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

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®*. Cary, NC: SAS Institute Inc.

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

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

February 2015

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.

- Ensure that the HDFS, MapReduce, YARN, and Hive services are running on the Hadoop cluster.
- 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.
- 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.
  - 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 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).

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

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 WebHDFS.

For more information, see "Configuring Hadoop JAR Files" on page 7 and "Using WebHDFS" on page 8.

3 Create a configuration file and make it available to the SAS client machine.

For more information, see "Making Hadoop Cluster Configuration Files Available to the SAS Client Machine" 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 11.

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 more information, see "Supported Hadoop Distributions for the FILENAME Statement and PROC HADOOP" on page 6.

3 Install Base SAS by following the instructions in your software order e-mail.

Supported Hadoop Distributions for the FILENAME Statement and PROC HADOOP

In the second maintenance release for SAS 9.4, the following Hadoop distributions are supported for the FILENAME statement and PROC HADOOP:

- Cloudera CDH 4.5x

  Cloudera CDH 5.x
- Hortonworks HDP 1.3.x

  Hortonworks HDP 2.x
- IBM InfoSphere BigInsights 2.1
- MapR 3.1
- Pivotal HD 2.0.1

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.

Configuring 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 4.5. 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 47 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 58.

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 42.

Note: To submit HDFS commands, you can also connect to the Hadoop server by using WebHDFS. WebHDFS is an HTTP REST API that supports the complete FileSystem interface for HDFS. Using WebHDFS removes the need for client-side JAR files for HDFS, but Pig JAR files are still needed. For more information, see "Using WebHDFS" on page 8.

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

WebHDFS is an HTTP REST API that supports the complete FileSystem interface for HDFS.

Note that using WebHDFS removes the need for client-side JAR files for HDFS, but to submit MapReduce programs or Pig language programs with PROC HADOOP, JAR files are still needed. To use WebHDFS 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 60.

2 Make sure the configuration file includes the properties for the WebHDFS location, which are the dfs.http.address or **`dfs.namenode.http-address`** properties. If the **`dfs.http.address`** 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 a configuration file with properties for a WebHDFS location:

```
<configuration>
<property>
   <name>fs.default.name</name>
   <value>hdfs://caesar.unx.sas.com:8020</value>
 </property>
 <property>
   <name>dfs.http.address</name>
   <value>caesar.unx.sas.com:50070</value>
 </property>
 <property>
   <name>dfs.namenode.http-address</name>
   <value>caesar.unx.sas.com:50070</value>
 </property>
 <property>
   <name>mapred.job.tracker</name>
   <value>caesar.unx.sas.com:8021</value>
 </property>
</configuration>
```

3 In the FILENAME statement or PROC HADOOP statement, identify the configuration file with the CFG= option.

For more information about the configuration file, see "Making Hadoop Cluster Configuration Files Available to the SAS Client Machine" on page 9.

Making Hadoop Cluster Configuration Files Available to the SAS Client Machine

To connect to a Hadoop server with the FILENAME statement or PROC HADOOP, a single configuration file must 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.

Note: The FILENAME statement and PROC HADOOP do not support the SAS_HADOOP_CONFIG_PATH environment variable to set the location of the configuration file. You must use the CFG= option.

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*
- Chapter 29, “HADOOP Procedure” in *Base SAS Procedures Guide*

4

Configuring SAS/ACCESS for Hadoop

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 the 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 15.

3 Review security and user access.

For more information, see "Security and User Access to Hadoop" on page 34.

4 Review Hive and HiveServer2 requirements.

For more information, see "Working with Hive and HiveServer2" on page 35.

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 33.

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 more information, see "Supported Hadoop Distributions for SAS/ACCESS Interface to Hadoop" on page 14.

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

Supported Hadoop Distributions for SAS/ACCESS Interface to Hadoop

In the second maintenance release for SAS 9.4, the following Hadoop and Hive combinations are supported for SAS/ACCESS Interface to Hadoop:

Cloudera CDH4.5.x
Cloudera CDH5

Hortonworks HDP 1.3.x

Hortonworks HDP 2.x

IBM InfoSphere BigInsights 2.1

MapR 3.1

Pivotal HD 2.0.1

Note: If a vendor assures upward compatibility, SAS/ACCESS supports newer combinations.

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.

Configuring Hadoop JAR and Configuration Files

Using the 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 the SAS Deployment Manager to make required Hadoop JAR and configuration files available to the SAS client machine. The SAS Deployment Manager, a tool that enables you to perform some administrative and configuration tasks, is included with each SAS software order. The 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. WebHDFS is an HTTP REST API that supports the complete FileSystem interface for HDFS. Using WebHDFS removes the need for client-side JAR files for HDFS, but Hive JAR files are still needed. For more information, see "Using WebHDFS" on page 32.

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
exit
```

  Here is an example:

```
sudo su - root
su - hdfs
kinit -kt hdfs.keytab hdfs
exit
```

Note:

Note: If you are running on a cluster with Kerberos, a keytab is required for the HDFS user who configures the Hadoop JAR and configuration files.

Note: You can run klist while you are running as the -hdfsuser user to check the status of your Kerberos ticket on the server. Here is an example:

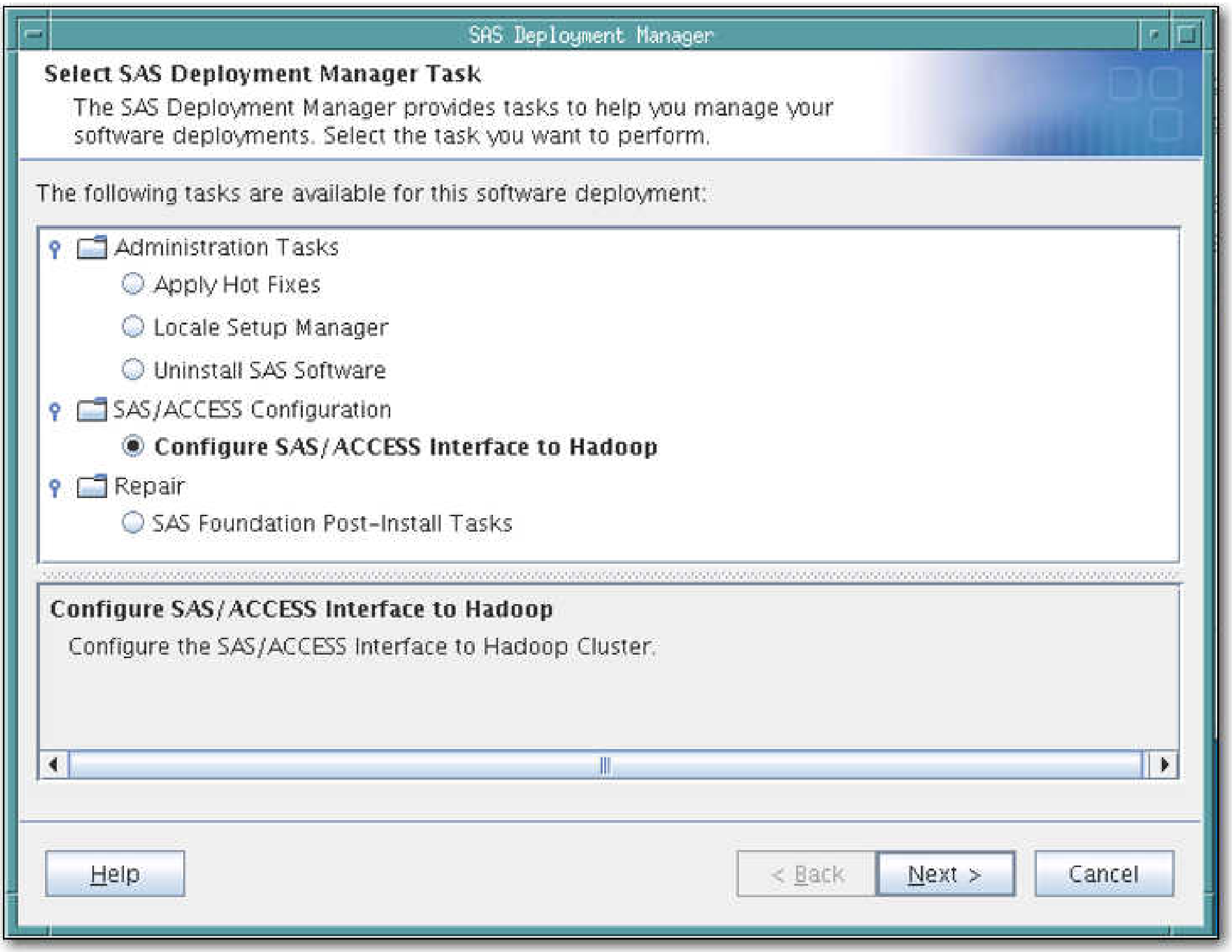
```
klist
Ticket cache: FILE/tmp/krb5cc_493
Default principal: hdfs@HOST.COMPANY.COM

