

An Insider's Guide to SAS/ACCESS to Hadoop

Jeff Bailey, SAS Institute Inc.

ABSTRACT

In the very near future Hadoop will likely find its way into your SAS environment. Hadoop is rapidly displacing database management systems in the corporate world and rearing its head in the SAS world. If you think now is the time to learn how to use SAS with Hadoop, you are in luck. This workshop is the jump start you need.

This workshop show you how to access Hadoop using SAS. We devote most of our time to exploring SAS/ACCESS to Hadoop. During the workshop you will learn:

- How to configure your SAS environment so that you can access Hadoop
- How to use the Hadoop FILENAME statement
- How to use Hadoop Procedure
- How to use SAS/ACCESS Interface to Hadoop.

INTRODUCTION

This workshop has been designed for people with no Hadoop experience. The vast majority of the workshop will be hands on. In fact, we begin all discussions with hands-on exercises. This is a departure from the traditional lecture, exercise, discuss approach typically used.

Why are we using an “exercise first” approach? The reason is simple, discussing topics which you have experience with is more meaningful than discussing some theoretical topic. Discussing a topic when you have no experience creates the illusion of understanding. The “exercise first” approach leads to deeper learning.

Unfortunately, “exercise first” is unfamiliar and you may not like it. The exercise-first approach will provide you with actual experience and a much deeper understanding of the topic. You will like the experience and understanding provided by this approach. After attending this workshop you will understand how and why to use SAS/ACCESS to Hadoop, the Hadoop FILENAME statement, and the Hadoop procedure. Because you will have actual experience using the products, you will be able to carry on meaningful conversations about SAS/ACCESS Interface to Hadoop.

During the workshop you will run SAS code (conveniently provided this code for you). This code is special. It includes learning materials: ideas and questions. These questions are the workshop. Please make an effort to answer them. It is best if you write the answers on a sheet of paper (or computer, etc.). You may want to discuss them with your or neighbor. There is no better way to fully understand a topic than to discuss it with real people. Your goal is to understand the topics well enough to have a meaningful conversation.

Once we complete an exercise we will discuss it and dive into some areas a little deeper. Our goal is to discover insights. These insights will not be detailed in the exercise code.

THE WORKSHOP ENVIRONMENT

SAS 9.4M3 - Is installed on each of the workshop computers. You can run SAS Display Manager or SAS Studio web app. The choice is up to you.

Cloudera 5.5 QuickStart – this is a virtual machine. We have installed this on all the lab machines. If you want to run it on your computer you can download the QuickStart from the Cloudera web site. We are running this virtual machine in the VMWare player. You can download this VM here:

http://www.cloudera.com/downloads/QuickStart_vms/5-5.html.

Workshop Files – The files for this workshop are in the following directory:

C:\HOW\bailey

The workshop directory will contain the following subdirectories:

\data – contains data files used for exercises

\code – contains the SAS code used for the exercises

\notes – contains the course notes

Reality check: We have gone to extraordinary lengths to ensure our lab environment works.

Unfortunately this environment is fragile. Fortunately, the fragility of the environment provides us with a great learning opportunity. The laptops you are running on only have 16GB of RAM. There is a chance that while the person sitting next to you is happily doing the exercises while your environment will not work, at all.

Do not get frustrated. We have prepared for this. The SAS log output is included in the appendix of this doc. Use it as a pencil and paper exercise. Or, partner with someone which a working environment.

EXERCISE 0 – SANITY CHECK (LET'S MAKE SURE THE ENVIRONMENT WORKS)

In order to complete the workshop we need to verify that your environment is working properly. Chances are the environment is not going to work, immediately. This is probably the first time anyone has said this in front of so many people.

Understand: The SAS environment has been properly configured to work. Any problem you encounter is a real problem. Exercise 0 is not a test; the code should run.

Type the following code into SAS and submit it:

```
libname mycdh hadoop server='quickStart.cloudera' user='cloudera'
password='cloudera';
```

There is a SAS file in the code directory (C:\NOW\bailey\code). You can use it if you want to avoid typing.

