Paper 232-2008

DocItOut: Publishing SAS® Code Documentation on the Web

Choon-Chern Lim, Mayo Clinic, Rochester, MN

ABSTRACT

Documenting your SAS code is almost as important as writing the code. As software developers, we are gifted with great imaginations to solve complex problems. Since there are usually several possible ways to achieve the same end results, it is crucial to document our thoughts clearly in plain language to assist other developers (and ourselves) to better understand the implementations in the future. However, the biggest challenge is that we take a great deal of time to document our code, yet nobody seems to be able to find them. Most of us find it annoying to traverse directory after directory just to search for the documentation that is embedded deep in the code.

This paper introduces DocItOut, a free open source tool, which is specifically developed to traverse these messy directories and generate properly formatted web documentation in HTML format based on the comments and statements in the SAS code. The latest release of DocItOut can be downloaded from http://docitout.sourceforge.net/.

INTRODUCTION

"Code Documentation" – This is one of those subjects we software developers preach about, yet most of us don't practice it. If we conduct a poll among the audience in a room on whether they frequently document their code, almost everyone would unanimously nod their heads in agreement, then, feel a little guilty about it.

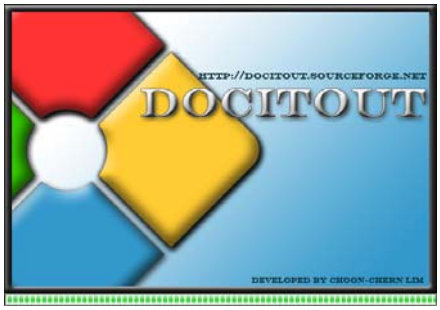
In most organizations, it is inevitable for one development team to inherit and maintain projects developed by other teams. It is always a nightmare to take over one of those projects that has minimal or absolutely no code documentation. We have all heard or made these excuses before: *"The coding implementation is so straightforward and almost pseudocode-like. Thus, there is no point documenting it"*, or, *"It is counter productive to document the code because the implementation changes frequently"*. In the software development world, there are many different solutions to satisfy the customer requirements. Depending on the software developer's skill level, some solutions could be badly implemented, yet they work. Some are nicely designed, albeit difficult to understand due to high level of code abstraction. Regardless of how these solutions are implemented, what good is the implementation when no one else understands it and knows how to maintain it?

On the other hand, there are also a few of these "rare breed" software developers that actually put generous effort in documenting their code… kudos to them. These individuals spend tremendous amount of time keeping the documentation up to date, yet nobody seems to be able to find them. This documentation gets cluttered with the actual code implementation in a file, and these files are usually hidden in a massive directory hive. This makes it difficult for others to search for code documentation because they have to either traverse directory after directory or scroll down thousand lines of code to look for them. In the end, what good is the code documentation when no one else could ever find it?

This paper introduces a handy utility tool known as DocItOut that enables you to quickly generate web documentation based on the comments and statements in the SAS code. The goal of this paper is twofold: to emphasize the importance of proper code documentation and to enable software developers to easily share the code documentation on the web.

Please note that the DocItOut features mentioned in this paper are based on version 1.2.0. Some of this information may slightly differ by the time this paper is actually presented.

WHAT IS DOCITOUT



DocItOut is a freely available open source tool that enables SAS developers to easily publish and share project code documentation through a more accessible medium - web. DocItOut provides an easy point-and-click graphical user interface (GUI). At this moment, there is currently no other available tool besides DocItOut that offers this functionality. By specifying the project code root directory, DocItOut recursively traverses each subdirectory to look for files with ".sas" suffix and parses the "preferred" comment styles (will be discussed in *"Enhancing Web Documentation Content"* section) and selected SAS statements from the files. Based on this information, DocItOut generates well formatted web documentation in a specified destination directory. DocItOut does not and will never alter the SAS files.

If you are one of those "brave souls" who rarely document your code, do not worry. DocItOut will still generate the web documentation for you, albeit with less information. However, it is highly encouraged to instill the habit of documenting your own code because you know your own implementations the best.

In effort to standardize code documentation between different languages, the DocItOut's "preferred" comment styles closely resemble Javadoc's version (for JAVA language). These comment styles are 100% compatible with SAS and you can safely execute your SAS code without getting any runtime errors. The DocItOut's generated web documentation look and feel is also similar to Javadoc's version.

GETTING STARTED

First, you need to visit http://docitout.sourceforge.net and click on "Download" link on the right menu navigation. This will bring you to a page that has a green download box (see Figure 1).

Figure 1: Download box

After clicking on the download box, please download *DocItOut-installer-**x.x.x**.exe* where ***x.x.x*** represents the latest release. Then, double click this executable file and proceed with the installation. By using the default installation settings, the installer will create a desktop icon that enables you to launch DocItOut.

DocItOut currently runs only in **Windows environment**. The main reason is because I have limited environments to test the DocItOut graphical user interface (GUI) for compatibility. However, depending on the user demands, I may consider implementing a command line interface (CLI) version in the future that will be cross-platform compatible. In the mean time, if your SAS code resides in UNIX environment, you can map a network drive (through Samba) to make your SAS code accessible in Windows environment before running DocItOut.

DocItOut requires **at least Java Runtime Environment (JRE) 6.x** to run. Most computers have JRE 1.4 by default. If you encounter difficulties in installing or launching DocItOut, please download and install the latest Java Runtime Environment (JRE) from http://java.sun.com/javase/downloads/index.jsp . Finally, restart your computer and proceed with DocItOut installation again.