Valid starting    Expires           Service principal
06/20/14 09:51:26 06/27/14 09:51:26 krbtgt/HOST.COMPANY.COM@HOST.COMPANY.COM
     renew until 06/22/14 09:51:26
```

2 Start the SAS Deployment Manager by running sasdm.exe for Windows or sashm.sh for UNIX.

The **Choose Language** page opens.

3 Select the language in which you want to perform the configuration of your software.

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



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:

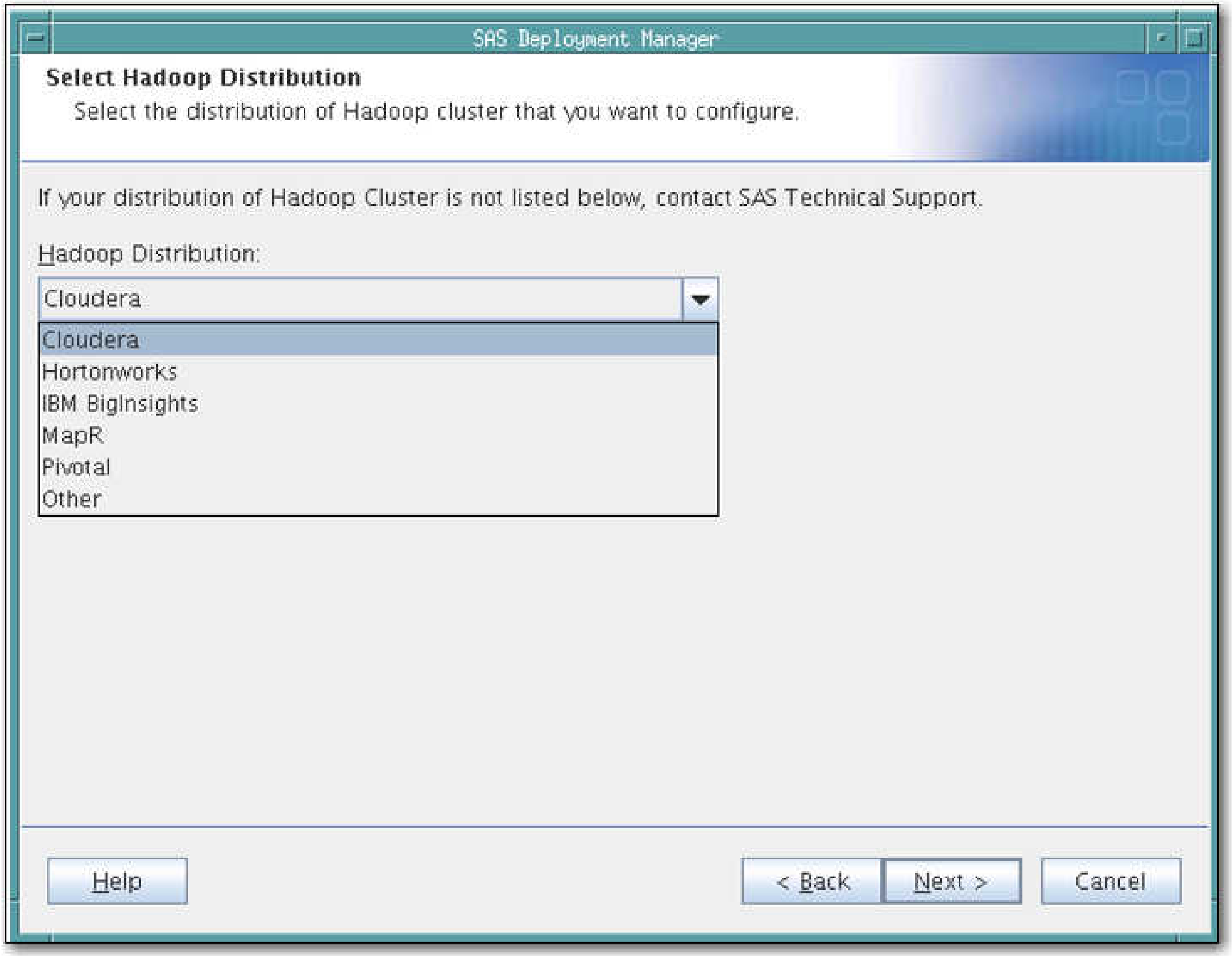
- Administration Tasks
  - ○ Apply Hot Fixes
  - ○ Locale Setup Manager
  - ○ Uninstall SAS Software
- SAS/ACCESS Configuration
  - ● **Configure SAS/ACCESS Interface to Hadoop**
- Repair
  - ○ SAS Foundation Post-Install Tasks

Configure SAS/ACCESS Interface to Hadoop

Configure the SAS/ACCESS Interface to Hadoop Cluster.

4 Under **SAS/ACCESS Configuration**, select **Configure SAS/ACCESS Interface to Hadoop**.

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



SAS Deployment Manager

Select Hadoop Distribution

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

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

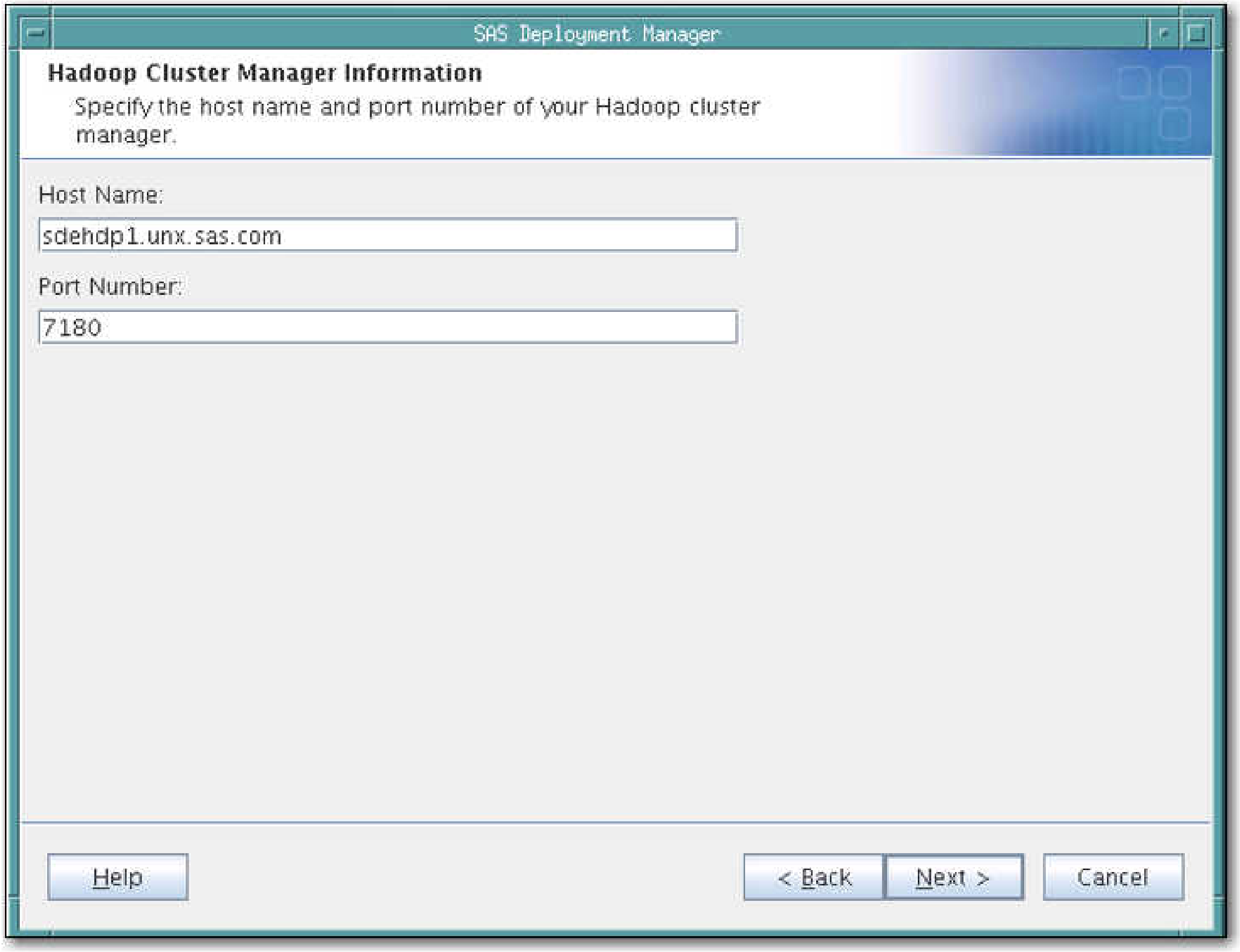
Hadoop Distribution: Cloudera

5 From the drop-down menu, select the distribution of Hadoop that you are using.

Note: If your distribution is not listed, exit the SAS Deployment Manager and contact SAS Technical Support.

Click **Next**.

If your distribution has an administrative client such as Cloudera Manager or Ambari, the **Hadoop Cluster Manager Information** page opens. Continue with Step 6 on page 19.



Hadoop Cluster Manager Information

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

Host Name: sdehdp1.unx.sas.com

Port Number: 7180

If your distribution does not have an administrative client, the **Hadoop Cluster Hive Node Information** page opens. Skip to Step 9 on page 22.

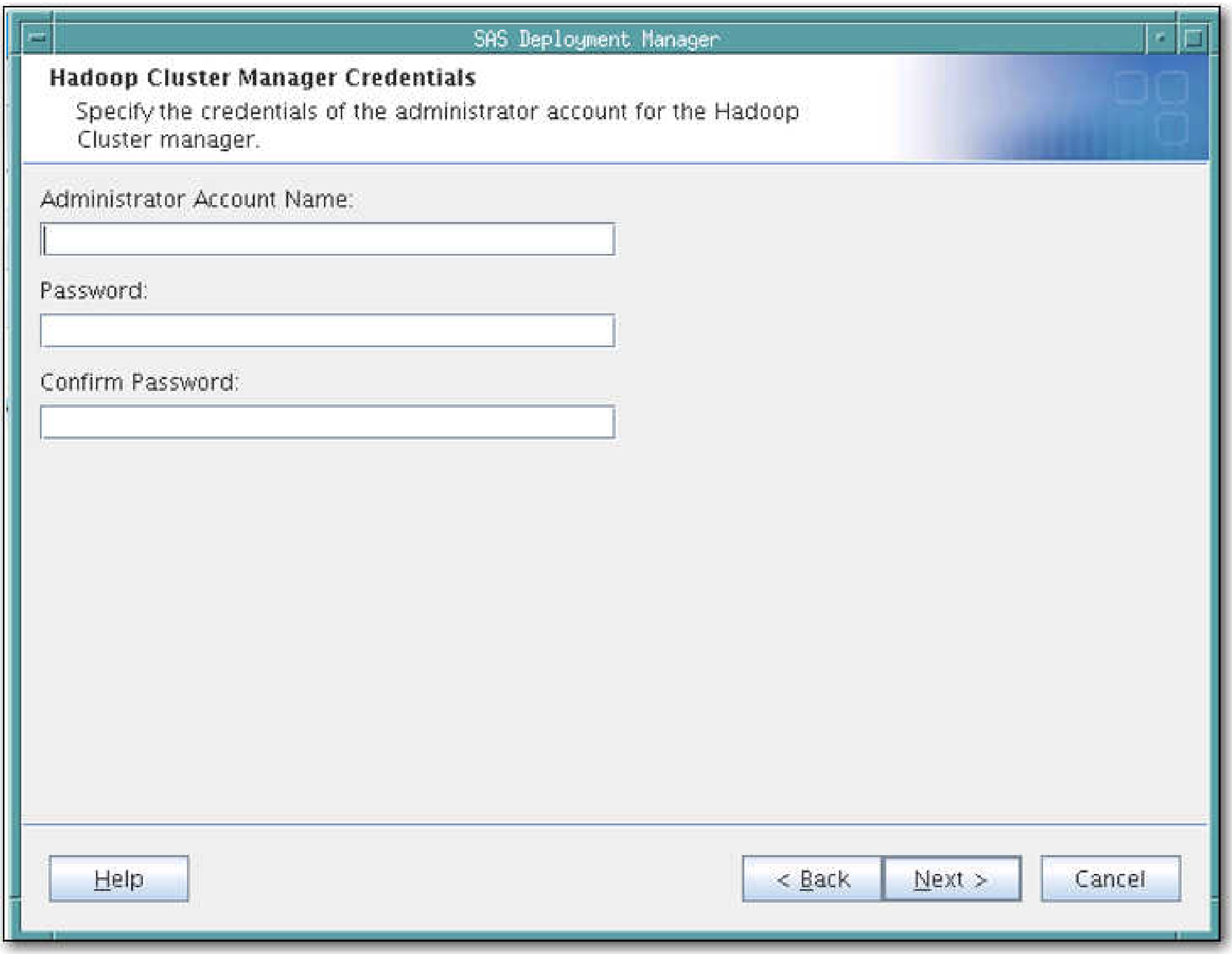
6 Enter the host name and port number for your Hadoop cluster.

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.



SAS Deployment Manager

