
Note taking with AsciiDoctor

Jens Getreu

Revision History

Revision 2.7

5.11.2016

JG

Table of Contents

1. How students take notes	3
1.1. The lesson starts	3
1.2. Taking notes about a file	5
2. Create a new note	7
2.1. Invoking the script	7
2.2. Directory as parameter	8
2.3. No parameter	9
2.4. Filename as parameter	10
3. Editing notes	13
3.1. Syntax	13
3.2. Example	13
4. Viewing notes	15
4.1. Syntax	15
4.2. Example	15
5. AsciiDoc title-filename sync	15
6. Installation and configuration	18
6.1. Linux	18
6.2. Windows	19
6.3. Integration with file manager	21
7. Source code	25
7.1. src/adoc-note	25
7.2. src/adoc-note.bat	31

Markup languages like `AsciiDoc` are perfectly suited for fast note taking. Type your notes with your favourite editor, view and browse them with `Chromium`, `Chrome` or `Firefox`.



This project is discontinued. Please use [tp-note¹](#) instead. The code of `tp-note` is more generic and works with all markup languages.

All you need is the `adoc-note` script and Chromium, Chrome or Firefox with the [Chrome AsciiDoctor.js Live Preview²](#) extension or the [Firefox AsciiDoctor.js Live Preview³](#) extension.

`adoc-note` runs under Linux and Windows.⁴ The script operates in four modes depending on it's options (see [Section 2.1, "Invoking the script"](#)):

Create a new note

Creates a new context depending asciidoc note file.

Edit notes

Launches an editor.

View notes

Launches a live-previewer (Firefox, Chrome of Chromium).

Sync filename

Renames the filename to be in sync with the document's title.

The project is hosted on Github: [getreu/asciidoc-notetaking⁵](#). The project's webpage is on [blog.getreu.net⁶](#). There you also find a [pdf rendition⁷](#) of this document.

Quickstart

1. Read the installation guide for [Section 6.1, "Linux"](#) or [Section 6.2, "Windows"](#).
2. Read about the 2 most common use cases [Section 1, "How students take notes"](#).

¹ <http://blog.getreu.net/projects/tp-note/>

² <https://chrome.google.com/webstore/detail/asciidocorjs-live-previe/iaalpfqpbocpdfblpn-hhgllgdbbchmia>

³ <https://addons.mozilla.org/en-US/firefox/addon/asciidocorjs-live-preview/>

⁴ For Mac follow Linux instructions (not tested).

⁵ <https://github.com/getreu/asciidoc-notetaking>

⁶ <http://blog.getreu.net/projects/asciidoc-notetaking/>

⁷ http://blog.getreu.net/_downloads/asciidoc-notetaking.pdf

1. How students take notes

A fellow student still uses paper and pen. I ask her why and she replied “I can better concentrate. My computer distracts me. I will do all other things, but not listening.”.

This is certainly true. As I am concerned I am not good at logistics. For me having all documents and notes in one little machine is a blessing.

To illustrate how to work with `adoc-note` here my most common workflows.

1.1. The lesson starts

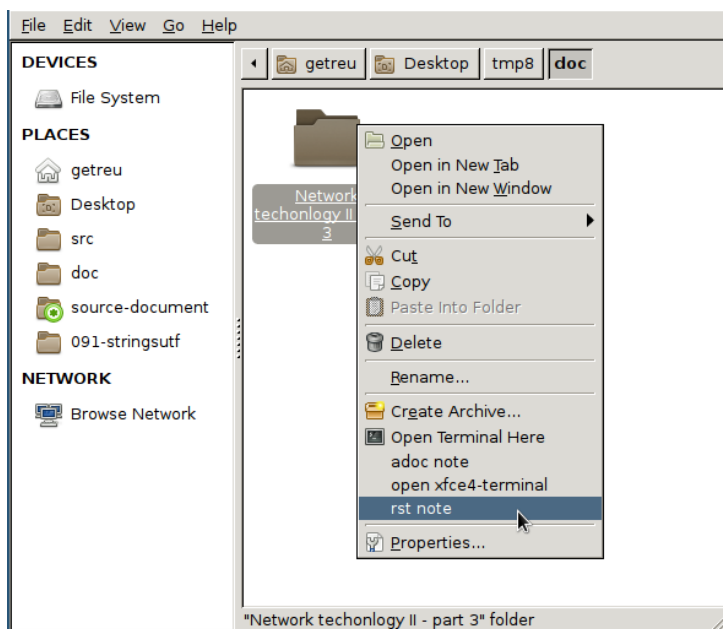


Figure 1. The folder in which the new note will be created.

Alternatively you can open the folder you want to create a new note in and right-click on some empty white space.