LAUNCHING DOCITOUT

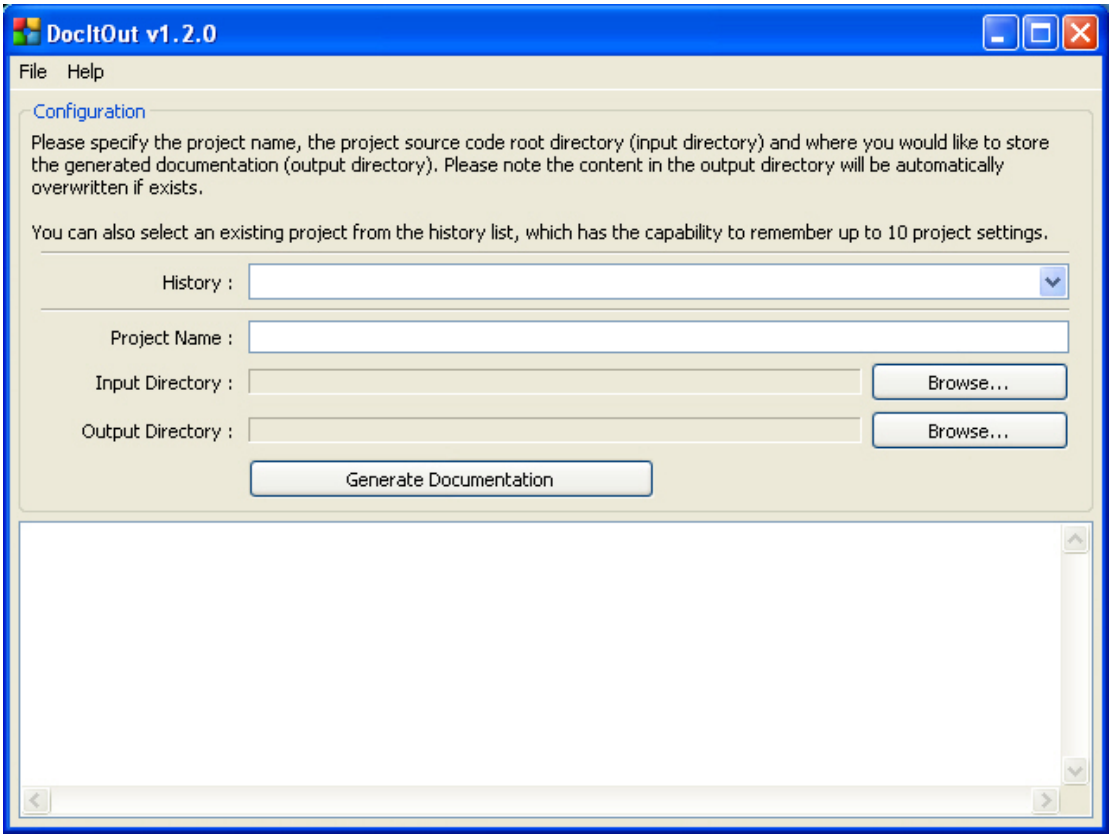


Figure 2: DocItOut graphical user interface (GUI)

Figure 2 shows a successfully launched DocItOut. The field descriptions are described in Table 1.

Table 1: DocItOut field descriptions

Field Name	Description
History	This field remembers previously configured project documentation settings. This prevents you from constantly retyping the project name, input directory and output directory each time you plan to regenerate the documentation.
Project Name	The project name to be printed on the main documentation page.
Input Directory	The directory where you store your SAS files. If the files are grouped into several subdirectories, please specify the root directory here. DocItOut automatically traverses the subdirectories (if there is any) to look for SAS files.
Output Directory	The directory where you want the generated web documentation to be stored at.

When the “Generate Documentation” button is clicked, DocItOut will first validate the project name, input and output directories before proceeding. The processing activities are logged into the log console. This enables you to check the log messages for any errors.

GENERATING WEB DOCUMENTATION

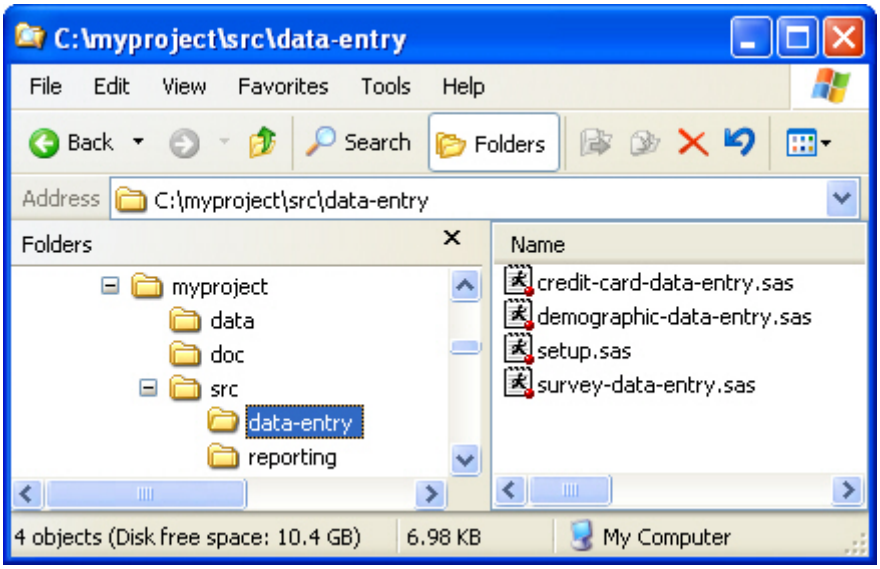


Figure 3: Sample project structure to demonstrate how DocItOut works

In order to show how DocItOut works, we will use the above project structure (Figure 3). The **data** directory stores all SAS data sets. The **doc** directory stores the generated web documentation. The **src** directory contains all SAS files. In this example, this directory has two subdirectories: one subdirectory contains data entry related program files, and another contains reporting related program files.