If you encounter a problem, verify that you typed the LIBNAME statement correctly. If the code is correct and it does not work, open a DOS Command Prompt and see if you can ping the CDH VM:

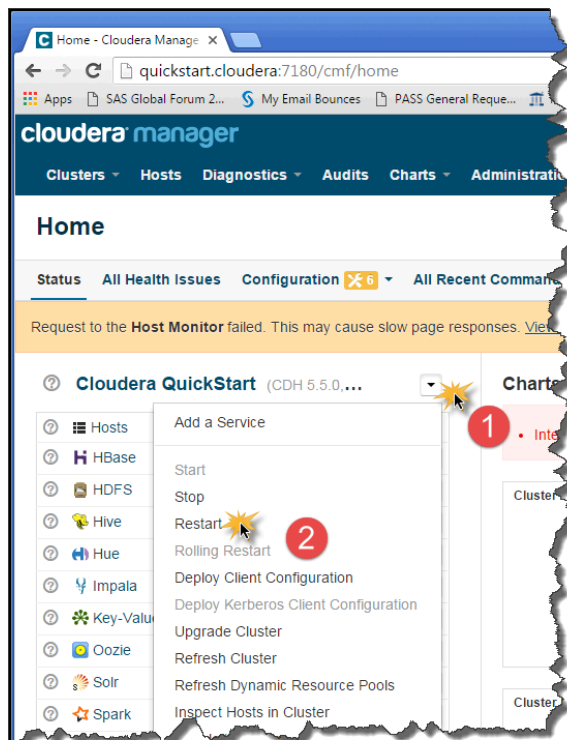
```
ping quickStart.cloudera
```

If you can ping the VM, when restart the Cloudera QuickStart. Here's how:

Open a browser on the workshop PC and enter this URL: <http://QuickStart.cloudera:7180>

Logon: cloudera/cloudera

Use Cloudera Manager to restart the Cloudera processes.



It will (should) take about 5 minutes for the cluster to restart. Retry the LIBNAME statement. If it doesn't work, you can reboot your VM. You may need to restart the Cloudera QuickStart cluster (repeat the above steps).

If rebooting doesn't fix the problem, raise your hand and ask for help.

Understand: the workshop environment is designed for functionality, not performance.

C:\HOW\Bailey – contains data and programs that will be used for the workshop. It is a great idea to open a Windows Explorer window and point it to this directory. That way you will remember where it is.

HUE – Hadoop User Experience. This is a web-based UI that you can use to explore the data that lives on your cluster. Use it to truly understand what is taking place and how it impacts Hadoop. The beauty of HUE is that it can help make working with SAS and Hadoop much easier. Here is the link:

<http://QuickStart.cloudera:8888> – Userid/Password: cloudera/cloudera

Insight: working with the QuickStart VM is fraught with peril. Sometimes it doesn't work well. If that happens during the workshop we may have to improvise. Fortunately, we have a plan B, and C...

EXERCISE 1 – ARCHITECTURE (THE PARTS THAT MAKE THIS WORK)

Goal: Understand the configuration points and what it takes to make all this stuff work together.

Consider the following questions:

- What is required for SAS/ACCESS to Hadoop connect to a cluster?
- Where do the Hadoop FILENAME statement and Hadoop procedure get their connection information?
- What is the Cloudera QuickStart?
- What does the SAS_HADOOP_RESTFUL= environment variable do?

It is extremely important that you not make any changes to the environment. If you inadvertently alter the environment it may stop work working.

Open a Windows Explorer and navigate to `C:\SAS_HADOOP_JAR_PATH`.

- Look at the list of JAR files – pay attention to the names. How many are there? What do you think would happen if you were to delete some of them?

Navigate to `C:\SAS_HADOOP_CONFIG_PATH`.

- Open `core_site.xml` using Notepad. What is the value of `Fs.defaultFS` (it is close to the top of the file)? Compare this value to the `EX0 LIBNAME` statement.
- Close Notepad. Do not save changes to the file.

Open the hosts file (`C:\Windows\System32\drivers\etc`).

- Notice that an entry for `quickStart.cloudera` has been added.
 - If there is a problem with your environment an incorrect IP address is likely the cause.
- Previously, we asked you to issue the ping command. If you didn't, do it now.
 - `ping QuickStart.cloudera`
 - Leave the Command Prompt window open.

