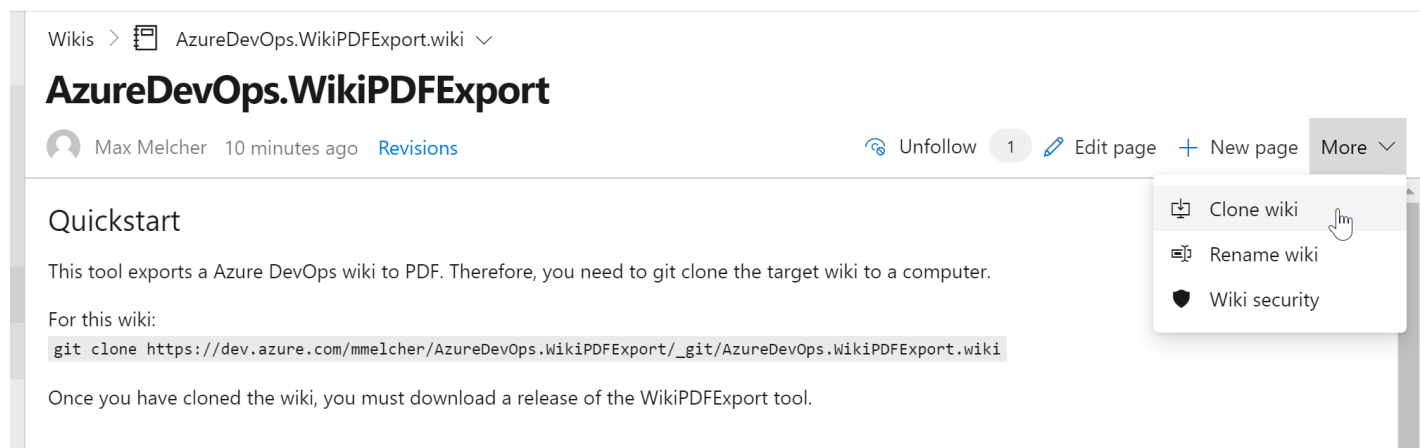


AzureDevOps.WikiPDFExport

Quickstart

This tool exports a Azure DevOps wiki as PDF. Therefore, you need to git clone the target wiki to a computer. You can get the clone link of the wiki in the top right of the wiki homepage:



Wikis > AzureDevOps.WikiPDFExport.wiki

AzureDevOps.WikiPDFExport

Max Melcher 10 minutes ago [Revisions](#)

Unfollow 1 Edit page + New page More

Clone wiki

Rename wiki

Wiki security

Quickstart

This tool exports a Azure DevOps wiki to PDF. Therefore, you need to git clone the target wiki to a computer.

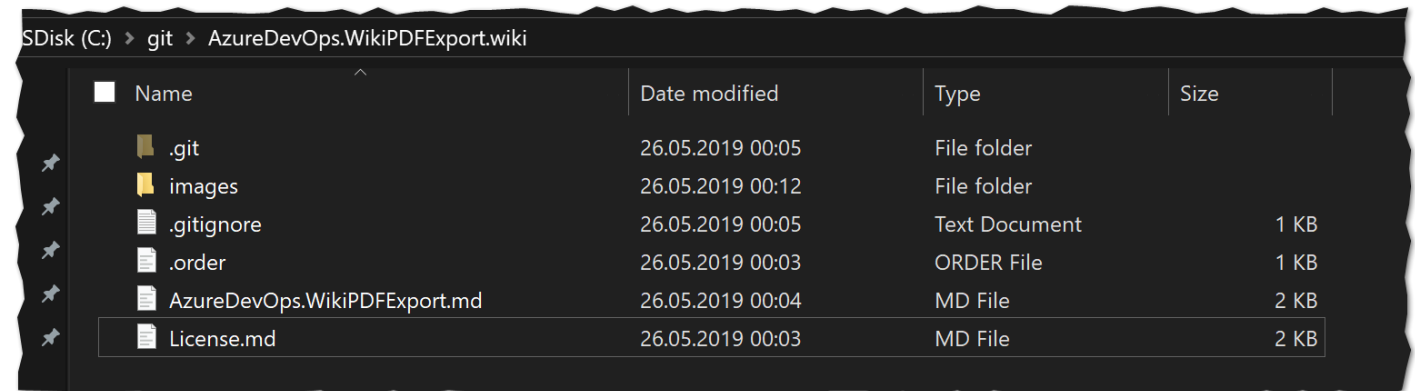
For this wiki:

```
git clone https://dev.azure.com/mmelcher/AzureDevOps.WikiPDFExport/_git/AzureDevOps.WikiPDFExport.wiki
```

