THE POWER TO KNOW.

SAS® Document Conversion 1.2

Developer's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2010. *SAS® Document Conversion 1.2: Developer's Guide*. Cary, NC: SAS Institute Inc.

SAS® Document Conversion 1.2: Developer's Guide

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, January 2010

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/publishing** or call 1-800-727-3228.

Contents

About This Book

Audience

SAS Document Conversion is designed for the C developer who installs SAS Document Conversion onto a server and who writes the code to convert an input document.

Prerequisites

Here are the prerequisites for using SAS Document Conversion:

- third-party programs for the documents that you want to convert
- SAS Document Conversion loaded onto your server
- access to documents that you want to convert into text

Conventions

This manual uses the following typographical conventions:

Convention	Description
`TGM_ROOT`	The root directory where SAS Document Conversion is installed, typically the following: **Windows:** `C:/Program Files/SAS/SAS Document Conversion`
`.sam`	The code examples for the C program are shown in a fixed-width font.
`Top`	The names of taxonomy nodes appear in a fixed-width font.
`www.sas.com`	The hypertext links are shown in a light blue, fixed-width font, and are underlined.

1 About SAS Document Conversion

- *What is SAS Document Conversion?*
- *Benefits to Using SAS Document Conversion*
- *How Does SAS Document Conversion Work?*

1.1 What is SAS Document Conversion?

Most organizations require access to information that is created by using a number of different software applications. This requirement poses a challenge because data is often stored in various types of formats. For example, you might use Microsoft Word to write proposals and Microsoft Excel to analyze financial data. Preprocessing is required in order to obtain the information in these documents.

Use SAS Document Conversion when you want to obtain metadata on your data using SAS Content Categorization Studio. You can also choose to use SAS Document Conversion to preprocess your files into `.txt` documents before you build an index.

SAS Document Conversion provides a client - server solution with a server that runs on Windows and a C API to develop your client. This configuration provides optimal conversion capabilities enabling you to address your organization’s textual information requirements.

SAS Document Conversion automatically leverages third-party software installed on your server, for example, Microsoft Office. When you load more third-party programs onto your server, you extend the conversion capabilities of SAS Document Conversion.

Easy file conversion

load the application onto your server and use the custom application that you write to automatically convert your files.

Custom application

use the program code provided to write your custom application.

Convert many types of files

easily convert the types of files created by the third-party software loaded on your server.

Use the output files with many applications

choose to use the output files with a wide variety of applications including SAS Content Categorization Studio and SAS Search and Indexing.

Expansive conversion capabilities

install several programs on your server to expand these document conversion capabilities.

1.2 Benefits to Using SAS Document Conversion

SAS Document Conversion provides the following benefits:

Empowers business owners by converting many forms of data into text

SAS Document Conversion provides the text conversion functionality that is required to convert your data into text.

Improves the business value of IT and the corporate data that it manages

SAS Document Conversion provides you with a simple, easy-to-use data conversion program that is flexible and customizable.

Saves money on training and support costs

SAS Document Conversion is so simple that you can quickly become self-sufficient, with minimal IT support and no need for extensive training. Once you start using SAS Document Conversion you are no longer dependent on the IT staff.

1.3 How Does SAS Document Conversion Work?

SAS Document Conversion enables you to load the program to the server and then to write a customized client that sends the documents to the server for preprocessing. Use the C language API included in this manual to develop your client.

2 Installation

- *Overview*
- *Prerequisite System Requirements*
- *Install SAS Document Conversion*

2.1 Overview

SAS Document Conversion comes with an installer and works with Microsoft Windows running on your server. SAS Document Conversion (using Microsoft IFilter technology) extracts text from your input documents, for example, Word documents.

SAS Document Conversion enables you to use your text files with SAS applications and third-party solutions. For example, you can use SAS Document Conversion to convert Word files into text before extracting their entities using SAS Content Categorization Studio.

When you install SAS Document Conversion on your server, where third-party programs are also running, you enable the converter to expand its capabilities. This means that plain text can be extracted from documents where it could not otherwise be obtained. For example, if you are running Microsoft Office, you could convert the text in PowerPoint presentations into text format.

