DocGen - a documentation tool

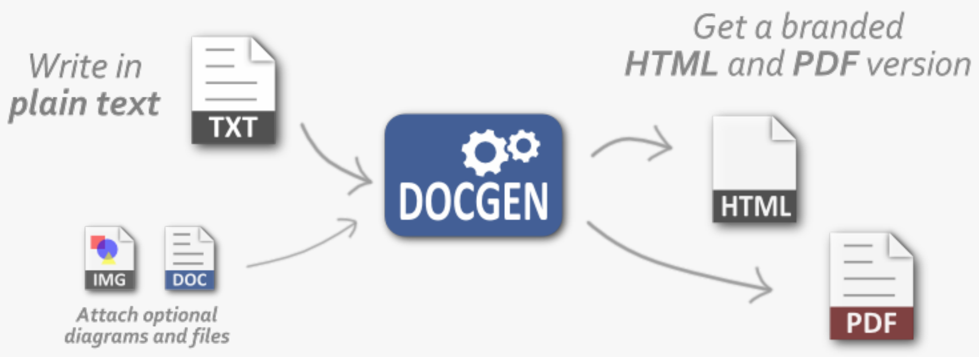
Owner: Mark Macdonald	**Version:** 2.0.0	**Released:** XX/XX/XXXX
Author: Mark Macdonald	**Contributors:**	
Module:	**ID:**	**Link:** Github

Summary

DocGen is a documentation tool for software products. It's a static website generator that takes plain text or Markdown as input, and outputs both a static website and a PDF copy.

Table of Contents

Better documentation for software projects.



Write in plain text
TXT
IMG
DOC
Attach optional diagrams and files
DOCGEN
Get a branded HTML and PDF version
HTML
PDF

Download

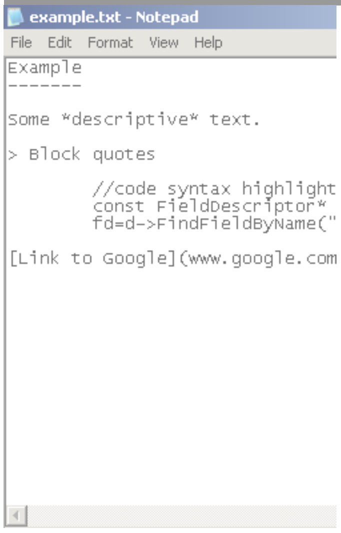
Report Issues

How it works

"You write in plain text or human-friendly Markdown"

"DocGen styles and publishes all your content as a website"

"DocGen also creates an equivalent PDF copy"



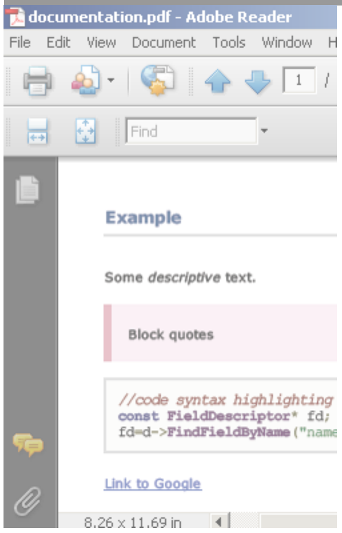
```
Example
-------

Some *descriptive* text.

> Block quotes

        //code syntax highlight
        const FieldDescriptor*
        fd=d->FindFieldByName("

[Link to Google](www.google.com
```



documentation.pdf - Adobe Reader
File Edit View Document Tools Window H
1
Find
8.26 x 11.69 in

Example

Some *descriptive* text.

Block quotes

```
//code syntax highlighting
const FieldDescriptor* fd;
fd=d->FindFieldByName("name
```

Link to Google

Features

DocGen makes it easy for anyone to make high-quality technical documentation. It is particularly intended for software library 'user guides' and tutorials which need to be versioned **in sync** with the software. It may be useful in other situations where versioned documentation needs to be made accessible, such as process guides or technical product manuals.

Input Formats

- Plain text
- Markdown
- HTML
- Image diagrams
- Attach other documents

Configurable Generator

- Code syntax highlighting
- Automatic tables of contents
- Branding and legal markings
- Ownership and versioning
- Change log feature

Flexible output formats

- Self-contained HTML website
- Publish to any web server
- Publish to CD, shared drive, local folder
- Author next to code in Subversion
- Automatic PDF version

> DocGen is probably not the right tool for precision PDF layout, or repeated generation of similar documents from standard templates (e.g. in product configurators). It is meant for free-form, human generated content which is regularly updated and refactored.