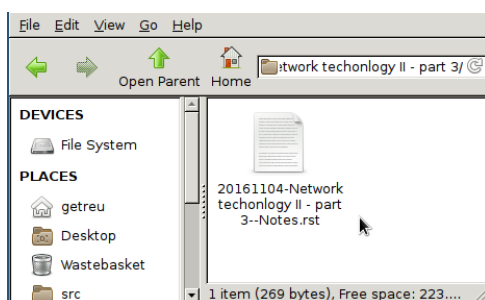


Figure 2. The new file

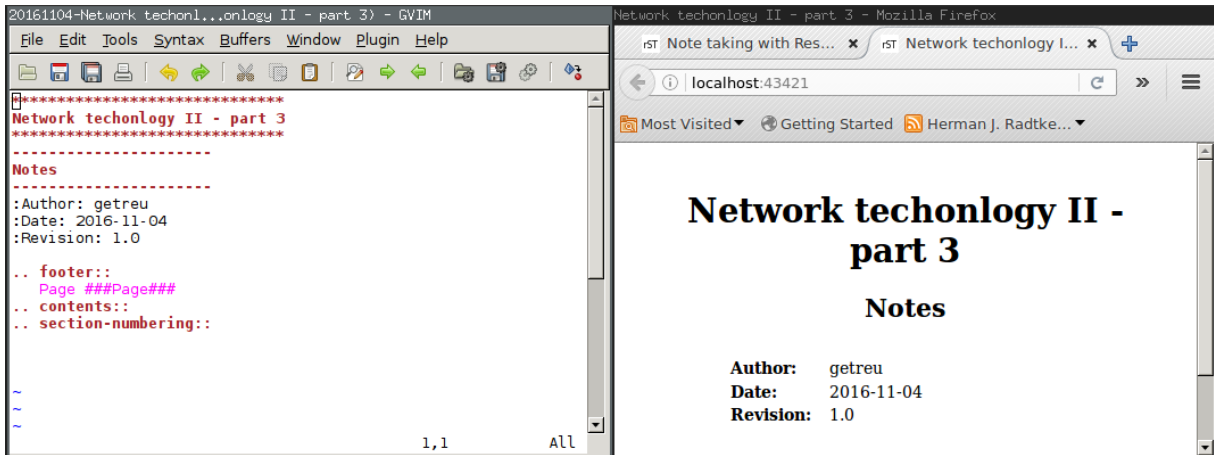


Figure 3. Editor and Browser windows with unmodified template

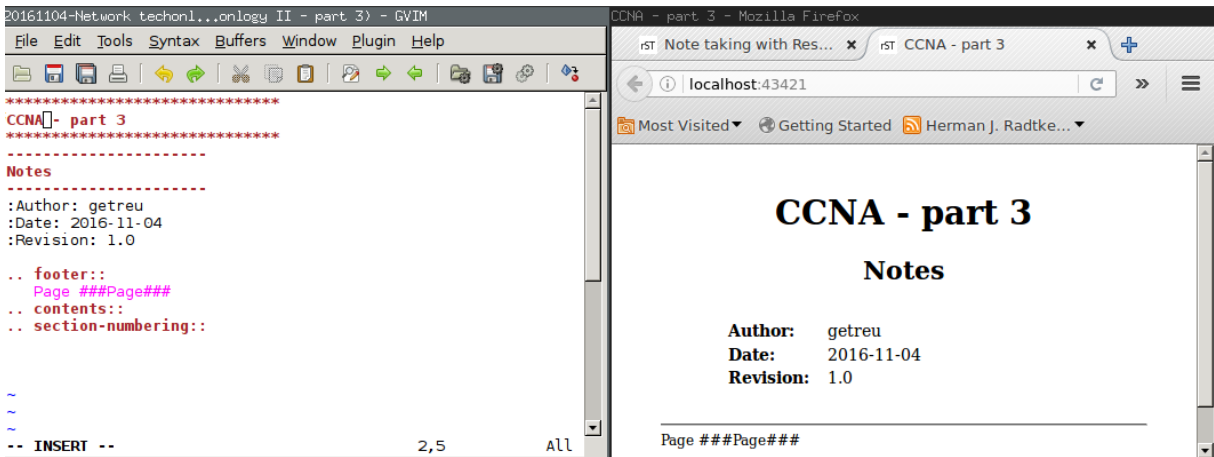


Figure 4. Title change

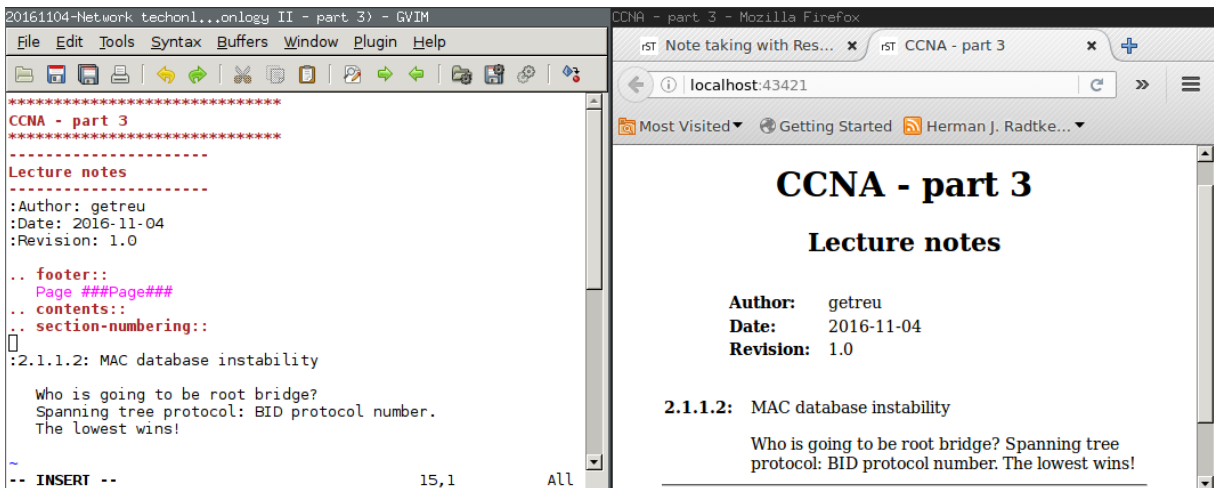


Figure 5. Some text added

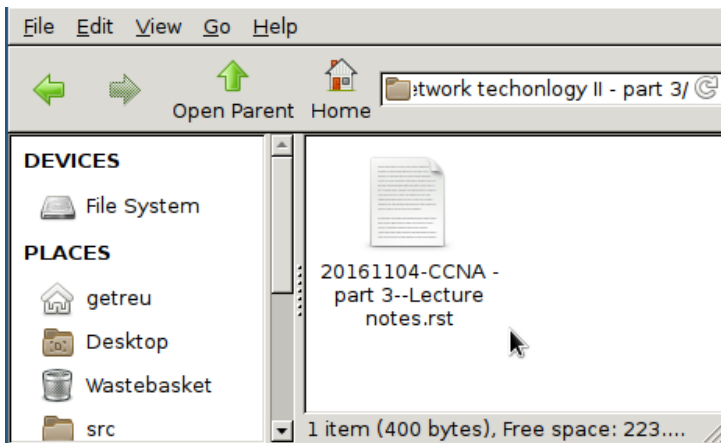


Figure 6. The new note file on disk after closing the editor



Before and after launching the editor `adoc-note` renames the file to be in sync with the asciidoc title. For more details see [Section 5, “AsciiDoc title-filename sync”](#).

