## Contents

**Chapter 1 • Verifying Your Hadoop Environment**
- Pre-Configuration Checklist for SAS Software That Interfaces with Hadoop ....... 1

**Chapter 2 • SAS Viya and SAS/ACCESS Software with Hadoop**
- Introduction .................. 3

**Chapter 3 • Configuring FILENAME Statement Hadoop Access Method and PROC HADOOP**
- Overview of Steps to Configure the FILENAME Statement and PROC HADOOP .... 5
- Prerequisites for the FILENAME Statement and PROC HADOOP .................. 6
- Making Hadoop JAR and Configuration Files Available to the SAS Client Machine ...... 6
- Using WebHDFS or HttpFS ........................................ 6
- Using Apache Knox Gateway Security .................................. 8
- Using Apache Oozie ................................................. 8
- Validating the FILENAME Statement and PROC HADOOP to Hadoop Connection .... 10
- Documentation for Using the FILENAME Statement and PROC HADOOP ............ 10

**Chapter 4 • Configuring SAS/ACCESS for Hadoop**
- Overview of Steps to Configure SAS/ACCESS Interface to Hadoop ................. 13
- Prerequisites for SAS/ACCESS Interface to Hadoop .................................. 14
- Making Hadoop JAR and Configuration Files Available to the SAS Client Machine ... 15
- Security and User Access to Hadoop ........................................... 15
- Using WebHDFS or HttpFS ............................................. 17
- Working with Hive .................................................. 18
- Validating Your SAS/ACCESS to Hadoop Connection ................................. 20
- Documentation for Using SAS/ACCESS Interface to Hadoop ......................... 21

**Appendix 1 • SAS Environment Variables for Hadoop**
- Dictionary .................................................. 23

**Recommended Reading** ........................................ 29

**Index** .................................................. 31
Chapter 1

Verifying Your Hadoop Environment

Pre-Configuration Checklist for SAS Software That Interfaces with Hadoop

A good understanding of your Hadoop environment is critical to a successful configuration of SAS software that interfaces with Hadoop.

Before you configure SAS software that interfaces with Hadoop, it is recommended that you verify your Hadoop environment by using this checklist:

• Gain working knowledge of the Hadoop distribution that you are using (for example, Cloudera or Hortonworks).
  
  You also need working knowledge of the Hadoop Distributed File System (HDFS), MapReduce 1, MapReduce 2, YARN, and HiveServer2 services. For more information, see the Apache website or the vendor’s website.

  For MapR, you must install the MapR client. The installed MapR client version must match the version of the MapR cluster that SAS connects to. For more information, see MapR: Setting Up the Client.

• Confirm that the HCatalog, HDFS, HiveServer2, MapReduce, Oozie, Sqoop, and YARN services are running on the Hadoop cluster. SAS software uses these various services, and this confirmation ensures that the appropriate JAR files are gathered during the configuration.

• Know the location of the MapReduce home.

• Know the host name of the Hive server and the name of the NameNode.

• Determine where the HDFS and Hive servers are running. If the Hive server is not running on the same machine as the NameNode, note the server and port number of the Hive server for future configuration.

• Request permission to restart the MapReduce service.

• Verify that you can run a MapReduce job successfully.

• Understand and verify your Hadoop user authentication.

• Understand and verify your security setup.

  It is highly recommended that you enable Kerberos or another security protocol for data security.
Verify that you can connect to your Hadoop cluster (HDFS and Hive) from your client machine outside of the SAS environment with your defined security protocol.
Chapter 2
SAS Viya and SAS/ACCESS Software with Hadoop

Introduction

This document provides post-installation configuration information that enables you to use the following SAS components that access Hadoop:

- SAS Viya components
  - `FILENAME Statement Hadoop Access Method`
    enables SAS Viya users to use Hadoop to read from or write to a file in HDFS.
  - HADOOP procedure
    enables SAS Viya users to submit HDFS commands, Pig language code, and MapReduce programs against Hadoop data. PROC HADOOP interfaces with the Hadoop JobTracker, which is the service within Hadoop that controls tasks to specific nodes in the cluster.

- SAS/ACCESS Interface to Hadoop
  enables you to interact with your data by using SQL constructs through HiveServer2. SAS/ACCESS Interface to Hadoop also enables you to access data directly from the underlying data storage layer, the Hadoop Distributed File System (HDFS).
Overview of Steps to Configure the FILENAME Statement and PROC HADOOP

1. Verify that all prerequisites have been satisfied.
   This step ensures that you understand your Hadoop environment. For more information, see “Prerequisites for the FILENAME Statement and PROC HADOOP” on page 6.

2. Determine whether you want to connect to the Hadoop server using Hadoop JAR files or an HTTP REST API.
   For more information, see “Making Hadoop JAR and Configuration Files Available to the SAS Client Machine” on page 6 and “Using WebHDFS or HttpFS” on page 6.

   Note: If you decide to connect to the Hadoop server using an HTTP REST API, you must make Hadoop configuration files available to the SAS client machine. The Hadoop JAR files are not required on the SAS client machine for the REST API.
3. If you use Apache Oozie, follow the configuration steps in “Using Apache Oozie” on page 8.

4. Run basic tests to confirm that your Hadoop connections are working.

For more information, see “Validating the FILENAME Statement and PROC HADOOP to Hadoop Connection” on page 10.

