

IQReport Documentation

Version 1.9

September 30, 2018

1 Contents

1	Installation and requirements	2
1.1	Required third party software	2
1.2	System requirements	2
1.3	Installation	2
1.4	Activation	2
1.5	Syntax highlighting of Report Markdown (RMD) language	2
2	Using IQReport (Example)	3
2.1	Example report as download	3
2.2	Compiling the example report	3
3	Syntax of '.rmd' files	3
3.1	Markdown	3
3.2	Mathematical expressions and Greek symbols	3
3.3	Title page	4
3.4	Approvals page	4
3.5	Removing the 'Intro'	4
3.6	Page break	4
3.7	Page orientation	4
3.8	Commented lines in an '.rmd' document	5
3.9	Un-numbered headings	5
3.10	Word style comments	5
3.11	URLs	5
3.12	Including external objects	5
3.12.1	Timestamp and compliance information	5
3.12.2	Including text files	5
3.12.3	Including figures and PDFs	6
3.12.4	Including tables	7
3.12.5	Including value tables	8

3.13	Cross-referencing elements	8
3.13.1	Cross-referencing Sections	8
3.13.2	Cross-referencing Figures, Text and Tables.....	9
3.13.3	Cross-referencing Table Cells and Value Tables	9
4	Template '.rmdt' File	10
5	IQReport R API.....	10
5.1	Installation of the IQReport R API	10
5.2	Where to find examples for the IQReport R API	10

1 Installation and requirements

IQReport can be downloaded from this webpage: <http://iqreport.intiquan.com>.

1.1 Required third party software

- Pandoc
- Ghostscript
- ImageMagick

1.2 System requirements

- Windows 64 bit or Linux 64 bit system
- Microsoft Word does not need to be present on the system on which IQReport is run. IQReport has been tested with Word 2010, 2013, and 2016.

1.3 Installation

Please follow instructions on the webpage <http://iqreport.intiquan.com>.

BOTH WINDOWS AND LINUX: Please locate the file "path_settings.txt" in the "IQReport/config" folder. Open it and check that the path settings are in-line with where you installed IQReport.

1.4 Activation

To run IQReport a license key is needed. It needs to be stored as file 'license.txt' in the 'config' folder. To obtain the license key, please contact IntiQuan at info@intiquan.com.

1.5 Syntax highlighting of Report Markdown (RMD) language

Using Notepad++ as a text editor, it is possible to syntax highlight the Report Markdown documents. A customized, portable NotePad++ version is provided for IQReport as download from <http://iqreport.intiquan.com>. In addition, this customized NotePad++ version allows to run IQReport and facilitates document writing by included Macros for recurrent text pieces. For Linux this is not available.

2 Using IQReport (Example)

The generation of a report is based on one main input file: the base document for the report in '.rmd' (Report Markdown) format. The desired Word style and other settings are provided by a template file that is stored in the 'IQReport/templates' folder. In the default web version, a single template file is provided 'DefaultStyle.rmdt'. Customized templates for desired corporate style and title pages can be provided on request. Additional files in different formats can be imported from within the Report Markdown document. Such files are figures in PNG/JPG/GIF format, text files, PDF files, and table files.

2.1 Example report as download

A detailed example is provided as a [zip file](#). Please download this zip file, unzip it and have a look at its contents. It provides the 'example.rmd' file and a folder 'material' in which elements are stored that are going to be included in the report (figures, tables, text files, PDF files). Additionally, it is shown how to include mathematical formulae.

2.2 Compiling the example report

Reports can be compiled using a command line interface (IQReport). This is explained here based on the example report:

- Obtain the example ([zip file](#)) and unzip it
- Open a command line interface in Windows or Linux (bash), change into the folder of the example
- Windows: execute the command: `IQReport.bat example.rmd`
- Linux: execute the command: `IQReport.sh example.rmd`
- A Word report is generated

It is also possible to run IQReport in R using the function 'IQReport' from IQRtools. For more information please refer to [IQRtools](#) and its documentation.

3 Syntax of '.rmd' files

The base document for the report is written in 'Report Markdown' (RMD). The file extension for 'Report Markdown' files is '.rmd', which is not to be confused with 'R Markdown' files which typically have the extension '.Rmd'.

3.1 Markdown

Report markdown is an extension of the [Markdown language](#). The Markdown syntax provides basic functionality to represent:

- Normal text
- Numbered section and subsection headers
- Itemized and numbered lists
- Certain font properties (bold, italic, etc.)

Please refer to the [Markdown cheat-sheet](#) for more information on this syntax.

