Red Hat Reference Architecture Series

Style Guide and Template

Roger Lopez

Version 1.3, 2015-08-07

100 East Davie Street

Raleigh NC 27601 USA

Phone: +1 919 754 3700

Phone: 888 733 4281

Fax: +1 919 754 3701

PO Box 13588

Research Triangle Park NC 27709 USA

Send feedback to refarch-feedback@redhat.com

1. Comments and Feedback

In the spirit of open source, we invite anyone to provide feedback and comments on any reference architectures. Although we review our papers internally, sometimes issues or typographical errors are encountered. Feedback allows us to not only improve the quality of the papers we produce, but allows the reader to provide their thoughts on potential improvements and topic expansion to the papers. Feedback on the papers can be provided by emailing refarch-feedback@redhat.com. Please refer to the title within the email.

1.1. Staying In Touch

Join us on some of the popular social media sites where we keep our audience informed on new reference architectures as well as offer related information on things we find interesting.

1.1.1. Like us on Facebook:

https://www.facebook.com/rhrefarch

1.1.2. Follow us on Twitter:

https://twitter.com/RedHatRefArch

1.1.3. Plus us on Google+

https://plus.google.com/u/0/b/114152126783830728030/

2. Questions and Concerns About using Asciidoc

2.1. *Asciidoctor.org*

Read as many of these as time allows. This will greatly increase efficiency while writing asciidocs.

- Asciidoctor.org
- AsciiDoc cheatsheet
- Asciidoc Userguide

2.2. *How to install Asciidoctor*

Asciidoctor works on Linux, OSX (Mac) and Windows and requires one of the following implementations of Ruby:

- MRI (Ruby 1.8.7, 1.9.3, 2.0.0 & 2.1.2)
- JRuby 1.7 (Ruby 1.8 and 1.9 modes)
- Rubinius 2.2.x
- Opal (JavaScript)

There are several methods to install Asciidoctor as documented within GitHub's AsciiDoctor, however, the steps below are how I installed it. I use asciidoctor for creation of HTML docs.

With a terminal window, execute the following command

```
$ gem install asciidoctor
```

> If you have an earlier version of Asciidoctor installed, you can update it using:

```
$ gem update asciidoctor
```

> Within Fedora 18 or greater, you can install asciidoctor using `yum` however, I did not chose this route as on Fedora 20 the version is 0.1.4 and the latest version is 1.5.2

Once Asciidoctor gem installed successfully the asciidoctor command line interface (CLI) willb e available on your PATH. To verify it is available, run the following command:

```
$ asciidoctor --version
```

The info provided is the asciidoctor version and ruby environment version

```
Asciidoctor 1.5.2 [http://asciidoctor.org]
Runtime Environment (ruby 2.0.0p353 (2013-11-22 revision 43784) [i386-linux]) (lc:UTF-8
fs:UTF-8 in:- ex:UTF-8)
```

2.3. *How to install Asciidoctor-PDF*

To install asciidoctor-pdf (alpha software) follow the steps below.

My preferred method to install asciidoctor-pdf is to use RVM. RVM allows one to have separate *workspaces* called *gemsets* that key each Ruby environment completely isolated from another. It also doesn't require root access. The nice part of this since asciidoctor-pdf is still alpha, it allows me to keep different versions in different workspaces making it easier to test between dev, latest alpha and alpha-1 versions.

With a terminal window, execute the following command to setup an RVM environment. For more info on RVM, visit https://rvm.io

1. Import the key of the RVM team which will be used to validate archives:

```
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys
409B6B1796C275462A1703113804BB82D39DC0E3
```

2. Next, install the RVM script in your home directory (resides in $HOME/.rvm):

```
$ \curl -sSL https://get.rvm.io | bash -s stable
```

3. Source your bash profile

```
$ source ~/.bash_profile
```

4. Install at least one version of Ruby, version 2.2 is the one used.

```
$ rvm install 2.2
```

5. Finally, you'll be ready to start creating gemsets (aka workspaces). Here's how you create and switch to a new gemset:

```
$ rvm use 2.2@asciidoctor-pdf-dev --create
```

> After the gemset is created, the `--create flag` is not required. Each time you call `rvm use`, it will switch to an isolated workspace.