1.2. Taking notes about a file

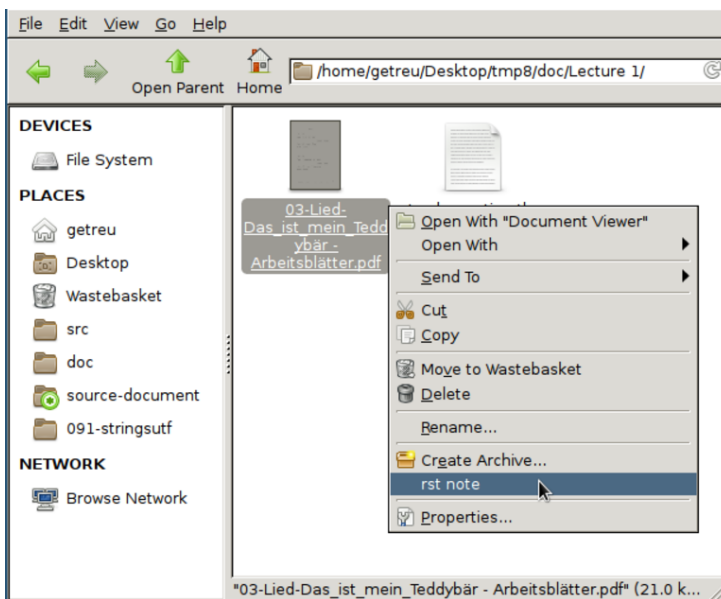


Figure 7. We want to take a note about a pdf

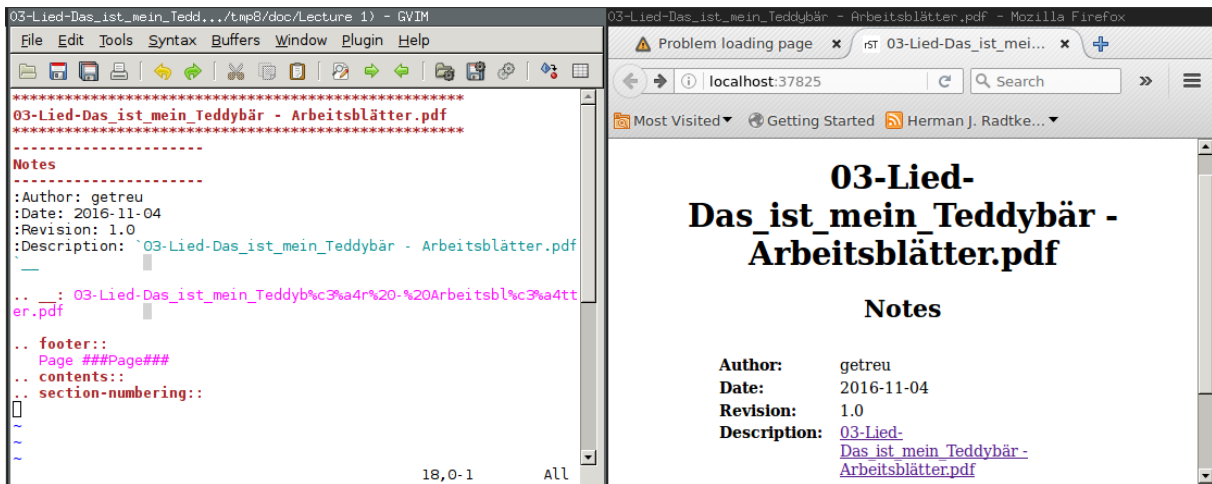


Figure 8. Editor and Browser windows with unmodified template

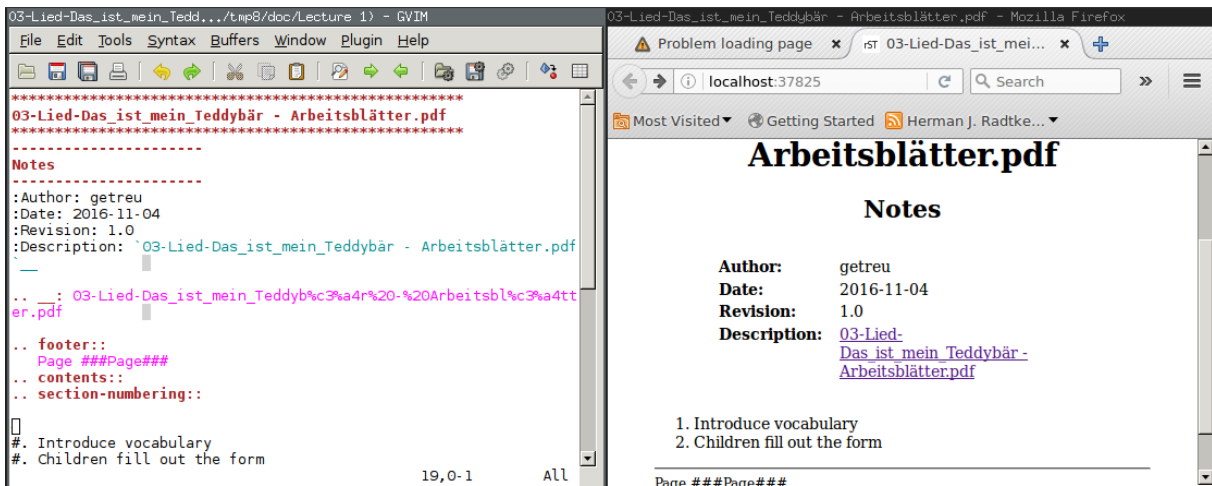


Figure 9. We add a note about the origin of the pdf

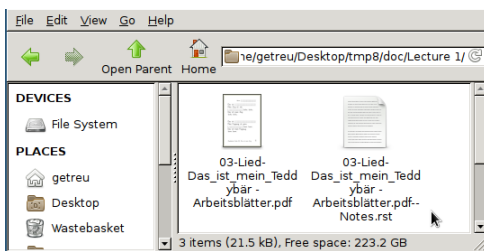


Figure 10. The new note file on disk after closing the editor

2. Create a new note

There are several ways to launch `adoc-note`.

Linux

- Launch the commands in shell as described the sections *Syntax* below.
- Right click on a file or directory in your file-manger and choose *adoc-note* in the context menu. See [Section 6.3, “Integration with file manager”](#).

Windows

- Drag and drop a file or directory on the shortcut `adoc-note.bat` on your desktop.

This method also works in very restricted environments. The only rights you need is to be allowed to execute portable binaries and batch scripts. You also need the permission to install the *AsciiDoctor.js Live Preview* plugin in Chromium, Chrome or Firefox.

- Double-click the shortcut `adoc-note.bat` on your desktop. This will place a new note on your desktop.
- Right click on a file or directory in file explorer and choose *adoc-note* in the context menu. See [Section 6.3, “Integration with file manager”](#).

2.1. Invoking the script

```
adoc-note -h
```

shows a short help text with available command line options:

```
/usr/local/bin/adoc-note creates, edits or views an asciidoc note.
```

usage:

```
/usr/local/bin/adoc-note [-h][-ro|-so|-eo] | [<File.adoc>|<Dir>|<File>]
```

<Dir>|<File>: directory where the new note file will be created (current directory if none).

If <File> is given a new adoc note will be created next to that file.

If <File.adoc> is given the file is edited.

Filename of <File> is changed when not in sync with title.

Options:

-ro Do not open editor, open viewer only.

-eo Do not open viewer, only new note or, sync filename and edit.

-so Do not open editor or viewer, only new note or sync filename.

Table 1. `adoc-note` options

Option	Create a new note	Launch editor	Launch viewer	Sync title-filename
<i>without</i>	Y *	Y	Y	Y
-ro	Y *	N	Y	N
-eo	Y *	Y	N	Y
-so	Y *	N	N	Y

Lengend

Y = yes

N = no

* = If a note with the same filename exists on disk already, no new note is created.

2.2. Directory as parameter

Syntax

```
adoc-note <path>/<dir>
```

creates the following document:

```
<path>/<dir>/YYYYMMDD-<dir>--Notes.adoc
```

Example 1. `<path>/<dir>/YYYYMMDD-<dir>--Notes.adoc`

```
= <dir> : Notes
<loginname>
v1.0, <date>
:numbered:
```



```
:toc!:  
:pagenums:
```

Example

```
adoc-note 'doc/Lecture 1'
```

creates the following document:

```
doc/Lecture 1/20151129-Lecture 1--Notes.adoc
```

Example 2. doc/Lecture 1/20151129-Lecture 1-- Notes.adoc

```
= Lecture 1 : Notes  
johndoe  
v1.0, 29.11.2015  
:numbered:  
:toc!:  
:pagenums:
```

2.3. No parameter

Syntax

It is also possible to invoke the script without options:

```
adoc-note
```

The result is the same as above but the current working directory is used for `<path>/<dir>`.

Example

```
cd 'doc/Lecture 1'  
adoc-note
```

creates the following document:

doc/Lecture 1/20151129-Lecture 1--Notes.adoc

Example 3. doc/Lecture 1/20151129-Lecture 1—Notes.adoc

```
= Lecture 1 : Notes
johndoe
v1.0, 29.11.2015
:numbered:
:toc!:
:pagenums:

== Notes
```

2.4. Filename as parameter

The filename should be some existing local file you want to annotate. For example I use this feature to note from where I have downloaded a .pdf.

