SAS® 9.4 Hadoop Configuration Guide for Base SAS® and SAS/ACCESS®

Second Edition

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2015. *SAS® 9.4 Hadoop Configuration Guide for Base SAS® and SAS/ACCESS®, Second Edition*. Cary, NC: SAS Institute Inc.

SAS® 9.4 Hadoop Configuration Guide for Base SAS® and SAS/ACCESS®, Second Edition

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513-2414.

January 2016

Contents

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, Hive, 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.
- Ensure that the HCatalog, HDFS, Hive, MapReduce, Oozie, Sqoop, and YARN services are running on the Hadoop cluster. SAS software uses these various services and this 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.

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 any host machine that you can connect to on a Hadoop cluster.
  - 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.
  - SQOOP procedure

    enables Base SAS users 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.
  - Scalable Performance Data (SPD) Engine

    enables Base SAS users to use Hadoop to store SAS data through the SAS Scalable Performance Data (SPD) Engine. The SPD Engine is designed for high-performance data delivery, reading data sets 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

  enables you to interact with your data by using SQL constructs through Hive and HiveServer2. It also enables you to access data directly from the underlying data storage layer, the Hadoop Distributed File System (HDFS).

- 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, 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 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*

3

Configuring FILENAME Statement Hadoop Access Method and PROC HADOOP

Overview of Steps to Configure the FILENAME Statement and PROC HADOOP

1 Verify that all prerequisites have been satisfied.

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

2 Determine whether you want to connect to the Hadoop server by using Hadoop JAR files or with an HTTP REST API.

   For more information, see “Configuring Hadoop JAR Files” on page 7 and “Using WebHDFS or HttpFS” on page 9.

3 Make Hadoop configuration files available to the SAS client machine.

   For more information, see “Making Hadoop Cluster Configuration Files Available to the SAS Client Machine” on page 11.

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

   For more information, see “Validating the FILENAME Statement and PROC HADOOP to Hadoop Connection” on page 14.

Prerequisites for the FILENAME Statement and PROC HADOOP

Setting Up Your Environment for the FILENAME Statement and PROC HADOOP

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

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

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

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

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

   Note: SAS 9.4 for AIX requires Cloudera CDH 4.5 or Hortonworks 1.3.2 or later when you use PROC HADOOP with Kerberos 5 Version 1.9.

   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.

Configuring Hadoop JAR Files

Configuring Hadoop JAR Files Using SAS Deployment Manager

If you license SAS/ACCESS Interface to Hadoop, you can use SAS Deployment Manager to make required Hadoop JAR and configuration files available to the SAS client machine for the FILENAME statement and PROC HADOOP. For more information about using SAS Deployment Manager for SAS/ACCESS Interface to Hadoop, see “Configuring Hadoop JAR and Configuration Files” on page 17.

If you do not license SAS/ACCESS Interface to Hadoop, you must follow the steps in “Making Required Hadoop JAR Files Available to the SAS Client Machine” on page 7 to submit the FILENAME statement or PROC HADOOP by using Hadoop JAR files.

Making Required Hadoop JAR Files Available to the SAS Client Machine

To submit the FILENAME statement or PROC HADOOP to a Hadoop server by using Hadoop JAR files, the required JAR files must be available to the SAS client machine. To make the required JAR files available, you must define the SAS_HADOOP_JAR_PATH environment variable to set the location of the JAR files:

1. Create a directory that is accessible to the SAS client machine.
2. From the specific Hadoop cluster, copy Hadoop HDFS and Hadoop authorization JAR files for the particular Hadoop distribution to the directory that was created in step 1.

   For example, here are the required JAR files for CDH 4.7. The set is different for other Hadoop distributions.

   guava
   hadoop-auth
   hadoop-common
   hadoop-core
   hadoop-hdfs
   hive-exec
   hive-jdbc
   hive-metastore
   hive-service
   libfb303
   pig
   protobuf-java

   Appendix 1, “Hadoop JAR Files,” on page 51 lists the required JAR files for each Hadoop distribution.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. Your Hadoop administrator can assist you in locating the appropriate JAR files.

Additional JAR files might be needed because of JAR file interdependencies and your Hadoop distribution. For more information, see "Supporting Multiple Hadoop Versions and Upgrading Hadoop Version" on page 8.

3 Define the SAS environment variable SAS_HADOOP_JAR_PATH. Set the variable to the directory path for the Hadoop JAR files.

For example, if the JAR files are copied to the location **C:\third_party\Hadoop\jars**, the following syntax sets the environment variable appropriately. If the pathname contains spaces, enclose the pathname value in double quotation marks.

```
-set SAS_HADOOP_JAR_PATH "C:\third_party\Hadoop\jars" /* SAS command line */
```

or

```
set SAS_HADOOP_JAR_PATH "C:\third_party\Hadoop\jars" /* DOS prompt */
```

or

```
export SAS_HADOOP_JAR_PATH="/third_party/hadoop/jars" /* SAS command UNIX */
```

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;C:\MyHadoopJars";
```

For more information about the environment variable, see "SAS_HADOOP_JAR_PATH Environment Variable" on page 76.

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 46.

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 9.

Supporting Multiple Hadoop Versions and Upgrading Hadoop Version

The JAR files in the SAS_HADOOP_JAR_PATH directory 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 files for each Hadoop version.

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

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

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. 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      /* SAS command line */
```

or

```
-set SAS_HADOOP_RESTFUL 1     /* DOS prompt */
```

or

```
export SAS_HADOOP_RESTFUL=1   /* UNIX */
```

For more information, see “SAS_HADOOP_RESTFUL Environment Variable” on page 78.

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

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

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

or

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

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

```
<property>
<name>dfs.http.address</name>
<value>maprserver1.unx.xyz.com:14000</value>
</property>
---- 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 Cluster Configuration Files Available to the SAS Client Machine” on page 11.

Using Apache Oozie

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

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

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

   ```
   set SAS_HADOOP_RESTFUL 1      /* SAS command line */
   ```

   or

   ```
   -set SAS_HADOOP_RESTFUL 1     /* DOS prompt */
   ```

   or

   ```
   export SAS_HADOOP_RESTFUL=1   /* UNIX */
   ```

   For more information, see “SAS_HADOOP_RESTFUL Environment Variable” on page 78.

2. Create a directory that is accessible to the SAS client machine.

3. From the specific Hadoop cluster, copy these configuration files to the directory created in step 2.

   core-site.xml

   hdfs-site.xml

4. Make sure the hdfs-site.xml configuration file includes the properties for the WebHDFS location. The configuration file includes the **dfs.http.address** property or the **dfs.namenode.http-address** property. If the **dfs.http.address** property is not in the configuration file, the **dfs.namenode.http-address** property is used if it is in the file.

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

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

   or

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

```
</property>
```

5 Define the SAS environment variable named SAS_HADOOP_CONFIG_PATH. Set the environment variable to the directory path for the Hadoop cluster configuration files. For example, if the cluster configuration files are copied to the location **`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;
```

Making Hadoop Cluster Configuration Files Available to the SAS Client Machine

Configuring Hadoop Cluster Configuration Files Using SAS Deployment Manager

If you license SAS/ACCESS Interface to Hadoop, you can use SAS Deployment Manager to make required Hadoop JAR and configuration files available to the SAS client machine for the FILENAME statement and PROC HADOOP. For more information about using SAS Deployment Manager for SAS/ACCESS

Interface to Hadoop, see “Configuring Hadoop JAR and Configuration Files” on page 17.

If you do not license SAS/ACCESS Interface to Hadoop, you must perform one of these tasks:

- Copy the configuration files to a physical location that is accessible to the SAS client machine and then set the SAS environment variable SAS_HADOOP_CONFIG_PATH to the location. For more information, see “Copying the Hadoop Cluster Configuration Files” on page 12.
- Create a single configuration file by merging the properties from the multiple Hadoop cluster configuration files and then identify the configuration file with the PROC HADOOP or FILENAME statement CFG= argument. For more information, see “Using a Single Configuration File” on page 12.

Copying the Hadoop Cluster Configuration Files

To connect to a Hadoop server, you must make the configuration files available to the SAS client machine:

1. Create a directory that is accessible to the SAS client machine.
2. From the specific Hadoop cluster, copy these configuration files to the directory created in step 1.

   core-site.xml

   hdfs-site.xml

   hive-site.xml

   mapred-site.xml

   yarn-site.xml

   Note: For a MapReduce 1 cluster, only the mapred-site.xml file is needed. For a MapReduce 2 cluster and a YARN cluster, the mapred-site.xml file and the yarn-site.xml file are needed.
3. 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"
   ```

Using a Single Configuration File

To connect to a Hadoop server with the FILENAME statement or PROC HADOOP, a single configuration file can be created. The configuration file must then be identified in the FILENAME statement or PROC HADOOP statement with the CFG= option. The configuration file must specify the name and JobTracker addresses for the specific server.

To create a single configuration file:

1. Create a directory that is accessible to the SAS client machine.

2 Create a single configuration file with the properties from the Hadoop core configuration file or by merging the properties from multiple Hadoop configuration files.

- The configuration file can be a copy of the core-site.xml configuration file.
- If your Hadoop cluster is running Kerberos security or with HDFS failover enabled, create a configuration file that combines the properties of core-site.xml and the hdfs-site.xml configuration files.
- If you are using MapReduce 1, merge the properties from the core-site.xml, hdfs-site.xml, and mapred-site.xml configuration files.
- If you are using MapReduce 2 and YARN, merge the properties from the core-site.xml, hdfs-site.xml, mapred-site.xml, and yarn-site.xml configuration files.

The merged configuration file must have one beginning <configuration> tag and one ending </configuration> tag. Only the properties should exist between the <configuration>…</configuration> tags. Here is an example of a configuration file with merged properties:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>abcdef.sas.com:8021</value>
  </property>

<!-- lines omitted for sake of brevity -->

  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://abcdef.sas.com:8020</value>
  </property>
</configuration>
```

3 Save the configuration file in the directory created in step 1.

4 In the FILENAME statement or PROC HADOOP statement, identify the configuration file with the CFG= option:

```
filename cfg1 'C:\Users\sasabc\hadoop\sample_config.xml';

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;
run;
```

Validating the FILENAME Statement and PROC HADOOP to Hadoop Connection

Validating the FILENAME Statement

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

```
filename out hadoop "/user/testing/myfile"
   cfg="C:\users\sasabc\hadoop\sample_config.xml"
   user="sasabc" pass="abcpass";

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

Validating PROC HADOOP

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

```
filename cfg 'C:\Users\sasabc\hadoop\sample_config.xml';
proc hadoop cfg=cfg 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;
```

Documentation for Using the FILENAME Statement and PROC HADOOP

The documentation can be found in the following documents:

- "FILENAME Statement, Hadoop Access Method" in *SAS Statements: Reference*
- "HADOOP" in *Base SAS Procedures Guide*

4

Configuring SAS/ACCESS for Hadoop

Overview of Steps to Configure SAS/ACCESS Interface to Hadoop

1. Verify that all prerequisites have been satisfied.

   This step ensures that you understand your Hadoop environment. For more information, see “Prerequisites for SAS/ACCESS Interface to Hadoop” on page 17.

2. Review security and user access.

   For more information, see “Security and User Access to Hadoop” on page 36.

3. 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 “Configuring Hadoop JAR and Configuration Files” on page 17.

4. Review the following sections for additional configuration information.

   - SAS/ACCESS Interface to Impala

     “Configuring SAS/ACCESS Interface to Impala” on page 34

   - PROC SQOOP

     “Configuring PROC SQOOP” on page 35

   - Hive and HiveServer2

     “Working with Hive and HiveServer2” on page 38

   - WebHDFS or HttpFS

     “Using WebHDFS or HttpFS” on page 37

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 40.

Prerequisites for SAS/ACCESS Interface to Hadoop

Setting Up Your Environment for SAS/ACCESS Interface to Hadoop

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

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

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

2 Review the supported Hadoop distributions.

For a list of supported Hadoop distributions and versions, see SAS 9.4 Supported Hadoop Distributions.

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

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

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

3 Install SAS/ACCESS Interface to Hadoop by following the instructions in your software order email.

Configuring Hadoop 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
- Oozie service host name

- SSH credentials of the administrator who has access to both Hive and Oozie services
- For clusters that have Kerberos security enabled, a valid ticket on the client machine and the Hive service
- 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.

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

In the February 2015 release, 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: 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 37.

After you have installed SAS/ACCESS Interface to Hadoop, 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
06/20/15 09:51:26 06/27/15 09:51:26 krbtgt/HOST.COMPANY.COM@HOST.COMPANY.COM
```

```
renew until 06/27/15 09:51:26
```

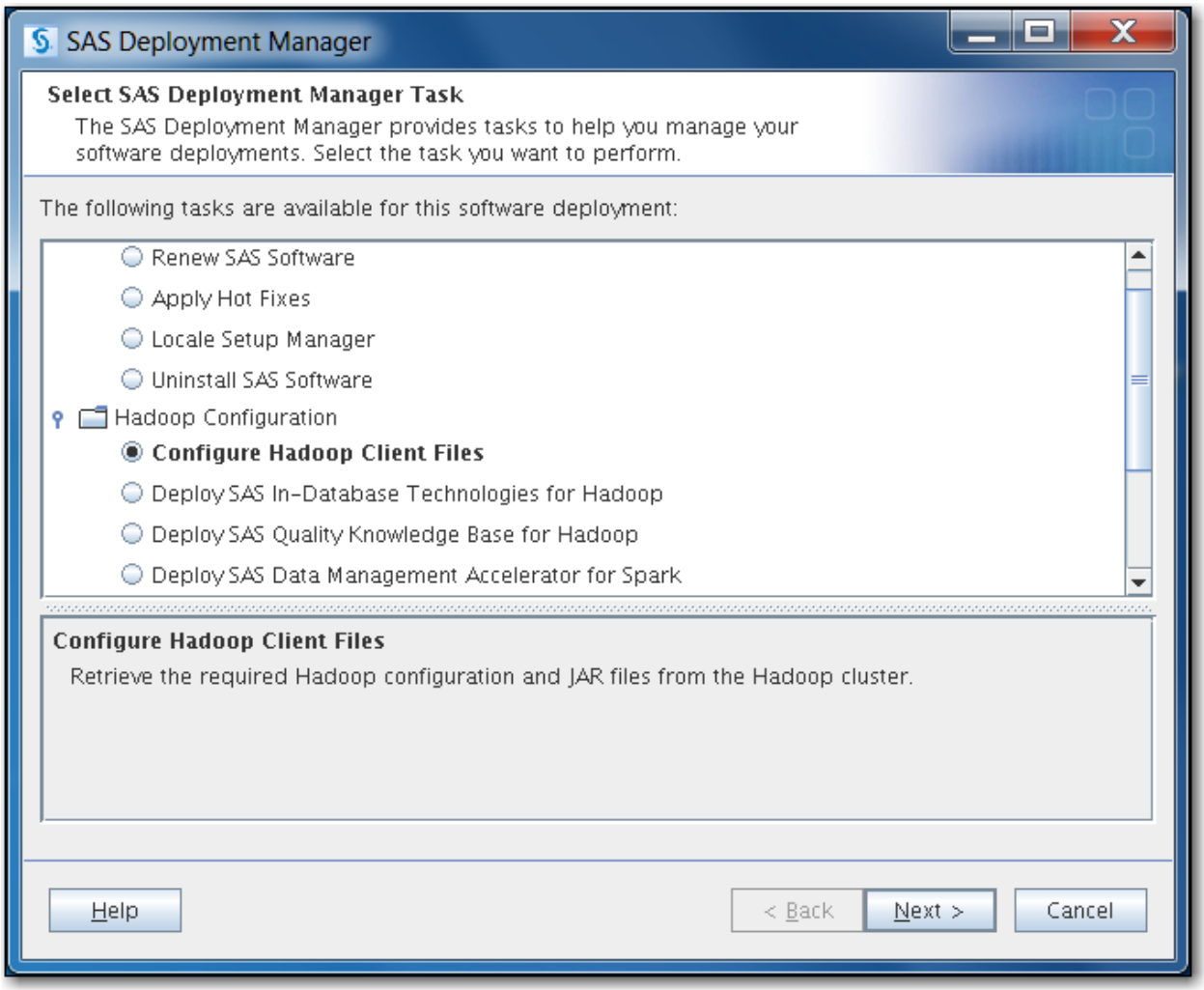
2 Start SAS Deployment Manager by running sasdm.exe for Windows or sasdm.sh for UNIX. The SAS Deployment Manager script is located in the `/SASHome/SASDeploymentManager/9.4` directory.