Hadoop Cluster Manager Credentials

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

Administrator Account Name:

Password:

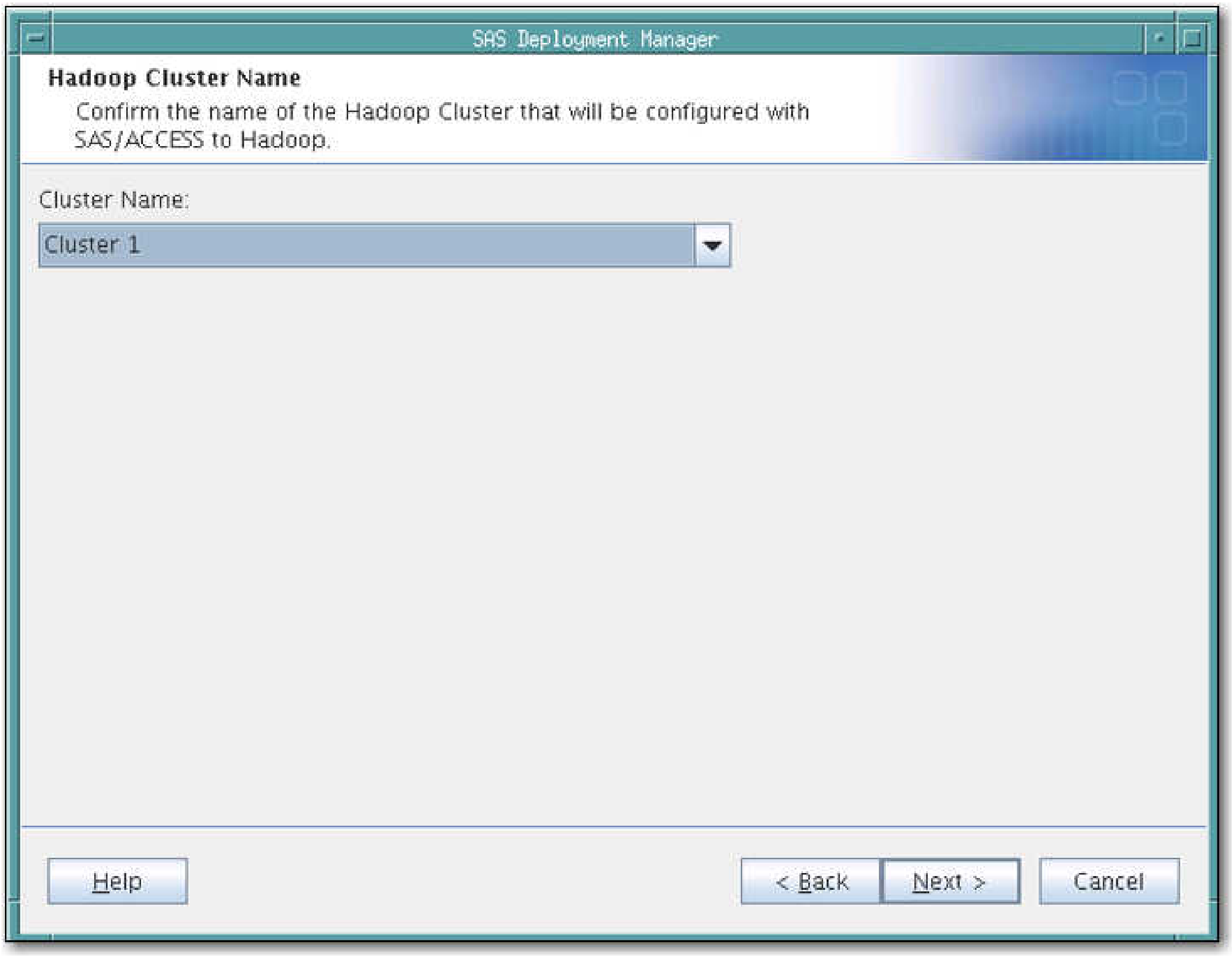
Confirm Password:

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

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. However, the account name does not have to be that of an administrator; it can be a read-only user.

Click **Next**.

If you are configuring a Cloudera cluster, the **Hadoop Cluster Name** page opens. Continue with Step 8 on page 21.



Hadoop Cluster Name

Confirm the name of the Hadoop Cluster that will be configured with SAS/ACCESS to Hadoop.

Cluster Name: Cluster 1

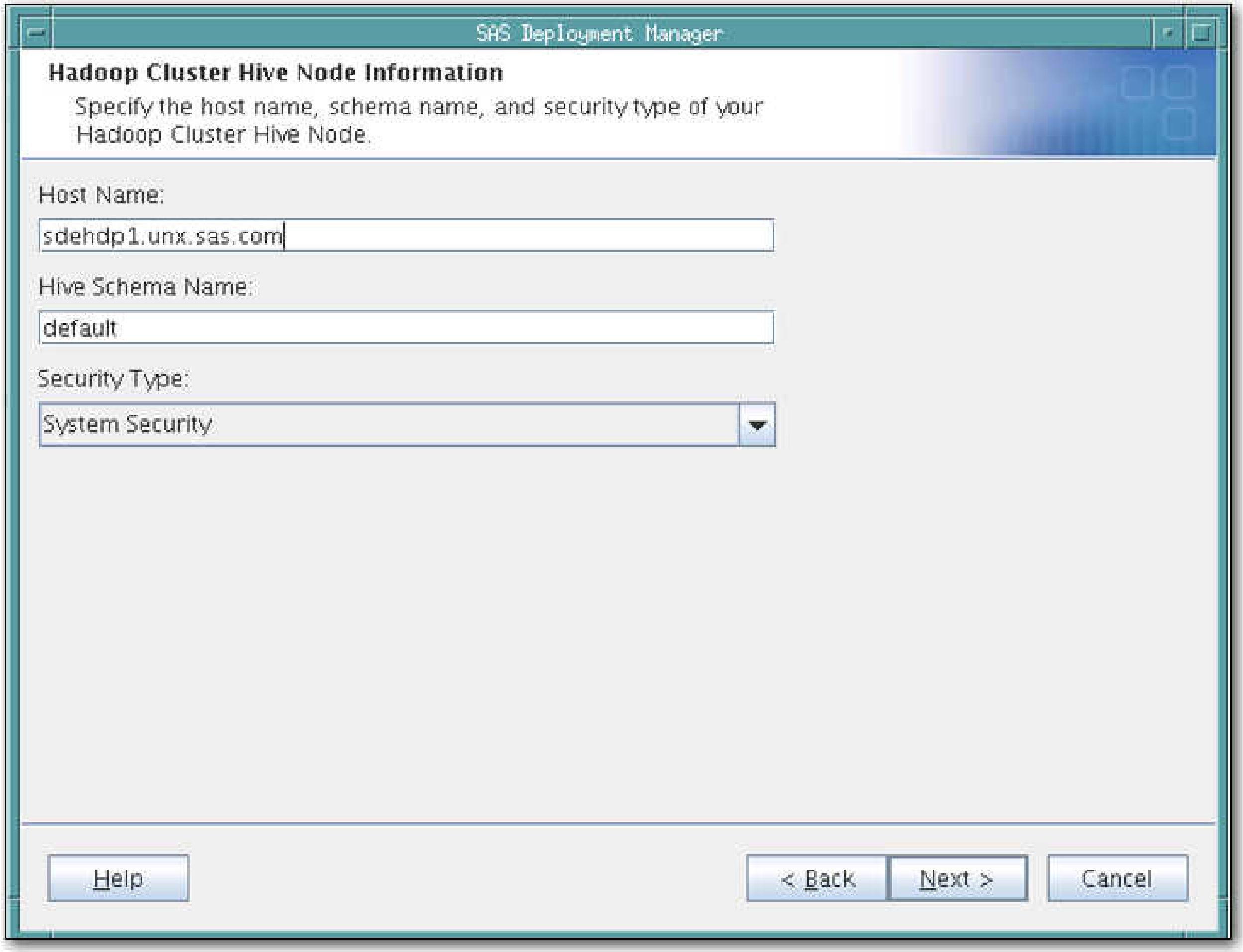
If you are configuring a Hortonworks cluster, the **Hadoop Cluster Hive Node Information** page opens. Skip to Step 9 on page 22.

Note: Ambari can manage only one cluster at a time.

8 From the drop-down menu, select the cluster name that you want to use to connect SAS/ACCESS to Hadoop.

Note: Multiple clusters can be managed with Cloudera Manager. Select which cluster you want to connect to.

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



SAS Deployment Manager

Hadoop Cluster Hive Node Information

Specify the host name, schema name, and security type of your Hadoop Cluster Hive Node.

Host Name: sdehdp1.unx.sas.com

Hive Schema Name: default

Security Type: System Security

9 Enter the Hive node information for the Hadoop cluster.

a Enter the Hive host name.

Note: For Cloudera and Hortonworks, this field is automatically populated based on your input in the previous steps. If the Hive host name that is displayed is not correct, return to Step 4 on page 17 and select **Other**.

b Enter the Hive schema name.

c Choose the security type:

- System Security

  Select this option if your Hadoop distribution uses a security protocol other than Kerberos.

- Kerberos

  Select this option if your cluster is Kerberized.

- No Credentials Available

  Select this option if you do not have the credentials to allow the SAS Deployment Manager to configure Hadoop.