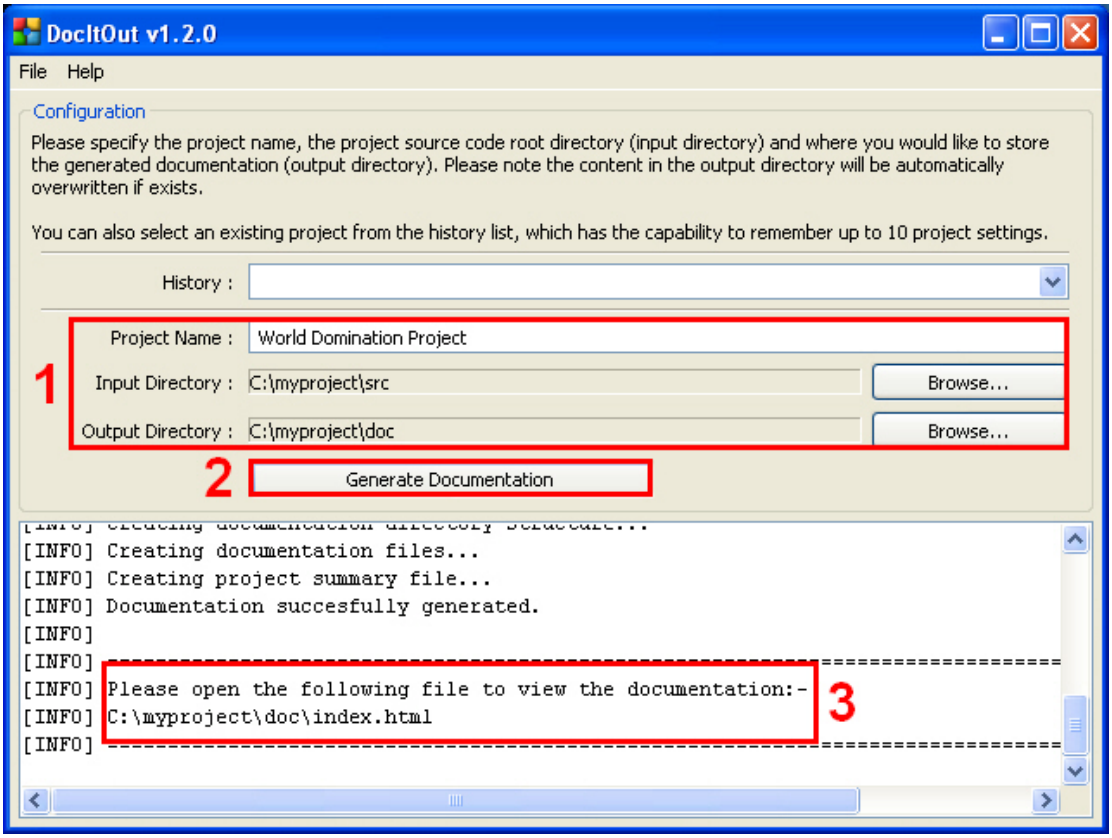


Figure 4: Generating web documentation with DocItOut

In DocItOut (Figure 4), we specify the pertinent information first before clicking the “Generate Documentation” button. Once successfully generated, we open the file path in the web browser to view the generated web documentation.

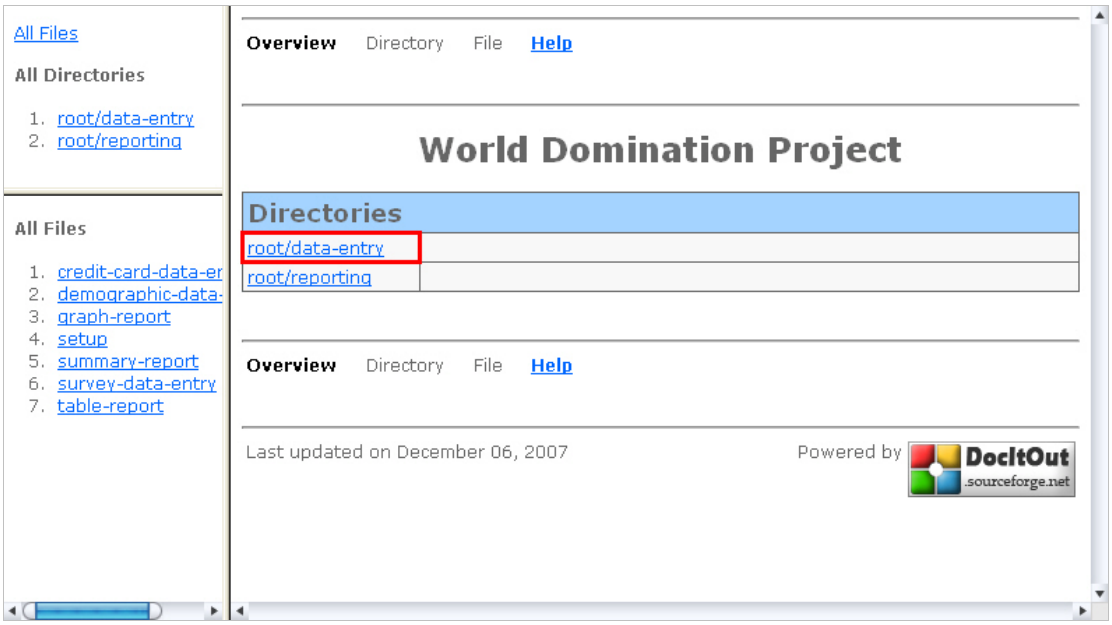


Figure 5: Overview page

Figure 5 shows the **Overview** page. This page provides a general summary of the directories that contain the SAS files. By clicking the directory link (highlighted in red), the **Directory** page will be displayed.