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

The **Choose Language** page opens.

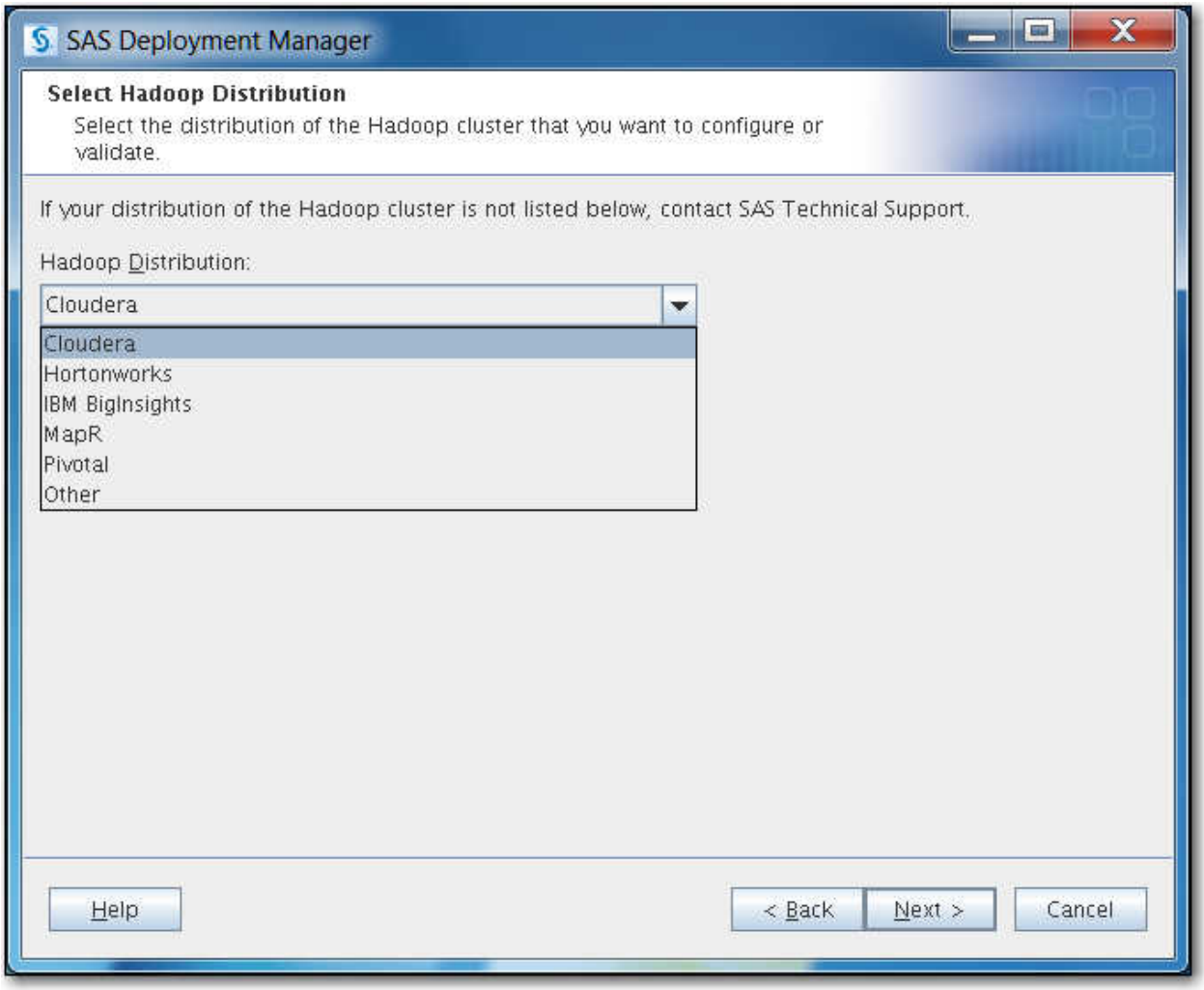
3 Select the language that you want to use to perform the configuration of your software.

Click **OK**. The **Select SAS Deployment Manager Task** page opens.



4 Under **Hadoop Configuration**, select **Configure Hadoop Client Files**.

Click **Next**. The **Select Hadoop Distribution** page opens.

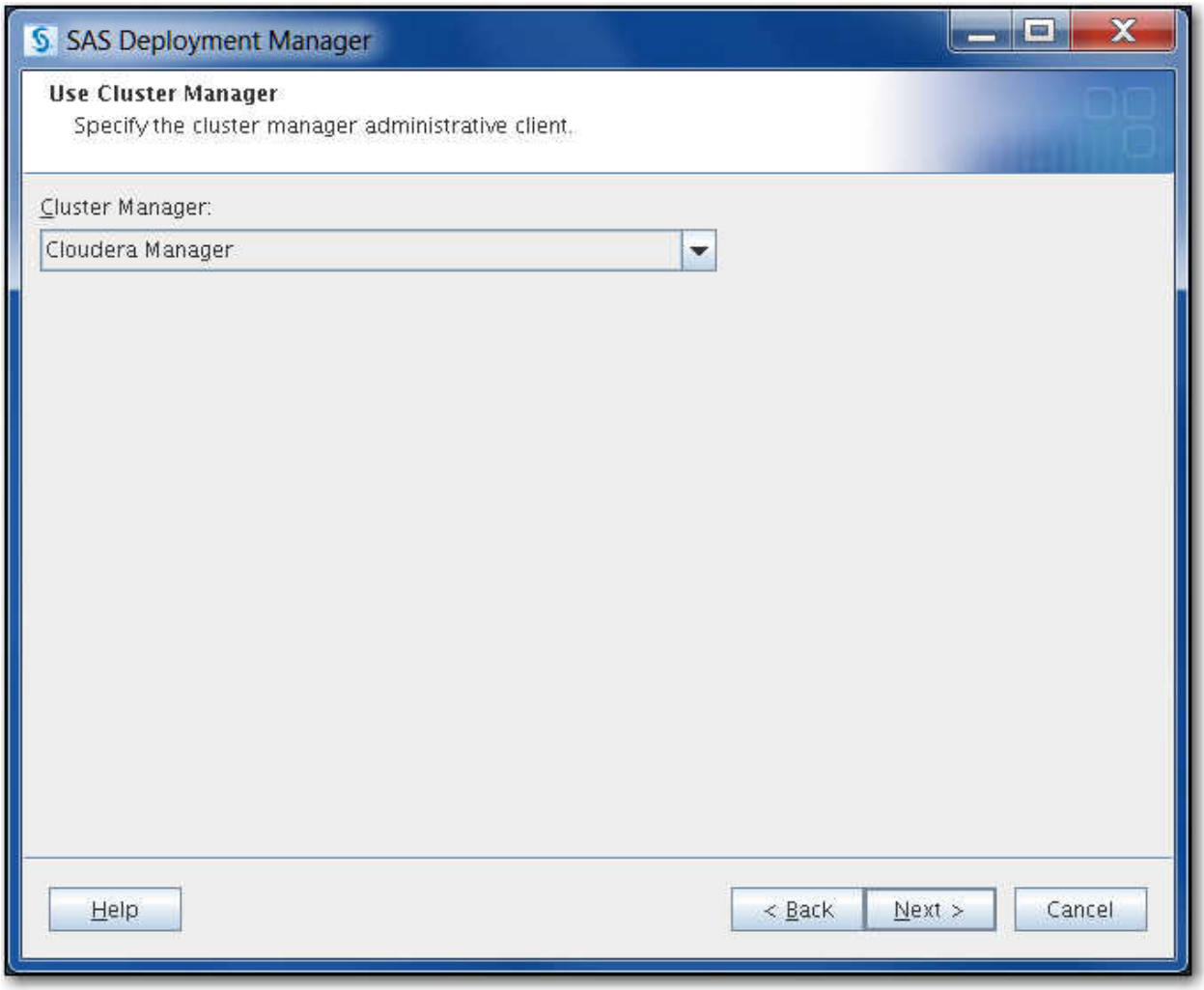


5 From the drop-down menu, select the distribution of Hadoop that you are using. (If your distribution is not listed, exit SAS Deployment Manager and contact SAS Technical Support.)

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 distribution has an administrative client such as Cloudera Manager or Ambari, the **Use Cluster Manager** page opens. Continue with Step 7 on page 22.



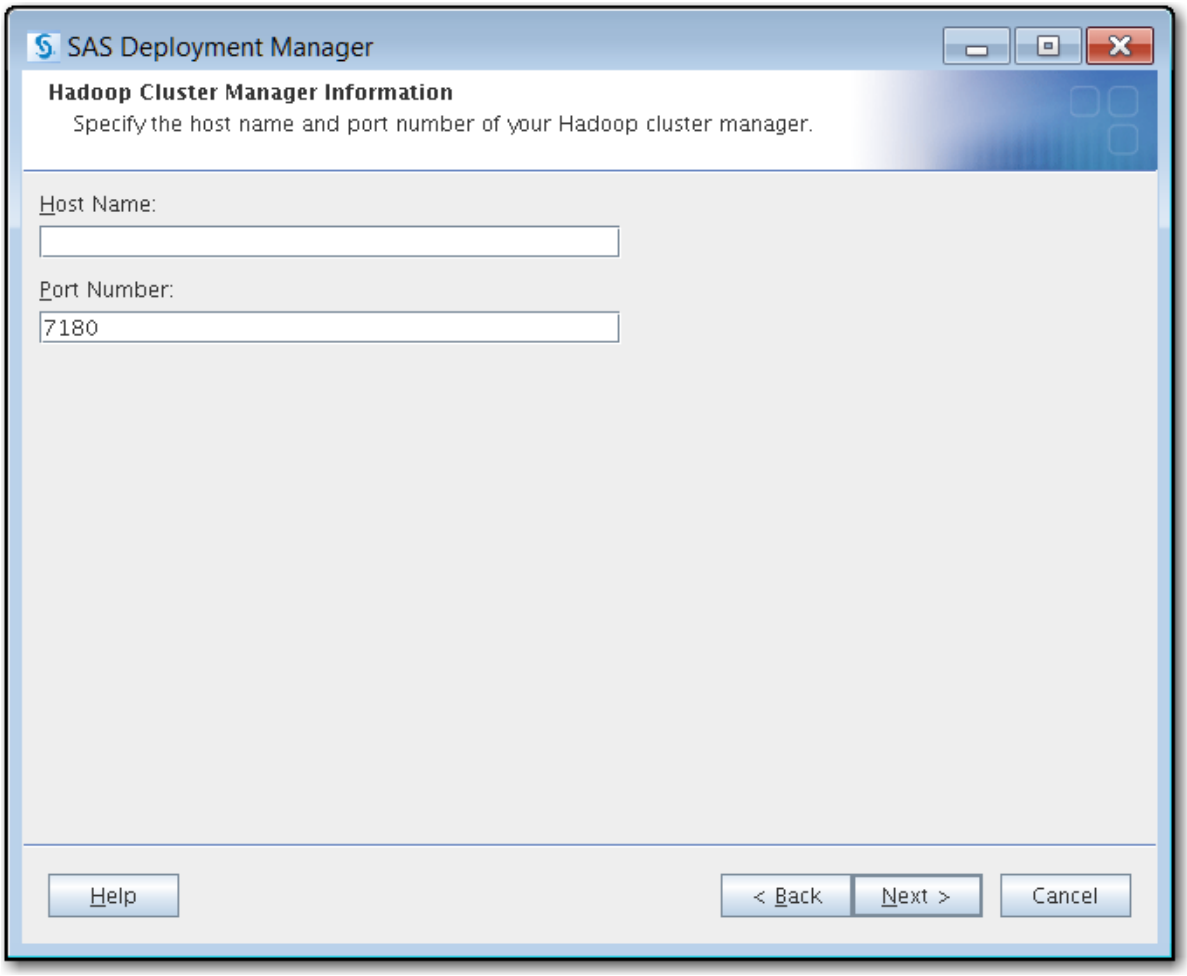
If your distribution does not have an administrative client, the **Hadoop Cluster Service Information and SSH Credentials** page opens. Skip to Step 10 on page 25.

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.

Click **Next**.

If you selected a cluster manager, the **Hadoop Cluster Manager Information** page opens. Continue with Step 7 on page 22.



If you selected **None**, the **Hadoop Cluster Service Information and SSH Credentials** page opens. Skip to Step 10 on page 25.

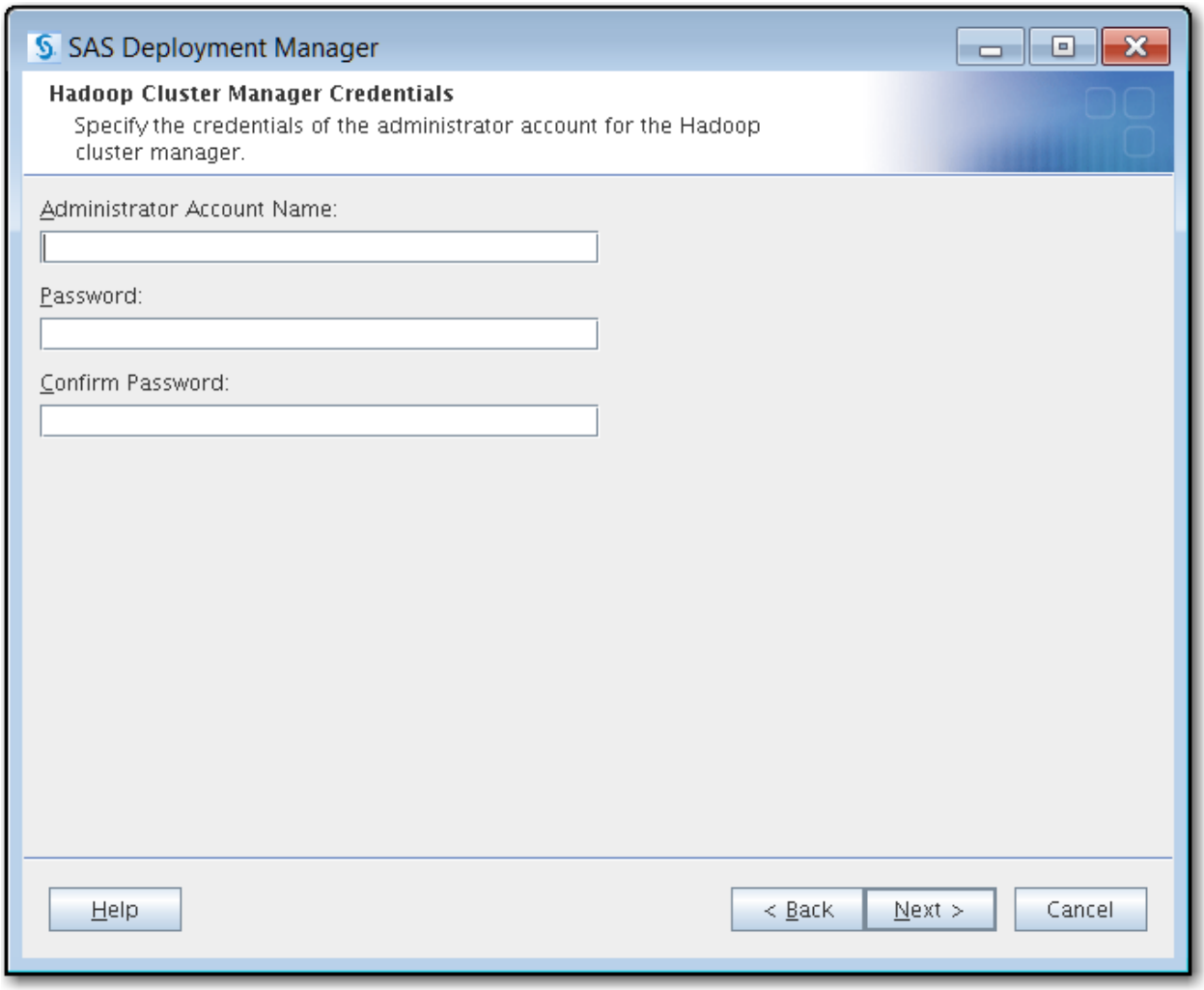
7 Enter the host name and port number for your Hadoop cluster.

