# Contents

*What’s New in SAS 9.4 Hadoop Configuration Guide for Base SAS and SAS/ACCESS*

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Verifying Your Hadoop Environment</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Pre-Installation Checklist for SAS Software That Interfaces with Hadoop</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Base SAS and SAS/ACCESS Software with Hadoop</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Configuration Information for Other SAS Software</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Configuring FILENAME Statement Hadoop Access Method and PROC HADOOP</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Overview of Steps to Configure PROC HADOOP and the FILENAME Statement</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Prerequisites for the PROC HADOOP and the FILENAME Statement</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Making Hadoop JAR and Configuration Files Available to the SAS Client Machine</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Using WebHDFS or HttpFS</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Using Apache Knox Gateway Security</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Using Apache Oozie</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Validating PROC HADOOP and the FILENAME Statement to Hadoop Connection</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Documentation for Using PROC HADOOP and the FILENAME Statement</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>Configuring SAS/ACCESS for Hadoop</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Overview of Steps to Configure SAS/ACCESS Interface to Hadoop</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Prerequisites for SAS/ACCESS Interface to Hadoop</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Making Hadoop JAR and Configuration Files Available to the SAS Client Machine</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Additional Configuration for All Hadoop Distributions</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Configuring SAS/ACCESS Interface to Impala</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Configuring PROC SQOOP</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Security and User Access to Hadoop</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Using WebHDFS or HttpFS</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Working with Hive</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Validating Your SAS/ACCESS to Hadoop Connection</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Documentation for Using SAS/ACCESS Interface to Hadoop</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>Configuring SPD Engine</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Overview of Steps to Configure the SPD Engine</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Prerequisites for the SPD Engine</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Making Hadoop JAR and Configuration Files Available to the SAS Client Machine</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Additional Requirements for MapR Systems</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Additional Configuration for All Hadoop Distributions</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Kerberos Security</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Validating the SPD Engine to Hadoop Connection</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>Documentation for Using SPD Engine to Hadoop</td>
<td>30</td>
</tr>
<tr>
<td>1</td>
<td>Using the SAS Deployment Manager to Obtain Hadoop JAR and Configuration Files</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Information and Credentials Required to Configure Hadoop</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Using SAS Deployment Manager</td>
<td>31</td>
</tr>
</tbody>
</table>
Using SAS Deployment Manager to Obtain the Hadoop JAR and
Configuration Files ............................................................... 32
Supporting Multiple Hadoop Versions and Upgrading Hadoop Version .... 53
When to Collect New JAR and Configuration Files ......................... 53

Appendix 2 • Using the Hadoop Tracer Script to Obtain the Hadoop JAR
and Configuration Files .......................................................... 55
Overview of the Hadoop Tracer Script ........................................ 55
Prerequisites for Using the Hadoop Tracer Script .......................... 56
Obtaining the Hadoop Tracer Script ......................................... 56
Running the Hadoop Tracer Script .......................................... 56
Setting Up the JAR and Configuration File Directories on the SAS Client Machine 57
Additional Configuration for IBM BigInsights ............................ 59
Additional Configuration for MapR ......................................... 60

Appendix 3 • SAS Environment Variables for Hadoop ......................... 61
Dictionary .............................................................................. 61

Recommended Reading .......................................................... 69
Index ..................................................................................... 71
What’s New in SAS 9.4
Hadoop Configuration Guide
for Base SAS and
SAS/ACCESS

Overview

In SAS Viya 3.3, the information from *SAS Viya and SAS/ACCESS: Hadoop Configuration Guide* has been merged into this book. *SAS Viya and SAS/ACCESS: Hadoop Configuration Guide* is no longer available.

In SAS 9.4M5, the following new features and enhancements were added:

- SAS Deployment Manager can now be run if the cluster manager is Transport Layer Security (TLS) enabled. SAS Deployment Manager validates if the Certificate of Authority (CA) exists for the host that you are trying to access. If the CA does not exist, a warning message is issued and asks to run another SAS Deployment Manager task to add the CA.
  
  *Note:* All discussion of TLS is also applicable to the predecessor protocol, Secure Sockets Layer (SSL).

- You now have the option of manually collecting the JAR and configuration files with the hadooptracer.py script while using SAS Deployment Manager.

- Three new options have been added to the hadooptracer.py script:
  
  - `winplatform` collects the correct files when running the Hadoop tracer script from a Windows client.
  - `debug` produces version and debug logging information.
  - `postprocess` removes version numbers from configuration files that are collected.

In SAS 9.4M4, the following new features and enhancements were added:

- SAS Deployment Manager is now available for Base SAS and the SPD Engine as a means to obtain JAR and configuration files.

- SAS Deployment Manager has been enhanced to be more efficient in gathering the JAR and configuration files needed to run SAS software.

- The FILENAME statement Hadoop access method, HADOOP procedure, and SAS/ACCESS Interface to Hadoop support Apache Knox Gateway authentication.

---

Enhancements to SAS Deployment Manager

In SAS 9.4M5, SAS Deployment Manager has been enhanced in the following areas:
• SAS Deployment Manager now supports a cluster manager that is enabled with Transport Layer Security (TLS). SAS Deployment Manager validates if the Certificate of Authority (CA) exists for the host that you are trying to access. If the CA does not exist, a warning message is issued and asks to run another SAS Deployment Manager task to add the CA.

Note: All discussion of TLS is also applicable to the predecessor protocol, Secure Sockets Layer (SSL).

• You now have the option of manually collecting the JAR and configuration files with the hadooptracer.py script while using SAS Deployment Manager.

• Three new options have been added to the hadooptracer.py script:
  • winplatform collects the correct files when running the Hadoop tracer script from a Windows client.
  • debug produces version and debug logging information.
  • postprocess removes version numbers from configuration files that are collected.

• SAS Deployment Manager now validates whether the following configuration files are pulled from the cluster and writes a message to the sashadoopconfig.log file: core-site.xml, hdfs-site.xml, yarn-site.xml, mapred-site.xml, and hive-site.xml.

• For MapR distributions that use Pig, you no longer need to set Filter JAR Files By to No filter to pull the correct JAR files.

In SAS 9.4M4, SAS Deployment Manager has been enhanced in the following areas:

• If your SAS software uses Impala or Oozie services, SAS Deployment Manager generates an inventory.json file.

• You can choose to provide either a private key file or the user ID and password as credentials for the UNIX user account that has SSH access to the machine that is hosting HiveServer2.

• You can filter the JAR files by obtaining the most recent version of the JAR files.

• You can specify a non-default Hive user name.

• You can specify a non-default port for the Hive service.

---

**Apache Knox Gateway**

In SAS 9.4M4, the FILENAME statement Hadoop access method, HADOOP procedure, and SAS/ACCESS Interface to Hadoop can be configured for Apache Knox Gateway authentication:

• For SAS/ACCESS Interface to Hadoop, see “Apache Knox Gateway Security” on page 19.

• For the FILENAME statement and PROC HADOOP, see “Using Apache Knox Gateway Security” on page 8.
Chapter 1
Verifying Your Hadoop Environment

Pre-Installation Checklist for SAS Software That Interfaces with Hadoop

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

Before you install SAS software that interfaces with Hadoop, it is recommended that you verify your Hadoop environment by using the following 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

Base SAS 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:

- **Base SAS components**
  - FILENAME Statement Hadoop Access Method
    enables Base SAS users to use Hadoop to read from or write to a file from HDFS.
  - HADOOP procedure
    enables Base SAS users to submit HDFS commands, Pig language code, and MapReduce programs against Hadoop data. PROC HADOOP interfaces with the Hadoop JobTracker. This is the service within Hadoop that controls tasks to specific nodes in the cluster.
  - Scalable Performance Data (SPD) Engine
    enables Base SAS users to use Hadoop to store data through the SAS Scalable Performance Data (SPD) Engine. The SPD Engine is designed for high-performance data delivery, reading data sources that contain billions of observations. The engine uses threads to read data very rapidly and in parallel. The SPD Engine reads, writes, and updates data in the HDFS.

- **SAS/ACCESS Interface to Hadoop**
  - HiveServer2
    enables you to interact with your data by using SQL constructs through HiveServer2.
  - Hadoop Distributed File System (HDFS)
    enables you to access data directly from the underlying data storage layer, the Hadoop Distributed File System (HDFS).
  - SQOOP procedure
enables you to transfer data between Hadoop and relational database management systems (RDBMs). Sqoop commands are passed to the cluster using the Apache Oozie Workflow Scheduler for Hadoop.

- SAS/ACCESS Interface to Impala

enables you to issue SQL queries to data that is stored in the Hadoop Distributed File System (HDFS) and Apache Hbase without moving or transforming data. Cloudera Impala is an open-source, massively parallel processing (MPP) query engine that runs natively on Apache Hadoop.

Configuration Information for Other SAS Software

There is other SAS software that builds on the foundation of Base SAS and SAS/ACCESS that uses Hadoop.

To use SAS software to perform in-database processing, contextual analysis, data quality operations, high-performance analytics, or in-memory analytics, additional installation and configuration steps are required.