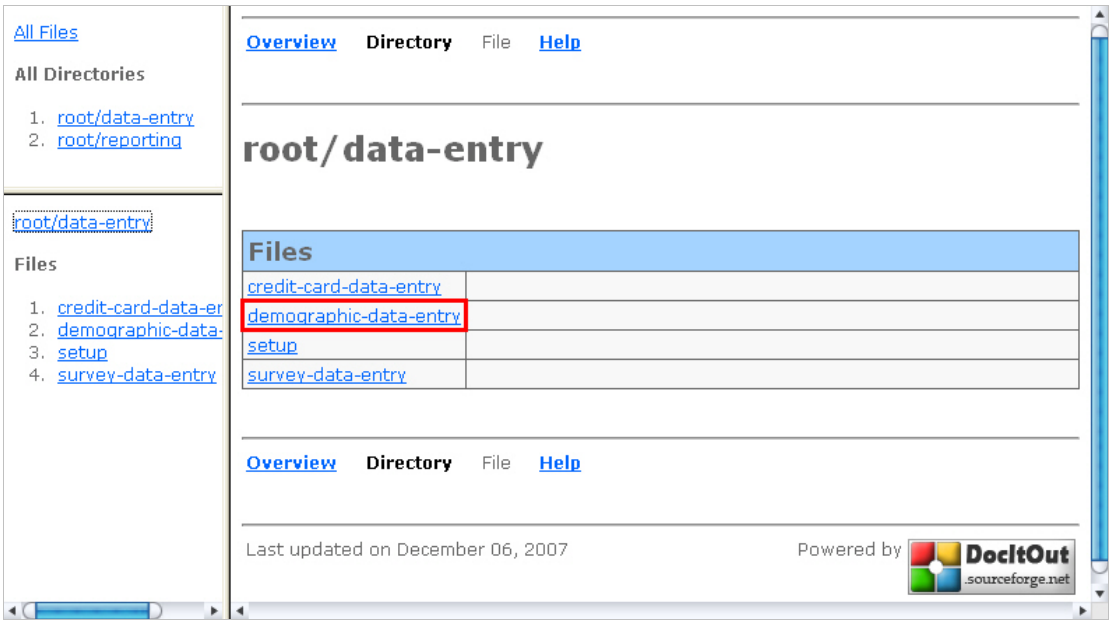


Figure 6: Directory page

Figure 6 shows the **Directory** page. This page provides a general summary of all SAS files within the selected directory. By clicking the file link (highlighted in red), the **File** page will be displayed.