Syntax

When invoke with a filename, no date stamp is prepended.

```
adoc-note <path>/<dir>/<filename>
```

The new file will look like this:

```
<path>/<dir>/<filename>--Notes.adoc
```

Example 4. <path>/<dir>/<filename>-- Notes.adoc

```
= <filename> : Notes
<loginname>
v1.0, <date>
:numbered:
:toc!:
:pagenums:

Annotations on file: link:++<filename>++[]
```

Example

```
adoc-note 'doc/Implementing the NIST Cybersecurity Framework.pdf'
```

creates the following document:

```
doc/Implementing the NIST Cybersecurity Framework.pdf--Notes.adoc
```

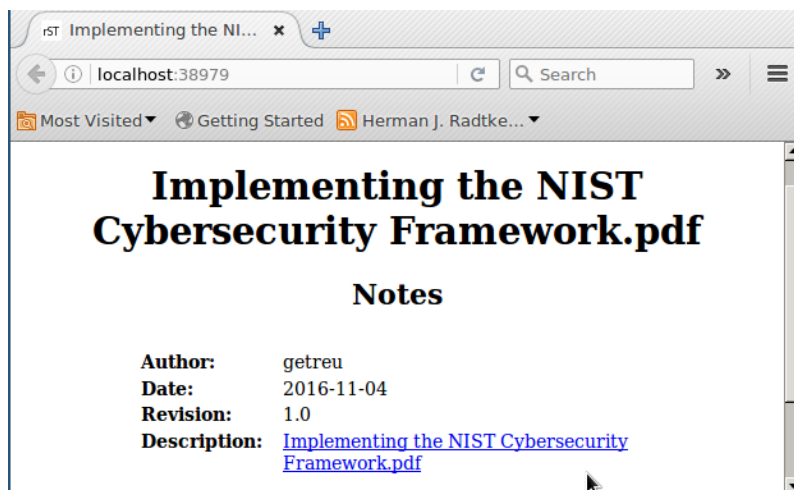
Example 5. doc/Implementing the NIST Cybersecurity Framework.pdf —Notes.adoc

```
= Implementing the NIST Cybersecurity Framework.pdf : Notes
johndoe
v1.0, 29.11.2015
:numbered!:
:toc!:
:pagenums:
```

```
Annotations on file: link:++Implementing-the-NIST-Cybersecurity-
Framework.pdf++[]
```

Before quitting the `adoc-note` script executes its *editing-mode*. This opens in your editor the completed template (see example above) and a live-previewr showing the rendered asciidoc file:

Example 6. Rendered note



3. Editing notes

3.1. Syntax

```
adoc-note <path>/<filename>.adoc
```

launches the the `gvim` editor by default. Replace `gvim` by any editor of your choice. At the same time a `chrome` or `chromium` window will pop up showing the live rendition of your asciidoc file. You need to have the [AsciiDoctor.js Live Preview](#)⁸ extension installed.

You may want to use some autosave editor feature in order to observe changes immediately in the rendered preview live.

At the beginning and the end of the `adoc-note` script another helper function `ChangeFilename` is called. It guarantees that any change in the asciidoc title of the file you have been editing will replicate in the filename of that asciidoc file. This guarantees that **the filenames of asciidoc notes always correspond to their asciidoc title** allowing you to find your notes quickly in your directory structure. See [Section 5, “AsciiDoc title-filename sync”](#) for more details.



You can disable the title-filename-sync feature by adding a blank line at the beginning of the asciidoc document.

3.2. Example

```
adoc-note 'Implementing the NIST Cybersecurity Framework.pdf--  
Notes.adoc'
```

Figure 11. adoc-edit invokation example

The same result is obtained by repeating the same command you used to create this note ⁹:

⁸ <https://chrome.google.com/webstore/detail/asciidocorjs-live-previe/iaalpfgpbocpdfblpn-hhgllgdbchmia>

⁹ This only works if you have not changed the original title in the meantime!

.....
adoc-note 'Implementing the NIST Cybersecurity Framework.pdf'
.....

4. Viewing notes

4.1. Syntax

```
adoc-note -ro <path>/<filename>.adoc
```

launches `firefox` or `chromium`. You need to have the [Firefox AsciiDoctor.js Live Preview¹⁰](#) extension or the [Chrome AsciiDoctor.js Live Preview¹¹](#) extension installed.

4.2. Example

```
adoc-note -ro 'Implementing the NIST Cybersecurity Framework.pdf--Notes.adoc'
```

Figure 12. adoc-view invokation example

All text in the above example was automatically generated by the `adoc-note` script only the last link *download URL* was added “by hand”. This is particularly useful for quickly writing down the source of a downloaded document.

5. AsciiDoc title-filename sync

Consider the following note file:

```
20151208-Make this world a better place--Suggestions.adoc
```

The filename has 3 parts:

```
<sort tag>-<simplified-title>--<simplified-subtitle>.adoc
```

A `<sort tag>` can be a

- *chronological sort tag* or

¹⁰ <https://addons.mozilla.org/en-US/firefox/addon/asciidocorjs-live-preview/>

¹¹ <https://chrome.google.com/webstore/detail/asciidocorjs-live-previe/iaalpfgpbocpdfblpn-hhgllgdbdbchmia>

Example 7. Chronological sort tags

20140211-
20151208-

- a sequence number sort tag.

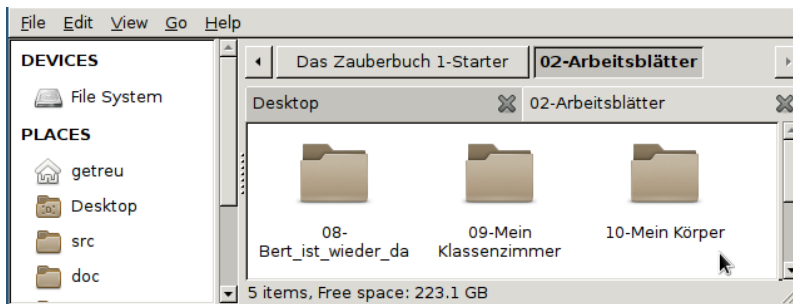
Example 8. Sequence number sort tags

02-
08-
09_02-

<sort tag> can be any combination of 0123456789-_.

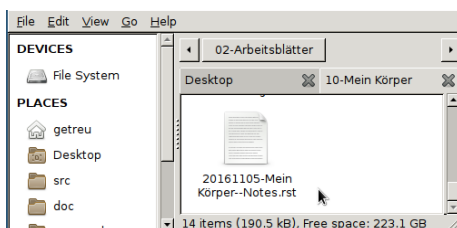
When `adoc-note` creates a new note based on a directory it prepends a *chronological sort tag* of today. The `<simplified-title>` part is derived from the parent directory name omitting its own *sort tag*.

Example 9. New note in a directory with sequence number sort tag



`adoc-note '10-Mein Körper'`

will result in a new file:



10-Mein Körper/20151209-Mein Körper--Notes.adoc



The parent directory's sort tag is never used to compose a filename for a new note.

When `adoc-note` creates a new note based on a filename no *sort tag* is prepended.

Before and after editing the `adoc-note` analyses the title and subtitle of the asciidoc file and simplifies them in a filesystem friendly form. If the result does not equal to `<simplified-title>--<simplified-subtitle>` the filename is changed on disk. Possible *sort tags* remain untouched.



You can disable this feature by inserting a blank line at the beginning of the asciidoc file.



`adoc-note` might change the note's filename but never changes an *sort tag*!

For details about how an AsciiDoc title line is simplified into a `<simplified-title>--<simplified-subtitle>` string refer to the `Sanitize-Filename()` function in [???](#).

6. Installation and configuration

`adoc-note` runs on Linux and Windows. For Windows installation and configuration see [Section 6.2, “Windows”](#) below.