For more information, see the following documentation:

- Installation and configuration information for in-database processing (including the SAS Embedded Process): *SAS In-Database Products: Administrator’s Guide*

- Installation and configuration information for contextual analysis: *SAS Contextual Analysis In-Database Scoring for Hadoop: Administrator’s Guide*

- Installation and configuration information for data quality operations: *SAS Data Loader for Hadoop: Installation and Configuration Guide*

- Installation and configuration of the High-Performance Analytics Infrastructure: *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*

- Basic installation (not part of a solution installation) of SAS In-Memory Statistics for Hadoop: *SAS LASR Analytic Server: Reference Guide*
Overview of Steps to Configure PROC HADOOP and the FILENAME Statement

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

2. Determine whether you want to connect to the Hadoop server by using Hadoop JAR files or with 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 7.
Note: If you decide to connect to the Hadoop server with 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 9.

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

   For more information, see “Validating PROC HADOOP and the FILENAME Statement to Hadoop Connection” on page 11.

---

Prerequisites for the PROC HADOOP and the FILENAME Statement

Setting Up Your Environment for PROC HADOOP and the FILENAME Statement

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 PROC HADOOP and the FILENAME statement.

   For SAS 9.4, see SAS 9.4 Support for Hadoop for a list of the supported Hadoop distributions and versions.

   For SAS Viya, see SAS Viya: Deployment Guide.

   Note: SAS 9.4 can access a MapR distribution only from a Linux or Windows 64 host.

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

---

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

PROC HADOOP and FILENAME Statement Hadoop JAR and Configuration Files for SAS 9.4

To submit PROC HADOOP or the FILENAME statement to a Hadoop server, a set of Hadoop JAR and configuration files must be available to the SAS client machine. To make the required JAR and configuration files available, you must obtain these files from the Hadoop cluster, copy the files to the SAS client machine, and define the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables to set the location of the JAR and configuration files.
You use SAS Deployment Manager to obtain the JAR and configuration files. For more information, see Appendix 1, “Using the SAS Deployment Manager to Obtain Hadoop JAR and Configuration Files,” on page 31.

Note: Gathering the JAR and configuration files is a one-time process. If you have already gathered the Hadoop JAR and configuration files for another SAS component, you do not need to gather the files again unless you make changes to your Hadoop distribution. For more information, see “When to Collect New JAR and Configuration Files” on page 53.

**PROC HADOOP and FILENAME Statement Hadoop JAR and Configuration Files for SAS Viya**

To submit PROC HADOOP or the FILENAME statement 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 with 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 66.

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.
Note: If your cluster is not secured, the \texttt{dfs.http.address} property or the \texttt{dfs.namenode.http-address} property is used.

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

\begin{verbatim}
<property>
  <name>dfs.https.address</name>
  <value>hwserver1.unx.xyz.com:50070</value>
</property>
\end{verbatim}

or

\begin{verbatim}
<property>
  <name>dfs.namenode.https-address</name>
  <value>hwserver1.unx.xyz.com:50070</value>
</property>
\end{verbatim}

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

\begin{verbatim}
<property>
  <name>dfs.https.address</name>
  <value>maprserver1.unx.xyz.com:14000</value>
</property>
\end{verbatim}

---- or ----

\begin{verbatim}
<property>
  <name>dfs.namenode.https-address</name>
  <value>maprserver1.unx.xyz.com:14000</value>
</property>
\end{verbatim}

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

\section*{Using Apache Knox Gateway Security}

To use PROC HADOOP and the FILENAME statement with a Hadoop cluster that includes Apache Knox Gateway authentication, you must complete these configuration steps:

\begin{itemize}
  \item Connect to the Hadoop server through WebHDFS by defining the \texttt{SAS_HADOOP_RESTFUL} 1 SAS environment variable. Here is an example:

    \begin{verbatim}
    options set=SAS_HADOOP_RESTFUL 1;
    \end{verbatim}

    For more information, see “\texttt{SAS_HADOOP_RESTFUL} Environment Variable” on page 66.

  \item Make sure the configuration files include the properties for the WebHDFS location. For more information, see “Using WebHDFS or HttpFS” on page 7.

  \item Set the SAS environment variable \texttt{KNOX_GATEWAY_URL} to the location of the Knox Gateway. Here is an example:

    \begin{verbatim}
    options set=KNOX_GATEWAY_URL='https://server:port/gateway/default';
    \end{verbatim}

    For more information, see “\texttt{KNOX_GATEWAY_URL} Environment Variable” on page 61.
\end{itemize}
• 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 or Encryption in SAS Viya: Data in Motion.

• Provide an authorized user ID and password in the PROC HADOOP statement or the FILENAME statement to authenticate on the Apache Knox Gateway server. Here is an example:

```sas
proc hadoop username='sasabc' password='sasabc' verbose;
  hdfs mkdir='/user/sasabc/new_directory';
  hdfs delete='/user/sasabc/temp2_directory';
  hdfs copytocalocal='/user/sasabc/testdata.txt'
      out='C:\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:

   ```sas
   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 66.

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.

   ```sas
   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:

   ```xml
   <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 C:\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 "C:\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>
    <value>server.yourcompany.com:50070</value>
    </configuration>

   Note: For the MapR distribution, the fs.default.name property value would include maprfs:///, and the mapred.job.tracker property value would include either maprfs:// or maprfs://server.yourcompany.com:8032.

7. In the PROC HADOOP statement, identify the configuration file with the CFG= argument:

   proc hadoop cfg=cfg1 username='sasabc' password='sasabc' verbose;
   hdfs mkdir='/user/sasabc/new_directory';
   hdfs delete='/user/sasabc/temp2_directory';
   hdfs copytolocal='/user/sasabc/testdata.txt'
   out='C:\Users\sasabc\Hadoop\testdata.txt' overwrite;
Validating PROC HADOOP and the FILENAME Statement to Hadoop Connection

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.

```sas
options set=SAS_HADOOP_CONFIG_PATH="C:\sasdata\hdcluster1\conf";
options set=SAS_HADOOP_JAR_PATH="C:\sasdata\hdcluster1\jars";
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='C:\Users\sasabc\Hadoop\testdata.txt' overwrite;
run;
```

Validating the FILENAME Statement

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

```sas
options set=SAS_HADOOP_CONFIG_PATH="C:\sasdata\hdcluster1\conf";
options set=SAS_HADOOP_JAR_PATH="C:\sasdata\hdcluster1\jars";

filename out hadoop "/user/testing/myfile"
   user="sasabc" pass="abcpass";

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

Documentation for Using PROC HADOOP and the FILENAME Statement

The documentation can be found in these documents:

- “HADOOP Procedure” in *Base SAS Procedures Guide*
- “FILENAME Statement, Hadoop Access Method” in *SAS Global Statements: Reference*
Chapter 4

Configuring SAS/ACCESS for Hadoop

Overview of Steps to Configure SAS/ACCESS Interface to Hadoop 14
Prerequisites for SAS/ACCESS Interface to Hadoop 14
Setting Up Your Environment for SAS/ACCESS Interface to Hadoop 14
Making Hadoop JAR and Configuration Files Available to the SAS Client Machine 15
SAS/ACCESS Hadoop JAR and Configuration Files for SAS 9.4 15
SAS/ACCESS Hadoop JAR and Configuration Files for SAS Viya 15
Additional Configuration for MapR 15
Additional Configuration for All Hadoop Distributions 16
Configuring SAS/ACCESS Interface to Impala 16
Impala ODBC Driver 16
Bulk Loading 17
Configuring PROC SQOOP 17
Prerequisites for PROC SQOOP 17
Configuration for PROC SQOOP 17
Security and User Access to Hadoop 18
Kerberos Security 18
Apache Knox Gateway Security 19
JDBC Read Security 20
HDFS Write Security 20
HDFS Permission Requirements for Optimized Reads 20
Using WebHDFS or HttpFS 20
Working with Hive 21
Starting with Hive 21
Running the Hive Service on Your Hadoop Server 22
Writing Data to Hive: HDFS /tmp and the “Sticky Bit” 23
Validating Your SAS/ACCESS to Hadoop Connection 23
Documentation for Using SAS/ACCESS Interface to Hadoop 24
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. Make Hadoop JAR and configuration files available to the SAS client machine.
   This step involves using SAS Deployment Manager to copy a set of JAR and configuration files to the SAS client machine that accesses Hadoop.
   For more information, see “Making Hadoop JAR and Configuration Files Available to the SAS Client Machine” on page 15.

3. Review the following sections for additional configuration information.
   • SAS/ACCESS Interface to Impala
     “Configuring SAS/ACCESS Interface to Impala” on page 16
   • PROC SQOOP
     “Configuring PROC SQOOP” on page 17
   • HiveServer2
     “Working with Hive” on page 21
   • WebHDFS or HttpFS
     “Using WebHDFS or HttpFS” on page 20

4. Review security and user access.
   For more information, see “Security and User Access to Hadoop” on page 18.

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 23.

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 SAS 9.4, see SAS 9.4 Support for Hadoop for a list of the supported Hadoop distributions and versions.

For SAS Viya, see SAS Viya: Deployment Guide.

Note: SAS 9.4 can access a MapR distribution only from a Linux or Windows 64 host.

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

SAS/ACCESS Hadoop JAR and Configuration Files for SAS 9.4

To use SAS/ACCESS with a Hadoop server, a set of Hadoop JAR and configuration files must be available to the SAS client machine. To make the required JAR and configuration files available, you must obtain these files from the Hadoop cluster, copy the files to the SAS client machine, and define the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables to set the location of the JAR and configuration files.

For more information, see Appendix 1, “Using the SAS Deployment Manager to Obtain Hadoop JAR and Configuration Files,” on page 31.

Note: Gathering the JAR and configuration files is a one-time process. If you have already gathered the Hadoop JAR and configuration files for another SAS component, you do not need to gather the files again unless you make changes to your Hadoop distribution. For more information, see “When to Collect New JAR and Configuration Files” on page 53.

SAS/ACCESS Hadoop JAR and Configuration Files for SAS Viya

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 MapR-based Hadoop systems:

• 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.
For example, on Windows, if the 64-bit MapRClient shareable library location is `C:\mapr\lib`, add this line to JREOPTIONS in the SAS configuration file:

```
-jreoptions (-Djava.library.path=C:\mapr\lib
-Djava.security.auth.login.config=C:\mapr\conf\mapr.login.conf)
```

**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=hybrid`

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

**Note:** SAS no longer supports the 32-bit Windows client.

---

**Additional Configuration for All Hadoop Distributions**

Depending on which in-database product you are running on Base SAS with the SAS Embedded Process, you might get a `java.lang.OutOfMemoryError: PermGen space` error. This error indicates that the Java Virtual Machine PermGen space is exhausted. The PermGen space is not part of the Java Heap memory. It is used to hold metadata that describes the classes and method used by an application.

The PermGen size can be increased to alleviate this issue by adding this line to JREOPTIONS in the SAS configuration file:

```
-jreoptions "(-XX:MaxPermSize=256m)"
```

---

**Configuring SAS/ACCESS Interface to Impala**

**Impala ODBC Driver**

If you are using SAS/ACCESS Interface to Impala to connect to an Impala server on a Cloudera cluster, you must set up the Cloudera Impala ODBC driver. For instructions, see Installation Guide for Cloudera ODBC 2.5.x Driver for Impala.

If you are using SAS/ACCESS Interface to Impala to connect to an Impala server on a MapR cluster, you must set up the MapR Impala ODBC driver. For instructions, see Configure the MapR Impala ODBC Driver for Linux and Mac OSX. In addition to setting up the MapR Impala ODBC driver, you need to set the LIBNAME option `DRIVER_VENDOR=MAPR` or use the `SAS_IMPALA_DRIVER_VENDOR=MAPR` environment variable.
Note: Cloudera ODBC driver for Impala version 2.5.17 or later is required for AIX.

**Bulk Loading**

Using bulk loading with SAS/ACCESS Interface to Impala requires additional configuration.

Bulk loading with the Impala engine is accomplished in two ways:

- By using the WebHDFS or HttpFS interface to Hadoop to push data to HDFS. The SAS environment variable SAS_HADOOP_RESTFUL must be defined and set to a value of 1. You can include the properties for the WebHDFS or HttpFS location in the Hadoop hdfs-site.xml file. Alternatively, specify the WebHDFS or HttpFS host name or the IP address of the server where the external file is stored using the BL_HOST= option. Set the BL_PORT option to either 50700 (WebHDFS) or 14000 (HttpFS). The BULKLOAD= option must be set to YES. No JAR files are needed. It is recommended that you also define the SAS_HADOOP_CONFIG_PATH environment variable.

  For more information, see “Using WebHDFS or HttpFS” on page 20 and Appendix 3, “SAS Environment Variables for Hadoop,” on page 61.

- By configuring a required set of Hadoop JAR files. The JAR files must be in a single location and available to the SAS client machine. The SAS environment variable SAS_HADOOP_JAR_PATH must be defined and set to the location of the Hadoop JAR files. It is recommended that you also define the SAS_HADOOP_CONFIG_PATH environment variable.

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

For more information about bulk loading with SAS/ACCESS Interface to Impala, see SAS/ACCESS for Relational Databases: Reference

---

**Configuring PROC SQOOP**

**Prerequisites for PROC SQOOP**

To use PROC SQOOP, the following prerequisites must be met:

- SAS/ACCESS Interface to Hadoop must be installed and configured.
- Apache Sqoop 1 and Apache Oozie must be installed.

  *Note:* Apache Sqoop Server 2 is not supported.

**Configuration for PROC SQOOP**

- The SAS_HADOOP_CONFIG_PATH environment variable must be defined to include the directory that contains your Hadoop cluster configuration files.

  *Note: *The directory must also contain the hive-site.xml file if you are using the --hive-import Sqoop option.

- The SAS_HADOOP_RESTFUL environment variable must be set to 1 and either WebHDFS or HttpFS must be enabled.
For more information, see “Using WebHDFS or HttpFS” on page 20.

- The generic JDBC Connector is shipped with Sqoop, and it works with most databases. However, because there might be performance issues, it is recommended that you use the specific connector for your database. Most Hadoop distributions are shipped with specialized connectors for DB2, Microsoft SQL Server, MySQL, Netezza, Oracle, and PostgreSQL. For information about connectors, see Understand Connectors and Drivers.

For Cloudera, connector JAR files must be located in the subdirectory of the Oozie shared library rather than the main shared library. Here is an example of an Oozie ADMIN command that you can run to see the contents and location of the shared library that Oozie is using:

```
oozie admin -oozie url-to-oozie-server -shareliblist sqoop
```

For Oracle, you must specify the value to be used for the --table option in Sqoop in uppercase letters because the JDBC Connector requires it. For information about case sensitivity for tables, see the documentation for your specific DBMS.

Connection strings should include the character set option that is appropriate for the data to be imported. For more information, see your connector documentation.

---

**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 might need to update or enable your Java Cryptography Extension (JCE) policy files. When JCE policy files are required, the version of the JRE that you have determines the action you take to obtain or enable the JCE policy files.

To determine your JRE, run this command:

```
SASHome/SASPrivateJavaRuntimeEnvironment/9.4/jre/bin/java -version
```

If your JRE is version 1.7.0_151, you must manually add the Java Cryptography Extension local_policy.jar file in every location where JAVA Home resides on the cluster. If you are located 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. The appropriate JAR files must 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 download the necessary unlimited strength policy files from the Oracle or IBM website.

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

If your JRE is version 1.7.0_161, you already have the unlimited strength policy files. You must enable `unlimited` in the java.security file. To do this, remove the # symbol from this line of the java.security file:

```
#crypto.policy=unlimited
```

If your JRE is version 1.7.0_181, no action is needed. You already have the unlimited strength policy files and `unlimited` is enabled by default.
• For SAS/ACCESS on AIX, if you are using Kerberos security and the Kerberos ticket cache is not stored in the user’s home directory, another line should be added to JREOPTIONS in the SAS configuration file. For example, if the Kerberos ticket caches are stored in `/var/krb5/security/creds`, then also add this line:

```
-DKRB5CCNAME=/var/krb5/security/creds/krb5cc_'id -u'
```

Another example is if the Kerberos ticket caches are stored in `/tmp`, then this line should be added:

```
-DKRB5CCNAME=/tmp/krb5cc_'id -u'
```

• For SAS/ACCESS on HP-UX, set the KRB5CCNAME environment variable to point to your ticket cache whose filename includes your numeric user ID:

```
KRBCNAME="/tmp/krb5cc_'id -u'
```

export KRBCNAME

• For SAS/ACCESS on Windows, ensure that your Kerberos configuration file is in your Java environment. The algorithm to locate the krb5.conf file is as follows:

- If the system property java.security.krb5.conf is set, its value is assumed to specify the path and filename:

```
-jreoptions '(-Djava.security.krb5.conf=C:\[krb5 file])'
```

- If the system property java.security.krb5.conf is not set, the configuration file is looked for in the following directory:

```
<java-home>\lib\security
```

- If the file is still not found, then an attempt is made to locate it:

```
C:\windows\krb5.ini
```

- To connect to a MapR cluster, the following JRE option must be set:

```
-Dhadoop.login=hybrid
```

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

**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 66.

• Make sure the configuration files include the properties for the WebHDFS location. For more information, see “Using WebHDFS or HttpFS” on page 20.

• 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 61.
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* or *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 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 66.

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

   Note: If your cluster is not secured, the `dfs.http.address` property or the `dfs.namenode.http-address` property is used.

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

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

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

```xml
<property>
  <name>dfs.https.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;

INFO : OK
+-------------+--+
|   test.c    |
+-------------+--+
| test        |
+-------------+--+
1 row selected (0.156 seconds)

To access this table from SAS, run this example code:

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:

$ export HIVE_PORT=10000
$ HIVE_HOME/bin/hive --service hiveserver2
**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.

$ hadoop fs -ls /
draxrwxrwt - hdfs hdfs 0 2016-01-21 13:25 /tmp
draxr-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.

*Note:* HiveServer1 was removed with the release of Hive 1.0.0 and in SAS 9.4M4, SAS/ACCESS Interface to Hadoop no longer supports a connection to HiveServer1. For more information, see Delete Hiveserver1.