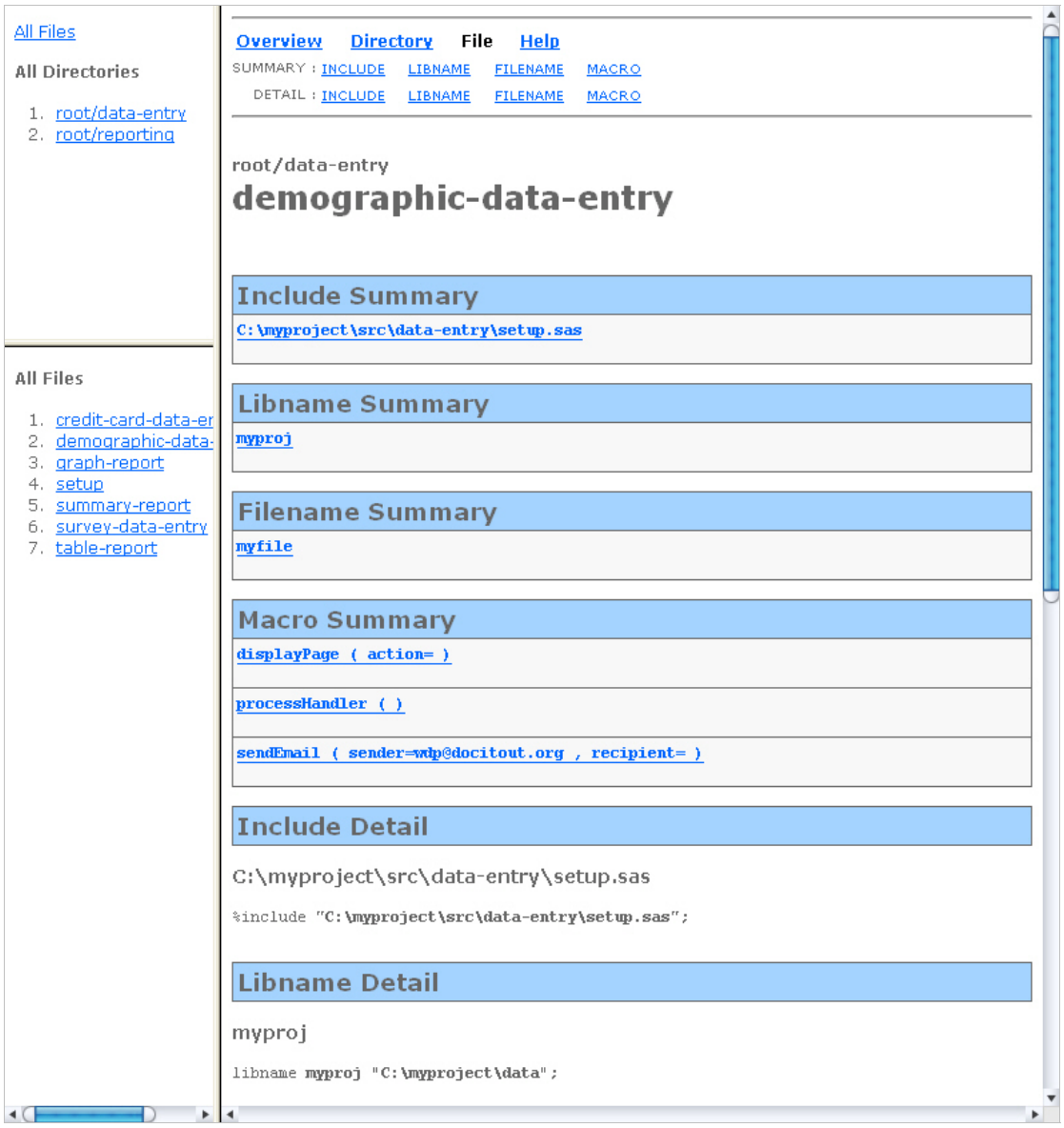


Figure 7: File page

Figure 7 shows the **File** page. This page provides both summary and detail sections. Since this file does not contain any code documentation, this page only displays information from the SAS statements.

ENHANCING WEB DOCUMENTATION CONTENT

As you can see now, the generated web documentation is good, yet it is not extremely useful. The reason is because these files are not documented at all and DocItOut only creates this documentation based on the directory structure and SAS statements from the files.

In previous Figure 5, while the directories are listed in the **Overview** page, there are no descriptions elaborating about them. To include the directory description, we need to create and place a **readme.txt** in each of these directories (Figure 8).

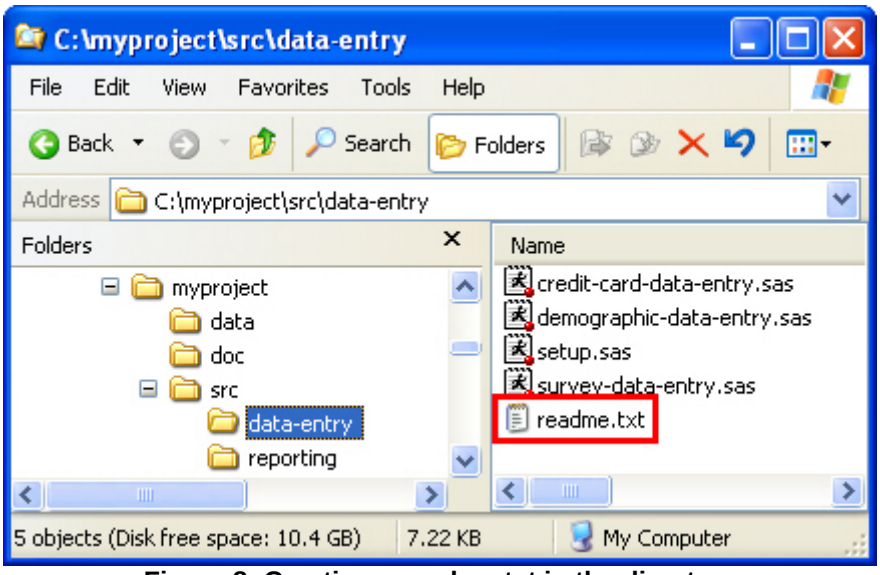


Figure 8: Creating a readme.txt in the directory

In this **readme.txt** file, you can either type the description in plain text or use HTML tags to decorate the texts. However, it is recommended to keep the HTML usage to the minimum for readability and consistency purpose. For example, the content in the **readme.txt** file may look like this:-

Table 2: Sample content in readme.txt

```
Contains all data entry related program files. There are currently three
data entries implemented: <code>credit card</code>, <code>demographic</code>
and <code>survey</code>.
<p>
These program files are currently maintained by <b>team WDP</b>.
```

Figure 9: Overview page with directory description

Figure 9 shows the **Overview** page that contains directory description. DocItOut displays the first sentence from **readme.txt** file in this page. The full directory description is placed in the **Directory** page.

