
Swift Document Generation

Contents

| | | |
|----------|--|----------|
| 1 | Overview | 1 |
| 2 | Structure | 1 |
| 2.1 | Directories | 1 |
| 2.2 | AsciiDoc File | 1 |
| 2.3 | Chapters | 1 |
| 2.4 | Including Chapters in a Document | 1 |
| 3 | Building Documentation Manually | 2 |
| 3.1 | Prerequisites | 2 |
| 3.2 | build_docs.sh | 2 |
| 4 | Document Automation | 2 |
| 4.1 | Cron Job | 2 |
| 5 | Website | 3 |

1 Overview

This guide provides a brief overview of how Swift Document Generation works. It should explain the structure of a docs directory, how to manually build documentation, and explain how document automation works.

2 Structure

A specific file and directory structure is required for each document in order to ease the process of automation, conversion, and copying.

2.1 Directories

Each subdirectory within docs/ should correspond to the name of the document. This name will also later be used in the URL used to point to the document.

Example Directory Names

```
docs/userguide
docs/tutorial
```

2.2 AsciiDoc File

Every document directory should contain one file with a .txt extension. This .txt file is the AsciiDoc document that later gets converted to various formats.

Example AsciiDoc Documents

```
docs/userguide/userguide.txt
docs/tutorial/tutorial.txt
```

2.3 Chapters

Each document directory should contain individual chapters that make up the document. These chapters should not have any file extensions.

Example Chapters

```
docs/userguide/overview
docs/userguide/language
docs/userguide/mappers
docs/userguide/commands
```

2.4 Including Chapters in a Document

The .txt file should then include the individual chapters with the AsciiDoc include command.

userguide.txt

```
include::overview[]

include::language[]

include::mappers[]

include::commands[]
```

**Warning**

Be sure to include spaces between includes. Putting them together with no spaces can cause formatting/compilation errors.

3 Building Documentation Manually

Before committing anything to SVN, it is a good idea manually build the documentation for testing purposes.

3.1 Prerequisites

There are several prerequisites for building documentation:

- asciidoc and a2x (<http://www.methods.co.nz/asciidoc>)
- dblatex (<http://dblatex.sourceforge.net>)
- source-highlight (<http://www.gnu.org/software/src-highlight>)
- For Mac OSX users only: coreutils (installation via macports recommended)

For CI users, compiled versions of these packages can be found in ~davidk.

3.2 build_docs.sh

The script to build the documentation is docs/build_docs.sh. This script will convert the documents into HTML and PDF, and then copy the files to a given output directory.

Usage

```
./build_docs.sh /path/to/copy/output
```

4 Document Automation

All .txt files in the docs directory get converted to HTML and PDF on a nightly basis. These documents then get copied to the Swift web site.

4.1 Cron Job

The job that processes these documents runs on Bridled at 7pm nightly. The script is /home/davidk/docscripts/update_and_build.sh. The script calls svn update on several different swift directories in ~davidk, and then builds the the documents. The documents should be built for each Swift branch later than 0.93, including trunk. As new versions of Swift are created, this script will need to be updated.

5 Website

Once the cron job runs, documents can be accessed at:

`http://www.ci.uchicago.edu/swift/guides/release-<version>/documentname/documentname.html` `http://www.ci.uchicago.edu/swift/guides/<version>/documentname/documentname.pdf`

For example, assume that you create a document called docs/newdoc. You check this guide into Swift 0.93. The AsciiDoc .txt file is called newdoc.txt. The URLs that would be generated would be:

`http://www.ci.uchicago.edu/swift/guides/release-0.93/newdoc/newdoc.html` `http://www.ci.uchicago.edu/swift/guides/release-0.93/newd`

[home](#)