Support for multiple platforms

DocGen can be used to publish documentation on **Microsoft Windows**, **Mac OS X**, and **Linux**.

Content published by DocGen is tested against these browsers:

- Internet Explorer 7+
- Firefox
- Safari
- Chrome

Installing DocGen

Overview

- Download the **latest released version** and save it anywhere, e.g. the Desktop.

> The first time DocGen is used some dependencies (listed below) must be installed. For help, see the setup instructions on this page.

- Ruby **1.9.x or newer**
- Kramdown RubyGem **0.10.0 or newer**
- Nokogiri RubyGem **1.4.4.1 or newer**
- Kwalify RubyGem **0.7.2 or newer**
- Systemu RubyGem **2.2.0 or newer**
- wkhtmltopdf **0.9.9 or newer**

Setup - Microsoft Windows

Download and run the following installers:

- Windows Ruby installer (select ***'Add Ruby executables to your PATH'*** and ***'Associate .rb and .rbw files with this Ruby installation'***)
- Windows wkhtmltopdf installer (select ***'Modify PATH'***)

Open the Windows Command Prompt and enter the following commands. (See help with proxy settings in case of network errors).

```
gem install kramdown
gem install nokogiri
gem install kwalify
gem install systemu
```

> The latest version of systemu is only compatible with Ruby 1.9.3 on Windows. For older Ruby versions, install systemu 2.2.0 by instead entering: **gem install systemu --version '=2.2.0'**

Confirm correct wkhtmltopdf installation by entering the following. (See help with wkhtmltopdf path in case of 'not recognised' errors).

```
wkhtmltopdf --version
```

Setup - Linux

Normally a suitable version of Ruby is already installed. Check this by opening a terminal and entering:

```
ruby -v
```

If Ruby is not installed or the version is too old, follow the installing or upgrading Ruby help.

Then, in the terminal, enter the following commands. (See help with proxy settings in case of network errors).

```
sudo gem install kramdown
sudo gem install kwalify
sudo gem install systemu
sudo apt-get install libxslt-dev libxml2-dev
sudo gem install nokogiri
```

Download the wkhtmltopdf Linux static binary to your desktop. If needed, rename it to **wkhtmltopdf** (i.e. remove version suffix). In the terminal, enter:

```
sudo cp Desktop/wkhtmltopdf /usr/local/bin
sudo chmod 755 /usr/local/bin/wkhtmltopdf
```

Setup - Apple OS X

OS X already has Ruby, but you probably need to upgrade the Ruby version. To check the version, open a terminal and enter:

```
ruby -v
```

After upgrading, enter the following commands in the terminal. (See help with proxy settings in case of network errors).

```
sudo gem install kramdown
sudo gem install nokogiri
sudo gem install kwalify
sudo gem install systemu
```

Download the wkhtmltopdf OS X static binary to your desktop. If needed, rename it to **wkhtmltopdf** (i.e. remove version suffix). In the terminal, enter:

```
sudo cp Desktop/wkhtmltopdf /usr/local/bin
sudo chmod 755 /usr/local/bin/wkhtmltopdf
```

OS X additionally needs to have **/usr/local/bin** added to the PATH. In the terminal, enter:

```
pico ~./bash_profile
```

Add the following line to the file. Save it and exit pico using the on-screen instructions (i.e. CTRL+O then CTRL+X)

```
export PATH=/usr/local/bin:/usr/local/sbin:$PATH
```

In ther terminal, enter:

```
source ~/.bash_profile
```

Test the installation

Run DocGen to test the installation. Use the Command Prompt on Windows or the terminal on Linux / OS X and enter (substituting the correct path):

```
cd /path/to/DocGen/Generator
ruby Generator.rb
```

Now open **documentation.html** in the DocGen Output folder. If everything is working, you will see an empty template website. If there were problems, see troubleshooting (below).

Troubleshooting

Help with proxy settings when installing Ruby gems

When behind a proxy, add the option **-p proxy_url**. For example:

```
gem install kramdown -p http://<IP>:8080
```

Alternatively, **download the gems** manually and install them from a local directory (using the normal instructions above).