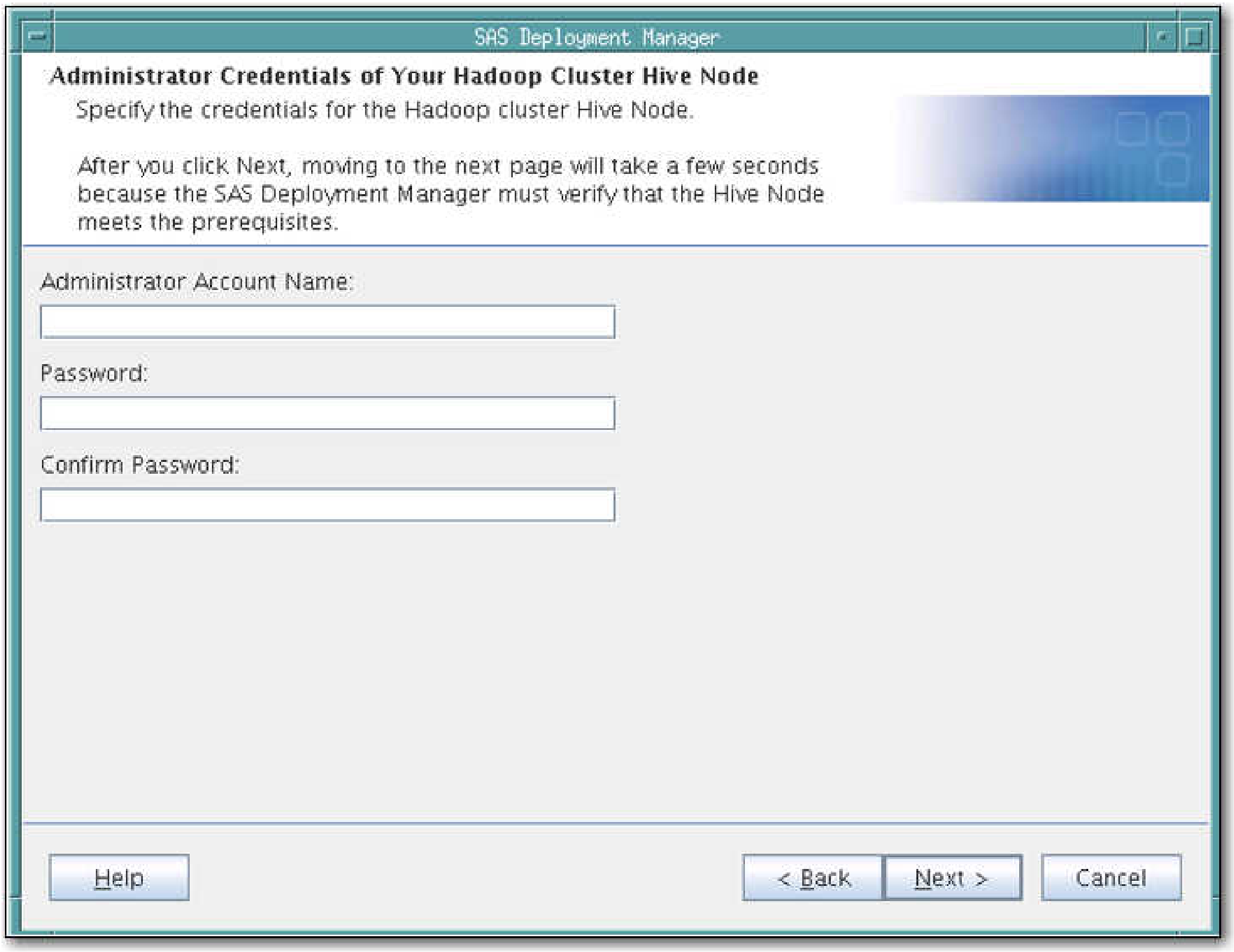
Click **Next**.

If you chose No Credentials Available, the SAS and Hadoop administrators must work together to complete the configuration manually. The **Required Manual Steps to Continue Configuring Hadoop** page opens. Skip to Step 12 on page 25.

If you chose System Security or Kerberos, the SAS Deployment Manager ensures that SAS can connect to the system, that the prerequisites are on the machine, and that you can run the software that is needed. These are the prerequisites:

- The HDFS `/tmp` exists.
- The HDFS `/tmp` is not specified for a sticky bit (-t).
- The Hadoop, hive, and pig commands exist and are in the path.
- A Kerberos ticket is available if Kerberos security is selected.
- strace is installed.

The **Administrator Credentials of Your Hadoop Cluster Hive Node** page opens. Continue with Step 10 on page 23.



SAS Deployment Manager

Administrator Credentials of Your Hadoop Cluster Hive Node

Specify the credentials for the Hadoop cluster Hive Node.

After you click Next, moving to the next page will take a few seconds because the SAS Deployment Manager must verify that the Hive Node meets the prerequisites.

Administrator Account Name:

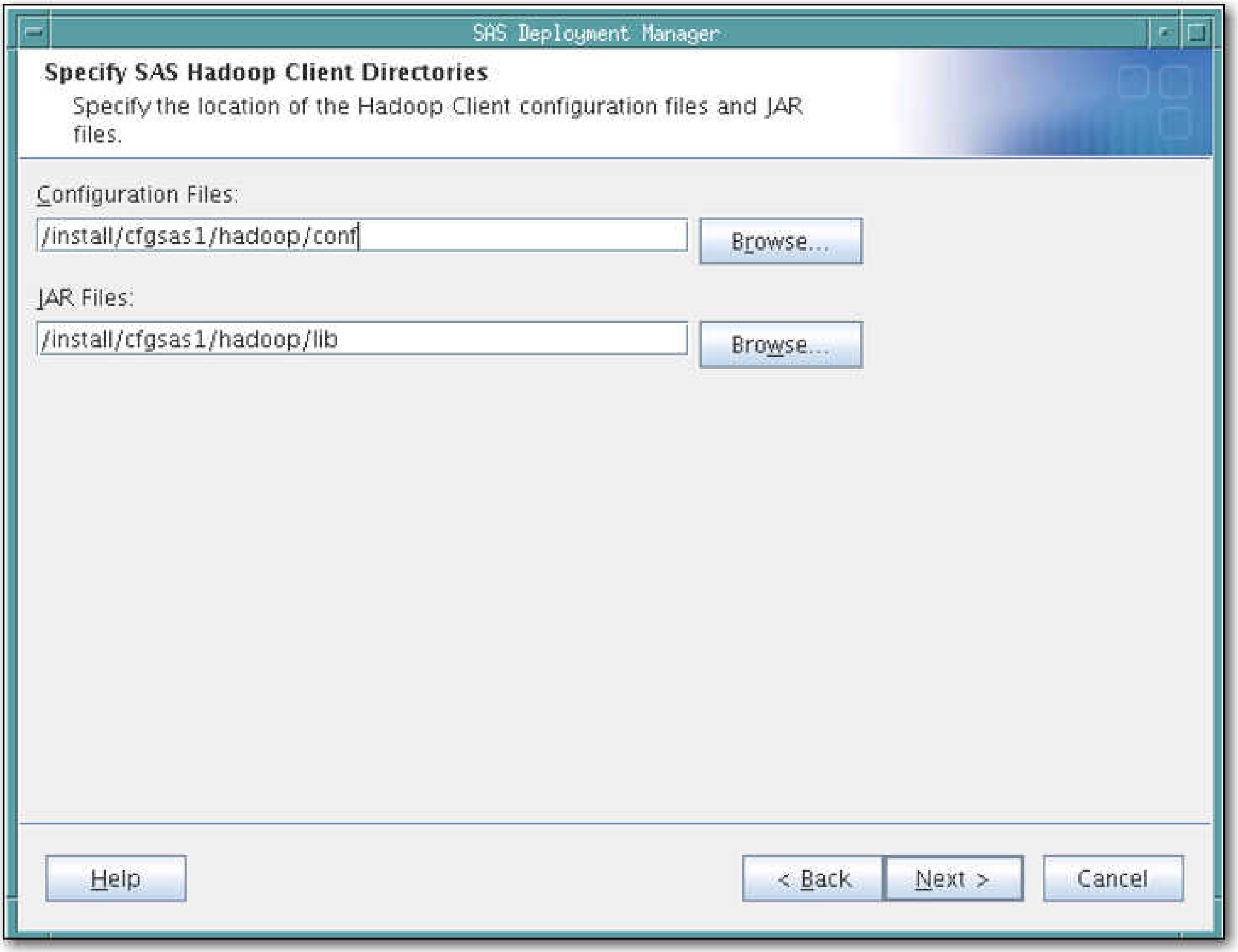
Password:

Confirm Password:

10 Enter the administrator account name and password.

The administrator account name is a local system user for the host that was identified on the **Hadoop Cluster Hive Node Information** page in the previous step.

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



SAS Deployment Manager

Specify SAS Hadoop Client Directories

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

Configuration Files: /install/cfgsas1/hadoop/conf

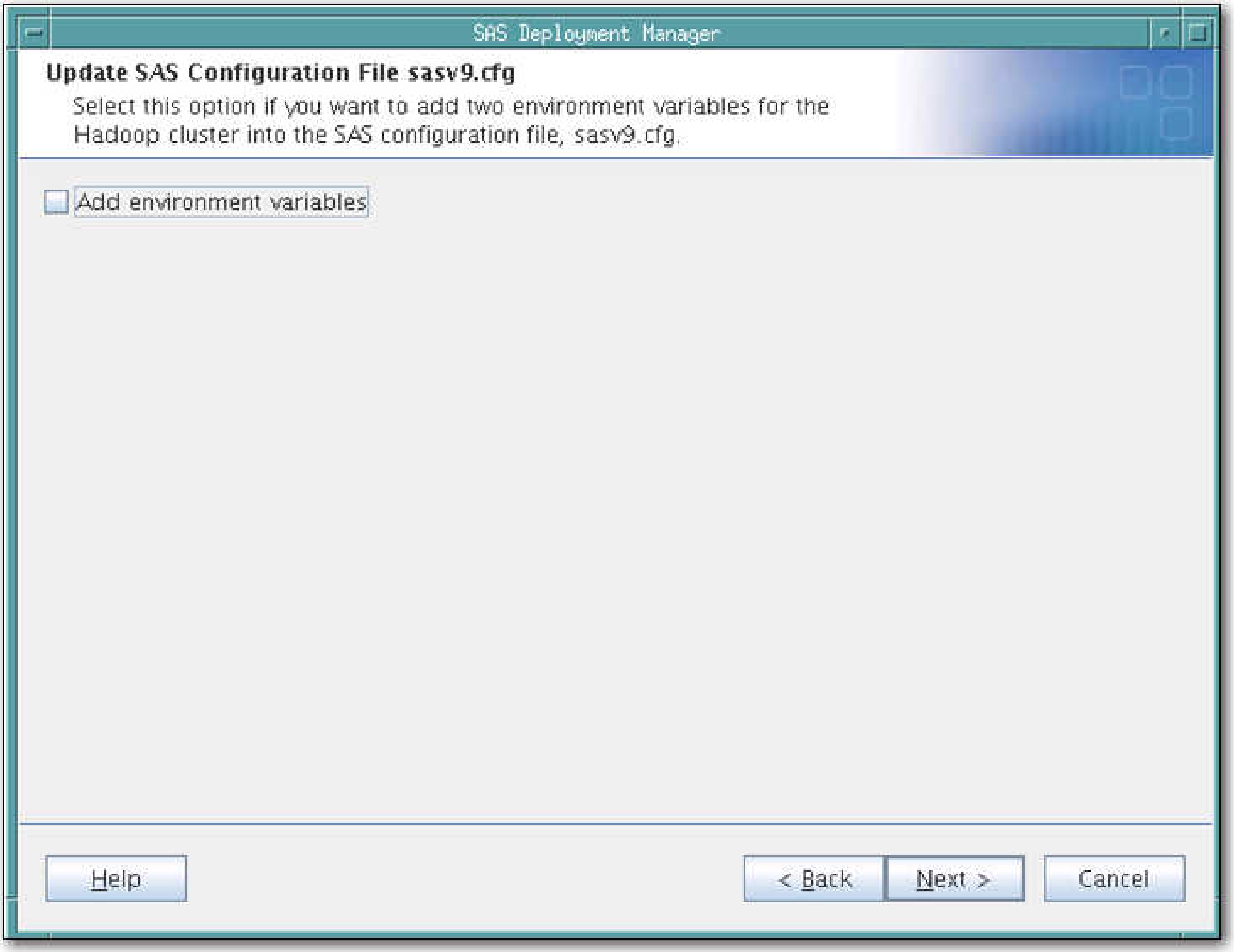
JAR Files: /install/cfgsas1/hadoop/lib

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. The directory you select must already exist. This step does not 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 could be a network path if multiple SAS servers are being configured to work with Hadoop.