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

It is assumed that the SAS_HADOOP_CONFIG_PATH and SAS_HADOOP_JAR_PATH environment variables have been set to the location of the client copy of the JAR and configuration files.

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

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:**

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:
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:

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.
Overview of Steps to Configure the SPD Engine

1. Verify that all prerequisites have been satisfied.
   This step ensures that you understand your Hadoop environment. For more information, see “Prerequisites for the SPD Engine” on page 26.

2. Make Hadoop JAR and configuration files available to the SAS client machine.
   For more information, see “Making Hadoop JAR and Configuration Files Available to the SAS Client Machine” on page 26.

3. Review security and user access.
   For more information, see “Kerberos Security” on page 28.

4. Run basic tests to confirm that your Hadoop connections are working.
   For more information, see “Validating the SPD Engine to Hadoop Connection” on page 29.
Prerequisites for the SPD Engine

Setting Up Your Environment for the SPD Engine

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 SPD Engine.
   For SAS 9.4, see SAS 9.4 Support for Hadoop for a list of the supported Hadoop distributions and versions.
   For SAS Viya, see SAS Viya: Deployment Guide.

   Note: SAS 9.4 can access a MapR distribution only from a Linux or Windows 64 host.

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

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

SPD Engine Hadoop JAR and Configuration Files for SAS 9.4

To use the SPD Engine to access files on a Hadoop server, a set of Hadoop JAR and configuration files must be available to the SAS client machine. To make the required JAR and configuration files available, you must obtain these files from the Hadoop cluster, copy the files to the SAS client machine, and define the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables to set the location of the JAR and configuration files.

In SAS 9.4M4, you use SAS Deployment Manager to obtain the JAR and configuration files. For more information, see Appendix 1, “Using the SAS Deployment Manager to Obtain Hadoop JAR and Configuration Files,” on page 31.

Note: Gathering the JAR and configuration files is a one-time process. If you have already gathered the Hadoop JAR and configuration files for another SAS component, you do not need to gather the files again unless you make changes to your Hadoop distribution. For more information, see “When to Collect New JAR and Configuration Files” on page 53.

SPD Engine Hadoop JAR and Configuration Files for SAS Viya

To use the SPD Engine 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.

Note: If you installed the SAS Embedded Process with SAS Viya, the tools.jar file is not included in the installation. This JAR file must be manually located and moved to a certain location.

For more information, see *SAS Viya: Deployment Guide*.

---

**Additional Requirements for MapR Systems**

In addition to the Hive, Hadoop HDFS, and Hadoop authorization JAR files, you need to set the SAS_HADOOP_JAR_PATH directory to point to the JAR files that are provided in the MapR client installation.

In the following example, `C:\third_party\Hadoop\jars` is as described in the previous topic, and `C:\mapr\hadoop\hadoop-0.20.2\lib` is the JAR directory that is specified by the MapR client installation software.

```
set SAS_HADOOP_JAR_PATH=C:\third_party\Hadoop\jars;C:\mapr\hadoop\hadoop-0.20.2\lib
```

In addition, 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 typically installed in the `MAPR_HOME/conf` directory.

For example, on Windows, if the 64-bit MapRClient shareable library location is `C:\mapr\lib`, then add this line to JREOPTIONS in the SAS configuration file:

```
-jreoptions (-Djava.library.path=C:\mapr\lib
-Djava.security.auth.login.config=C:\mapr\conf\mapr.login.conf)
```

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 the 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.

---

**Additional Configuration for All Hadoop Distributions**

Depending on which in-database product you are running on Base SAS with the SAS Embedded Process, you might get a `java.lang.OutOfMemoryError: PermGen space` error. This error indicates that the Java Virtual Machine PermGen space is exhausted. The PermGen space is not part of the Java Heap memory. It is used to hold metadata that describes the classes and method used by an application.

The PermGen size can be increased to alleviate this issue by adding this line to JREOPTIONS in the SAS configuration file:
Kerberos Security

The SPD Engine can be configured for cache-based logon authentication by using MIT Kerberos 5 Version 1.9.

- If you are using Advanced Encryption Standard (AES) encryption with Kerberos, you might need to update or enable your Java Cryptography Extension (JCE) policy files. When JCE policy files are required, the version of the JRE that you have determines the action you take to obtain or enable the JCE policy files.

To determine your JRE, run this command:

```
SASHome/SASPrivateJavaRuntimeEnvironment/9.4/jre/bin/java -version
```

If your JRE is version 1.7.0_151, you must manually add the Java Cryptography Extension local_policy.jar file in every location where JAVA_HOME resides on the cluster. If you are located 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. The appropriate JAR files must 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 download the necessary unlimited strength policy files from the Oracle or IBM website.

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

If your JRE is version 1.7.0_161, you already have the unlimited strength policy files. You must enable **unlimited** in the java.security file. To do this, remove the # symbol from this line of the java.security file:

```
#crypto.policy=unlimited
```

If your JRE is version 1.7.0_181, no action is needed. You already have the unlimited strength policy files and **unlimited** is enabled by default.

- For the SPD Engine on AIX, add this option to your SAS command:
  ```
  -sasoptsappend '(-jreoptions "(-Djavax.security.auth.useSubjectCredsOnly=false)")'
  ```

- For the SPD Engine on HP-UX, set the KRB5CCNAME environment variable to point to your ticket cache whose filename includes your numeric user ID:
  ```
  KRB5CCNAME="/tmp/krb5cc_\'id -u\'"
  export KRB5CCNAME
  ```

- For the SPD Engine on Windows, ensure that your Kerberos configuration file is in your Java environment. The algorithm to locate the krb5.conf file is as follows:
  - If the system property java.security.krb5.conf is set, its value is assumed to specify the path and filename:
    ```
    -jreoptions "(-Djava.security.krb5.conf=C:\\[krb5 file])"
    ```
  - If the system property java.security.krb5.conf is not set, then the configuration file is looked for in this directory:
    ```
    <java-home>\lib\security
    ```
  - If the file is still not found, an attempt is made to locate it as follows:
C:\winnt\krb5.ini

- To connect to a MapR cluster, this JRE option must be set:
  Dhadoop.login=hybrid

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

Validating the SPD Engine to Hadoop Connection

Use the following code to connect to a Hadoop cluster with the SPD Engine. Replace the Hadoop cluster configuration files and JAR files directories with the pathnames for a Hadoop cluster at your site. In addition, replace the primary pathname in the LIBNAME statement with a fully qualified pathname to a directory in your Hadoop cluster.

```sas
options msglevel=i;
options set=SAS_HADOOP_CONFIG_PATH="configuration-files-pathname";
options set=SAS_HADOOP_JAR_PATH="JAR-files-pathname";
libname myspde spde 'primary-pathname' hdfshost=default;
data myspde.class;
  set sashelp.class;
run;
proc datasets library=myspde;
  contents data=class;
run;
  delete class;
run;
quit;
```

Here is the SAS log from a successful connection.
Log 5.1  Successful SPD Engine Connection

16   options msglevel=i;
17   options set=SAS_HADOOP_CONFIG PATH="\mycompany\hadoop\ConfigDirectory\cdh45p1";
18   options set=SAS_HADOOP JAR PATH="\mycompany\hadoop\JARDirectory\cdh45";
19   libname myspde spde '/user/sasabc' hdfshost=default;
NOTE: Libref MYSPDE was successfully assigned as follows:
   Engine:        SPDE
   Physical Name: /user/sasabc/
20   data myspde.class;
21      set sashelp.class;
22   run;
NOTE: There were 19 observations read from the data set SASHELP.CLASS.
NOTE: The data set MYSPDE.CLASS has 19 observations and 5 variables.
NOTE: DATA statement used (Total process time):
       real time           57.00 seconds
       cpu time            0.15 seconds

23
24   proc datasets library=myspde;
25      contents data=class;
26   run;
27
28      delete class;
29   run;
NOTE: Deleting MYSPDE.CLASS (memtype=DATA).
30   quit;
NOTE: PROCEDURE DATASETS used (Total process time):
       real time           37.84 seconds
       cpu time            0.25 seconds

Documentation for Using SPD Engine to Hadoop

The documentation can be found in *SAS SPD Engine: Storing Data in the Hadoop Distributed File System*. 
Appendix 1

Using the SAS Deployment Manager to Obtain Hadoop JAR and Configuration Files

Information and Credentials Required to Configure Hadoop Using SAS Deployment Manager

Using SAS Deployment Manager to Obtain the Hadoop JAR and Configuration Files

Using SAS Deployment Manager to Make Required Hadoop JAR and Configuration Files Available to the SAS Client Machine

Location of Original JAR and Configuration Files after a Redeployment

Supporting Multiple Hadoop Versions and Upgrading Hadoop Version

When to Collect New JAR and Configuration Files

Information and Credentials Required to Configure Hadoop Using SAS Deployment Manager

You need the following information and credentials to use SAS Deployment Manager to configure the Hadoop JAR and configuration files:

- For the Hadoop cluster manager:
  - host name and port
  - credentials (account name and password)
- Hive service host name and port number

Note: Ensure that your Hive service is working properly. One way to do that is to issue this command to check if the cluster is responding in a timely manner:

```
-bash-4.1$ time hive -e 'set -v'
```

This test is also performed by SAS Deployment Wizard and SAS Deployment Manager. When it fails, the message "The SAS Deployment Manager/Wizard cannot instantiate a Hive session. Make sure that the hive command runs successfully on the Hive node host."

appears.

