THE POWER TO KNOW.

SAS® Data Loader 2.2 for Hadoop

Administrator's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2015. *SAS® Data Loader for Hadoop 2.2: Administrator's Guide*. Cary, NC: SAS Institute Inc.

SAS® Data Loader for Hadoop 2.2: Administrator's Guide

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

Printing 1, March 2015

Contents

Part 1

Installation

1

Introduction

About SAS In-Database Technologies for Hadoop

SAS In-Database Technologies for Hadoop supports the operation of SAS Data Loader for Hadoop. SAS Data Loader for Hadoop is web-client software that is separately downloaded by the user, installed as a vApp, and run in a virtual machine. The complete SAS Data Loader for Hadoop library consists of the following books:

For business analysts, data stewards, and other SAS Data Loader users:

- The *SAS Data Loader for Hadoop: vApp Deployment Guide* documents the installation, configuration, and settings of the SAS Data Loader for Hadoop vApp on the client machine. Install the vApp after your system administrator has deployed the SAS In-Database Technologies for Hadoop offering.
- The *SAS Data Loader for Hadoop: User's Guide* documents how to use SAS Data Loader for Hadoop, provides examples, and demonstrates how to update the vApp. It also explains how to update your vApp and manage your license.

For System and Hadoop Administrators:

- This *SAS Data Loader for Hadoop: Administrator's Guide* documents the installation, configuration, and administration of SAS In-Database Technologies for Hadoop on the Hadoop cluster. This offering must be installed first and before the installation of the SAS Data Loader for Hadoop vApp in order for the vApp to communicate successfully with the Hadoop cluster.

Support for the vApp User

You must configure the Hadoop cluster and provide certain values to the vApp user. See "Configuring Components on the Cluster" on page 9 and "End-User Support" on page 43 for specific information about what you must provide.

System Requirements

System requirements for the SAS Data Loader for Hadoop vApp are specified in the *SAS Data Loader for Hadoop: vApp Deployment Guide*.

Hadoop Environment

Note: SAS In-Database Technologies for Hadoop must be installed before the installation of the SAS Data Loader for Hadoop vApp in order for the vApp to communicate successfully with the Hadoop cluster.

System requirements for the Hadoop environment are as follows:

- Cloudera CDH 5.2 or Hortonworks HDP 2.1.

  Note: Both Hive 2 and YARN (MapReduce 2) are supported. MapReduce 1 is not supported.

  See Chapter 4, "The In-Database Deployment Package," on page 13.

- If Kerberos security is supported on the Hadoop cluster, then the vApp must be configured for Kerberos on the client machine.

  See Chapter 6, "Security," on page 37.

- SAS Data Loader for Hadoop uses the SQOOP and OOZIE components of your Hadoop deployment to move data into or out of a DBMS. These components must be enabled in your Hadoop cluster in order to communicate with the DBMS to which SAS Data Loader for Hadoop users need access.

  See "SQOOP and OOZIE" on page 9.

- The JDBC drivers required by the DBMS that the SAS Data Loader for Hadoop users need access to must be installed on the Hadoop cluster.

  See "JDBC Driver" on page 9.

SAS Server Environment

System requirements for the SAS Server environment are as follows:

- Supported operating systems: AIX, HP IPF, Linux for x64, Solaris SPARC, Solaris for x64, Windows 32-bit server or workstation, and Windows for x64 server or workstation.
- Base SAS 9.4, second maintenance release

2

SAS In-Database Technologies for Hadoop

Installing with the SAS Download Manager

You receive a Software Order Email (SOE) with your licensed order of SAS In-Database Technologies for Hadoop. The SOE describes how to install your order and links to tools that you use to do so.

Note: SAS In-Database Technologies for Hadoop must be installed before to the installation of the SAS Data Loader for Hadoop vApp for the vApp to communicate successfully with the Hadoop cluster. The SAS Data Loader for Hadoop vApp installation is described in a separate SOE.

To install your software:

1. Download the SAS Download Manager.

   The SAS Download Manager is the application that you use to download your software. Follow the link in your SOE to install and download your SAS Download Manager.

   After you launch the SAS Download Manager, it guides you through a series of steps in which you insert the order number listed in your SOE. The SAS Download Manager downloads your software to the SAS Software Depot, which is a repository for your SAS software media. If you do not have a SAS Software Depot, the SAS Download Manager creates one for you.

2. After the SAS Download Manager downloads the SAS software, return to the SOE for additional instructions to begin your SAS installation using the SAS Deployment Wizard. Review the products in the **Select Products to Install** dialog box. Select SAS Quality Knowledge Base in addition to the pre-selected products.

3. After completing your installation, you must configure SAS In-Database Technologies for Hadoop. You might also need to install additional components, if you do not already have them, and then configure them. See the configuration section of this guide for complete instructions.

Part 2

Configuration

3

The Hadoop Cluster

Configuring Components on the Cluster

Overview

You must configure several components and settings on the Hadoop cluster in order for SAS Data Loader for Hadoop to operate correctly. These are explained in the following four topics:

- “SQOOP and OOZIE” on page 9
- “JDBC Driver” on page 9
- “Hadoop Configuration Files” on page 10
- “User ID, Permissions, and Configuration Values” on page 11

SQOOP and OOZIE

Your Hadoop cluster must be configured to use SQOOP commands and OOZIE scripts.

Note: You must add `sqoop-action-0.4.xsd` as an entry in the list for the oozie.service.SchemaService.wf.ext.schemas property.

JDBC Driver

SAS Data Loader for Hadoop leverages the SQOOP and OOZIE components installed with Hadoop cluster to move data to and from a DBMS. The SAS Data Loader for Hadoop vApp client also accesses the databases directly using JDBC for the purpose of selecting either source or target schemas and tables to move.

You must install on the Hadoop cluster the JDBC driver that is required by the DBMS that users need to access. Follow the JDBC driver vendor installation instructions.

SAS Data Loader for Hadoop supports the Teradata and Oracle DBMSs directly. You can support additional databases selecting **Other** in the **Type** option on the SAS Data Loader for Hadoop Database Configuration dialog box. See the *SAS Data Loader for Hadoop: User's Guide* for more information about the dialog box.

For Teradata and Oracle, SAS recommends that you download the following JDBC files from the vendor site:

Table 3.1 *JDBC Files*

Database	Required Files
Oracle	ojdbc6.jar
Teradata	tdgssconfig.jar and terajdbc4.jar **Note:** You must also download the Teradata connector JAR file that is matched to your cluster distribution.

The JDBC driver and Teradata clients are installed under the OOZIE shared lib directory in the Hadoop file system as follows:

- Hortonworks Hadoop clusters: `/user/oozie/share/lib/sqoop`
- Cloudera Hadoop clusters: `/user/oozie/share/lib/sharelib<version>/sqoop`

The JDBC and connector JAR files must be located in the OOZIE shared libs directory in HDFS, not in `/var/lib/sqoop`. The correct path is available from the oozie.service.WorkflowAppService.system.libpath property.

You must have, at a minimum, -rw-r--r-- permissions on the JDBC drivers.

After JDBC drivers have been installed and configured along with SQOOP and OOZIE, you must restart OOZIE.

SAS Data Loader for Hadoop users must also have the same version of the JDBC drivers on their client machines in the `SASWorkspace\JDBCDrivers` directory. Provide a copy of the JDBC drivers to SAS Data Loader for Hadoop users.

Hadoop Configuration Files

You must make the following configuration files from the Hadoop cluster available to be copied to the client machine of the SAS Data Loader for Hadoop vApp user:

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

Note:

- For a MapReduce 2 and YARN cluster, both the mapred-site.xml and yarn-site.xml files are required.