Once you have cloned the wiki, you must download a release of the WikiPDFExport tool.

To clone this wiki, use the following command: `git clone https://dev.azure.com/mmelcher/AzureDevOps.WikiPDFExport/_git/AzureDevOps.WikiPDFExport.wiki`

The result should look like this:



Name	Date modified	Type	Size
.git	26.05.2019 00:05	File folder	
images	26.05.2019 00:12	File folder	
.gitignore	26.05.2019 00:05	Text Document	1 KB
.order	26.05.2019 00:03	ORDER File	1 KB
AzureDevOps.WikiPDFExport.md	26.05.2019 00:04	MD File	2 KB
License.md	26.05.2019 00:03	MD File	2 KB

Once you have cloned the wiki, you must download the Azure DevOps WikiPDFExport tool. [azuredevops-export-wiki.exe](#) (~40MB)

You can drop it right into the cloned folder and execute it there. Launched without parameters, the tool will detect all wiki files next to it and convert it to a PDF called export.pdf right next to it. Similar to this [pdf](#).

If you need more control over the output, please see the Configuration Options below or by launching the tool with --help parameter.

Features

The tool currently supports the following:

- Export all wiki pages (and sub pages) in the correct order including styles and formatting.
- Includes pictures (remote and relative urls)
- Creates PDF bookmarks to all pages for easier navigation within the PDF
- If you link to other wiki pages, the link in the PDF will work, too.
- Everything self-contained. Download the .exe file, run it, done.
- Tool can be used as part of a build, see [BuildTask](#)
- Supports emoticons 😊 ⚠️ ⓘ
- It is fast. A PDF with 160 pages is created in less than a second. 1000 pages in ~8 seconds.

Requirements

The tool is developed as .NET 5 application, therefore you need to have the runtime installed. Download is available [here](#). Currently it requires a x64 runtime.

Download

The download is available [here](#)

⚙ Configuration Options

--help

Help - outputs all the flags/parameters

-o / --output

The path to the export file including the filename, e.g. c:\export.pdf

-b / --breakPage

For every wiki page a new page in the PDF will be created

-h / --heading

For every wiki page create a heading in the PDF. If the file is called Home.md a new #Home-heading is added to PDF.

-s / --single

Path to a single markdown file to convert to PDF. If you want to write your changelog in the wiki, this is your parameter to only convert a single page. -p parameter is required, too.

-p / --path

Path to the wiki folder. If not provided, the current folder of the executable is used.

If you only want to convert a subfolder and have images, then you must provide the path to the attachments folder with --attachments-path.

--attachments-path

Path to the .attachments folder. If not provided, the .attachments is assumed to be located under the folder of the wiki (-p/--path).

-m / --mermaid

Convert mermaid diagrams to SVG. Will download latest chromium, if chrome-path is not defined.

--mermaidjs-path

Path of the mermaid.js file. It'll be used if mermaid diagrams support is turned on (-m/--mermaid). If not specified, 'https://cdn.jsdelivr.net/npm/mermaid/8.6.4/mermaid.min.js' will be downloaded.

--chrome-path

Path of the chrome or chromium executable. It'll be used if mermaid diagrams support is turned on (-m/--mermaid). If not specified, a headless version will be downloaded.

--css

Path to the stylesheet to overwrite the look of certain components in the PDF. See [styles.css](#) for examples. To get the html file, use the [--debug flag](#) to inspect and style it.