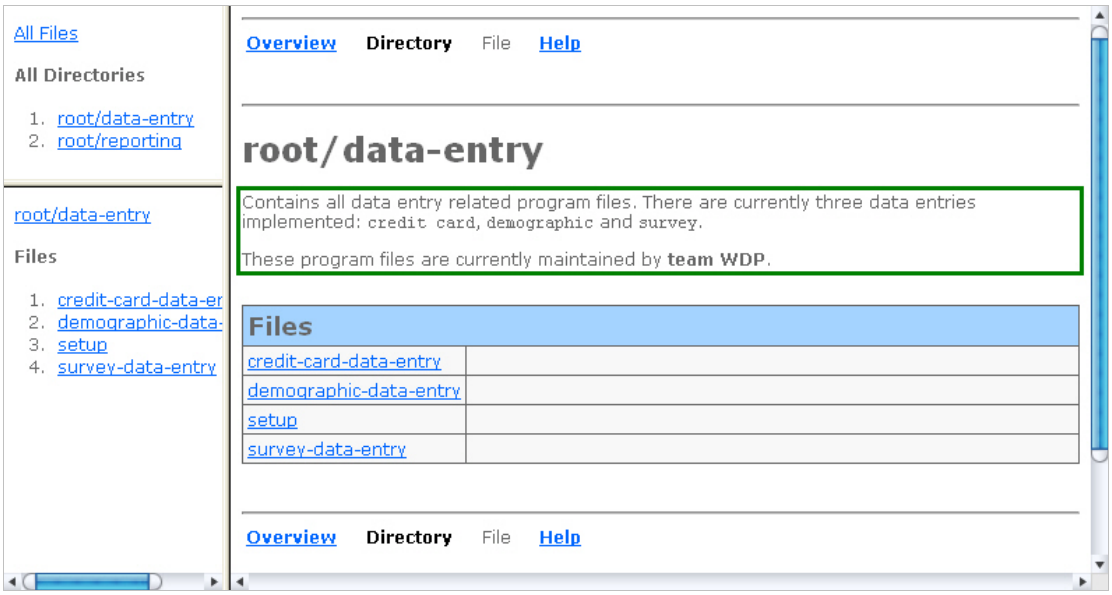


Figure 10: Full directory description in Directory page

Figure 10 shows the **Directory** page that contains full directory description. Since we have used HTML tags in **readme.txt** file, the texts are also properly rendered here. However, similar to the **Overview** page, while the files are listed in the **Directory** page, there are no descriptions elaborating about them. To include the file description, we need to add a file header comment block in each SAS file.

Table 3: File header comment block

Syntax

```
/*!
 *     <description>
 *
 *     @author       <author-name>
 *     @created      <creation-date>
 */
```

Notes

1. Every SAS file can have only one file header.
2. This comment block must begin with `/*!` and ends with `*/`. It must be placed at the top of the file.
3. The file header captures three types of information:-
    a. A description of the particular SAS file.
    b. Author(s).
    c. Creation date.
4. Two available tags: `@author` and `@created`.
5. The order of the tags is important or they will get ignored.

Example

```
/*!
 *     This file collects demographic information from end users.
 *     Whenever the user deletes a record from the demographic database,
 *     this program will notify the administrator by email.
 *
 *     @author       Choon-Chern Lim (Mike)
 *     @author       Popeye the Intern
 *     @created      01/01/2007
 */
```

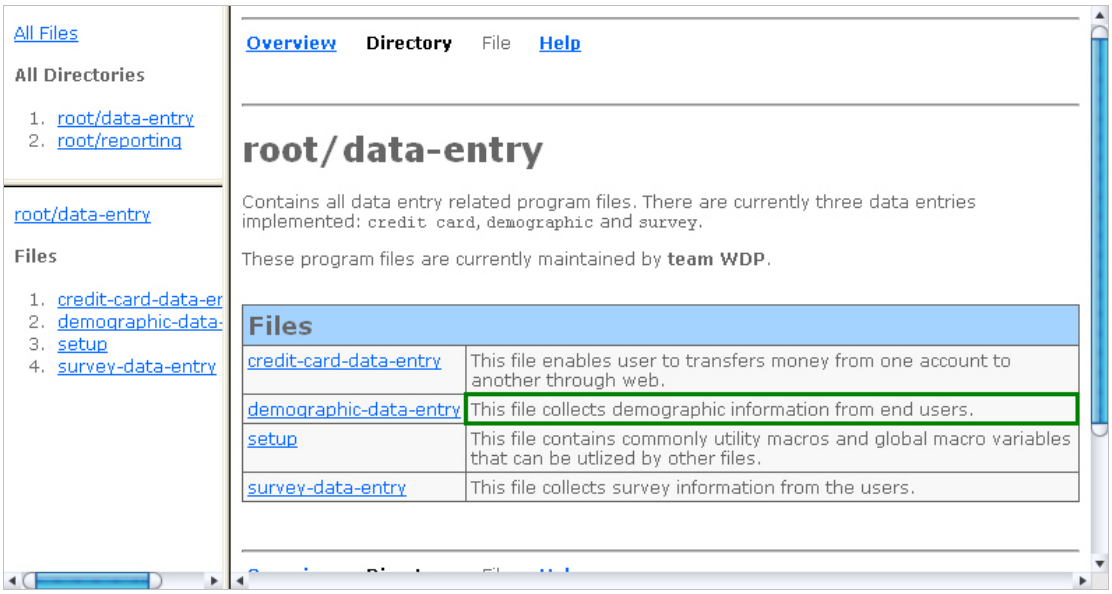


Figure 11: Directory page with file description

Figure 11 shows the **Directory** page that contains file descriptions. DocItOut only displays the first sentence from the file header. The full file description is placed in the **File** page. The below are the rest of the comment blocks that can be placed in the SAS file.

Table 4: Include comment block

Syntax

```
/**
 *     <description>
 */
%include "<file-path>";
```

Notes

1. This comment block does not have any tag.
2. It begins with `/**` and ends with `*/`. This comment block must be placed right above `%include` statement.

Example

```
/**
 *     Utility macros for generating the emails. Why? Because this
 *     is the coolest!
 */
%include "c:\project\pgm\utility.sas";
```

Table 5: Libname comment block

Syntax

```
/**
 *     <description>
 */
libname <library-name-ref> "<library-path>";
```

Notes

1. This comment block does not have any tag.
2. It begins with `/**` and ends with `*/`. This comment block must be placed right above `libname` statement.