3.2 Mathematical expressions and Greek symbols

Report markdown (RMD) supports mathematical expressions and subsequently Greek and other symbols. Mathematical expressions need to be written in the [LaTeX syntax](#). Inline mathematical expressions need to be enclosed in '\$' signs. 'Stand-alone' mathematical expressions need to be enclosed in '\$\$' signs. An example is shown below:

This is an inline equation: $y = \sum_{i=1}^n g(x_i)$

This is a 'stand-alone' equation:

$$y = \sum_{i=1}^n g(x_i)$$

3.3 Title page

There is a specific RMD command for the definition of the contents of a title page. For the default style it looks as follows (example as in the ZIP file):

```
=====
TITLE:      Example report illustrating functioning of IQReport
SUBTITLE:   Totally based on Fake Data
DATE:       dd-Mmm-yyyy
TEMPLATE:   DefaultStyle.rmdt
=====
```

This command has to appear as first element in an '*.rmd' file and is only allowed to appear once. Different template styles might have additional information in the title definition (e.g. Authors, Version, Status, Document Type, etc.). The common element (and the only one that is required is the TEMPLATE information. Thus, also the following title definition would be acceptable (TITLE, SUBTITLE, and DATE would be set to default values):

```
=====
TEMPLATE:   DefaultStyle.rmdt
=====
```

This latter example is also used in the case that the title page information is missing from the RMD file.

3.4 Approvals page

The RMD command `!APPROVALS` allows to add a custom approvals / signature page after the title page. In the default style this functionality is not included and the `!APPROVALS` command will be ignored. Custom built templates can include such a feature. The command can be located at any position in the '*.rmd' file after the title definition.

3.5 Removing the 'Intro'

In some cases, the user might not want a title page, table of contents, lists of tables and figures. If the RMD command `!NOINTRO` is located anywhere in the RMD document, then these 'Intro' information are removed from the Word file.

3.6 Page break

The command `!NEWPAGE` inserts a page-break into an RMD document. There have to be empty lines before and after this command.

3.7 Page orientation

The command `!LANDSCAPE` switches to landscape and the command `!PORTRAIT` to portrait page orientation. There has to be some text in these sections, otherwise the orientation switching will not work (e.g. with only an included table in it).

It is assumed that the first page is always in portrait mode.

3.8 Commented lines in an '.rmd' document

RMD documents can contain meta-information that is not desired to be included in the generated Word file. Examples are comments or sections of the document that should be kept in the '.rmd' document but are to be commented out in the Word file. Two different RMD commands are available:

- The character '%' used as the first character in a line will comment out the complete line.

3.9 Un-numbered headings

By default, all headings and sub-headings are numbered. It is sometimes useful to allow for un-numbered (sub)headings. This can be accomplished by the RMD command '#*' for a level 1 section and '##*' for a level 2 section, etc.

3.10 Word style comments

Word style comments can be added to an RMD document by the RMD command '!COM'. An example is the following:

```
!COM(This is a comment that will appear in Word on the side bar of a document)
```

These comments need to be on a separate line in the RMD file. They cannot be used as inline with text.

3.11 URLs

URLs can be included into an RMD document by the RMD command '@URL'. The syntax is as follows:

```
@URL[caption](link)
```

The caption is optional. In case the caption is not provided, the link will be used as caption. An example is the following:

```
@URL(http://www.google.com)
```

or

```
@URL[Search Page](http://www.google.com)
```

3.12 Including external objects

IQReport can process graphics in PDF, PNG, JPG, and GIF format. In addition, text files can be included (these text files will not be processed as RMD code but rendered in Word as present on the disk). Table import to Word is realized by importing text files that contain the desired table in a custom format. As most of our users use R to generate information to be reported, we provide an R library in the IQReport installation folder (see Section 5) that makes the generation of such table text files seamless. Any data.frame or R matrix can directly be exported into the required format.

3.12.1 Timestamp and compliance information

Using the IQRoutput* functions of the provided IQReport R API (or using IQRtools) with activated compliance mode leads to the generation of compliance log files. The information in the log files is used to trace back the time of generation, path to the source file, and script for the respective file. Whenever a log-file exists, the file-name, script-name and date of creation are included beneath the object in the Word file.

3.12.2 Including text files

Whole '.txt' documents can be included into the report using the command:

```
!TXT[optionalArguments](textFilePath)
```

Optional arguments: `caption`, `style`, `label`

Example:

```
!TXT[caption:My caption for the text - when label is defined then the  
caption is numbered,style:CODE(10), label:textlabel3](material/text.txt)
```

The specified caption is added above the text in the file found under 'material/text.txt' and the text itself is added using Word DOCX template `style CODE` and text size 10. The character string to be used for cross-referencing is specified as `label`. The user can define other Word styles to use, as long as these styles are defined in the Word document. The `label` is allowed to have a max length of 40 characters.