-v / --verbose

Verbose mode. Logging will added to the console window

-d / --debug

Debug mode. Logs tons of stuff and even exports the intermediate html file

--pathToHeading

Add path of the file to the header

--footer-left, --footer-center, --footer-right, --header-left, --header-center, --header-right,

Headers and footers can be added to the document by the --header-* and --footer* arguments respectfully. In header and footer text string supplied to e.g. --header-left, the following variables will be substituted.

- [page] Replaced by the number of the pages currently being printed
- [frompage] Replaced by the number of the first page to be printed
- [topage] Replaced by the number of the last page to be printed
- [webpage] Replaced by the URL of the page being printed

- [section] Replaced by the name of the current section
- [subsection] Replaced by the name of the current subsection
- [date] Replaced by the current date in system local format
- [isodate] Replaced by the current date in ISO 8601 extended format
- [time] Replaced by the current time in system local format
- [title] Replaced by the title of the of the current page object
- [doctitle] Replaced by the title of the output document
- [sitepage] Replaced by the number of the page in the current site being converted
- [sitepages] Replaced by the number of pages in the current site being converted

--disableTelemetry

Disables the telemetry tracking, see [Telemetry](#)

--open

Opens the PFD file after conversion. Great for development, not great in a build task.

--filter

Filters the pages depending on the page [yaml tags](#).

Limitations

So far the following limitations are known:

- TOC (Table of Contents) tag is not supported and removed from the pdf.
- The tool, sometimes shows an error "Qt: Could not initialize OLE (error 80010106)" - this can be ignored.
- If headers are not formatted properly (#Header instead of # Header), they are rendered incorrectly. I might fix that in the future.
- The tool lacks proper testing because I only have two realistic wikis available. Want to contribute one?

License

See [license](#)

Telemetry

The tool uses Application Insights for basic telemetry:

- The duration of the export and the count of wiki pages is tracked and submitted to Azure.
- In the case of an error, the exception is submitted.
- No wiki data/content is submitted.

Thanks

In this tool uses open source libraries that do the actual work - I just combined them to get the export as PDF:

1. [CommandLineParser](#) to parse the command line
2. [MarkDig](#) to parse markdown files to HTML.
3. [DinkToPdf](#) to export HTML to PDF
4. [dotnet-warp](#) to release a self-contained exe file
5. [puppeteer-sharp](#) to convert mermaid markdown to SVG

License

MIT License

Copyright (c) 2019 Max Melcher

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Build Task

Using Azure DevOps WikiPDFExport as build task is straightforward.

1. Create a new build definition
2. Git Source is "Other Git"
3.
 1. Add the clone url to the wiki to the details and username / password if required
4. Ensure that the agent is a windows agent
5. Add a powershell task with the following code:

```
#Download url to the export tool
$url = "https://dev.azure.com/mmelcher/8036eca1-fd9e-4c0f-8bef-646b32fdba0b/_apis/git/repositories/e08d1ada-7794-4b89-a3ea-cb64a26683c3/Items?path=/"

#filename of the tool
$output = "azuredevops-export-wiki.exe"

#download the file
Invoke-WebRequest -Uri $url -OutFile $output

#launch the tool - adjust the parameters if required
./azuredevops-export-wiki.exe
```

5. Add a second task to publish the PDF as build artifact.

Once the build succeeds, you can download the PDF file from the build page or use it in a release.

Pictures

Windows Agent

The screenshot shows the 'Agent job 1' configuration in the Azure DevOps Build pipeline. The pipeline is named 'Build pipeline' and contains three tasks: 'Get sources' (repository: wikiMaster), 'PowerShell Script' (PowerShell), and 'Publish Artifact: drop' (Publish Build Artifacts). The 'Agent job 1' is selected, and its configuration is shown on the right. The 'Display name' is 'Agent job 1', the 'Agent selection' is 'Hosted VS2017', and the 'Demands' are empty.