6.1. Linux

1. Install `chromium` (alternatively you can use `chrome` or `firefox`).¹²
2. Install the [Chrome AsciiDoctor.js Live Preview](#)¹³ extension. (Or [Firefox AsciiDoctor.js Live Preview](#)¹⁴)
3. Download the note-taking-script `bin/adoc-note` from Github [getreu/asciiDoctor-notetaking](#)¹⁵. Alternatively you can copy and paste the listing at end of this document.
4. Copy it in a location of your `$PATH` and make it executable for everyone.

```
sudo cp adoc-note /usr/local/bin
sudo chmod a+rx /usr/local/bin/adoc-note
```

5. Install some helper packages (most should be already on your system)¹⁶.

```
sudo apt-get install sed uni2ascii vim-gtk
```

Replace `vim-gtk` with an editor of your choice. Configure `adoc-note` accordingly.

6. Configuration:

At the beginning of `adoc-note` you will find a section enclosed in the comments `CONFIGURATION SECTION START` and `CONFIGURATION SECTION END`.

¹² At the time of this writing only `chromium` and `chrome` refresh automatically and can be launched in application interface mode. My preferred choice.

¹³ <https://chrome.google.com/webstore/detail/asciiDoctorjs-live-privie/iaalpfgpbocpdfblpn-hhgllgdbchmia>

¹⁴ <https://addons.mozilla.org/en-US/firefox/addon/asciiDoctorjs-live-preview/>

¹⁵ <https://github.com/getreu/asciiDoctor-notetaking>

¹⁶ `uni2ascii` is only needed for `chrome` and `chromium`



The only file you edit to change the default choices for *editor* and *viewer* under Linux is `adco-note`. **Not** `adoc-note.bak` !

Here you can specify what editor you want to use, if you choose `firefox`, `chrome` or `chromium` and you can specify the parameters these programs require. Make sure that your editor does not fork when launched. If it does fork, the script will still work but when you quit, no `asciidoc-title-filename` sync will occur. Outside the `CONFIGURATION SECTION` no changes should be necessary.



The environment variables `ADOC_EDITOR` or `ADOC_VIEWER` -when defined- have precedence over settings in the `CONFIGURATION SECTION` and will override them.

7. Test the installation: open a console window and type `adoc-note`. An *editor* and *viewer* window containing a note template should open.
8. Optional: integrate the scripts with your file-manager (see [the section called “Linux file manager configuration”](#)).

6.2. Windows

1. Install `chromium` (alternatively you can use `chrome` or `firefox`).¹⁷
2. Install the [Chrome AsciiDoctor.js Live Preview](#)¹⁸ extension. (Or [Firefox AsciiDoctor.js Live Preview](#)¹⁹)
3. Download the the files `bin/adoc-note`, `bin/adoc-note.bat` and `bin/busybox.exe` from Github [getreu/asciidoc-notetaking](#)²⁰. Alternatively you can copy and paste the listings at end of this document. The version of `busybox` in the above repository is probably outdated. Please get a newer version from here: <http://frippery.org/busybox/>
4. Copy the 3 files in a directory of your choice. In the following I call this directory `BIN_DIR`.

¹⁷ At the time of this writing only `chromium` and `chrome` refresh automatically and can be launched in application interface mode. My preferred choice.

¹⁸ <https://chrome.google.com/webstore/detail/asciidocorjs-live-previe/iaalpfgpbocpdfblpn-hhgllgdbchmia>

¹⁹ <https://addons.mozilla.org/en-US/firefox/addon/asciidocorjs-live-preview/>

²⁰ <https://github.com/getreu/asciidoc-notetaking>

5. Configuration:

At the beginning of `adoc-note.bat` you will find a section enclosed in the comments `CONFIGURATION SECTION START` and `CONFIGURATION SECTION END`.



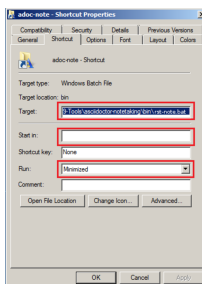
The only file you edit to change the default choices for *editor* and *viewer* under Windows is `adco-note.bat`. **Not** `adoc-note`!



New notes are created with an Unicode BOM indicating Unicode encoding. Do not use the `notepad` editor coming with Windows older than Vista. These old versions of `notepad` do not handle Unicode correctly. Use `Wordpad` or and other modern Unicode editor instead.

Here you can specify what browser you want to use, e.g. if you choose `firefox.exe` or `chromium.exe` and you can specify the parameters these programs require. Do the same for the editor of your choice. Make sure that your editor does not fork when launched. If it does fork, the script will still work but when you quit, no `asciidoc-title-filename` sync will occur. Outside the `CONFIGURATION SECTION` no changes should be necessary.

6. Create a shortcut to `adoc-note.bat` on your desktop, click on properties and change the shortcut to *run minimized*. Let the *start in* path empty.



7. Test the installation: drag a file or directory on the `adoc-note Shortcut` on your desktop. An *editor* and *viewer* window should open.

8. Optional: integrate the scripts with your file-manager (see [the section called "Windows file explorer configuration"](#)).

6.3. Integration with file manager

`adoc-note` integrates easily with your favorite [file-manager](#) under Linux and [Explorer](#) under Windows.

Linux file manager configuration

The example below shows the *Thunar* filebrowser's custom actions.

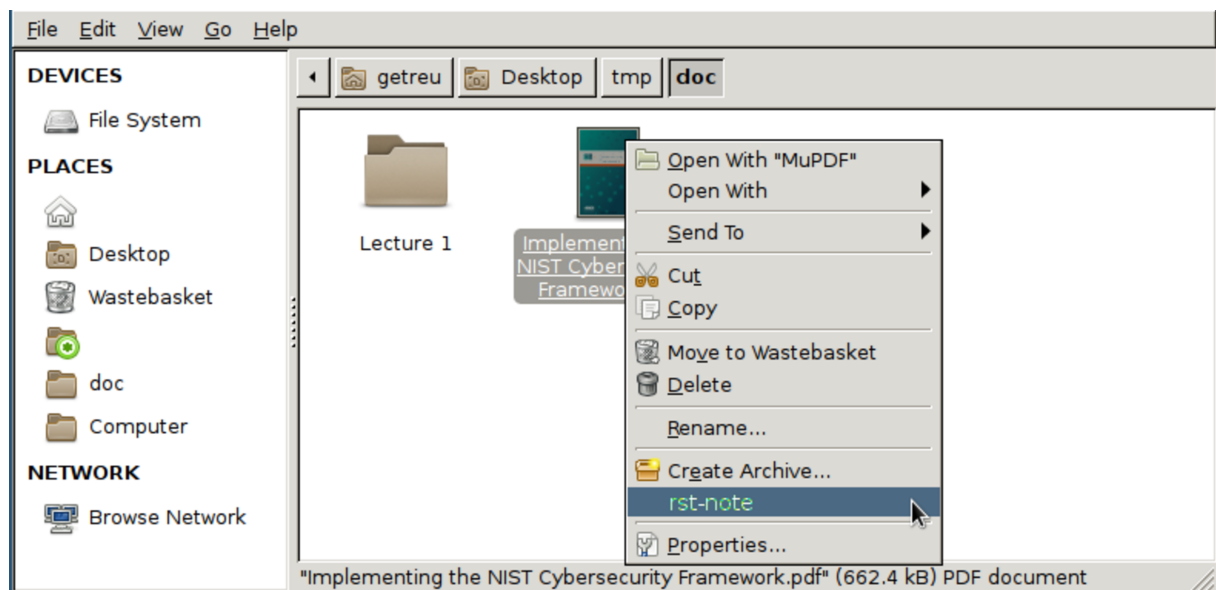


Figure 13. Integration with file-manager

Most file-manager allow extending the context menu. As an example the following images show the configuration of the Thunar-file-manger.

