THE POWER TO KNOW.

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

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS® 9.4 Hadoop Configuration Guide for Base SAS® and SAS/ACCESS®, Fourth Edition*. Cary, NC: SAS Institute Inc.

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

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2016

9.4-P1:hadoopbacg

Contents

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

Overview

In the fourth maintenance release for SAS 9.4, SAS Deployment Manager is now available for Base SAS and the SPD Engine as a means to obtain JAR and configuration files.

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

In the fourth maintenance release for SAS 9.4, the FILENAME statement Hadoop access method, HADOOP procedure, and SAS/ACCESS Interface to Hadoop support Apache Knox Gateway authentication.

Enhancements to SAS Deployment Manager

In the fourth maintenance release for SAS 9.4, SAS Deployment Manager has been enhanced in the following areas:

- If your SAS software uses Impala or Oozie services, SAS Deployment Manager generates an inventory.json file.
- You can choose to provide either a private key file or the user ID and password as credentials for the UNIX user account that has SSH access to the machine that is hosting HiveServer2.
- You can filter the JAR files by obtaining the most recent version of the JAR files.
- You can specify a non-default Hive user name.
- You can specify a non-default port for the Hive service.

Apache Knox Gateway

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

- For SAS/ACCESS Interface to Hadoop, see “Apache Knox Gateway Security” on page 18.
- For the FILENAME statement and PROC HADOOP, see “Using Apache Knox Gateway Security” on page 8.

Chapter 1
Verifying Your Hadoop Environment

Pre-Installation Checklist for SAS Software That Interfaces with Hadoop

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

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

- Gain working knowledge of the Hadoop distribution that you are using (for example, Cloudera or Hortonworks).

  You also need working knowledge of the Hadoop Distributed File System (HDFS), MapReduce 1, MapReduce 2, YARN, and HiveServer2 services. For more information, see the Apache website or the vendor's website.

  For MapR, you must install the MapR client. The installed MapR client version must match the version of the MapR cluster that SAS connects to. For more information, see MapR: Setting Up the Client.
- Confirm that the HCatalog, HDFS, HiveServer2, MapReduce, Oozie, Sqoop, and YARN services are running on the Hadoop cluster. SAS software uses these various services, and this confirmation ensures that the appropriate JAR files are gathered during the configuration.
- Know the location of the MapReduce home.
- Know the host name of the Hive server and the name of the NameNode.
- Determine where the HDFS and Hive servers are running. If the Hive server is not running on the same machine as the NameNode, note the server and port number of the Hive server for future configuration.
- Request permission to restart the MapReduce service.
- Verify that you can run a MapReduce job successfully.
- Understand and verify your Hadoop user authentication.
- Understand and verify your security setup.

  It is highly recommended that you enable Kerberos or another security protocol for data security.

Verify that you can connect to your Hadoop cluster (HDFS and Hive) from your client machine outside of the SAS environment with your defined security protocol.

Chapter 2

Base SAS and SAS/ACCESS Software with Hadoop

Introduction

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

- Base SAS components
  - FILENAME Statement Hadoop Access Method

    enables Base SAS users to use Hadoop to read from or write to a file from HDFS.
  - HADOOP procedure

    enables Base SAS users to submit HDFS commands, Pig language code, and MapReduce programs against Hadoop data. PROC HADOOP interfaces with the Hadoop JobTracker. This is the service within Hadoop that controls tasks to specific nodes in the cluster.
  - 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 data through the SAS Scalable Performance Data (SPD) Engine. The SPD Engine is designed for high-performance data delivery, reading data sources that contain billions of observations. The engine uses threads to read data very rapidly and in parallel. The SPD Engine reads, writes, and updates data in the HDFS.
- SAS/ACCESS Interface to Hadoop

  enables you to interact with your data by using SQL constructs through HiveServer2. SAS/ACCESS Interface to Hadoop also enables you to access data directly from the underlying data storage layer, the Hadoop Distributed File System (HDFS).
- SAS/ACCESS Interface to Impala

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

Configuration Information for Other SAS Software

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

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

For more information, see the following documentation:

- Installation and configuration information for in-database processing (including the SAS Embedded Process): *SAS In-Database Products: Administrator's Guide*
- Installation and configuration information for contextual analysis: *SAS Contextual Analysis In-Database Scoring for Hadoop: Administrator's Guide*
- Installation and configuration information for data quality operations: *SAS Data Loader for Hadoop: Installation and Configuration Guide*
- Installation and configuration of the High-Performance Analytics Infrastructure: *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*
- Basic installation (not part of a solution installation) of SAS In-Memory Statistics for Hadoop: *SAS LASR Analytic Server: Reference Guide*

Chapter 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 “Making Hadoop JAR and Configuration Files Available to the SAS Client Machine” on page 6 and “Using WebHDFS or HttpFS” on page 7.

   Note: If you decide to connect to the Hadoop server with an HTTP REST API, you must make Hadoop configuration files available to the SAS client machine. The Hadoop JAR files are not required on the SAS client machine for the REST API.

3. If you use Apache Oozie, follow the configuration steps in "Using Apache Oozie" on page 9.
4. Run basic tests to confirm that your Hadoop connections are working.

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

Prerequisites for the FILENAME Statement and PROC HADOOP

Setting Up Your Environment for the FILENAME Statement and PROC HADOOP

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

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

   For more information, see Chapter 1, "Verifying Your Hadoop Environment," on page 1.
2. Review the Hadoop distributions that are supported for the FILENAME statement and PROC HADOOP.

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

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

To submit the FILENAME statement or PROC HADOOP to a Hadoop server, a set of Hadoop JAR and configuration files must be available to the SAS client machine. To make the required JAR and configuration files available, you must obtain these files from the Hadoop cluster, copy the files to the SAS client machine, and define the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables to set the location of the JAR and configuration files.

In the fourth maintenance release for SAS 9.4, you use SAS Deployment Manager to obtain the JAR and configuration files. For more information, see Appendix 1, "Using the SAS Deployment Manager to Obtain Hadoop JAR and Configuration Files," on page 31.

Note: Gathering the JAR and configuration files is a one-time process. If you have already gathered the Hadoop JAR and configuration files for another SAS component, you do not need to gather the files again unless you make changes to your Hadoop distribution. For more information, see "When to Collect New JAR and Configuration Files" on page 52.

Using WebHDFS or HttpFS

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

Using WebHDFS or HttpFS removes the need for client-side JAR files for HDFS, but JAR files are still needed to submit MapReduce programs and Pig language programs.

Note: If you decide to connect to the Hadoop server with an HTTP REST API, you must make Hadoop configuration files available to the SAS client machine. The Hadoop JAR files are not required on the SAS client machine for the REST API. For more information, see "Making Hadoop JAR and Configuration Files Available to the SAS Client Machine" on page 6.

To use WebHDFS or HttpFS instead of the HDFS service:

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

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

   or

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

   or

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

   For more information, see "SAS_HADOOP_RESTFUL Environment Variable" on page 58.

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 JAR and Configuration Files Available to the SAS Client Machine" on page 6.

Using Apache Knox Gateway Security

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

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

  ```
  options set=SAS_HADOOP_RESTFUL 1;
  ```

  For more information, see "SAS_HADOOP_RESTFUL Environment Variable" on page 58.

- Make sure the configuration files include the properties for the WebHDFS location. For more information, see "Using WebHDFS or HttpFS" on page 7.

- Set the SAS environment variable KNOX_GATEWAY_URL to the location of the Knox Gateway. Here is an example:

  ```
  options set=KNOX_GATEWAY_URL='https://server:port/gateway/default';
  ```

  For more information, see "KNOX_GATEWAY_URL Environment Variable" on page 53.

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

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

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

Using Apache Oozie

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

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

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

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

   or

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

   or

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

   For more information, see "SAS_HADOOP_RESTFUL Environment Variable" on page 58.

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;
```

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**.

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

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

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

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.

```
options set=SAS_HADOOP_CONFIG_PATH "C:\sasdata\hdcluster1\conf";
options set=SAS_HADOOP_JAR_PATH="C:\sasdata\hdcluster1\jars";
proc hadoop username='sasabc' password='sasabc' verbose;
   hdfs mkdir='/user/sasabc/new_directory';
   hdfs delete='/user/sasabc/temp2_directory';
   hdfs copytolocal='/user/sasabc/testdata.txt'
   out='C:\Users\sasabc\Hadoop\testdata.txt' overwrite;
run;
```

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*

Chapter 4
Configuring SAS/ACCESS for Hadoop

Overview of Steps to Configure SAS/ACCESS Interface to Hadoop

1. Verify that all prerequisites have been satisfied.

This step ensures that you understand your Hadoop environment. For more information, see "Prerequisites for SAS/ACCESS Interface to Hadoop" on page 14.

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

   This step involves using SAS Deployment Manager to copy a set of JAR and configuration files to the SAS client machine that accesses Hadoop.

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

3. Review the following sections for additional configuration information.

   - SAS/ACCESS Interface to Impala

     "Configuring SAS/ACCESS Interface to Impala" on page 16
   - PROC SQOOP

     "Configuring PROC SQOOP" on page 17
   - HiveServer2

     "Working with Hive" on page 21
   - WebHDFS or HttpFS

     "Using WebHDFS or HttpFS" on page 20

4. Review security and user access.

   For more information, see "Security and User Access to Hadoop" on page 17.

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 22.

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.

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

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

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

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

Note: Gathering the JAR and configuration files is a one-time process. If you have already gathered the Hadoop JAR and configuration files for another SAS component, you do not need to gather the files again unless you make changes to your Hadoop distribution. For more information, see “When to Collect New JAR and Configuration Files” on page 52.

Additional Configuration for MapR

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

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

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

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

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

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

- MapR requires this JRE option for a Kerberos connection:

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

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

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

Configuring SAS/ACCESS Interface to Impala

Impala ODBC Driver

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

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

Note: Cloudera ODBC driver for Impala version 2.5.17 or later is required for AIX.

Bulk Loading

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

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

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

  For more information, see "Using WebHDFS or HttpFS" on page 20 and Appendix 2, "SAS Environment Variables for Hadoop," on page 53.

- 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 "Making Hadoop JAR and Configuration Files Available to the SAS Client Machine" on page 15.

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

Configuring PROC SQOOP

Prerequisites for PROC SQOOP

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

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

  Note: Apache Sqoop Server 2 is not supported.

Configuration for PROC SQOOP

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

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

- The SAS_HADOOP_RESTFUL environment variable must be set to 1 and either WebHDFS or HttpFS must be enabled.

  For more information, see "Using WebHDFS or HttpFS" on page 20.

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

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

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

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

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

Security and User Access to Hadoop

Kerberos Security

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

- If you are using Advanced Encryption Standard (AES) encryption with Kerberos, you 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:\windows\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.

Apache Knox Gateway Security

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

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

  ```
  options set=SAS_HADOOP_RESTFUL 1;
  ```

For more information, see "SAS_HADOOP_RESTFUL Environment Variable" on page 58.

- Make sure the configuration files include the properties for the WebHDFS location. For more information, see "Using WebHDFS or HttpFS" on page 20.
- Set the SAS environment variable KNOX_GATEWAY_URL to the location of the Knox Gateway. Here is an example:

  ```
  options set=KNOX_GATEWAY_URL='https://server:port/gateway/default';
  ```

  For more information, see "KNOX_GATEWAY_URL Environment Variable" on page 53.
- Set up the SSL encryption protocol. For example, the SSLCALISTLOC= system option must be submitted to specify the location of the public certificate or certificates for trusted certificate authorities (CAs). For more information about the SSL encryption protocol and the SSLCALISTLOC= system option, see *Encryption in SAS*.
- Use the URI= option in the LIBNAME statement option to connect to Knox. The URI= option is required to fully qualify the JDBC connection string to a Hive cluster that is behind a Knox gateway. Here is an example:

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

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

JDBC Read Security

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

HDFS Write Security

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

HDFS Permission Requirements for Optimized Reads

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

Using WebHDFS or HttpFS

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

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

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

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

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

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

   For more information, see "SAS_HADOOP_RESTFUL Environment Variable" on page 58.

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 "Making Hadoop JAR and Configuration Files Available to the SAS Client Machine" on page 15.

Working with Hive

Starting with Hive

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

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