2.2 Prerequisite System Requirements

The installation prerequisites include optional third-party software, for example, Microsoft Office. However, you can install third-party software after you install SAS Document Conversion. In this case, SAS Document Conversion also works with the newly-installed program.

Configure the server where you install SAS Document Conversion according to the recommended system configuration:

CPU: x86 with 1 GHz or higher required. 2+ CPUs of 2 GHz or higher, each, are recommended.

RAM: 1 GB or higher is recommended, but this base number depends on the size of the project that you have loaded.

Use Table 2-1 below to learn about the supporting operating systems and the platforms required to run SAS Document Conversion:

Table 2-1: Supported Operating Systems

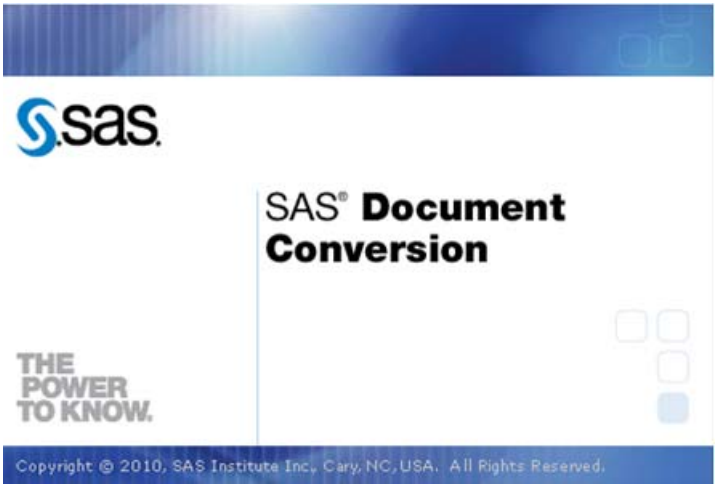
Operating System	Platform
Linux, (Red Hat 7.x, 8, 9, Fedora 1-3, RHEL 2.1 and higher), Suse	x86, x86-64
IBM AIX	PPC
FreeBSD	x86
HP-UX 32	PA-RISC
Sun Solaris (32-bit)	x86
Sun Solaris (32-bit)	SPARC
Sun Solaris (64-bit)	UltraSPARC
Tru64 UNIX	HP Alpha
Windows	x86, x86-64

Run the SAS Document Conversion application on a Windows machine running .NET, or on a UNIX machine running Python.

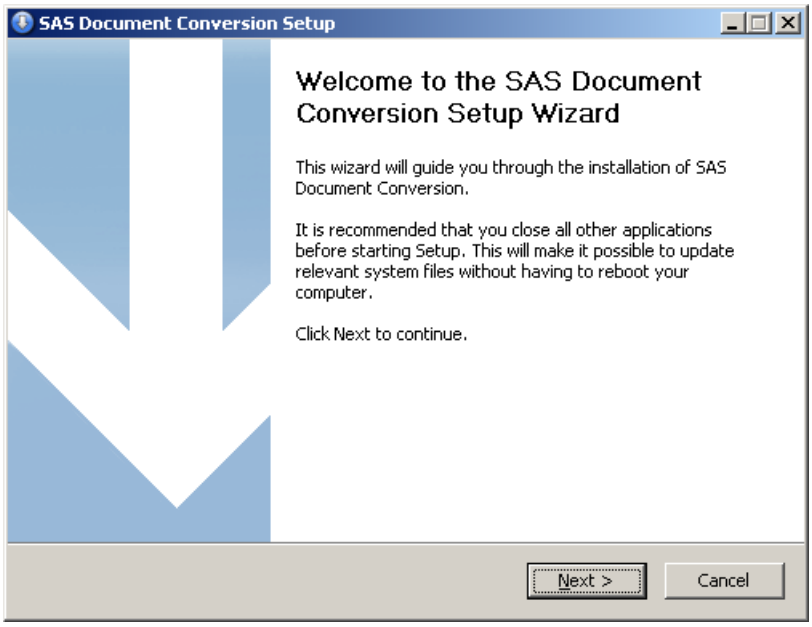
2.3 Install SAS Document Conversion

To install SAS Document Conversion on your server, complete these steps:

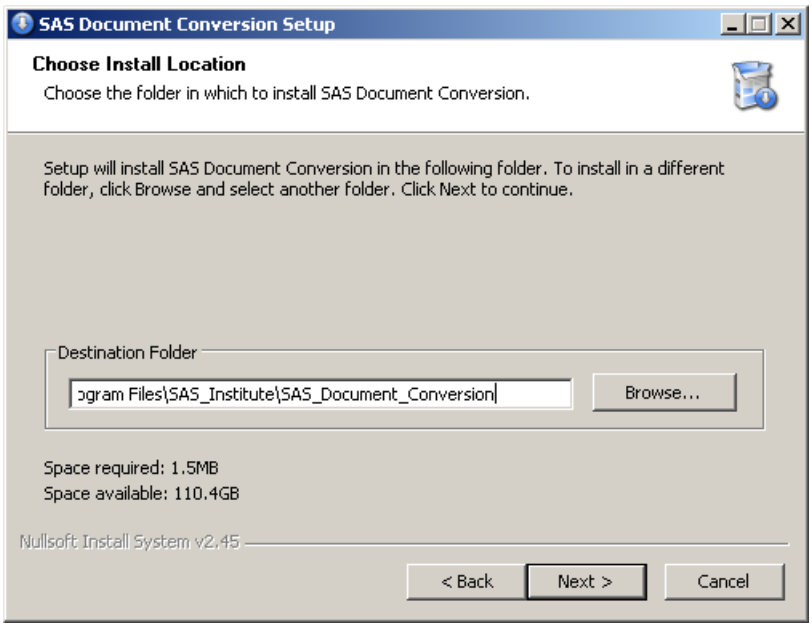
1. Double-click on the `SAS_DocConversion_Setup.exe` and the splash page appears.



2. Click **Next** in the Welcome page that appears.

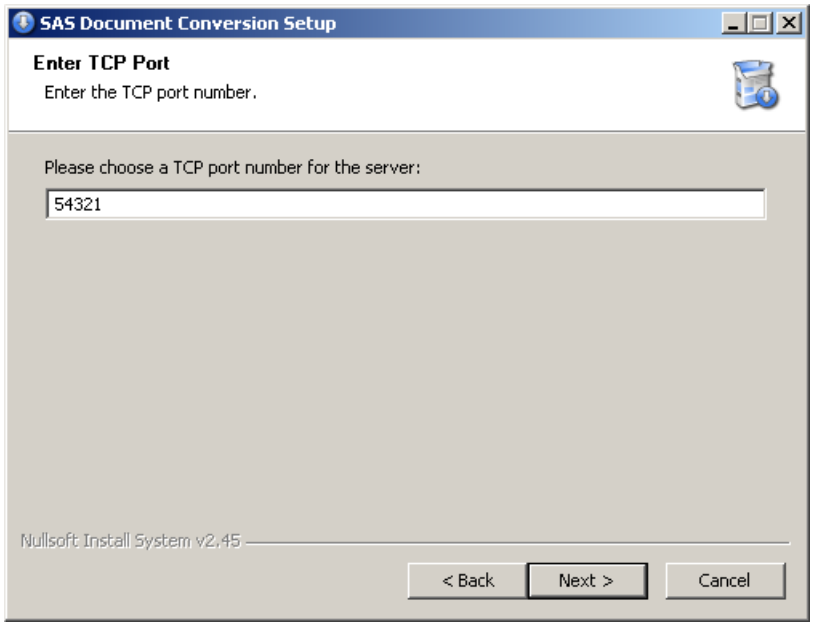


The Choose Install Location page appears.