PowerShell Task

The screenshot shows the 'PowerShell Script' task configuration in the Azure DevOps Build pipeline. The task is selected, and its configuration is shown on the right. The 'Task version' is '2.*', the 'Display name' is 'PowerShell Script', the 'Type' is 'Inline', the 'Script' content is as follows:

```
$url = "https://dev.azure.com/mmelcher/8036eca1-fd9e-4c0f-8bef-646b32fdba0b/_apis/git/repositories/e08d1ada-7794-4b89-a3ea-cb64a26683c3/Items?path=/"
$output = "azuredevops-export-wiki.exe"
Invoke-WebRequest -Uri $url -OutFile $output
./azuredevops-export-wiki.exe
```

The 'ErrorActionPreference' is set to 'Stop'.

Publish Task

TasksVariablesTriggersOptionsRetentionHistory

Save & queueDiscardSummaryQueue

PipelineBuild pipeline

Get sourcesrepositorywikiMaster

Agent job 1Run on agent

PowerShell ScriptPowerShell

Publish Artifact: dropPublish Build Artifacts

Publish Build Artifacts

Task version1.*

Display name *Publish Artifact: drop

Path to publish *\$(Build.ArtifactStagingDirectory)\export.pdf

Artifact name *drop

Artifact publish location *Azure Pipelines/TFS

Control Options

Output Variables

Test

Guidance

This page implements most of the markdown examples given on the [DevOps wiki docs](#)

It is not claimed or stated that all these features should work. However, this is a quite complete test for the PDF generation.

At some points the official docs show images instead of the rendered content. I suspect that this is to support multiple browsers and even the 'print to PDF' functionality in a browser.

Table of Contents

The table of contents is not supported. Should render just the tag in PDF. See [#47](#)

Headers

Structure your comments using headers. Headers segment longer comments, making them easier to read.

Start a line with a hash character `#` to set a heading. Organize your remarks with subheadings by starting a line with additional hash characters, for example `####`. Up to six levels of headings are supported.

Example:

```
# This is a H1 header
## This is a H2 header
### This is a H3 header
#### This is a H4 header
##### This is a H5 header
```

Result:

This a H1 header

This a H2 header

This a H3 header

This a H4 header

This a H5 header

Paragraphs and line breaks

Make your text easier to read by breaking it up with paragraphs or line breaks.

In pull request comments, select **Enter** to insert a line break, and begin text on a new line.

In a Markdown file or widget, enter two spaces before the line break to begin a new paragraph, or enter two consecutive line breaks to begin a new paragraph.

Example - pull request comment:

```
Add lines between your text with the Enter key.
This spaces your text better and makes it easier to read.
```

Result: Add lines between your text with the Enter key. This action spaces your text better and makes it easier to read.

Example - Markdown file or widget:

```
Add two spaces before the end of the line. (space, space)
This adds space in between paragraphs.
```

Result:

Add two spaces before the end of the line.
Space is added in between paragraphs.

Blockquotes

Quote previous comments or text to set the context for your comment or text.

Quote single lines of text with `>` before the text. Use many `>` characters to nest quoted text. Quote blocks of lines of text by using the same level of `>` across many lines.

Example:

```
> Single line quote
>> Nested quote
>> multiple line
>> quote
```

Result:

```
Single line quote
Nested quote
multiple line
quote
```

Horizontal rules

To add a horizontal rule, add a line that's a series of dashes `---`. The line above the line containing the `---` must be blank.

Example:

```
above
----
below
```

Result:

above

below

Emphasis (bold, italics, strikethrough)

You can emphasize text by applying bold, italics, or strikethrough to characters:

- To apply italics: surround the text with an asterisk `*` or underscore `_`.
- To apply bold: surround the text with double asterisks `**`.
- To apply strikethrough: surround the text with double tilde characters `~~`.

Combine these elements to apply emphasis to text.

There is no Markdown syntax that supports underlining text. Within a wiki page, you can use the HTML `<u>` tag to generate underlined text. For example, `<u>underlined text</u>` yields underlined text.