---

**Prerequisites for the FILENAME Statement and PROC HADOOP**

**Setting Up Your Environment for the FILENAME Statement and PROC HADOOP**

To ensure that your Hadoop environment and SAS software are ready for configuration:

1. Verify that you have set up your Hadoop environment correctly prior to installation of any SAS software.

   For more information, see Chapter 1, “Verifying Your Hadoop Environment,” on page 1.

2. Review the Hadoop distributions that are supported for the FILENAME statement and PROC HADOOP.

   For a list of the supported Hadoop distributions and versions, see SAS Viya: Deployment Guide.

3. Install SAS Viya by following the instructions in your software order email.

---

**Making Hadoop JAR and Configuration Files Available to the SAS Client Machine**

To submit the FILENAME statement or PROC HADOOP to a Hadoop server, a set of Hadoop JAR and configuration files must be available to the SAS client machine.

For SAS Viya, the Hadoop JAR and configuration files are made available when you configure the SAS Data Connector to Hadoop. For more information, see SAS Viya: Deployment Guide.

---

**Using WebHDFS or HttpFS**

WebHDFS is an HTTP REST API that supports the complete FileSystem interface for HDFS. MapR Hadoop distributions call this functionality HttpFS. WebHDFS and HttpFS essentially provide the same functionality.

Using WebHDFS or HttpFS removes the need for client-side JAR files for HDFS, but JAR files are still needed to submit MapReduce programs and Pig language programs.
Note: If you decide to connect to the Hadoop server using an HTTP REST API, you must make Hadoop configuration files available to the SAS client machine. The Hadoop JAR files are not required on the SAS client machine for the REST API. For more information, see “Making Hadoop JAR and Configuration Files Available to the SAS Client Machine” on page 6.

To use WebHDFS or HttpFS instead of the HDFS service:

1. Define the SAS environment variable SAS_HADOOP_RESTFUL 1. Here are three examples:
   ```
   set SAS_HADOOP_RESTFUL 1 /* DOS prompt */
   or
   -set SAS_HADOOP_RESTFUL 1 /* SAS command line */
   or
   export SAS_HADOOP_RESTFUL=1 /* UNIX */
   ```
   For more information, see “SAS_HADOOP_RESTFUL Environment Variable” on page 27.

2. Make sure the configuration files include the properties for the WebHDFS or HttpFS location. The configuration files include the `dfs.http.address` property or the `dfs.namenode.http-address` property. If the `dfs.http.address` property is not in the configuration file, the `dfs.namenode.http-address` property is used if it is in the file.

   Here is an example of configuration file properties for a WebHDFS location:

   ```
   <property>
   <name>dfs.http.address</name>
   <value>hwserver1.unx.xyz.com:50070</value>
   </property>
   ```
   or
   ```
   <property>
   <name>dfs.namenode.http-address</name>
   <value>hwserver1.unx.xyz.com:50070</value>
   </property>
   ```

   Here is an example of configuration file properties for an HttpFS location:

   ```
   <property>
   <name>dfs.http.address</name>
   <value>maprserver1.unx.xyz.com:14000</value>
   </property>
   ```

   For more information about the configuration files, see “Making Hadoop JAR and Configuration Files Available to the SAS Client Machine” on page 6.
Using Apache Knox Gateway Security

To use the FILENAME statement and PROC HADOOP with a Hadoop cluster that includes Apache Knox Gateway authentication:

• Connect to the Hadoop server through WebHDFS by defining the SAS_HADOOP_RESTFUL 1 SAS environment variable. Here is an example:
  
    options set=SAS_HADOOP_RESTFUL 1;

  For more information, see “SAS_HADOOP_RESTFUL Environment Variable” on page 27.

• Make sure the configuration files include the properties for the WebHDFS location. For more information, see “Using WebHDFS or HttpFS” on page 6.

• Set the SAS environment variable KNOX_GATEWAY_URL to the location of the Knox Gateway. Here is an example:
  
    options set=KNOX_GATEWAY_URL='https://server:port/gateway/default';

  For more information, see “KNOX_GATEWAY_URL Environment Variable” on page 23.

• Set up the SSL encryption protocol. For example, the SSLCALISTLOC= system option must be submitted to specify the location of the public certificate or certificates for trusted certificate authorities (CAs). For more information about the SSL encryption protocol and the SSLCALISTLOC= system option, see Encryption in SAS Viya: Data in Motion.

• Provide an authorized user ID and password in the FILENAME statement or PROC HADOOP statement to authenticate on the Apache Knox Gateway server. Here is an example:
  
    proc hadoop username='sasabc' password='sasabc' verbose;
    hdfs mkdir='/user/sasabc/new_directory';
    hdfs delete='/user/sasabc/temp2_directory';
    hdfs copytolocal='/user/sasabc/testdata.txt'
      out='/Users/sasabc/Hadoop/testdata.txt' overwrite;
    run;

Using Apache Oozie

Apache Oozie is a workflow scheduler system that manages Apache Hadoop jobs. Apache Oozie supports running MapReduce and Pig jobs by using WebHDFS or HttpFS.

Using Apache Oozie removes the need for client-side JAR files. To use Apache Oozie to submit MapReduce programs and Pig language code:

1. Define the SAS environment variable SAS_HADOOP_RESTFUL 1. Here are three examples:
  
    set SAS_HADOOP_RESTFUL 1 /* DOS prompt */

    or
-set SAS_HADOOP_RESTFUL 1 /* SAS command line */

or
export SAS_HADOOP_RESTFUL=1 /* UNIX */

For more information, see “SAS_HADOOP_RESTFUL Environment Variable” on page 27.