For the Copy to Hadoop and Copy from Hadoop directives to work correctly, you must set the following entries to False:

- the dfs.permissions.enabled entry in the hdfs-site.xml file on the Hadoop cluster
- the hive.resultset.use.unique.column.names entry in the hive-site.xml file on the target Hadoop cluster

User ID, Permissions, and Configuration Values

Your Hadoop cluster can use Kerberos authentication or a different type of authentication of users. If you are using Kerberos authentication, see Chapter 6, "Security," on page 37.

For clusters that do not use Kerberos, you must create one or more user IDs and enable certain permissions for the SAS Data Loader for Hadoop vApp user.

Use the following procedure to establish user IDs:

1. Choose one of the following options for user IDs:
   - create one Hadoop Configuration User ID for all vApp users

     Note: Do not use the Hadoop super user, which is typically hdfs.
   - create one Hadoop Configuration User ID for every vApp user
2. Create UNIX users in **`/etc/passwd`** and **`/etc/groups`**.
3. Create a map reduce staging hdfs directory defined in the mapreduce configuration. The default is **`/users/myuser`**
4. Change the permissions and owner of hdfs **`/users/myuser`** to match the UNIX user.

   Note: The user ID must have at least the following permissions:
   - Read,Write, and Delete permission for files in the HDFS directory (used for Oozie jobs)
   - Read, Write, and Delete permission for tables in Hive
5. If you are using Hive, create the Hive database and set appropriate permissions on the folder.
6. The user= value in the SAS LIBNAME must match the hdfs user that you created on the Hadoop cluster, as in the following example:

```
libname testlib hadoop
server = "hadoopdb"
user = user2
database = mydatabase
;
```

You must provide the vApp user with values for fields in the SAS Data Loader for Hadoop Configuration dialog box. See the *SAS Data Loader for Hadoop: vApp Deployment Guide* for more information about the SAS Data Loader for Hadoop Configuration dialog box. The fields are as follows:

User ID
the Hadoop Configuration User ID user account that you have created on your Hadoop cluster for each user or all of the vApp users.

Host
the full host name of the machine on the cluster running the Hive server.

Port
the number of the Hadoop port on the host that supports your cluster.

- For Cloudera, the HiveServer2 port default is 10000.
- For HortonWorks, the Hive server port default is 10000.

Oozie URL
the URL to the Oozie Web Console, which is an interface to the Oozie server. The URL is similar to the following example: http://*host_name*:*port_number*/oozie/.

Confirm that the Oozie Web UI is enabled before providing it to the vApp user. If it is not, use Oozie Web Console to enable it.

4

The In-Database Deployment Package

Introduction

Configuring the in-database deployment package for Hadoop needs to be done only once for each Hadoop cluster.

In-Database Deployment Package for Hadoop

Prerequisites

The following are required before you install and configure the in-database deployment package for Hadoop:

- You have working knowledge of the Hadoop vendor distribution that you are using.

  You also need working knowledge of the Hadoop Distributed File System (HDFS), MapReduce 2, YARN, Hive, and HiveServer2 services. For more information, see the Apache website or the vendor's website.
- The HDFS, MapReduce, YARN, and Hive services must be running on the Hadoop cluster.
- You have root or sudo access. Your user has Write permission to the root of HDFS.
- You know the location of the MapReduce home.
- You know the host name of the Hive server and the NameNode.
- You understand and can verify your Hadoop user authentication.
- You understand and can verify your security setup.

  If you are using Kerberos, you need the ability to get a Kerberos ticket.
- You have permission to restart the Hadoop MapReduce service.
- In order to avoid SSH key mismatches during installation, add the following two options to the SSH **config** file, under the user's home .ssh folder. An example of a home .ssh folder is **/root/.ssh/**. *nodes* is a list of nodes separated by a space.

  ```
  host nodes
     StrictHostKeyChecking no
     UserKnownHostsFile /dev/null
  ```

  For more details about the SSH **config** file, see the SSH documentation.
- All machines in the cluster are set up to communicate with passwordless SSH. Verify that the nodes can access the node that you chose to be the master node by using SSH.

  Traditionally, public key authentication in Secure Shell (SSH) is used to meet the passwordless access requirement. SSH keys can be generated with the following example.

  ```
  [root@raincloud1 .ssh]# ssh-keygen -t rsa
  Generating public/private rsa key pair.
  Enter file in which to save the key (/root/.ssh/id_rsa):
  Enter passphrase (empty for no passphrase):
  Enter same passphrase again:
  Your identification has been saved in /root/.ssh/id_rsa.
  Your public key has been saved in /root/.ssh/id_rsa.pub.
  The key fingerprint is:
  09:f3:d7:15:57:8a:dd:9c:df:e5:e8:1d:e7:ab:67:86 root@raincloud1

  add id_rsa.pub public key from each node to the master node authorized
     key file under /root/.ssh/authorized_keys
  ```

For Secure Mode Hadoop, GSSAPI with Kerberos is used as the passwordless SSH mechanism. GSSAPI with Kerberos not only meets the passwordless SSH requirements, but also supplies Hadoop with the credentials required for users to perform operations in HDFS with SAS LASR Analytic Server and SASHDAT files. Certain options must be set in the SSH daemon and SSH client configuration files. Those options are as follows and assume a default configuration of sshd.

To configure passwordless SSH to use Kerberos, follow these steps:

1 In the sshd_config file, set:

```
GSSAPIAuthentication yes
```

2 In the ssh_config file, set:

```
Host *.domain.net
GSSAPIAuthentication yes
GSSAPIDelegateCredentials yes
```

where *domain.net* is the domain name used by the machine in the cluster.

TIP Although you can specify Host *, this is not recommended because it allows GSSAPI Authentication with any host name.

Overview of the In-Database Deployment Package for Hadoop

This section describes how to install and configure the in-database deployment package for Hadoop (SAS Embedded Process).

The in-database deployment package for Hadoop must be installed and configured before you can transform data in Hadoop and extract transformed data out of Hadoop for analysis.

The in-database deployment package for Hadoop includes the SAS Embedded Process and two SAS Hadoop MapReduce JAR files. The SAS Embedded Process is a SAS server process that runs within Hadoop to read and write data. The SAS Embedded Process contains macros, run-time libraries, and other software that is installed on your Hadoop system.

The SAS Embedded Process must be installed on all nodes capable of executing MapReduce 2 and YARN tasks. The SAS Hadoop MapReduce JAR files must be installed on all nodes of a Hadoop cluster.

The SAS Embedded Process must be installed on all nodes capable of executing MapReduce 2 and YARN tasks, that is, nodes where a NodeManager is running. Usually, every DataNode node has a YARN NodeManager running. By default, the SAS Embedded Process install script (sasep-servers.sh) discovers the cluster topology and installs the SAS Embedded Process on all DataNode nodes, including the host node from where you run the script (the Hadoop master NameNode). This occurs even if a DataNode is not present. If you want to limit the list of nodes on which you want the SAS Embedded Process installed, you should run the sasep-servers.sh script with the **-host <hosts>** option. The SAS Hadoop MapReduce JAR files must be installed on all nodes of a Hadoop cluster.

Hadoop Installation and Configuration

Hadoop Installation and Configuration Steps

1 If you are upgrading from or reinstalling a previous release, follow the instructions in "Upgrading from or Reinstalling a Previous Version" on page 16 before installing the in-database deployment package.

2 Move the SAS Embedded Process and SAS Hadoop MapReduce JAR file install scripts to the Hadoop master node (the NameNode).

For more information, see "Moving the SAS Embedded Process and SAS Hadoop MapReduce JAR File Install Scripts" on page 18.

Note: Both the SAS Embedded Process install script and the SAS Hadoop MapReduce JAR file install script must be transferred to the SASEPHome directory.