```
0: jdbc:hive2://cdh58d1hive:10000/default> !connect jdbc:hive2://cdh58d1hive:10000/default
Connecting to jdbc:hive2://cdh58d1hive:10000/default
Enter username for jdbc:hive2://cdh58d1hive:10000/default: hadoop
Enter password for jdbc:hive2://cdh58d1hive:10000/default: *******
Connected to: Apache Hive (version 1.1.0-cdh5.8.0)
.
.
1: jdbc:hive2://cdh58d1hive:10000/default> create table test (c char(10) );
.
.
INFO  : OK
.
.
1: jdbc:hive2://cdh58d1hive:10000/default> insert into table test values ('test');
.
.
INFO  : OK
No rows affected (16.668 seconds)
1: jdbc:hive2://cdh58d1hive:10000/default> select * from test;
.
.
INFO  : OK
+-------------+--+
|   test.c    |
+-------------+--+
| test        |
+-------------+--+
1 row selected (0.156 seconds)
```

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

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

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

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

Running the Hive Service on Your Hadoop Server

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

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

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

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 2016-01-21 13:25 /tmp
drwxr-xr-x - hdfs supergroup 0 2016-01-21 11:46 /user
```

Validating Your SAS/ACCESS to Hadoop Connection

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

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

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

Sample libref connection to HiveServer2 (default):

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

```
NOTE: Libref HDPLIB was successfully assigned as follows:
Engine: HADOOP
Physical Name: jdbc:hive2://hadoop01:10000/default
```

Sample PROC SQL connection:

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

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

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

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

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

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

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

```
libname hdplib hadoop server=hadoop01 port=10000 user=hadoop_usr
   password=hadoop_usr_pwd;
ERROR: java.lang.NoClassDefFoundError:
org/apache/hadoop/hive/metastore/api/MetaException
ERROR: Unable to connect to server or to call the Java Drivermanager.
ERROR: Error trying to establish connection.
ERROR: Error in the LIBNAME statement.
```

Documentation for Using SAS/ACCESS Interface to Hadoop

The documentation can be found in "SAS/ACCESS Interface to Hadoop" in *SAS/ACCESS for Relational Databases: Reference*.

Chapter 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 26.

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

   For more information, see “Making Hadoop JAR and Configuration Files Available to the SAS Client Machine” on page 26.

3. Review security and user access.

   For more information, see “Kerberos Security” on page 27.

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 28.

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.

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

Overview

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

In the fourth maintenance release for SAS 9.4, you use the SAS Deployment Manager to obtain the JAR and configuration files. For more information, see Appendix 1, "Using the SAS Deployment Manager to Obtain Hadoop JAR and Configuration Files," on page 31.

Note: Gathering the JAR and configuration files is a one-time process. If you have already gathered the Hadoop JAR and configuration files for another SAS component, you do not need to gather the files again unless you make changes to your Hadoop distribution. For more information, see "When to Collect New JAR and Configuration Files" on page 52.

Additional Requirements for MapR Systems

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

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

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

In addition, set the **`java.library.path`** property to the directory that contains the 64-bit MapRClient shareable library. Set the **`java.security.auth.login.config`** property to the **`mapr.login.conf`** file, which is 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.

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

Kerberos Security

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

- If you are using Advanced Encryption Standard (AES) encryption with Kerberos, you 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). As a best practice, first back up the existing local_policy.jar and US_export_policy.jar files in case they need to be restored.

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

- For the SPD Engine on AIX, add this option to your SAS command:

  ```
  -sasoptsappend '(-jreoptions "(-Djavax.security.auth.useSubjectCredsOnly=false)")'
  ```

- For the SPD Engine on HP-UX, set the KRB5CCNAME environment variable to point to your ticket cache whose filename includes your numeric user ID:

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

- For the SPD Engine on Windows, ensure that your Kerberos configuration file is in your Java environment. The algorithm to locate the krb5.conf file is as follows:

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

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

  - If the system property java.security.krb5.conf is not set, then the configuration file is looked for in 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="\\mycompany\hadoop\ConfigDirectory
\cdh45p1";
18   options set=SAS_HADOOP_JAR_PATH="\\mycompany\hadoop\JARDirectory\cdh45";
19   libname myspde spde '/user/sasabc' hdfshost=default;
NOTE: Libref MYSPDE was successfully assigned as follows:
      Engine:        SPDE
      Physical Name: /user/sasabc/
20   data myspde.class;
21      set sashelp.class;
22   run;

NOTE: There were 19 observations read from the data set SASHELP.CLASS.
NOTE: The data set MYSPDE.CLASS has 19 observations and 5 variables.
NOTE: DATA statement used (Total process time):
      real time           57.00 seconds
      cpu time            0.15 seconds



23
24   proc datasets library=myspde;
25      contents data=class;
26   run;

27
28      delete class;
29   run;

NOTE: Deleting MYSPDE.CLASS (memtype=DATA).
30   quit;

NOTE: PROCEDURE DATASETS used (Total process time):
      real time           37.84 seconds
      cpu time            0.25 seconds
```

Documentation for Using SPD Engine to Hadoop

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

Appendix 1
Using the SAS Deployment Manager to Obtain Hadoop JAR and Configuration Files

Information and Credentials Required to Configure Hadoop Using SAS Deployment Manager

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

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

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

  ```
  -bash-4.1$ time hive -e 'set -v'
  ```
- Oozie service host name and port number
- Impala service host name and port number
- Either the private key file or the user ID and password as credentials for the UNIX user account that has SSH access to the machine that is hosting the Hive, Oozie, and Impala services
- Ensure that Python 2.6 or later and strace are installed. Contact your system administrator if these packages are not installed on the system.

- For clusters secured with Kerberos, a valid ticket for the user on the client machine and the Hive service
- The HDFS user home directory, **/user/*user-account***, must exist and have Write permission for the *user-account* or the mapred account must have a drwxrwxrwx permission for the HDFS**/user** directory.

  Note: This is a critical prerequisite for a MapR cluster.
- Authorization to issue HDFS and Hive commands. During SAS Deployment Manager processing, a simple validation test is run to see whether HDFS (**hadoop**) and Hive (**hive**) commands can be issued. If the validation test fails, the script that pulls the JAR and configuration files is not be executed.
- If you have a Cloudera cluster with Sentry and RecordService services, the Sentry and RecordService services must be configured on the HiveServer2 node. This ensures that the configuration files that are required for Sentry and RecordService services are obtained when SAS Deployment Manager is run.

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

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

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

Note: Gathering the JAR and configuration files is a one-time process. If you have already gathered the Hadoop JAR and configuration files for another SAS component, you do not need to gather the files again unless you make changes to your Hadoop distribution. For more information, see “When to Collect New JAR and Configuration Files” on page 52.

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

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

1. If you are running on a cluster with Kerberos, you must kinit the HDFS user.
   a. Log on to the server using SSH as root with sudo access.

   ```
   ssh username@serverhostname
   sudo su - root
   ```

   b. Enter the following commands to kinit the HDFS user. The default HDFS user is **hdfs**.

   ```
   su - hdfs | hdfs-userid
   ```

```
kinit -kt location of keytab file
   user for which you are requesting a ticket
```

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

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

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

Valid starting    Expires           Service principal
06/20/16 09:51:26 06/27/16 09:51:26 krbtgt/HOST.COMPANY.COM@HOST.COMPANY.COM
     renew until 06/27/16 09:51:26
```

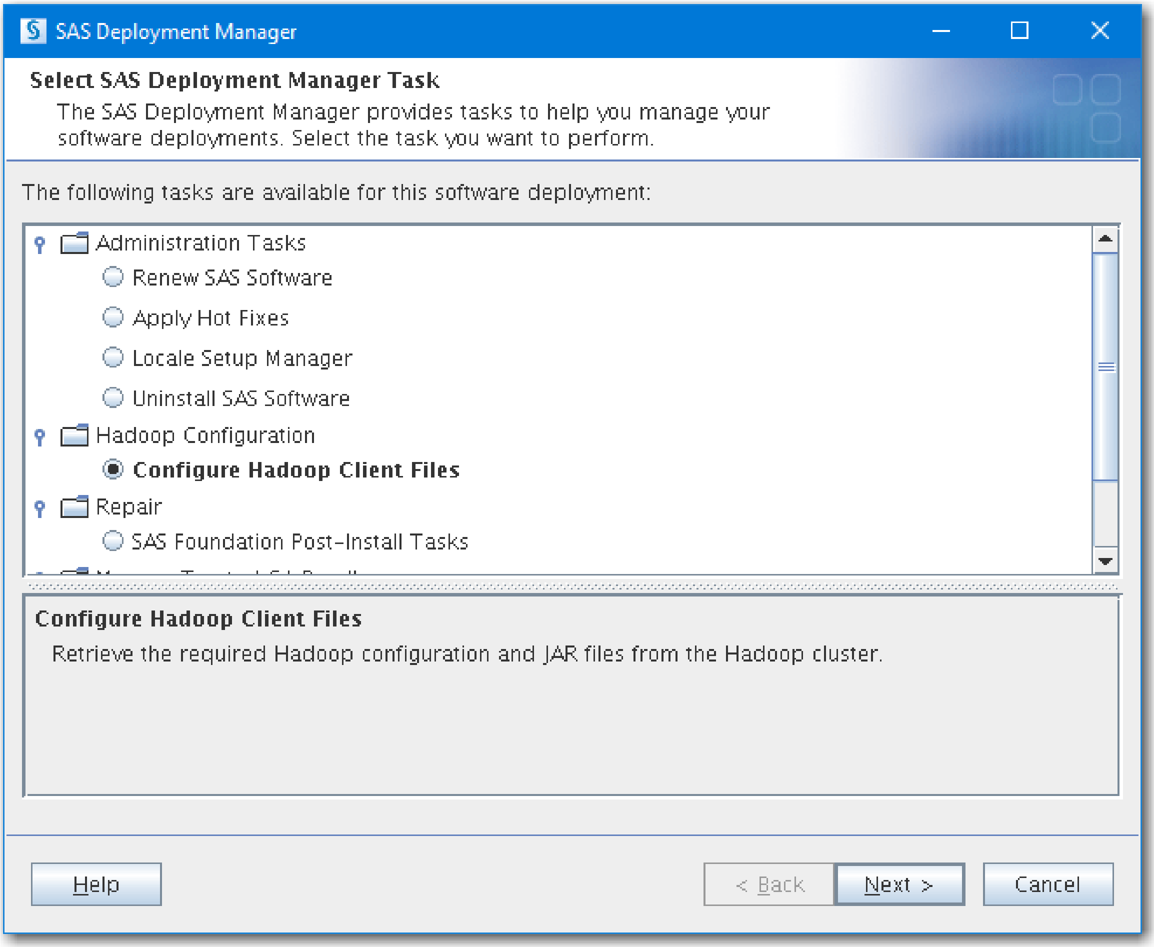
2. Start SAS Deployment Manager by running sasdm.exe for Windows or sasdm.sh for UNIX. The SAS Deployment Manager script is located in the **`/`*`SASHome`*`/SASDeploymentManager/9.4`** directory.

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

   The **Choose Language** page opens.

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

   Click **OK**. The **Select SAS Deployment Manager Task** page opens. The items listed under **Hadoop Configuration** depend on the SAS software that you have licensed.



SAS Deployment Manager

Select SAS Deployment Manager Task

The SAS Deployment Manager provides tasks to help you manage your software deployments. Select the task you want to perform.