6. Once the development asciidoctor-pdf-dev workspace is created, git clone the repository to get the latest version.

7. Confirm the rvm workspace is using asciidoctor-pdf-dev via:

```
$ rvm current
ruby-2.2.1@asciidoctor-pdf-dev
```

8. Git clone the asciidoctor-pdf-dev repository

```
$ git clone https://github.com/asciidoctor/asciidoctor-pdf
```

9. Install bundler using gem

```
$ gem install bundler
```

10. Change directory into asciidoctor-pdf-dev directory.

```
$ cd /path/to/asciidoctor-pdf-dev
```

11. Use `bundle` command to install dependences

```
$ bundle
```

12. Install the syntax highlight source listings.

```
$ gem install coderay
$ gem install rouge
$ gem install pygments.rb
```

13. Confirm the required gems install properly, and verify the `asciidoctor-pdf` version

```
$ /path/to/asciidoctor-pdf/bin/asciidoctor-df -v
Asciidoctor PDF 1.5.0.dev using Asciidoctor 1.5.2 [http://asciidoctor.org]
Runtime Environment (ruby 2.2.1p85 (2015-02-26 revision 49769) [x86_64-linux])
(lc:UTF-8 fs:UTF-8 in:- ex:UTF-8)
```

14. This completes the steps to create and have the latest dev version of asciidoctor-pdf-dev. The next steps are to install a new workspace that has the latest alpha/beta stable release.

15. Once the development asciidoctor-pdf-dev workspace is created, go ahead and create the latest alpha version workspace.

> 🛈 | Latest version at the time of this writing is alpha9

```
$ rvm use 2.2@asciidoctor-pdf-alpha9 --create
```

16. Confirm rvm version

```
$ rvm current
ruby-2.2.1@asciidoctor-pdf-alpha9
```

17. Install asciidoctor-pdf

```
$ gem install --pre asciidoctor-pdf
```

18. Install the syntax highlight source listings.

```
$ gem install coderay
$ gem install rouge
$ gem install pygments.rb
```

> 🛈 | If you have an issue installing *pygments.rb*, it is probably due to requiring `ruby-devel` package.

19. Verify install of asciidoctor-pdf-alpha via

```
$ asciidoctor-pdf -v
Asciidoctor PDF 1.5.0.alpha.9 using Asciidoctor 1.5.2 [http://asciidoctor.org]
Runtime Environment (ruby 2.2.1p85 (2015-02-26 revision 49769) [x86_64-linux])
(lc:UTF-8 fs:UTF-8 in:- ex:UTF-8)
```

2.4. *Convert Asciidoc to PDF*

To convert your asciidoc to pdf, using ruby. Follow the steps below.

1. `cd` into the /path/to/asciidoctor-pdf/ , example below.

```
$ /HOMEDIR/.rvm/gems/ruby-2.2.1@asciidoctor-pdf-alpha9/gems/asciidoctor-pdf-
1.5.0.alpha.9/data/themes
```

2. Get the latest copy of the refarch template and place it in the data/themes directory of your specified rvm workspace

```
$ wget
https://gist.githubusercontent.com/rlopez133/600279abdc7d05ad91f4/raw/d86523c418855228
68d9f4303557df8fe94fbdd1/asciidoctor-theme.yml
```

> **ℹ** Within the theme file, you must fix the appropriate locations for your images.

3. Create a pdf from the asciidoc file.

```
$ ./bin/asciidoctor-pdf -a pdf-style=asciidoctor <name>.adoc
```

This will place a <name>.pdf into your current directory.

3. Style Guide

- System names should be short functional names.

- Avoid long references if possible – folder names, urls, filenames.

3.1. *Good References*

- Red Hat – WordUsage

  - https://home.corp.redhat.com/node/67041

- Grammar style guide

  - https://home.corp.redhat.com/wiki/red-hat-style-guide

- Grammar Questions

  - http://www.drgrammar.org/frequently-asked-questions

3.2. *Things to Avoid*

- Words

  - we

  - us

  - I

  - will

  - you

  - Two spaces next to each other

  - Misspellings

- Single item lists or sub-sections

  - This is a perfect example of what not to do

- Tense

  - Try not to write in past tense (e.g. did)

  - Try not to write in future tense (e.g. will)

  - Write in present tense.