For Cloudera, enter the location where Cloudera Manager is running. For Hortonworks, enter the location where the Ambari server is running.

The port number is set to the appropriate default after Cloudera or Hortonworks is selected.

Note: The host name must be a fully qualified domain name. The port number must be valid, and the cluster manager must be listening.

Click **Next**. The **Hadoop Cluster Manager Credentials** page opens.

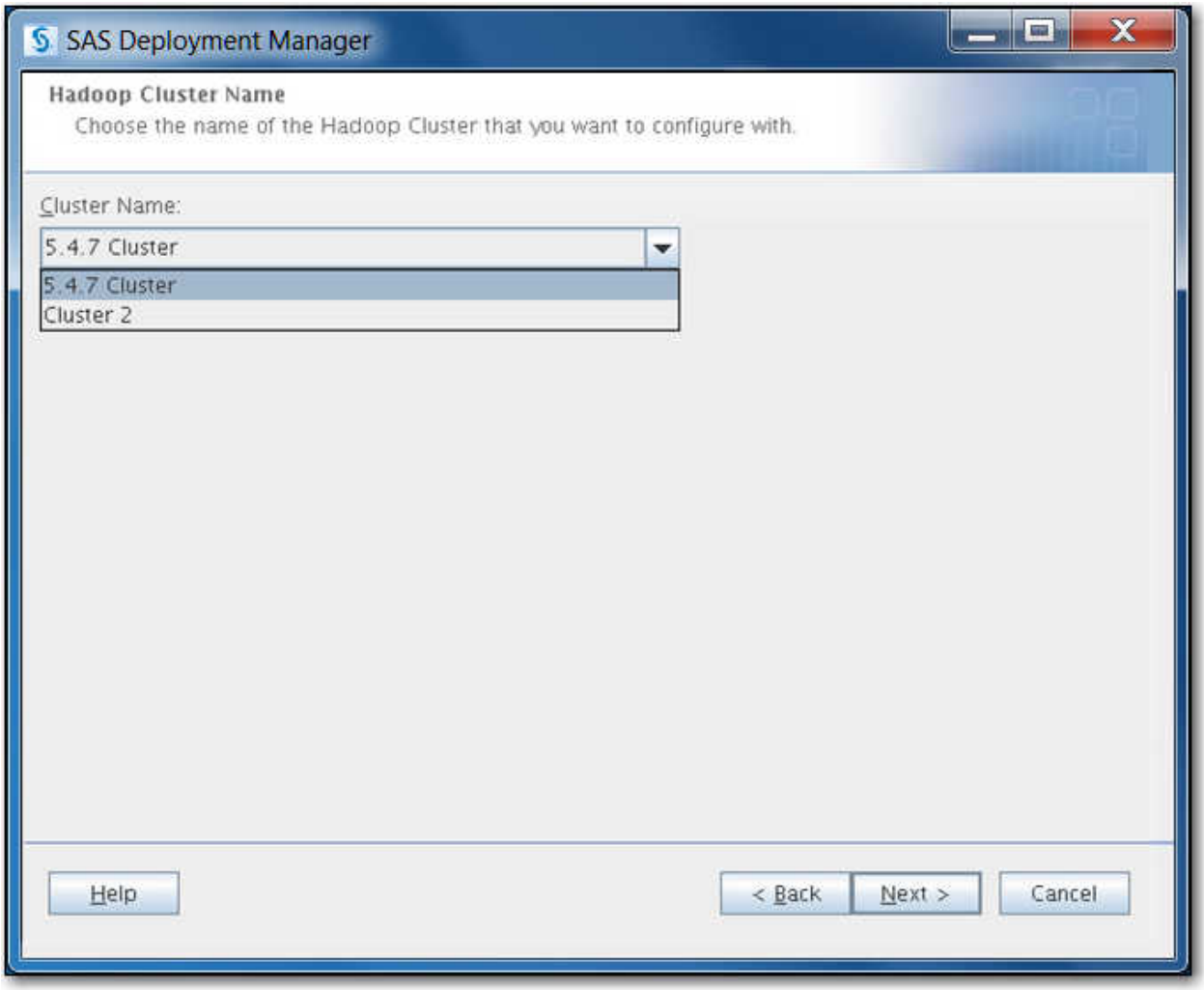


8 Enter the Cloudera Manager or Ambari administrator account name and password. If your distribution is not listed, exit SAS Deployment Manager and contact SAS Technical Support.

Note: Using the credentials of the administrator account to query the Hadoop cluster and to find the Hive node eliminates guesswork and removes the chance of a configuration error.

Click **Next**.

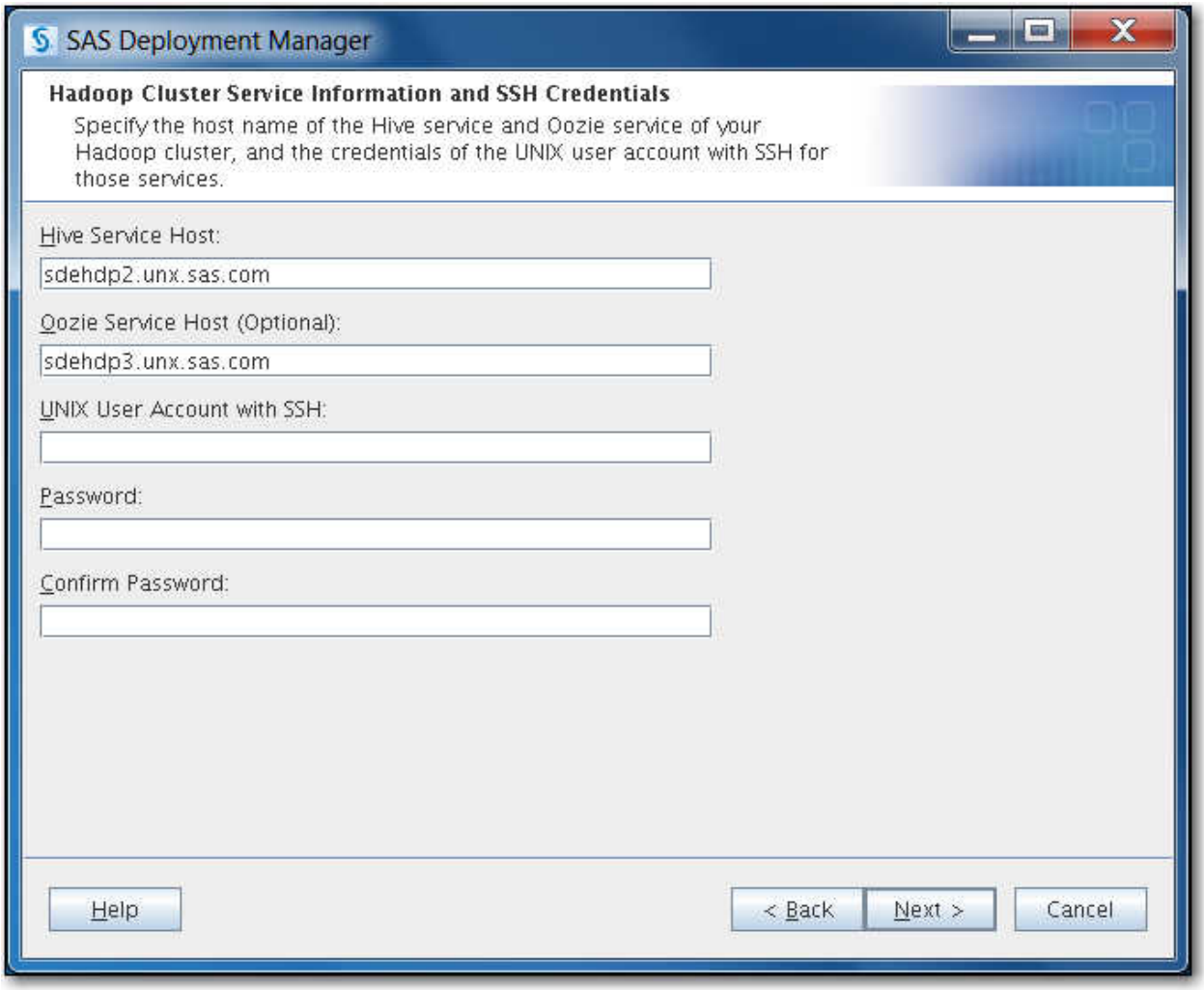
If you are using Cloudera Manager and multiple Hadoop clusters are being managed by the same cluster manager, the **Hadoop Cluster Name** page opens. Continue with Step 9 on page 24.



Otherwise, the **Hadoop Cluster Service Information and SSH Credentials** page opens. Skip to Step 10 on page 25.

9 Select the cluster from the drop-down list.

Click **Next**. The **Hadoop Cluster Service Information and SSH Credentials** page opens.



10 Enter the following information:

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

  Note: The Oozie service host name is optional. However, if your SAS software (for example, SAS Data Loader for Hadoop) uses Oozie, you need to enter the Oozie service host name so that the correct JAR files and configuration files are collected.

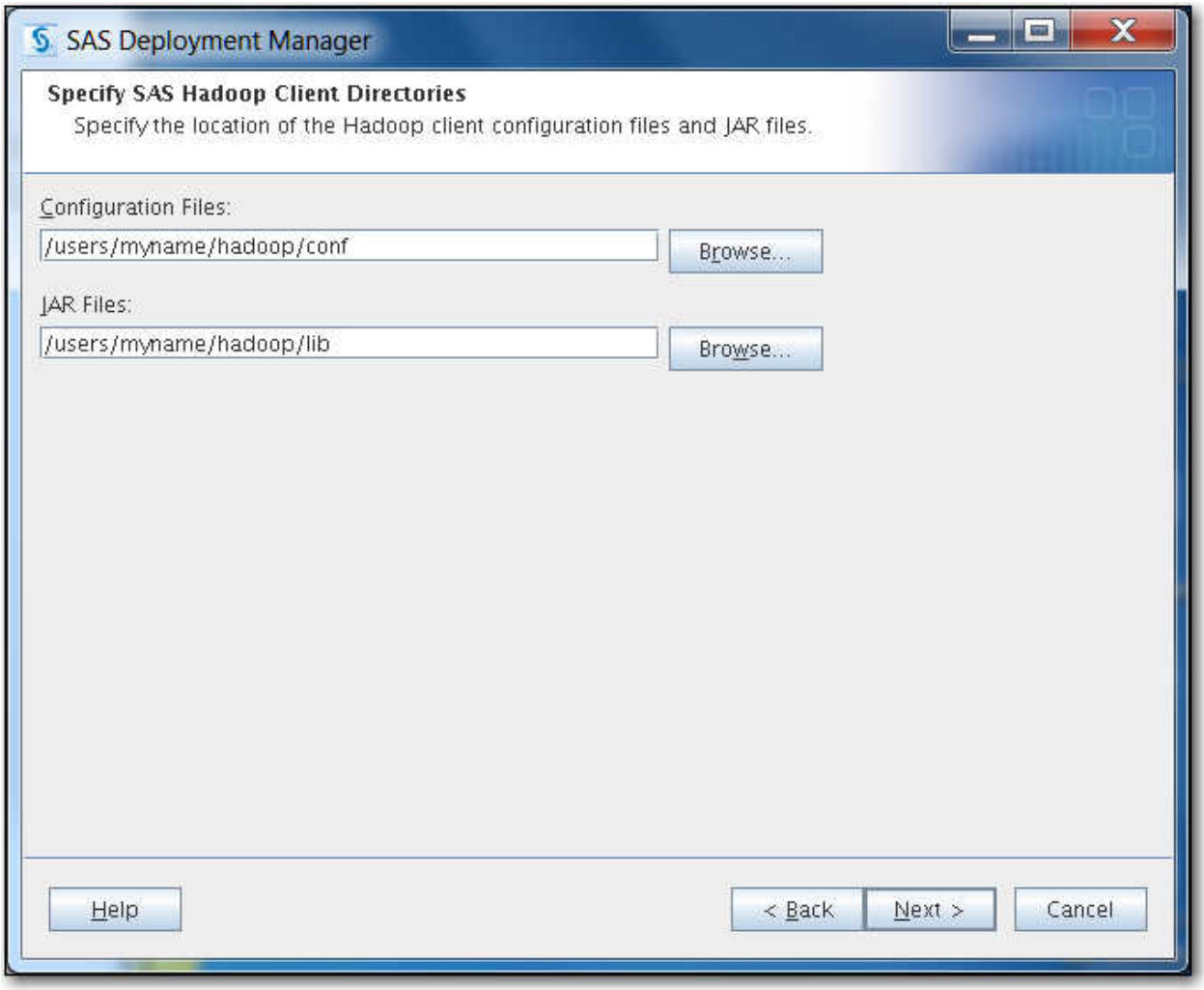
- The SSH-enabled administrator account name and password that have access to both the Hive and Oozie services. This information is required to move and copy files to and from hosts.

Note: When you provide SSH credentials, a directory named **`/user/ssh-account/test1`** is created to validate the ability to create an HDFS directory. For most Hadoop distributions, SAS Deployment Manager deletes this directory automatically. However, if you are using Hortonworks 1.3.2, this directory is not automatically deleted. If you need to run SAS Deployment Manager a second time to configure the Hadoop client files on this cluster (for example, a hot fix), an error occurs. You must manually remove the **`/user/ssh-account/test1`** directory by using the following command:

```
hadoop fs -rmr -skipTrash /user/ssh-account/test1
```

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

Click **Next**. The **Specify SAS Hadoop Client Directories** page opens.