The following tasks are available for this software deployment:

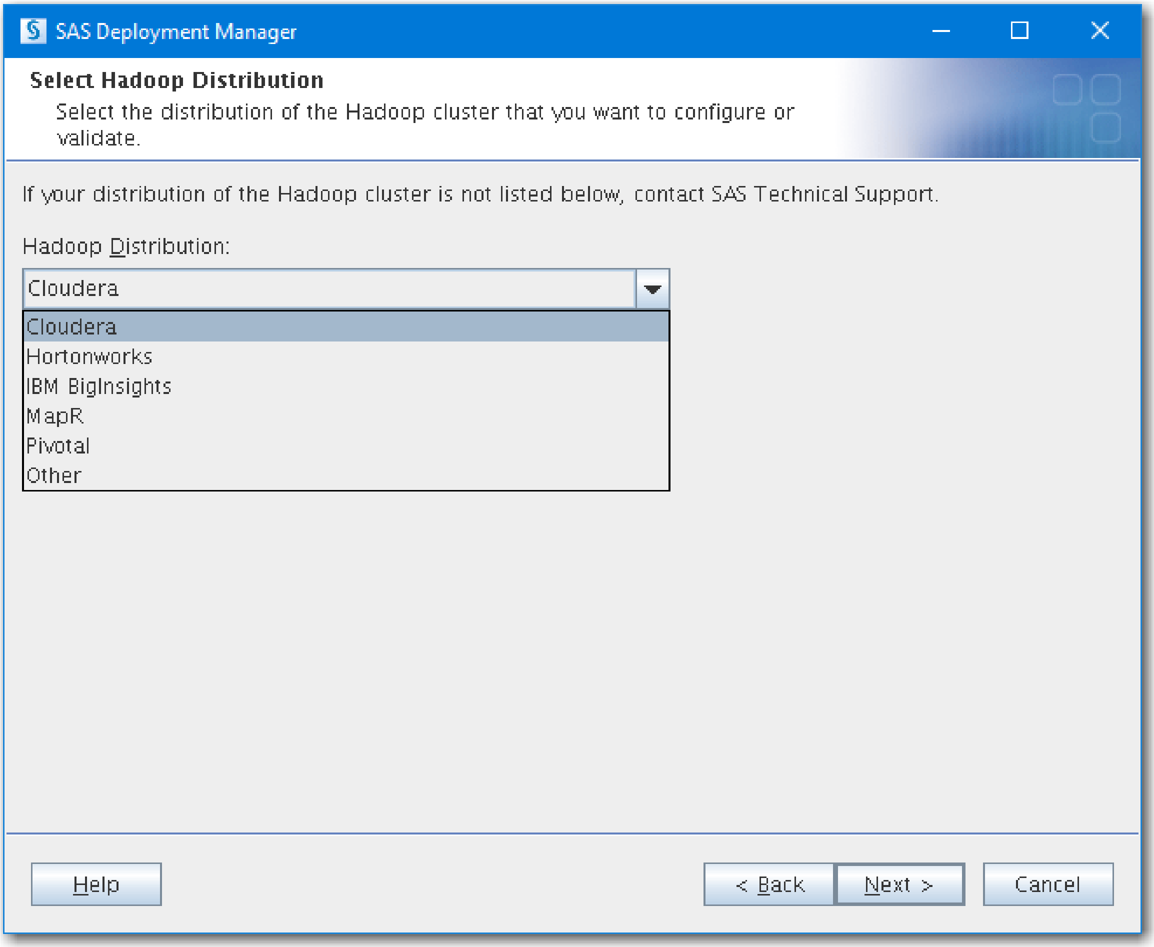
- Administration Tasks
  - ○ Renew SAS Software
  - ○ Apply Hot Fixes
  - ○ Locale Setup Manager
  - ○ Uninstall SAS Software
- Hadoop Configuration
  - ● **Configure Hadoop Client Files**
- Repair
  - ○ SAS Foundation Post-Install Tasks

Configure Hadoop Client Files

Retrieve the required Hadoop configuration and JAR files from the Hadoop cluster.

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

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



SAS Deployment Manager

Select Hadoop Distribution

Select the distribution of the Hadoop cluster that you want to configure or validate.

If your distribution of the Hadoop cluster is not listed below, contact SAS Technical Support.

Hadoop Distribution: Cloudera

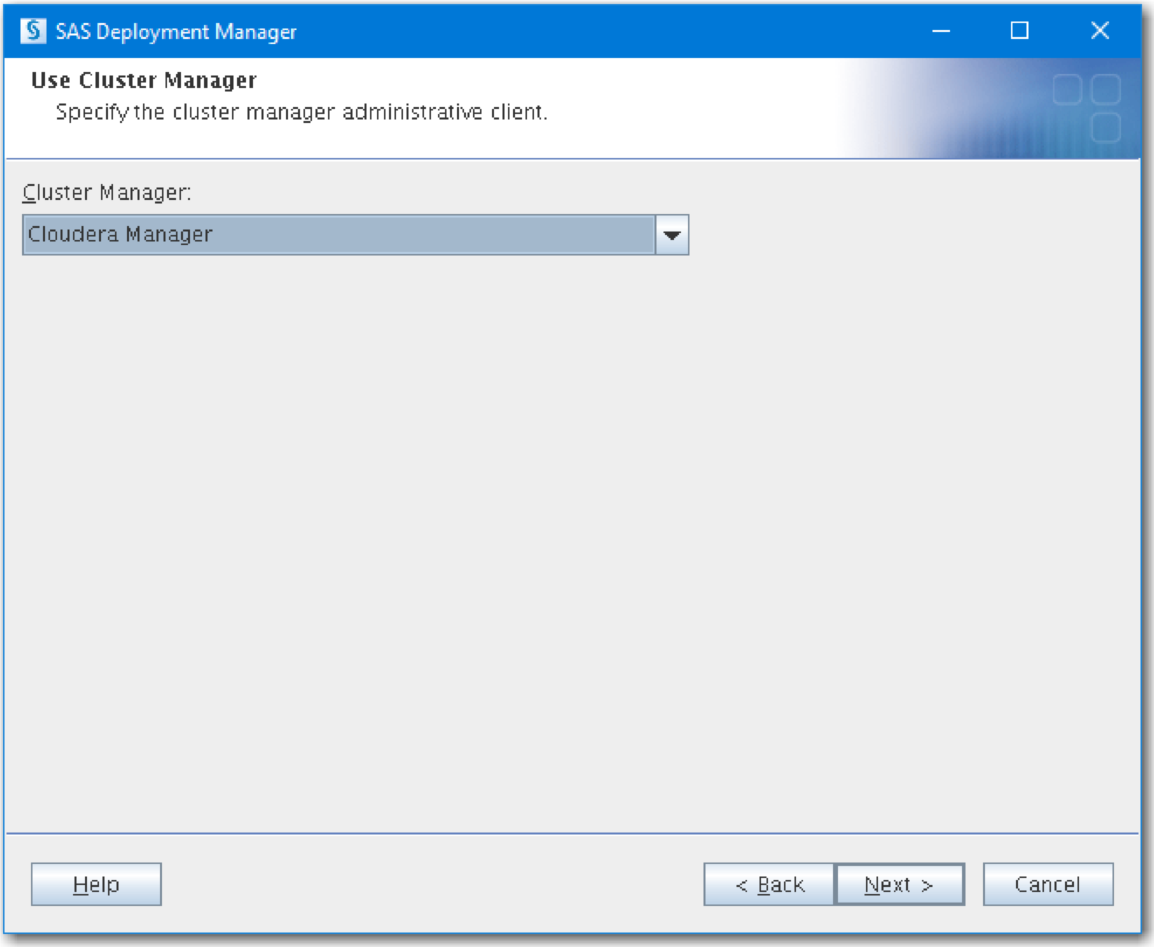
5. From the drop-down menu, select the distribution of Hadoop that you are using. If your distribution is not listed, exit SAS Deployment Manager and contact SAS Technical Support.

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

   Click **Next**.

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

   If your Hadoop distribution has an administrative client such as Cloudera Manager or Ambari, the **Use Cluster Manager** page opens.



Use Cluster Manager

Specify the cluster manager administrative client.

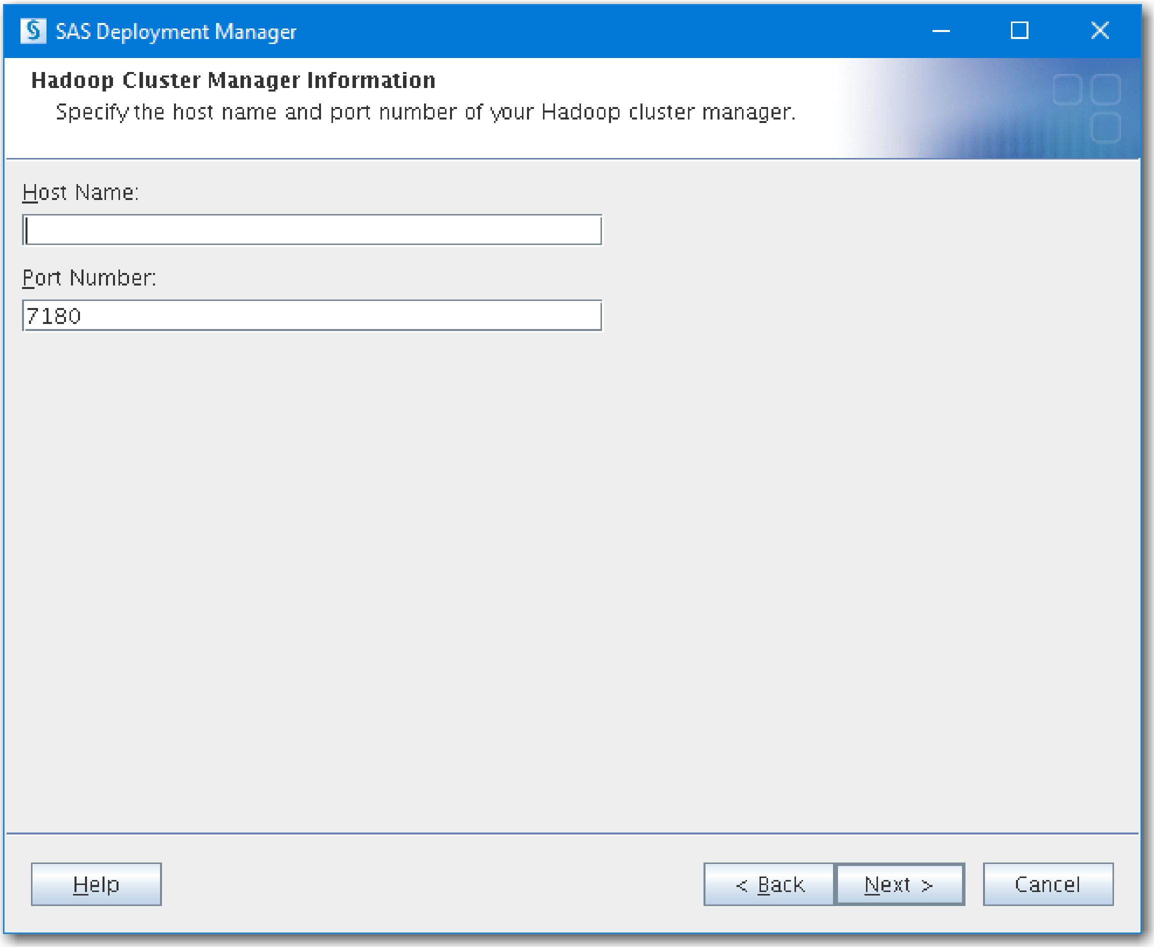
Cluster Manager: Cloudera Manager

6. Select the cluster manager administrative tool from the list.

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

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

   If you select a cluster manager and click **Next**, the **Hadoop Cluster Manager Information** page opens.



Hadoop Cluster Manager Information

Specify the host name and port number of your Hadoop cluster manager.