2. Create a directory that is accessible to the SAS client machine.
3. From the specific Hadoop cluster, copy these configuration files to the directory created in step 2.
   - core-site.xml
   - hdfs-site.xml
4. Make sure the hdfs-site.xml configuration file includes the properties for the WebHDFS location. The configuration file includes the dfs.http.address property or the dfs.namenode.http-address property. If the dfs.http.address property is not in the configuration file, the dfs.namenode.http-address property is used if it is in the file.
   Here is an example of configuration file properties for a WebHDFS location:

```
<property>
  <name>dfs.http.address</name>
  <value>server.yourcompany.com:50070</value>
</property>
```
   or

```
<property>
  <name>dfs.namenode.http-address</name>
  <value>server.yourcompany.com:50070</value>
</property>
```
5. Define the SAS environment variable named SAS_HADOOP_CONFIG_PATH. Set the environment variable to the directory path for the Hadoop cluster configuration files. For example, if the cluster configuration files are copied to the location /sasdata/cluster1/config, then the following syntax sets the environment variable appropriately. If the pathname contains spaces, enclose the pathname value in double quotation marks.

- set SAS_HADOOP_CONFIG_PATH "/sasdata/cluster1/config"

6. Create a single configuration file with properties that are specific to Oozie (for example, the Hadoop Oozie Server HTTP port, Hadoop NameNode, and Hadoop Job Tracker). Save the file to a directory that is accessible to the SAS client machine.
Here is an example of a single configuration file with properties that are specific to Oozie:

```
<configuration>
  <name>oozie_http_port</name>
  <value>server.yourcompany.com:11000</value>
  <name>fs.default.name</name>
  <value>server.yourcompany.com:8020</value>
  <name>mapred.job.tracker</name>
  <value>server.yourcompany.com:8032</value>
  <name>dfs.http.address</name>
```
Validating the FILENAME Statement and PROC HADOOP to Hadoop Connection

Validating the FILENAME Statement

This FILENAME example writes the file `myfile` to the directory `testing`.

```plaintext
filename out hadoop "\user\testing\myfile"
   user="sasabc" pass="abcpass";

data _null_
   file out;
   put "here is a line in myfile";
run;
```

Validating PROC HADOOP

This PROC HADOOP example submits HDFS commands to a Hadoop server. The statements create a directory, delete a directory, and copy a file from HDFS to a local output location.

```plaintext
proc hadoop username='sasabc' password='sasabc' verbose;
   hdfs mkdir='/user/sasabc/new_directory';
   hdfs delete='/user/sasabc/temp2_directory';
   hdfs copytolocal='/user/sasabc/testdata.txt'
      out="/Users/sasabc/Hadoop/testdata.txt" overwrite;
run;
```

Documentation for Using the FILENAME Statement and PROC HADOOP

The documentation can be found in these documents:
• “FILENAME Statement, Hadoop Access Method” in SAS Viya Statements: Reference

Chapter 4
Configuring SAS/ACCESS for Hadoop

Overview of Steps to Configure SAS/ACCESS Interface to Hadoop

1. Verify that all prerequisites have been satisfied.
   This step ensures that you understand your Hadoop environment. For more information, see “Prerequisites for SAS/ACCESS Interface to Hadoop” on page 14.

2. Configure the SAS Data Connector to Hadoop.
   This step involves obtaining Hadoop JAR and configuration files, which are necessary to use SAS/ACCESS with a Hadoop server. For more information, see...
“Making Hadoop JAR and Configuration Files Available to the SAS Client Machine” on page 15.

*Note:* Although it is not required for basic SAS/ACCESS functionality, if you have licensed SAS In-Database Technologies for Hadoop on SAS Viya, you should deploy the SAS Embedded Process for Hadoop. For more information, see *SAS Viya: Deployment Guide*.

3. Review the following sections for additional configuration information:
   - HiveServer2
     “Working with Hive” on page 18
   - WebHDFS or HttpFS
     “Using WebHDFS or HttpFS” on page 17

4. Review security and user access.
   For more information, see “Security and User Access to Hadoop” on page 15.

5. Run basic tests to confirm that your Hadoop connections are working.
   For more information, see “Validating Your SAS/ACCESS to Hadoop Connection” on page 20.

---

**Prerequisites for SAS/ACCESS Interface to Hadoop**

**Setting Up Your Environment for SAS/ACCESS Interface to Hadoop**

To ensure that your Hadoop environment and SAS software are ready for configuration:

1. Verify that you have set up your Hadoop environment correctly prior to installation of any SAS software.
   For more information, see Chapter 1, “Verifying Your Hadoop Environment,” on page 1.

2. Review the supported Hadoop distributions.
   For a list of supported Hadoop distributions and versions, see *SAS Viya: Deployment Guide*.

   *Note:* SAS takes advantage of the advanced Hadoop types, including DATE, TIMESTAMP, and VARCHAR when the version of Hive is .12 or later.

   *Note:* SAS/ACCESS can be configured for Kerberos ticket cache-based logon authentication by using Kerberos 5 Version 1.9 and by running HiveServer2.

3. Install SAS/ACCESS Interface to Hadoop by following the instructions in your software order email.
Making Hadoop JAR and Configuration Files Available to the SAS Client Machine

**Making Hadoop JAR and Configuration Files Available to the SAS Client Machine**

To use SAS/ACCESS with a Hadoop server, a set of Hadoop JAR and configuration files must be available to the SAS client machine.