Help with wkhtmltopdf path issues (Windows only)

The wkhtmltopdf 0.9.9 installer does not have an option to add wkhtmltopdf to the PATH environment variable. Instead:

- either use wkhtmltopdf 0.10.0 or later (recommended)
- or manually add wkhtmltopdf.exe to the PATH environment variable: (My Computer | Properties | Advanced | Environment Variables).

Missing Nokogiri dependencies (Linux and OS X)

The Nokogiri RubyGem requires libxml2 and libxslt on Linux and OS X. Usually these are already present. If not, follow:

- detailed instructions for installing nokogiri

Installing or upgrading Ruby (Linux and OS X)

These instructions install a fresh copy of the latest stable Ruby and make it the default interpreter. Existing vendor Ruby installations are unaffected.

- download and unzip the Ruby source code
- (OS X only) install XCode if not already present

To make and install Ruby, open a terminal and enter (substituting the correct path):

```
cd /path/to/ruby-download
autoconf
./configure --prefix=/usr/local --enable-shared
make
sudo make install
```

Extra help for Debian Linux

Make sure these packages are installed ***before*** configuring Ruby:

- build-essential
- libssl-dev
- libreadline5
- libreadline5-dev
- zlib1g
- zlib1g-dev

How to create content with DocGen

Overview

Using DocGen involves working with three folders:

- **Source** (the document master, where all your content goes)
- **Generator** (the only task you perform here is to run **Generator.rb** when publishing the content)
- **Output** (where the generated HTML website and PDF will appear)

'Source' and 'Generator' are included in the DocGen download, and 'Output' will appear the first time you run Generator.rb.

Writing content

DocGen takes any text file (extension .txt) in the Source folder and converts it. Each text file will be a separate page in the website and a separate chapter in the PDF. You can create as many new files as you wish (start by editing the example file called index.txt).

> Make sure you save content files (.txt) with **UTF-8 encoding**. This makes non-standard characters (ø © é etc) work.

You mostly write paragraphs of plain text using any text editor. Stylistic effects can be achieved by using a plain-text publishing format called Markdown. The best way to learn is to see an example (below), and then read the Markdown reference.

What you type:

```
Headings give structure
-----------------------

Paragraphs are big blocks of text with lots of sentences.
Some sentences have *significant* concepts or **very important ones**.

A new paragraph starts when the line before looks blank.
Some paragraphs will contain example links.

    //To make a code block, just indent with a tab. "Hello World" in Ruby:
    5.times { puts "Hello!" }
```

What it looks like in the website and the PDF:

Headings give structure

Paragraphs are big blocks of text with lots of sentences. Some sentences have ***significant*** concepts or **very important ones** .

A new paragraph starts when the line before looks blank. Some paragraphs will contain example links.

```
#To make a code block, just indent with a tab. "Hello World" in Ruby:
5.times { puts "Hello!" }
```

Adding diagrams and other files

- Any diagrams (in image form, e.g. JPEG, PNG, GIF etc) should be put the Source/Images folder.
- Any other files you want to attach should go into Source/Docs.

See the Markdown reference for how to attach them. Images are embedded directly into the website and PDF, but files are only attached as links.

Markdown Reference

Markdown is a human-friendly, plain text publishing format (the text files are easily readable without being obfuscated by markup tags). The most useful Markdown features are shown below. Some specialist topics are explained on the advanced content page. See also the original Markdown specification and the documentation for the kramdown engine used by DocGen.

Note: a page will only appear in the PDF table of contents if it has a **page heading** (see below).

Style	What you type	What it looks like
Page Heading	`Page Heading` `============`	Page heading
Section Heading	`Section Heading` `---------------`	Section Heading
Minor Heading	`###Minor Heading`	Minor Heading
False Heading (not in PDF contents list)	`<p class="false-heading">False</p>`	False
Emphasis (italic)	`This text has` *`emphasis`*	This text has ***emphasis***
Strong (bold)	`This text is` **`bold`**	This text is **bold**
Block quotes	`> This is a block quote.`	This is a block quote
Code block (indent with tab)	`    int some_code=0`	`int some_code=0`
Unordered List	*`unordered list`* `(items)`	• unordered list • (items)
Ordered List	`1. ordered list` `2. (items)`	1. ordered list 2. (items)
External Links	link to Google	link to Google
Links to a local file	user guide	user guide
Diagrams (embedding images)		
Simple table	`| Cell 1 | Cell 2 | Cell 3 |` `| Cell 4 | Cell 5 | Cell 6 |`	Cell 1 Cell 2 Cell 3 Cell 4 Cell 5 Cell 6

Writing Advanced Content

Overview

Any standard **HTML** can be entered in a content (.txt) file if needed (useful in cases where Markdown features are too simple).