Finally, let us look at a handful of environment variables. In a Command Prompt window (hopefully, you left it open) run the following command:

- `env | find "SAS_HADOOP"`
- What do you think these environment variables do? Hint: two of them will point to directories you reviewed previously.
- Challenge: Can you figure-out what `SAS_HADOOP_RESTFUL=` does?

Insight: Understanding how all the pieces fit together enables you to solve problems when you begin using SAS and Hadoop. What first appears to be trivia becomes very important when you have a deadline and you cannot connect to your Hadoop environment via SAS.

Once you have finished this exercise feel free to wave at me!

EXERCISE 2 – TABLES AND FILES

Goal: Hadoop has the concept of tables and files. They are similar, but different.

Consider the following questions:

- If I drop a Hive table is the underlying data deleted?
- Can I define a Hive table so that it points to an existing HDFS file?
- What happens if the table definition doesn't match the data?
- Why would I use the SAS Hadoop FILENAME statement?
- Why would I use the Hadoop procedure?

Use the following SAS file for this exercise:

`C:\HOW\bailey\exercises\ex02_sas_hadoop_sgf_2016.sas`

```

/*****/
/* Exercise 02 - SAS3880 */
/* An Insider's Guide to SAS/ACCESS to Hadoop */
/* */
/* Explore Hive tables, HDFS files, and how they relate. */
/*****/

/*****/
/* We use to have to do this type of thing... now we don't */
/* filename cfg 'C:\SAS_HADOOP_CONFIG_PATH\core-hdfs-merged.xml'; */
/*****/

/*****/
/* Setup the Cloudera environment */
/* PROC HADOOP is great for setup and clean up */
/*****/

/*****/
/* - create a directory in the Hadoop File System (HDFS) */
/* - copy a file to HDFS */
/*****/
proc hadoop username="cloudera" verbose;
    HDFS mkdir='/user/cloudera/textfile';
    HDFS copyfromlocal='C:\HOW\bailey\data\sgf_characters.txt'
        out='/user/cloudera/textfile/sgf_characters.txt';
run;

/*****/
/* Locate the HDFS file using the HUE file browser */
/* and login */
/* */
/* http://quickstart.cloudera:8888 */
/* cloudera/cloudera */
/*****/

/*****/
/* Display the contents of the sgf_characters.txt file using SAS */
/*****/

/*****/
/* Open core-hdfs-merged.xml using NOTEPAD. */
/* What do you think the purpose of the file is? */
/* Why do you think SAS requires the file? */
/* Where does the sgf_characters.txt file live? */
/*****/

filename sgftext hadoop '/user/cloudera/textfile/sgf_characters.txt'
user=cloudera;

data _null_;
    infile sgftext;
    input mycharacter $1.;

```

```

    put mycharacter;
run;

/*****
/* Using ACCESS to Hadoop */
*****/

/*****
/* Create a table using ACCESS to Hadoop */
*****/
proc sql;
    connect to hadoop(server='quickstart.cloudera' user=cloudera);
    execute (create external table mytext(c1 varchar(1))
            stored as textfile
            location '/user/cloudera/textfile') by hadoop;
quit;

/* let's verify that the table is there and SAS can read it. */

libname mycdh hadoop server='quickstart.cloudera' user=cloudera;
options fullstimer;

/*****
/* We created a table but did not load it.      */
/* - Where did the data come from?              */
*****/
proc sql;
    connect to hadoop(server='quickstart.cloudera' user=cloudera);
    select count(*) from connection to hadoop
        (select * from mytext);
quit;

/*****
/* Let's ask the same question in a slightly different way. */
/* - Passing the count() function to Hadoop should make      */
/*   the query perform better, right?                          */
/* - Do the results match your expectations?                  */
/* - Why is there a performance difference?                    */
*****/
proc sql;
    connect to hadoop(server='quickstart.cloudera' user=cloudera);
    select * from connection to hadoop
        (select count(*) from mytext);
quit;

/* Why is the second version is much faster. Point out that */
/* the count(*) was passed to Hive. Or was it? How do we know? */
/* Revisit this. It will keep their attention.                */

/*****
/* Drop the table.                                          */
/* - What happens to the HDFS file?                        */
*****/