Note: The location where you transfer the install scripts becomes the SAS Embedded Process home and is referred to as SASEPHome throughout this chapter.

3 Install the SAS Embedded Process and the SAS Hadoop MapReduce JAR files.

For more information, see "Installing the SAS Embedded Process and SAS Hadoop MapReduce JAR Files" on page 19.

Note: If you are installing the SAS High-Performance Analytics environment, you must perform additional steps after you install the SAS Embedded Process. For more information, see *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*.

Upgrading from or Reinstalling a Previous Version

To upgrade or reinstall a previous version, follow these steps.

1 If you are upgrading from SAS 9.3, follow these steps. If you are upgrading from SAS 9.4, start with Step 2.

a Stop the Hadoop SAS Embedded Process.

```
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.35/bin/sasep-stop.all.sh
```

SASEPHome is the master node where you installed the SAS Embedded Process.

b Delete the Hadoop SAS Embedded Process from all nodes.

```
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.35/bin/sasep-delete.all.sh
```

c Verify that all files named sas.hadoop.ep.*distribution-name*.jar have been deleted.

The JAR files are located at ***HadoopHome*/lib**.

For Cloudera, the JAR files are typically located here:

```
/opt/cloudera/parcels/CDH/lib/hadoop/lib
```

For Hortonworks, the JAR files are typically located here:

```
/usr/lib/hadoop/lib
```

d Continue with Step 3.

2 If you are upgrading from SAS 9.4, follow these steps.

a Stop the Hadoop SAS Embedded Process.

```
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.*/bin/sasep-servers.sh
   -stop -hostfile host-list-filename | -host <">host-list<">
```

SASEPHome is the master node where you installed the SAS Embedded Process.

For more information, see "SASEP-SERVERS.SH Script" on page 23.

b Remove the SAS Embedded Process from all nodes.

```
SASEPHome/SAS/SASTKInDatabaseForServerHadoop/9.*/bin/sasep-servers.sh
   -remove -hostfile host-list-filename | -host <">host-list<">
   -mrhome dir
```

Note: This step ensures that all old SAS Hadoop MapReduce JAR files are removed.

For more information, see "SASEP-SERVERS.SH Script" on page 23.

c Verify that all files named sas.hadoop.ep.apache*.jar have been deleted.

The JAR files are located at ***HadoopHome*/lib**.

For Cloudera, the JAR files are typically located here:

```
/opt/cloudera/parcels/CDH/lib/hadoop/lib
```

For Hortonworks, the JAR files are typically located here:

```
/usr/lib/hadoop/lib
```

Note: If all the files have not been deleted, then you must delete them. Open-source utilities are available that can delete these files across multiple nodes.

d Verify that all the SAS Embedded Process directories and files have been deleted on all nodes, except the node from which you are running the script. The sasep-servers.sh -remove script removes the files everywhere except on the node from which you ran the script.

Note: If all the directories and files have not been deleted, then you must delete them. Open-source utilities are available that can delete these directories and files across multiple nodes.

Manually remove the SAS Embedded Process directories and files on the node from which you ran the script.

The sasep-servers.sh -remove script displays instructions that are similar to the following example:

```
localhost WARN: Apparently, you are trying to uninstall SAS Embedded Process
for Hadoop from the local node.
The binary files located at
local_node/SAS/SASTKInDatabaseServerForHadoop/local_node/
```

```
SAS/SASACCESStoHadoopMapReduceJARFiles will not be removed.
localhost WARN: The init script will be removed from /etc/init.d and the
SAS Map Reduce JAR files will be removed from /usr/lib/hadoop-mapreduce/lib.
localhost WARN: The binary files located at local_node/SAS
should be removed manually.
```

3 Continue the installation process.

For more information, see "Installing the SAS Embedded Process and SAS Hadoop MapReduce JAR Files" on page 19.

Moving the SAS Embedded Process and SAS Hadoop MapReduce JAR File Install Scripts

Creating the SAS Embedded Process Directory

Before you can install the SAS Embedded Process and the SAS Hadoop MapReduce JAR files, you must move the SAS Embedded Process and SAS Hadoop MapReduce JAR file install scripts to a directory on the Hadoop master node (the NameNode).

Create a new directory that is not part of an existing directory structure, such as **/sasep**.

This path is created on each node in the Hadoop cluster during the SAS Embedded Process installation. Do not use existing system directories such as **/opt or /usr**. This new directory becomes the SAS Embedded Process home and is referred to as SASEPHome throughout this chapter.

Moving the SAS Embedded Process Install Script

The SAS Embedded Process install script is contained in a self-extracting archive file named tkindbsrv-9.42-*n*_lax.sh where *n* is a number that indicates the latest version of the file. If this is the initial installation, *n* has a value of 1. Each time you reinstall or upgrade, *n* is incremented by 1. The self-extracting archive file is located in the **[SASHome]/SASTKInDatabaseServer/9.4/HadooponLinuxx64** directory.

Using a method of your choice, transfer the SAS Embedded Process install script to your Hadoop master node.

This example uses secure copy, and *SASEPHome* is the location where you want to install the SAS Embedded Process.

```
scp tkindbsrv-9.42-n_lax.sh username@hadoop:/SASEPHome
```

Note: The location where you transfer the install script becomes the SAS Embedded Process home.

Note: Both the SAS Embedded Process install script and the SAS Hadoop MapReduce JAR file install script must be transferred to the **SASEPHome** directory.

Moving the SAS Hadoop MapReduce JAR File Install Script

The SAS Hadoop MapReduce JAR file install script is contained in a self-extracting archive file named hadoopmrjars-9.42-*n*_lax.sh where *n* is a number that indicates the latest version of the file. If this is the initial installation, *n* has a

value of 1. Each time you reinstall or upgrade, *n* is incremented by 1. The self-extracting archive file is located in the **[SASHome]/SASACCESStoHadoopMapReduceJARFiles/9.42** directory.

Using a method of your choice, transfer the SAS Hadoop MapReduce JAR file install script to your Hadoop master node.

This example uses Secure Copy, and *SASEPHome* is the location where you want to install the SAS Hadoop MapReduce JAR files.

```
scp hadoopmrjars-9.42-n_lax.sh username@hadoop:/SASEPHome
```

Note: Both the SAS Embedded Process install script and the SAS Hadoop MapReduce JAR file install script must be transferred to the **SASEPHome** directory.

Installing the SAS Embedded Process and SAS Hadoop MapReduce JAR Files

To install the SAS Embedded Process, follow these steps.

Note: Permissions are needed to install the SAS Embedded Process and SAS Hadoop MapReduce JAR files. For more information, see "Hadoop Permissions" on page 29.

1 Log on to the server using SSH as root with sudo access.

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

2 Move to your Hadoop master node where you want the SAS Embedded Process installed.

```
cd /SASEPHome
```

SASEPHome is the same location to which you copied the self-extracting archive file. For more information, see "Moving the SAS Embedded Process Install Script" on page 18.

Note: Before continuing with the next step, ensure that each self-extracting archive file has Execute permission.

3 Use the following script to unpack the tkindbsrv-9.42-*n*_lax.sh file.

```
./tkindbsrv-9.42-n_lax.sh
```

n is a number that indicates the latest version of the file. If this is the initial installation, *n* has a value of 1. Each time you reinstall or upgrade, *n* is incremented by 1.

Note: If you unpack in the wrong directory, you can move it after the unpack.