Click **Next**. The **Update SAS Configuration File sasv9.cfg** page opens. Skip to Step 13 on page 27.

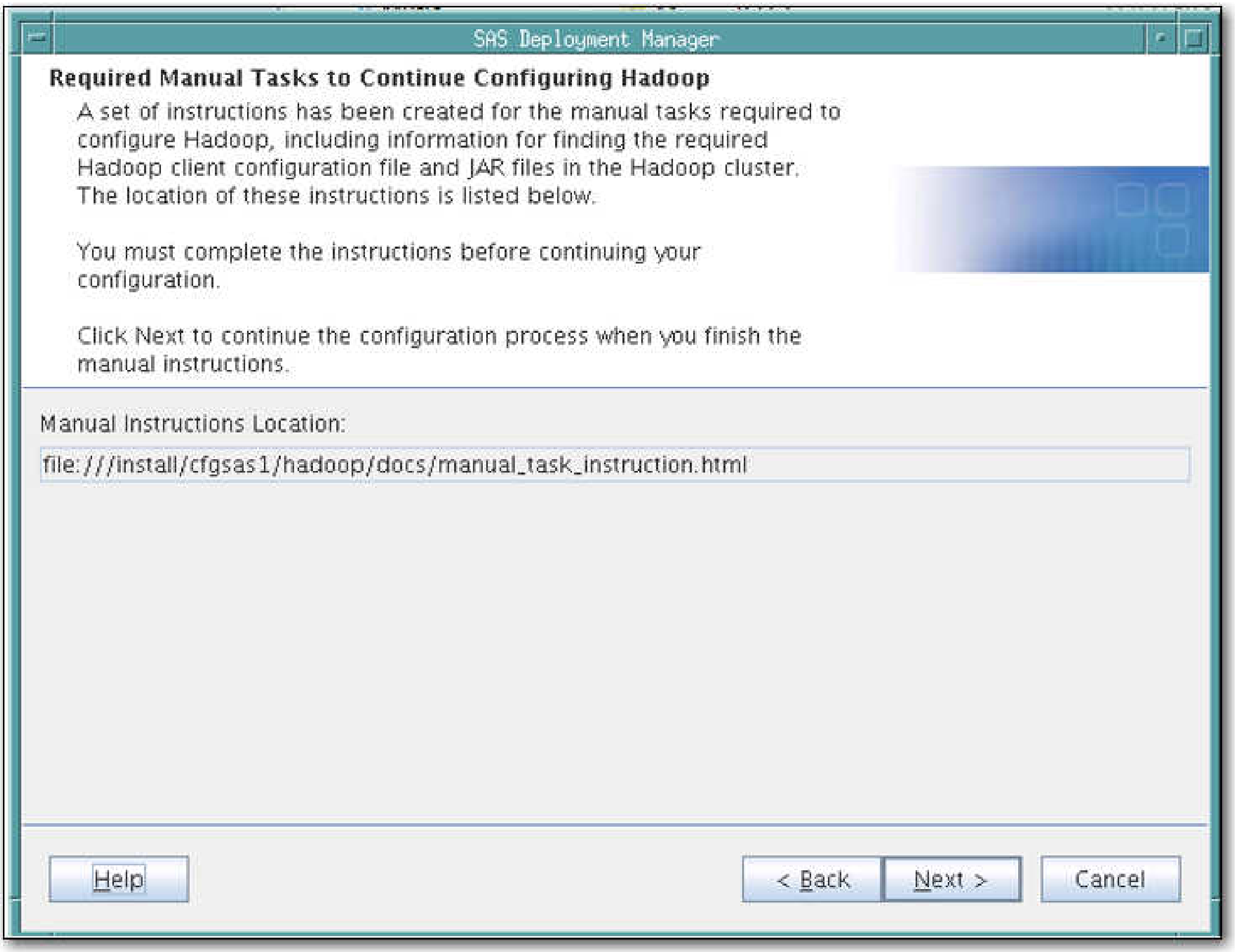


Update SAS Configuration File sasv9.cfg

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

12 If you indicated in Step 9c that you did not have the credentials to allow the SAS Deployment Manager to gather the Hadoop Hive node information, you must gather that information manually. The **Required Manual Tasks to Continue Configuring Hadoop** page provides the location of the instructions required to manually collect the needed configuration files and JAR files. Copy and paste this location into a browser window.



Required Manual Tasks to Continue Configuring Hadoop

A set of instructions has been created for the manual tasks required to configure Hadoop, including information for finding the required Hadoop client configuration file and JAR files in the Hadoop cluster. The location of these instructions is listed below.

You must complete the instructions before continuing your configuration.

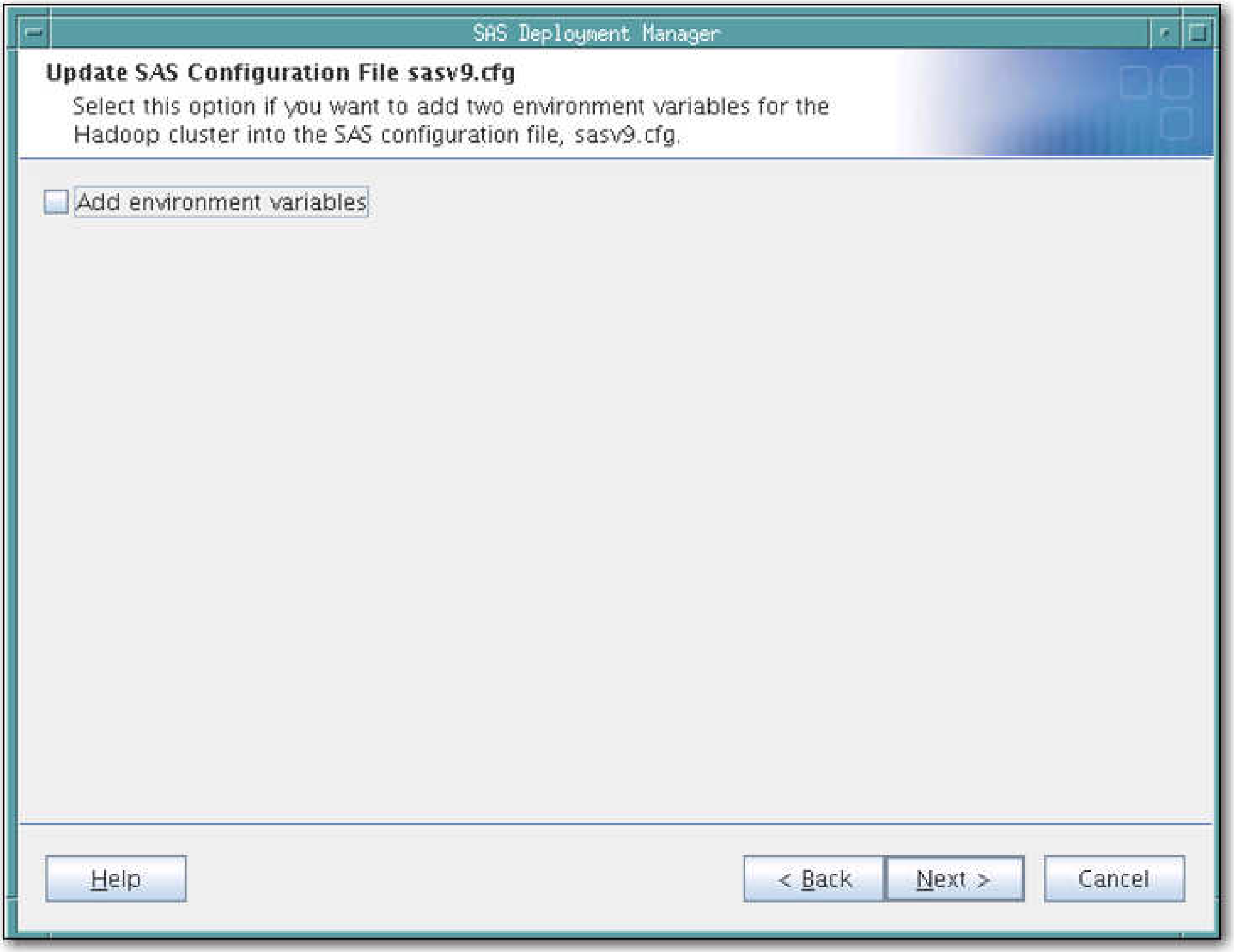
Click Next to continue the configuration process when you finish the manual instructions.

Manual Instructions Location:

file:///install/cfgsas1/hadoop/docs/manual_task_instruction.html

TIP When you have completed the instructions for manual configuration, note the location of the configuration files and JAR files for use after the SAS Deployment Manager closes.

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

13 If you want the SAS Deployment Manager to define two Hadoop cluster environment variables to the SAS configuration file, sasv9.cfg, select this option. Otherwise, leave it deselected.

The two configuration variable are as follows:

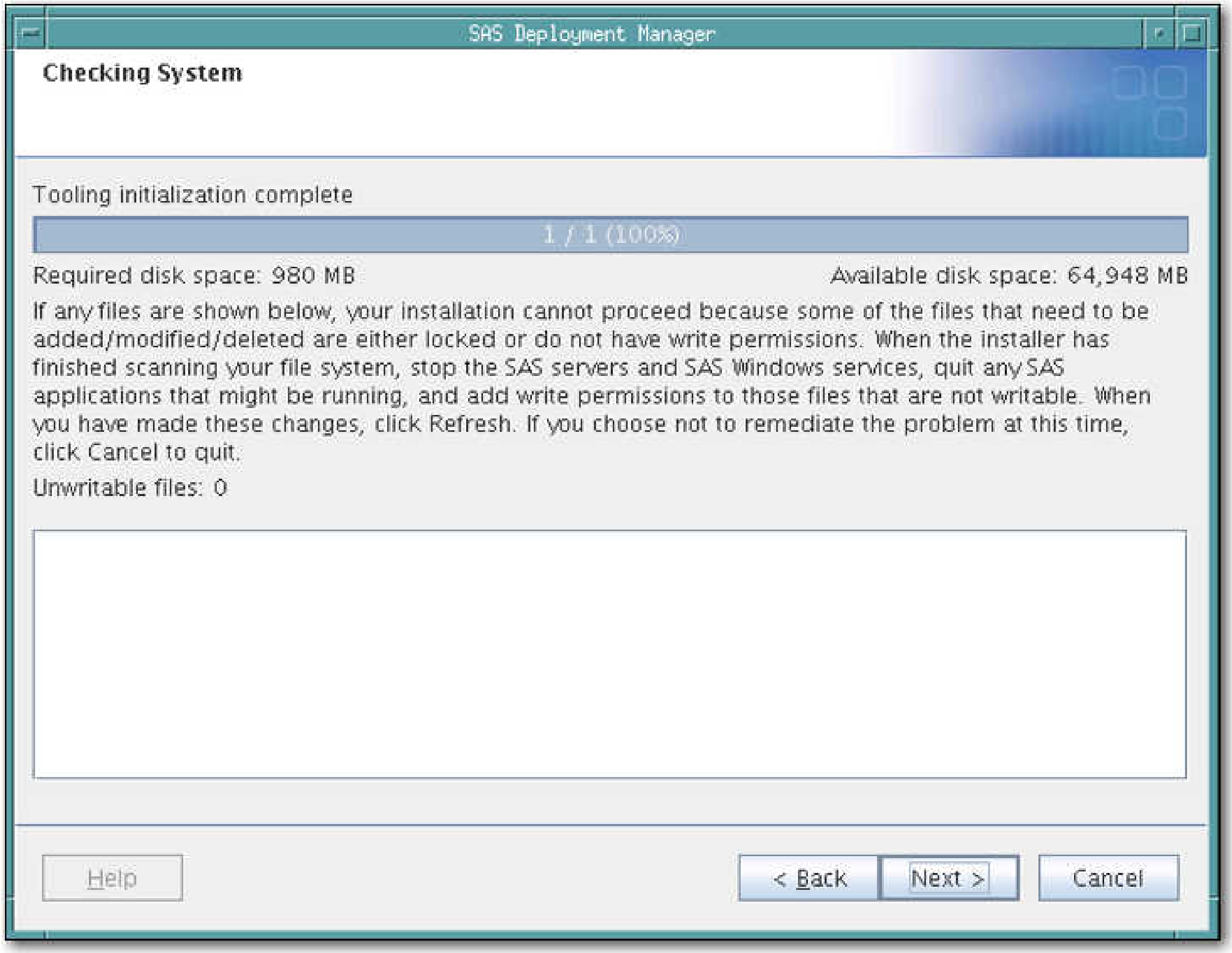
- SAS_HADOOP_CONFIG_PATH

  This environment variable sets the location of the Hadoop cluster configuration files. The environment variable is used by SAS/ACCESS Interface to Hadoop and the SPD Engine.

- SAS_HADOOP_JAR_PATH