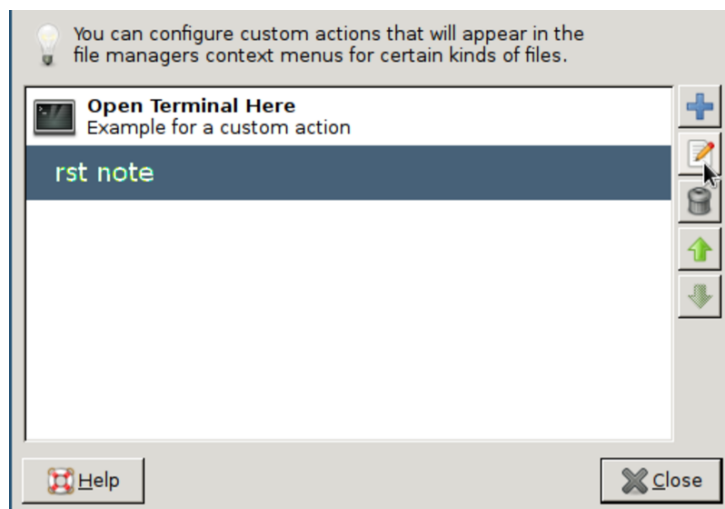


Figure 14. Thunar custom action configuration

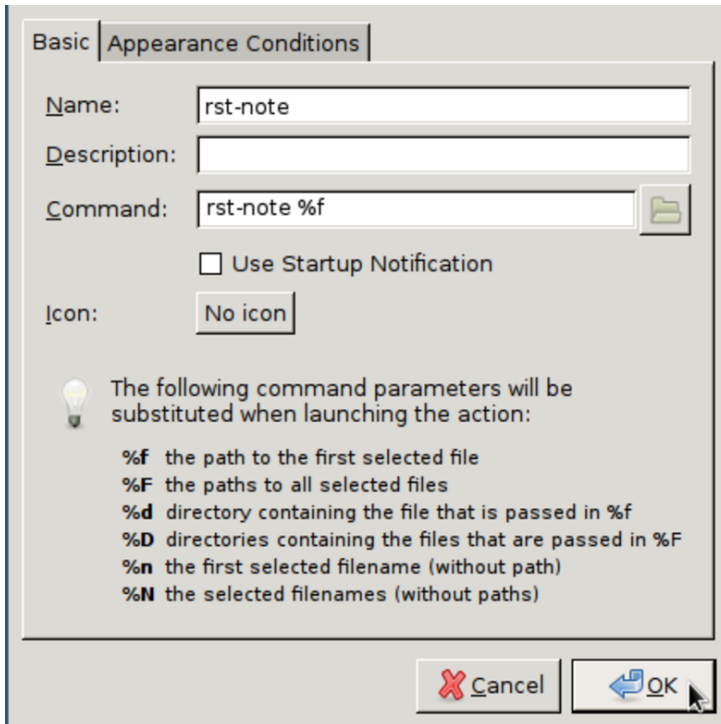


Figure 15. Edit custom action

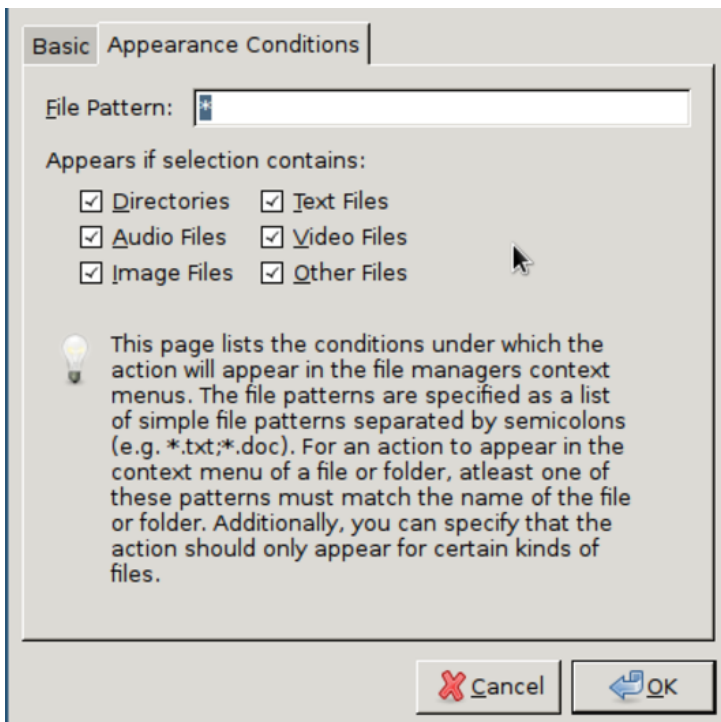
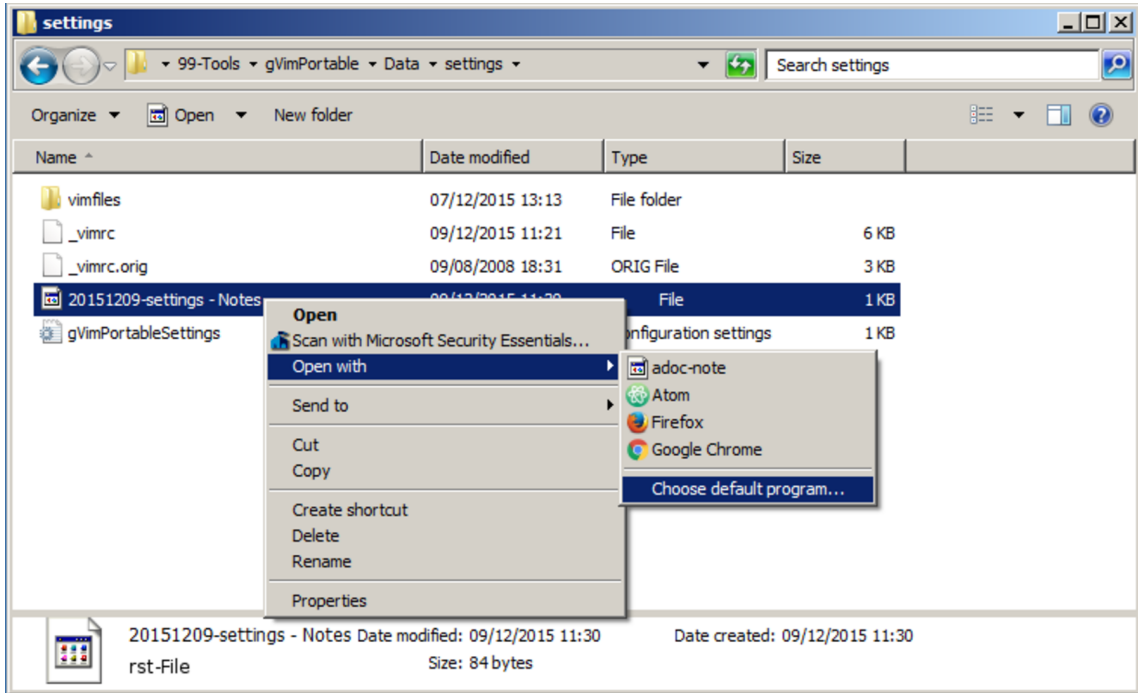