- Oozie service host name and port number
- Impala service host name and port number (Cloudera only)
Either the private key file or the user ID and password as credentials for the UNIX user account that has SSH access to the machine that is hosting the Hive, Oozie, and Impala services

Ensure that Python 2.6 or later and strace are installed. Contact your system administrator if these packages are not installed on the system.

For clusters secured with Kerberos, a valid ticket for the user on the client machine and the Hive service

For cluster managers that are enabled with Transport Layer Security (TLS), a trusted certificate of authority (CA) must be on the host that you access. You must run the SAS Deployment Manager task Add Certificate to Trusted CA Bundle if the trusted CA has not been added to your SAS installation.

Note: All discussion of TLS is also applicable to the predecessor protocol, Secure Sockets Layer (SSL).

The HDFS user home directory, /user/user-account, must exist and have Write permission for the user-account or the mapred account must have a drwxrwxrwx permission for the HDFS/user directory.

Note: This is a critical prerequisite for a MapR cluster.

Authorization to issue HDFS and Hive commands. During SAS Deployment Manager processing, a simple validation test is run to see whether HDFS (hadoop) and Hive (hive) commands can be issued. If the validation test fails, the script that pulls the JAR and configuration files is not be executed.

If you have a Cloudera cluster with Sentry and RecordService services, the Sentry and RecordService services must be configured on the HiveServer2 node. This ensures that the configuration files that are required for Sentry and RecordService services are obtained when SAS Deployment Manager is run.

Using SAS Deployment Manager to Obtain the Hadoop JAR and Configuration Files

Using SAS Deployment Manager to Make Required Hadoop JAR and Configuration Files Available to the SAS Client Machine

You can use SAS Deployment Manager to make required Hadoop JAR and configuration files available to the SAS client machine. SAS Deployment Manager, a tool that enables you to perform some administrative and configuration tasks, is included with each SAS software order. SAS Deployment Manager is located in your SASHome directory, in the\SASDeploymentManager\9.4 folder.

Note: Gathering the JAR and configuration files is a one-time process. If you have already gathered the Hadoop JAR and configuration files for another SAS component, you do not need to gather the files again unless you make changes to your Hadoop distribution. For more information, see “When to Collect New JAR and Configuration Files” on page 53.

Note: When you submit HDFS commands with SAS/ACCESS, you can also connect to the Hadoop server by using WebHDFS or HttpFS. WebHDFS and HttpFS are HTTP REST APIs that support the complete FileSystem interface for HDFS. Using WebHDFS or HttpFS removes the need for client-side JAR files for HDFS, but Hive
JAR files are still needed. For more information, see “Using WebHDFS or HttpFS” on page 20.

After you have installed your SAS software, complete these steps to configure your Hadoop distribution:

1. If you are running on a cluster with Kerberos, you must kinit the HDFS user.
   a. Log on to the server using SSH as root with sudo access.
      
      `ssh username@serverhostname`
      
      `sudo su - root`
   
   b. Enter the following commands to kinit the HDFS user. The default HDFS user is `hdfs`.
      
      `su - hdfs | hdfs-userid`
      
      `kinit -kt location of keytab file user for which you are requesting a ticket`

   Note: For all Hadoop distributions except MapR, the default HDFS user is `hdfs`. For MapR distributions, the default HDFS user is `mapr`.

   Note: If you are running on a cluster with Kerberos, a valid keytab is required for the HDFS user who configures the Hadoop JAR and configuration files. To check the status of your Kerberos ticket on the server, run klist while you are running as the -hdfsuser user. Here is an example:

   `klist`

   Ticket cache: FILE/tmp/krb5cc_493
   Default principal: hdfs@HOST.COMPANY.COM

   Valid starting    Expires           Service principal
   06/20/17 09:51:26 06/27/17 09:51:26 krbtgt/HOST.COMPANY.COM@HOST.COMPANY.COM
   renew until 06/27/17 09:51:26

2. Start SAS Deployment Manager by running sasdm.exe for Windows or sasdm.sh for UNIX. The SAS Deployment Manager script is located in the `/SASHome/SASDeploymentManager/9.4` directory.

   Note: For more information about SAS Deployment Manager pages, click Help on each page.

   The Choose Language page opens.

3. Select the language that you want to use to perform the configuration of your software.

   Click OK. The Select SAS Deployment Manager Task page opens. The items listed under Hadoop Configuration depend on the SAS software that you have licensed.
4. Under Hadoop Configuration, select Configure Hadoop Client Files.

Click Next. The Select Hadoop Distribution page opens.
5. From the drop-down menu, select the distribution of Hadoop that you are using. If your distribution is not listed, exit SAS Deployment Manager and contact SAS Technical Support.

   Note: If your MapR client is on Windows, the MAPR_HOME and JAVA_HOME environment variables must be set. For more information, see MapR: Setting Up the Client.

   Click Next.

   If your Hadoop distribution does not have an administrative client, the Hadoop Cluster Service Information page opens. Skip to Step 9 on page 39.

   If your Hadoop distribution has an administrative client such as Cloudera Manager or Ambari, the Use Cluster Manager page opens.
6. Select the cluster manager administrative tool from the list.

The Hive and Oozie services information that SAS Deployment Manager needs to configure the Hadoop client files can be retrieved from the cluster manager. Select the cluster manager that you want to use to retrieve the information, or select None if you want to specify the information yourself.

If you select None and click Next, the Hadoop Cluster Service Information page opens. Skip to Step 9 on page 39.

If you select a cluster manager and click Next, the Hadoop Cluster Manager Information page opens.
Complete the following information:

- Enter the host name and port number for your Hadoop cluster manager.
  
  For Cloudera, enter the location where Cloudera Manager is running. For Hortonworks, IBM BigInsights, or Pivotal, enter the location where the Ambari server is running.

  The port number is set to the appropriate default after Cloudera, Hortonworks, IBM BigInsights, or Pivotal is selected in step 5.

  Note: The host name must be a fully qualified domain name. The port number must be valid, and the cluster manager must be listening. SAS Deployment Manager validates a constructed URL based on the host name and port number and issues an error message if the URL is not accessible.

- Indicate whether the cluster manager is enabled with Transport Layer Security (TLS).

  Note: All discussion of TLS is also applicable to the predecessor protocol, Secure Sockets Layer (SSL).

  Note: If you choose Yes and a trusted certificate authority (CA) is not found for the host that you are trying to access, an error occurs. A dialog box that prompts you to run the SAS Deployment Manager Add Certificate to Trusted CA Bundle task is displayed. For more information about how to run this task, see “Add Certificate to Trusted CA Bundle” in SAS Deployment Wizard and SAS Deployment Manager: User’s Guide. After you run this task...
to add the certificate, you must restart the Configure Hadoop Client Files task.

Click Next. The Hadoop Cluster Manager Credentials page opens.

8. Enter the Cloudera Manager or Ambari administrator account name and password.

   Note: Use a cluster manager user account whose role is allowed to query the cluster and find the Hive, Oozie, and Impala (Cloudera only) node. Using the credentials of the account that has this role or administrator account eliminates guesswork and removes the chances of a configuration error.

   Click Next. The Hadoop Cluster Service Information page opens.
9. Enter the following information:

   - The host names of the Hive, Impala, and Oozie services for the Hadoop cluster. If you use the cluster manager, this field is populated for you.

   Note: The Impala and Oozie service host names are optional. However, if your SAS software uses Impala or Oozie, you need to enter the Impala or Oozie service host name so that the correct JAR files and configuration files are collected. In addition, the host names for Impala and Oozie are added to the inventory.json file that SAS Deployment Manager creates in this step.

   - The method of Hadoop JAR and configuration file collection. Choose one of these three methods:

     - To use a password of the UNIX user account with SSH for the Hive, Impala, and Oozie hosts, select **SAS Deployment Manager with the UNIX Account with SSH - Password**.

     Click Next. The **Hadoop Cluster SSH Credentials** page opens and asks for a password. Continue with Step 10 on page 42.
To use a private key file (id_rsa) of the UNIX user account with SSH for the Hive, Impala, and Oozie hosts, select **SAS Deployment Manager with the UNIX Account with SSH - Private Key Files**.

Click **Next**. The **Hadoop Cluster SSH Credentials** page opens and asks for a private key file. Skip to **Step 11 on page 42**.
If you do not have SSH credentials or you prefer to manually collect the Hadoop JAR and configuration files, select **Manual Collection**.

The following dialog box appears.

- **Note:** These files must be collected before proceeding with the SAS Deployment Manager task. The configuration fails if the Hadoop JAR and configuration files are not available during the remaining configuration tasks that are executed by SAS Deployment Manager.
For information about how to manually collect the Hadoop JAR and configuration files, see Appendix 2, “Using the Hadoop Tracer Script to Obtain the Hadoop JAR and Configuration Files,” on page 55.