For SAS Viya, the Hadoop JAR and configuration files are made available when you configure the SAS Data Connector to Hadoop. For more information, see *SAS Viya: Deployment Guide*.

**Additional Configuration for MapR**

The following requirements are needed for Hadoop systems based on MapR:

- Set the `java.library.path` property to the directory that contains the 64-bit MapRClient shareable library. Set the `java.security.auth.login.config` property to the `mapr.login.conf` file, which is normally installed in the `MAPR_HOME/conf` directory.

  *Note:* The MapR 64-bit library must be selected. The MapR 32-bit library produces undesirable results.

  *Note:* As reported by MapR case #00038839, when using MapR 5.0 or later, setting `-Djava.library.path` can result in various class errors. The workaround is to remove the `-Djava.library.path` from Java JRE options. This workaround might allow the connection to work, causing MapR 5.x to extract its native libraries from the JAR file to the `/tmp` directory on a per-user basis. MapR is working on a solution to this issue.

- MapR requires this JRE option for a Kerberos connection:

  `-Dhadoop.login=kerberos`

  For more information, see *Configuring Hive on a Secure Cluster: Using JDBC with Kerberos*.

**Security and User Access to Hadoop**

**Kerberos Security**

SAS/ACCESS can be configured for a Kerberos ticket cache-based logon authentication by using MIT Kerberos 5 Version 1.9 and by running HiveServer2.

- If you are using Advanced Encryption Standard (AES) encryption with Kerberos, you must manually add the Java Cryptography Extension `local_policy.jar` file in every place that JAVA Home resides on the cluster. If you are outside the United States, you must also manually add the `US_export_policy.jar` file. The addition of these files is governed by the United States import control restrictions.
If you are using the Oracle JRE or the IBM JRE, the appropriate JAR file must also replace the existing local_policy.jar and US_export_policy.jar files in your JRE location. This location is typically the `Java_Home/jre/lib/security/` directory. You can obtain the appropriate file from the Oracle or IBM website.

It is recommended to back up the existing local_policy.jar and US_export_policy.jar files first in case they need to be restored.

If you are using the OpenJDK, the files do not need to be replaced.

**Apache Knox Gateway Security**

To use the SAS/ACCESS Interface to Hadoop with a Hadoop cluster that includes Apache Knox Gateway authentication, you must complete these configuration steps:

- Connect to the Hadoop server through WebHDFS by defining the SAS_HADOOP_RESTFUL 1 SAS environment variable. Here is an example:
  
  ```
  options set=SAS_HADOOP_RESTFUL 1;
  ```

  For more information, see “SAS_HADOOP_RESTFUL Environment Variable” on page 27.

- Make sure the configuration files include the properties for the WebHDFS location. For more information, see “Using WebHDFS or HttpFS” on page 17.

- Set the SAS environment variable KNOX_GATEWAY_URL to the location of the Knox Gateway. Here is an example:
  
  ```
  options set=KNOX_GATEWAY_URL='https://server:port/gateway/default';
  ```

  For more information, see “KNOX_GATEWAY_URL Environment Variable” on page 23.

- Set up the SSL encryption protocol. For example, the SSLCALISTLOC= system option must be submitted to specify the location of the public certificate or certificates for trusted certificate authorities (CAs). For more information about the SSL encryption protocol and the SSLCALISTLOC= system option, see *Encryption in SAS Viya: Data in Motion*.

- Use the URI= option in the LIBNAME statement option to connect to Knox. The URI= option is required to fully qualify the JDBC connection string to a Hive cluster that is behind a Knox gateway. Here is an example:

  ```
  uri='jdbc:hive2://server:port/default;ssl=true;transportMode=http;httpPath=gateway/default/hive'
  ```

  For more information about the JDBC Knox connection options, see Apache Knox.

**JDBC Read Security**

SAS/ACCESS can access Hadoop data through a JDBC connection to a HiveServer2 service. Depending on what release of Hive you have, Hive might not implement Read security. A successful connection from SAS can allow Read access to all data accessible to the Hive service. HiveServer2 can be secured with Kerberos. SAS/ACCESS supports Kerberos 5 Version 1.9 or later.
**HDFS Write Security**

SAS/ACCESS creates and appends to Hive tables by using the HDFS service. HDFS can be unsecured, user and password secured, or Kerberos secured. Your HDFS connection needs Write access to the HDFS `/tmp` directory. After data is written to `/tmp`, a Hive LOAD command is issued on your JDBC connection to associate the data with a Hive table. Therefore, the JDBC Hive session also needs Write access to `/tmp`.

**HDFS Permission Requirements for Optimized Reads**

To optimize big data reads, SAS/ACCESS creates a temporary table in the HDFS `/tmp` directory. This action requires that the SAS JDBC connection have Write access to `/tmp`. The temporary table is read using HDFS, so the SAS HDFS connection needs Read access to the temporary table that is written to `/tmp`.

---

**Using WebHDFS or HttpFS**

WebHDFS is an HTTP REST API that supports the complete FileSystem interface for HDFS. MapR Hadoop distributions call this functionality HttpFS. WebHDFS and HttpFS essentially provide the same functionality.

To use WebHDFS or HttpFS instead of the HDFS service, complete these steps. Although using WebHDFS or HttpFS removes the need for client-side JAR files for HDFS, JAR files are still needed to submit MapReduce programs and Pig language programs.