Example

```
/**
 *     Data storage for Project MI350.
 */
libname mylib "c:\project\data";
```

Table 6: Filename comment block

Syntax

```
/**
 *     <description>
 */
filename <file-name-ref> "<file-path>";
```

Notes

1. This comment block does not have any tag.
2. It begins with `/**` and ends with `*/`. This comment block must be placed right above `filename` statement.

Example

```
/**
 *     File storage for Project MI350, which is physically stored in
 *     Windows 2003 environment.
 */
filename myfile "c:\project\file";
```

Table 7: Macro comment block

Syntax

```
/**
 *     <description>
 *
 *     @param        <param-name> <param-description>
 *     @return       <return-description>
 */
%macro sendEmail(<param-name>=<default-value>);
       <some-implementation>
%mend;
```

Notes

1. This comment block has two tags: `@param` and `@return`.
2. It begins with `/**` and ends with `*/`. This comment block must be placed right above `macro` statement.
3. The order of the tags is important or they will get ignored.
4. The `<param-name>` in the comment block must match the `<param-name>` in the `macro` parameter. Otherwise, it will be ignored.

Examples

```
/**
 *     Sends email to the intended recipient.
 */
%macro sendEmail;
       <some-implementation>
%mend;
```

```
/**
 *     Sends email to the intended recipient.
 *
 *     @param        sender        Sender's email address.
 */
%macro sendEmail(sender=wdp@docitout.org);
       <some-implementation>
%mend;
```

```
/**
 *     Sends email to the intended recipient.
 *
 *     @param        sender        Sender's email address.
 *     @param        recipient     Recipient email address.
 *     @return       Status "1" if successful, "0" otherwise.
 */
%macro sendEmail(sender= wdp@docitout.org, recipient=);
       <some-implementation>
%mend;
```

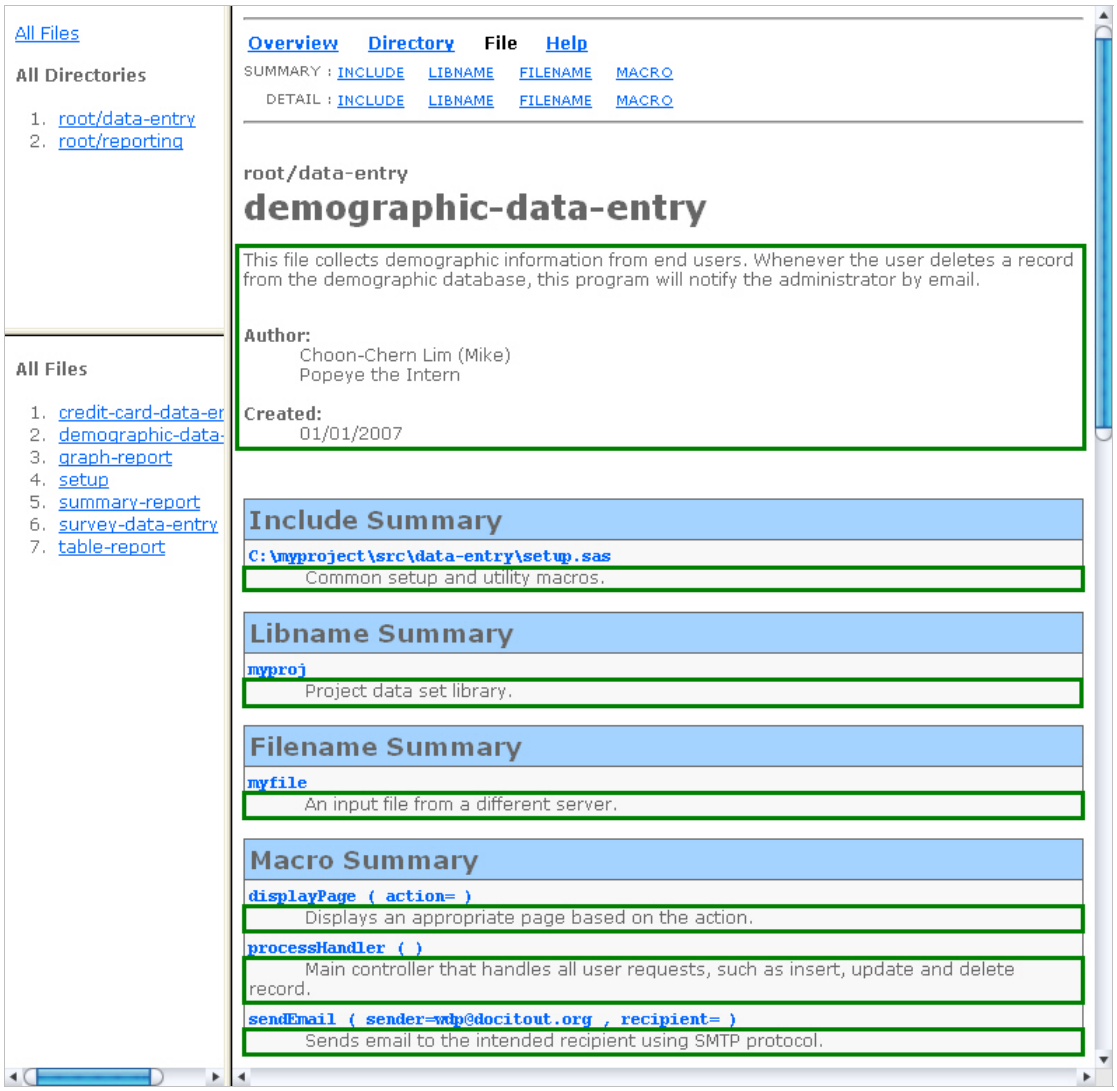


Figure 12: File page with full file description and summary sections

Figure 12 and Figure 13 show the **File** page that contains a fully documented SAS file. Each file contains general file information as well as summary and detail sections. There are also quick links at the top of the page that enables you to jump directly into a certain section.

