

TREAT DOCUMENTATION AS CODE

hagen.bauer@rusticus-consulting.de @hagen_bauer

ABOUT ME

- Hagen Bauer
- IT Consultant Information Security
- Privacy Advocate
- Serveradmin
- Web Shop Owner



REQUIREMENTS FOR COMPLEX DOCUMENTATION

- Input from various sources
 - text, csv input, charts, pdf reports
- different versions
- multiple authors different output formats (PDF, Word, HTML ...)
- different content for different reader groups

DEMO

IT SECURITY CONCEPTS

SOLUTION: TREAD DOCUMENTATION LIKE CODE

- Write plain text files
- Keep versions in version control systems
- Use continuous integration for document compilation

ASCIIDOCTOR ALLOWS FOR THIS

VOCABULARY

- AsciiDoc = Text Format
 - A plain-text writing format for authoring articles, documentation, books, web pages, blog posts, man pages and more.
- AsciiDoctor = Tooling
 - AsciiDoctor is a fast, open source text processor and publishing toolchain for converting AsciiDoc content to HTML5 PDF and other formats.

TYPICAL WORKFLOW?

- Use your favorite text editor (z.b. Atom, Vim...)
- Check in version in Git
- Automatic creation of target documents
- Local / Git CI

TEXT FORMAT

```
= AsciiDoc is Writing Zen
Doc Writer <doc.writer@example.com>
:icons: font

_Zen_ in the art of writing `plain text` with
http://asciidoc.org[AsciiDoc].

[TIP]
Use http://asciidoctor.org[Asciidoctor] for the best AsciiDoc
experience.footnote:[Not to mention the best looking output!]
Then icon:twitter[role=aqua] about it!

== Sample Section

[square]
* item 1
* item 2

[source,ruby]
----
puts "Hello, World!"
----
```

AsciiDoc source

AsciiDoc is Writing Zen

Doc Writer – doc.writer@example.com

Zen in the **art** of writing **plain text** with [AsciiDoc](#).



Use [Asciidoctor](#) for the best AsciiDoc experience.^[1] Then  about it!

Sample Section

- item 1
- item 2

```
puts "Hello, World!"
```

¹ Not to mention the best looking output!

Rendered HTML

STUFF I LOVE THE MOST

- Possible Post/Pre Processing
- Different output
- Syntax Highlighting
- Conditional inclusion
- Comments
- Attributes

OPTIONAL PRE/POST PROCESSING

- since we are on the command line
 - Massage inputs via sed/grep/awk....
 - Create graphs via gnuplot
 - Combine PDF with consistent page numbering
 - Add Watermarks

OUTPUT FORMAT

- native:
 - html5, xhtml, docbook, manpage
- addon:
 - pdf, epub3, latex
- pandoc:
 - the rest

SYNTAX HIGHLIGHTING

```
.app.rb  
[#src-listing]  
[source,ruby]  
----  
require 'sinatra'  
  
get '/hi' do  
  "Hello World!"  
end  
----
```

```
app.rb  
  
require 'sinatra'  
  
get '/hi' do  
  "Hello World!"  
end
```

ATTRIBUTES

activate features (behaviors, styles, integrations, etc)
or hold replacement (i.e., variable) content

```
= Document Title  
:imagesdir: new/path/to/images  
  
asciidoctor -a output=internal doc.adoc
```

DYNAMIC INCLUSION

include all or parts of external documents/data

```
include::filename.txt[lines="1..10,15..20"]
```

```
include::big-configuration-file.ini[tags=customer]
```

```
include::https://raw.githubusercontent.com/asciidoctor/asciidocto
```

CONDITIONAL INCLUSION

Define attributes the control part of your content

```
ifdef::linux[]  
Linux Server will have to add .....  
endif::linux[]
```


COMMENTS

you ever wanted to write down stuff that is not part of the final result?

```
// A single-line comment.  
  
////  
A multi-line comment.  
  
Notice it's a delimited block.  
////
```

DEMO

(WARNING: IMPLEMENTATION DETAILS SHOULD NOT BE REUSED)

TREAT DOCUMENTATION AS CODE

hagen.bauer@rusticus-consulting.de @hagen_bauer