1. Define the SAS environment variable SAS_HADOOP_RESTFUL 1. Here are three examples:

   /* SAS command line */
   set SAS_HADOOP_RESTFUL 1

   /* DOS prompt */
   -set SAS_HADOOP_RESTFUL 1

   /* UNIX */
   export SAS_HADOOP_RESTFUL=1

   For more information, see “SAS_HADOOP_RESTFUL Environment Variable” on page 27.

2. Make sure the configuration files include the properties for the WebHDFS or HttpFS location. If the `dfs.http.address` property is not in the configuration file, the `dfs.namenode.http-address` property is used if it is in the file.

   Here is an example of configuration file properties for a WebHDFS location:

   ```
   <property>
   <name>dfs.http.address</name>
   <value>hwserver1.unx.xyz.com:50070</value>
   </property>
   ```

   ```
   <property>
   <name>dfs.namenode.http-address</name>
   <value>hwserver1.unx.xyz.com:50070</value>
   </property>
   ```
Here is an example of configuration file properties for an HttpFS location:

```
<property>
<name>dfs.http.address</name>
<value>maprserver1.unx.xyz.com:14000</value>
</property>

---- or ----

<property>
<name>dfs.namenode.http-address</name>
<value>maprserver1.unx.xyz.com:14000</value>
</property>
```

For more information about the configuration files, see “Making Hadoop JAR and Configuration Files Available to the SAS Client Machine” on page 15.

---

**Working with Hive**

**Starting with Hive**

If you do not currently run Hive on your Hadoop server, then your Hadoop data likely resides in HDFS files initially invisible to Hive. To make HDFS files (or other formats) visible to Hive, a Hive CREATE TABLE is issued.

The following example of a simple table demonstrates how to access HDFS files using the Beeline interface with a JDBC connection string. Informational lines returned by the Beeline interface have been removed for brevity.

```
0: jdbc:hive2://cdh58d1hive:10000/default> !connect jdbc:hive2://cdh58d1hive:10000/default
Connecting to jdbc:hive2://cdh58d1hive:10000/default
Enter username for jdbc:hive2://cdh58d1hive:10000/default: hadoop
Enter password for jdbc:hive2://cdh58d1hive:10000/default: *******
Connected to: Apache Hive (version 1.1.0-cdh5.8.0)

1: jdbc:hive2://cdh58d1hive:10000/default> create table test (c char(10) );

INFO : OK

1: jdbc:hive2://cdh58d1hive:10000/default> insert into table test values ('test');

INFO : OK
No rows affected (16.668 seconds)

1: jdbc:hive2://cdh58d1hive:10000/default> select * from test;

+-------------+--+
|   test.c    |
+-------------+--+
| test        |
```
To access this table from SAS, run this example code:

```sas
libname hdplib hadoop server=hadoop_cluster user=hadoop_usr
    password=hadoop_usr_pwd;
data work.test;
    set hdplib.test;
    put _all_
    run;

proc sql;
    select c from hdplib.test;
quit;
```

This is a complete but intentionally simple scenario intended for new Hive users. To explore Hive in detail, consult Hadoop and Hive documentation such as Apache Hive. For more information about how SAS/ACCESS interacts with Hive, see SAS/ACCESS for Relational Databases: Reference.

**Running the Hive Service on Your Hadoop Server**

SAS/ACCESS reads Hadoop data via a JDBC connection to a HiveServer2 service. As a best practice, launch the service as a daemon that kicks off on system restarts. This launch ensures consistent service.

This example starts a HiveServer2 service at an operating system prompt:

```bash
$ export HIVE_PORT=10000
$ HIVE_HOME/bin/hive --service hiveserver2
```

*Note:* For Hive operations such as submitting HiveQL, the Hadoop engine requires access to the Hive service that runs on the Hadoop cluster, often port 10000. For HDFS operations, such as writing data to Hive tables, the Hadoop engine requires access to the HDFS service that runs on the Hadoop cluster, often port 8020. If the Hadoop engine cannot access the HDFS service, its full functionality is not available.

**Writing Data to Hive: HDFS /tmp and the “Sticky Bit”**

SAS/ACCESS assumes that HDFS /tmp exists, and writes data there. After data is written, SAS/ACCESS issues a LOAD command to move the data to the Hive warehouse. If the “sticky bit” is set on HDFS /tmp, the LOAD command can fail. One option to resolve this LOAD failure is to disable the “sticky bit” on HDFS /tmp. If the “sticky bit” cannot be disabled, SAS data can be written to an alternate location specified by the HDFS_TEMPDIR= option.

In this example of a Hadoop file system command, the “sticky bit” is set for HDFS/tmp. It is denoted by the 't' attribute.

```bash
$ hadoop fs -ls /
  drwxrwxrwt - hdfs hdfs 0 2016-01-21 13:25 /tmp
drwxr-xr-x - hdfs supergroup 0 2016-01-21 11:46 /user
```
Validating Your SAS/ACCESS to Hadoop Connection

SAS code connects to HiveServer2 either with a libref or a PROC SQL CONNECT TO statement. The libref writes information upon a successful connection, whereas PROC SQL is silent on a successful connection.

In these examples, Hive is listening on default port 10000 on Hadoop node **hadoop01**.

**Sample libref connection to HiveServer2 (default):**

```sas
libname hdplib hadoop server=hadoop01 user=hadoop_usr password=hadoop_usr_pwd;
```

NOTE: Libref HDPLIB was successfully assigned as follows:

- Engine: HADOOP
- Physical Name: jdbc:hive2://hadoop01:10000/default

**Sample PROC SQL connection:**

```sas
proc sql;
connect to hadoop (server=hadoop01 user=hadoop_usr password=hadoop_usr_pwd);
```

A failure to connect can have different causes. Error messages can help diagnose the issue.