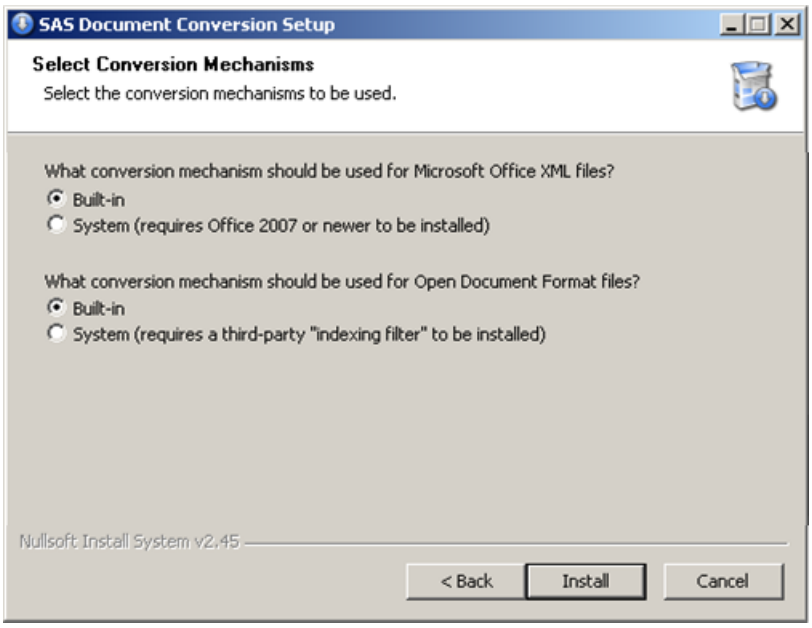
3. (Optional) Click **Browse** to open the Browse For Folder dialog box where you can select a different install location.
4. Compare the **Space required** with the **Space available** numbers to see if there is enough room on your hard drive to install this program.

5. Click **Next** and the Enter TCP Port page appears.



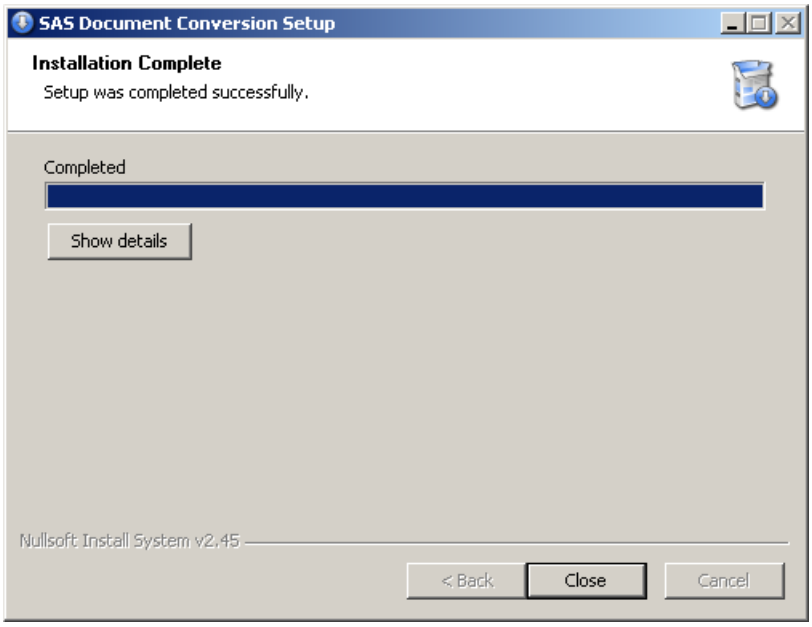
   a. (Optional) Type a new TCP port number in the **Please enter a TCP port number for the server** field.

b. Click **Next** and the Select Conversion Mechanisms page appears.



Note: You should only select a **System** operation if you are sure that the specified software is installed, otherwise unexpected behaviors might occur.

6. Click **Install** and the Installation Complete page appears.



7. (Optional) Click **Show details** to view a list of the extracted files.
8. Click **Close**.

3
C API and Sample Program

- *Overview*
- *The C API*
- *Client Class*
- *Input Class*
- *Output Class*
- *Sample Program*

3.1 Overview

This chapter contains the client, input, and output classes and their methods for the C API. A sample program is also included in this chapter for reference purposes. Use the C API methods to build the SAS Document Conversion application that converts files of various formats into text.