3.12.3 Including figures and PDFs

Figures in PNG, GIF, PDF, or JPG format can be included using the command:

```
!FIG[optionalArguments](figurePath){optional figure legend}
```

Optional Arguments: `ignoreCaption`, `crop`, `scale`, `caption`, `label`, `pages` (if multi-page PDF file).

- **ignoreCaption:true** will not produce a figure caption and a figure number (default: false)
- **crop:true** will crop whitespace around the figure (default: true)
- **scale:90** will scale the figure to x% of the page width (default: 100)
- **caption:captionText** Caption text for the figure. "caption:" identifier can be omitted if caption provided as first argument. (default: default caption text)
- **label:figLabel** defines a label that can be used for cross-referencing (default: no label)
- **pages:1 2 3 6-** providing page numbers to include as figures under same caption. Only works for multi-page PDFs.

If several !FIG commands come after each other at least an empty line needs to be between these elements in the RMD file.

Example:

```
!FIG[caption: My chosen caption ,crop:true, scale:75, crop:true,  
label:Figurelabel3], (material/fig1.png){ This is a legend }
```

Inclusion of the plot 'fig.png' with a specified caption. With the option 'crop:true' the blank space around the figure is removed and 'scale:75' then sets the width of the figure to 75% of the Word document width. The legend written in curly brackets is added under the figure.

3.12.3.1 Including multi-page PDF documents

The inclusion of PDFs is possible using two different methods.

- `!FIG[pages:1 3 5-7,scale:30,crop:true](material/fig2.pdf)`
- `!PDF[pages:1 3 5-7,repeatLegend:true](material/fig2.pdf)){ legend under every figure }`

With the optional argument 'pages' specifying the chosen pages for inclusion in the .pdf document, in this case page 1,3,5,6, and 7. If the legend is to be shown under every included pdf page then the option 'repeatLegend' should be set to true. As of now it is not possible to place figures next to each other - which can, however, be handled when generating the figure files and thus is not a limitation. The other optional arguments are to be used as specified under the inclusion of figures.

If several !PDF commands come after each other at least an empty line needs to be between these elements in the RMD file.

3.12.4 Including tables

The inclusion of tables requires the tables to be in a specific format saved in a .txt file. An example for the contents of such a table text file is the following:

```
<TT> This is the table title
-----
<TH> Characteristics | Category1 | Category2
-----
<TR> Variable 1 | 68 (13.8), [21-100]* | 67 (11.2), [39-97]*
<TR> Variable 2 | 59.5 (14.5), [31.4-99.8] | 63.3 (16.5), [27.7-121]
<TR> Variable 3 | 1.61 (0.121), [1.31-2.01] | 1.65 (0.147), [1.27-2.02]
<TR> Variable 4 | 23.6 (5.67), [12.5-42.8] | 23.7 (5.54), [8.9-47.1]
<TR> Variable 5 | 1.64 (0.224), [1.1-2.21] | 1.7 (0.26), [1.1-2.52]
-----
<TF> *Median (Standard deviation), [Minimum-Maximum]
```

The table content is specified using the row names

- <TT> for the table title (optional, but should then be specified later),
- <TH> for the table row containing column headers (required),
- <TR> for table rows (required) and
- <TF> for the table footer (optional).

In the example above the lines without these tags do not serve any purpose for parsing the table files.

If the IQReport R API (see Section 5) or IQRtools is used, data frames and matrices can be exported to the explained format using the function 'IQRoutputTable'.

The table can then be included using the command:

```
!TAB[optionalArguments](tableFilePath)
```

With the optional arguments: caption, label, style

Example:

```
!TAB[caption:This is the table title ,style:TABLE(8),label:tableref1]
(material/table.txt)
```

In this case the caption specified in the .txt file is overwritten. The table is included using Word DOCX template style TABLE and font size 8. The label tableref1 might be used for cross-referencing. Other table styles can be used but need to be defined in the Word template.

Line breaks in table cells

Line breaks cannot be coded in arbitrary table cells. This is a limitation from markdown and Pandoc at the moment. What can be done is to define line breaks in the table footer text (following the <TF> tag). Line breaks here can be coded by either the '
' tag.