Example:

```
Use _emphasis_ in comments to express **strong** opinions and point out ~~corrections~~
**_Bold, italicized text_**
~~Bold, strike-through text~~
```

Result:

Use *emphasis* in comments to express **strong** opinions and point out ~~corrections~~
Bold, italicized text **~~Bold, strike-through text~~**

Code highlighting

Highlight suggested code segments using code highlight blocks. To indicate a span of code, wrap it with three backtick quotes (`````) on a new line at both the start and end of the block. To indicate code inline, wrap it with one backtick quote (```).

Code highlighting entered within the Markdown widget renders code as plain preformatted text.

Example:

```
***
sudo npm install vsoagent-installer -g
***
```

Result:

```
sudo npm install vsoagent-installer -g
```

Example:

```
To install the Microsoft Cross Platform Build & Release Agent, run the following: '$ sudo npm install vsoagent-installer -g'.
```

Result:

To install the Microsoft Cross Platform Build & Release Agent, run the following command: `$ sudo npm install vsoagent-installer -g`.

Within a Markdown file, text with four spaces at the beginning of the line automatically converts to a code block.

Set a language identifier for the code block to enable syntax highlighting for any of the supported languages in [highlightjs](#), version v9.10.0.

```
*** language
code
***
```

Additional examples:

```
*** js
const count = records.length;
***
```

```
const count = records.length;
```

```
*** csharp
Console.WriteLine("Hello, World!");
***
```

```
Console.WriteLine("Hello, World!");
```

Tables

Organize structured data with tables. Tables are especially useful for describing function parameters, object methods, and other data that have a clear name to description mapping. You can format tables in pull requests, wiki, and Markdown files such as README files and Markdown widgets.

- Place each table row on its own line
- Separate table cells using the pipe character `|`
- The first two lines of a table set the column headers and the alignment of elements in the table
- Use colons (`:`) when dividing the header and body of tables to specify column alignment (left, center, right)
- To start a new line, use the HTML break tag (`
`) (Works within a Wiki but not elsewhere)
- Make sure to end each row with a CR or LF.
- A blank space is required before and after work item or pull request (PR) mentions inside a table cell.

Example:

```
| Heading 1 | Heading 2 | Heading 3 |
|-----|:-----|:-----|
| Cell A1 | Cell A2 | Cell A3 |
| Cell B1 | Cell B2 | Cell B3<br/>second line of text |
```

Result:

```
| Heading 1 | Heading 2 | Heading 3 |
|-----|:-----|:-----|
| Cell A1 | Cell A2 | Cell A3 |
| Cell B1 | Cell B2 | Cell B3
second line of text |
```

Lists

Organize related items with lists. You can add ordered lists with numbers, or unordered lists with just bullets.

Ordered lists start with a number followed by a period for each list item. Unordered lists start with a `-`. Begin each list item on a new line. In a Markdown file or widget, enter two spaces before the line break to begin a new paragraph, or enter two line breaks consecutively to begin a new paragraph.

Ordered or numbered lists

Example:

```
1. First item.
1. Second item.
1. Third item.
```

Result:

- 1. First item.
- 2. Second item.
- 3. Third item.

Bullet lists

Example:

```
- Item 1
- Item 2
- Item 3
```

Result:

- Item 1
- Item 2
- Item 3

Nested lists

Example:

```
1. First item.
  - Item 1
  - Item 2
  - Item 3
1. Second item.
  - Nested item 1
  - Nested item 2
  - Nested item 3
```

Result:

- 1. First item.
 - Item 1
 - Item 2
 - Item 3
- 2. Second item.
 - Nested item 1

- Nested item 2
- Nested item 3

Links

In pull request comments and wikis, HTTP and HTTPS URLs are automatically formatted as links. You can link to work items by entering the # key and a work item ID, and then choosing the work item from the list.

Avoid auto suggestions for work items by prefixing # with a backslash (\). This action can be useful if you want to use # for color hex codes.

In Markdown files and widgets, you can set text hyperlinks for your URL using the standard Markdown link syntax:

```
[Link Text](Link URL)
```

When linking to another Markdown page in the same Git or TFVC repository, the link target can be a relative path or an absolute path in the repository.