  This environment variable sets the location of the Hadoop JAR files. The environment variable is used by the FILENAME statement Hadoop access method, HADOOP procedure, SAS/ACCESS Interface to Hadoop, and the SPD Engine.

Click **Next**. The **Checking System** page opens and a check for locked files and Write permissions is performed.



Checking System

Tooling initialization complete

1 / 1 (100%)

Required disk space: 980 MB

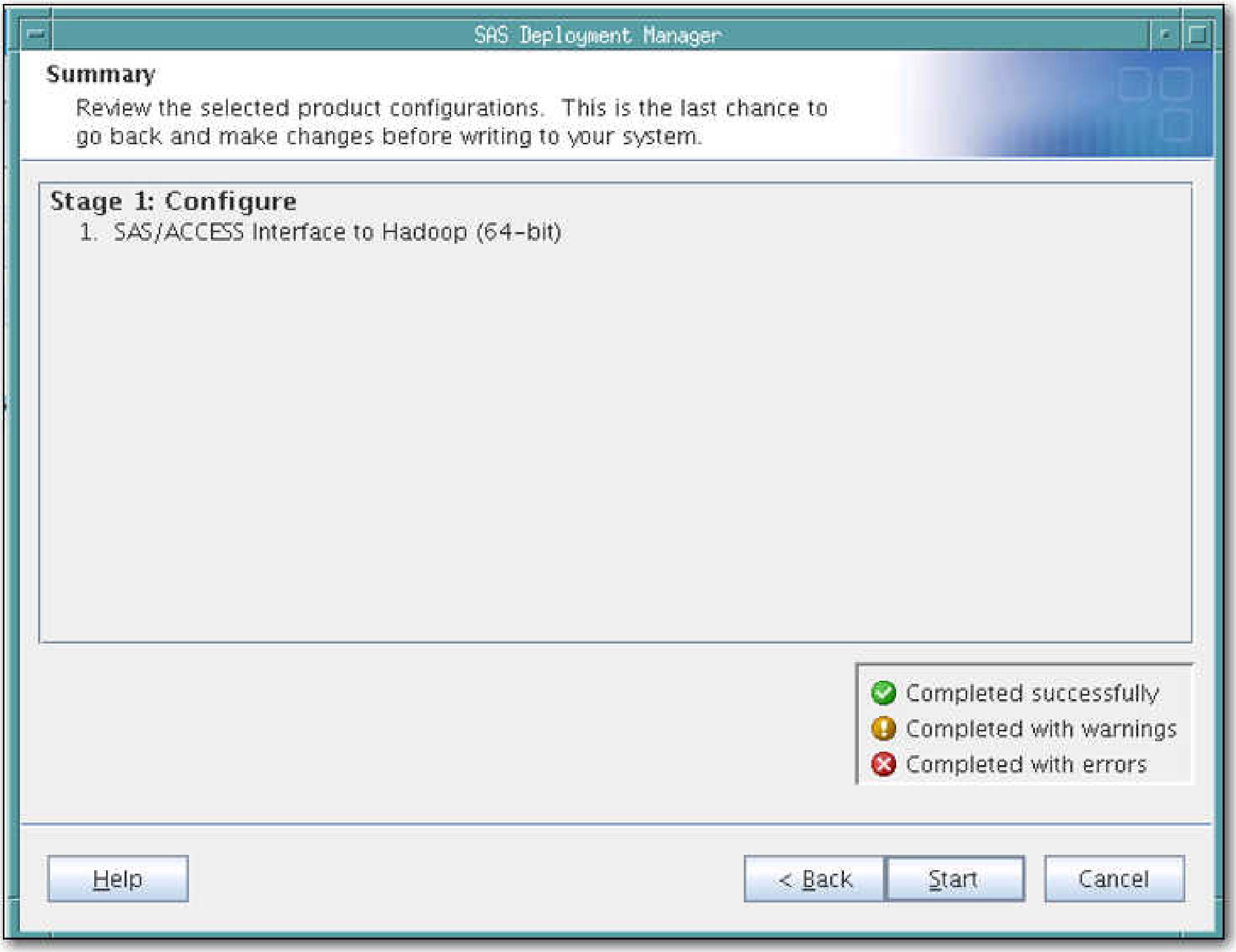
Available disk space: 64,948 MB

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

14 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.



SAS Deployment Manager

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/ACCESS Interface to Hadoop (64-bit)

✓ Completed successfully

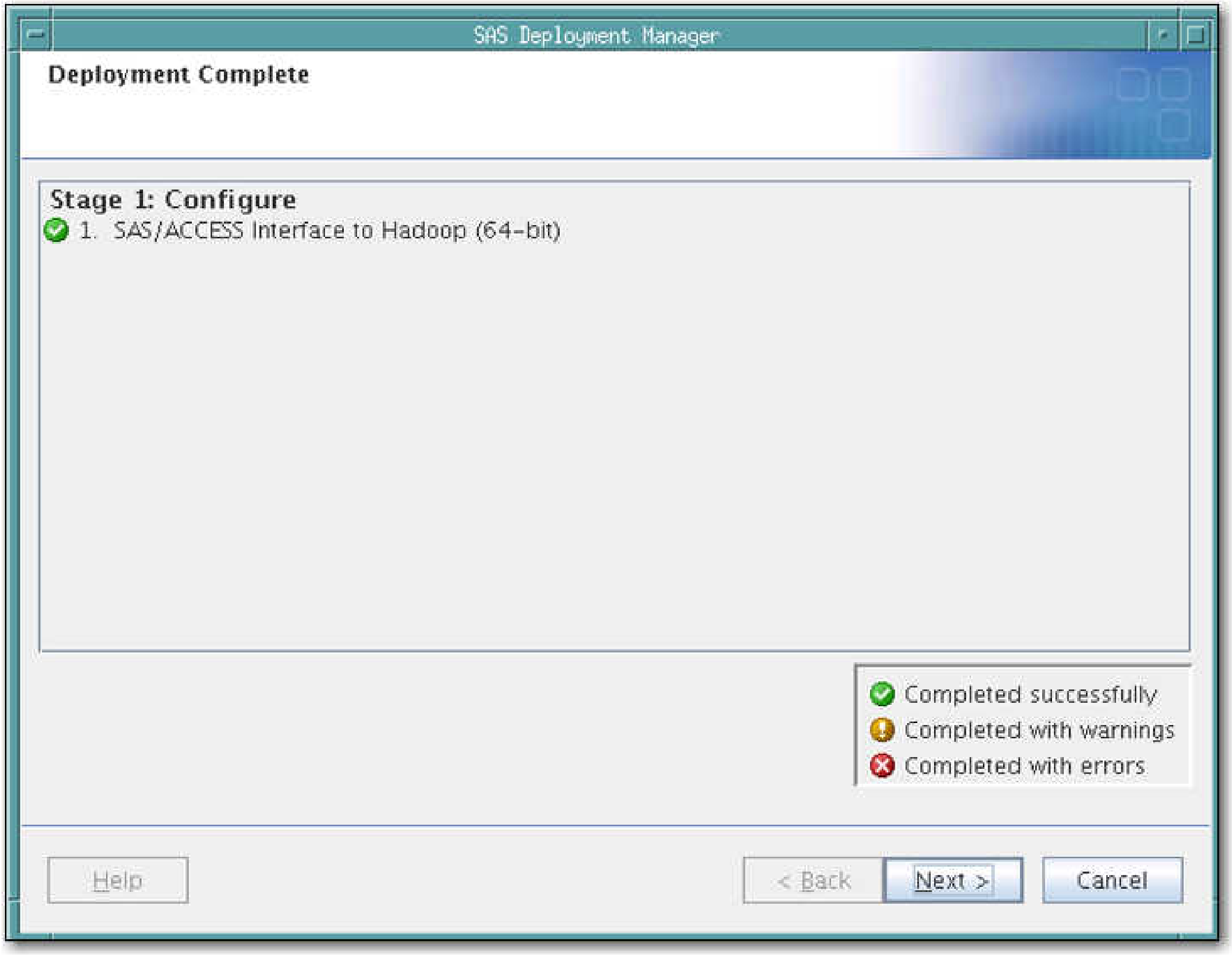
! Completed with warnings

✗ Completed with errors

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

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

If the configuration is successful, the page title changes to **Deployment Complete** and a green check mark is displayed beside SAS/ACCESS Interface to Hadoop.



Note: Part of the configuration process runs SAS code to validate the environment. A green check mark indicates that the 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.

16 Click **Next** to close the SAS Deployment Manager.

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.

Supporting Multiple Hadoop Versions and Upgrading Your 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. 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, you must set up the Cloudera Impala ODBC driver. For instructions, see Installation Guide for Cloudera ODBC 2.5.x Driver for Implala.

Note: At the time of the second maintenance release for SAS 9.4, Cloudera did not have an ODBC driver available for AIX (R64). Instead, SAS/ACCESS Interface to Impala linked to the DataDirect ODBC Driver Manager. Starting with version 2.5.17, the Cloudera ODBC driver for Impala includes support for AIX (R64). If you use Cloudera ODBC driver for Impala version 2.5.17 or later, enter the following commands in the directory that contains your ODBC driver manager library:

```
unix_prompt mv libodbc.a libodbc.a.bak
unix_prompt cp libodbc.so.2 odbc.so
unix_prompt ar -X64 -r libdoc.a odbc.so
```

This creates the **libodbc.a** archive with the internal shared object name that SAS/ACCESS Interface to Impala is expecting.

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 interface to Hadoop to push data to HDFS. The SAS environment variable SAS_HADOOP_RESTFUL must be defined and set to the value of 1. You can include the properties for the WebHDFS location in the Hadoop hdfs-site.xml file. Alternatively, specify the WebHDFS host name or the IP address of the server where the external file is stored using the BL_HOST= option. 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” on page 32 and “Using the SAS Deployment Manager to Make Required Hadoop JAR and Configuration Files Available to the SAS Client Machine” on page 15.

- 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 the SAS Deployment Manager to Make Required Hadoop JAR and Configuration Files Available to the SAS Client Machine” on page 15.

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

Using WebHDFS

WebHDFS is an HTTP REST API that supports the complete FileSystem interface for HDFS.

To use WebHDFS instead of the HDFS service, follow these steps. Note that using WebHDFS removes the need for client-side JAR files for HDFS, but Hive JAR files are still needed.

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 60.

2. Make sure the configuration files include the properties for the WebHDFS location, which include the dfs.http.address 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>
```