Note: You copy the Hadoop JAR and configuration files to specific directory paths that are requested on the Specify SAS Hadoop Client Directories page in Step 13 on page 44.

After you collect the files, click Next. Skip to Step 12 on page 43.

10. Enter the account name and password of the UNIX user who has SSH access to the machine that is hosting the Hive, Impala, and Oozie services. This information is required to move and copy files to and from hosts.

Note: If Kerberos is installed on your Hadoop cluster, the user should have a Kerberos principal configured.

Click Next. The Hadoop Cluster Service Port Information page opens. Skip to Step 12 on page 43.

11. Enter the following information. This information is required to move and copy files to and from hosts.

• The account name of the UNIX user who has SSH access to the machine that is hosting HiveServer2.

• The path to the location of the private key file.

Note: If Kerberos is installed on your Hadoop cluster, the user should have a Kerberos principal configured.

Click Next. The Hadoop Cluster Service Port Information page opens. Skip to Step 12 on page 43.
12. Enter the port numbers of the Hive, Impala, and Oozie services of your Hadoop cluster.

   Click Next. The Specify SAS Hadoop Client Directories page opens.
13. Specify the locations of the configuration files and JAR files for the Hadoop client and choose whether to select the most recent version of the configuration and JAR files.

The default paths for the configuration and JAR files are created in `hadoop/conf` and `hadoop/lib` in the same parent directory as your `SASHome` directory. In this screen capture, `SASHome` is `/local/install/yourdir/`. Therefore, the default paths are `/local/install/yourdir/hadoop/conf` and `/local/install/yourdir/hadoop/lib`.

**CAUTION:**

The configuration files and JAR files for the Hadoop client must reside in the `/conf` and `/lib` directories, respectively. You can specify a non-default path to the `/conf` and `/lib` directories. If you do not have the `/conf` and `/lib` directories, SAS software cannot find the required files to run successfully.

**Note:** If you want to specify a directory other than the default directory, click **Browse** and select another directory. This step can also create a new directory. However, SAS Deployment Manager creates the `sas_hadoop_config.properties` file and a repository directory in the `hadoop` directory. The repository directory contains the resulting configuration and JAR files each time you run SAS Deployment Manager to pull configuration and JAR files. The files can be found in `/SASHome/hadoop/repository/hive/hive-host-name/time-stamp/`. 
Note: Each time this configuration process is run, the resulting files and libraries are stored in the paths provided here. This path could be a network path if multiple SAS servers are being configured to work with Hadoop.

Click Next. The Update SAS Configuration File sasv9.cfg page opens.

14. If you do not want SAS Deployment Manager to add two Hadoop cluster environment variables to the SAS configuration file, sasv9.cfg, deselect this option.
If you do not use SAS Deployment Manager to define the environment variables, you must manually set the variables later.

The two environment variables are as follows:

- **SAS_HADOOP_CONFIG_PATH**
  This environment variable sets the location of the Hadoop cluster configuration files.

- **SAS_HADOOP_JAR_PATH**
  This environment variable sets the location of the Hadoop JAR files.

Click Next.

If you are obtaining JAR and configuration files for Base SAS or the SPD Engine, the Checking System page opens. Skip to Step 18 on page 50.
If you are obtaining JAR and configuration files for SAS/ACCESS, the Run Validation page opens.
15. (Optional) Validate the configuration of SAS/ACCESS Interface to Hadoop.

If you want to collect the JAR and configuration files without validation, deselect this option.

If there are problems with the validation, an error message appears. You can check the log file, checkaccesshdp_timestamp.log, for the cause of the error. By default, the validation log file can be found in your account home directory:

- UNIX: /your-home/.SASAppData/SASDeploymentWizard
- Windows: C:\users\your-account\AppData\Local\SASDeploymentWizard

Click Next. The Hadoop Cluster Hive Service Information page opens.
16. Enter the schema name for the cluster’s Hive service and select whether Kerberos is enabled on the cluster.

A valid Kerberos ticket must be available on the client machine and Hive service. If a ticket is not available, you must go out to the client machine, cluster, or both and obtain the Kerberos ticket. When the ticket is obtained, you can resume the deployment using SAS Deployment Manager.

**Note:** If you are using Advanced Encryption Standard (AES) encryption with Kerberos, you might need to update or enable your Java Cryptography Extension (JCE) policy files. When JCE policy files are required, the version of the JRE that you have determines the action you take to obtain or enable the JCE policy files. To determine your JRE, run this command:

```
SASHome/SASPrivateJavaRuntimeEnvironment/9.4/jre/bin/java -version
```

- If your JRE is version 1.7.0_151, you must manually add the Java Cryptography Extension local_policy.jar file in every location where JAVA Home resides on the cluster. If you are located 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. The appropriate JAR files must 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 download the necessary unlimited strength policy files from the Oracle or IBM website.
Note: It is recommended to back up any existing local_policy.jar and US_export_policy.jar files first in case they need to be restored.

- If your JRE is version 1.7.0_161, you already have the unlimited strength policy files. You must enable unlimited in the java.security file. To do this, remove the # symbol from this line of the java.security file:

```java
#crypto.policy=unlimited
```

- If your JRE is version 1.7.0_181, no action is needed. You already have the unlimited strength policy files and unlimited is enabled by default.

Click Next. The Hive User Name Credentials page opens.

17. Enter a Hive user name that has access to the Hive service and the password for that Hive user name.

Note: Check with your Hadoop administrator for a valid user name and password.

Click Next. SAS Deployment Manager verifies the prerequisites for the validation and checks for locked files and Write permissions. Checking the system might take several seconds. The Checking System page opens.
18. If any files are shown in the text box after the system check, follow the instructions on the Checking System page to fix any problems.

Click Next. The Summary page opens.
19. Click **Start** to begin the configuration.

*Note:* It takes several minutes to complete the configuration. If Kerberos is installed on your Hadoop cluster, the configuration could take longer.

If the configuration is successful, the name of the page changes to **Deployment Complete** and a green check mark is displayed beside SAS Hadoop Configuration Libraries.
Note: Part of the configuration process runs SAS code to validate the environment. A green check mark indicates that SAS Deployment Manager could connect to Hadoop, run a tracer script, pull back files, and run SAS code to validate the setup. If warnings or errors occur, fix the issues and restart the configuration.

20. Click **Next** to close SAS Deployment Manager.

**Location of Original JAR and Configuration Files after a Redeployment**

If you run SAS Deployment Manager again to redeploy the Hadoop client files, the current JAR and configuration files are placed in the following repository directories on the client machine in the **SASHome** root directory. These files can be retrieved to revert to your previous deployment in case of a problem.

On a Windows client:

C:\SASHome\repository\service-name\host-name-of-service\lib
C:\SASHome\repository\service-name\host-name-of-service\conf

On a UNIX client:

SASHome/hadoop/repository/service-name/host-name-of-service/lib
SASHome/hadoop/repository/service-name/host-name-of-service/conf

service-name is either hive or oozie.

Here are some examples where C:\test\hadoop\ is the SASHome location for Windows and where /test/hadoop/ is the SASHome location for UNIX:

C:\test\hadoop\repository\oozie\oozienode1\lib
C:\test\hadoop\repository\oozie\oozienode1\conf

/test/hadoop/repository/oozie/oozienode1/lib
/test/hadoop/repository/oozie/oozienode1/conf

---

Supporting Multiple Hadoop Versions and Upgrading Hadoop Version

The JAR and configuration files in the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH directories must match the Hadoop server to which SAS connects. If you have multiple Hadoop servers running different Hadoop versions, create and populate separate directories with version-specific Hadoop JAR and configuration files for each Hadoop version.

The SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH directories must be dynamically set depending on which Hadoop server a SAS job or SAS session connects to. To dynamically set SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH, either use the OPTION statement or create a wrapper script associated with each Hadoop version. SAS is invoked via the option or a wrapper script that sets SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH appropriately to pick up the JAR and configuration files that match the target Hadoop server.

Upgrading your Hadoop server version might involve multiple active Hadoop versions. The same multi-version instructions apply.

---

When to Collect New JAR and Configuration Files

Once you gather the Hadoop JAR and configuration files for a SAS component using SAS Deployment Manager, you do not need to do it again unless changes are made to your Hadoop system.

You need to re-run SAS Deployment Manager to collect new JAR files if either of the following conditions occur:

- you upgrade your Hadoop installation
- you install a new Hadoop parcel, package, service, or component on an existing cluster

You need to re-run SAS Deployment Manager to collect new configuration files if either of the following conditions occur:

- you upgrade your Hadoop installation
- you install a new Hadoop parcel, package, service, or component on an existing cluster
you make any configuration changes to the Hadoop services or components

*Note:* If you chose to manually collect the Hadoop JAR and configuration files, you can re-run the Hadoop tracer script. For more information, see Appendix 2, “Using the Hadoop Tracer Script to Obtain the Hadoop JAR and Configuration Files,” on page 55.
Overview of the Hadoop Tracer Script

Part of the configuration process with SAS Deployment Manager involves connection to the Hadoop cluster and automatically running the Hadoop tracer script to collect JAR and configuration files. If you do not have SSH credentials or if you prefer to manually collect the Hadoop JAR and configuration files, there is an option to manually run the Hadoop tracer script. For more information, see Step 9 on page 39.