In this sample failure, Hive is not active on port 10000 on Hadoop node **hadoop01**:

```sas
libname hdplib hadoop server=hadoop01 port=10000 user=hadoop_usr password=hadoop_usr_pwd;
```

**ERROR:** java.sql.SQLException: Could not establish connection to hadoop01:10000/default:

```
    java.net.ConnectException: Connection refused: connect
    ERROR: Unable to connect to server or to call the Java Drivermanager.
    ERROR: Error trying to establish connection.
    ERROR: Error in the LIBNAME statement.
```

In this sample failure, the hive-metastore JAR file is missing from **SAS_HADOOP_JAR_PATH**:

```sas
libname hdplib hadoop server=hadoop01 port=10000 user=hadoop_usr password=hadoop_usr_pwd;
```

**ERROR:** java.lang.NoClassDefFoundError: org/apache/hadoop/hive/metastore/api/MetaException

```
    ERROR: Unable to connect to server or to call the Java Drivermanager.
    ERROR: Error trying to establish connection.
    ERROR: Error in the LIBNAME statement.
```
Documentation for Using SAS/ACCESS Interface to Hadoop

The documentation can be found in “SAS/ACCESS Interface to Hadoop” in SAS/ACCESS for Relational Databases: Reference.
Appendix 1
SAS Environment Variables for Hadoop

Dictionary

KNOX_GATEWAY_URL Environment Variable
Sets the location of the Apache Knox Gateway.

Valid in: SAS configuration file, SAS invocation, OPTIONS statement, SAS System Options window

Used by: FILENAME statement Hadoop access method, HADOOP procedure, SAS/ACCESS Interface to Hadoop

See: “Using Apache Knox Gateway Security” on page 8

Syntax

KNOX_GATEWAY_URL URL

Required Argument

URL specifies the HTTPS URL for the Knox Gateway website. The URL is site specific.
The format of the Knox gateway URL is as follows:
https://gateway-host:gateway-port/gateway-pathname/cluster-name

For example, the following OPTIONS statement syntax sets the environment variable:
options set=KNOX_GATEWAY_URL "https://server:port/gateway/default";
Details

The Apache Knox Gateway is a REST API gateway for interacting with Hadoop clusters. The Apache Knox Gateway runs as a reverse proxy, which provides a single point of authentication and access for Apache Hadoop services in one or more Hadoop clusters.

How you define the SAS environment variables depends on your operating environment. For most operating environments, you can define the environment variables either locally (for use only in your SAS session) or globally. For example, you can define the SAS environment variables with the SET system option in a SAS configuration file, at SAS invocation, with the OPTIONS statement, or in the SAS System Options window. In addition, you can use your operating system to define the environment variables.

The following table includes examples of defining the KNOX_GATEWAY_URL environment variable.

<table>
<thead>
<tr>
<th>Operating Environment</th>
<th>Method</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX *</td>
<td>SAS configuration file</td>
<td>-set KNOX_GATEWAY_URL &quot;<a href="https://server:port/gateway/default">https://server:port/gateway/default</a>&quot;</td>
</tr>
<tr>
<td></td>
<td>SAS invocation</td>
<td>-set KNOX_GATEWAY_URL &quot;<a href="https://server:port/gateway/default">https://server:port/gateway/default</a>&quot;</td>
</tr>
<tr>
<td></td>
<td>OPTIONS statement</td>
<td>options <a href="https://server:port/gateway/default">https://server:port/gateway/default</a>;</td>
</tr>
</tbody>
</table>

* In the UNIX operating environment, the SAS environment variable name must be in uppercase characters and the value must be the full pathname of the directory. That is, the name of the directory must begin with a slash.

SAS_HADOOP_CONFIG_PATH Environment Variable

Sets the location of the Hadoop cluster configuration files.

Valid in: SAS configuration file, SAS invocation, OPTIONS statement, SAS System Options window

Used by: FILENAME statement Hadoop access method, HADOOP procedure, SAS/ACCESS Interface to Hadoop

Requirement: The SAS_HADOOP_CONFIG_PATH environment variable must be set regardless of whether you are using JAR files or WebHDFS or HttpFS.

Note: This environment variable is automatically set if you accept the default configuration values in SAS Deployment Manager when you configure SAS/ACCESS Interface to Hadoop.

Syntax

SAS_HADOOP_CONFIG_PATH pathname
Required Argument

\textit{pathname}

specifies the directory path for the Hadoop cluster configuration files. If the
pathname contains spaces, enclose the pathname value in double quotation marks.

For example, if the cluster configuration files are copied from the Hadoop cluster to
the location /sasdata/cluster1/conf, then the following OPTIONS statement
syntax sets the environment variable appropriately.

\begin{verbatim}
options set=SAS_HADOOP_CONFIG_PATH "/sasdata/cluster1/conf";
\end{verbatim}

Details

Your Hadoop administrator configures the Hadoop cluster that you use. The
administrator defines defaults for system parameters such as block size and replication
factor that affect the Read and Write performance of your system. In addition, Hadoop
cluster configuration files contain information such as the host name of the computer
that hosts the Hadoop cluster and the TCP port.

How you define the SAS environment variables depends on your operating environment.
For most operating environments, you can define the environment variables either
locally (for use only in your SAS session) or globally. For example, you can define the
SAS environment variables with the SET system option in a SAS configuration file, at
SAS invocation, with the OPTIONS statement, or in the SAS System Options window.
In addition, you can use your operating system to define the environment variables.