Supported links for Welcome pages:

- Relative path: `[text to display](/target.md)`
- Absolute path in Git: `[text to display](/folder/target.md)`
- Absolute path in TFVC: `[text to display]($/project/folder/target.md)`
- URL: `[text to display](http://address.com)`

Supported links for Markdown widget:

- URL: `[text to display](http://address.com)`

Supported links for Wiki:

- Absolute path of Wiki pages: `[text to display](/parent-page/child-page)`
- URL: `[text to display](http://address.com)`

Anchor links

Within Markdown files, anchor IDs are assigned to all headings when rendered as HTML. The ID is the heading text, with the spaces replaced by dashes (-) and all lower case. In general, the following conventions apply:

- Punctuation marks and leading white spaces within a file name are ignored
- Upper case letters are converted to lower
- Spaces between letters are converted to dashes (-).

Example:

```
###Link to a heading in the page
```

Result:

The syntax for an anchor link to a section...

```
[Link to a heading in the page](#link-to-a-heading-in-the-page)
```

The ID is all lower case, and the link is case-sensitive, so be sure to use lower case, even though the heading itself uses upper case. You can also reference headings within another Markdown file:

```
[text to display](./target.md#heading-id)
```

In wiki, you can also reference heading in another page:

```
[text to display](/page-name#section-name)
```

Images

To highlight issues or make things more interesting, you can add images and animated GIFs to the following aspects in your pull requests:

- Comments
- Markdown files
- Wiki pages

Use the following syntax to add an image:

```
! [Text] (URL)
```

The text in the brackets describes the image being linked and the URL points to the image location.

Example:

```
! [Illustration to use for new users] (https://azurecomodn.azureedge.net/cvt-779fa2985e70b1ef1c34d319b505f7b4417add09948df4c5b81db2a9bad966e5/images/page/services/devops/hero-
```

Result:



The path to the image file can be a relative path or the absolute path in Git or TFVC, just like the path to another Markdown file in a link.

- Relative path: `! [Image alt text] (./image.png)`
- Absolute path in Git: `! [Image alt text] (/media/markdown-guidance/image.png)`
- Absolute path in TFVC: `! [Image alt text] ($/project/folder/media/markdown-guidance/image.png)`
- Resize image: `IMAGE_URL =WIDTHxHEIGHT`

Be sure to include a space before the equal sign.

- **Example:** `![[Image alt text]]($/project/folder/media/markdown-guidance/image.png){width=500 height=250}`
- It's also possible to specify only the WIDTH by leaving out the HEIGHT value: `IMAGE_URL =WIDTHx`

Checklist or task list

You can Use `[]` or `[x]` to support checklists. Precede the checklist with either `-<space>` or `1.<space>` (any numeral).

Example - Format a list as a task list

```
- [ ] A
- [ ] B
- [ ] C
- [x] A
- [x] B
- [x] C
```

Result:

- ☐ A
- ☐ B
- ☐ C
- ☒ A
- ☒ B
- ☒ C

A checklist within a table cell isn't supported.

Emoji

Example:

```
:smile:
:angry:
```

Result:



To escape emojis, enclose them using the ``` character.

Example:

```
`:smile:` `:)` `:angry:`
```

Result:

```
:smile: :) :angry:
```

See a full list of emoji and how they render [on the Emoji page](#) (this is also an example of a link wiht an anchor to another pase)

Ignore or escape Markdown syntax to enter specific or literal characters

Syntax	Example/notes
--------	---------------

To insert one of the following characters, prefix with a `\` (backslash).