After this script is run and the files are unpacked, the script creates the following directory structure where *SASEPHome* is the master node from Step 1.

```
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/misc
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/sasexe
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/utilities
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/build
```

The content of the ***SASEPHome*/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin** directory should look similar to this.

```
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/sas.ep4hadoop.template
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/sasep-servers.sh
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/sasep-common.sh
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/sasep-server-start.sh
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/sasep-server-status.sh
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/sasep-server-stop.sh
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/InstallTKIndbsrv.sh
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/MANIFEST.MF
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/qkbpush.sh
SASEPHome/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/sas.tools.qkb.hadoop.jar
```

4 Use this command to unpack the SAS Hadoop MapReduce JAR files.

```
./hadoopmrjars-9.42-1_lax.sh
```

After the script is run, the script creates the following directory and unpacks these files to that directory.

```
SASEPHome/SAS/SASACCESStoHadoopMapReduceJARFiles/9.42-1/lib/ep-config.xml
SASEPHome/SAS/SASACCESStoHadoopMapReduceJARFiles/9.42-1/lib/
   sas.hadoop.ep.apache023.jar
SASEPHome/SAS/SASACCESStoHadoopMapReduceJARFiles/9.42-1/lib/
    sas.hadoop.ep.apache023.nls.jar
SASEPHome/SAS/SASACCESStoHadoopMapReduceJARFiles/9.42-1/lib/
   sas.hadoop.ep.apache121.jar
SASEPHome/SAS/SASACCESStoHadoopMapReduceJARFiles/9.42-1/lib/
   sas.hadoop.ep.apache121.nls.jar
SASEPHome/SAS/SASACCESStoHadoopMapReduceJARFiles/9.42-1/lib/
   sas.hadoop.ep.apache205.jar
SASEPHome/SAS/SASACCESStoHadoopMapReduceJARFiles/9.42-1/lib/
   sas.hadoop.ep.apache205.nls.jar
```

5 Use the sasep-servers.sh script with the `-add` option to deploy the SAS Embedded Process installation across all nodes. The SAS Embedded Process is installed as a Linux service.

Note: If you are running on a cluster with Kerberos, complete both steps a and b. If you are not running with Kerberos, complete only step b.

a If you are running on a cluster with Kerberos, you must kinit the HDFS user.

```
sudo su - root
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: The default HDFS user is **hdfs**. You can specify a different user ID with the `-hdfsuser` argument when you run the `sasep-servers.sh -add` command.

Note: If you are running on a cluster with Kerberos, a keytab is required when running the `sasep-servers.sh -add` command.

Note: You can run klist while you are running as an HDFS 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
```

b Run the sasep-servers.sh script. Review all of the information in this step before running the script.

```
cd SASEPHOME/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin
./sasep-servers.sh -add
```

TIP There are many options available when installing the SAS Embedded Process. We recommend that you review the script syntax before running it. For more information, see "SASEP-SERVERS.SH Script" on page 23.

During the install process, the script asks whether you want to start the SAS Embedded Process. If you choose **Y** or **y**, the SAS Embedded Process is started on all nodes after the install is complete. If you choose **N** or **n**, you can start the SAS Embedded Process later by running the `./sasep-servers.sh -start` command.

Note: When you enter the `sasep-servers.sh -add` command, a user and group named **sasep** is created. You can specify a different user and group name with the `-epuser` and `-epgroup` arguments when you enter the `sasep-servers.sh -add` command.

Note: The sasep-servers.sh script can be run from any location. You can also add its location to the PATH environment variable.

TIP Although you can install the SAS Embedded Process in multiple locations, the best practice is to install only one instance. Only one version of the SASEP JAR files is installed in your *HadoopHome*/lib directory.

Note: The SAS Embedded Process must be installed on all nodes capable of executing MapReduce 2 tasks. For MapReduce 2, this would be nodes where a NodeManager is running. Usually, every DataNode node has a YARN NodeManager running. By default, the SAS Embedded Process install script (sasep-servers.sh) discovers the cluster topology and installs the SAS Embedded Process on all DataNode nodes, including the host node from where you run the script (the Hadoop master NameNode). This occurs even if a DataNode is not present. If you want to limit the list of nodes on which you want the SAS Embedded Process installed, run the sasep-servers.sh script with the **-host <hosts>** option.

Note: If you install the SAS Embedded Process on a large cluster, the SSHD daemon might reach the maximum number of concurrent connections. The `ssh_exchange_identification: Connection closed by remote host` SSHD error might occur. To work around the problem, edit the `/etc/ssh/sshd_config` file, change the MaxStartups option to the number that accommodates your cluster, and save the file. Then, reload the SSHD daemon by running the `/etc/init.d/sshd reload` command.

6 Verify that the SAS Embedded Process is installed and running. Change directories and then run the sasep-servers.sh script with the -status option.

```
cd SASEPHOME/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin
./sasep-servers.sh -status
```

This command returns the status of the SAS Embedded Process running on each node of the Hadoop cluster. Verify that the SAS Embedded Process home directory is correct on all the nodes.

Note: The `sasep-servers.sh -status` command cannot run successfully if the SAS Embedded Process is not installed.

7 Verify that the sas.hadoop.ep.apache*.jar files are now in place on all nodes.

The JAR files are located at ***HadoopHome*/lib**.

For Cloudera, the JAR files are typically located here:

```
/opt/cloudera/parcels/CDH/lib/hadoop/lib
```

For Hortonworks, the JAR files are typically located here:

```
/usr/lib/hadoop/lib
```

8 Restart the Hadoop YARN or MapReduce service.

This enables the cluster to reload the SAS Hadoop JAR files (sas.hadoop.ep.*.jar).

Note: It is preferable to restart the service by using Cloudera Manager or Hortonworks Ambari.

9 Verify that an init.d service with a sas.ep4hadoop file was created in the following directory.

```
/etc/init.d/sas.ep4hadoop
```

View the sas.ep4hadoop file and verify that the SAS Embedded Process home directory is correct.

The init.d service is configured to start at level 3 and level 5.

Note: The SAS Embedded Process needs to run on all nodes in the Hadoop cluster.

10 Verify that configuration files were written to the HDFS file system.

```
hadoop fs -ls /sas/ep/config
```

Note: If you are running on a cluster with Kerberos, you need a Kerberos ticket. If not, you can use the WebHDFS browser.

Note: The **`/sas/ep/config`** directory is created automatically when you run the install script.

SASEP-SERVERS.SH Script

Overview of the SASEP-SERVERS.SH Script

The sasep-servers.sh script enables you to perform the following actions.

- Install or uninstall the SAS Embedded Process and SAS Hadoop MapReduce JAR files on a single node or a group of nodes.
- Start or stop the SAS Embedded Process on a single node or on a group of nodes.
- Determine the status of the SAS Embedded Process on a single node or on a group of nodes.
- Write the installation output to a log file.
- Pass options to the SAS Embedded Process.

Note: The sasep-servers.sh script can be run from any folder on any node in the cluster. You can also add its location to the PATH environment variable.

Note: You must have sudo access to run the sasep-servers.sh script.

SASEP-SERVERS.SH Syntax

sasep-servers.sh

-add | -remove | -start | -stop | -status | -restart
<-mrhome *path-to-mr-home*>
<-hdfsuser *user-id*>
<-epuser>*epuser-id*
<-epgroup>*epgroup-id*
<-hostfile *host-list-filename* | -host <">*host-list*<">>
<-epscript *path-to-ep-install-script*>
<-mrscript *path-to-mr-jar-file-script*>
<-options "*option-list*">
<-log *filename*>
<-version *apache-version-number*>
<-getjars>

Arguments

-add
installs the SAS Embedded Process.

Note The `-add` argument also starts the SAS Embedded Process (same function as -start argument). You are prompted and can choose whether to start the SAS Embedded Process.

Tip You can specify the hosts on which you want to install the SAS Embedded Process by using the `-hostfile` or `-host` option. The `-hostfile` or `-host` options are mutually exclusive.

See -hostfile and -host option on page 25

-remove
removes the SAS Embedded Process.