Host Name:

Port Number: 7180

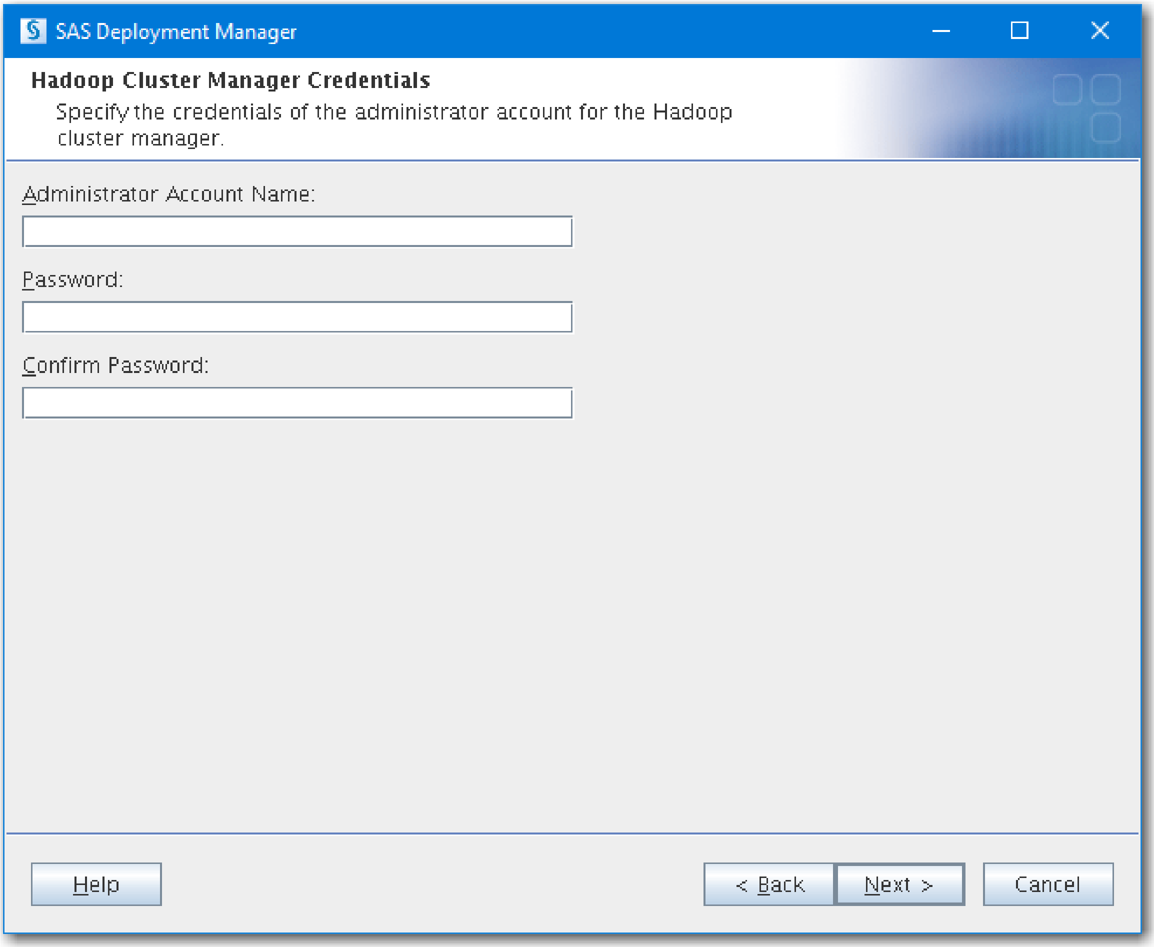
7. Enter the host name and port number for your Hadoop cluster manager.

   For Cloudera, enter the location where Cloudera Manager is running. For Hortonworks, IBM BigInsights, or Pivotal, enter the location where the Ambari server is running.

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

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

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



Hadoop Cluster Manager Credentials

Specify the credentials of the administrator account for the Hadoop cluster manager.

Administrator Account Name:

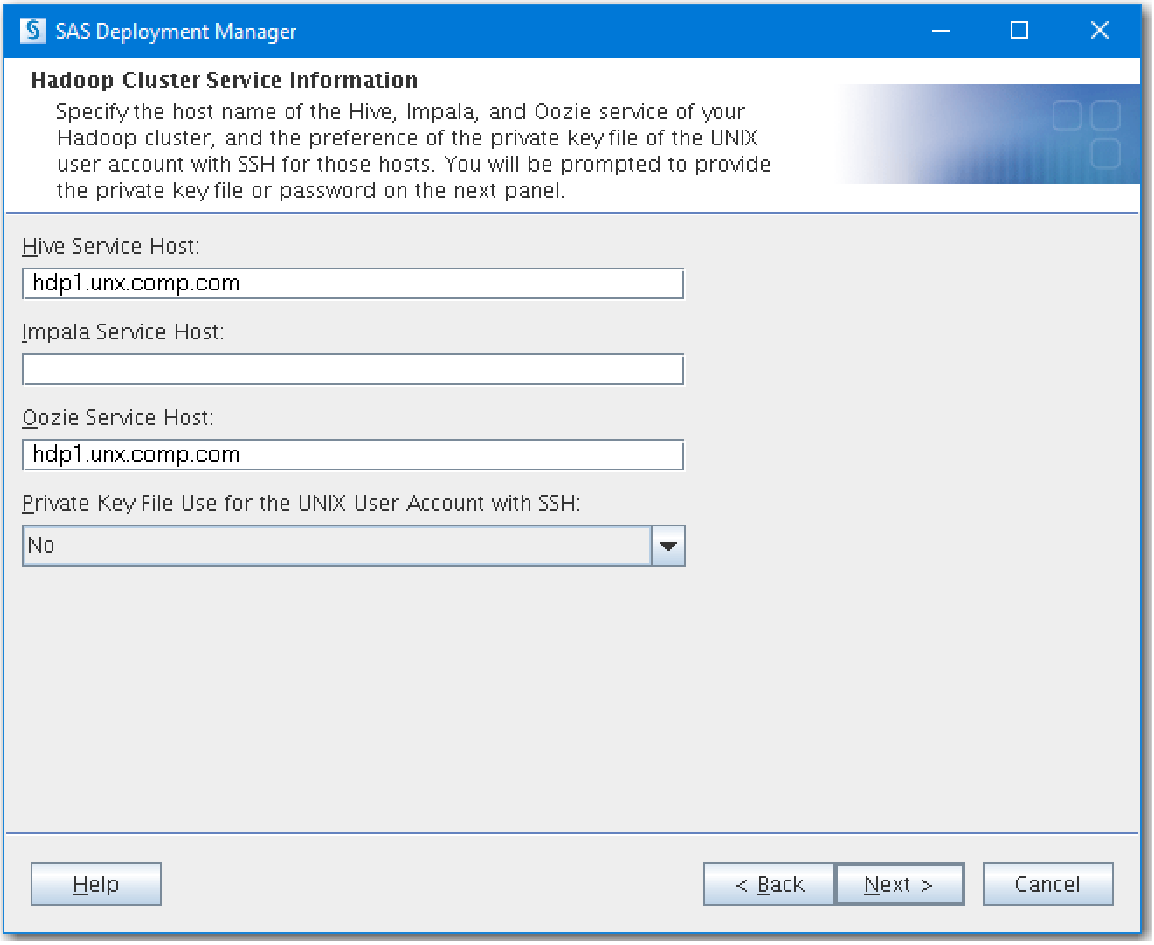
Password:

Confirm Password:

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 chances of a configuration error.

   Click **Next**. The **Hadoop Cluster Service Information** page opens.



Hadoop Cluster Service Information

Specify the host name of the Hive, Impala, and Oozie service of your Hadoop cluster, and the preference of the private key file of the UNIX user account with SSH for those hosts. You will be prompted to provide the private key file or password on the next panel.

Hive Service Host: hdp1.unx.comp.com

Impala Service Host:

Oozie Service Host: hdp1.unx.comp.com

Private Key File Use for the UNIX User Account with SSH: No

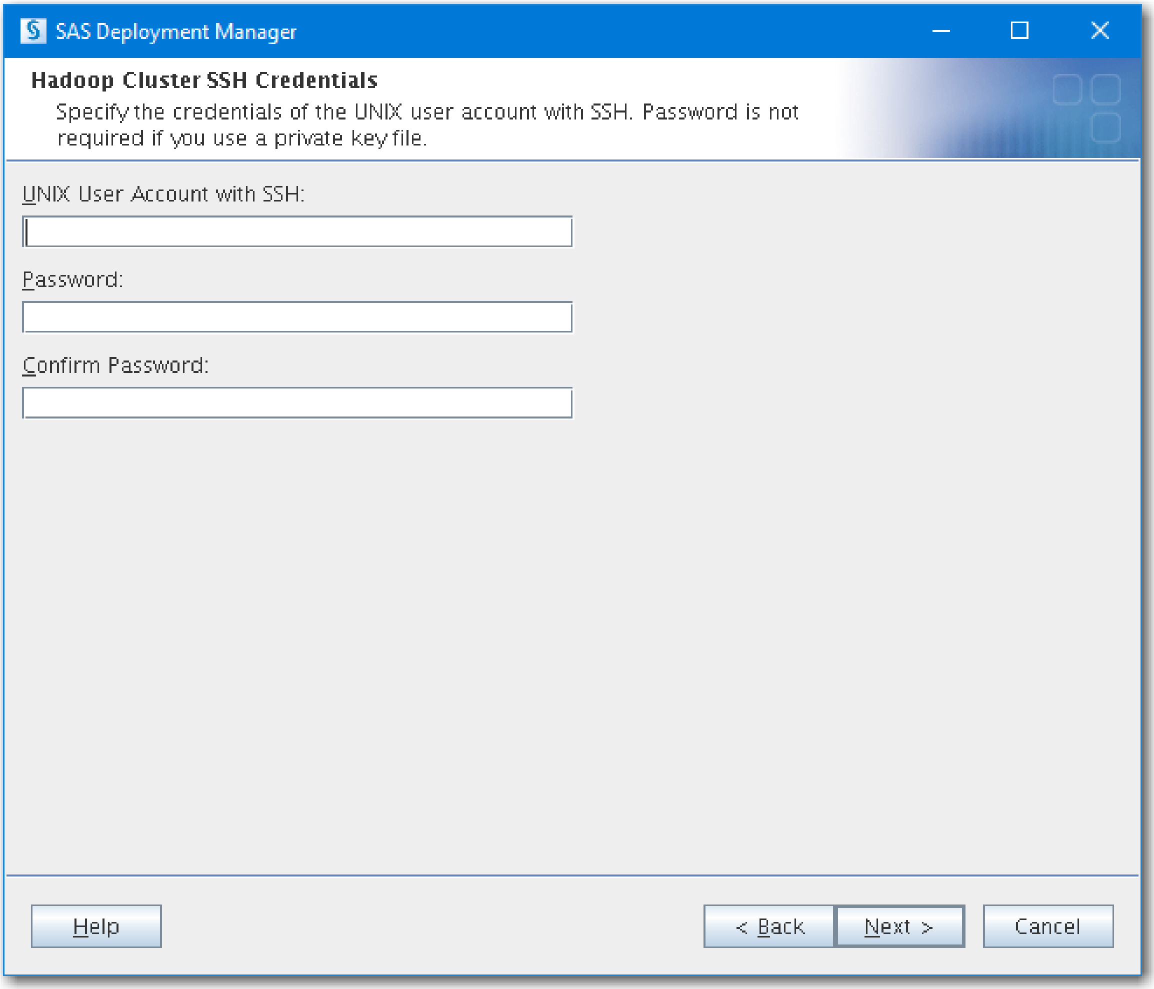
9. Enter the following information:

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

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

   - The preference to provide the private key file of the UNIX user account with SSH for the Hive, Impala, and Oozie hosts. Select **Yes** if you want to use the private key file (id_rsa). Select **No** if you want to use the password.