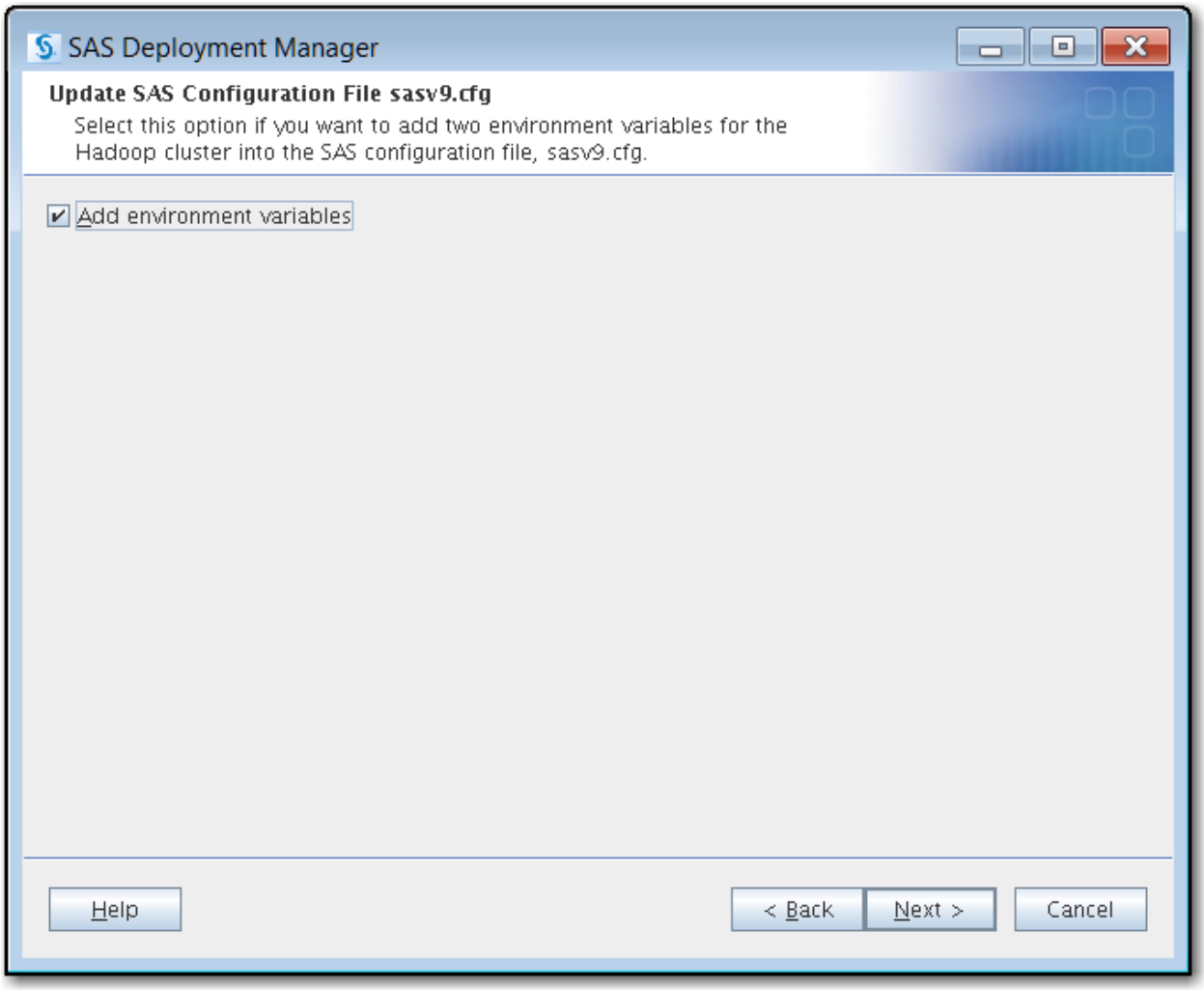
11 Specify the locations of the configuration files and JAR files for the Hadoop client.

Note: The default value is a path outside the configuration home because SAS/ACCESS does not have to be a planned deployment. Therefore, the configuration directories do not exist. 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.

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.

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.

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



12 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:

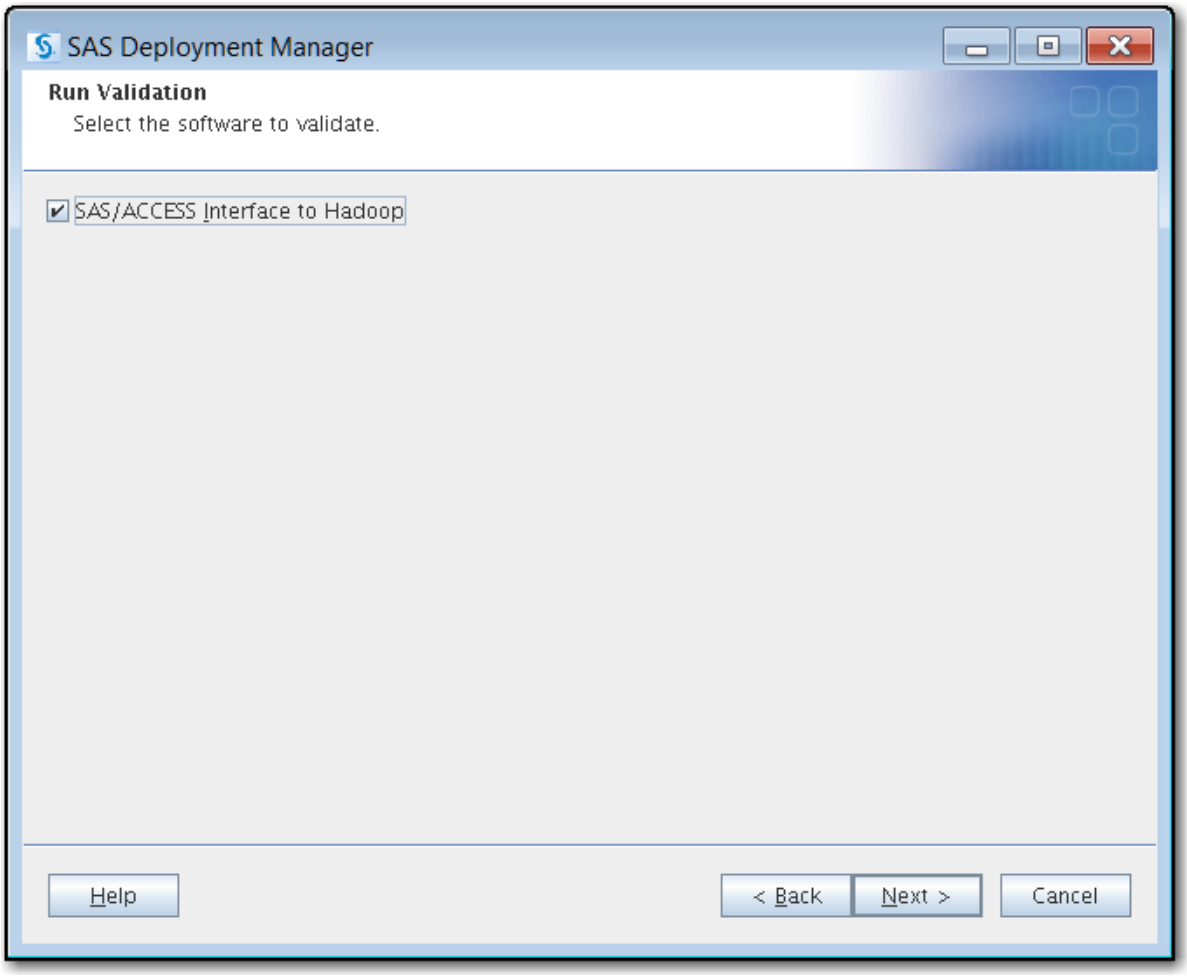
- SAS_HADOOP_CONFIG_PATH

  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**. The **Run Validation** page opens.

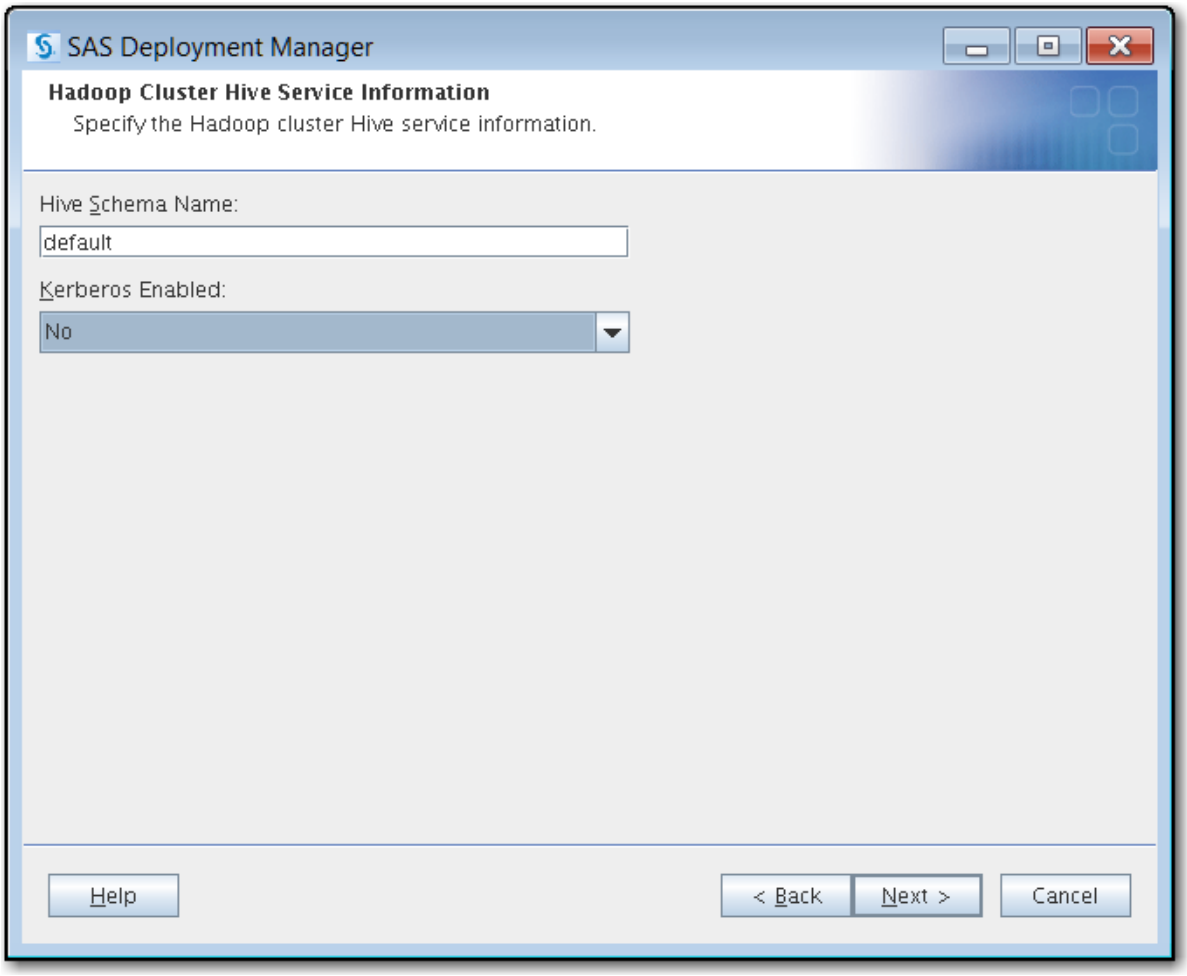


13 Validate the configuration of SAS/ACCESS Interface to Hadoop.

Note: If you are using Advanced Encryption Standard (AES) encryption with Kerberos, you must manually add the Java Cryptography Extension local_policy.jar file in every place 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. For more information, see "Kerberos Security" on page 36.

If there are problems with the validation, an error message appears. You can check the log files for the cause of the error. By default, log files are saved under the `/install/home` directory.

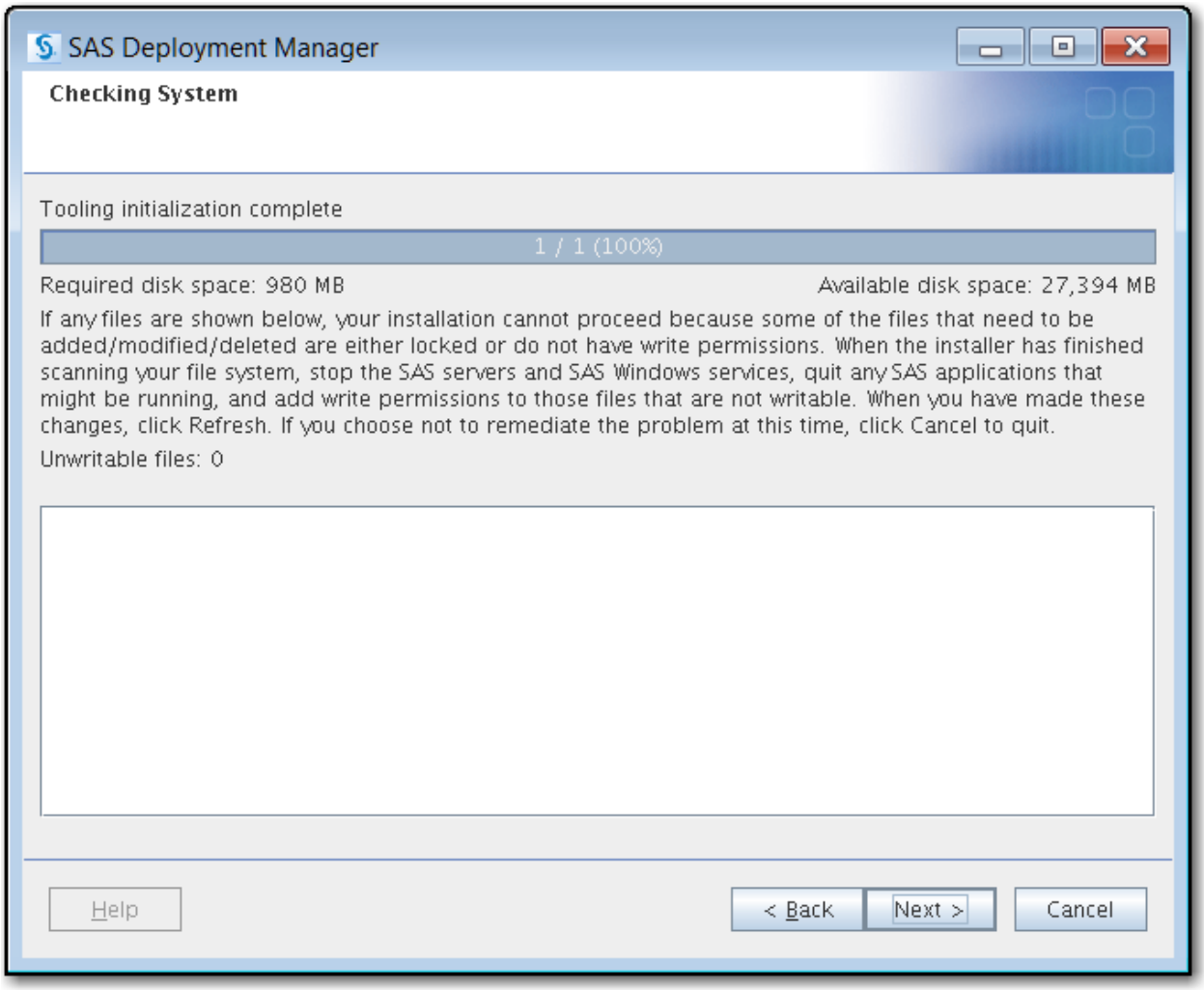
Click **Next**. The **Hadoop Cluster Hive Service Information** page appears.