Each summary section displays only the first sentence of the description from the comment block. The main purpose of having the summary sections is to allow you to quickly browse through the whole file content. If you are interested in a particular item within a section, you can click on the link to jump directly into the detail section.

The detail sections contain all the information from the comment blocks.

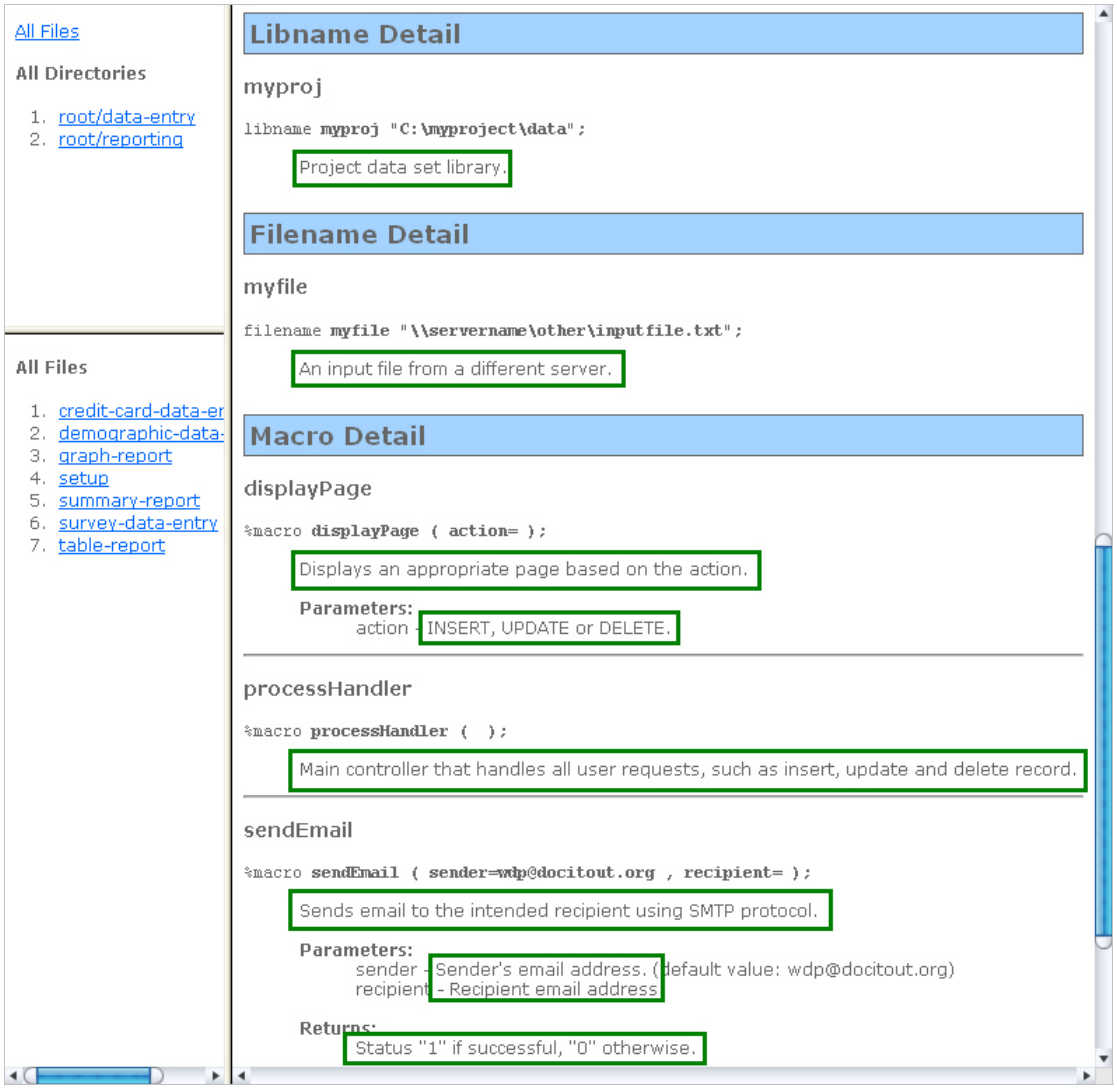


Figure 13: File page with detail sections

CONCLUSION

This paper introduces DocItOut, a freely available open source tool, which SAS developers can use to generate web documentation based on the comments and statements from the SAS files. This paper emphasizes the importance of code documentation and the ability to easily share them on the web. Instead of burying useful project code documentation in a place where nobody could find them, it is much more efficient and easier to make this documentation accessible on the web. The latest release of DocItOut can be downloaded from http://docitout.sourceforge.net/.

REFERENCES

Choon-Chern Lim. 2007. "DocItOut (SAS Documentation Generator)". http://docitout.sourceforge.net/

Sun Microsystems, Inc. 2007. "Javadoc Tool". http://java.sun.com/j2se/javadoc/

ACKNOWLEDGEMENTS

Special thanks to Dawn Finnie for taking the precious time to review this paper. Bow before her for she is the greatest! ☺

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Choon-Chern Lim
Mayo Clinic
200 First Street SW
Rochester, MN 55905
Email: limc@mayo.edu