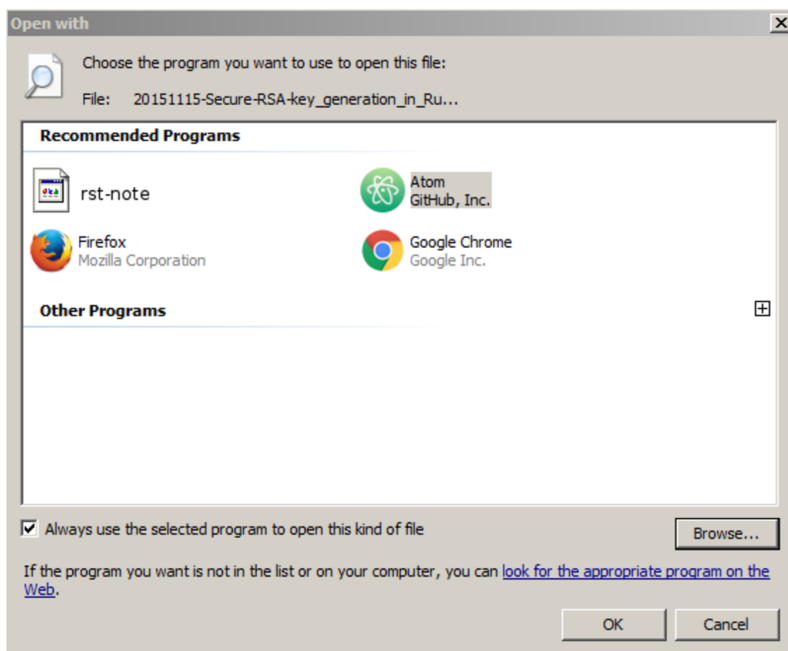
Figure 16. Appearance Condition

Windows file explorer configuration

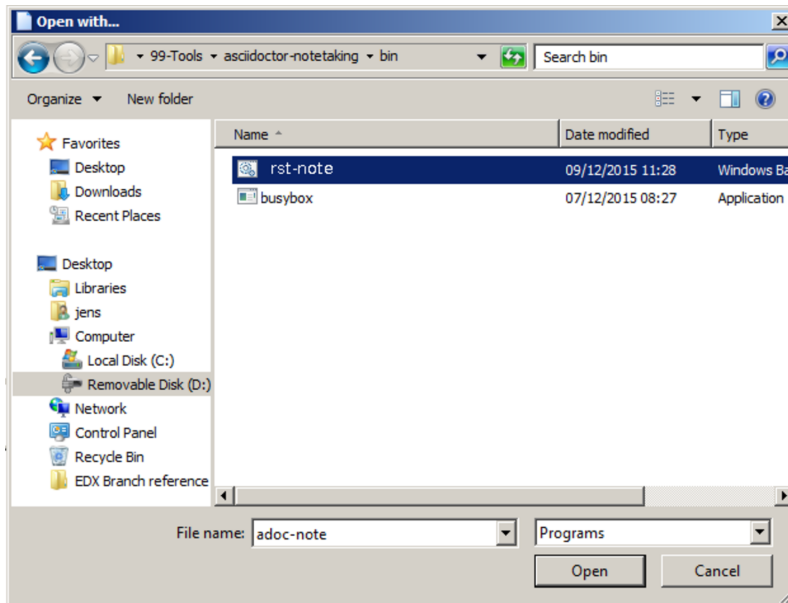
1. Open a folder containing an `.adoc` file.
2. Right-click the `.adoc` file and point to *Open with* and then click *Choose default program*.



3. Select the *Always use the selected program* and then click *Browse...*



4. Click *Browse...* then browse to your `BIN_DIR` directory, select `adoc-note.bak` and click *Open* and later *Ok*.



More information

- [Change the program that opens a type of file²¹](#)
- [How to Add Any Application Shortcut to Windows Explorer's Context Menu²²](#).

²¹ <http://windows.microsoft.com/en-us/windows/change-file-open-program#1TC=windows-7>

²² <http://www.howtogeek.com/107965/how-to-add-any-application-shortcut-to-windows-explorers-context-menu/>

7. Source code

7.1. src/adoc-note

Source code: adoc-note

```
#!/bin/sh
# Author: Jens Getreu
# 16.2.2016
# license: MIT

### CONFIGURATION SECTION START

LaunchViewer () {
# WARNING: if the environment variable ADOC_VIEWER is defined
# the following code is never executed!

# Note: we need & at the end of the line!
# Uncomment one line only!
# Chromium AsciiDoctor-plugin follows your changes without "reload".
#chromium "file:/// $1" &
# Alternative:
#chrome "file:/// $1" &
# Extra package needed for this example: apt-get install uni2ascii
#chromium "--app=file:$(echo "$1"|uni2ascii -aJ)" &
# Press "reload current page" in firefox to see your changes.
firefox "file:/// $1" &
# If your editor has an integrated viewer (like Atom)
# do not uncomment any line above, but we need this:
return 0
}

LaunchEditor () {
# WARNING: if the environment variable ADOC_EDITOR is defined
# the following code is never executed!

# Note there is no & in this function. The editor should not fork!
# Uncomment one line only!

# Very simple Linux editor.
leafpad "$1"
# Linux geany. Optional: enable autosave in geany's save plugin.
#geany "$1"
# Alternatively run gvim.
#gvim --nofork "$1"
```

```
return 0
}

### CONFIGURATION SECTION END

Readlink () {
# Path=$(readlink -f "$1") # not available in busybox
# We actually only need absolute paths so the following will do.
local Path
Path="$(echo "$(cd "$(dirname "$1")" && pwd -P)"/"${(basename "$1")}")"
# return global variable
Readlink="$Path"
}

Main () {
if [ "-h" = "$1" ]; then
echo "\n${0} creates, edits or views an asciidoc note."
echo "\nusage:"
echo "\n  $_ [-h][-ro|-so|-eo] | [<File.adoc>|<Dir>|<File>]"
echo "\n<Dir>|<File>: directory where the new note file will be created"
echo "(current directory if none)."
echo "If <File> is given a new adoc note will be created next to that
file."
echo "If <File.adoc> is given the file is edited."
echo "Filename of <File> is changed when not in sync with title."
echo "\nOptions:"
echo "-ro\tDo not open editor, open viewer only."
echo "-eo\tDo not open viewer, only new note or, sync filename and
edit."
echo "-so\tDo not open editor or viewer, only new note or sync
filename."
exit 0
fi

local Path
local Option
if [ "-ro" = "$1" ] || [ "-so" = "$1" ] || [ "-eo" = "$1" ] ; then
Option="$1"
Readlink "$2"
else
Option=""

