Delinea

Delinea Community Reference Guide

Delinea Professional Services

2022-09-09

Contents

Chapter 1

Introduction

This Community Guide, written in Bookdown, is intended for Delinea Professional Services and their customers to assist with unique tasks or "how-tos" that are not covered in official training. Articles are written by Delinea Professional Services consultants and architects from real-world experiences so it should serve as a great resource for everyone.

In addition to being an online resource, this format will also be exported into a .pdf file for offline viewing.

1.1   How to add an article

This section will describe the process to create the file and folder structure needed to contribute an article to the project.

1.1.1   File/Folder Structure

To contribute an article, create a folder named after your article. Then convert the name of the folder to lowercase and replace all spaces with a hyphen `-`. For example, if your article name is `How to install XYZ`, then the folder should be named `how-to-install-xyz`.

Inside that folder, create a new RMarkdown `.Rmd` file. This file should also follow the same naming convention mentioned above. Continuing with the same example, if your article name is `How to install XYZ`, then the RMarkdown file should be named `how-to-install-xyz.Rmd`.

This should create a file/folder structure that looks like the following:

`how-to-install-xyz\how-to-install-xyz.Rmd`

This is the bare minimum needed for a proper article contribution to this project. Any supporting graphics or other files should also go into this folder. Supporting files do not need to follow this special naming convention.

1.1.2   RMarkdown File Structure

The RMarkdown file needs to follow a simple format for proper inclusion into the project. There just needs to be a single, level 1 header `#` at the top of the document. You can use standard Markdown or RMarkdown syntax for whatever you wish to document. Continuing with the same example from above, the .Rmd file should look like the following:

```
1   # How to install XYZ
2
3   This article is to show how to install XYZ.
4
5   ## Step 1
6
7   This is a second level header for my article.
```

1.1.3 Including Graphics

While Markdown has a syntax for inserting graphics, it is recommended to use the following to ensure full compatibility with both web and pdf forms of this Bookdown project.

To insert fully compatible graphics, the following syntax should be used; continuing from our earlier example:

```
```{r echo=FALSE, fig.cap="Graphic Caption", out.width = '75%', fig.align='center',
↪  fig.pos = 'H'}
knitr::include_graphics("articles/how-to-install-xyz/Example Graphic.png")
```
```

1.1.4 Updating _bookdown.yml

Once all the files for your article are in the proper format, there needs to be a change to the base `_bookdown.yml` file in the root project folder. This file controls meta data about the Bookdown's project structure.

The part that needs to be updated is a line for the property `rmd_files`. This array controls the listing of the Rmd articles in order as it appears in this array. For example, let's take a look at a basic version of the property with just the index.Rmd file and another article.

```
11   rmd_files: [
12     "index.Rmd",
13     "articles/how-to-configure-abc/how-to-configure-abc.Rmd"
14   ]
```

To include your article in the Bookdown project, simply add a line to this file, separated by a comma and included in quotes. To continue from our earlier example, the following is adding the "How to Install XYZ" article into this property:

```
11   rmd_files: [
12     "index.Rmd",
13     "articles/how-to-configure-abc/how-to-configure-abc.Rmd",
14     "articles/how-to-install-xyz/how-to-install-xyz.Rmd"
15   ]
```

Once these changes are made to `_bookdown.yml`, submit these changes as a pull request to the Bookdown project. Once these changes are merged your article should now appear as part of the project.

Chapter 2

Bookdown Code Samples

This article describes some code examples to handling formating of special objects like images and tables as part of this Bookdown project:

2.1   Code Blocks

Code blocks should be displayed as the following:

```
~~~{.lineAnchors .numberLines startFrom="350"}
Code can go here
The .lineAnchors option makes code carriage return if the text is too long on one line
↪  for .pdfs.
The .numberLines option enables line numbers
The startFrom="350" options makes line numbers start with 350
~~~
```

The following is the above block as fully rendered:

```
350  Code can go here
351  The .lineAnchors option makes code carriage return if the text is too long on one line
     ↪  for .pdfs.
352  The .numberLines option enables line numbers
353  The startFrom="350" options makes line numbers start with 350
```

2.2   Images

Images needs to list the full path from the root project folder. Additional details such as positioning can be controlled via the R code block.

```
```{r echo=FALSE, fig.cap="Graphic Caption", out.width = '75%', fig.align='center',
↪  fig.pos = 'H'}
knitr::include_graphics("articles/how-to-install-xyz/Example Graphic.png")
```
```

Table 2.1: Version History

Version Number	Date	Author	Description
1	08/02/2022	YOURNAME	Initial version

2.3 Tables

While Markdown has simpler syntax for tables, the following should be used for table data to ensure full compatibility with both web and pdf versions of this Bookdown project:

```
```{r echo=FALSE, message=FALSE, warning=FALSE, results='asis'}
# header row is first
tabledata = "Version Number,Date,Author,Description
1.0,08/02/2022,YOURNAME,Initial version
"
# read the above text as a csv file
thistable <- read.csv(text = tabledata, header=TRUE, check.names=FALSE)
# turn it into a kable table
kbl(thistable, align = "c", caption = "Version History") %>% # name of table
  kable_styling(latex_options = c("striped", "responsive")) %>%
  column_spec(1, border_left = T) %>% # column 1 for border left line
  column_spec(4, border_right = T) %>% # last column for border right line
  row_spec(0, bold = T, background = "#241556", color = "white")
```
```

This method already includes Delinea's color branding.

This code block uses the `kable` package to design and build the table. Essentially, the tabledata shown above is treated as a String object, which is then read as a .csv. The `kbl` function then turns that csv into the table.

Check out (https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_html.html) for all kinds of options.

The following is the example code block from above:

Chapter 3

PowerShell Snippets

This section will contain various, helpful PowerShell snippets used by Professional Services.

3.1 Privilege-related

3.1.1 Test if prompt is administrative

```
([Security.Principal.WindowsPrincipal]
↪  [Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole(
↪  [Security.Principal.WindowsBuiltInRole] "Administrator")
```

Returns a `[System.Boolean]` `True` if prompt is an admin. `False` if it is not.

3.2 .NET Framework-related

3.2.1 Display current .NET Framework version

```
(Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\NET Framework Setup\NDP\v4\Full").Version
```

Returns the currently installed .NET Framework version as a `[System.String]`. For example `4.8.04084`.

3.3 Active Directory-related

3.3.1 Show current forest

```
([System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest()).Name
```

Returns the currently joined forest as a `[System.String]`. For example `domain.com`.

3.3.2 Set SPN for a user

Sets an SPN for a designated user:

```
Get-ADUser svc-username | Set-ADUser -ServicePrincipalNames
→   @{Add='MSSQLSvc/MSSQLSERVER01.domain.com:1433'}
```

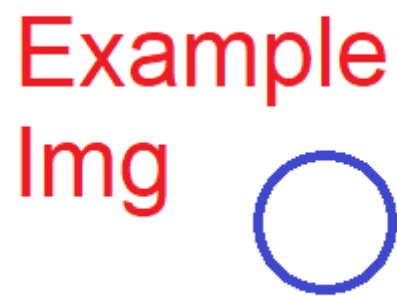## Chapter 4

How to install XYZ

This is how you install XYZ.

Figure 4.1: Face