Creating internal links to page sections

To create links to other sections within one content page, put hyphens between the words in the heading and prepend with #:

```
[link to heading](#this-is-a-heading)
... other page content here ...
This is a heading
-----------------
```

Creating internal links to other pages

To create links to *other* content pages provide the relative url to the page:

```
[link to heading](example-page.html)
```

Control of page breaks in the PDF

DocGen does not provide precision control over PDF layout. However, some steps can be taken in case of page break issues (the most common problem).

To ***force*** a page break, insert the following before the item you want to appear on a new page:

```
<span class="force-break"></span>
```

DocGen already tries to eliminate unsightly page breaks ***inside*** code blocks, block quotes, and table rows. To apply the same technique to other elements, revert to HTML and apply the **avoid-break** class. For example:

```
<p class="avoid-break">A long paragraph</p>
```

Basic Tables

An extended Markdown syntax allows basic tables to be created. Basic tables:

- have fixed width
- do not allow control of column widths
- do not provide for custom styling

What you type

```
| Heading     | Heading           | Heading |
|:-----------|:----------------:|--------:|
| Something   | Some commentary   | ?       |
| Another     | More commentary   | ?       |
| One more    | More commentary   | ?       |
```

What it looks like

Heading	Heading	Heading
Something	Some commentary	?
Another	More commentary	?
One more	More commentary	?

Custom Tables

For better control of table styling, revert to HTML. Either embed custom CSS styles in the content page (total control), or use the DocGen built-in style called **styled-table**. With **styled-table** it is possible to:

- adjust the table width and set the column widths exactly

- have zebra (alternate shaded) rows and bold columns
- have red or green text in cells, or ticks and crosses

What you type

```
<table class="styled-table" style="width:100%;">

 <tr>
  <th style="width:200px;">Heading</th>
  <th style="width:500px;">Heading</th>
  <th>Heading</th>
 </tr>

 <tr>
  <td class="bold">Example</td>
  <td>Example</td>
  <td>Example</td>
 </tr>

 <tr class="shaded">
  <td class="bold">Example</td>
  <td>Example</td>
  <td></td>
 </tr>

 <tr>
  <td class="bold">Example</td>
  <td class="green">Example</td>
  <td class="tick"></td>
 </tr>

 <tr class="shaded">
  <td class="bold">Example</td>
  <td class="red">Example</td>
  <td class="cross"></td>
 </tr>

</table>
```

What it looks like

Heading	Heading	Heading
Example	Example	Example
Example	Example	
Example	Example	
Example	Example	

How to publish output with DocGen

Overview

You publish content by running **Generator.rb**. This

- **creates a static website** (each text file in Source is converter to an HTML file in Output, e.g. content.txt becomes content.html)
- **converts the whole website into a single PDF**

Before running the Generator

Update the configuration files

These are in the 'Source' folder.

- ***doc-parameters.yml*** sets the attributes that will appear inside the document. Most only need to entered the first time you create a new document, **but you should make sure to always update the release field when releasing a new product version**
- ***tool-behaviours.yml*** allows how DocGen operates to be customised (e.g. to modify the path to 'Output' folder)

> All config files (.yml) **must** be saved with UTF-8 encoding (their default format).

Complete table-of-contents.yml

This is also in the 'Source' folder. It sets:

1. what to call each file in the HTML table of contents and which column it should appear in.
2. which order the chapters in the PDF will be.

Note that the table of contents in the PDF is based on the page headings in your text files. The only bearing that 'table-of-contents.yml' has on the PDF is to decide the order of the chapters. You can include links to other documents or external websites in the table of contents, but they won't appear anywhere in the PDF copy.

Update change_log.txt

If you are releasing a new version of a product, you should summarise the changes for this version and add it to the top of the page.