\textit{Note:} Only one SAS\_HADOOP\_CONFIG\_PATH path is used per Hadoop cluster. To
see the path, enter the following command:

\begin{verbatim}
%put %sysget(SAS_HADOOP_CONFIG_PATH);
\end{verbatim}

The following table includes examples of defining the SAS\_HADOOP\_CONFIG\_PATH
environment variable.

\begin{table}[h]
\centering
\caption{Defining the SAS\_HADOOP\_CONFIG\_PATH Environment Variable}
\begin{tabular}{|c|c|l|}
\hline
\textbf{Operating Environment} & \textbf{Method} & \textbf{Example} \\
\hline
UNIX & SAS configuration file & \texttt{-set SAS\_HADOOP\_CONFIG\_PATH "/sasdata/cluster1/conf"} \\
& SAS invocation & \texttt{-set SAS\_HADOOP\_CONFIG\_PATH "/sasdata/cluster1/conf"} \\
& OPTIONS statement & \texttt{options set=SAS\_HADOOP\_CONFIG\_PATH="/sasdata/cluster1/conf";} \\
\hline
\end{tabular}
\end{table}

\* In the UNIX operating environment, the SAS environment variable name must be in uppercase characters and the value must be the
full pathname of the directory. That is, the name of the directory must begin with a slash.

\section*{SAS\_HADOOP\_JAR\_PATH Environment Variable}

Sets the location of the Hadoop JAR files.
Valid in: SAS configuration file, SAS invocation, OPTIONS statement, SAS System Options window

Used by: FILENAME statement Hadoop access method, HADOOP procedure, SAS/ACCESS Interface to Hadoop

Note: This environment variable is automatically set if you accept the default configuration values in SAS Deployment Manager when you configure SAS/ACCESS Interface to Hadoop.

Tip: If SAS_HADOOP_RESTFUL is set to 1 and you are using the FILENAME Statement Hadoop access method, you do not need to set the SAS_HADOOP_JAR_PATH environment variable.

Syntax

SAS_HADOOP_JAR_PATH pathname(s)

Required Argument

pathname(s)

specifies the directory path for the Hadoop JAR files. If the pathname contains spaces, enclose the pathname value in double quotation marks. To specify multiple pathnames, concatenate pathnames by separating them with a colon (:) in a UNIX environment.

For example, if the JAR files are copied to the location /third_party/Hadoop/jars/lib, then the following OPTIONS statement syntax sets the environment variable appropriately.

```sas
options set=SAS_HADOOP_JAR_PATH="/third_party/Hadoop/jars/lib";
```

Details

Unless you are using WebHDFS or HttpFS, SAS components that interface with Hadoop require that a set of Hadoop JAR files be available to the SAS client machine. The SAS environment variable named SAS_HADOOP_JAR_PATH must be defined to set the location of the Hadoop JAR files.

How you define the SAS environment variables depends on your operating environment. For most operating environments, you can define the environment variables either locally (for use only in your SAS session) or globally. For example, you can define the SAS environment variables with the SET system option in a SAS configuration file, at SAS invocation, with the OPTIONS statement, or in the SAS System Options window. In addition, you can use your operating system to define the environment variables.

Note: Only one SAS_HADOOP_JAR_PATH path is used. To see the path, enter the following command:

```sas
%put %sysget(SAS_HADOOP_JAR_PATH);
```

The following table includes examples of defining the SAS_HADOOP_JAR_PATH environment variable.
Table A1.3  Defining the SAS_HADOOP_JAR_PATH Environment Variable

<table>
<thead>
<tr>
<th>Operating Environment</th>
<th>Method</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX *</td>
<td>SAS configuration file</td>
<td>-set SAS_HADOOP_JAR_PATH &quot;/third_party/Hadoop/jars/lib&quot;</td>
</tr>
<tr>
<td></td>
<td>SAS invocation</td>
<td>-set SAS_HADOOP_JAR_PATH &quot;/third_party/Hadoop/jars/lib&quot;</td>
</tr>
<tr>
<td></td>
<td>OPTIONS statement</td>
<td>options set=SAS_HADOOP_JAR_PATH=&quot;/third_party/Hadoop/jars/lib&quot;;</td>
</tr>
</tbody>
</table>

* In the UNIX operating environment, the SAS environment variable name must be in uppercase characters and the value must be the full pathname of the directory. That is, the name of the directory must begin with a slash.

Note: A SAS_HADOOP_JAR_PATH directory must not have multiple versions of a Hadoop JAR file. Multiple versions of a Hadoop JAR file can cause unpredictable behavior when SAS runs.

Note: For SAS/ACCESS Interface to Hadoop to operate properly, your SAS_HADOOP_JAR_PATH directory must not contain any Thrift JAR files such as libthrift*.jar.

SAS_HADOOP_RESTFUL Environment Variable

Determines whether to connect to the Hadoop server through JAR files, HttpFS, or WebHDFS.

Valid in: SAS configuration file, SAS invocation, OPTIONS statement, SAS System Options window

Used by: FILENAME statement Hadoop access method, HADOOP procedure, SAS/ACCESS Interface to Hadoop

Default: 0, which connects to the Hadoop server with JAR files

Syntax

SAS_HADOOP_RESTFUL 0 | 1

Required Arguments

0

specifies to connect to the Hadoop server by using Hadoop client side JAR files. This is the default setting.