Tip You can specify the hosts for which you want to remove the SAS Embedded Process by using the `-hostfile` or `-host` option. The `-hostfile` or `-host` options are mutually exclusive.

See -hostfile and -host option on page 25

-start
starts the SAS Embedded Process.

Tip You can specify the hosts on which you want to start the SAS Embedded Process by using the `-hostfile` or `-host` option. The `-hostfile` or `-host` options are mutually exclusive.

See -hostfile and -host option on page 25

-stop
stops the SAS Embedded Process.

Tip You can specify the hosts on which you want to stop the SAS Embedded Process by using the `-hostfile` or `-host` option. The `-hostfile` or `-host` options are mutually exclusive.

See -hostfile and -host option on page 25

-status
provides the status of the SAS Embedded Process on all hosts or the hosts that you specify with either the `-hostfile` or `-host` option.

Tips The status also shows the version and path information for the SAS Embedded Process.

You can specify the hosts for which you want the status of the SAS Embedded Process by using the `-hostfile` or `-host` option. The `-hostfile` or `-host` options are mutually exclusive.

See -hostfile and -host option on page 25

-restart
restarts the SAS Embedded Process.

Tip You can specify the hosts on which you want to restart the SAS Embedded Process by using the `-hostfile` or `-host` option. The `-hostfile` or `-host` options are mutually exclusive.

See -hostfile and -host option on page 25

-mrhome *path-to-mr-home*
specifies the path to the MapReduce home.

-hdfsuser *user-id*
specifies the user ID that has Write access to HDFS root directory.

Default hdfs

Note The user ID is used to copy the SAS Embedded Process configuration files to HDFS.

-epuser *epuser-name*
specifies the name for the SAS Embedded Process user.

Default sasep

-epgroup *epgroup-name*
specifies the name for the SAS Embedded Process group.

Default sasep

-hostfile *host-list-filename*
specifies the full path of a file that contains the list of hosts where the SAS Embedded Process is installed, removed, started, stopped, or status is provided.

Default If you do not specify -hostfile, the sasep-servers.sh script will discover the cluster topology and uses the retrieved list of data nodes.

Tip You can also assign a host list filename to a UNIX variable, **`sas_ephosts_file`**.

```
export sasep_hosts=/etc/hadoop/conf/slaves
```

See "-hdfsuser *user-id*" on page 24

Example
```
-hostfile /etc/hadoop/conf/slaves
```

-host <">*host-list*<">
specifies the target host or host list where the SAS Embedded Process is installed, removed, started, stopped, or status is provided.

Default If you do not specify -host, the sasep-servers.sh script will discover the cluster topology and uses the retrieved list of data nodes.

Requirement If you specify more than one host, the hosts must be enclosed in double quotation marks and separated by spaces.

Tip You can also assign a list of hosts to a UNIX variable, **`sas_ephosts`**.

```
export sasep_hosts="server1 server2 server3"
```

See "-hdfsuser *user-id*" on page 24

Example
```
-host "server1 server2 server3"
-host bluesvr
```

-epscript *path-to-ep-install-script*
copies and unpacks the SAS Embedded Process install script file to the host.

Restriction Use this option only with the `-add` option.

Requirement You must specify either the full or relative path of the SAS Embedded Process install script, tkindbsrv-9.42-*n*_lax.sh file.

Example
```
-epscript /home/hadoop/image/current/tkindbsrv-9.42-1_lax.sh
```

-mrscript *path-to-mr-jar-file-script*
copies and unpacks the SAS Hadoop MapReduce JAR files install script on the hosts.

Restriction Use this option only with the -add option.

Requirement You must specify either the full or relative path of the SAS Hadoop MapReduce JAR file install script, hadoopmrjars-9.42-*n*_lax.sh file.

Example
```
-mrscript /home/hadoop/image/current/tkindbsrv-9.42-1_lax.sh
```

-options "*option-list*"
specifies options that are passed directly to the SAS Embedded Process. The following options can be used.

-trace *trace-level*
specifies what type of trace information is created.

0	no trace log
1	fatal error
2	error with information or data value
3	warning
4	note
5	information as an SQL statement
6	critical and command trace
7	detail trace, lock
8	enter and exit of procedures
9	tedious trace for data types and values
10	trace all information

Default 02

Note The trace log messages are stored in the MapReduce job log.

-port *port-number*
specifies the TCP port number where the SAS Embedded Process accepts connections.

Default 9261

Requirement The options in the list must be separated by spaces, and the list must be enclosed in double quotation marks.

-log *filename*
writes the installation output to the specified filename.

-version *Apache-version-number*
specifies the Hadoop version of the JAR file that you want to install on the cluster. The *apache-version-number* can be one of the following values.

0.23
installs the SAS Hadoop MapReduce JAR files that are built from Apache Hadoop 0.23 (sas.hadoop.ep.apache023.jar and sas.hadoop.ep.apache023.nls.jar).

1.2
installs the SAS Hadoop MapReduce JAR files that are built from Apache Hadoop 1.2.1 (sas.hadoop.ep.apache121.jar and sas.hadoop.ep.apache121.nls.jar).

2.0
installs the SAS Hadoop MapReduce JAR files that are built from Apache Hadoop 0.2.3 (sas.hadoop.ep.apache023.jar and sas.hadoop.ep.apache023.nls.jar).

2.1
installs the SAS Hadoop MapReduce JAR files that are built from Apache Hadoop 2.0.5 (sas.hadoop.ep.apache205.jar and sas.hadoop.ep.apache205.nls.jar).

Default If you do not specify the -version option, the sasep.servers.sh script will detect the version of Hadoop that is in use and install the JAR files associated with that version. For more information, see "Installing the SAS Embedded Process and SAS Hadoop MapReduce JAR Files" on page 19.

Interaction The -version option overrides the version that is automatically detected by the sasep.servers.sh script.

-getjars
creates a HADOOP_JARZIP file in the local folder. This ZIP file contains all required client JAR files.

Starting the SAS Embedded Process

There are three ways to manually start the SAS Embedded Process.

Note: Root authority is required to run the sasep-servers.sh script.

- Run the sasep-servers.sh script with the `-start` option on the master node.

  This starts the SAS Embedded Process on all nodes. For more information about running the sasep-servers.sh script, see "SASEP-SERVERS.SH Syntax" on page 23.

- Run sasep-server-start.sh on a node.

  This starts the SAS Embedded Process on the local node only. The sasep-server-start.sh script is located in the ***SASEPHome*/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/** directory. For more information, see "Installing the SAS Embedded Process and SAS Hadoop MapReduce JAR Files" on page 19.

- Run the UNIX `service` command on a node.

  This starts the SAS Embedded Process on the local node only. The `service` command calls the init script that is located in the **/etc/init.d** directory. A

symbolic link to the init script is created in the `/etc/rc3.d` and `/etc/rc5.d` directories, where `3` and `5` are the run level at which you want the script to be executed.

Because the SAS Embedded Process init script is registered as a service, the SAS Embedded Process is started automatically when the node is rebooted.

Stopping the SAS Embedded Process

The SAS Embedded Process continues to run until it is manually stopped. The ability to control the SAS Embedded Process on individual nodes could be useful when performing maintenance on an individual node.

There are three ways to stop the SAS Embedded Process.

Note: Root authority is required to run the sasep-servers.sh script.

- Run the sasep-servers.sh script with the `-stop` option from the master node.

  This stops the SAS Embedded Process on all nodes. For more information about running the sasep-servers.sh script, see "SASEP-SERVERS.SH Syntax" on page 23.

- Run sasep-server-stop.sh on a node.

  This stops the SAS Embedded Process on the local node only. The sasep-server-stop.sh script is located in the ***SASEPHome*/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/** directory. For more information, see "Installing the SAS Embedded Process and SAS Hadoop MapReduce JAR Files" on page 19.