14 Enter the schema name for the cluster's Hive service and select whether Kerberos is enabled on the cluster.

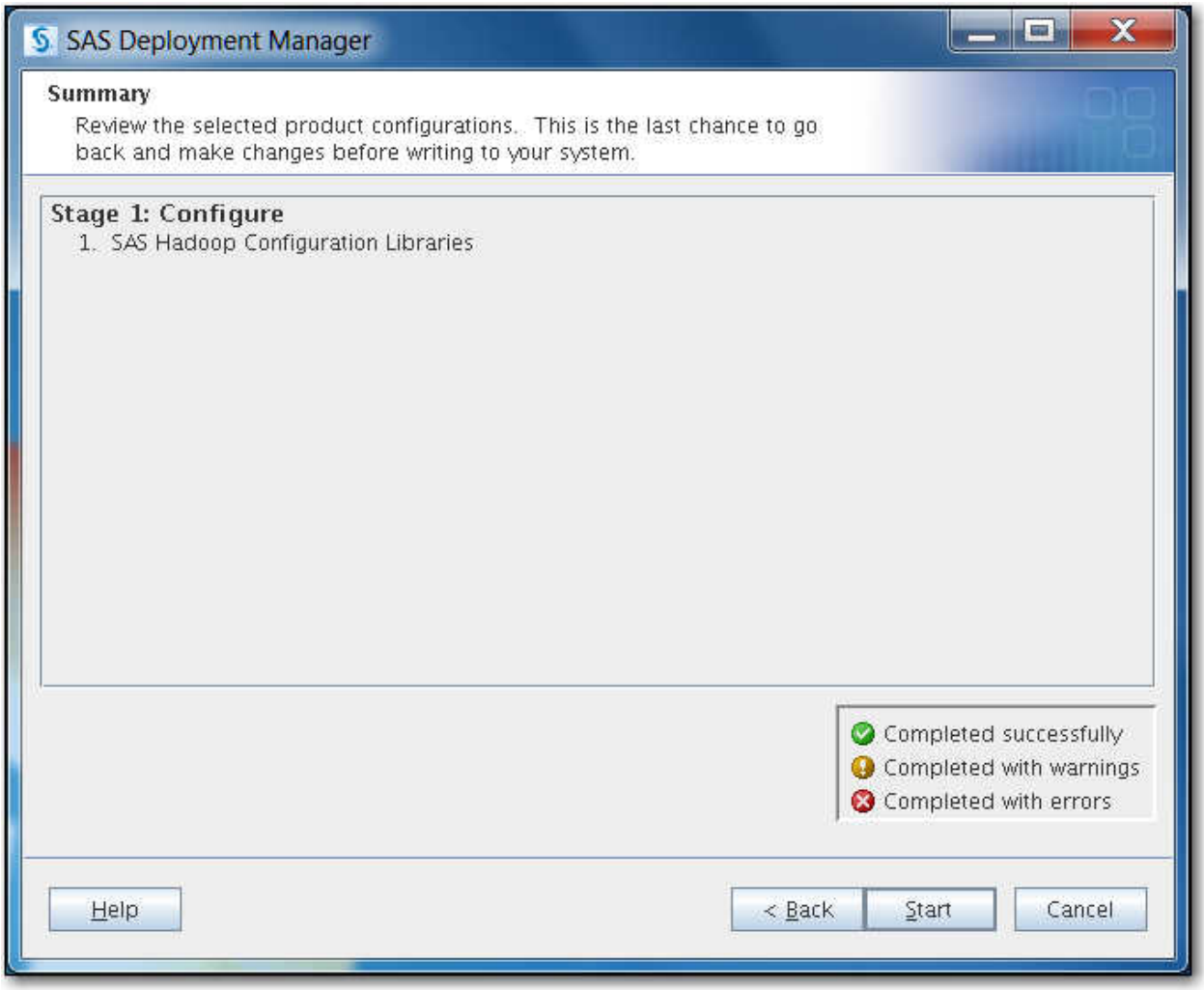
A valid Kerberos ticket must be available on the client machine and Hive service. If a ticket is not available, you must go out to the client machine, cluster, or both and obtain the Kerberos ticket. When the ticket is obtained, you can resume the deployment using SAS Deployment Manager.

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.



15 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.

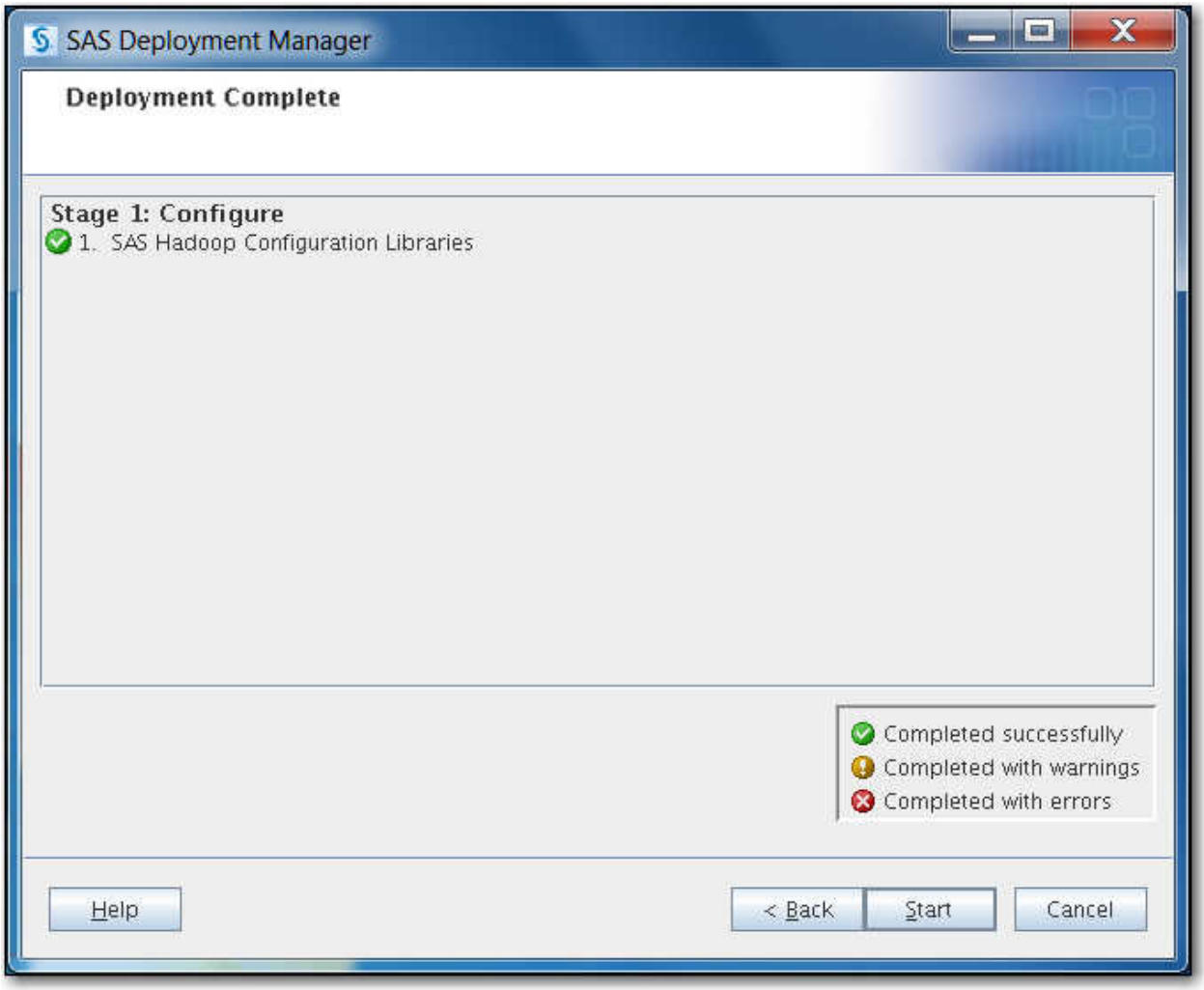
Click **Next**. The **Summary** page opens.



16 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 page title 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.

17 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
```

Additional Configuration for MapR

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

- In the third maintenance release for SAS 9.4, using SAS Deployment Manager automatically copies the requisite JAR files. SAS Deployment Manager enables you to define the SAS_HADOOP_JAR_PATH environment variable to point to those files and save the environment variable location in the sasv9.cfg file. This action eliminates the need to manually configure the MapR client JAR files and set the environment variable to point to them.
- 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 located in the directory that contains the configuration files. This directory was specified when you used SAS Deployment Manager.

  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.security.auth.login.config=
     C:\SDM-config-file-directory\conf\mapr.login.conf)
  ```

  Note: The MapR 64-bit library must be selected. The MapR 32-bit library produces undesirable results.
- MapR requires this JRE option for a Kerberos connection:

  ```
  -Dhadoop.login=kerberos
  ```

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

Note: In the third maintenance release for SAS 9.4, SAS no longer supports the 32-bit Windows client.

Additional Configuration for IBM BigInsights 3.0

The hive-site.xml configuration file is not automatically copied to the SAS client when you run SAS Deployment Manager.

Copy the configuration file to the Hadoop client configuration directory that was specified in Step 11 on page 26.

Supporting Multiple Hadoop Versions and Upgrading Your Hadoop Version

The version of the JAR files in the SAS_HADOOP_JAR_PATH directory must match the version of the JAR files on 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 files for each Hadoop version.

The SAS_HADOOP_JAR_PATH directory must be dynamically set depending on which Hadoop server a SAS job or SAS session connects to. One way to dynamically set SAS_HADOOP_JAR_PATH is to create a wrapper script that is associated with each Hadoop version. SAS is invoked via a wrapper script that sets SAS_HADOOP_JAR_PATH appropriately to pick up the JAR files that match the target Hadoop server.

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

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 37 and "Using SAS Deployment Manager to Make Required Hadoop JAR and Configuration Files Available to the SAS Client Machine" on page 18.

- By configuring a required set of Hadoop JAR files. The JAR files must be located in one 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 "Using SAS Deployment Manager to Make Required Hadoop JAR and Configuration Files Available to the SAS Client Machine" on page 18.

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 37.

- 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 must manually add the Java Cryptography Extension local_policy.jar file in every place that JAVA Home resides on the cluster. If you are outside the United States, you must also manually add the US_export_policy.jar file. The addition of these files is governed by the United States import control restrictions.

  These two JAR files also need to replace the existing local_policy.jar and US_export_policy.jar files in the SAS JRE location that is the ***SASHome*/SASPrivateJavaRuntimeEnvironment/9.4/jre/lib/security/** directory. It is recommended to back up the existing local_policy.jar and US_export_policy.jar files first in case they need to be restored.

  These files can be obtained from the IBM or Oracle website.

- 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:

  ```
  KRB5CCNAME="/tmp/krb5cc_'id -u'"
  export KRB5CCNAME
  ```

- 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:\winnt\krb5.ini
  ```

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

  ```
  Dhadoop.login=kerberos
  ```

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

JDBC Read Security

SAS/ACCESS can access Hadoop data through a JDBC connection to a HiveServer or 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:

```
set SAS_HADOOP_RESTFUL 1      /* SAS command line */
```

or

```
-set SAS_HADOOP_RESTFUL 1     /* DOS prompt */
```

or

```
export SAS_HADOOP_RESTFUL=1   /* UNIX */
```

For more information, see “SAS_HADOOP_RESTFUL Environment Variable” on page 78.

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

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

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

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

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

For more information about the configuration files, see “Configuring Hadoop JAR and Configuration Files” on page 17.

Working with Hive and HiveServer2

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 simple scenario demonstrates how to access HDFS files from Hive by using the Hive CLI. For more information, perform a web search for “Hive CLI” and locate the appropriate Apache documentation.

Assume there are HDFS files weblog1.txt and weblog2.txt with data lines that contain in order, a date field, a text integer field, and a string field. The fields are comma-delimited and lines \n terminated.

```
$ hadoop fs -ls /user/hadoop/web_data
Found 2 items
-rw-r--r-- 3 hadoop [owner] [size/date]
/user/hadoop/web_data/weblog1.txt
-rw-r--r-- 3 hadoop [owner] [size/date]
/user/hadoop/web_data/weblog2.txt
```

To make these HDFS files visible to Hive:

1. Terminate the Hive service if it is running. Next, at a Linux prompt, execute the Hive CLI:

   ```
   $ hive
   ```

2. At the Hive command prompt, make the weblogs visible to Hive:

   ```
   hive> CREATE EXTERNAL TABLE weblogs (extract_date STRING,
   extract_type INT, webdata STRING) ROW FORMAT DELIMITED FIELDS
   TERMINATED BY ',' STORED AS TEXTFILE LOCATION
   '/user/hadoop/web_data';
   ```

3. At the Hive command prompt, test that weblog1.txt is now accessible to Hive:

   ```
   hive> SELECT * FROM weblogs LIMIT 1;
   ```

4. If the SELECT statement works, quit the Hive CLI and start the Hive Service on default port 10000.

   For example, if you start the Hive service on node **`hadoop_cluster`**, a test access from SAS would be as follows:

   ```
   libname hdplib hadoop server=hadoop_cluster user=hadoop_usr
   password=hadoop_usr_pwd;
   data work.weblogs;
   set hdplib.weblogs(obs=1);
   put _all_;
   run;
   ```

This is a complete but intentionally simple scenario intended for new Hive users. It is not representative of a mature Hive environment because the default Hive schema is used implicitly and the Hive default Derby metadata store might be in use. Consult Hadoop and Hive documentation such as Apache Hive to begin to explore Hive in detail. For more information about how SAS/ACCESS interacts with Hive, see *SAS/ACCESS for Relational Databases: Reference*.

Running the Hive or HiveServer2 Service on Your Hadoop Server

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

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

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

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

Writing Data to Hive: HDFS /tmp and the "Sticky Bit"

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

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

```
$ hadoop fs -ls /
drwxrwxrwt - hdfs hdfs 0 2013-01-21 13:25 /tmp
drwxr-xr-x - hdfs supergroup 0 2013-01-21 11:46 /user
```

Validating Your SAS/ACCESS to Hadoop Connection

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

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

Sample libref connection to HiveServer2 (default):

```
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);
```

Sample libref connection to Hive:

```
libname hdplib hadoop server=hadoop user=hadoop_usr password=hadoop_usr_pwd
   subprotocol=hive;
```