`\` , backslash
``` , backtick  
`_` , underscore  
`{}` , curly braces  
`[]` , square brackets  
`()` , parentheses  
`#` , hash mark  
`+` , plus sign  
`-` , minus sign (hyphen)  
`.` , period  
`!` , exclamation mark  
`*` , asterisk

Some examples on inserting special characters:

Enter `\\` to get `\`

Enter `\_` to get `_`

Enter `\#` to get `#`

Enter `\(` to get `(`

Enter `\.` to get `.`

Enter `\!` to get `!`

Enter `\*` to get `*`

## Mathematical notation and characters

Supported in: Pull Requests | Wikis

Both inline and block [KaTeX](#) notation is supported in wiki pages and pull requests. The following supported elements are included:

- Symbols
- Greek letters
- Mathematical operators
- Powers and indices
- Fractions and binomials
- Other KaTeX supported elements

To include mathematical notation, surround the mathematical notation with a `$` sign, for inline, and `$$` for block, as shown in the following examples:

This feature is supported within Wiki pages and pull requests for TFS 2018.2 or later versions.

### Example: Greek characters

```
$
\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \kappa, \lambda, \mu, \nu, \omicron, \pi, \rho, \sigma, \tau, \upsilon, \phi, ...
$

$\Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Upsilon, \Phi, \Psi, \Omega$
```

### Result:

`$ \alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \kappa, \lambda, \mu, \nu, \omicron, \pi, \rho, \sigma, \tau, \upsilon, \phi, ... $`

`\(\Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Upsilon, \Phi, \Psi, \Omega\)`

### Example: Algebraic notation

```
Area of a circle is πr^2

And, the area of a triangle is:

$$
A_{\text{triangle}}=\frac{1}{2}(\text{b}\cdot\text{h})
$$
```

### Result:

Area of a circle is  $\pi r^2$

And, the area of a triangle is:

$A_{\text{triangle}}=\frac{1}{2}(\text{b}\cdot\text{h})$

### Example: Sums and Integrals

```
$$
\sum_{i=1}^{10} t_i
$$

$$
\int_0^{\infty} \mathrm{e}^{-x} \mathrm{d}x
$$
```

**Result:**

$$\sum_{i=1}^{10} t_i$$
$$\int_0^{\infty} \mathrm{e}^{-x} \mathrm{d}x$$

**Additional**

I tried to recreate the [Microsoft Docs Alerts](#)

This one needs a red background

**⚠ CAUTION**

There will be irreversible consequences

This one needs a yellow background

**⚠ WARNING**

Dangerous certain consequences of an action can apply

This one needs a purple background

**📌 Important**

Essential information required for success

This one needs a light blue background

**💡 Tip**

Optional information to help the user be more successful!

This one needs a green background

**📝 Note**

Information the user should note when skimming the document!

**Resources**

Emoji cheat sheet: <https://www.webfx.com/tools/emoji-cheat-sheet/>

## TEST PAGE

The following are tests for the export page

### Copy & Paste Picture



### Table of Contents

#### Table Test

| a | b |
|---|---|
| 0 | 1 |

#### Formular

$$E=mc^2$$

#### Mention

@<7C0C20F7-3AB6-6232-8E80-E35152712AF5>

##Malformed header (no space!)

#### Lists

- 1 Entry
- 2 Entry
- 3 Entry

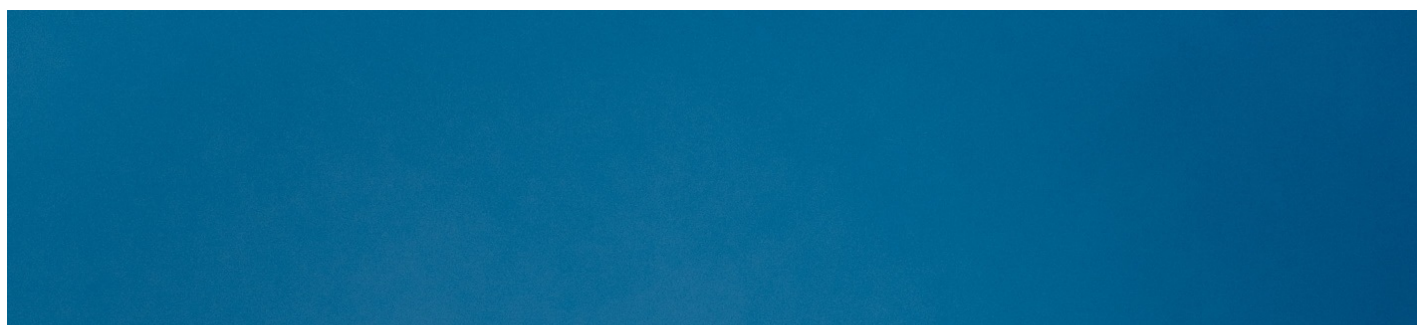
#### Formats

**bold** *italic* [link](#) `code`

#### Mermaid