- Inconsistency

  - Be consistent with the writing. Don't change from one style to another during the paper.

  - Ensure all items in a list end with a period or do not

  - Headings are similar.

- Do:Writing, Reading, Erasing

- Don't:Writing, How to read, Erasing

- Tables are the same width on all pages.

- Formatting

  - Never manually format by using inserted newlines or inserting page breaks

  - Instead, use "Format" menu and adjust items in this way

  - If the change is something that applies to everything in the document (e.g. All heading level 2 items should start on a new page), edit the style instead (better yet, change this in the template).

- Cross-References

  - Only use a cross-reference when the object to be referenced is not directly below or above the reference

  - See https://home.corp.redhat.com/node/49989

3.3. *Headers*

Heading 1 (Level 0)

```
= Heading 1 (Level 0)
```

Heading 2 (Level 1)

```
== Heading 2 (Level 1)
```

Heading 3 (Level 2)

```
=== Heading 3 (Level 2)
```

Heading 4 (Level 3)

```
==== Heading 4 (Level 3)
```

Heading 5 (Level 4)

```
===== Heading 5 (Level 4)
```

```
====== Heading 6 (Level 5)
```

3.4. *Entity Callouts*

An *entity* is basically a word or set of words that are not normal English terms that need to be called out in some manner. When using these words in normal paragraphs, they need to be called out with something like **bold**, *italics*, "quotes", or a different font. The point is to ensure that the reader is aware that these terms are special, and not to be confused with a normal English word. A simple example would be a sentence as such:

> When a node is detected to be not communicating with the cluster it needs to be fenced by fenced.

Without a callout of some kind, this could be confusing. However, with the command fenced in a different style it makes more sense:

Listing 1. Bold a word syntax

```
When a node is detected to be not communicating with the cluster it needs to be fenced by
*fenced*.
```

> When a node is detected to be not communicating with the cluster it needs to be fenced by **fenced**.

3.5. *Character Styles*

These styles only apply to a word or a set of characters.

3.5.1. *Files*

File names should appear in italics if in a paragraph.

Listing 2. Italizes a file name syntax

```
The _/etc/resolv.conf_ file configures name lookup settings.
```

Example 1. Result

> The */etc/resolv.conf* file configures name lookup settings.

3.5.2. *Software Name*

Callout a command, package name, channel name, project names, products, etc.

Listing 3. Software Callout Syntax

```
`*CloudForms*` is composed of `*Aeolus Conductor*`, `*Application Engine*`,
`*deltacloud*`, and many others.
```

Result: callout syntax showing in bold monospace font.

> `CloudForms` is composed of `Aeolus Conductor`, `Application Engine`, `deltacloudd`, and many others.

3.5.3. *UI Element*

Use when something needs to be clicked on a UI, selections need to be chosen, radio boxes, check boxes, input field names, etc.

Listing 4. Button Macro Syntax

```
Press the btn:[OK] button when you are finished.
Select a file in the file navigator and click btn:[Open].
```

Example 2. Result: macros displaying UI buttons

> Press the **[OK]** button when you are finished. Select a file in the file navigator and click **[Open]**.

3.5.4. *Important Term*

When defining a word, make the word(s) italized, and mono space.

Listing 5. Important Term Syntax

```
`_DM Multipath_` - Provides multipathing ...
```

> `DM Multipath` - Provides multipathing ...

3.5.5. *Variable Data*

User input that changes based on the circumstance.

```
Connect to the server via *_ssh username@domain.name_*
```

Connect to the server via ***ssh** username@domain.name*

3.5.6. *Generic Entity*

When none of these character styles apply but something needs to be called out, use *italics*.

3.6. Paragraph Styles

These styles apply to the entire paragraph.

3.6.1. *Code*

File contents are highlighted using a background color and `monospace font`.

- Command prompts should be simple, # for root, $ for users

- Command prompts should **not** be bolded.

- Commands should be **bolded.**

- Show the commands in a way that the reader can cut and paste them.