- Run the UNIX `service` command on a node.

  This stops the SAS Embedded Process on the local node only.

Determining the Status of the SAS Embedded Process

You can display the status of the SAS Embedded Process on one node or all nodes. There are three ways to display the status of the SAS Embedded Process.

Note: Root authority is required to run the sasep-servers.sh script.

- Run the sasep-servers.sh script with the `-status` option from the master node.

  This displays the status of the SAS Embedded Process on all nodes. For more information about running the sasep-servers.sh script, see "SASEP-SERVERS.SH Syntax" on page 23.

- Run sasep-server-status.sh from a node.

  This displays the status of the SAS Embedded Process on the local node only. The sasep-server-status.sh script is located in the ***SASEPHome*/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin/** directory. For more information, see "Installing the SAS Embedded Process and SAS Hadoop MapReduce JAR Files" on page 19.

- Run the UNIX `service` command on a node.

This displays the status of the SAS Embedded Process on the local node only.

Hadoop Permissions

The person who installs the SAS Embedded Process must have sudo access.

5

The SAS Quality Knowledge Base

Introduction

The SAS Quality Knowledge Base (QKB) is a collection of files that store data and logic that support data management operations. SAS software products reference the QKB when performing data management operations on your data. In order to use the Cleanse Data in Hadoop directives in SAS Data Loader for Hadoop, you must deploy a QKB on your Hadoop cluster. This chapter describes how to perform this deployment.

Choosing a QKB

You can deploy a QKB in the Hadoop cluster in one of three ways:

- deploy the QKB that was provided with your SAS Data Loader order
- deploy a QKB that you are already using with other SAS software in your enterprise
- deploy a new QKB downloaded from the software Downloads page (support.sas.com/downloads)

After your initial deployment, it is advisable to update the QKB in your Hadoop cluster periodically to make sure that you are using the latest QKB updates provided by SAS. See “Updating and Customizing the SAS Quality Knowledge Base” on page 33 for more information.

Deployment Overview

SAS Data Loader provides a script for deploying your QKB on the Hadoop cluster. Before you can run this script, you must copy your QKB to the Hadoop cluster. This can be done by transferring the directory structure to the Hadoop master node via FTP, or by mounting the file system where the QKB is located on the Hadoop master node.

It is recommended that you run the script, named qkbpush.sh, on the Hadoop master node (Name Node). The script automatically discovers all nodes in the cluster and deploys the QKB on them by default. Flags are available to enable you to deploy the QKB on individual nodes, or on a subset of nodes instead.

The qkbpush.sh script performs two tasks:

- It copies the specified QKB directory to a fixed location (`/opt/qkb/default`) on the specified nodes and sets the QKB's permissions so that the QKB is owned by the user account that is owned by the SAS Embedded Process.
- It generates an index file from the contents of the QKB and pushes this index file to HDFS. This index file, named default.idx, is created in the `/sas/qkb` directory in HDFS. The default.idx file provides a list of QKB definition and token names to SAS Data Loader. SAS Data Loader surfaces the names in its graphical user interface.

Creating the index file requires special permissions in a Kerberos security environment. For more information, see "Kerberos Security Requirements" on page 32.

Only one QKB and one index file are supported in the Hadoop framework at a time. Subsequent QKB and index pushes replace prior ones.

After the QKB deployment is complete, you must restart the SAS Embedded Process on each Hadoop node so that each instance of the SAS Embedded Process loads the newly deployed QKB. Use the sasep-servers.sh script to restart the SAS Embedded Process. For information about the sasep-servers.sh script, see the information for Hadoop in the *SAS In-Database Products: Administrator's Guide*.

Kerberos Security Requirements

In a Kerberos environment, a Kerberos ticket (TGT) is necessary to run the qkbpush.sh script. This script copies the QKB to each data node in the cluster and uploads metadata about it to HDFS.

To create the ticket:

1. Log on as root.
2. Change to the hdfs user.
3. Run kinit.
4. Exit back to root.

5 Run the qkbpush.sh script.

The following are examples of commands that you might use to obtain the ticket.

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

Note: You must supply the root password for the first command.

Using the QKBPUSH.SH Script

The qkbpush.sh script is located in the ***<SASEPHome>*/SAS/SASTKInDatabaseServerForHadoop/9.42-1/bin** directory of your In-Database Technologies package installation. It can be run at any time to put a new QKB in the cluster. However, the SAS Embedded Process must be restarted afterward for the changes to take effect. You must execute the script from the ***<SASEPHome>*** directory.

The following are sample commands:

- To automatically deploy the QKB to all nodes in the Hadoop cluster, execute:

  ```
  qkbpush.sh qkb_path
  ```

- To deploy to one or more nodes on a command line, execute:

  ```
  qkbpush.sh -h hostname1 [-h hostname2] qkb_path
  ```

- To deploy using a file containing a list of node names, execute:

  ```
  qkbpush.sh -f hostfile qkb_path
  ```

- To deploy to one or more nodes on a command line and suppress QKB index creation, execute:

  ```
  qkbpush.sh -i -h hostname1 [-h hostname2] qkb_path
  ```

For ***qkb_path***, specify the path to the QKB source directory. See "QKBPUSH.SH: Reference" on page 34 for syntax details.

Updating and Customizing the SAS Quality Knowledge Base

SAS provides regular updates to the QKB. It is recommended that you update your QKB each time a new one is released. For a listing of the latest enhancements to the QKB, refer to "What's New in SAS Quality Knowledge Base." The What's New document is available on the SAS Quality Knowledge Base product documentation page at support.sas.com. To find this page, either search on the name SAS Quality Knowledge Base or locate the name in the product index and click the Documentation tab. Check the What's New for each QKB to determine which definitions have been added, modified, or deprecated, and to learn about new locales that might be supported. Contact your SAS software representative to order updated QKBs and locales.Use the qkbpush.sh script to load updates as described in "Deployment Overview" on page 32.

The definitions delivered in the QKB are sufficient for performing most data quality operations. However, if you have DataFlux Data Management Studio, you can use the Customize feature to modify your QKB to meet specific needs. See your SAS representative for information to license DataFlux Data Management Studio.

If you want to customize your QKB, then, as a best practice, we recommend that you customize your QKB on a local workstation before copying it to the Hadoop master node for deployment. Then, when updates to the QKB are required, you can merge your customizations into an updated QKB locally and deploy a copy of the updated, customized QKB to the Hadoop cluster as you would a standard QKB. Refer to the online Help provided with your SAS Quality Knowledge Base for information about how to merge any customizations that you have made into an updated QKB.

Note: To add support for additional QKB locales, you must license the new locales and then reinstall and re-deploy the QKB.

Removing the QKB from Hadoop

The QKB can be removed from a Hadoop cluster by executing the qkbpush.sh script with the `-r` flag. The `-r` flag removes the QKB index file from HDFS and the QKB from all nodes by default. Specify the `-h` or `-f` flags in conjunction with the `-r` flag, as appropriate, to remove the QKB from a subset of the nodes. After running the script, restart the SAS Embedded Process, so that the changes can take effect.

The QKB index file is not removed from HDFS when the `-h` or `-f` flag is specified.

QKBPUSH.SH: Reference

Overview

The qkbpush.sh script deploys a SAS Quality Knowledge Base in a Hadoop cluster. When executed without options, the script automatically discovers all nodes in the cluster and deploys the QKB on them. The script also generates an index file from the contents of the QKB and pushes this index file to HDFS. Flags are available that enable you to deploy the QKB on individual nodes or on a subset of nodes instead. Flags are also available to enable you to suppress index creation or perform only index creation.

qkbpush.sh should be run as the root user. It becomes the HDFS user in order to detect the nodes in the cluster. It sets the QKB permissions on the Hadoop nodes so that the QKB is owned by the default user name for the SAS Embedded Process. Flags are available to specify the HDFS and SAS Embedded Process user names if user names other than the defaults were configured.

