

AsciiDoc NetBeans Plugin User Guide

Release 2.1.1

Table of Contents

Introduction.....	1
AsciiDoctor NB-Plugin by Example	2
AsciiDoc file recognition.....	2
Publish a pdf book.....	2
Publish an html article	2
To repurpose an AsciiDoc file for both pdf and html publishing.....	3
The project structure.....	4
Using alternative source and generated documents folders.....	5
To assemble multiple AsciiDoc files for pdf and/or html publishing.....	5
Save Before Publishing	6
Using the AsciiDoc NBM	8
Adding the NBM to NetBeans	8
Useful Links	8

Introduction

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

In non Project Mode, it provides a publish action for any `.adoc` file, which can create either an html or pdf document (depending upon the header of the document). The resulting file will be placed in the same folder as the `.adoc` file.

Publishing actions are in the file node's popup actions, and also in the editor window's popup actions.

The NetBeans spelling checker is enabled when editing an AsciiDoc file.

Automatically saving modified documents before Publishing is available. It is always enabled in non Project Mode.

In Project Mode, an AsciiDoc Project is recognised 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 their associated resources. Any folder structure within the `src` folder is acceptable. All `.adoc` files will have a publishing action. The published documents will be placed within another root level folder (named `generated_documents`). This will include a folder structure that reflects the source folder structure.

The default source root folder and `generated_documents` root folder can be redefined by use of properties in the `asciidoc.properties` file.

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

Automatically saving modified documents before Publishing is available in Project Mode. It is optional and is enabled/disabled by a property in the `asciidoc.properties` file. By default it is enabled.

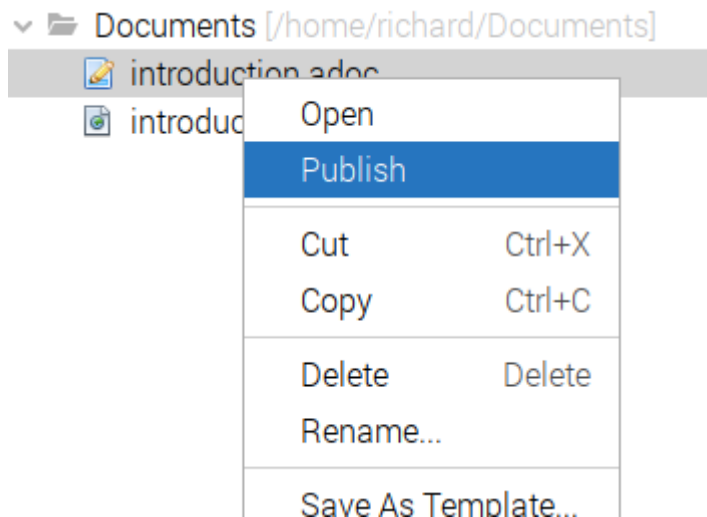
CAUTION

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

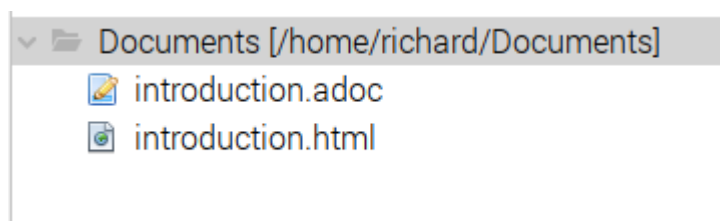
AsciiDoctor NB-Plugin by Example

AsciiDoc file recognition

Any AsciiDoc file (*.adoc*) will be recognised by NetBeans and will have a Publish action included in it's popup 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.



Publish a pdf book

To publish an asciidoctor-pdf book 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

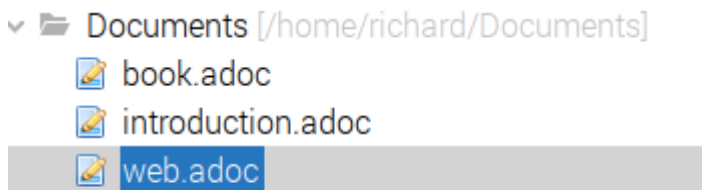
To publish an asciidoctor article your AsciiDoc file will need to define options in it's header

```
: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 to publish in multiple formats, create 3 files - the source file (without headers) and two files which contain the necessary headers and an include statement to add the source.

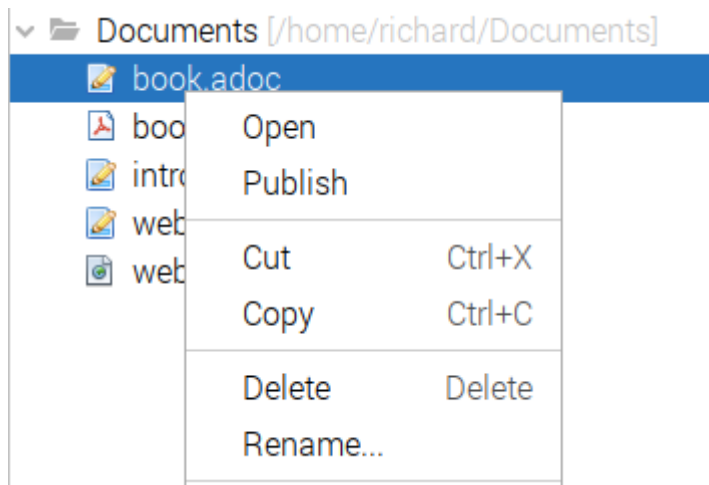


In *book.adoc* set the options for a book and an include statement.

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

include::introduction.adoc[]
```

Select Publish on the *book.adoc* file to create the pdf.