```

```
Readlink "$1"
fi
Path="$Readlink"

# If file extension is not .adoc then create a new note.
if [ ! -n "$Path" ] || \
 [ ! "$(echo "$Path"|awk -F . '{print $NF}')" = "adoc" ] ; then
  AdocNewNote "$Path" || exit 1
  Path="$AdocNewNote"
fi

if [ -f "$Path" ] ; then
  if [ ! "-ro" = "$Option" ] ; then
    SyncFilename "$Path"
    Path="$SyncFilename"
  fi
  if [ ! "-eo" = "$Option" ] && \
 [ ! "-so" = "$Option" ] ; then
    AdocView "$Path"
  fi
  if [ ! "-ro" = "$Option" ] && \
 [ ! "-so" = "$Option" ] ; then
    AdocEdit "$Path"
    SyncFilename "$AdocEdit"
    Path="$SyncFilename"
  fi
  echo "$Path"
  exit 0
else
  echo "Error: Can not open '$Path'." >&2
  exit 1
fi
}

SanitizeFilename () {
# line 1:  tab -> space
# line 2:  Delete control characters.
# line 3:  :\\/?~,;=  ->  _
# line 4:
# Exclude NTFS critical characters:      <>:"\\/?*
# https://msdn.microsoft.com/en-us/library/windows/desktop/
aa365247%28v=vs.85%29.aspx
# Exclude restricted in fat32:           +,;=[]
# https://en.wikipedia.org/wiki/Filename#Reserved_characters_and_words
# These are considered unsafe in URLs:   <>#%{ }|\^~[]`
```

```
# https://perishablepress.com/stop-using-unsafe-characters-in-urls/
# line 5: Strip all until the first alpha char at the beginning
#         and all spaces and _ at the end of the line.
# line 6: Remove spaces and _ before and after --.
# Return global variable
SanitizeFilename="$(echo "$1" | \
tr -s '[:blank:]' ' ' | \
tr -d '[:cntrl:]' | \
tr -s ':\\/|?~,;=' ' _' | \
tr -s '<>:"\||?*<>#%{}|\^~[+ ,;=[]`[:blank:]' ' ' | \
sed -e 's/[[:blank:]]*\.*/\1/g; s/[[:blank:]]*$/g' \
-e 's/[[:blank:]]*--[[:blank:]]*/--/g'
)"
}
```

```
SyncFilename () {
# Pathname of the asciidoc-file
local Path
Path="$1"

# Change filename according to the title in the
# first line of the .adoc file.

# Extract title from the first line of the .adoc file
local FirstLine
FirstLine="$(head -n 1 "$Path")" || exit 1
local Title
# line 1: stream string
# line 2: omit BOM and =
# line 3: splitt title and subtitle, then concat with -- in between
Title="$(echo "$FirstLine" | \
sed -e 's/[^=]*=[[:blank:]]*\.*/\1/' \
-e 's/[[:blank:]]*$/g; s/\(.*\):\(.*)/\1--\2/g' )"
# A title was found when Title <> FirstLine
if [ ! "$FirstLine" = "$Title" ] ; then

# extract leading numbers including "-" and "_"
# example "20150912-hallo.adoc" -> "20150912-"
# example "01-abstract.adoc" -> "01-"
# example "04-01_03-abstract.adoc" -> "04-01_03-"
# example "hallo02-abstract.adoc" -> ""
local BaseName
BaseName="$(basename "$Path")"
local LeadingNumbers
LeadingNumbers="$(echo "$BaseName" | \
```

```
sed -e 's/\([[[:digit:]]_]*\)\"(.*)\"/\1/'"
# if no leading numbers found the following equals
if [ "$BaseName" = "$LeadingNumbers" ] ; then
  LeadingNumbers="" #no leading no. found
fi

# Substitute special chars with _
SanitizeFilename "$Title"
local FileTitle
FileTitle="$(echo "$SanitizeFilename"|
  sed -e 's/\([[[:digit:]]_]*\)\"(.*)\"/\2/')"

# Construct new name
local DirName
DirName="$(dirname "$Path)"
local NewPath="$DirName}/${LeadingNumbers}${FileTitle}.adoc"
mv -n "$Path" "$NewPath" >/dev/null 2>&1
Path="$NewPath"
fi

# Return global variable
SyncFilename="$Path"
}
```

```
AdocNewNote () {
  local Suffix="Notes"
  local TitleSuffix=" : $Suffix"
  local FileSuffix="--$Suffix"
  local Dir
  local Basename
  local DocRef
  local NewFileName
  if [ -n "$1" ] && [ -f "$1" ] ; then
    Dir="$(dirname "$1")"
    Basename="$(basename "$1")"
    DocRef="Annotations on file: link:++$Basename++[]"
    SanitizeFilename "$Basename"
    NewFileName="$Dir/${SanitizeFilename}${FileSuffix}.adoc"
  elif [ -n "$1" ] && [ -d "$1" ] ; then
    Dir="$1"
    Basename="$(basename "$1")"
    DocRef=""
    # omit leading numbers, "-" and "_"
  fi
}
```

```

    Basename="$(echo "$Basename" | \
        sed -e 's/\([[[:digit:]]_]*\)\.*/\2/')"
    SanitizeFilename "$Basename"
    NewFileName="$Dir/$(date +%Y%m%d)-${SanitizeFilename}${FileSuffix}.adoc"
else
    Dir="$(pwd)"
    Basename="$(basename "$Dir")"
    DocRef=""
    # omit leading numbers, "-" and "_"
    Basename="$(echo "$Basename" | \
        sed -e 's/\([[[:digit:]]_]*\)\.*/\2/')"
    SanitizeFilename "$Basename"
    NewFileName="$Dir/$(date +%Y%m%d)-${SanitizeFilename}${FileSuffix}.adoc"
fi
local Datestr
Datestr="$(date +%x)"

if [ ! -e "$NewFileName" ] ; then

    # Create new file according to template (with unicode BOM)
    printf "\357\273\277= ${Basename}${TitleSuffix}
$USER
v1.0, $Datestr
:numbered!:
:toc!:
:pagenums:

$DocRef

" > "$NewFileName"

else
    echo "$NewFileName already exists. No new document created.">&2
fi
# return global variable
AdocNewNote="$NewFileName"
}

AdocEdit () {
    AdocEdit="$1"
    if [ -n "$ADOC_EDITOR" ]; then
        "$ADOC_EDITOR" "$AdocEdit"
    else

```

```
    LaunchEditor "$AdocEdit"  
  fi  
  # Return global variable AdocEdit  
}
```

```
AdocView () {  
  AdocView="$1"  
  if [ -n "$ADOC_VIEWER" ]; then  
    "$ADOC_VIEWER" "$ADOC_VIEWER_OPT$AdocView" &  
  else  
    LaunchViewer "$AdocView"  
  fi  
  # Return global variable AdocView  
}
```

```
Main "$1" "$2"
```

7.2. src/adoc-note.bat

Source code: adoc-note.bat

```
@echo off > NUL  
  
rem CONFIGURATION SECTION START (WINDOWS ONLY)  
  
rem uncomment TWO lines only  
set "ADOC_VIEWER=C:\Program Files (x86)\Mozilla Firefox\firefox.exe"  
set "ADOC_VIEWER_OPT=file:///"  
rem set "ADOC_VIEWER=C:\Program Files\Mozilla Firefox\firefox.exe"  
rem set "ADOC_VIEWER_OPT=file:///"  
rem set "ADOC_VIEWER=C:\Program Files (x86)\Google\Chrome\Application  
\Chrome.exe"  
rem set "ADOC_VIEWER_OPT=file:///"  
  
rem uncomment one line only  
set "ADOC_EDITOR=C:\Program Files (x86)\Windows NT\Accessories  
\Wordpad.exe"  
rem set "ADOC_EDITOR=C:\windows\system32\wordpad.exe"  
rem set "ADOC_EDITOR=C:\Program Files (x86)\Vim\vim74\gvim.exe"  
rem set "ADOC_EDITOR=D:\99-Tools\gVimPortable\gVimPortable.exe"
```

```
rem CONFIGURATION SECTION END (WINDOWS ONLY)
```

```
"%~dp0\busybox" ash "%~dp0\adoc-note" "%~1"
```