To simplify maintenance, the QKB directory is copied to a fixed location (`/opt/qkb/default`) on each node. The QKB index file is created in the `/sas/qkb` directory in HDFS.

Syntax

```
qkbpush <options> qkb_path
```

Required Arguments

qkb_path
specifies the path to the source QKB directory.

Optional Arguments

Authentication Options

-s *hdfs-user*
specifies the user name to associate with HDFS, when the default user name (`hdfs`) is not used.

-g *ep-group*
specifies the group name to associate with the SAS Embedded Process, when the default group name (`sasep`) is not used.

-u *ep-user*
specifies the user name to associate with the SAS Embedded Process, when the default user name (`sasep`) is not used.

QKB Index Options

-i
creates and pushes the QKB index only.

-x
suppresses QKB index creation.

Subsetting Options

-h *hostname*
specifies the host name or IP address of the computer on which to perform the deployment.

-f *hostfile*
specifies the name of a file that contains a list of the host names or IP addresses on which to perform the deployment.

General Options

-?
prints usage information.

-l *logfile*
directs status information to a log file, instead of to standard output.

-r
removes the QKB from the Hadoop nodes and the QKB index file from HDFS.

-v
specifies verbose output.

Troubleshooting the QKB Deployment

The QKB deployment can fail for the following reasons:

- The SAS Embedded Process was not restarted after the QKB was deployed.
- You did not obtain a Kerberos ticket before attempting to run the qkbpush.sh script in a Kerberos environment.
- You did not execute the qkbpush.sh script as the root user.
- You executed the qkbpush.sh script from the `<SASHome>` directory. The script must be run from `<SASEPHome>`.
- You had insufficient space in the `/tmp` directory for qkbpush.sh to run. (Please clear space and try again.)
- You did not specify the correct user name for the SAS Embedded Process.
- The QKB does not contain the intended locales. See “Updating and Customizing the SAS Quality Knowledge Base” on page 33 for information about how locales are obtained. Try running the qkbpush.sh script again and restart the SAS Embedded process. Verify that you executed the script as the root user and with the user name representing the SAS Embedded Process.

6

Security

Overview

If your enterprise uses Kerberos security, you must take specific steps to configure it to allow authentication to flow from the client machine that is hosting the SAS Data Loader for Hadoop vApp virtual machine through to the Hadoop cluster.

Note: SAS Data Loader for Hadoop does not provide Kerberos validation. All of the following configuration values must be entered correctly in the SAS Data Loader for Hadoop vApp or errors result during its operation.

Client Configuration

Host Name

Client authentication using Kerberos requires:

- accessing SAS Data Loader for Hadoop using a host name, not an IP address
- configuring the browser to use Kerberos when accessing the vApp host name

Accessing the vApp using a host name depends on the client browser being able to resolve the host name to the internal NAT IP of the SAS Data Loader for

Hadoop vApp. You must create a host name for use on the client machine. For example, you might create a name similar to dltest1.vapps.sas.com.

Hosts File

You must modify the hosts file on the client machine to include the host name. You must also modify this file to include the static IP address of the vApp that is installed on the host. VMware Player Pro displays this address in a welcome window when the vApp is started on the client machine. The hosts file requiring modification is: `%SystemRoot%\system32\drivers\etc\hosts`. The editor must run in UAC-permitted mode. This requires administrative privileges on the machine. To modify the file:

1. Click the **Start** button.
2. Enter `notepad %SystemRoot%\system32\drivers\etc\hosts` in the search box.
3. Press Ctrl+Shift+Enter to execute as the administrator.
4. Accept the UAC prompt.
5. Enter the host name and IP address in the proper format. For example, you might enter `192.168.212.132 dltest1.vapps.sas.com`.

   Note: The IP address of the vApp can change. Anytime the IP changes, you must repeat this process.

Supported Browsers and Integrated Windows Authentication

SAS Data Loader for Hadoop supports the Firefox and Chrome browsers for single sign-on. The browser must be configured to support Integrated Windows Authentication (IWA). See Support for Integrated Windows Authentication for more information.

The browser on the client vApp machine must be configured as follows:

Firefox

1. Enter `about:config` in the address bar.
2. Enter `negotiate` in the filter text box.
3. Set the `network.negotiate-auth.delegation-uris` value to the domain of the host name assigned to the vApp.
4. Set the network.negotiate-auth.trusted-uris value to the domain of the host name assigned to the vApp.
5. Close the browser.

Chrome

1. Close all open Chrome browser instances.
2. Open **Control Panel** ▸ **Internet Options** from the Windows Start menu.
3. Click the **Security** tab.

4 Click **Local intranet**.

5 Click **Sites**, and then click **Advanced**.

6 Enter the domain of the host name assigned to the vApp in the **Add this website to the zone** field.

7 Click **Add**, click **Close**, and then click **OK**.

8 Click the **Advanced** tab.

9 Scroll down to the Security section.

10 Select the **Enable Integrated Windows Authentication** option.

11 Click **OK** to close the Internet Properties Control Panel.

12 Click the Windows **Start** button.

13 Enter `regedit` in the search box, and then press the Enter key.

14 In the Registry Editor, expand **HKEY_LOCAL_MACHINE**, and then expand **SOFTWARE**.

15 Right-click **Policies**, and then select **New** ▸ **Key**.

16 As appropriate, enter `Google` as the name of the new key.

17 As appropriate, right-click `Google`, and then select **New** ▸ **Key**.

18 As appropriate, enter `Chrome` as the name of the new key.

19 Right-click `Chrome`, and then select **New** ▸ **String Value**. The right pane displays a new registry entry of type REG_SZ.

20 Enter the following name for the new string value:
`AuthNegotiateDelegateWhitelist`

21 Right-click `AuthNegotiateDelegateWhitelist` and select **Modify**.

22 In the Edit String window, in the **Value data** field, enter the host name that is or will be used in Kerberos to refer to the client.

23 Click **OK** to close the Edit String window.

24 Exit the Registry Editor.

25 Restart the Chrome browser.

Kerberos Configuration

Overview

The Kerberos topology contains multiple tiers, all of which are configured to communicate with the Kerberos Key Distribution Center (KDC) to allow authentication to flow from the SAS Data Loader for Hadoop client machine through to the Hadoop cluster. When you log on to the client machine, the KDC

issues a ticket granting ticket (TGT), which is time stamped. This TGT is used by the browser to issue a ticket to access SAS Data Loader for Hadoop.

Two different types of Kerberos systems are available: AD (Windows Active Directory) and MIT. You might have either a realm for only AD Kerberos or mixed AD and MIT realms. A realm for only AD Kerberos protects the client machine, the vApp virtual machine, and the Hadoop cluster all through the AD domain controller. A realm for only AD Kerberos is simpler because it requires no further client configuration.

In a common configuration of mixed realms, AD Kerberos protects both the client machine and the vApp virtual machine, whereas MIT Kerberos protects only the Hadoop cluster. The mixed realms can be configured such that AD Kerberos protects only the client machine, whereas MIT Kerberos protects both the Hadoop cluster and the vApp virtual machine. Which realm configuration is in use determines how you must configure Kerberos.

vApp

Overview

You must generate a Service Principal Name (SPN) and Kerberos keytab for the host, SAS, and HTTP service instances.