   If you select **No** and click **Next**, the **Hadoop Cluster SSH Credentials** page opens and asks for a password. Continue with Step 10 on page 41.



Hadoop Cluster SSH Credentials

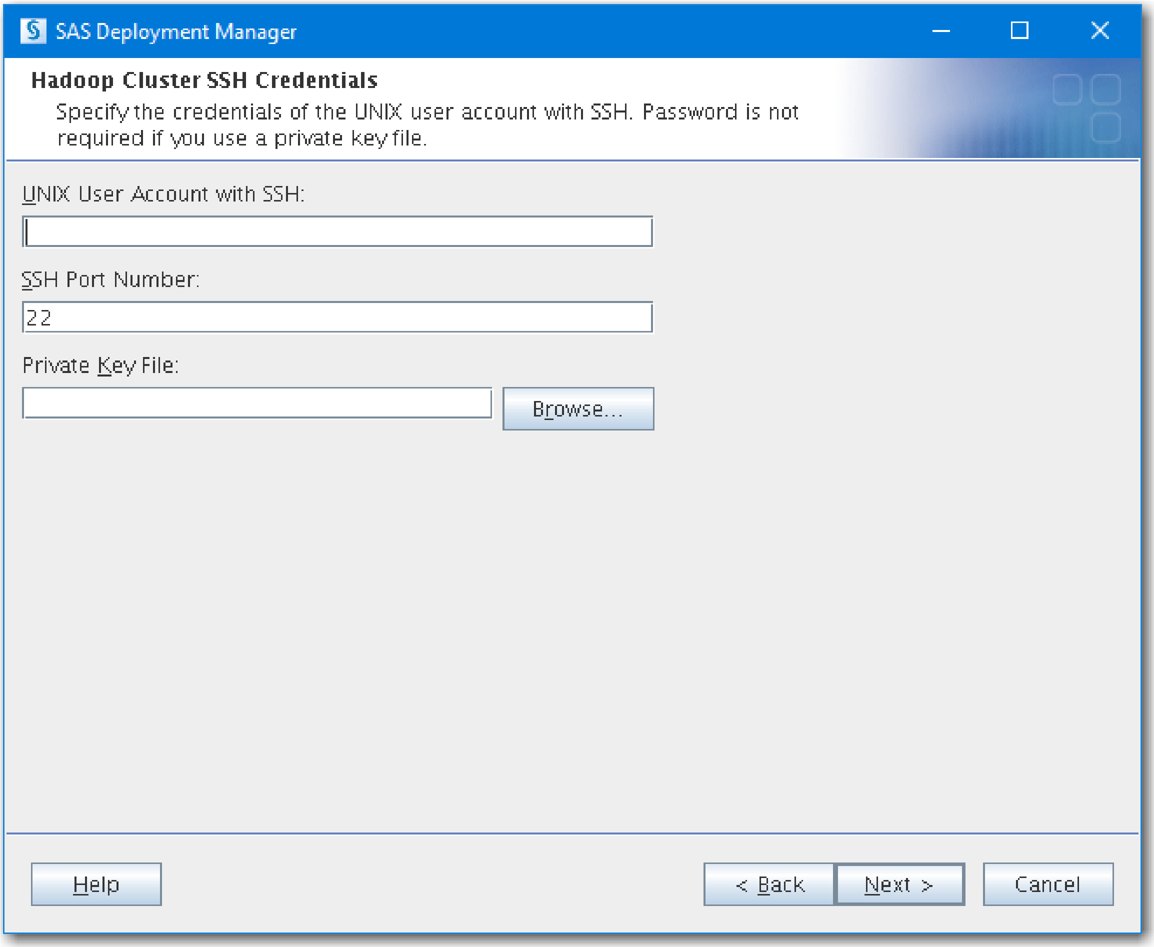
Specify the credentials of the UNIX user account with SSH. Password is not required if you use a private key file.

UNIX User Account with SSH:

Password:

Confirm Password:

If you select **Yes** and click **Next**, the **Hadoop Cluster SSH Credentials** page opens and asks for a private key file. Skip to Step 11 on page 41.



Hadoop Cluster SSH Credentials

Specify the credentials of the UNIX user account with SSH. Password is not required if you use a private key file.

UNIX User Account with SSH:

SSH Port Number: 22

Private Key File:

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

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

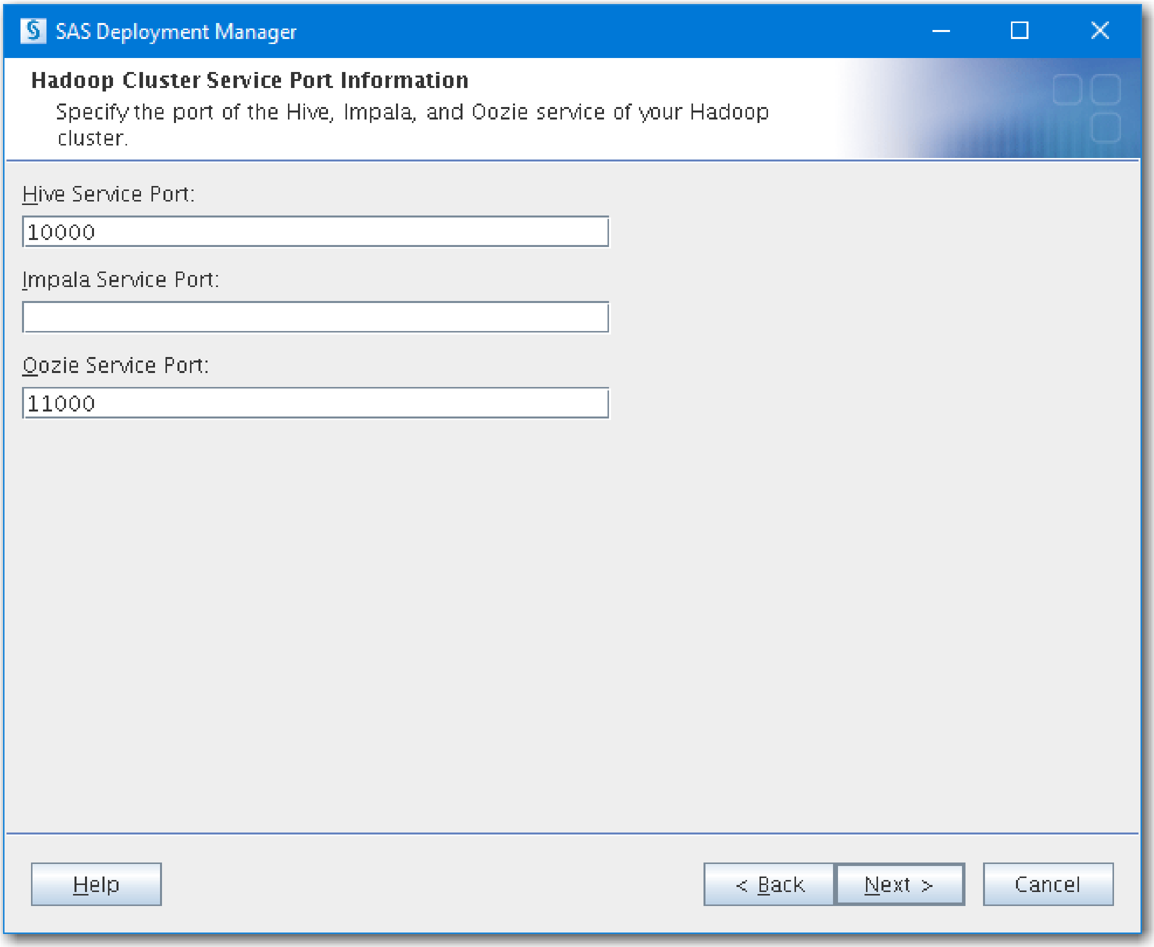
    Click **Next**. The **Hadoop Cluster Service Port Information** page opens. Skip to Step 12 on page 42.

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

    - The account name of the UNIX user who has SSH access to the machine that is hosting HiveServer2.
    - The path to the location of the private key file.

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

    Click **Next**. The **Hadoop Cluster Service Port Information** page opens.



Hadoop Cluster Service Port Information

Specify the port of the Hive, Impala, and Oozie service of your Hadoop cluster.

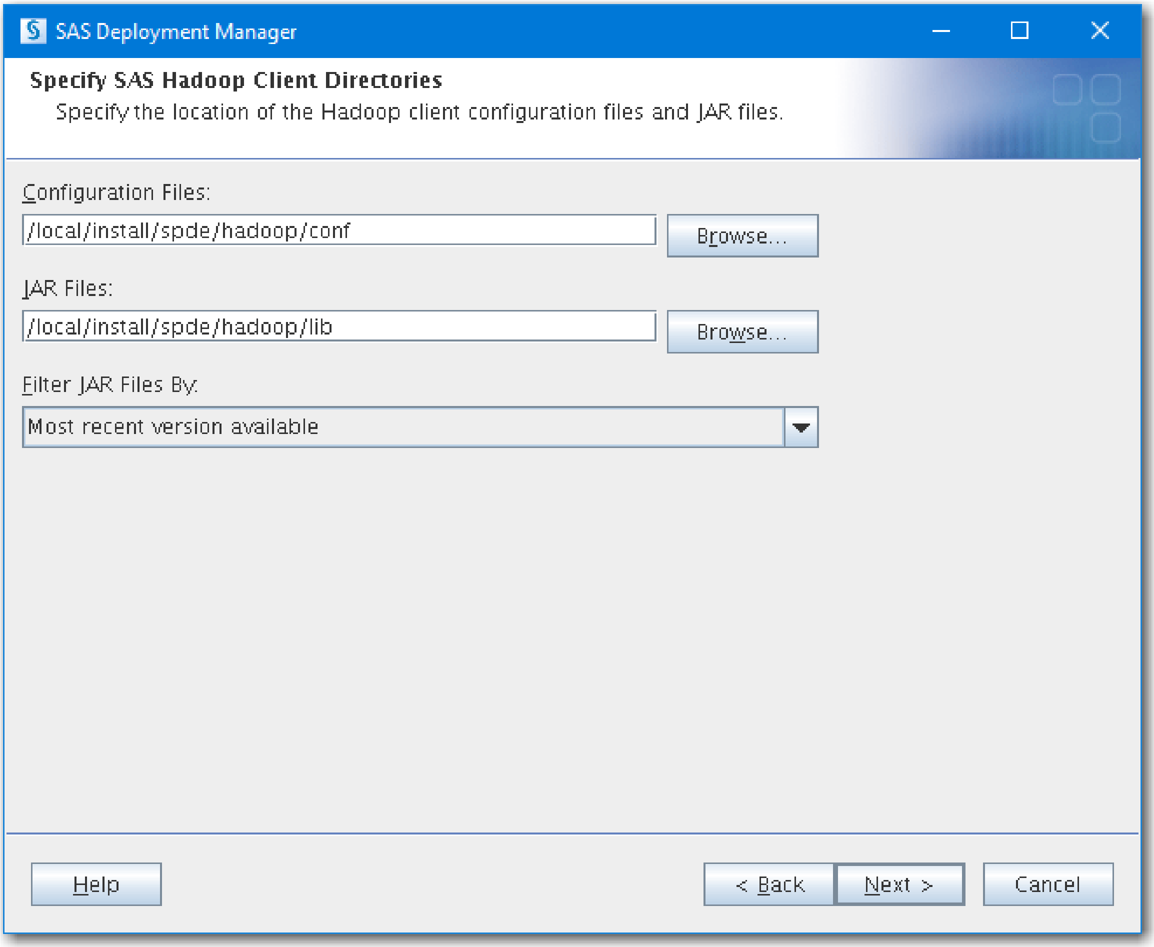
Hive Service Port: 10000

Impala Service Port:

Oozie Service Port: 11000

12. Enter the port numbers of the Hive, Impala, and Oozie services of your Hadoop cluster.

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



Specify SAS Hadoop Client Directories

Specify the location of the Hadoop client configuration files and JAR files.