3.12.5 Including value tables

Value tables allow the referencing of values in the table in text passages. To do so a value table requires the presence of a "value" and a "label" column. The overall set-up of value tables is identical to normal tables.

```
<TT>  Caption of my value table
=====

<TH>  Name                | value                | label
-----|-----|-----
<TR>  Age (years)          | 1234                 | AGE
<TR>  Weight (kg)         | 5678                 | WEIGHT
<TR>  Height (cm)         | 9101112              | HEIGHT
<TR>  Body Mass Index (kg/m2) | 13141516            | BMI
<TR>  Body Surface Area (m2) | 17181920            | BSA
-----|-----|-----
<TF>  *Median (Standard deviation), [Minimum-Maximum]
```

A value table is specified by setting the option `valueTable` to true:

```
!TAB[valueTable:true, label:valuetable1](material/valuetable.txt).
```

Optional Arguments include: `caption`, `label` and `style` and are specified as described under section 3.12.4.

In case the user would like to override the default names for the value and label columns in valueTables, this can be done by the optional arguments: `VTlabel` and `VTvalue`.

Example

```
!TAB[valueTable:true, label:valuetable1, VTlabel:newNameLabelColumn,
VTvalue:newNameValueColumn](material/valuetable.txt).
```

3.13 Cross-referencing elements

Cross-referencing is used to improve readability of the report. All cross-referenced objects need to have a specified label.

3.13.1 Cross-referencing Sections

Sections can be cross-referenced if a label was introduced for the specific section before as shown below:

```
# Section header {#sectionlabel}
```

Later referencing is then executed with the following commands:

```
@SEC(sectionlabel)
```

```
@SEC[my customized caption](sectionlabel)
```

If no caption is provided 'Section x.y' will be shown in the text. If a caption is provided, this caption will be shown. The cross-reference is hyper-linked.

3.13.2 Cross-referencing Figures, Text and Tables

Figures, Texts and tables are all cross-referenced using the same schemata:

- For text : @TXT[caption](label)
- For figures: @FIG[caption](label)
- For tables : @TAB[caption](label)

Caption is optional and specifies alternative text to show instead of the automatically assigned caption with number (e.g. Text 1, Figure 1, Table 1...). As label, the label assigned to the respective object at inclusion need to be used.

Example:

```
!FIG[caption: My chosen  
caption ,crop:true,scale:75,crop:true,label:Figurelabel3],  
(material/fig1.png){ this is a typical legend }
```

Cross-Reference:

```
@FIG(Figurelabel3)
```

Or

```
@FIG[see this cool figure](Figurelabel3)
```

3.13.3 Cross-referencing Table Cells and Value Tables

While writing the Report it might be necessary to include content of table cells of tables and value tables in-line. Table content is referenced by stating the row and column of the respective cell. However, if the content changes the reference is not updated. In this case value tables are a better choice. Here, the label of the respective value is referenced. With changing table content the inline reference is updated.

- For tables: @VAL[row*,column](tablelabel) *header rows are not counted
- For value tables: @VAL[label of value to be referenced](label of value table)

The following examples refer to the table and valueTable shown under the inclusion of tables and value tables (see sections 3.12.4 and 3.12.5):

```
This is the value found for Category1 and Variable2 in the text:  
@VAL[2,2](tableref1) .
```

→ This is the value found for Category1 and Variable2 in the text: **59.5 (14.5), (34.4-99.8)**.

The value @VAL[AGE](valuetable1) corresponds to the label AGE in the referenced value table cell.

→ The value **1234** corresponds to the label AGE in the referenced value table cell.

4 Template '.rmdt' File

'*.rmdt' template files are basically zip files that contain a template Word DOCX document and an additional settings.txt document. Using these two files it is possible to realize almost any corporate requirements on Word Style, Title Page format, and signature pages. An IQReport installation can handle an unlimited number of different template files. The name of the desired template is simply defined in the Title definition in an RMD document.

Users with specific formatting requests are welcome to contact IntiQuan at info@intiquan.com for support.

5 IQReport R API

The installation folder of IQReport contains an R package that serves the following purposes:

- It allows to construct RMD files programmatically directly from R.
- It allows to run IQReport on RMD files.
- It allows to export tables (data.frame or matrix) into the IQReport table text format.
- It allows to generate compliance log information for each output that has been exported using the IQRoutput* functions.

5.1 Installation of the IQReport R API

The R source package of "IQReportAPI" is located in the "IQReport/Rlibrary" folder. The current version is: IQReportAPI_0.1.0.tar.gz. For installation, open R, change working directory to the "IQRtools/Rlibrary" folder and execute:

```
install.packages("IQReportAPI_0.1.0.tar.gz", repos = NULL, type = "source")
```

5.2 Where to find examples for the IQReport R API

In the installation folder of the IQReportAPI package you find a folder "examples". In this folder, 3 detailed examples are provided that can get you started using the R API.