Run the Generator

Use the Command Prompt on Windows or the terminal on Linux / OS X and enter (substituting the correct path):

```
cd /path/to/DocGen/Generator
ruby Generator.rb
```

Advanced Usage

DocGen looks for the 'Source' files in this order:

- the sourcePath option (e.g. run `ruby Generator --sourcePath "C:/Example Source/"` noting forward slashes and quote marks)
- the working directory (only if it contains doc-parameters.yml)
- **default**: ../Source (relative to the working directory)

The DocGen Output location can be changed in tool-behaviours.yml

Review the generated content

Open **Output/documentation.html** and check the website and PDF are as desired.

Deploy your documentation

The self contained website can be placed on any web server, or used locally from a computer folder.

Troubleshooting

Why is the PDF font size or layout different when generated on Windows vs. Linux?

DocGen does not guarantee that the PDF will be precisely identical when generated on different platforms. If this matters, choose one platform and use it consistently.

Why does Chrome PDF Viewer open or print the PDF incorrectly?

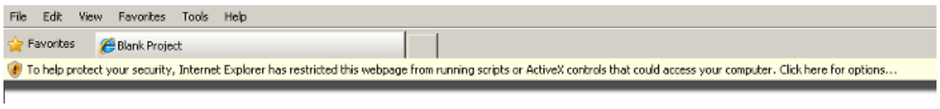
These are known issues. Only **Adobe Reader** is officially supported at this time.

Why do the links not work in the PDF?

This feature is unsupported in **wkhtmltopdf 0.9.9** but will be supported in version **0.10.0**.

Why is there an "Internet Explorer has restricted this webpage" warning?

It is a known issue that Internet Explorer prevents pages opened from a local folder from running scripts. The following warning may be seen:



To help protect your security, Internet Explorer has restricted this webpage from running scripts or ActiveX controls that could access your computer. Click here for options...

There are two solutions:

- view the pages from any web server (simply upload them)
- disable scipts in tool-behaviours.yml by setting `suppress_IE_warning: true`. This is not recommended for production use and will also disable the code syntax highlighting in Internet Explorer.

Other issues

For any other problems, please submit a [an issue ticket] with the following information:

- Describe the problem
- Describe the steps needed to reproduce the problem
- Attach any sample files needed to reproduce the problem

> Please always attach **log.txt** (in the Generator folder) when requesting support.

How to use DocGen with Subversion

Recommended practice

When using DocGen with Subversion, the following practice is recommended (see the setup steps for help)

- **Add** the **Source** directory and its files directly to your repository (e.g. in any suitable docs directory)
- **Ignore** the **Output** directory, using Subversion properties
- Use **Subversion externals** to reference a specific release of the DocGen **Generator** and **User Guide** directories
- **Ignore log.txt** in the **Generator** directory, using Subversion properties

> Do not try to directly version the Output Folder (DocGen overwrites it). If you want to also version control the generated content files (not necessary), you must move them from the **Output** directory into another suitable folder, taking care to manually perform Subversion commands like file renames, deletions, or additions.

Setup - using DocGen in a Subversion project

i) Create a directory in your Subversion repository for DocGen documentation and perform **svn add**

ii) Apply the following Subversion property to the directory (e.g. on Windows, right click and choose **TortoiseSVN | Properties**): property - **svn:ignore**, value - **Output**

iii) Similarly, create the **svn:externals** property and set its value as follows, substituting X and A.B.C for the the desired release version from the DocGen repository tags directory:

```
^/../svnccpcommonplatformtoolbox/tags/DocGen/releases/Xx/A.B.C/Generator/ Generator
^/../svnccpcommonplatformtoolbox/tags/DocGen/releases/Xx/A.B.C/DocGen/User%20Guide/ "Generator
```

iv) Create a child directory called **Source**, copy in the desired content files and perform **svn add**

v) Perform **svn update**

vi) Apply the following Subversion properties to the **Generator** directory: property - **svn:ignore** value - **log.txt**

How to upgrade to new versions of DocGen

If you want to upgrade to a newer version of DocGen but still use your existing content from a previous project, this is trivial:

1. Download the latest version of DocGen
2. Delete the Source folder
3. Replace with your existing Source folder

Sometimes new features will require you to update the configuration files to match those expected by the latest version. In this case look at the reference example in the DocGen Source folder.

> You should check the styling in the latest version still works well with your old content.