

# AzureDevOps.WikiPDFExport

## Quickstart

This tool exports a Azure DevOps wiki to 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

## 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:

SDisk (C:) > git > AzureDevOps.WikiPDFExport.wiki

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 K
.order	26.05.2019 00:03	ORDER File	1 K
AzureDevOps.WikiPDFExport.md	26.05.2019 00:04	MD File	2 K
License.md	26.05.2019 00:03	MD File	2 K

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

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.
- It is fast. A PDF with 160 pages is created in less than a second.

## Requirements

The tool is developed as .NET Core 2.2 application, therefore you need to have the runtime installed. Download is available [here](#).

## Download

The download is available [here](#)

## Configuration Options

`-o / --output`

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

`-d / --date`

The current date will be added to the footer

`-b / --breakPage`

01.06.2019 15:47

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

**-t / --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.

**-v / --verbose**

Verbose mode. Logging will added to the console window

**-g / --debug**

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

**-h / --help**

Help - outputs the parameters that can be used

## Limitations

So far the following limitations are known:

- TOC (Table of Contents) tag is not supported and will exported as tag
- 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.
- The tool lacks proper testing because I only have two wikis available

## License

See [license](#)

## Thanks

In this tool uses three open source libraries are doing the 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

## TEST-PAGE

tags:

- test
- page
- yaml title: Test Page

The following are tests for the export page

### Copy & Paste Picture



### Table of Contents

[[TOC]]

### Video

... video

...

### 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`

## 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-646b32fba0b/_apis/git/repositories/e08d1ada-7794-4b89-a3ea-cb64a26683c3/Items?path=%2Fazuredevops-export-wiki.exe&
```

```
#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 task is named 'Agent job 1' and is set to 'Run on agent'. The configuration panel on the right shows the following details:

- Display name \***: Agent job 1
- Agent selection ^**: Agent pool
- Agent pool**: Hosted VS2017 (with links for Pool information and Manage)
- Demands**: None

### PowerShell Task

The screenshot shows the 'PowerShell Script' task configuration in the Azure DevOps build pipeline. The task is named 'PowerShell Script' and is set to 'Run on agent'. The configuration panel on the right shows the following details:

- Task version**: 2.\*
- Display name \***: PowerShell Script
- Type**: Inline (selected over File Path)
- Script \***:
 

```
$url = "https://dev.azure.com/mmelcher/8036eca1-fd9e-4c0f-8bef-646b32fba0b/_apis/git/repositories/e08d1ada-7794-cb64a26683c3/Items?path=%2Fazuredevops-export-wiki.exe&versionDescriptor%5BversionOptions%5D=0&versionDescriptor%5BversionType%5D=0&versionDescriptor%5Bdownload=true&resolveLfs=true&%24format=octetStream&api-version=5.0-preview.1"

$output = "azuredevops-export-wiki.exe"

Invoke-WebRequest -Uri $url -OutFile $output

./azuredevops-export-wiki.exe
```
- ErrorActionPreference**: Stop

### Publish Task

Tasks
Variables
Triggers
Options
Retention
History

Save & queue
Discard
Summary
Queue
...

Pipeline
Build pipeline

Get sources
repository
wikiMaster

Agent job 1
Run on agent

PowerShell Script
PowerShell

Publish Artifact: drop
Publish Build Artifacts

Publish Build Artifacts

Task version
1.\*

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

## 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.