The Hadoop tracer script performs these tasks:

- Pulls the necessary Hadoop JAR and configuration files and places them in the \( /tmp/jars \) directory and the \( /tmp/sitexmls \) directory, respectively.

- Creates a hadooptracer.json file in the \( /tmp \) directory. You can specify a custom path.

- Creates a log file, hadooptracer.log, in the \( /tmp \) directory.

If a problem occurs and you need to contact SAS Technical Support, include this log file with your support track.

Note: Typically, the Hadoop administrator manually collects the required Hadoop JAR and configuration files and provides the SAS administrator with either the files or access to the \( /tmp/jars \) and \( /tmp/sitexmls \) directories.
Prerequisites for Using the Hadoop Tracer Script

These are prerequisites to ensure that the Hadoop tracer script runs successfully:

• Ensure that Python 2.6 or later and strace are installed. Contact your system administrator if these packages are not installed on the system.

• Ensure that the user running the script has authorization to issue HDFS and Hive commands.

• If Hadoop is secured with Kerberos, obtain a Kerberos ticket for the user before running the script.

Note: There is no requirement for SSH access if you manually run the Hadoop tracer script. However, if you use SAS Deployment Manager to obtain the Hadoop JAR and configuration files, a UNIX account with SSH access is required.

Obtaining the Hadoop Tracer Script

To obtain the Hadoop tracer script:

Note: These steps are typically performed by the SAS administrator.

1. Navigate to the `SASHome/SASHadoopConfigurationLibraries/2.5/data/` directory.
2. Locate the hadooptracer.py script.
   - If your Hadoop administrator runs the script, send the administrator the script and the instructions in this appendix.

Running the Hadoop Tracer Script

To run the Hadoop tracer script:

Note: These steps are typically performed by the Hadoop administrator.

1. Change permissions on the file so that it has EXECUTE permission.
   ```
   chmod 755 ./hadooptracer.py
   ```
2. Run the tracer script.
   ```
   python ./hadooptracer.py --filterby=latest --postprocess
   ```

   **TIP** The `filterby=latest` option ensures that if duplicate JAR or configuration files exist, the latest version is selected. If you want to pull the necessary JAR files without filtering, use `filterby=none` or do not use the `filterby` argument at all.

   **TIP** The `postprocess` option ensures that the `${hdp.version}` tokens are replaced. It is highly recommended that you run the tracer script with this option.
Here is the syntax for the hadooptracer.py script:

```
hadooptracer.py --filterby=latest | none --winplatform --postprocess
--debug --version --nothreads -f /custom-path/hadooptracer.json -h
```

--filterby=latest
ensures that if duplicate JAR or configuration files exist, the latest version is selected.

--filterby=none
no filter is applied when JAR or configuration files are collected.

--winplatform
must use if hadooptracer.py is run from a Windows client that connects to a Linux Hadoop cluster. This option enables the mapreduce.app-submission.cross-platform property to be set in the mapred-site.xml file.

--postprocess
removes version numbers from configuration files that are collected.

**Tip** It is highly recommended that you run the tracer script with this option.
This option is ignored for Cloudera clusters.

--debug
prints verbose logging information.

--version
prints the version of the hadooptracer.py script that is run.

--nothreads
is useful for small Hadoop clusters or for clusters with low memory.

-f /custom-path/hadooptracer.json
specifies a custom path for the JSON output file.

-h
specifies the options available for the hadooptracer.py script.

3. Provide the SAS administrator with either a copy of these files or access to the /tmp/jars and /tmp/sitexmls directories.

---

### Setting Up the JAR and Configuration File Directories on the SAS Client Machine

**Note:** These steps are typically performed by the SAS administrator.

To set up the JAR and configuration file directories on the SAS client machine:

1. Create two directories that are accessible to the SAS client machine to hold the JAR and configuration files. Examples would be /opt/sas/hadoopfiles/conf and /opt/sas/hadoopfiles/lib.

**CAUTION:**

The JAR and configuration files for the Hadoop client must reside in the /lib and /conf directories, respectively. You can specify a non-default path to the /lib and /conf directories. If you do not have the /lib and /conf directories, SAS software cannot find the required files to run successfully.
2. Using a method of your choice (for example, PSFTP, SFTP, SCP, or FTP), copy the files in the `/tmp/jars` and the `/tmp/sitexmls` directories on the Hadoop server to the directories on the SAS client machine that you created in step 1, such as `/opt/sas/hadoopfiles/lib` and `/opt/sas/hadoopfiles/conf`, respectively.

Copy the files that you received from the Hadoop administrator to the SAS client machine.

- If you received the files directly, copy the files to their respective directories on the SAS client machine that you created in step 1, such as `/opt/sas/hadoopfiles/lib` (JAR files) and `/opt/sas/hadoopfiles/conf` (configuration files).

- If you received access to the `/tmp/jars` and the `/tmp/sitexmls` directories on the Hadoop server, use a method of your choice (for example, PSFTP, SFTP, SCP, or FTP) and copy the files in their respective directories to the directories on the SAS client machine that you created in step 1, such as `/opt/sas/hadoopfiles/lib` (JAR files) and `/opt/sas/hadoopfiles/conf` (configuration files).

**Note:** If you connect to the Hadoop server with an HTTP REST API, you do not need the Hadoop JAR files on the SAS client machine.

3. Additional JAR and configuration files might be needed because of JAR file interdependencies and your Hadoop distributions. For more information, see “Supporting Multiple Hadoop Versions and Upgrading Hadoop Version” on page 53.

If needed, repeat steps 1 and 2 and add these JAR and configuration files to different file directories.

4. (Optional) Define the SAS environment variable SAS_HADOOP_JAR_PATH. Set the variable to the directory path for the Hadoop JAR files.

**Note:** Step 14 on page 45 of SAS Deployment Manager automatically sets the SAS_HADOOP_JAR_PATH for you, if you prefer.

For example, if the JAR files are copied to the location `C:\opt\sas\hadoopfiles\lib`, the following syntax sets the environment variable appropriately. If the pathname contains spaces, enclose the pathname value in double quotation marks.

```sas
/* SAS command line */
-set SAS_HADOOP_JAR_PATH "C:\opt\sas\hadoopfiles\lib"
```

or

```dos
/* DOS prompt */
set SAS_HADOOP_JAR_PATH "C:\opt\sas\hadoopfiles\lib"
```

or

```sas
/* SAS command UNIX */
export SAS_HADOOP_JAR_PATH="/opt/sas/hadoopfiles/lib"
```

To concatenate pathnames, the following OPTIONS statement in the Windows environment sets the environment variable appropriately:

```sas
options set=SAS_HADOOP_JAR_PATH="C:\opt\sas\hadoopfiles\lib; C:\MyHadooplib";
```

For more information about the environment variable, see “SAS_HADOOP_JAR_PATH Environment Variable” on page 64.
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. For more information, see “Supporting Multiple Hadoop Versions and Upgrading Hadoop Version” on page 53.

5. (Optional) Define the SAS environment variable SAS_HADOOP_CONFIG_PATH. Set the variable to the directory path for the Hadoop configuration files.

Note: Step 14 on page 45 of SAS Deployment Manager automatically sets the SAS_HADOOP_CONFIG_PATH for you, if you prefer.

For example, if the configuration files are copied to the location C:\opt\sas\hadoopfiles\conf, the following syntax sets the environment variable appropriately. If the pathname contains spaces, enclose the pathname value in double quotation marks.

    /* SAS command line */
    -set SAS_HADOOP_CONFIG_PATH "C:\opt\sas\hadoopfiles\conf"

or

    /* DOS prompt */
    set SAS_HADOOP_CONFIG_PATH "C:\opt\sas\hadoopfiles\conf"

or

    /* SAS command UNIX */
    export SAS_HADOOP_CONFIG_PATH="/opt/sas/hadoopfiles/conf"

To concatenate pathnames, the following OPTIONS statement in the Windows environment sets the environment variable appropriately:

    options set=SAS_HADOOP_CONFIG_PATH="C:\opt\sas\hadoopfiles\conf;C:\MyHadoopConf";

For more information about the environment variable, see “SAS_HADOOP_CONFIG_PATH Environment Variable” on page 62.

6. If you are using PROC HADOOP and your Hadoop distribution is MapR, additional configuration is needed. For more information, see “Additional Configuration for MapR” on page 60.

Note: To submit HDFS commands, you can also connect to the Hadoop server by using WebHDFS or HttpFS. Using WebHDFS or HttpFS removes the need for client-side JAR files for HDFS, but Pig JAR files are still needed. For more information, see “Using WebHDFS or HttpFS” on page 7.

Additional Configuration for IBM BigInsights

If you run the Hadoop tracer script on an IBM BigInsights cluster that is secured with Kerberos before obtaining a Kerberos ticket, you must manually copy the automaton.1.11.8.jar JAR file to the location of the other JAR files on the SAS client machine. An example is C:\opt\sas\hadoopfiles\lib.
Additional Configuration for MapR

If you run the Hadoop tracer script on MapR, you must copy the following JAR files to the location of the other JAR files on the SAS client machine. An example is C:\opt\sas\hadoopfiles\lib.