- Use long command line options instead of the single character options if possible.

```
# virsh list --all
 Id Name                    State
----------------------------------
  0 Domain-0                running
20 x-RHEL4.8-64-FV          idle
23 x-RHEL6.0-32-FV          idle
 - x-RHEL5.6-64-FV          shut off
 - x-RHEL5.6-PV             shut off
 - x-W2K3-64-FV             shut off
```

3.6.2. File Contents

Listing 6. File Contents Syntax

```
; generated by /sbin/dhclient-script
search cloud.lab.eng.bos.redhat.com
nameserver 10.16.143.247
nameserver 10.16.143.248
nameserver 10.16.255.2
```

```
; generated by /sbin/dhclient-script
search cloud.lab.eng.bos.redhat.com
nameserver 10.16.143.247
nameserver 10.16.143.248
nameserver 10.16.255.2
```

4. Tables

Should be consistent and same or similar style. If you use cross references for some tables, use them on all tables. Tables should not be the first item on a page.

Table 1. An example table

Col 1	Col 2	Col 3
1	Item 1	a
2	Item 2	b
3	Item 3	c
6	Three items	d

Table 2. CSV data, 15% each column

1	2	3	4
a	b	c	d
A	B	C	D

Table 3. Table Examples

ID	FName	LName	Address	Phone
1	Vasya	**Pupkin**	London	+123
2	X	**Y**	A,B	45678

Table 4. Multiline cells, row/col span

Date	Duration	Avg HR	Notes
22-Aug-08	10:24	157	Worked out MSHR (max sustainable heart rate) by going hard for this interval.
22-Aug-08		152	Back-to-back with previous interval.
24-Aug-08	none		

Table 5. RA-Column Heading

Column	Headings
This is	Data
And still	More Data
Data	

5. Figures

Should be consistent. If you use cross references for some figures, use them for all figures. Figures should not be the first item on a page. Use **inkscape** to create the figures in `Scalable Vector Graphics` format. This makes it easy to re-size the image as needed without losing quality. It also allows easy sharing of images among the teams. Export the image as a png once it is ready and include it in the document.

Listing 7. Different Ways of Adding an Image

```
image::images/rh-ra-banner.png[]
image::images/rh-ra-banner.png[RA-Banner]

[[img-rabanner]]
image::images/rh-ra-banner.png[caption="Figure 1: ", title="The RA Banner", alt="Banner",
width="400", height="300"]
```


Figure 1: The RA Banner

6. Admonition

There are certain statements that you may want to draw attention to by taking them out of the content's flow and labeling them with a priority. These are called admonitions. It's rendered style is determined by the assigned label (i.e., value). Asciidoctor provides five admonition style labels:

- NOTE
- TIP
- IMPORTANT
- CAUTION
- WARNING

When you want to call attention to a single paragraph, start the first line of the paragraph with the label you want to use. The label must be uppercase and followed by a colon (:).

Listing 8. Example of AsciiDoc Syntax

```
WARNING: Continued usage of Libreoffice is harmful to your health. Test Warning.
```

Continued usage of Libreoffice is harmful to your health. Test Warning.

A single line. Test tip.

A single line. Test important.

A single line. Test caution.

A single line. Test note.

Multi-line

Lorem Ipsum dolor sit amet, consectetur adipiscing elit. Nullam nulla magna, egestas ut porttitor a, facilisis ac risus. Curabitur faucibus pharetra libero et malesuada. Integer sed congue enim. Aliquam condimentum iaculis risus, id aliquam velit blandit sed. Donec venenatis eu velit vel congue. Sed dignissim, nunc nec consequat euismod, nibh risus vehicula sapien, in condimentum nisi sapien vitae turpis. Donec at orci in tellus pharetra malesuada. Suspendisse potenti.

Appendix A: Example Appendix

AsciiDoc article appendices are just just article sections with 'specialsection' titles.

Appendix B: Another Appendix

AsciiDoc article appendices are just just article sections with 'specialsection' titles.

B.1. Appendix Sub-section

Appendix sub-section at level 2.

7. Example Glossary

Glossaries are optional. Glossaries entries are an example of a style of AsciiDoc labeled lists.

A glossary term

The corresponding (indented) definition.

A second glossary term

The corresponding (indented) definition.