PharmaSUG 2025 - Paper RW-328

Using the S3 Procedure for Object Management in Amazon S3

Kevin Russell, SAS Inc.

Russ Tyndall, SAS Inc.

Abstract

Amazon S3 (AWS S3) is an object storage service that enables customers to store, manage, analyze, and protect their data. Millions of customers worldwide now use the scalability, data availability, improved security, and performance of AWS S3 to store their data. Many of the companies now taking advantage of AWS S3 storage are also SAS customers, so SAS has provided its customers with multiple ways to interface with AWS S3. One of the primary methods SAS provides to interface with AWS S3 is the S3 procedure.

PROC S3 is a Base SAS® procedure that is available in both SAS® 9.4 and SAS® Viya®, which enables SAS customers to perform object management in AWS S3. PROC S3 enables customers to create buckets, directories, and files in AWS S3. PROC S3 also enables customers to list, copy, and delete objects.

The purpose of this paper is to do as follows:

- Provide an overview of PROC S3 and the basic steps needed to start using the procedure.
- Discuss and provide examples of various tasks that users can perform with PROC S3.
- Explain the best method to gather information about the execution of the procedure in the log to assist in debugging when necessary.

Introduction

SAS Technical Support has seen a surge in the number customers who are taking advantage of the ability within AWS S3 to manage and store data. Understanding PROC S3 will enable you to incorporate your AWS S3 environment to SAS.

PROC S3 is an extremely easy-to-use procedure that provides multiple utility tools to help you manage your AWS S3 locations. The procedure consists of the following:

- the PROC statement
- the other statements needed to perform various tasks in your AWS S3 location.

This paper provides examples of some of the statements that are available with PROC S3. Although this paper does not cover all of the available statements, it includes the most commonly used statements.

Table of Contents

Before Using PROC S3

Before you can use PROC S3, you need to obtain your AWS Access Key ID, your Secret Access Key, your Session ID, and your AWS region. Typically, your AWS S3 administrator would provide this information.

Once you obtain this information, it is referenced in the PROC S3 statement. PROC S3 needs this information to make a connection to your AWS S3 environment.

There are a few ways to list these values. The most straightforward method is to list each value with the corresponding option name in the PROC S3 statement.

Here is an example:

```
PROC S3
keyid='your- keyid'
secret='your-secret-key'
```

```
session='your-session-token'
region='your-region';
```

However, you can also list your credentials in the AWS credentials file. The contents of this file could look similar to the following:

```
[default]
aws_access_key_id=
aws_secret_access_key=
aws_session_token=
```

Next, list the other parameters that are used to access AWS S3 in the AWS config file, including parameters like the region. This file could look similar to the following:

```
[default]
region = us-east-1
```

Then, list location of these files as the values of the AWSCRENDENTIALS= and AWSCONFIG= options in the PROC S3 statement.

```
PROC S3 awscredentials='the-complete-path-to-your-aws-
credentials-file' awsconfig='the-complete-path-to-your-aws-
config-file';
```

Finally, you can specify your AWS S3 credentials and other connection parameters in the PROC S3 configuration file. The default PROC S3 configuration file is as follows:

- .tks3.conf in your home directory in UNIX
- tks3.conf in your home directory in Windows

Here is an example of how this file might look:

```
sessionToken=
keyID=
secret=
region=useast
```

Note: If you decide to store this file in a location other than the default, you will need to use the CONFIG= option in the PROC S3 statement to list the file's location.

```
PROC S3 config='the-complete-path-to-your-proc-s3-config-file'
```

You can find a more complete discussion on the PROC S3 configuration file at PROC S3 Configuration File.

Place a File in the AWS S3 Location

MKDIR, CREATE, and PUT Statements

This section explains how to do the following:

- Create a directory.
- Create a bucket.
- Place a file within a bucket in an AWS S3 location.

Three statements within PROC S3 can easily help you complete these tasks: MKDIR, CREATE, and PUT.

1. Use the MKDIR statement to create the new directory in an AWS S3 location. Specify a fully qualified path within quotation marks from the bucket name to the directory name that you are trying to create. In the following example, a new directory is created titled *mynewdir* in the root location. This directory uses the credential settings that were discussed in the **Before Using PROC S3** section.

   The syntax for the MKDIR statement is as follows: `MKDIR "s3-location";`.

   ```
   Proc s3;
     keyid='…'
     secret='…'
     session='…'
     region='us-east-1';
     mkdir '/mynewdir';
   run;
   ```

   Note: The forward slash is necessary prior to the directory name.

2. Use the CREATE statement to create a bucket within the new directory. In the following example, a bucket titled *mynewbucket* is created within the *mynewdir* directory from step one.

   The syntax for the CREATE statement is as follows: `CREATE "bucket-name";`.