```
NOTE: Libref HDPLIB was successfully assigned as follows:
Engine: HADOOP
Physical Name: jdbc:hive://hadoop:10000/default
```

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

Note: HiveServer1 has been removed with the release of Hive 1.0.0 and in a future release, SAS/ACCESS Interface to Hadoop will no longer support a connection to HiveServer1. For more information, see Delete Hiveserver1.

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*.

5

Configuring SPD Engine

Overview of Steps to Configure 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 SPD Engine” on page 44.

2 Make Hadoop JAR files available to the SAS client machine.

   For more information, see “Configuring Hadoop JAR Files” on page 44.

3 Make Hadoop configuration files available to the SAS client machine.

   For more information, see “Making Hadoop Cluster Configuration Files Available to the SAS Client Machine” on page 47.

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 49.

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

   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.

Configuring Hadoop JAR Files

Configuring Hadoop JAR Files Using SAS Deployment Manager

If you license SAS/ACCESS Interface to Hadoop, you can use SAS Deployment Manager to make required Hadoop JAR and configuration files available to the SAS client machine for the SPD Engine. For more information about using SAS Deployment Manager for SAS/ACCESS Interface to Hadoop, see “Configuring Hadoop JAR and Configuration Files” on page 17.

Note: The Apache Curator JAR files that are required for the SPD Engine distributed locking are not made available by SAS Deployment Manager. If you want distributed locking that is provided by the SPD Engine, you must make the following Apache Curator JAR files available to the SAS client machine by following the instructions in “Making Required Hadoop JAR Files Available to the SAS Client Machine” on page 45.

- curator-client
- curator-framework
- curator-recipes

If you do not license SAS/ACCESS Interface to Hadoop, in order to use the SPD Engine, you must follow the instructions in “Making Required Hadoop JAR Files Available to the SAS Client Machine” on page 45 to use the SPD Engine.

Making Required Hadoop JAR Files Available to the SAS Client Machine

To connect the SPD Engine to a Hadoop server, the required Hadoop JAR files must be available to the SAS client machine. To make the required JAR files available, you must define the SAS_HADOOP_JAR_PATH environment variable to set the location of the JAR files:

1. Create a directory that is accessible to the SAS client machine.
2. From the Hadoop cluster, copy the required JAR files for the particular Hadoop distribution to the directory that was created in step 1.

   For example, here are the required JAR files for CDH 4.7. The set is different for other Hadoop distributions.

   commons-beanutils
   commons-cli
   commons-collections
   commons-configuration
   commons-lang
   commons-logging
   guava
   hadoop-auth
   hadoop-common
   hadoop-core
   hadoop-hdfs
   hive-exec
   hive-jdbc
   hive-metastore
   hive-service
   jackson-core-asl
   jackson-jaxrs
   jackson-mapper-asl
   jackson-xc
   libfb303
   pig
   protobuf-java
   slf4j-api
   slf4j-log4j12

   Appendix 1, “Hadoop JAR Files,” on page 51 lists the required JAR files for each Hadoop distribution.

   Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. You might need assistance from your Hadoop administrator to locate the appropriate JAR files.

Additional JAR files might be needed due to JAR file interdependencies and your Hadoop distribution. For more information, see "Supporting Multiple Hadoop Versions and Upgrading Hadoop Version" on page 46.

3 If you want distributed locking, which is provided by the SPD Engine, you must also make the following Apache Curator JAR files available to the SAS client machine. Copy these JAR files for the particular Hadoop distribution to the directory that was created in step 1.

- curator-client
- curator-framework
- curator-recipes

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

For example, if the JAR files are copied to the location **`C:\third_party\Hadoop\jars`**, 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_JAR_PATH "C:\third_party\Hadoop\jars" /* SAS command line */
```

or

```
set SAS_HADOOP_JAR_PATH "C:\third_party\Hadoop\jars" /* DOS prompt */
```

or

```
export SAS_HADOOP_JAR_PATH="/third_party/hadoop/jars" /* SAS command UNIX */
```

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;C:\MyHadoopJars";
```

For more information, see "SAS_HADOOP_JAR_PATH Environment Variable" on page 76.

Note: A SAS_HADOOP_JAR_PATH directory must not have multiple versions of a Hadoop JAR file. Multiple versions can cause unpredictable behavior when SAS runs. For more information, see "Supporting Multiple Hadoop Versions and Upgrading Hadoop Version" on page 46.

Supporting Multiple Hadoop Versions and Upgrading Hadoop Version

The JAR files in the SAS_HADOOP_JAR_PATH directory 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 files for each Hadoop version.

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

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

Additional Requirements for MapR-Based Hadoop 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 normally 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.

Making Hadoop Cluster Configuration Files Available to the SAS Client Machine

Configuring Hadoop Cluster Configuration Files Using SAS Deployment Manager

If you license SAS/ACCESS Interface to Hadoop, you can use SAS Deployment Manager to make required Hadoop JAR and configuration files available to the SAS client machine for the SPD Engine. For more information about using SAS Deployment Manager for SAS/ACCESS Interface to Hadoop, see “Configuring Hadoop JAR and Configuration Files” on page 17.

If you do not license SAS/ACCESS Interface to Hadoop, you must copy the configuration files to a physical location that is accessible to the SAS client machine and then set the SAS environment variable SAS_HADOOP_CONFIG_PATH. See “Copying the Hadoop Cluster Configuration Files” on page 47.

Copying the Hadoop Cluster Configuration Files

To connect to a Hadoop server, you must make the configuration files available to the SAS client machine:

1 Create a directory that is accessible to the SAS client machine.

2 From the specific Hadoop cluster, copy these configuration files to the directory created in step 1.

core-site.xml
hdfs-site.xml
hive-site.xml
mapred-site.xml
yarn-site.xml

Note: For a MapReduce 1 cluster, the core-site.xml, hdfs-site.xml, and mapred-site.xml files are needed. For a MapReduce 2 cluster, the yarn-site.xml file is needed.

3 Define the SAS environment variable named SAS_HADOOP_CONFIG_PATH. Set it 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"
```

For more information, see "SAS_HADOOP_CONFIG_PATH Environment Variable" on page 75.

Kerberos Security

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

- For the SPD Engine on AIX, add this line to the JREOPTIONS in the SAS configuration file:

```
-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 the following 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, the following JRE option must be set:

  ```
  Dhadoop.login=kerberos
  ```

  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.

```
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="\\sashq\root\u\sasabc\hadoop
\ConfigDirectory\cdh45p1";
18   options set=SAS_HADOOP_JAR_PATH="\\sashq\root\u\sasabc\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

Hadoop JAR Files

Cloudera JAR Files

Cloudera 4.7.x JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop. SAS Deployment Manager automatically copies the requisite JAR and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

guava-11.0.2.jar
hadoop-auth-2.0.0-cdh4.7.0.jar
hadoop-common-2.0.0-cdh4.7.0.jar
hadoop-core-2.0.0-mr1-cdh4.7.0.jar
hadoop-hdfs-2.0.0-cdh4.7.0.jar
hive-exec-0.10.0-cdh4.7.0.jar
hive-jdbc-0.10.0-cdh4.7.0.jar
hive-metastore-0.10.0-cdh4.7.0.jar
hive-service-0.10.0-cdh4.7.0.jar
libfb303-0.9.0.jar
pig-0.11.0-cdh4.7.0-withouthadoop.jar
protobuf-java-2.4.0a.jar

For the SPD Engine on Cloudera 4.7, include these JAR files as well:

commons-beanutils-1.7.0.jar
commons-cli-1.2.jar
commons-collections-3.2.1.jar
commons-configuration-1.6.jar
commons-lang-2.5.jar
commons-logging-1.1.1.jar
jackson-core-asl-1.8.8.jar
jackson-jaxrs-1.8.8.jar
jackson-mapper-asl-1.8.8.jar
jackson-xc-1.8.8.jar
slf4j-api-1.6.1.jar
slf4j-log4j12-1.6.1.jar

Cloudera 5.0 JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop. SAS Deployment Manager automatically copies the requisite JAR and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

guava-12.0.1.jar
hadoop-auth-2.3.0-cdh5.0.0.jar
hadoop-client-2.3.0-mr1-cdh5.0.0.jar
hadoop-common-2.3.0-cdh5.0.0.jar
hadoop-core-2.3.0-mr1-cdh5.0.0.jar
hadoop-hdfs-2.3.0-cdh5.0.0.jar
hive-exec-0.12.0-cdh5.0.0.jar
hive-jdbc-0.12.0-cdh5.0.0.jar
hive-metastore-0.12.0-cdh5.0.0.jar
hive-service-0.12.0-cdh5.0.0.jar
httpclient-4.2.5.jar
httpcore-4.2.5.jar
libfb303-0.9.0.jar
pig-0.12.0-cdh5.0.0-withouthadoop.jar
protobuf-java-2.5.0.jar

For the SPD Engine on Cloudera 5, include these JAR files as well:

commons-beanutils-1.7.0.jar
commons-cli-1.2.jar
commons-collections-3.2.1.jar
commons-configuration-1.6.jar
commons-lang-2.6.jar
commons-logging-1.1.3.jar
jackson-core-asl-1.8.8.jar
jackson-jaxrs-1.8.8.jar
jackson-mapper-asl-1.8.8.jar
jackson-xc-1.8.8.jar
slf4j-api-1.7.5.jar
slf4j-log4j12.jar

Cloudera 5.2 JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop. SAS Deployment Manager automatically copies the requisite JAR

and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

guava-12.0.1.jar
hadoop-auth-2.5.0-cdh5.2.0.jar
hadoop-common-2.5.0-cdh5.2.0.jar
hadoop-hdfs-2.5.0-cdh5.2.0.jar
hive-exec-0.13.1-cdh5.2.0.jar
hive-jdbc-0.13.1-cdh5.2.0.jar
hive-metastore-0.13.1-cdh5.2.0.jar
hive-service-0.13.1-cdh5.2.0.jar
httpclient-4.2.5.jar
httpcore-4.2.5.jar
libfb303-0.9.0.jar
pig-0.12.0-cdh5.2.0-withouthadoop.jar
protobuf-java-2.5.0.jar

For the SPD Engine on Cloudera 5, include these JAR files as well:

commons-beanutils-1.7.0.jar
commons-cli-1.2.jar
commons-collections-3.2.1.jar
commons-configuration-1.6.jar
commons-lang-2.6.jar
commons-logging-1.1.3.jar
jackson-core-asl-1.8.8.jar
jackson-jaxrs-1.8.8.jar
jackson-mapper-asl-1.8.8.jar
jackson-xc-1.8.8.jar
slf4j-api-1.7.5.jar
slf4j-log4j12.jar

Cloudera 5.4 JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator

for Hadoop. SAS Deployment Manager automatically copies the requisite JAR and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

guava-12.0.1.jar
hadoop-auth-2.6.0-cdh5.4.2.jar
hadoop-common-2.6.0-cdh5.4.2.jar
hadoop-core-2.6.0-mr1-cdh5.4.2.jar
hadoop-hdfs-2.6.0-cdh5.4.2.jar
hive-exec-1.1.0-cdh5.4.2.jar
hive-jdbc-1.1.0-cdh5.4.2-standalone.jar
hive-metastore-1.1.0-cdh5.4.2.jar
httpclient-4.2.5.jar
httpcore-4.2.5.jar
pig-0.12.0-cdh5.4.2.jar
protobuf-java-2.5.0.jar

For the SPD Engine on Cloudera 5, include these JAR files as well:

commons-beanutils-1.7.0.jar
commons-cli-1.2.jar
commons-collections-3.2.1.jar
commons-configuration-1.6.jar
commons-lang-2.6.jar
commons-logging-1.1.3.jar
jackson-core-asl-1.8.8.jar
jackson-jaxrs-1.8.8.jar
jackson-mapper-asl-1.8.8.jar
jackson-xc-1.8.8.jar
slf4j-api-1.7.5.jar
slf4j-log4j12.jar

Hortonworks JAR Files

Hortonworks HDP 1.3.2 JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly

recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop. SAS Deployment Manager automatically copies the requisite JAR and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

guava-11.0.2.jar
hadoop-core-1.2.0.1.3.2.0-111.jar
hive-exec-0.11.0.1.3.2.0-111.jar
hive-jdbc-0.11.0.1.3.2.0-111.jar
hive-metastore-0.11.0.1.3.2.0-111.jar
hive-service-0.11.0.1.3.2.0-111.jar
libfb303-0.9.0.jar
pig-0.11.1.1.3.2.0-111.jar
pig-0.11.1.1.3.2.0-111-core.jar
protobuf-java-2.4.1.jar

Hortonworks HDP 2.0.x JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop. SAS Deployment Manager automatically copies the requisite JAR and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

guava-12.0.1.jar
hadoop-auth-2.2.0.2.0.6.0-101.jar

hadoop-common-2.2.0.2.0.6.0-101.jar
hadoop-hdfs-2.2.0.2.0.6.0-101.jar
hive-exec-0.12.0.2.0.6.1-101.jar
hive-jdbc-0.12.0.2.0.6.1-101.jar
hive-metastore-0.12.0.2.0.6.1-101.jar
hive-service-0.12.0.2.0.6.1-101.jar
httpclient-4.2.5.jar
httpcore-4.2.5.jar
libfb303-0.9.0.jar
pig-0.12.0.2.0.6.1-101-withouthadoop.jar
protobuf-java-2.5.0.jar

For the SPD Engine on HDP 2.0, include these JAR files as well:

commons-beanutils-1.7.0.jar
commons-cli-1.2.jar
commons-collections-3.2.1.jar
commons-configuration-1.6.jar
commons-lang-2.5.jar
commons-logging-1.1.1.jar
jackson-core-asl-1.8.8.jar
jackson-jaxrs-1.8.8.jar
jackson-mapper-asl-1.8.8.jar
jackson-xc-1.8.8.jar
slf4j-api-1.7.5.jar
slf4j-log4j12-1.7.5.jar

Hortonworks HDP 2.1.x JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop. SAS Deployment Manager automatically copies the requisite JAR and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

automaton-1.11-8.jar

guava-11.0.2.jar
hadoop-auth-2.4.0.2.1.5.0-695.jar
hadoop-common-2.4.0.2.1.5.0-695.jar
hadoop-hdfs-2.4.0.2.1.5.0-695.jar
hive-exec-0.13.0.2.1.5.0-695.jar
hive-jdbc-0.13.0.2.1.5.0-695.jar
hive-metastore-0.13.0.2.1.5.0-695.jar
hive-service-0.13.0.2.1.5.0-695.jar
httpclient-4.2.5.jar
httpcore-4.2.5.jar
jline-0.9.94.jar
libfb303-0.9.0.jar
pig-0.12.1.2.1.5.0-695-withouthadoop.jar
protobuf-java-2.5.0.jar

For the SPD Engine on HDP 2.1, include these JAR files as well:

commons-beanutils-1.7.0.jar
commons-cli-1.2.jar
commons-collections-3.2.1.jar
commons-configuration-1.6.jar
commons-lang-2.6.jar
commons-logging-1.1.3.jar
jackson-core-asl-1.8.8.jar
jackson-jaxrs-1.8.8.jar
jackson-mapper-asl-1.8.8.jar
jackson-xc-1.8.8.jar
slf4j-api-1.7.5.jar
slf4j-log4j12-1.7.5.jar

Hortonworks HDP 2.2.x JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop. SAS Deployment Manager automatically copies the requisite JAR and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these

individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

automaton-1.11-8.jar
guava-11.0.2.jar
hadoop-auth-2.6.0.2.2.0.0-2041.jar
hadoop-common-2.6.0.2.2.0.0-2041.jar
hadoop-hdfs-2.6.0.2.2.0.0-2041.jar
hive-exec-0.14.0.2.2.0.0-2041.jar
hive-jdbc-0.14.0.2.2.0.0-2041.jar
hive-metastore-0.14.0.2.2.0.0-2041.jar
httpclient-4.2.5.jar
httpcore-4.2.5.jar
jline-0.9.94.jar
pig-0.14.0.2.2.0.0-2041-core-h2.jar
protobuf-java-2.5.0.jar

For the SPD Engine on HDP 2.1, include these JAR files as well:

commons-beanutils-1.7.0.jar
commons-cli-1.2.jar
commons-collections-3.2.1.jar
commons-configuration-1.6.jar
commons-lang-2.6.jar
commons-logging-1.1.3.jar
jackson-core-asl-1.9.13.jar
jackson-jaxrs-1.9.13.jar
jackson-mapper-asl-1.9.13.jar
jackson-xc-1.9.13.jar
slf4j-api-1.7.5.jar
slf4j-log4j12-1.7.5.jar

Hortonworks HDP 2.3.x JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop. SAS Deployment Manager automatically copies the requisite JAR and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology

directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

automaton-1.11-8.jar
guava-11.0.2.jar
hadoop-auth-2.7.1.2.3.0.0-2257.jar
hadoop-common-2.7.1.2.3.0.0-2257.jar
hadoop-hdfs-2.7.1.2.3.0.0-2257.jar
hive-exec-1.2.1.2.3.0.0-2557.jar
hive-jdbc-1.2.1.2.3.0.0-2557.jar
hive-metastore-1.2.1.2.3.0.0-2557.jar
httpclient-4.2.5.jar
httpcore-4.2.5.jar
jline-2.12.jar
pig-0.15.0.2.3.0.0-2557-core-h2.jar
protobuf-java-2.5.0.jar