Configuration Files: /local/install/spde/hadoop/conf

JAR Files: /local/install/spde/hadoop/lib

Filter JAR Files By: Most recent version available

13. Specify the locations of the configuration files and JAR files for the Hadoop client and choose whether to select the most recent version of the configuration and JAR files.

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

    Note: If you want to specify a directory other than the default directory, click **Browse** and select another directory. This step can also create a new directory. However, SAS Deployment Manager creates the sas_hadoop_config.properties file and a repository directory in the **`hadoop`** directory. The repository directory contains the resulting configuration and JAR files each time you run SAS Deployment Manager to pull configuration and JAR files. The files can be found in **`/`*`SASHome`*`/hadoop/repository/hive/`*`hive-host-name`*`/`*`time-stamp`*`/`**.

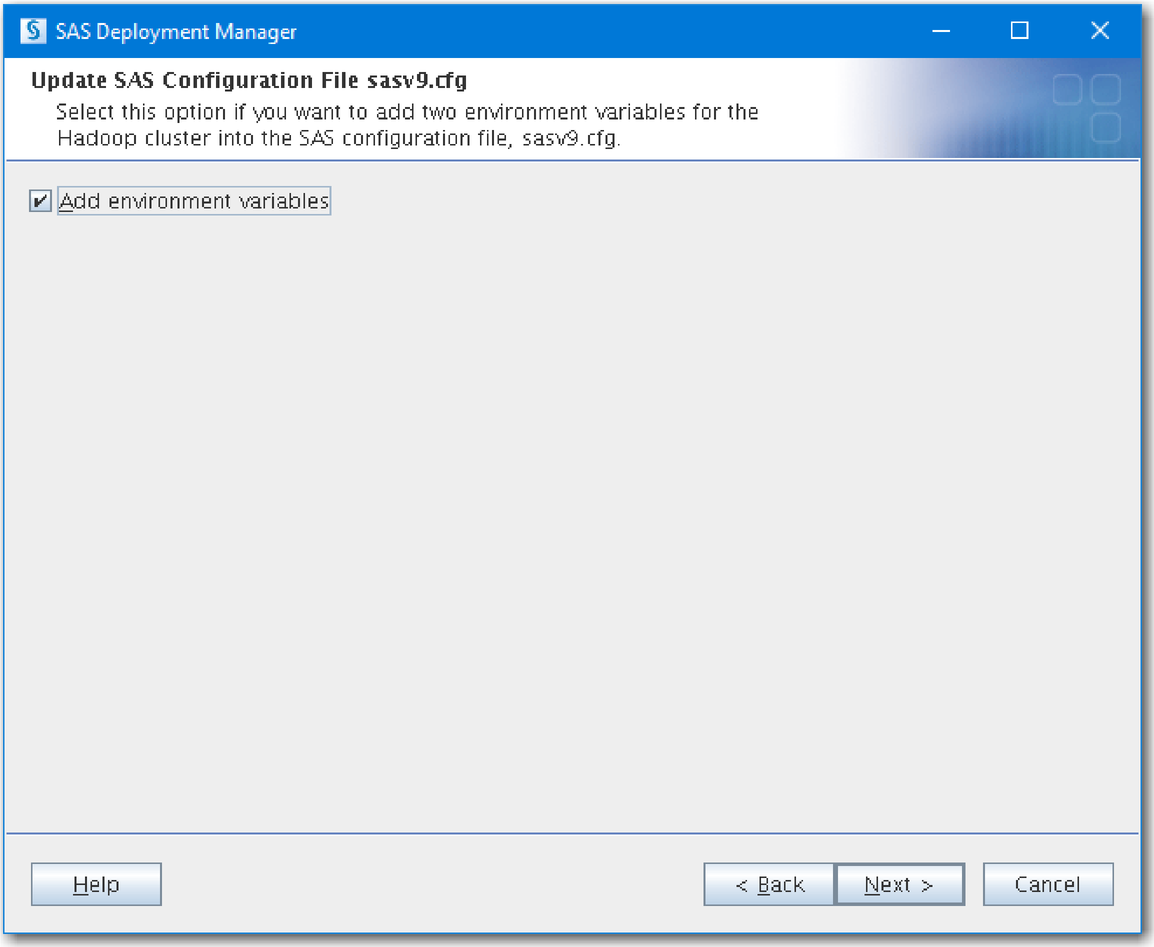
    Note: Each time this configuration process is run, the resulting files and libraries are stored in the paths provided here. This path could be a network path if multiple SAS servers are being configured to work with Hadoop.

    Note: For MapR distributions that need to use Pig, set **Filter JAR Files By** to No filter. This setting enables the correct JAR files to be pulled.

    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.



SAS Deployment Manager

Update SAS Configuration File sasv9.cfg

Select this option if you want to add two environment variables for the Hadoop cluster into the SAS configuration file, sasv9.cfg.

☑ Add environment variables

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

    The two environment variables are as follows:

    - SAS_HADOOP_CONFIG_PATH

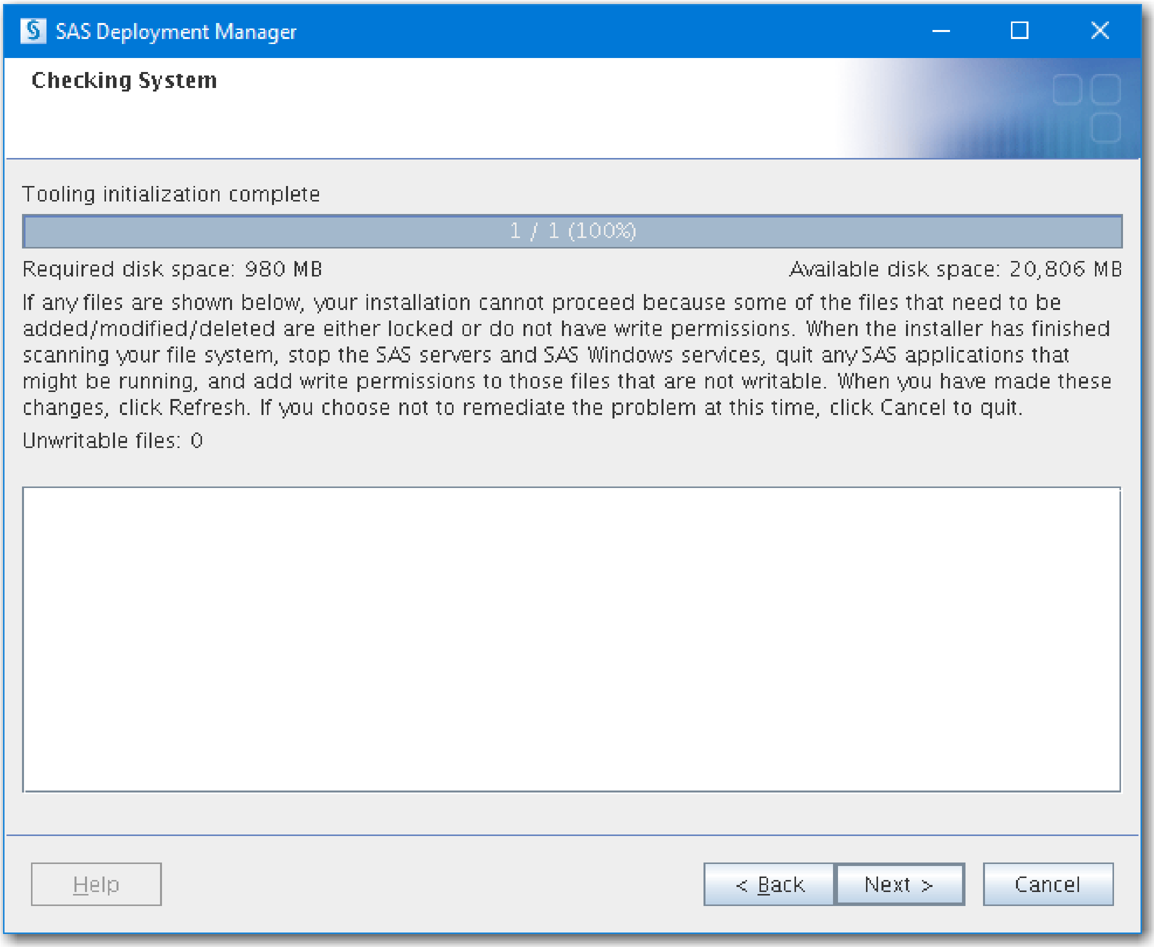
      This environment variable sets the location of the Hadoop cluster configuration files.

    - SAS_HADOOP_JAR_PATH

      This environment variable sets the location of the Hadoop JAR files.

    Click **Next**.

    If you are obtaining JAR and configuration files for Base SAS or SPD Engine, the **Checking System** page opens. Skip to Step 18 on page 49.



Tooling initialization complete

1 / 1 (100%)

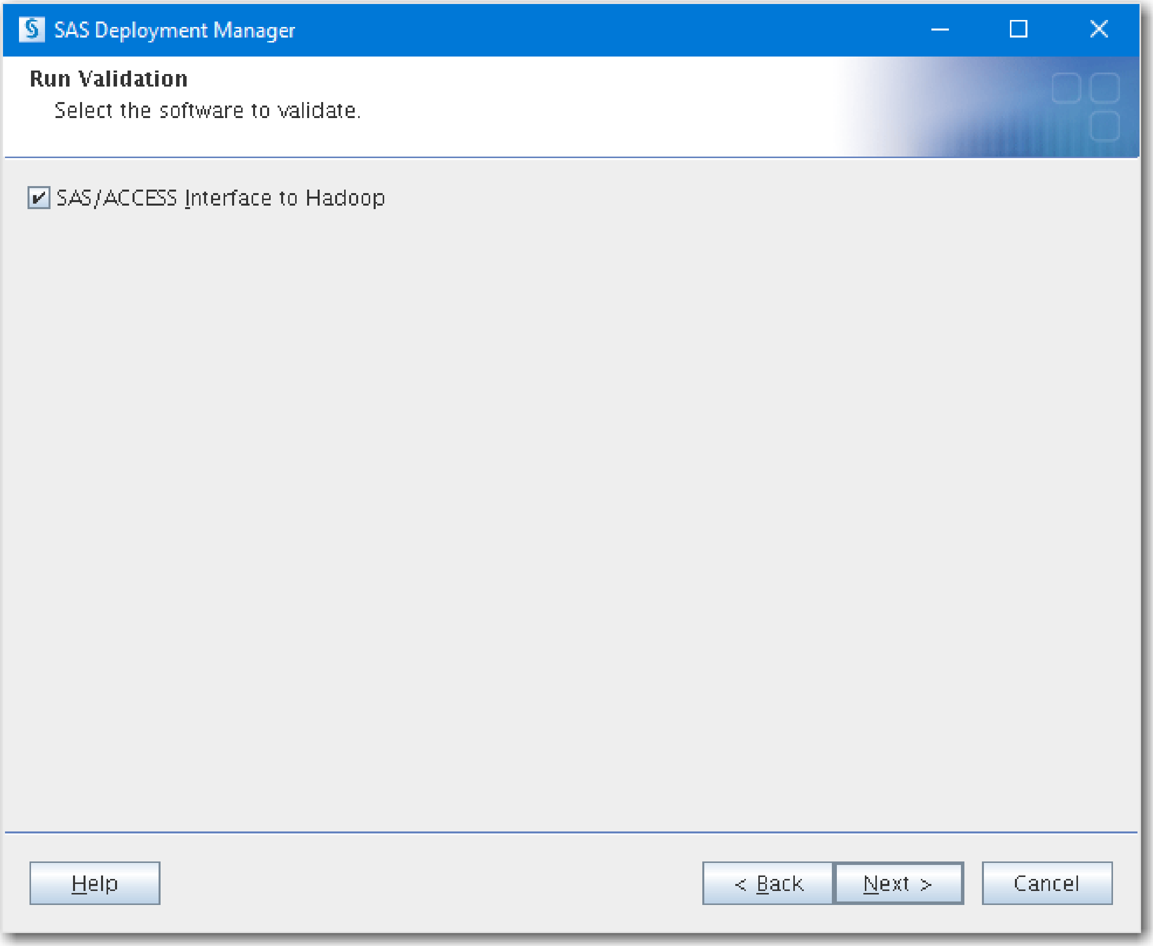
Required disk space: 980 MB

Available disk space: 20,806 MB

If any files are shown below, your installation cannot proceed because some of the files that need to be added/modified/deleted are either locked or do not have write permissions. When the installer has finished scanning your file system, stop the SAS servers and SAS Windows services, quit any SAS applications that might be running, and add write permissions to those files that are not writable. When you have made these changes, click Refresh. If you choose not to remediate the problem at this time, click Cancel to quit.