```
graph LR; A[Wiki supports Mermaid] --> B[Visit https://mermaidjs.github.io/ for Mermaid syntax];
```

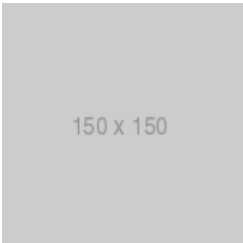
#### Large Pictures



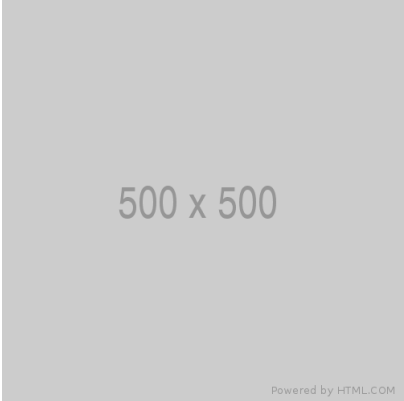




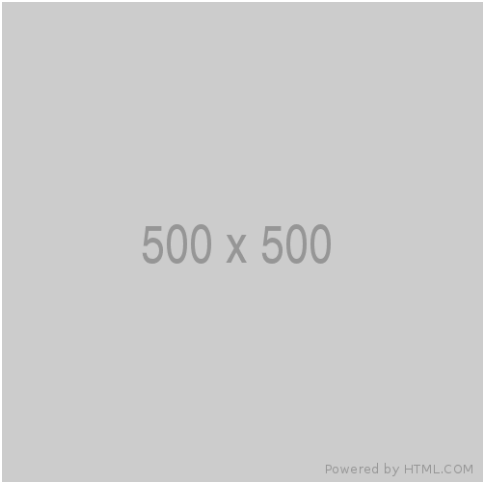
Pictures with Space in Filename



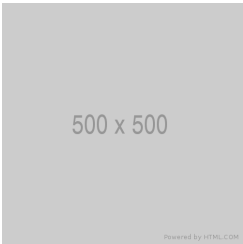
Pictures with Width/Height



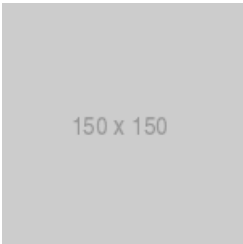
Pictures with Width only



Pictures with Height only



Picture as html



Table

| Revision | Datum | Änderungen zur vorherigen Revision| |--|--|--| | A | 09.02.2021 | Erste Veröffentlichung der Bedienungsanleitung|

| Pin Nummer                       | Pin Name | Funktion                                    | Signalrichtung       |
|----------------------------------|----------|---------------------------------------------|----------------------|
| 1, 10, 19, 28, 37, 46            | GND      | Ground                                      | -                    |
| 47, 49                           | VCC      | +5VDC<br>Versorgungsspannung                | -                    |
| 2...9;<br>11...15;<br>17, 51, 52 | A0...A15 | 16-Bit<br>Adressbuss                        | Eingang              |
| 38                               | /CS      | Chipselect                                  | Eingang (Low active) |
| 39                               | /RD      | Read Befehl                                 | Eingang (Low active) |
| 43                               | Sync1    | Resettakt                                   | Ausgang              |
| 45                               | /Busy    | Briefmarke aktiv                            | Ausgang (Low active) |
| 20...27;<br>29...36              | D0...D15 | 16-Bit<br>Datenbus                          | Ein-/Ausgang         |
| 41                               | /BHE     | Byte High Enable<br>für 16-Bit Adressierung | Eingang (Low active) |
| 40                               | /WR      | Write Befehl                                | Eingang (Low active) |