For the SPD Engine on HDP 2.1, include these JAR files as well:

commons-beanutils-1.7.0.jar
commons-cli-1.2.jar
commons-collections-3.2.1.jar
commons-configuration-1.6.jar
commons-lang-2.6.jar
commons-logging-1.1.3.jar
jackson-core-asl-1.9.13.jar
jackson-jaxrs-1.9.13.jar
jackson-mapper-asl-1.9.13.jar
jackson-xc-1.9.13.jar
slf4j-api-1.7.5.jar
slf4j-log4j12-1.7.5.jar

IBM InfoSphere BigInsights 2.1.2 JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop. SAS Deployment Manager automatically copies the requisite JAR and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

activation-1.1.jar
adaptive-mr.jar
ant-1.7.1.jar
ant-launcher-1.7.1.jar
antlr-runtime-3.4.jar
automaton-1.11-8.jar
avro-1.7.4.jar
avro-mapred-1.7.4.jar
biginsights-gpfs-2.2.0.jar
biginsights-sftpfs-1.0.0.jar
bigsql-serdes.jar
bonecp-0.7.1.RELEASE.jar
core-3.1.1.jar
datanucleus-api-jdo-3.2.4.jar
datanucleus-core-3.2.6.jar
datanucleus-rdbms-3.2.5.jar
db2jcc-10.5.jar
db2jcc_license_cisuz-10.5.jar
derby-10.8.3.1.jar
findbugs-annotations-1.3.9-1.jar
guardium-proxy.jar
guava-11.0.2.jar
hadoop-core-2.2.0-mr1.jar
hadoop-core.jar
hadoop-example.jar
hadoop-mr1-examples-2.2.0.jar
hadoop-streaming.jar
hbase-client-0.96.0.jar
hbase-common-0.96.0.jar
hbase-hadoop2-compat-0.96.0-tests.jar
hbase-hadoop2-compat-0.96.0.jar
hbase-prefix-tree-0.96.0.jar
hbase-protocol-0.96.0.jar
hbase-server-0.96.0-tests.jar
hbase-server-0.96.0.jar
hive-beeline-0.12.0.jar
hive-cli-0.12.0.jar
hive-common-0.12.0.jar
hive-contrib-0.12.0.jar

hive-exec-0.12.0.jar
hive-hwi-0.12.0.jar
hive-jdbc-0.12.0.jar
hive-metastore-0.12.0.jar
hive-service-0.12.0.jar
hive-shims-0.12.0.jar
htrace-core-2.01.jar
httpclient-4.2.5.jar
httpcore-4.2.4.jar
ibm-compression.jar
jamon-runtime-2.3.1.jar
jansi-1.9.jar
JavaEWAH-0.3.2.jar
javolution-5.5.1.jar
jdo-api-3.0.1.jar
jersey-core-1.8.jar
jersey-json-1.8.jar
jersey-server-1.8.jar
jettison-1.3.1.jar
jetty-6.1.26.jar
jetty-sslengine-6.1.26.jar
jetty-util-6.1.26.jar
jline-0.9.94.jar
joda-time-2.1.jar
jsch-0.1.43.jar
JSON4J_Apache-1.0.jar
jsp-2.1-6.1.14.jar
jsr305-1.3.9.jar
jython-standalone-2.5.3.jar
libfb303-0.9.0.jar
log4j-1.2.15.jar
log4j-1.2.17.jar
metrics-core-2.1.2.jar
netty-3.2.4.Final.jar
netty-3.6.6.Final.jar
pig-0.12.0.jar
piggybank.jar
protobuf-java-2.5.0.jar
stax-api-1.0-2.jar
stax-api-1.0.1.jar
ST4-4.0.4.jar
tempus-fugit-1.1.jar
workflowScheduler.jar
xz-1.0.jar
zookeeper-3.4.5.jar

MapR JAR Files

MapR 3.1 JAR Files

To install the client side JAR files for MapR, follow the instructions at MapR: Setting Up the Client. The installed MapR client version must match the version of the MapR cluster that SAS connects to.

Note: In the third maintenance release for SAS 9.4, use SAS Deployment Manager to install and configure your SAS MapR client. If you use SAS Deployment Manager to deploy Hadoop client files, SAS Deployment Manager automatically copies the requisite JAR files, defines the SAS_HADOOP_JAR_PATH environment variable to point to those files, and saves the environment variable location in the sasv9.cfg file.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

For more information, see “Additional Configuration for MapR” on page 33.

hadoop-mapreduce-client-*.jar
hadoop-yarn-*.jar
hive-common-0.12-mapr-1403.jar
hive-contrib-0.12-mapr-1403.jar
hive-exec-0.12-mapr-1403.jar
hive-jdbc-0.12-mapr-1403.jar
hive-metastore-0.12-mapr-1403.jar
hive-service-0.12-mapr-1403.jar
httpclient-4.1.1.jar
httpcore-4.1.jar
pig-0.12.1-mapr-1403-withouthadoop.jar
zookeeper-3.4.5-mapr-1401.jar

For the SPD Engine on MapR 3.1, include these JAR files as well:

activation-1.1.jar
amazon-s3.jar
asm-3.2.jar
aspectjrt-1.6.5.jar
aspectjtools-1.6.5.jar
avro-1.7.1.jar
aws-java-sdk-1.3.26.jar
commons-cli-1.2.jar
commons-codec-1.5.jar

commons-configuration-1.8.jar
commons-daemon-1.0.1.jar
commons-el-1.0.jar
commons-httpclient-3.0.1.jar
commons-httpclient-3.1.jar
commons-lang-2.6.jar
commons-logging-1.0.4.jar
commons-logging-1.1.1.jar
commons-logging-api-1.0.4.jar
commons-math-2.1.jar
commons-net-1.4.1.jar
commons-net-3.1.jar
core-3.1.1.jar
emr-metrics-1.0.jar
eval-0.5.jar
gson-1.4.jar
guava-13.0.1.jar
guava-r09-jarjar.jar
hadoop-0.20.2-auth.jar
hadoop-0.20.2-dev-capacity-scheduler.jar
hadoop-0.20.2-dev-core.jar
hadoop-0.20.2-dev-fairscheduler.jar
hadoop-metrics-0.20.2-dev.jar
hadoop-metrics2-0.20.2-dev.jar
hive-beeline-0.12-mapr-1403.jar
hive-cli-0.12-mapr-1403.jar
hive-hbase-handler-0.12-mapr-1403.jar
hive-hwi-0.12-mapr-1403.jar
hive-serde-0.12-mapr-1403.jar
hive-shims-0.12-mapr-1403.jar
hsqldb-1.8.0.10.jar
jackson-core-asl-1.5.2.jar
jackson-mapper-asl-1.5.2.jar
jasper-compiler-5.5.12.jar
jasper-runtime-5.5.12.jar
jersey-core-1.8.jar
jersey-json-1.8.jar
jersey-server-1.8.jar
jets3t-0.6.1.jar
jetty-6.1.26.jar
jetty-plus-6.1.26.jar
jetty-util-6.1.26.jar
junit-4.5.jar
kfs-0.2.2.jar
libfb303-0.9.0.jar

log4j-1.2.15.jar
logging-0.1.jar
mapr-hbase-1.0.3-mapr-3.1.0.jar
mockito-all-1.8.2.jar
mockito-all-1.8.5.jar
mysql-connector-java-5.1.25-bin.jar
oro-2.0.8.jar
pig-0.10.1.jar
protobuf-java-2.4.1.jar
servlet-api-2.5-20081211.jar
servlet-api-2.5-6.1.14.jar
xmlenc-0.52.jar

MapR 4.0 JAR Files

To install the client side JAR files for MapR, follow the instructions at MapR: Setting Up the Client. The installed MapR client version must match the version of the MapR cluster that SAS connects to.

Note: In the third maintenance release for SAS 9.4, use SAS Deployment Manager to install and configure your SAS MapR client. If you use SAS Deployment Manager to deploy Hadoop client files, SAS Deployment Manager automatically copies the requisite JAR files, defines the SAS_HADOOP_JAR_PATH environment variable to point to those files, and saves the environment variable location in the sasv9.cfg file.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

For more information, see “Additional Configuration for MapR” on page 33.

data-nucleus-api-jdo-3.2.6.jar
data-nucleus-core-3.2.10.jar
data-nucleus-rdbms-3.2.9.jar
hadoop-mapreduce-client-*.jar
hadoop-yarn-*.jar
hive-common-0.13-mapr-1408.jar
hive-contrib-0.13-mapr-1408.jar
hive-exec-0.13-mapr-1408.jar
hive-hcatalog-hbase-storage-handler-0.13-mapr-1408.jar
hive-hcatalog-hbase-server-extensiom-0.13-mapr-1408.jar
hive-hcatalog-hbase-pig-adapter-0.13-mapr-1408.jar
hive-jdbc-0.13-mapr-1408.jar
hive-metastore-0.13-mapr-1408.jar
hive-service-0.13-mapr-1408.jar

httpclient-4.2.5.jar
httpcore-4.2.5.jar
jdo-api-3.0.1.jar
pig-withouthadoop.jar
zookeeper-3.4.5-mapr-1406.jar

For the SPD Engine on MapR 4.0, include these JAR files as well:

activation-1.1.jar
antlr-runtime-3.4.jar
aopalliance-1.0.jar
asm-3.2.jar
avro-1.7.4.jar
baseutils-4.0.1-mapr.jar
central-logging-4.0.1-mapr.jar
commons-beanutils-1.7.0.jar
commons-beanutils-core-1.8.0.jar
commons-cli-1.2.jar
commons-codec-1.4.jar
commons-collections-3.2.1.jar
commons-compress-1.4.1.jar
commons-configuration-1.6.jar
commons-daemon-1.0.13.jar
commons-digester-1.8.jar
commons-el-1.0.jar
commons-httpclient-3.1.jar
commons-io-2.4.jar
commons-lang-2.5.jar
commons-lang-2.6.jar
commons-logging-1.1.1.jar
commons-logging-1.1.3.jar
commons-math3-3.1.1.jar
commons-net-3.1.jar
derby-10.10.1.1.jar
eval-0.5.jar
guava-11.0.2.jar
guava-13.0.1.jar
guice-3.0.jar
guice-servlet-3.0.jar
hadoop-annotations-2.4.1-mapr-1408.jar
hadoop-auth-2.4.1-mapr-1408.jar
hadoop-common-2.4.1-mapr-1408-tests.jar
hadoop-common-2.4.1-mapr-1408.jar
hadoop-hdfs-2.4.1-mapr-1408-tests.jar
hadoop-hdfs-2.4.1-mapr-1408.jar
hadoop-hdfs-nfs-2.4.1-mapr-1408.jar

hadoop-mapreduce-examples-2.4.1-mapr-1408.jar
hadoop-nfs-2.4.1-mapr-1408.jar
hamcrest-core-1.1.jar
hbase-0.94.21-mapr-1407.jar
hive-cli-0.13.0-mapr-1408.jar
jackson-core-asl-1.8.8.jar
jackson-jaxrs-1.8.8.jar
jackson-mapper-asl-1.8.8.jar
jackson-xc-1.8.8.jar
jasper-compiler-5.5.23.jar
jasper-runtime-5.5.23.jar
java-xmlbuilder-0.4.jar
javax.inject-1.jar
jaxb-api-2.2.2.jar
jaxb-impl-2.2.3-1.jar
jersey-client-1.9.jar
jersey-core-1.9.jar
jersey-guice-1.9.jar
jersey-json-1.9.jar
jersey-server-1.9.jar
jets3t-0.9.0.jar
jettison-1.1.jar
jetty-6.1.26.jar
jetty-util-6.1.26.jar
jline-0.9.94.jar
jruby-complete-1.6.5.jar
jsch-0.1.42.jar
json-20080701.jar
jsp-api-2.1.jar
jsr305-1.3.9.jar
junit-4.10.jar
junit-4.8.2.jar
leveldbjni-all-1.8.jar
libfb303-0.9.0.jar
libprotodefs-4.0.1-mapr.jar
log4j-1.2.17.jar
mapr-hbase-4.0.1-mapr.jar
maprfs-4.0.1-mapr.jar
maprfs-core-4.0.1-mapr-tests.jar
maprfs-core-4.0.1-mapr.jar
maprfs-diagnostic-tools-4.0.1-mapr.jar
maprfs-jni-4.0.1-mapr.jar
mockito-all-1.8.5.jar
mysql-connector-java-5.1.25-bin.jar
netty-3.6.2.Final.jar

paranamer-2.3.jar
protobuf-java-2.5.0.jar
servlet-api-2.5-6.1.14.jar
servlet-api-2.5.jar
slf4j-api-1.7.5.jar
slf4j-log4j12-1.7.5.jar
snappy-java-1.0.4.1.jar
stax-api-1.0-2.jar
templates
xmlenc-0.52.jar
xz-1.0.jar

Pivotal HD JAR Files

Pivotal HD 1.1.1 JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop. SAS Deployment Manager automatically copies the requisite JAR and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

activation-1.1.jar
asm-3.2.jar
avro-1.5.3.jar
guava-11.0.2.jar
hadoop-annotations-2.0.5-alpha-gphd-2.1.1.0.jar
hadoop-auth-2.0.5-alpha-gphd-2.1.1.0.jar
hadoop-common-2.0.5-alpha-gphd-2.1.1.0.jar
hadoop-hdfs-2.0.5-alpha-gphd-2.1.1.0.jar
hadoop-mapreduce-client-app-2.0.5-alpha-gphd-2.1.1.0.jar
hadoop-mapreduce-client-common-2.0.5-alpha-gphd-2.1.1.0.jar
hadoop-mapreduce-client-core-2.0.5-alpha-gphd-2.1.1.0.jar

hadoop-mapreduce-client-jobclient-2.0.5-alpha-gphd-2.1.1.0.jar
hadoop-mapreduce-client-shuffle-2.0.5-alpha-gphd-2.1.1.0.jar
hadoop-vaidya-2.0.5-alpha-gphd-2.1.1.0.jar
hadoop-yarn-api-2.0.5-alpha-gphd-2.1.1.0.jar
hadoop-yarn-client-2.0.5-alpha-gphd-2.1.1.0.jar
hadoop-yarn-common-2.0.5-alpha-gphd-2.1.1.0.jar
hadoop-yarn-server-common-2.0.5-alpha-gphd-2.1.1.0.jar
hive-beeline-0.11.0-gphd-2.1.1.0.jar
hive-cli-0.11.0-gphd-2.1.1.0.jar
hive-common-0.11.0-gphd-2.1.1.0.jar
hive-contrib-0.11.0-gphd-2.1.1.0.jar
hive-exec-0.11.0-gphd-2.1.1.0.jar
hive-hwi-0.11.0-gphd-2.1.1.0.jar
hive-jdbc-0.11.0-gphd-2.1.1.0.jar
hive-metastore-0.11.0-gphd-2.1.1.0.jar
hive-service-0.11.0-gphd-2.1.1.0.jar
hive-shims-0.11.0-gphd-2.1.1.0.jar
httpclient-4.1.3.jar
httpcore-4.1.3.jar
javax.servlet-2.5.0.v201103041518.jar
jersey-core-1.8.jar
jersey-json-1.8.jar
jersey-server-1.8.jar
jets3t-0.6.1.jar
jettison-1.1.jar
jetty-continuation-7.6.10.v20130312.jar
jetty-http-7.6.10.v20130312.jar
jetty-io-7.6.10.v20130312.jar
jetty-security-7.6.10.v20130312.jar
jetty-server-7.6.10.v20130312.jar
jetty-servlet-7.6.10.v20130312.jar
jetty-util-7.6.10.v20130312.jar
jetty-webapp-7.6.10.v20130312.jar
jetty-xml-7.6.10.v20130312.jar
jsch-0.1.42.jar
jsr305-1.3.9.jar
libfb303-0.9.0.jar
log4j-1.2.17.jar
paranamer-2.3.jar
pig-0.12.0-gphd-2.1.1.0-withouthadoop.jar
protobuf-java-2.5.jar
stax-api-1.0.1.jar
xmlenc-0.52.jar