```

```

/* - Why would this behavior be valuable? */
/*****/

proc sql;
    drop table mycdh.mytext;
quit;

/*****/
/* View the mycdh library in SAS Explorer to make sure the table is gone. */
/*****/

/*****/
/* Run the DATA step again. */
/* - Does the DATA step work? If so, why? If not, why? */
/*****/
data _null_;
    infile sgftext;
    input mycharacter $1.;
    put mycharacter;
run;

/*****/
/* delete the textfile HDFS directory. */
/* What happens to the sgf_characters.txt file? */
/*****/

proc hadoop username="cloudera" verbose;
    HDFS delete='/user/cloudera/textfile';
run;

/*****/
/* Now for something really interesting... */
/* Run the following code describe what happens. */
/* - Why is this a big deal? */
/*****/

proc hadoop username="cloudera" verbose;
    HDFS mkdir='/user/cloudera/digitsfile';
    HDFS copyfromlocal='C:\HOW\bailey\data\sgf_digits.txt'
        out='/user/cloudera/digitsfile/sgf_digits.txt';
run;

proc sql;
    connect to hadoop (server='quickstart.cloudera' user=cloudera);
    execute (create table mydigits(myint int)
        stored as textfile
        location '/user/cloudera/digitsfile') by hadoop;
quit;

proc sql;
    select * from mycdh.mydigits;
quit;

```

```

/*****/
/* If this works it may help you see what this is a big deal */
/*****/

filename sgfdigit hadoop '/user/cloudera/digitsfile/sgf_digits.txt'
user=cloudera;

data _null_;
    infile sgfdigit;
    input mydigits $1.;
    put mydigits;
run;

/*****/
/* Clean up */
/*****/

proc sql;
    drop table mycdh.mydigits;
quit;

/*****/
/* End Exercise 02 */
/*****/

```

EXERCISE 3 – JOINS AND CTAS

Goal: JOINS are expensive, it pays to do them where the data lives. Create Table AS is very similar. It pays to do this work on the data source without moving data into SAS.

Consider the following questions:

- How do I find the SQL that SAS is sending to Hadoop?
- Why is it important that Hive perform join operations?
- What, exactly, causes SAS to perform the join operation?
- What is CTAS?
- Why is it important for Hive to perform CTAS processing?

Use the following SAS file for this exercise:

C:\HOW\bailey\code\ex03_sas_hadoop_sgf_2016.sas

```

/*****/
/* Exercise 03 - SAS3880 */
/* An Insider's Guide to SAS/ACCESS to Hadoop */
/*
/* Explore CREATE TABLE AS and multi-libname Join Processing */
/*****/

/*****/
/* Setup the Cloudera environment */
/* Move data from SASHELP to CDH */
/* Create schemas using explicit pass-thru */
/* CTAS - CREATE TABLE AS */
/*****/

```



```

/* create the test tables */
/* Create two simple tables */
libname cdh hadoop server="quickStart.cloudera" user=cloudera
password=cloudera;

data cdh.table1;
    x=3; output;
    x=2; output;
    x=1; output;
run;
data cdh.table2;
    x=3; y=3.3; z='three'; output;
    x=2; y=2.2; z='two'; output;
    x=1; y=1.1; z='one'; output;
    x=4; y=4.4; z='four'; output;
    x=5; y=5.5; z='five'; output;
run;

/*****/
/* verify that both tables were created */
/*****/

/*****/
/* TEST: change the following line to display the SQL */
/* that is being passed to Hive. */
/* */
/* type in correct code and submit it. */
/*****/
* options ??????=',,,d' sastraceloc=????? nostsuffix;
options sastrace=',,,d' sastraceloc=saslog nostsuffix;

/* single schema join - does Hive process the join? */
proc sql;
    select t1.x, t2.y, t2.z
        from cdh.table1 t1
            , cdh.table2 t2
        where t1.x=t2.x;
quit;

/*****/
/* multi-LIBNAME join */
/* - does Hive process the join? */
/* - if not, why? */
/*****/

libname cdh1 hadoop server="QuickStart.cloudera" user=cloudera
password=cloudera;
libname cdh2 hadoop server="QuickStart.cloudera" user=cloudera
password=cloudera;

proc sql;
    select t1.x, t2.y,t2.z
        from cdh1.table1 t1