   ```
   Proc s3;
     keyid='…'
     secret='…'
     session='…'
   ```

```
  region='us-east-1';
  mkdir '/mynewdir';
  create '/mynewdir/mynewbucket';
run;
```

Note: The forward slash is necessary prior to the directory name. In addition, the name of the AWS S3 bucket that you create must be unique across AWS S3.

3. Lastly, use the PUT statement to take a local file and place it in the newly created AWS S3 bucket. In the following example, the *mytest.csv* file that is located within the **`c:\test`** directory is copied to the *mynewbucket* bucket from step two.

   The syntax for the PUT statement is as follows: `PUT <ENCKEY="key-name"> "local-path" "s3-location";.`

```
Proc s3;
  keyid='…'
  secret='…'
  session='…'
  region='us-east-1';
  mkdir '/mynewdir';
  create '/mynewdir/mynewbucket';
  put "c:\test\mytest.csv"
"/mynewdir/mynewbucket/mytest.csv";
run;
```

Write a Local Directory to an AWS S3 Location

PUTDIR Statement

If you want to write the entire local directory to an AWS S3 location rather than place one file, use the PUTDIR statement.

The syntax for the PUTDIR statement is as follows: `PUTDIR <ENCKEY="key-name"> "local-path" "s3-location";.`

Using the previous examples, suppose you wanted to duplicate the **`c:\test`** directory in an AWS S3 location. You can easily duplicate the directory by using the PUTDIR statement, as shown in the following example:

```
putdir "c:\test" "/mynewdir/mynewbucket";
```

This statement would create the **`c:\test`** directory under the *mynewbucket* bucket. This new directory under the bucket within the AWS S3 location would have all the files that the **`c:\test`** directory contained.

Copy the Contents of an AWS S3 Directory to a Local Path

GETDIR Statement

SAS can also do the opposite of PUTDIR. You can retrieve the contents of an AWS S3 directory and copy them to a local path by using the GETDIR statement.

The syntax for the GETDIR statement is as follows: `GETDIR <ENCKEY="key-name"> "s3-location" "local-path";.`

Here is an example scenario: You have an AWS S3 bucket named *testbucket* that contains two data sets—dsn1 and dsn2. You want to re-create that bucket on a local path called *test*. The following illustrates how to accomplish this task using the GETDIR statement:

```
getdir 'testbucket' '/test';
```

The *test* local directory now contains the dsn1 and dsn2 data sets that the AWS S3 bucket (named *testbucket*) contained.

If the AWS S3 location contains subfolders, they will be copied to the local path as well.
For example, if the *testbucket* contained a folder titled *myfolder*, and *myfolder* contained a data set titled *dsn3*, running the GETDIR statement above would create a *myfolder* subdirectory within the *test* directory. That subdirectory would contain the dsn3 data set.

Copy One File to a Local Path

GET Statement

There might be instances where you do not want to copy the entire AWS S3 bucket contents to a local path. You might want to copy only one file to the local path. You can easily accomplish this task by using the GET statement.

The syntax for the GET statement is as follows: `GET <ENCKEY="key-name"> "s3-location-object" "local-file-path";.`

Here is an example scenario: You can use the GET statement to copy a data set titled *one* from the *testbucket* AWS S3 bucket and place that data set in the local path location of */test*.

```
get '/testbucket/one.sas7bdat' '/test/one.sas7bdat';
```

Note: The forward slash is necessary prior to the bucket name.

Copy an Object from One AWS S3 Source Location to Another AWS S3 Destination

COPY Statement

You can use the COPY statement to copy an object from one AWS S3 source location to another AWS S3 destination.

The syntax for the COPY statement is as follows: `COPY <SRCKEY="key-name"> "source-s3-location" <ENCKEY="key-name"> "destination-S3-location";.`

Here is an example scenario: You have a data set titled *one* that existed in the *testbucket* AWS S3 bucket. However, you want to copy that data set to a different bucket titled *test2bucket*. You can copy that data set to the new bucket by using the COPY statement.

```
copy '/testbucket/one.sas7bdat' '/test2bucket/one.sas7bdat';
```

Note: The forward slash is necessary prior to the directory name.

List the Contents of an AWS S3 Bucket

LIST Statement

PROC S3 can also list the contents of a bucket with the LIST statement. In addition, starting in SAS® 9.4M7 (TS1M7) and SAS® Viya® 3.5, you can use the OUT= option to write the list contents to a file.