For the SPD Engine on Pivotal HD 1.1.1, include these JAR files as well:

commons-beanutils-1.7.0.jar
commons-beanutils-core-1.8.0.jar
commons-cli-1.2.jar
commons-codec-1.4.jar
commons-collections-3.2.1.jar
commons-compress-1.4.1.jar
commons-configuration-1.6.jar
commons-digester-1.8.jar
commons-el-1.0.jar
commons-httpclient-3.1.jar
commons-io-2.1.jar
commons-lang-2.5.jar
commons-logging-1.1.1.jar
commons-math-2.1.jar
commons-net-3.1.jar
jackson-core-asl-1.8.8.jar
jackson-jaxrs-1.8.8.jar
jackson-mapper-asl-1.8.8.jar
jackson-xc-1.8.8.jar
jasper-compiler-5.5.23.jar
jasper-runtime-5.5.23.jar
jaxb-api-2.2.2.jar
jaxb-impl-2.2.3-1.jar
jsp-api-2.1.jar
slf4j-api-1.6.1.jar
slf4j-log4j12-1.6.1.jar

Pivotal HD 2.1 JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop. SAS Deployment Manager automatically copies the requisite JAR and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

activation-1.1.jar
asm-3.2.jar
avro-1.7.4.jar
guava-11.0.2.jar
hadoop-annotations-2.2.0-gphd-3.1.0.0.jar
hadoop-auth-2.2.0-gphd-3.1.0.0.jar
hadoop-common-2.2.0-gphd-3.1.0.0.jar
hadoop-hdfs-2.2.0-gphd-3.1.0.0.jar
hadoop-hdfs-nfs-2.2.0-gphd-3.1.0.0.jar
hadoop-mapreduce-client-app-2.2.0-gphd-3.1.0.0.jar
hadoop-mapreduce-client-common-2.2.0-gphd-3.1.0.0.jar
hadoop-mapreduce-client-core-2.2.0-gphd-3.1.0.0.jar
hadoop-mapreduce-client-jobclient-2.2.0-gphd-3.1.0.0.jar
hadoop-mapreduce-client-shuffle-2.2.0-gphd-3.1.0.0.jar
hadoop-nfs-2.2.0-gphd-3.1.0.0.jar
hadoop-vaidya-2.2.0-gphd-3.1.0.0.jar
hadoop-yarn-api-2.2.0-gphd-3.1.0.0.jar
hadoop-yarn-client-2.2.0-gphd-3.1.0.0.jar
hadoop-yarn-common-2.2.0-gphd-3.1.0.0.jar
hadoop-yarn-server-common-2.2.0-gphd-3.1.0.0.jar
hive-cli-0.12.0-gphd-3.1.0.0.jar
hive-common-0.12.0-gphd-3.1.0.0.jar
hive-exec-0.12.0-gphd-3.1.0.0.jar
hive-jdbc-0.12.0-gphd-3.1.0.0.jar
hive-metastore-0.12.0-gphd-3.1.0.0.jar
hive-service-0.12.0-gphd-3.1.0.0.jar
httpclient-4.2.5.jar
httpcore-4.2.4.jar
javax.servlet-2.5.0.v201103041518.jar
jersey-core-1.9.jar
jersey-json-1.9.jar
jersey-server-1.9.jar
jets3t-0.6.1.jar
jettison-1.1.jar
jetty-continuation-7.6.10.v20130312.jar
jetty-http-7.6.10.v20130312.jar
jetty-io-7.6.10.v20130312.jar
jetty-security-7.6.10.v20130312.jar
jetty-server-7.6.10.v20130312.jar
jetty-servlet-7.6.10.v20130312.jar
jetty-util-7.6.10.v20130312.jar
jetty-webapp-7.6.10.v20130312.jar
jetty-xml-7.6.10.v20130312.jar
jsch-0.1.42.jar
jsr305-1.3.9.jar

libfb303-0.9.0.jar
log4j-1.2.17.jar
netty-3.6.2.Final.jar
paranamer-2.3.jar
pig-0.12.0-gphd-3.1.0.0-withouthadoop.jar
protobuf-java-2.5.0.jar
stax-api-1.0.1.jar
xmlenc-0.52.jar
xz-1.0.jar

For the SPD Engine on Pivotal HD 2.1, include these JAR files as well:

commons-beanutils-1.7.0.jar
commons-beanutils-core-1.8.0.jar
commons-cli-1.2.jar
commons-codec-1.4.jar
commons-collections-3.2.1.jar
commons-compress-1.4.1.jar
commons-configuration-1.6.jar
commons-digester-1.8.jar
commons-el-1.0.jar
commons-httpclient-3.1.jar
commons-io-2.1.jar
commons-lang-2.5.jar
commons-logging-1.1.1.jar
commons-math-2.1.jar
commons-net-3.1.jar
jackson-core-asl-1.8.8.jar
jackson-jaxrs-1.8.8.jar
jackson-mapper-asl-1.8.8.jar
jackson-xc-1.8.8.jar
jasper-compiler-5.5.23.jar
jasper-runtime-5.5.23.jar
jaxb-api-2.2.2.jar
jaxb-impl-2.2.3-1.jar
jsp-api-2.1.jar
slf4j-api-1.7.5.jar
slf4j-log4j12-1.7.5.jar

Pivotal HD 3.0 JAR Files

Note: This list of JAR files is used when you manually configure the HADOOP procedure, the FILENAME statement Hadoop access method, and the SPD Engine. If you license SAS/ACCESS Interface to Hadoop, it is strongly recommended that you use SAS Deployment Manager to configure the JAR and configuration files. In addition to these JAR files, which are required for basic SAS/ACCESS functionality, SAS Deployment Manager gathers other JAR files

for software that uses SAS/ACCESS Interface to Hadoop. Examples of this software are SAS Embedded Process and SAS In-Database Code Accelerator for Hadoop. SAS Deployment Manager automatically copies the requisite JAR and configuration files, places these files in a specified directory, and defines the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables.

Note: JAR files include version numbers. For example, the Pig JAR file might be pig-0.10.0, pig-0.11.1, and so on. The version numbers can change frequently. The latest JAR files are located in your Hadoop `client` directory. The Hadoop `client` directory includes symbolic links to the various technology directories such as HDFS and Hive. The latest JAR files are contained in these individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

activation-1.1.jar
asm-3.2.jar
avro-1.7.4.jar
guava-11.0.2.jar
hadoop-annotations-2.6.0.3.0.0.0-249.jar
hadoop-auth-2.6.0.3.0.0.0-249.jar
hadoop-common-2.6.0.3.0.0.0-249.jar
hadoop-hdfs-2.6.0.3.0.0.0-249.jar
hadoop-hdfs-nfs-2.6.0.3.0.0.0-249.jar
hadoop-mapreduce-client-app-2.6.0.3.0.0.0-249.jar
hadoop-mapreduce-client-common-2.6.0.3.0.0.0-249.jar
hadoop-mapreduce-client-core-2.6.0.3.0.0.0-249.jar
hadoop-mapreduce-client-jobclient-2.6.0.3.0.0.0-249.jar
hadoop-mapreduce-client-shuffle-2.6.0.3.0.0.0-249.jar
hadoop-nfs-2.6.0.3.0.0.0-249.jar
hadoop-yarn-api-2.6.0.3.0.0.0-249.jar
hadoop-yarn-client-2.6.0.3.0.0.0-249.jar
hadoop-yarn-common-2.6.0.3.0.0.0-249.jar
hadoop-yarn-server-common-2.6.0.3.0.0.0-249.jar
hive-cli-0.14.0.3.0.0.0-249.jar
hive-common-0.14.0.3.0.0.0-249.jar
hive-exec-0.14.0.3.0.0.0-249.jar
hive-jdbc-0.14.0.3.0.0.0-249.jar
hive-metastore-0.14.0.3.0.0.0-249.jar
hive-service-0.14.0.3.0.0.0-249.jar
httpclient-4.2.5.jar
httpcore-4.2.5.jar
jersey-core-1.9.jar
jersey-json-1.9.jar
jersey-server-1.9.jar
jets3t-0.9.0.jar
jettison-1.1.jar
jetty-6.1.26.hwx.jar

jetty-util-6.1.26.hwx.jar
jsr305-1.3.9.jar
jsr305-2.0.3.jar
libfb303-0.9.0.jar
log4j-1.2.17.jar
netty-3.6.2.Final.jar
paranamer-2.3.jar
pig-0.14.0.3.0.0.0-249-core-h2.jar
protobuf-java-2.5.0.jar
stax-api-1.0.2.jar
xmlenc-0.52.jar
xz-1.0.jar

For the SPD Engine on Pivotal HD 3.0, include these JAR files as well:

commons-beanutils-1.7.0.jar
commons-beanutils-core-1.8.0.jar
commons-cli-1.2.jar
commons-codec-1.4.jar
commons-collections-3.2.1.jar
commons-compress-1.4.1.jar
commons-configuration-1.6.jar
commons-digester-1.8.jar
commons-el-1.0.jar
commons-httpclient-3.1.jar
commons-io-2.1.jar
commons-lang-2.5.jar
commons-logging-1.1.1.jar
commons-math-2.1.jar
commons-net-3.1.jar
jackson-core-asl-1.8.8.jar
jackson-jaxrs-1.8.8.jar
jackson-mapper-asl-1.8.8.jar
jackson-xc-1.8.8.jar
jasper-compiler-5.5.23.jar
jasper-runtime-5.5.23.jar
jaxb-api-2.2.2.jar
jaxb-impl-2.2.3-1.jar
jsp-api-2.1.jar
slf4j-api-1.7.5.jar
slf4j-log4j12-1.7.5.jar

Appendix 2

SAS Environment Variables for Hadoop

Dictionary

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

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

Syntax

SAS_HADOOP_CONFIG_PATH *pathname*

Required Argument

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.

```
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. To see the path, enter the following command:

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

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

Table A2.1 *Defining the SAS_HADOOP_CONFIG_PATH Environment Variable*

Operating Environment	Method	Example
UNIX *	SAS configuration file	`-set SAS_HADOOP_CONFIG_PATH "/sasdata/cluster1/conf"`
	SAS invocation	`-set SAS_HADOOP_CONFIG_PATH "/sasdata/cluster1/conf"`
	OPTIONS statement	`options set=SAS_HADOOP_CONFIG_PATH="/sasdata/cluster1/conf";`
Windows	SAS configuration file	`-set SAS_HADOOP_CONFIG_PATH "C:\sasdata\cluster1\conf"`
	SAS invocation	`-set SAS_HADOOP_CONFIG_PATH "C:\sasdata\cluster1\conf"`
	OPTIONS statement	`options set=SAS_HADOOP_CONFIG_PATH="C:\sasdata\cluster1\conf";`

* 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

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

Syntax

SAS_HADOOP_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;
   C:\MyHadoopJars\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:

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

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

Table A2.2 Defining the SAS_HADOOP_JAR_PATH Environment Variable

Operating Environment	Method	Example
UNIX *	SAS configuration file	`-set SAS_HADOOP_JAR_PATH "/third_party/Hadoop/jars/lib"`
	SAS invocation	`-set SAS_HADOOP_JAR_PATH "/third_party/Hadoop/jars/lib"`
	OPTIONS statement	`options set=SAS_HADOOP_JAR_PATH="/third_party/Hadoop/jars/lib";`
Windows	SAS configuration file	`-set SAS_HADOOP_JAR_PATH "C:\third_party\Hadoop\jars/lib"`
	SAS invocation	`-set SAS_HADOOP_JAR_PATH "C:\third_party\Hadoop\jars\lib"`
	OPTIONS statement	`options set=SAS_HADOOP_JAR_PATH="C:\third_party\Hadoop\jars\lib";`

* 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 Your Hadoop Version" on page 34.

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

SAS_HADOOP_RESTFUL Environment Variable

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

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

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

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

Syntax

SAS_HADOOP_RESTFUL 0 | 1

Required Arguments

0

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

1

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

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

Details

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

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

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

Table A2.3 *Defining the SAS_HADOOP_RESTFUL Environment Variable*

Method	Example
SAS configuration file	`-set SAS_HADOOP_RESTFUL 1`
SAS invocation	`-set SAS_HADOOP_RESTFUL 1`
OPTIONS statement	`options set=SAS_HADOOP_RESTFUL 1;`

Recommended Reading

- *Base SAS Procedures*
- *SAS/ACCESS to Relational Databases: Reference*
- *SAS SPD Engine: Storing Data in the Hadoop Distributed File System*
- *SAS Statements: Reference*
- *SAS and Hadoop Technology: Overview*

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

C

D

E

F

H

I

J

K

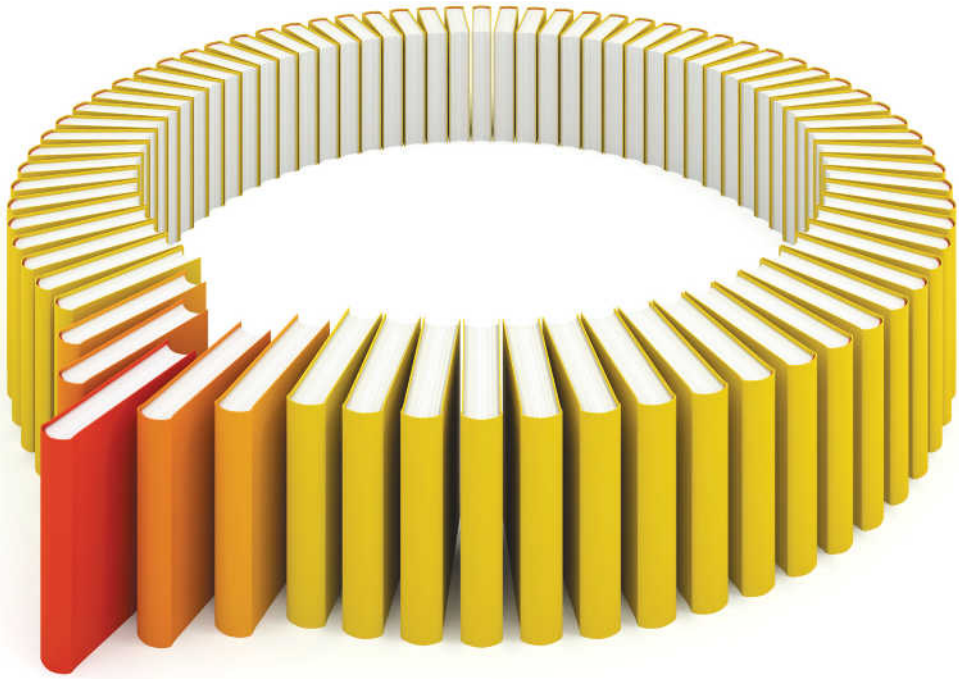
M

P

S

V

W



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.