```

```

        , cdh2.table2 t2
    where t1.x=t2.x;
quit;

/*****
/* multi-LIBNAME join */
/* - does Hive process the join? */
/* - if not, why? */
*****/
libname cdh1 hadoop server="QuickStart.cloudera" ;
libname cdh2 hadoop server="QuickStart.cloudera"
            user=cloudera password=cloudera;

proc sql;
    select t1.x, t2.y, t2.z
        from cdh1.table1 t1
            , cdh2.table2 t2
        where t1.x=t2.x;
quit;

/*****
/* multi-LIBNAME join */
/* - does Hive process the join? */
/* - if not, why? */
*****/
libname cdh1 hadoop server="QuickStart.cloudera";
libname cdh2 hadoop server="quickStart2.cloudera";

proc sql;
    select t1.x, t2.y, t2.z
        from cdh1.table1 t1
            , cdh2.table2 t2
        where t1.x=t2.x;
quit;

/*****
/* Setup Multi-schema Test */
*****/

/*****
/* create two Hive schemas for testing */
*****/ proc
sql;
    connect to hadoop (server="QuickStart.cloudera"
                        user=cloudera password=cloudera);
    execute(create schema schema1) by hadoop;
    execute(create schema schema2) by hadoop;
quit;

/*****
/* CREATE TABLE AS */
/* - view the SQL that is passed to Hive and */
/* describe how they are different. */
*****/

```

```

/*      - which is better?                                */
/*****/
libname cdhsch1 hadoop server="QuickStart.cloudera"
                user=cloudera password=cloudera
                schema=schema1;

libname cdhsch2 hadoop server="QuickStart.cloudera"
                user=cloudera password=cloudera
                schema=schema2;

options dbdirectexec;

proc sql;
    create table cdhsch1.table1 as select * from cdh.table1;
quit;

options nodbidirectexec;

proc sql;
    create table cdhsch2.table2 as select * from cdh.table2;
quit;

/*****/
/* multi-LIBNAME multi-SCHEMA join                        */
/*      - do you think Hive will process the join        */
/*      when SCHEMA= values are different?                */
/*****/

libname cdhsch1 hadoop server="QuickStart.cloudera"
                user=cloudera password=cloudera schema=schema1;

libname cdhsch2 hadoop server="QuickStart.cloudera"
                user=cloudera password=cloudera schema=schema2;

proc sql;
    select t1.x, t2.y, t2.z
        from cdh1.table1 t1
            , cdh2.table2 t2
        where t1.x=t2.x;
quit;

/*****/
/* Clean up */
/*****/

proc sql;
    drop table cdh.table1;
    drop table cdh.table2;
    drop table cdhsch1.table1;
    drop table cdhsch2.table2;
quit;

/*****/
/* drop Hive schemas */
/*****/

```

```

proc sql;
    connect to hadoop (server="quickStart.cloudera"
                      user=cloudera password=cloudera);
    execute(drop schema schema1) by hadoop;
    execute(drop schema schema2) by hadoop;
quit;

/*****
/* End Exercise 03 */
*****/

```

EXERCISE 4 – BULKLOAD

SAS/ACCESS Interface to Hadoop includes the capability to rapidly load data. In order to do this SAS/ACCESS Interface to Hadoop by-passes the Hive JDBC driver and writes directly to the Hadoop distributed file system (HDFS). This is an extremely important feature.

Goal: Know how to effectively move data from SAS into Hadoop.

Consider the following questions:

- What is bulk loading?
- How do I tell Hive to use a Parquet file to store the Hive data?