1

specifies to connect to the Hadoop server by using the WebHDFS or HttpFS REST API.

Requirement The Hadoop configuration file must include the properties of the WebHDFS location or the HttpFS location.
Details

WebHDFS is an HTTP REST API that supports the complete FileSystem interface for HDFS. MapR Hadoop distributions call this functionality HttpFS. WebHDFS and HttpFS essentially provide the same functionality.

How you define the SAS environment variables depends on your operating environment. For most operating environments, you can define the environment variables either locally (for use only in your SAS session) or globally. For example, you can define the SAS environment variables with the SET system option in a SAS configuration file, at SAS invocation, with the OPTIONS statement, or in the SAS System Options window. In addition, you can use your operating system to define the environment variables.

The following table includes examples of defining the SAS_HADOOP_RESTFUL environment variable.

<table>
<thead>
<tr>
<th>Method</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS configuration file</td>
<td>\texttt{-set SAS_HADOOP_RESTFUL 1}</td>
</tr>
<tr>
<td>SAS invocation</td>
<td>\texttt{-set SAS_HADOOP_RESTFUL 1}</td>
</tr>
<tr>
<td>OPTIONS statement</td>
<td>\texttt{options set=SAS_HADOOP_RESTFUL 1;}</td>
</tr>
</tbody>
</table>
Recommended Reading

• *Encryption in SAS Viya: Data in Motion*
• SAS and Hadoop Technology: Overview
• *SAS/ACCESS for Relational Databases: Reference*
• *SAS Viya: Deployment Guide*
• *SAS Viya Statements: Reference*
• *SAS Viya Visual Data Management and Utility Procedures Guide*

For a complete list of SAS publications, go to sas.com/store/books. If you have questions about which titles you need, please contact a SAS Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-0025
Fax: 1-919-677-4444
Email: sasbook@sas.com
Web address: sas.com/store/books
# Index

<table>
<thead>
<tr>
<th>Letter</th>
<th>Topic</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Apache Knox Gateway security</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FILENAME statement</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>PROC HADOOP</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>SAS/ACCESS interface to Hadoop</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Apache Oozie</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PROC HADOOP</td>
<td>8</td>
</tr>
<tr>
<td>C</td>
<td>configuration files</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FILENAME statement</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>PROC HADOOP</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>SAS/ACCESS interface to Hadoop</td>
<td>15</td>
</tr>
<tr>
<td>D</td>
<td>documentation for using</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FILENAME statement</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>PROC HADOOP</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>SAS/ACCESS interface to Hadoop</td>
<td>21</td>
</tr>
<tr>
<td>E</td>
<td>environment variable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KNOX_GATEWAY_URL</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>SAS_HADOOP_CONFIG_PATH</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>SAS_HADOOP_JAR_PATH</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>SAS_HADOOP_RESTFUL</td>
<td>27</td>
</tr>
<tr>
<td>F</td>
<td>FILENAME statement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>configuration files</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>documentation for using</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Hadoop distributions</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Hadoop JAR files</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>HttpFS</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>prerequisites</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>validating Hadoop connection</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>WebHDFS</td>
<td>6</td>
</tr>
<tr>
<td>H</td>
<td>Hadoop connection</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FILENAME statement</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>PROC HADOOP</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>SAS/ACCESS interface to Hadoop</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Hadoop distributions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FILENAME statement</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>PROC HADOOP</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>SAS/ACCESS interface to Hadoop</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Hadoop JAR files</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FILENAME statement</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>PROC HADOOP</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>SAS/ACCESS interface to Hadoop</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>HiveServer2, SAS/ACCESS interface to Hadoop</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>HttpFS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FILENAME statement</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>PROC HADOOP</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>SAS/ACCESS interface to Hadoop</td>
<td>17</td>
</tr>
<tr>
<td>J</td>
<td>JAR files</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FILENAME statement</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>PROC HADOOP</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>SAS/ACCESS interface to Hadoop</td>
<td>15</td>
</tr>
<tr>
<td>K</td>
<td>Kerberos security</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SAS/ACCESS interface to Hadoop</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Knox security</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FILENAME statement</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>PROC HADOOP</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>SAS/ACCESS interface to Hadoop</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>KNOX_GATEWAY_URL</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>environment variable</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>prerequisites</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FILENAME statement</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>PROC HADOOP</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>SAS/ACCESS interface to Hadoop</td>
<td>14</td>
</tr>
<tr>
<td>Index</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>PROC HADOOP</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Apache Oozie</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>configuration files</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>documentation for using</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Hadoop distributions</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Hadoop JAR files</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>HttpFS</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>prerequisites</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>validating Hadoop connection</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>WebHDFS</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>HttpFS</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>prerequisites</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>security</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>validating Hadoop connection</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>WebHDFS</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>security</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>SAS/ACCESS interface to Hadoop</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>system requirements</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAS_HADOOP_CONFIG_PATH environment variable</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>SAS_HADOOP_JAR_PATH environment variable</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>SAS_HADOOP_RESTFUL environment variable</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>SAS/ACCESS interface to Hadoop configuration files</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Hadoop distributions</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Hadoop JAR files</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>HiveServer2</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>validating Hadoop connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILENAME statement</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>PROC HADOOP</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>SAS/ACCESS interface to Hadoop</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WebHDFS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILENAME statement</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>PROC HADOOP</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>SAS/ACCESS interface to Hadoop</td>
<td>17</td>
<td></td>
</tr>
</tbody>
</table>
Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore for additional books and resources.