3.2 The C API

The C API is comprised of the following methods:

Table 3-1:

Method	Description
Client Class	
init_file_converter_client	Initializes a FILE_CONVERTER_CLIENT object.
Input Class	
init_file_converter_input	Initializes an input document.
free_file_converter_input	Frees a FILE_CONVERTER_CLIENT_INPUT object.
clear_file_converter_input	Resets a FILE_CONVERTER_CLIENT_INPUT object to empty, for reuse purposes.
set_content_file_converter_input	Sets the content to be converted.
set_file_name_file_converter_input	Sets the file name.
set_mime_type_file_converter_input	Sets the mime type.
Output Class	
init_file_converter_output	Initializes a File Converter output object.
free_file_converter_output	Frees a FILE_CONVERTER_OUTPUT object.
clear_file_converter_output	Resets a FILE_CONVERTER_CLIENT_OUTPUT object to empty for reuse purposes.
get_title_file_converter_output	Gets the title of the File Converter output object.
get_author_file_converter_output	Gets the author of the File Converter output object.
get_text_file_converter_output	Gets the text of the File Converter output object.

3.3 Client Class

This method is used to initialize the client object.

init_file_converter_client

Initializes a `FILE_CONVERTER_CLIENT` object.

C Synopsis

```
FILE_CONVERTER_CLIENT init_file_converter_client
                                (char *hostname, int port);
```

Description

This C method initializes a `FILE_CONVERTER_CLIENT` that communicates with the server and its hosts on the connections that are made to the specified port.

Arguments

Arguments	Description
`hostname`	Specifies the client for the File Converter server.
`port`	The number of the port for the server.

Return Values

- `FILE_CONVERTER_CLIENT` object: success
- `NULL`: failure

3.4 Input Class

These methods are used for input documents.

init_file_converter_input

Initializes an input document.

C Synopsis

```
FILE_CONVERTER_INPUT init_file_converter_input(void);
```

Description

This C method initializes an input object that is used to specify the input parameters for document conversion.

free_file_converter_input

Frees a `FILE_CONVERTER_CLIENT_INPUT` object.

C Synopsis

```
void free_file_converter_input (FILE_CONVERTER_INPUT
                                input);
```

Description

This C method frees a `FILE_CONVERTER_CLIENT_INPUT` object.

clear_file_converter_input

Resets a FILE_CONVERTER_CLIENT_INPUT object to empty for reuse purposes.

C Synopsis

```
void clear_file_converter_input
                (FILE_CONVERTER_INPUT input);
```

Description

This C method resets a FILE_CONVERTER_CLIENT_INPUT object to empty for reuse purposes.

Argument

Argument	Description
input	Specifies the FILE_CONVERTER_CLIENT_INPUT object.

set_content_file_converter_input

Sets the content to be converted.

C Synopsis

```
void set_content_file_converter_input
                  (FILE_CONVERTER_INPUT input,
                   unsigned char *content,
                    int32_t length);
```

Description

This C method sets the content to be converted for SAS Document Conversion.

Arguments

Argument	Description
`input`	Specifies the File Converter input.
`content`	The opaque byte array.
`length`	The length of the byte array.

set_file_name_file_converter_input

Sets the file name.

C Synopsis

```
void set_file_name_file_converter_input
                (FILE_CONVERTER_INPUT input,
                 unsigned char* name);
```

Description

This C method sets the file name.

Argument

Argument	Description
input	Specifies the File Converter input.
name	This is the name of the passed file.

set_mime_type_file_converter_input

Sets the mime type.

C Synopsis

```
void set_mime_type_file_converter_input
                                (FILE_CONVERTER_INPUT input,
                                 unsigned char* MimeType);
```

Description

This C method sets the mime type.

Argument

Argument	Description
`input`	Specifies the File Converter input.
`MimeType`	The file format that is reported by either a Web server or an e-mail client, in other words, `.html` or application.

3.5 Output Class

The following methods are used for output objects.

init_file_converter_output

Initializes a File Converter output object.

C Synopsis

```
FILE_CONVERTER_OUTPUT init_file_converter_output
                              (void);
```

Description

This C method initializes a File Converter output object.

free_file_converter_output

Frees a `FILE_CONVERTER_OUTPUT` object.

C Synopsis

```
void free_file_converter_output (FILE_CONVERTER_OUTPUT
                                 output);
```

Description

This C method frees a `FILE_CONVERTER_CLIENT_OUTPUT` object.

clear_file_converter_output

Resets a FILE_CONVERTER_CLIENT_OUTPUT object to empty for reuse purposes.

C Synopsis

```
void clear_file_converter_output
                (FILE_CONVERTER_OUTPUT output);
```

Description

This C method resets a FILE_CONVERTER_CLIENT_OUTPUT object to empty for reuse purposes.