Use the following SAS file for this exercise:

C:\SAS_Hadoop_Workshop\code\ex04_sas_hadoop_sgf_2016.sas

```

/*****
/* Exercise 04 - SAS3880 */
/* An Insider's Guide to SAS/ACCESS to Hadoop */
/*
/* Explore BULKLOAD= */
*****/
/*****
/*****
/* First: bulkload=yes the sashelp.cars SAS data set into CDH */
/* Second: bulkload=no sashelp.cars and compare to first. */
/* Third: bulkload=yes sashelp.cars underlying parquet file. */
*****/

libname mycdh hadoop server="QuickStart.cloudera"
                    user=cloudera password=cloudera;

options sastrace=',,,d' sastraceloc=saslog nostsuffix;

/* First we bulk load then we run a non-load append */

/*****
/* The following warning occurs because the cars data set has in index */
/* defined on it. You can ignore it. */
/*
/*
/* WARNING: Engine HADOOP does not support SORTEDBY operations. SORTEDBY*/
/* information cannot be copied. */
*****/

proc append base=mycdh.cars (bulkload=yes)
            data=sashelp.cars;

```

```

run;

/*****
/* clean up so we can rerun */
*****/
proc sql;
    drop table mycdh.cars;
quit;

/*****
/* How does bulkloading differ from non-bulkload loading? */
*****/

proc append base=mycdh.cars
            data=sashelp.cars;
run;

/*****
/* clean up so we can rerun */
*****/
proc sql;
    drop table mycdh.cars;
quit;

/*****
/* Is loading into a parquet file different? */
*****/

proc append base=mycdh.cars (dbcreate_table_opts='stored as parquetfile')
            data=sashelp.cars;
run;

/*****
/* Clean up */
*****/
proc sql;
    drop table mycdh.cars;
quit;

/*****
/* End Exercise 04 */
*****/

```

EXERCISE 5 – 32K STRING THING

Goal: Understand the impact that the 32k string thing has in your code. Ask me about the SASWORK tech support track.

Consider the following questions:

- What on earth is the 32K String Thing?
- What is the impact of having 32K strings in my SAS data sets?
- How do I know this has happened?

- How do I stop it from happening?

Use the following SAS file for this exercise:

C:\SAS_Hadoop_Workshop\code\ex05_sas_hadoop_sgf_2016.sas

```

/*****
/* Exercise 05 - SAS3880 */
/* An Insider's Guide to SAS/ACCESS to Hadoop */
/* */
/* 32k string thing */
*****/

/* setup the environment */
/* Copy single_string_column.txt to HDFS. */
proc hadoop username="cloudera" verbose;
HDFS COPYFROMLOCAL="C:\HOW\bailey\data\single_string_column.txt"
OUT='/user/cloudera/letters/single_string_column.txt';
run;

proc sql;
  connect to hadoop (server='QuickStart.cloudera'
                    user=cloudera password=cloudera);
  execute (create table letters (single_character string)
          row format delimited fields terminated by ','
          location '/user/cloudera/letters') by hadoop;
  disconnect from hadoop;
quit;

/*****
/* How many rows (a single character) are in the table? */
*****/
libname mycdh hadoop server='QuickStart.cloudera'
                    user=cloudera password=cloudera;

proc sql;
  select count(*) from mycdh.letters;
quit;

/*****
/* Using implicit pass-thru, create a SAS data set from letters */
*****/
data work.letters;
  set mycdh.letters;
run;

/*****
/* Before they added the warning, this was a lot more fun. */
/* How big is the work.letters data set? */
/* Does the size (180 characters) seem reasonable? */
*****/
proc datasets library=work;
quit;

/*****

```

```

/* Let's fix it */
/*****/
proc sql;
    connect to hadoop (server='QuickStart.cloudera'
                      user=cloudera password=cloudera);
    execute (alter table letters set tblproperties
            ('SASfmt:single_character'='CHAR(1)')) by hadoop;
quit;

/*****/
/* Did the ALTER TABLE fix it */
/*****/
data work.letters_fixed;
    set mycdh.letters;
run;

proc datasets library=work;
quit;

/*****/
/* Now for something completely different */
/*****/
data mycdh.letters_implicit (drop=i);
    do i = 1 to 180;
        single_character='a';
        output;
    end;
run;

/*****/
/* copy the new table from Hive into Hadoop and check the file size */
/*
/*
/* How large is the resulting work.letters_implicit
/* SAS data set? Guess then run the following code.
/*****/
data work.letters_implicit;
    set mycdh.letters_implicit;
run;

proc datasets library=work;
quit;

/*****/
/* Why did it work? */
/*****/
proc sql;
    connect to hadoop (server='QuickStart.cloudera'
                      user=cloudera password=cloudera);
    select * from connection to hadoop
            (describe formatted letters_implicit);
    disconnect from hadoop;
quit;

/*****/