- pig-core-h2.jar
- jline-1.0.jar or jruby.complete.1.6.7.jar
- automaton.1.11.8.jar
Appendix 3
SAS Environment Variables for Hadoop

Dictionary

SAS Environment Variables

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

Restriction: This environment variable is not valid on the CAS server.

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";
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>&quot;;</td>
</tr>
<tr>
<td>Windows</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>&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.

**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, SPD Engine
- **Restriction:** This environment variable is not valid on the CAS server.
- **Requirement:** The SAS_HADOOP_CONFIG_PATH environment variable must be set regardless of whether you are using JAR files or WebHDFS or HttpFS.
This environment variable is automatically set if you accept the default configuration values in SAS Deployment Manager when you run the Configure Hadoop Client Files task.

**Syntax**

SAS_HADOOP_CONFIG_PATH *pathname*

**Required Argument**

*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 C:\sasdata\cluster1\conf, then the following OPTIONS statement syntax sets the environment variable appropriately.

```sas
options set=SAS_HADOOP_CONFIG_PATH "C:\sasdata\cluster1\conf";
```

**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.

*Note:* Only one SAS_HADOOP_CONFIG_PATH path is used per Hadoop cluster. To see the path, enter the following command:

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

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

**Table A3.2 Defining the SAS_HADOOP_CONFIG_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_CONFIG_PATH &quot;\sasdata\cluster1\conf&quot;</td>
</tr>
<tr>
<td></td>
<td>SAS invocation</td>
<td>-set SAS_HADOOP_CONFIG_PATH &quot;\sasdata\cluster1\conf&quot;</td>
</tr>
<tr>
<td></td>
<td>OPTIONS statement</td>
<td>options set=SAS_HADOOP_CONFIG_PATH=&quot;\sasdata/cluster1/conf&quot;</td>
</tr>
</tbody>
</table>
### Operating Environment

<table>
<thead>
<tr>
<th>Method</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS configuration file</td>
<td>-set SAS_HADOOP_CONFIG_PATH &quot;C:sasdata\cluster1\conf&quot;</td>
</tr>
<tr>
<td>SAS invocation</td>
<td>-set SAS_HADOOP_CONFIG_PATH &quot;C:sasdata\cluster1\conf&quot;</td>
</tr>
<tr>
<td>OPTIONS statement</td>
<td>options set=SAS_HADOOP_CONFIG_PATH=&quot;C:sasdata\cluster1\conf&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.

---

### 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, SPD Engine

**Restriction:** This environment variable is not valid on the CAS server.

**Note:** This environment variable is automatically set if you accept the default configuration values in SAS Deployment Manager when you run the **Configure Hadoop Client Files** task.

**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 semicolon (;) in the Windows environment or a colon (:) in a UNIX environment.

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

    options set=SAS_HADOOP_JAR_PATH="C:\third_party\Hadoop\jars\lib";

To concatenate pathnames, the following OPTIONS statement in the Windows environment sets the environment variable appropriately.

    options set=SAS_HADOOP_JAR_PATH="C:\third_party\Hadoop\jars\lib;\mine\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>
<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>
<tr>
<td>Windows</td>
<td>SAS configuration file</td>
<td>-set SAS_HADOOP_JAR_PATH &quot;C:\third_party\Hadoop\jars\lib&quot;</td>
</tr>
<tr>
<td></td>
<td>SAS invocation</td>
<td>-set SAS_HADOOP_JAR_PATH &quot;C:\third_party\Hadoop\jars\lib&quot;</td>
</tr>
<tr>
<td></td>
<td>OPTIONS statement</td>
<td>options set=SAS_HADOOP_JAR_PATH=&quot;C:\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. For more information, see “Supporting Multiple Hadoop Versions and Upgrading Hadoop Version” on page 53.

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**

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

<table>
<thead>
<tr>
<th>Valid in:</th>
<th>SAS configuration file, SAS invocation, OPTIONS statement, SAS System Options window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used by:</td>
<td>FILENAME statement Hadoop access method, HADOOP procedure, SAS/ACCESS Interface to Hadoop, SAS/ACCESS Interface to Impala</td>
</tr>
<tr>
<td>Default:</td>
<td>0, which connects to the Hadoop server with JAR files</td>
</tr>
<tr>
<td>Restriction:</td>
<td>This environment variable is not valid on the CAS server.</td>
</tr>
</tbody>
</table>

**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>-set SAS_HADOOP_RESTFUL 1</td>
</tr>
<tr>
<td>SAS invocation</td>
<td>-set SAS_HADOOP_RESTFUL 1</td>
</tr>
<tr>
<td>Method</td>
<td>Example</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>OPTIONS statement</td>
<td><code>options set=SAS_HADOOP_RESTFUL 1;</code></td>
</tr>
</tbody>
</table>
Recommended Reading

- Base SAS Procedures Guide
- Encryption in SAS
- SAS/ACCESS for Relational Databases: Reference
- SAS Global Statements: Reference
- SAS SPD Engine: Storing Data in the Hadoop Distributed File System
- SAS and Hadoop Technology: Overview
- SAS Viya for Linux: Deployment 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

### A
- Apache Knox Gateway security
  - FILENAME statement 8
  - PROC HADOOP 8
  - SAS/ACCESS interface to Hadoop 19
- Apache Oozie
  - PROC HADOOP 9
  - PROC SQOOP 17

### C
- configuration files
  - FILENAME statement 6
  - PROC HADOOP 6
  - SAS/ACCESS interface to Hadoop 15
  - SPD Engine 26
  - when to refresh 53

### D
- documentation for using
  - FILENAME statement 11
  - PROC HADOOP 11
  - SAS/ACCESS interface to Hadoop 24
  - SPD Engine 30

### E
- environment variable
  - KNOX_GATEWAY_URL 61
  - SAS_HADOOP_CONFIG_PATH 62
  - SAS_HADOOP_JAR_PATH 64
  - SAS_HADOOP_RESTFUL 66

### F
- FILENAME statement
  - configuration files 6
  - documentation for using 11
  - Hadoop distributions 6
  - Hadoop JAR files 6
  - HttpFS 7
  - multiple Hadoop versions 53
  - validating Hadoop connection 11

### H
- Hadoop connection
  - FILENAME statement 11
  - PROC HADOOP 11
  - SAS/ACCESS interface to Hadoop 23
  - SPD Engine 29
- Hadoop distributions
  - FILENAME statement 6
  - PROC HADOOP 6
  - SAS/ACCESS interface to Hadoop 14
  - SPD Engine 26
- Hadoop JAR files
  - FILENAME statement 6
  - PROC HADOOP 6
  - SAS/ACCESS interface to Hadoop 15
  - SPD Engine 26
- Hadoop tracer script
  - additional configuration for IBM
    - BigInsights 59
  - additional configuration for MapR 60
  - directories on SAS client 57
  - obtaining 56
  - prerequisites 56
  - running 56
- HiveServer2, SAS/ACCESS interface to Hadoop 21
- HttpFS
  - FILENAME statement 7
  - PROC HADOOP 7
  - SAS/ACCESS interface to Hadoop 20

### J
- JAR files
  - FILENAME statement 6
  - PROC HADOOP 6
  - SAS/ACCESS interface to Hadoop 15
  - SPD Engine 26
  - when to refresh 53
K
Kerberos security
SAS/ACCESS interface to Hadoop 18
SPD Engine 28
Knox security
FILENAME statement 8
PROC HADOOP 8
SAS/ACCESS interface to Hadoop 19
KNOX_GATEWAY_URL environment variable 61

M
multiple Hadoop versions
FILENAME statement 53
PROC HADOOP 53
SAS/ACCESS interface to Hadoop 53

P
prerequisites
SAS/ACCESS interface to Hadoop 14
PROC HADOOP
Apache Oozie 9
configuration files 6
documentation for using 11
Hadoop distributions 6
Hadoop JAR files 6
HttpFS 7
multiple Hadoop versions 53
validating Hadoop connection 11
WebHDFS 7
PROC SQOOP
configuration 17

S
SAS Deployment Manager
FILENAME statement 6
PROC HADOOP 6
SAS_HADOOP_CONFIG_PATH environment variable 62
SAS_HADOOP_JAR_PATH environment variable 64
SAS_HADOOP_RESTFUL environment variable 66
SAS/ACCESS interface to Hadoop
collection files 15
Hadoop distributions 14
Hadoop JAR files 15
HiveServer2 21
HttpFS 20
multiple Hadoop versions 53
prerequisites 14
security 18
validating Hadoop connection 23
WebHDFS 20
SAS/ACCESS to Impala
configuration 16
security
SAS/ACCESS interface to Hadoop 18
SPD Engine 28
SPD Engine
configuration files 26
documentation for using 30
Hadoop distributions 26
Hadoop JAR files 26
security 28
validating Hadoop connection 29
system requirements
SAS/ACCESS 14

V
validating Hadoop connection
FILENAME statement 11
PROC HADOOP 11
SAS/ACCESS interface to Hadoop 23
SPD Engine 29

W
WebHDFS
FILENAME statement 7
PROC HADOOP 7
SAS/ACCESS interface to Hadoop 20
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.