Unwritable files: 0

If you are obtaining JAR and configuration files for SAS/ACCESS, the **Run Validation** page opens.



Run Validation

Select the software to validate.

☑ SAS/ACCESS Interface to Hadoop

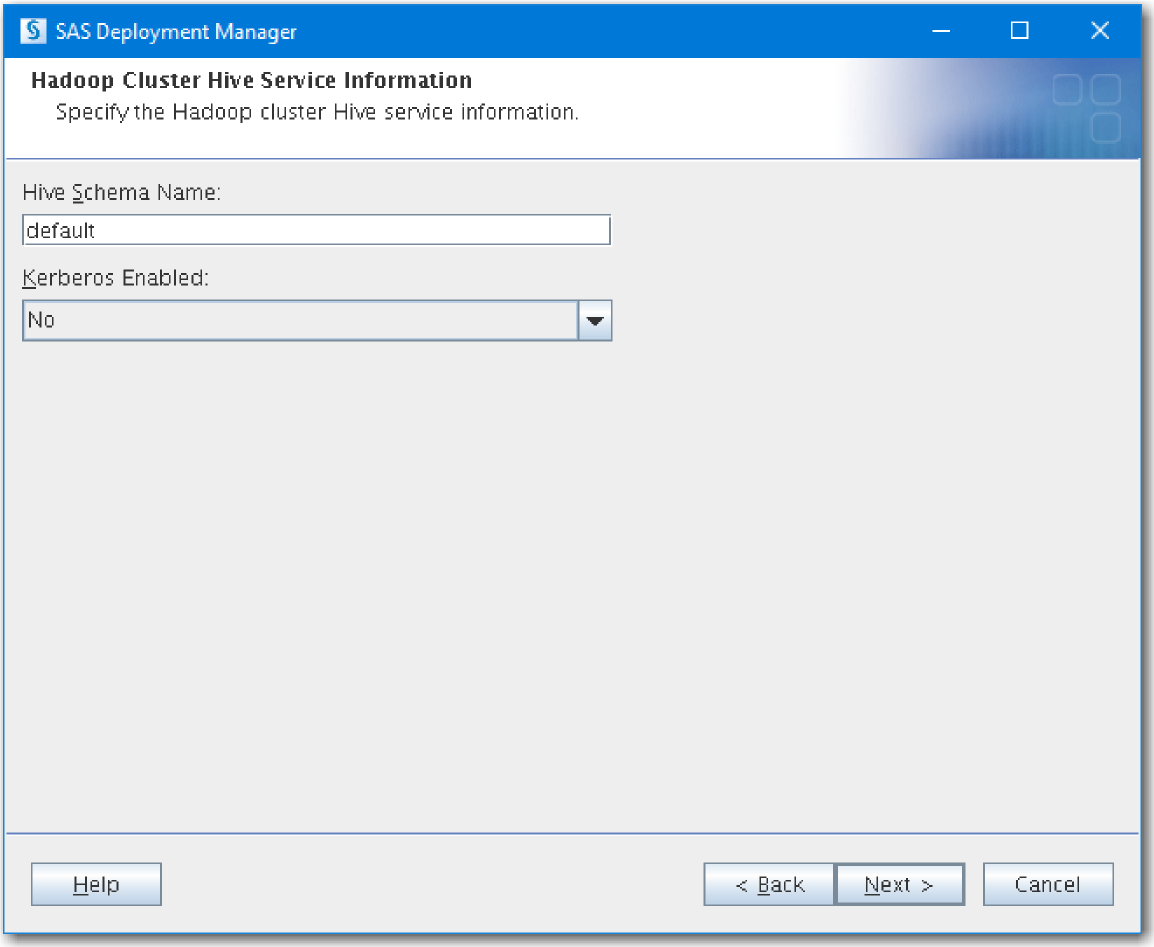
15. (Optional) Validate the configuration of SAS/ACCESS Interface to Hadoop.

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

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

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

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



Hadoop Cluster Hive Service Information

Specify the Hadoop cluster Hive service information.

Hive Schema Name: default

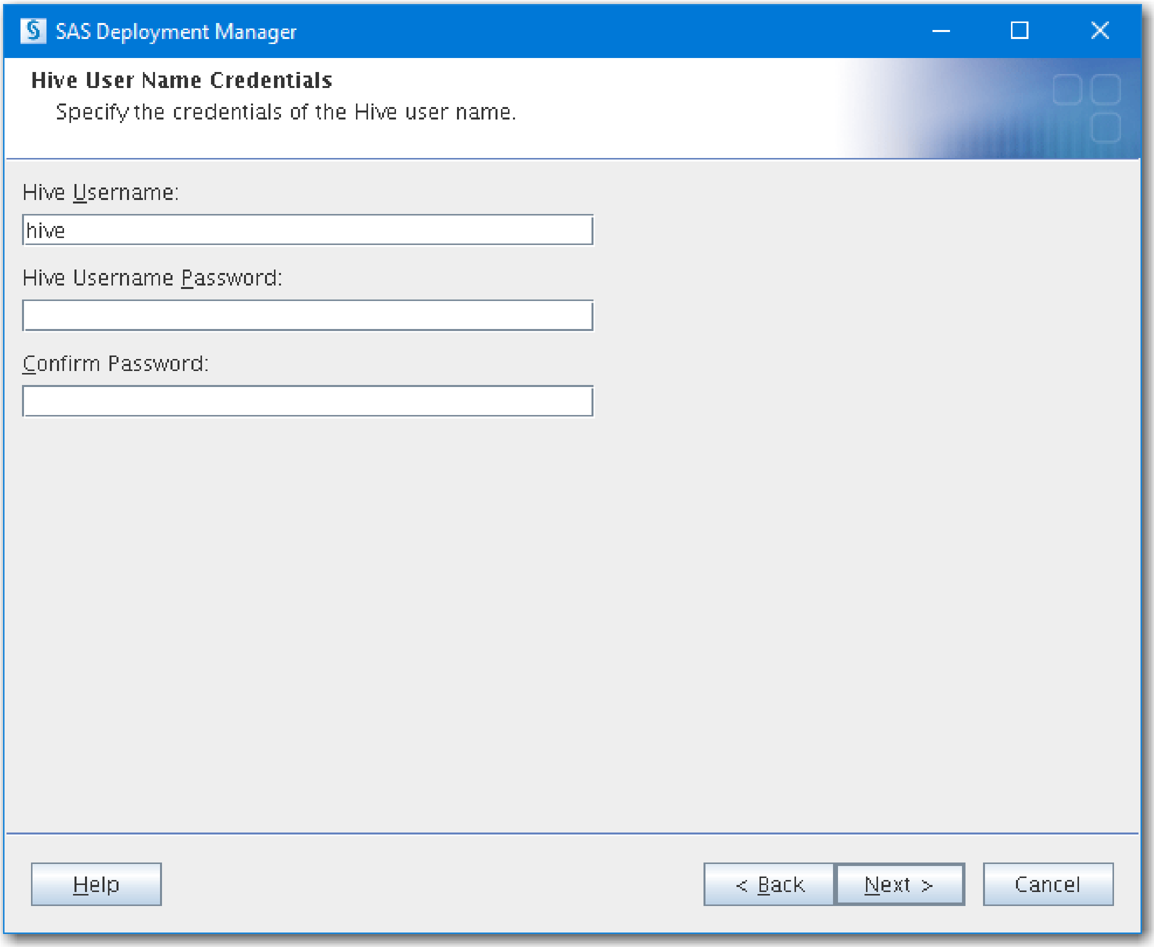
Kerberos Enabled: No

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

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

    Note: If you are using Advanced Encryption Standard (AES) encryption with Kerberos, you must manually add the Java Cryptography Extension local_policy.jar file in every location where JAVA Home resides on the cluster. If you are located outside the United States, you must also manually add the US_export_policy.jar file. The addition of these files is governed by the United States import control restrictions. These two JAR files also need to replace the existing local_policy.jar and US_export_policy.jar files in the SAS JRE location, which is the ***`SASHome`*** **`/SASPrivateJavaRuntimeEnvironment/9.4/jre/lib/security/`** directory. As a best practice, 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 websites.

    Click **Next**. The **Hive User Name Credentials** page opens.



Hive User Name Credentials

Specify the credentials of the Hive user name.

Hive Username: hive

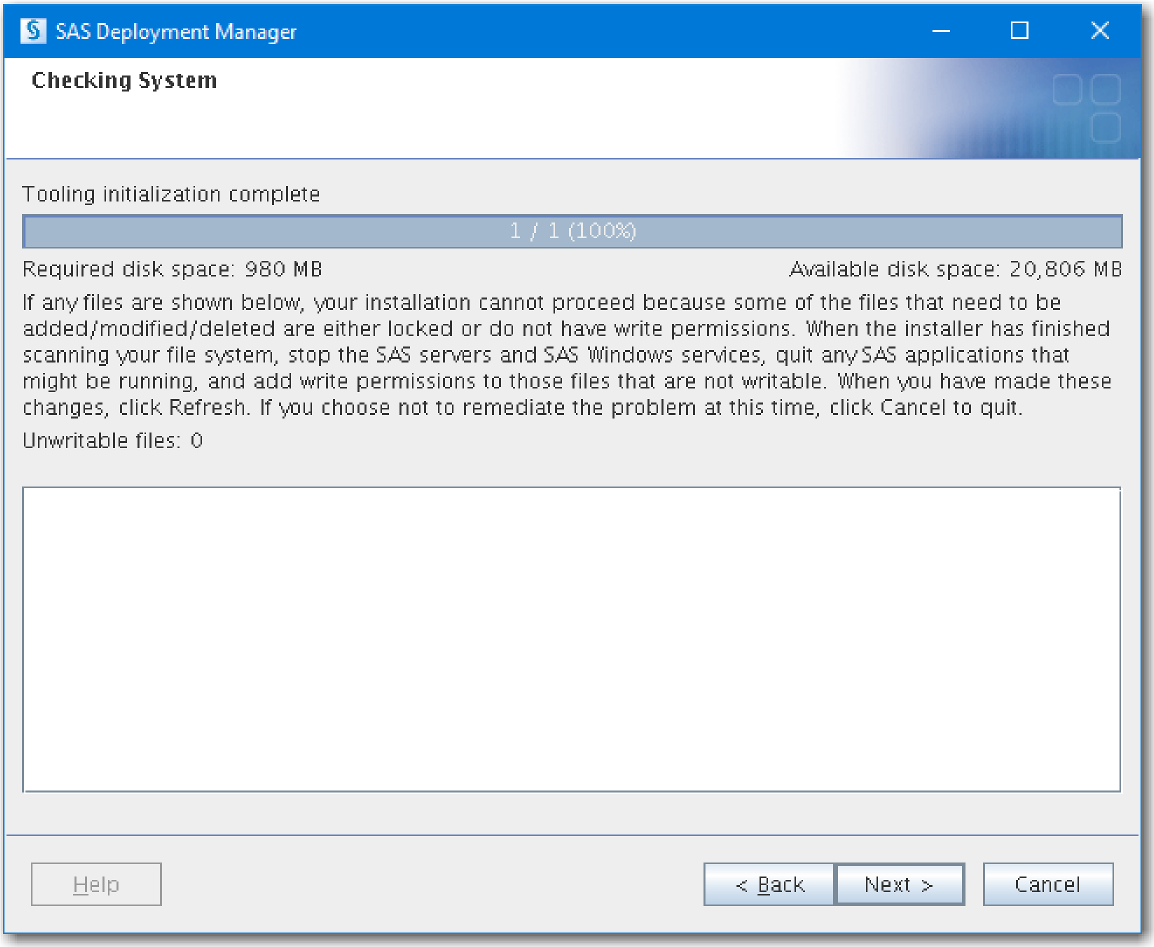
Hive Username Password:

Confirm Password:

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

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

    Click **Next**. SAS Deployment Manager verifies the prerequisites for the validation and checks for locked files and Write permissions. Checking the system might take several seconds. The **Checking System** page opens.