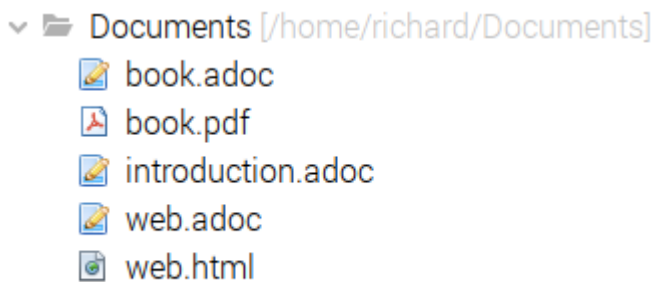


In *web.adoc* set the options for an html document and an include statement. Select Publish on the *web.adoc* file to create the html.

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

include::introduction.adoc[]
```

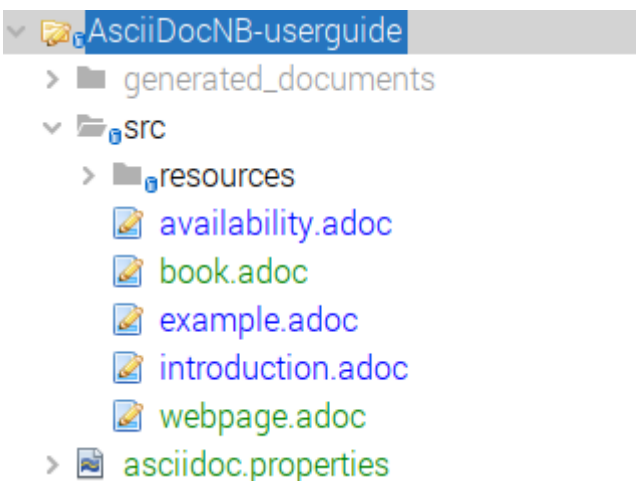
The resulting folder structure which will contain the two published documents:



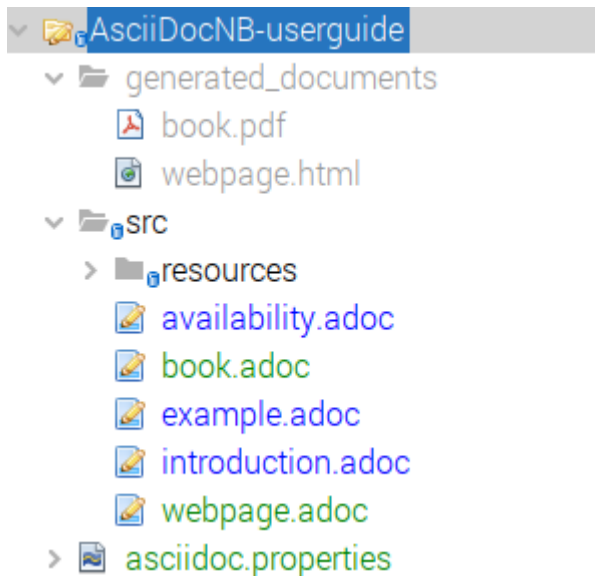
The project structure

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

- an *asciidoc.properties* file (content is optional).
- asciidoc files, located within the src folder
- resources (images etc) to be used in the Publishing process



The publishing process will create a *generated_documents* folder.



There are no specific project level Actions.

The standard output and error streams are routed to the NetBeans Output Window, while the published document is created in the *generated_documents* folder.

Using alternative source and generated documents folders

While there are default folders (*src* and *generated_documents*) it is possible to change these folders by addition of properties in the *asciidoc.properties* file.

```
source_root_folder: sources ①  
generated_root_folder: books_and_pages ②
```

① source root folder - relative to the project root folder

② generated documents root folder - relative to the project root folder

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

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.

Using similar techniques seen in previous examples, an AsciiDoc file for each output is created, defining the options, and now a series of include statements 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[]
```

Save Before Publishing

Save Before Publishing, if enabled, saves any AsciiDoc files which have been modified but not saved, prior to executing the Publish action.

In non Project Mode this feature is always enabled for the file which the Publish Action has been

applied.

In Project Mode the feature is controlled by a property in the *asciidoc.properties* file.

```
save_before_publishing: XXX ①
```

① where XXX is one of NO, YES or ALL. This property is optional and its default value is ALL.

This property controls how any modified files may be saved prior to running the publish action. Only files modified by Netbeans will be considered as a Save candidates.

The possible values for the property are:

- NO - No modified file(s) are saved prior to publishing, so the publish action will use the version of files as last saved (ignoring the modifications).
- YES - Only the file selected by the publish action (either via the file node or the editor action), will be saved, if modified, prior to running the publish action.
- ALL - All modified files that are within the source folder will be saved prior to running the publish action.

Using the AsciiDoc NBM

Adding the NBM to NetBeans

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. It is possible to download an NBM from Packages.

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 be installed with or prior to installing this NBM.

Useful 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](#).