

AsciiDoc NetBeans Plugin User Guide

Release 2.0.2: Early Preview Edition

Table of Contents

Introduction.....	1
Examples of the AsciiDoctor NB-Plugin.....	2
Basic file recognition.....	2
Publish a pdf article.....	2
Publish an html article.....	2
To repurpose an asciidoc file for both pdf and html publishing.....	2
The project structure.....	3
To assemble multiple asciidoc files for pdf and/or html publishing.....	3
Utilising the AsciiDoc NBM.....	5
Installing the NBM.....	5
Links.....	5

Introduction

The AsciiDoc NetBeans plugin is designed to support the publishing of AsciiDoc files using the `asciidocctor` and `asciidocctor-pdf` executables.

In its most basic form, it will provide a publish action for any `.adoc` file, which will create either an html or pdf document (depending upon the header of the document). The resulting published file will be placed in the same folder as the source `.doc` file.

An AsciiDoc Project is defined by the presence of an `asciidoc.properties` file at the root level of the project. There will also be a folder at the root level (named `src`) which will contain the asciidoc files which make up the project and associated resources. Any folder structure within the `src` folder is acceptable. All asciidoc will have a publishing action (as per the basic form of operation). However the published documents will be placed within another root level folder (named `generated_documents`), placed with a folder structure that reflects the source folder structure.

AsciiDoc projects are suited to publishing an assembly of AsciiDoc files and/or repurposing documents so that they can be published in multiple format. However they can be used as containers for a collection of individual documents.

CAUTION

The predefined `.adoc` file actions are limited to unix family systems (ie Linux and MacOS). No Windows implementation have yet been completed.

Examples of the AsciiDoctor NB-Plugin

Basic file recognition

Any AsciiDoc file (.adoc) will be recognised by Netbeans and will have a Publish action included in it's pull down menu. If the file is not located within a AsciiDoc project then the Publish action will create the published document in a same folder as the source.

ADD AN IMAGE HERE

Publish a pdf article

To publish an asciidoctor-pdf article your asciidoc file will need to define options in it's header

```
:doctype: book
:backend: pdf
:toc: ①
:revnumber: 2.0.2 ①
:revremark: Early Preview Edition ①
:version-label: Release ①
= AsciiDoc NetBeans Plugin User Guide
```

① optional options

Publish an html article

To publish an asciidoctor article your asciidoc file will need to define options in it's header

```
:toc2: ①
:revnumber: 2.0.2 ①
:revremark: Early Preview Edition ①
:version-label: Release ①
= AsciiDoc NetBeans Plugin User Guide
```

① optional options

To repurpose an asciidoc file for both pdf and html publishing

To repurpose a file for multiple publishing formats, it can be easily handled by creating 3 files - the source file (without headers) and two file which contain the necessary options headers and an include statement to add the actual source.

ADD AN IMAGE HERE

In *book.adoc* set the options for a book and use an include to add the source.

```
:doctype: book
:backend: pdf
:toc:
:revnumber: 2.0.2
:revremark: Early Preview Edition
:version-label: Release
= AsciiDoc NetBeans Plugin User Guide

include::userguide.adoc[]
```

In *web.adoc* set the options for an html document and use an include to add the same source.

```
:toc2:
:revnumber: 2.0.2
:revremark: Early Preview Edition
:version-label: Release
= AsciiDoc NetBeans Plugin User Guide

include::userguide.adoc[]
```

The project structure

The project structure that is recognised by this plug-in is:

- an *asciidoc.properties* file (no content is required).
- asciidoc files, located within the src folder
- resources (images etc) to be used in the Publishing process

ADD AN IMAGE HERE image::resources/projectstructure.png[project structure]

The publishing process will create a *generated_documents* folder.

There are no specific project level Actions.

The standard output of the Publishing command is routed to the NetBeans Output Window, while the published document is created in the *generated_documents* folder.

To assemble multiple asciidoc files for pdf and/or html publishing

As one of the major use cases for an AsciiDoc project is to be able to assemble multiple AsciiDoc source files into a book and/or html file.

So developing the previous examples, an asciidoc file for each output is created which define the

options and a series of include statements to define the assembly order.

```
:doctype: book
:backend: pdf
:toc:
:revnumber: 2.0.2
:revremark: Early Preview Edition
:version-label: Release
= AsciiDoc NetBeans Plugin User Guide

include::introduction.adoc[]

include::example.adoc[]

include::availability.adoc[]
```

or if you have the source in a subfolder structure.

```
:doctype: book
:backend: pdf
:toc:
:revnumber: 2.0.2
:revremark: Early Preview Edition
:version-label: Release
= AsciiDoc NetBeans Plugin User Guide

:imagesdir: part1/
include::part1/introduction.adoc[]

include::part1/example.adoc[]

:imagesdir: part2/
include::part2/availability.adoc[]
```

and html version

```
:toc2:
:revnumber: 2.0.2
:revremark: Early Preview Edition
:version-label: Release
= AsciiDoc NetBeans Plugin User Guide

include::part1/introduction.adoc[]

include::part1/example.adoc[]

include::part2/availability.adoc[]
```

Utilising the AsciiDoc NBM

Installing the NBM

To be able to use the Plugin it must be downloaded and then installed in Netbeans.

Releases of this product are stored in a Maven repository on GitHub. The NBM can be downloaded from the Package available.

Once you have a downloaded copy on your machine, you follow the standard Netbeans process to install the downloaded file (Tools>Plugins). The [ActionsSupport NBM](#) must also be installed with or prior to installing this NBM.

Links

- Releases are stored in a Packages Maven repository on [GitHub](#)
- Source Code is stored in a Git repository on [GitHub](#) as part of a monorepo containing NBMs.
- Documentation can be found on the [TRP website](#).
- Documentation source is stored in a Git repository on [GitHub](#) as part of a monorepo containing documentation.
- Development Process Guide for The-Retired-Programmer products can be found on the [TRP website](#).