```

```

/* Do you see it? */
/* Checkout the data type for the single_character column. */
/* It's funny that the above code generates the 32k message */
/*****/

/*****/
/* Clean up */
/*****/
proc sql;
    drop table work.letters;
    drop table work.letters_fixed;
    drop table work.letters_implicit;
    drop table mycdh.letters;
    drop table mycdh.letters_implicit;
quit;

/*****/
/* End Exercise 05 */
/*****/

```

EXERCISE 6 – IN-DATABASE PROCEDURES (IF WE HAVE TIME)

Goal: See an example of in-database PROC push-down to Hadoop.

Consider the following questions:

- What does in-database PROC push-down mean?
- How do I get it to happen?
- How do I stop it from happening?

Use the following SAS file for this exercise:

C:\SAS_Hadoop_Workshop\code\ex06_sas_hadoop_sgf_2016.sas

```

/*****/
/* Exercise 06 - SAS3880 */
/* An Insider's Guide to SAS/ACCESS to Hadoop */
/* */
/* In-Database PROC Example */
/*****/

libname mycdh hadoop server='QuickStart.cloudera'
                    user=cloudera password=cloudera schema='default';

data mycdh.class;
    setsashelp.class;
run;

/*****/
/* Run RANK procedure in-database */
/* How can you tell it was processed in-database? */
/*****/
options sastrace=',,,d' sastraceloc=saslog nostsuffix;

/*****/

```



```

/* By default it runs in-database */
/*****/
proc rank data=mycdh.class out=work.class_rank;
    by descending weight;
run;

/*****/
/* You can turn it off. */
/* But you probably should not do that. */
/*****/
options sqlgeneration=(DBMS EXCLUDEDB='HADOOP');

proc rank data=mycdh.class out=work.class_rank;
    by descending weight;
run;
proc print data=work.class_rank;
run;

/*****/
/* Clean up */
/*****/
proc sql;
    drop table work.class_rank;
    drop table mycdh.class;
quit;

/*****/
/* End Exercise 06 */
/*****/

```

CONCLUSION

Your next step is to use this new knowledge in your everyday work. Make sure you read the SAS/ACCESS documentation. It will help you get the most from your SAS and Hadoop investment. You will be happy you did.

REFERENCES

SAS 9.4 Hadoop Configuration Guide for Base SAS and SAS/ACCESS. Available at <https://support.sas.com/resources/thirdpartysupport/v94/hadoop/hadoopbacg.pdf>

SAS/ACCESS 9.4 for Relational Databases: Reference, Seventh Edition. Available at <http://support.sas.com/documentation/cdl/en/acreldb/68028/PDF/default/acreldb.pdf>.

ACKNOWLEDGMENTS

The author extends his heartfelt thanks and gratitude to the following individuals:

Pat Buckley, SAS Institute Inc.

Marie Dexter, SAS Institute Inc.

Antonio Gianni, SAS Institute Inc.

Tanya Gillikin, SAS Institute Inc.

Jerry Hosking, SAS Institute Inc.

Salman Maher, SAS Institute Inc.

Bill Oliver, SAS Institute Inc.

Michael Williams, SAS Institute Inc.

RECOMMENDED READING

- *SAS® and Hadoop Technology: Overview*. Available at <http://support.sas.com/documentation/cdl/en/hadoopov/68100/PDF/default/hadoopov.pdf>.
- *SAS/ACCESS® 9.4 for Relational Databases: Reference, Seventh Edition*. Available at <http://support.sas.com/documentation/cdl/en/acreldb/68028/PDF/default/acreldb.pdf>.

CONTACT INFORMATION

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

Jeff Bailey
SAS Institute Inc.
Jeff.Bailey@sas.com
www.linkedin.com/in/jeffreydbailey

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies