



# **AsciiDoc reference**

## **Contributor's Guide**

Ben Cammett, Aksel Davis  
April 23, 2021

This PDF was generated from [https://docs.netapp.com/us-en/contribute/asciidoc\\_syntax.html](https://docs.netapp.com/us-en/contribute/asciidoc_syntax.html) on May 31, 2021. Always check docs.netapp.com for the latest.

# Table of Contents

- AsciiDoc reference ..... 1
  - The basics ..... 1
  - Advanced stuff ..... 5

# AsciiDoc reference

AsciiDoc is a lightweight markup language, similar to Markdown. We chose AsciiDoc over standard Markdown because it provides more out-of-box capabilities. While it's more powerful, it's still simple to use. Refer to this page to get started writing in AsciiDoc.

See the [AsciiDoctor User Manual](#) for additional help.

## The basics

You need to know a few things to contribute simple doc updates.

### Headings

```
= Page title
== Level 1 section
=== Level 2 section
==== Level 3 section
===== Level 4 section
```

You can have only one page title, but you can have multiple section titles. For example, you might have three level 1 sections that include level 2 and 3 sections:

```
= Page title
== Level 1 section
=== Level 2 section
== Level 1 section
== Level 1 section
=== Level 2 section
==== Level 3 section
```

### Bold text

```
*Text*
```

### Italic text

```
_Text_
```

### Bulleted lists

```
* Item 1
+
Continuation text for the previous list item.

* Item 2
** Item 2a

* Item 3
```



The + is a list continuation. It keeps the text inline with the list item. Omitting the + affects the formatting of that line.

## Labeled lists

```
Item 1::
Description 1

Item 2::
Description 2
```

or

```
[horizontal]
Item 1::
Description 1

Item 2::
Description 2
```

When you add [horizontal] above item 1, the label and description appear on the same line. That works well when you have very short descriptions.

### Example without [horizontal]

#### Item 1

Description 1

#### Item 2

Description 2

### Example with [horizontal]

**Item 1**    Description 1

## Steps

```
.Steps

. Step 1

. Step 2
+
Info for step 2

. Step 3
.. Step 3a
.. Step 3b

. Step 4
```



The + is a list continuation. It keeps the text inline with the list item. Omitting the + affects the formatting of that line.

## Images

```
image::file.gif["alt text"]
```

*alt text* means alternative text. It describes the image that appears on the page. The primary use is for visually-impaired users who use screen readers.



It's best to enclose alt text in quotes because punctuation like commas can affect the ability to transform the content from AsciiDoc to HTML.

## Videos

Hosted on YouTube:

```
video::id[youtube]
```

Hosted locally in GitHub:

```
video::https://docs.netapp.com/us-en/contribute//media/file.mp4
```

## Links

The syntax that you should use depends on what you're linking to:

- [Link to an external site](#)
- [Link to a section on the same page](#)
- [Link to another page in the docs](#)

### Link to an external site

```
url[link text^]
```

The ^ opens the link in a new browser tab.

### Link to a section on the same page

```
<<section_title>>
```

For example:

```
For more details, see <<Headings>>.
```

The link text can be something other than the section title:

```
<<section_title>,Different link text>>
```

For example:

```
<<Headings>,Learn the syntax for headings>>.
```

### Link to another page in the docs

The file needs to be in the same GitHub repository:

```
xref:{relative_path}<file_name>.html[Link text]
```

To link directly to a section in the file, add a hash (#) and the section's title:

```
xref:{relative_path}<file_name>.html#<section-name-using-dashes-and-all-lower-case>[Link text]
```

For example:

```
xref:{relative_path}style.html#use-simple-words[Use simple words]
```

## Notes, tips, and cautions

You might want to draw attention to certain statements by using notes, tips, or caution statements. Format them as follows:

```
NOTE: text
```

```
TIP: text
```

```
CAUTION: text
```

Use each of these sparingly. You don't want to create pages that are full of notes and tips. They become less meaningful if you do.

Here's what each of these looks like when the AsciiDoc content is turned into HTML:



This is a note. It includes extra info that a reader might need to know.



A tip provides useful information that can help a user do something or understand something.



A caution advises the reader to act carefully. Use this in rare circumstances.

## Advanced stuff

If you're authoring new content, you'll want to review this section for some nitty-gritty details.

### Document headers

Each AsciiDoc file includes two types of headers. The first is for GitHub and the second is for AsciiDoctor, which is the publishing tool that turns the AsciiDoc content into HTML.

The GitHub header is the very first set of content in the .adoc file. It needs to include the following:

```
---
sidebar: sidebar
permalink: <file_name>.html
keywords: keyword1, keyword2, keyword3, keyword4, keyword5
summary: "A summary."
---
```

The keywords and summary directly affect search results. In fact, the summary itself displays in the search results. You should make sure that it's user friendly. The best practice is to have the summary mirror your lead paragraph.



It's best to enclose the summary in quotes because punctuation like colons can affect the ability to transform the content from AsciiDoc to HTML.

The next header goes directly underneath the document title (see [Headings](#)). This header should include the following:

```
:hardbreaks:
:nofooter:
:icons: font
:linkattrs:
:relative_path: ./
:imagesdir: {root_path}{relative_path}./media/
```

You won't need to touch any of the parameters in this heading. Just paste it in and forget it.

## Lead paragraph

The first paragraph that appears under the document title should include the following syntax directly above it:

```
[.lead]
This is my lead paragraph for this content.
```

[.lead] applies CSS formatting to the lead paragraph, which has a different format than the text that follows it.

## Tables

Here's syntax for a basic table:

```
[cols=2*,options="header",cols="25,75"]
|===
| heading column 1
| heading column 2
| row 1 column 1 | row 1 column 2
| row 2 column 1 | row 2 column 2
|===
```

There are *many* ways to format a table. Refer to the [AsciiDoctor User Manual](#) for additional help.



If a cell contains formatted content like bulleted lists, it's best to add an "a" in the column header to enable formatting. For example:  
[cols="2,2,4a" options="header"]



[See the AsciiDoc Syntax Quick Reference for more table examples.](#)

## Task headings

If you're explaining how to perform a task, you might include introductory information before you get to the steps. And you might need to say what to do after completing the steps. If you do, it's best to organize that information using headers, which enables scanning.

Use the following headings as needed:

### **What you'll need**

*The information the user needs to complete the task.*

### **About this task**

*Some extra contextual info the user might need to know about this task.*

### **Steps**

*The individual steps to complete the task.*

### **What's next?**

*What the user should do next.*

Each of these should include a `.` right before the text, like so:

```
.What you'll need
>About this task
.Steps
.What's next?
```

This syntax applies bold text in a larger font.

## Command syntax

When providing command input, enclose the command within ``` to apply monospace font:

```
`volume show -is-encrypted true`
```

Here's what that looks like:

```
volume show -is-encrypted true
```

For command output or command examples, use the following syntax:

```

----
cluster2::> volume show -is-encrypted true

Vserver  Volume  Aggregate  State  Type  Size  Available  Used
-----  -
vs1      vol1    aggr2      online  RW   200GB   160.0GB   20%
-----

```

The four dashes enable you to enter separate lines of text that appear together.  
Here's the result:

```

cluster2::> volume show -is-encrypted true

Vserver  Volume  Aggregate  State  Type  Size  Available  Used
-----  -
vs1      vol1    aggr2      online  RW   200GB   160.0GB   20%
-----

```

## Variable text

In commands and command output, enclose variable text within underscores to apply italics.

```
`vserver nfs modify -vserver _name_ -showmount enabled`
```



Underscores aren't supported with code syntax highlighting at this time.

## Code syntax highlighting

Code syntax highlighting provides a developer-focused solution for documenting the most popular languages.

### Output example 1

```

POST https://netapp-cloud-account.auth0.com/oauth/token
Header: Content-Type: application/json
Body:
{
  "username": "<user_email>",
  "scope": "profile",
  "audience": "https://api.cloud.netapp.com",
  "client_id": "UaVhOIXMWQs5i1WdDxauXe5Mqkb34NJQ",
  "grant_type": "password",
  "password": "<user_password>"
}

```

## Output example 2

```
[
  {
    "header": {
      "requestId": "init",
      "clientId": "init",
      "agentId": "init"
    },
    "payload": {
      "init": {}
    },
    "id": "5801"
  }
]
```

## Supported languages

- bash
- curl
- https
- json
- powershell
- puppet
- python
- yaml

## Implementation

Copy and paste the following syntax and then add a supported language and the code:

```
[source,<language>]
<code>
```

For example:

```
[source,curl]
curl -s https:///v1/ \
-H accept:application/json \
-H "Content-type: application/json" \
-H api-key: \
-H secret-key: \
-X [GET,POST,PUT,DELETE]
```

## Content reuse

If you have a chunk of content that's repeated across different pages, you can easily write it once and reuse it across those pages. Reuse is possible from within the same repository and across repositories. Here's how it works.

1. Create a folder in your repository named `_include`

For example, take a look at the [Cloud Tiering repository](#).

2. Add a `.adoc` file in that folder that includes the content that you'd like to reuse.

It can be a sentence, a list, a table, one or more sections, and so on. Don't include anything else in the file—no headers or anything.

3. Now go to the files where you'd like to reuse that content.

4. If you're reusing the content from within the *same* GitHub repository, use the following syntax on a line by itself:

```
include::_include/<filename>.adoc[]
```

For example:

```
include::_include/s3regions.adoc[]
```

5. If you're reusing the content in a *different* repository, use the following syntax on a line by itself:

```
include::https://raw.githubusercontent.com/NetAppDocs/<reponame>/main/_include/<filename>.adoc[]
```

For example:

```
include::https://raw.githubusercontent.com/NetAppDocs/cloud-tiering/main/_include/s3regions.adoc[]
```

That's it!

If you want to learn more about the include directive, [check out the AsciiDoctor User Manual](#).

## Images with links

You can add a link to an image so it acts like a "button."

### Output example

In the following screenshot, "View in Documentation Center" is an image that contains a link.

# View information in the NetApp HCI Doc Center

Want details about NetApp HCI? See the NetApp HCI Documentation Center.



View in Documentation Center

## Syntax

Use the following syntax when adding your image:

```
image:<file_name>.<ext>[alt=<text>,link=<url>,window=_blank]
```

"window=\_blank" opens the link in a new browser tab (or window).

For example:

```
image:btn-view-in-doc-center.png[alt=A button titled View in Documentation  
Center that links to the NetApp HCI Doc  
Center,link=http://docs.netapp.com/hci/index.jsp,window=_blank]
```

## Copyright Information

Copyright © 2021 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system-without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## Trademark Information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.