Checking System

Tooling initialization complete

1 / 1 (100%)

Required disk space: 980 MB

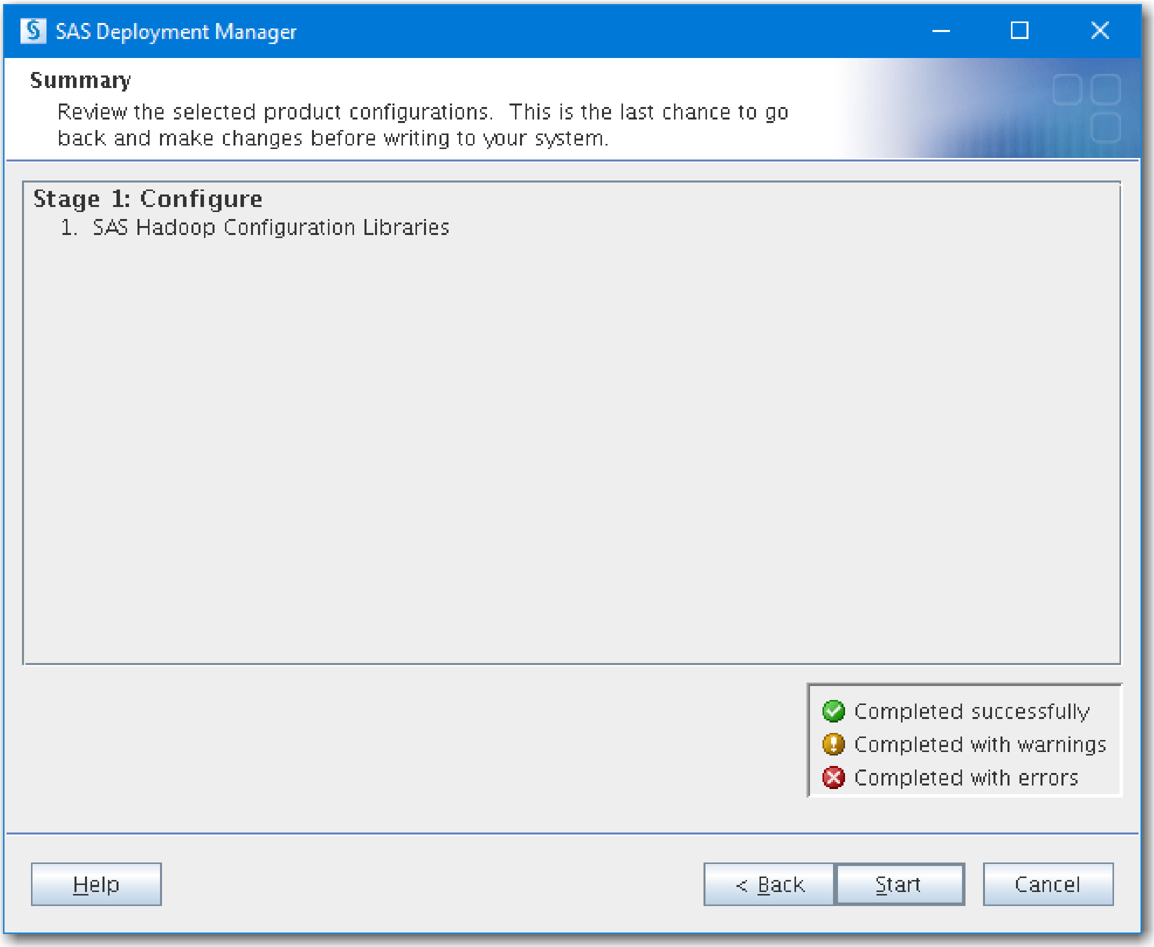
Available disk space: 20,806 MB

If any files are shown below, your installation cannot proceed because some of the files that need to be added/modified/deleted are either locked or do not have write permissions. When the installer has finished scanning your file system, stop the SAS servers and SAS Windows services, quit any SAS applications that might be running, and add write permissions to those files that are not writable. When you have made these changes, click Refresh. If you choose not to remediate the problem at this time, click Cancel to quit.

Unwritable files: 0

18. If any files are shown in the text box after the system check, follow the instructions on the **Checking System** page to fix any problems.

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



Summary

Review the selected product configurations. This is the last chance to go back and make changes before writing to your system.

Stage 1: Configure

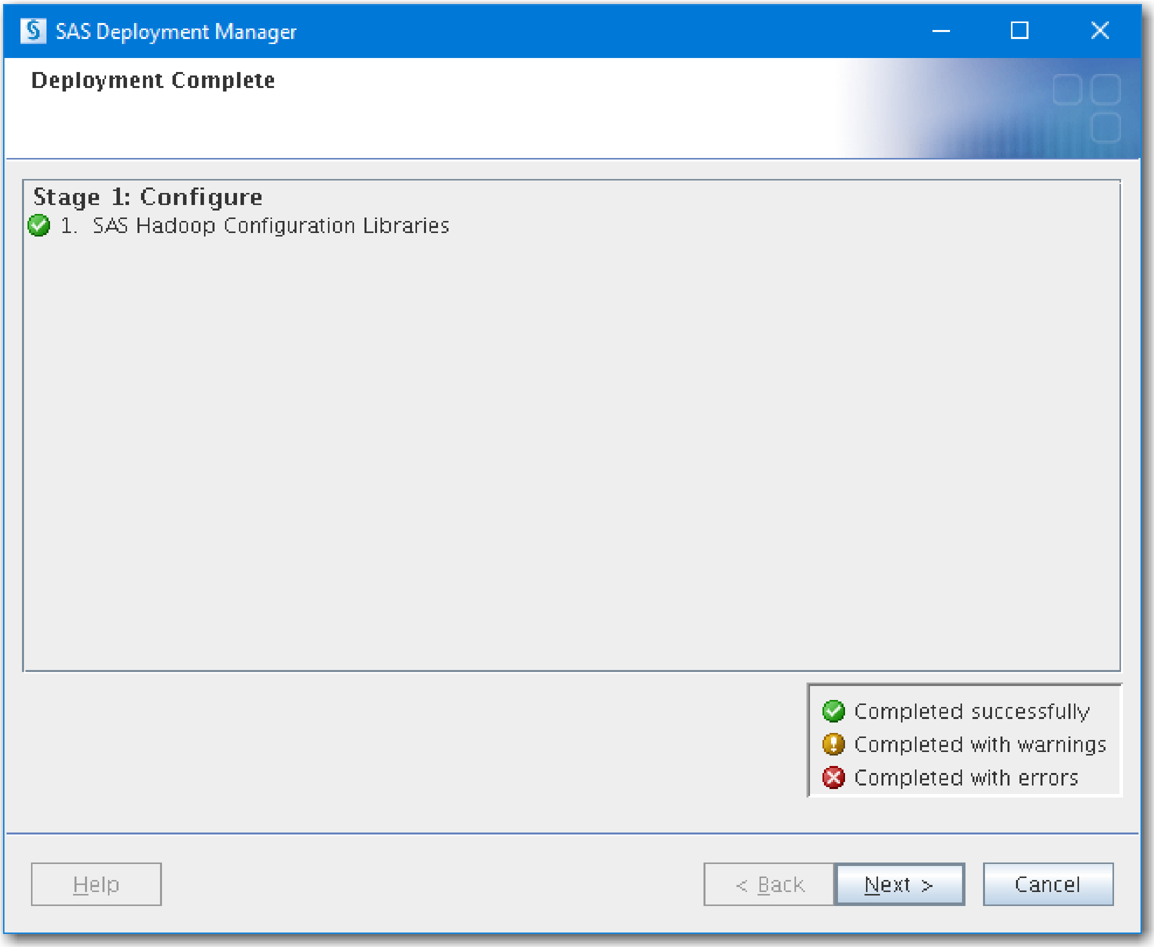
1. SAS Hadoop Configuration Libraries

✓ Completed successfully
! Completed with warnings
✗ Completed with errors

19. Click **Start** to begin the configuration.

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

    If the configuration is successful, the name of the page changes to **Deployment Complete** and a green check mark is displayed beside SAS Hadoop Configuration Libraries.



Deployment Complete

Stage 1: Configure

✓ 1. SAS Hadoop Configuration Libraries

✓ Completed successfully
! Completed with warnings
✗ Completed with errors

Note: Part of the configuration process runs SAS code to validate the environment. A green check mark indicates that SAS Deployment Manager could connect to Hadoop, run a tracer script, pull back files, and run SAS code to validate the setup.

If warnings or errors occur, fix the issues and restart the configuration.

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

Location of Original JAR and Configuration Files after a Redeployment

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

On a Windows client:

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

On a UNIX client:

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

service-name is either **hive** or **oozie**.

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

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

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

Supporting Multiple Hadoop Versions and Upgrading Hadoop Version

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

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

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

When to Collect New JAR and Configuration Files

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

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

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

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

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

Appendix 2
SAS Environment Variables for Hadoop

Dictionary

KNOX_GATEWAY_URL Environment Variable

Sets the location of the Apache Knox Gateway.

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

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

See: "Using Apache Knox Gateway Security" on page 8

Syntax

KNOX_GATEWAY_URL *URL*

Required Argument

URL

specifies the HTTPS URL for the Knox Gateway website. The URL is site specific. The format of the Knox gateway URL is as follows:

```
https://gateway-host:gateway-port/gateway-pathname/cluster-name
```

For example, the following OPTIONS statement syntax sets the environment variable:

```
options set=KNOX_GATEWAY_URL "https://server:port/gateway/default";
```

Details

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

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

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

Table A2.1 *Defining the KNOX_GATEWAY_URL Environment Variable*

Operating Environment	Method	Example
UNIX *	SAS configuration file	**-set KNOX_GATEWAY_URL "https://*server*:*port*/gateway/default"**
	SAS invocation	**-set KNOX_GATEWAY_URL "https://*server*:*port*/gateway/default"**
	OPTIONS statement	**options https://*server*:*port*/gateway/default";**
Windows	SAS configuration file	**-set KNOX_GATEWAY_URL "https://*server*:*port*/gateway/default"**
	SAS invocation	**-set KNOX_GATEWAY_URL "https://*server*:port/gateway/default"**
	OPTIONS statement	**options https://*server*:*port*/gateway/default";**

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

SAS_HADOOP_CONFIG_PATH Environment Variable

Sets the location of the Hadoop cluster configuration files.

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

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

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

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

Syntax

SAS_HADOOP_CONFIG_PATH *pathname*

Required Argument

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 per Hadoop cluster. 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 A2.2 *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";`**

Operating Environment	Method	Example
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.

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

Syntax

SAS_HADOOP_JAR_PATH *pathname(s)*

Required Argument

pathname(s)

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

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

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

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

```
options set=SAS_HADOOP_JAR_PATH="C:\third_party\Hadoop\jars\lib;
   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 A2.3 *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 Hadoop Version” on page 52.

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 A2.4 *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