The syntax for the LIST statement is as follows: `LIST <_SHORT_> "s3-location" <OUT=filename >;.`

In the following scenario, you have a bucket titled *testbucket* that contains three files. The example below uses the LIST statement to view the contents of that bucket:

```
proc s3
keyid='xxx'
secret='xxx'
session='xxx'
region='us-east-1';
list 'testbucket';
run;
```

The output generated in the log is as follows:

```
aa.txt                    5 2025-03-10T15:35:29.000Z
one.sas7bdat 131072 2025-03-10T15:34:53.000Z
test.xlsx             5562 2025-03-10T15:35:28.000Z
```

You can list just the file names themselves with the _SHORT_ option as follows: `list _short_ 'testbucket';.`

The output generated in the log is as follows:

```
aa.txt
one.sas7bdat
test.xlsx
```

The OUT= option can also write the contents to a SAS data set rather than the log. You must list the value for OUT= as a two-level name like *WORK.DSN* or list a fileref where a FILENAME points to the file that you want to create. Here is an example:

```
proc s3
keyid='xxx'
secret='xxx'
session='xxx'
region='us-east-1';
list 'testbucket' out=work.mylist;
run;

proc print data=mylist;
run;
```

The output generated in the SAS data set is as follows:

Obs	name	size	lastModified	owner	hash
1	aa.txt	5	10MAR25:15:35:29	awssandboxroot01	"5ba48b6e5a7c4d4930fda256f411e55b"
2	one.sas7bdat	131072	10MAR25:15:34:53	awssandboxroot01	"60a2a430bb90864f21f515be200add8e"
3	test.xlsx	5562	10MAR25:15:35:28	awssandboxroot01	"662cacd8d2fb7752cd1b3b8618779f70"

Remove the Contents from an AWS S3 Bucket

DELETE Statement

If the wrong file was copied by mistake or you want to remove an object from an AWS S3 bucket, use the DELETE statement.

The syntax for the DELETE statement is as follows: `DELETE "s3-location";`.

For example, you have a bucket titled *testbucket* and it contains the *test.csv* file, which you want to remove. You can use the DELETE statement to remove the file, as shown below. (Note that you must fully qualify the AWS S3 location from the bucket name to the object name.)

```
Delete '/testbucket/test.csv';
```

Note: The forward slash is necessary prior to the bucket name.

Enhanced Logging to Help with PROC S3 Debugging

When an error occurs during the execution of PROC S3, it is almost always necessary to enable enhanced logging to help determine the cause of the error. You can enable enhanced logging with the SAS-supplied log4SAS autocall macros. These macros provide additional information about the underlying SAS connection to the AWS S3 storage system.

You can invoke these macros right before your PROC S3 step, as shown in the following example:

```
%log4sas();
%log4sas_logger("App.tk.s3","level=trace");
%log4sas_logger("App.tk.htclient","level=trace");
<proc s3 or filename s3 statements go here>
```

Note that it is necessary to invoke the log4sas macro first and then the lo4sas_logger macros to specify the AWS S3 and HTCLIENT logging.

When running SAS 9.4 or SAS Viya 3.5, the log files are written to the log location specified by the LOGCONFIGLOC system option. When executing in the SAS® Viya® platform, though, you need to complete the following steps to generate the log.

1. A `SAS Studio compute context` message is displayed in the top right corner of SAS® Studio when the compute session is ready. When the session is ready, find the pod for that session. The pod should be one of the newest sas-compute-server pods.

   ```
   NS=<your namespace>
   kubectl -n $NS get pods | grep sas-compute-server
   ```

   Note: You can also filter for jobs on your user name.

```
kubectl -n $NS get pod -l launcher.sas.com/username=<your
username here>
```

2. Once you find the pod, send the logs to a file. (Use `logs -f` as shown below to tail or follow the log.)

```
kubectl -n $NS logs -f sas-compute-server-pod-name-found-above >
/tmp/comp.log
```

3. While the logs are being sent, run the full PROC S3 program with the following statements at the top:

```
%log4sas();
%log4sas_logger("App.tk.s3","level=trace");
%log4sas_logger("App.tk.htclient","level=trace");
```

4. Once the program finishes, you can kill kubectl. Send /tmp/comp.log, which contains the program that you just executed and the trace logging.

If you are still unable to determine the cause of the error, contact SAS Technical Support. Be sure to include the following:

- your PROC S3 code
- the enhanced logging

In addition, run the following %PUT statements and send in the resulting log. These statements write out your host name, your operating system, and your exact release of SAS.

```
%PUT &=SYSHOSTNAME;

%PUT &=SYSSCPL;

%PUT &=SYSVLONG;

%PUT &=SYSSITE;
```

Conclusion

The use of AWS S3 is growing at a staggering rate in today's business world. PROC S3 provides SAS users with a simple tool to interact with their AWS S3 environment. PROC S3 enables you to move, copy, and list files as well as create new buckets and directories within each bucket. SAS is continuously enhancing its ability to interact with the AWS S3 environment, so see Cloud Options: Advanced Cloud Analytics & Services for more information about SAS and AWS S3.

Acknowledgments

We would like to thank Ben Kitchens for all of his help with this paper as well as Sierra Thoemmes for her exceptional editing.