For more information about the configuration files, see “Configuring Hadoop JAR and Configuration Files” on page 15.

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:

```
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:hive://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 HiveServer2:

```
libname hdplib hadoop server=hadoop_h2 user=hadoop_usr password=hadoop_usr_pwd
   subprotocol=hive2;

NOTE: Libref HDPLIB was successfully assigned as follows:
Engine: HADOOP
Physical Name: jdbc:hive2://hadoop_h2:10000/default
```

A failure to connect can have different causes. Error messages assist in diagnosing 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 connecton 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.
```

Security and User Access to Hadoop

Kerberos Security

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

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

  ```
  -Djavax.security.auth.useSubjectCredsOnly=false
  ```

- 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
    ```

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**.

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 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
```

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

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 40.

2 Make Hadoop JAR files available to the SAS client machine by defining a SAS environment variable to set the location of the files. This step involves copying a set of JAR files to the SAS client machine that accesses Hadoop.

For more information, see "Configuring Hadoop JAR Files" on page 40.

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 42.

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 43.

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 more information, see “Supported Hadoop Distributions for SPD Engine” on page 40.

3. Install Base SAS by following the instructions in your software order e-mail.

Supported Hadoop Distributions for SPD Engine

In the second maintenance release for SAS 9.4, the following Hadoop distributions are supported for the SPD Engine:

- Cloudera CDH 4.5
- Cloudera CDH 5.0
- Hortonworks HDP 2.0
- Pivotal HD 2.0.1

Configuring Hadoop JAR Files

Making Required Hadoop JAR Files Available to the SAS Client Machine

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 4.5. 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 47 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 42.

3 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 58.

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 42.

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.

Making Hadoop Cluster Configuration Files Available to the SAS Client Machine

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 and YARN cluster, the mapred-site.xml and yarn-site.xml files are 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 57.

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
    ```

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 JAR Files

Cloudera 4.5.x JAR Files

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 can be found by going to 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 the 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-cdh4.5.0.jar
hadoop-common-2.0.0-cdh4.5.0.jar
hadoop-core-2.0.0-mr1-cdh4.5.0.jar
hadoop-hdfs-2.0.0-cdh4.5.0.jar
hive-exec-0.10.0-cdh4.5.0.jar
hive-jdbc-0.10.0-cdh4.5.0.jar
hive-metastore-0.10.0-cdh4.5.0.jar
hive-service-0.10.0-cdh4.5.0.jar
libfb303-0.9.0.jar
pig-0.11.0-cdh4.5.0-withouthadoop.jar

protobuf-java-2.4.0a.jar

For the SPD Engine on Cloudera 4.5, 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-2.5.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 JAR Files

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 can be found by going to 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 the 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

Hortonworks JAR Files

Hortonworks HDP 1.3.x JAR Files

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 can be found by going to 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 the 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: 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 can be found by going to 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 the 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: 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 can be found by going to 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 the individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

automation-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

IBM InfoSphere BigInsights 2.1 JAR Files

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 can be found by going to 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 the individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

JSON4J_Apache-1.0.jar
JavaEWAH-0.3.2.jar
ST4-4.0.4.jar
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
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
jsp-2.1-6.1.14.jar
jsr305-1.3.9.jar
jython-standalone-2.5.3.jar
libfb303-0.9.0.jar
libthrift-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
tempus-fugit-1.1.jar
workflowScheduler.jar
xz-1.0.jar
zookeeper-3.4.5.jar

MapR 3.1 JAR Files

To install the client side JAR files for MapR, follow the instructions at Setting Up the Client on the MapR website.

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 can be found by going to 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 the individual technology directories. Your Hadoop administrator can assist you in locating the appropriate JAR files.

After completing the installation of those JAR files, copy these JAR files to the same location:

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

Pivotal HD 2.0.1 JAR Files

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 can be found by going to 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 the 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-3.0.0.0.jar
hadoop-auth-2.2.0-gphd-3.0.0.0.jar
hadoop-common-2.2.0-gphd-3.0.0.0.jar
hadoop-hdfs-2.2.0-gphd-3.0.0.0.jar
hadoop-hdfs-nfs-2.2.0-gphd-3.0.0.0.jar
hadoop-mapreduce-client-app-2.2.0-gphd-3.0.0.0.jar
hadoop-mapreduce-client-common-2.2.0-gphd-3.0.0.0.jar
hadoop-mapreduce-client-core-2.2.0-gphd-3.0.0.0.jar
hadoop-mapreduce-client-jobclient-2.2.0-gphd-3.0.0.0.jar
hadoop-mapreduce-client-shuffle-2.2.0-gphd-3.0.0.0.jar
hadoop-nfs-2.2.0-gphd-3.0.0.0.jar
hadoop-vaidya-2.2.0-gphd-3.0.0.0.jar
hadoop-yarn-api-2.2.0-gphd-3.0.0.0.jar
hadoop-yarn-client-2.2.0-gphd-3.0.0.0.jar
hadoop-yarn-common-2.2.0-gphd-3.0.0.0.jar
hadoop-yarn-server-common-2.2.0-gphd-3.0.0.0.jar
hive-beeline-0.12.0-gphd-3.0.0.0.jar
hive-cli-0.12.0-gphd-3.0.0.0.jar
hive-common-0.12.0-gphd-3.0.0.0.jar
hive-contrib-0.12.0-gphd-3.0.0.0.jar
hive-exec-0.12.0-gphd-3.0.0.0.jar
hive-hwi-0.12.0-gphd-3.0.0.0.jar
hive-jdbc-0.12.0-gphd-3.0.0.0.jar
hive-metastore-0.12.0-gphd-3.0.0.0.jar
hive-service-0.12.0-gphd-3.0.0.0.jar
hive-shims-0.12.0-gphd-3.0.0.0.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-3.0.0.0-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 2.0.1, 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

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: SAS/ACCESS Interface to Hadoop, SPD Engine

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\config**, then the following OPTIONS statement syntax sets the environment variable appropriately.

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

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.

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/config"`
	SAS invocation	`-set SAS_HADOOP_CONFIG_PATH "/sasdata/cluster1/config"`
	OPTIONS statement	`options set=SAS_HADOOP_CONFIG_PATH="/sasdata/cluster1/config";`
Windows	SAS configuration file	`-set SAS_HADOOP_CONFIG_PATH "C:\sasdata\cluster1\config"`
	SAS invocation	`-set SAS_HADOOP_CONFIG_PATH "C:\sasdata\cluster1\config"`
	OPTIONS statement	`options set=SAS_HADOOP_CONFIG_PATH="C:\sasdata\cluster1\config";`

* 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

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**, then the following OPTIONS statement syntax sets the environment variable appropriately.

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

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";
```

Details

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.

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"`
	SAS invocation	`-set SAS_HADOOP_JAR_PATH "/third_party/Hadoop/jars"`
	OPTIONS statement	`options set=SAS_HADOOP_JAR_PATH="/third_party/Hadoop/jars";`
Windows	SAS configuration file	`-set SAS_HADOOP_JAR_PATH "C:\third_party\Hadoop\jars"`
	SAS invocation	`-set SAS_HADOOP_JAR_PATH "C:\third_party\Hadoop\jars"`

Operating Environment	Method	Example
	OPTIONS statement	`options set=SAS_HADOOP_JAR_PATH="C:\third_party \Hadoop\jars";`

* 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 31.

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 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 REST API.

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

Details

WebHDFS is an HTTP REST API that supports the complete FileSystem interface for HDFS.

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*

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

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-3228
Fax: 1-919-677-8166
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

Index