Argument

Argument	Description
output	Specifies the FILE_CONVERTER_CLIENT_OUTPUT object.

get_title_file_converter_output

Gets the title of the File Converter output object.

C Synopsis

```
unsigned char* get_title_file_converter_output (
                          FILE_CONVERTER_OUTPUT output);
```

Description

This C method gets the title of the File Converter output object.

Arguments

Argument	Description
output	Specifies the File Converter output object.

Returns

- document title: as extracted from the document
- empty string: if the document title is unavailable

get_author_file_converter_output

Gets the author of the File Converter output.

C Synopsis

```
unsigned char* get_author_file_converter_output
                              (FILE_CONVERTER_OUTPUT output);
```

Description

This C method gets the author of the File Converter output.

Arguments

Argument	Description
output	Specifies the File Converter output object.

Returns

- document author: as extracted from the document
- empty string: if the document author is unavailable

get_text_file_converter_output

Gets the text of the File Converter output object.

C Synopsis

```
unsigned char* get_text_file_converter_output
                    (FILE_CONVERTER_OUTPUT output);
```

Description

This C method gets the text of the File Converter output.

Arguments

Argument	Description
output	Specifies the File Converter output object.

Returns

- document text: as extracted from the document
- empty string: if the document text is unavailable

3.6 Sample Program

The following SAS Document Conversion sample program is for the C API:

```
#ifdef HAVE_CONFIG_H
#include <config.h>
#endif

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <tg/tg_std.h>

#include "file_converter_client.h"

static void usage(char *program_name)
{
     fprintf(stderr, "Usage: %s --server <hostname>:<port>
         [ --mime-type <type> ] <filename>\n", program_name);
     exit(1);
}
int main(int argc, char **argv)
{
     int i;
     char hostname[256];
     int port = -1;
     unsigned char *mime_type = NULL;
     unsigned char *file_name = NULL;

     prarg(&argc, &argv);

     for (i = 1; i < argc; i++)
     {
          if (strcmp(argv[i], "--help") == 0)
```

```
        {
              usage(argv[0]);
        }
        else if (strcmp(argv[i], "--server") == 0)
        {
             if (++i == argc) usage(argv[0]);
             if (sscanf(argv[i], "%256[^:]:%d", hostname,
                         &port) != 2) usage(argv[0]);
        }
        else if (strcmp(argv[i], "--mime-type") == 0)
        {
             if (++i == argc) usage(argv[0]);
             mime_type = (unsigned char *)(argv[i]);
        }
        else
        {
             file_name = (unsigned char *)(argv[i]);
        }
   }
   if ((port < 0) || (file_name == NULL)) usage(argv[0]);
   {
        D_STRING content = init_d_string(1024);
        FILE_CONVERTER_CLIENT client =
                    init_file_converter_client(hostname, port);
        FILE_CONVERTER_INPUT input =
                     init_file_converter_input();
        FILE_CONVERTER_OUTPUT output =
                     init_file_converter_output();

        if (load_filename_d_string_3(content, file_name))
        {
               if (mime_type != NULL)
                   set_mime_type_file_converter_input
                   (input, mime_type);
          set_file_name_file_converter_input(input, file_name);
```

```
                set_content_file_converter_input(input,
                                get_buff_d_string(content),
                                 last_c_d_string(content));

                 if (convert_file_converter_client(client, input,
                           output) == FILE_CONVERTER_STATUS_SUCCESS)
                 {
                      printf("Title: %s\n",
                            get_title_file_converter_output(output));
                      printf("Author: %s\n",
                          get_author_file_converter_output(output));
                      printf("%s\n",
                             get_text_file_converter_output(output));
                 }
                  else
                 {
                           fprintf(stderr, "Error: Cannot convert
                                                 file.\n");
                 }
             }
             else
             {
                           fprintf(stderr, "Error: Cannot load
                                                 file.\n");
             }

             free_d_string(content);
             free_file_converter_client(client);
             free_file_converter_input(input);
             free_file_converter_output(output);
       }

       return 0;
}
```

Index

C

F

G

I

M

O

S

T

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to `yourturn@sas.com`. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to `suggest@sas.com`.