The following SPNs must be created to allow ticket delegation, where *hostname* represents the host name that you have created and *krbrealm* represents your Kerberos realm:

- host/*hostname*@*krbrealm*.
- SAS/*hostname*@*krbrealm*. This allows single sign-on from the mid-tier to the SAS Object Spawner.
- HTTP/*hostname*@*krbrealm*. This allows single sign-on with tc Server and the SASLogon web application.

Protecting the vApp with MIT Kerberos

When protecting the vApp using MIT Kerberos, you must configure the client machine to acquire tickets for the vApp from the correct realm. To do this, you must run the **ksetup** command to add a KDC and to assign the vApp host name to that KDC. For example, if the KDC host is server2.unx.zzz.com and the host name is dladtest1.vapps.zzz.com, issue the following commands:

```
ksetup /AddKdc DMM.KRB.ZZZ.COM server2.unx.zzz.com
ksetup /AddHostToRealmMap dladtest1.vapps.zzz.com DMM.KRB.SAS.COM
```

On a machine that is configured to communicate with the MIT Kerberos realm, generate the three SPNs and corresponding keytabs. For example, if the fully qualified domain name is dladtest1.vapps.zzz.com issue the following commands:

```
$ kadmin -p user2/admin -kt /opt/keytabs/admin/user2.dmm.keytab
  kadmin: addprinc -randkey +ok_as_delegate host/dladtest1.vapps.zzz.com
  kadmin: ktadd -k $hostname/host.dladtest1.keytab host/dladtest1.vapps.zzz.com
  kadmin: addprinc -randkey +ok_as_delegate SAS/dladtest1.vapps.zzzcom
  kadmin: ktadd -k $hostname/SAS.dladtest1.keytab SAS/dladtest1.vapps.zzz.com
  kadmin: addprinc -randkey +ok_as_delegate HTTP/dladtest1.vapps.zzz.com
  kadmin: ktadd -k $hostname/HTTP.dladtest1.keytab HTTP/dladtest1.vapps.zzz.com
```

Protecting the vApp with AD Kerberos

To generate SPNs and keytabs in AD Kerberos on Windows Server 2012, you must have administrator access to the Windows domain and do the following:

1 Create SPN users.

   a Launch the Server Manager on the domain controller.

   b Select **Tools** ▸ **Active Directory Users and Computers**.

   c Select **<domain name>** ▸ **Managed Service Accounts**.

   d In the right pane, click **New** ▸ **User**.

   e In the **User logon name** field, enter **`host/`*`fully-qualified-hostname`*** . For example, enter **`host/dladtest1.vapps.zzz.com`**, and then click **Next**.

   f Enter and confirm a password.

   g If you are configuring a server with an operating system older than Windows 2000, change the logon name to **`HTTP/`*`simple-hostname`*** . For example, enter **`host/dladtest1`**.

   h Deselect **User must change password at next logon** and the select **Password never expires**.

   i Click **Finish**.

   j Repeat the previous steps for SAS and HTTP SPN users.

2 Create SPNs for each SPN user. At a command prompt on the domain controller, enter the following commands using a fully qualified host name and simple host name. For example, you might use dladtest1.vapps.zzz.com and **`dladtest1`**:

```
> setspn -A host/dladtest1.vapps.zzz.com HTTP_dladtest1
> setspn -A SAS/dladtest1.vapps.zzz.com SAS_dladtest1
> setspn -A HTTP/dladtest1.vapps.zzz.com host_dladtest1
```

3 Authorize ticket delegation:

   a Launch the Server Manager on the domain controller.

   b Select **View** ▸ **Advanced Features**.

   c Select **host/<vapp> user** ▸ **Properties**.

   d On the **Delegation** tab, select **Trust this user for delegation to any service (Kerberos only)**, and then click **OK**.

   e Select **host/<vapp> user** ▸ **Properties**.

   f On the **Attribute Editor** tab, locate the **msDS-KeyVersionNumber** attribute. Record this number.

   g Repeat the previous steps to authorize ticket delegation for the SAS and HTTP users.

4 Create keytabs for each SPN:

- On UNIX:

  1 At a command prompt, use the ktutil utility to create keytabs. Enter the following commands using a fully qualified host name, the realm for your domain, the password that you created, and the msDS-KeyVersionNumber. In the following host SPN keytab example, **dladtest1.vapps.zzz.com**, **PROXY.KRB.ZZZ.COM**, **Psword**, and **-k 2 -e arcfour-hmac** are used for these values:

```
ktutil
ktutil:  addent -password -p host/dladtest1.vapps.zzz.com@PROXY.KRB.ZZZ.COM -k 2 -e arcfour-hmac
Psword for host/dladtest1.vapps.zzz.com@PROXY.KRB.ZZZ.COM :
ktutil:  addent -password -p host/dladtest1.vapps.zzz.com@PROXY.KRB.ZZZ.COM -k 2 -e aes128-cts-hmac-sha1-96
Psword for host/dladtest1.host.zzz.com@PROXY.KRB.ZZZ.COM :
ktutil:  addent -password -p host/dladtest1.vapps.zzz.com@PROXY.KRB.ZZZ.COM -k 2 -e aes256-cts-hmac-sha1-96
Psword for host/dladtest1.vapps.zzz.com@PROXY.KRB.ZZZ.COM :
ktutil:  wkt host.dladtest1.keytab
ktutil:  quit
```

  2 Repeat the previous steps to create the SAS and HTTP keytabs.

- On Windows:

  1 At a command prompt, use the ktpass utility to create keytabs. Enter the following commands using a fully qualified host name, the realm for your domain, and any password (it does not have to be the password that you created earlier). In the following host SPN keytab example, **dladtest1.vapps.zzz.com**, **NA.ZZZ.COM**, and **Psword** are used for these values:

```
ktpass.exe -princ host/dladtest1.vapps.zzz.com@NA.ZZZ.COM -mapUser Server\dladtest1-host -pass "Psword"
         -pType KRB5_NT_PRINCIPAL -out dladtest1-host.keytab -crypto All
```

  2 Repeat the previous steps to create the SAS and HTTP keytabs.

Hadoop

Overview

The Hadoop cluster must be configured for Kerberos according to the instructions provided for the specific distribution that you are using.

Hortonworks

You must make the following specific change for Hortonworks:

```
* hive.server2.enable.doAs = true
```

When a Hortonworks cluster is protected by MIT Kerberos, you must set auth_to_local as follows:

```
RULE:[1:$1@$0](.*@\AD_DOMAIN_REALM\E$)s/@\AD_DOMAIN_REALM\E$//
RULE:[2:$1@$0](.*@\AD_DOMAIN_REALM\E$)s/@\AD_DOMAIN_REALM\E$//
* hive.security.authorization.enabled = false
* hadoop.proxyuser.HTTP.hosts = *
* hadoop.proxyuser.HTTP.groups = *
* hadoop.proxyuser.hive.groups = *
```

Cloudera

When a Cloudera cluster is protected by MIT Kerberos, add `AD_DOMAIN_REALM` to Trusted Kerberos Realms under the HDFS configuration.

SAS LASR Analytic Server

Integration of SAS Data Loader for Hadoop with a SAS LASR Analytic Server is possible only in an AD Kerberos environment. SAS Data Loader for Hadoop cannot be integrated with SAS LASR Analytic Server in a mixed AD and MIT Kerberos environment.

A public key is created as part of SAS Data Loader for Hadoop vApp configuration and is placed in the SAS Data Loader for Hadoop shared folder. This public key must also exist on the SAS LASR Analytic Server grid. The public key must be appended to the authorized_keys file in the .ssh directory of that user.

End-User Support

The SAS Data Loader for Hadoop vApp that runs on the client machine contains a **Settings** dialog box in the SAS Data Loader: Information Center. See the *SAS Data Loader for Hadoop: User's Guide* and the *SAS Data Loader for Hadoop: vApp Deployment Guide* for more information about the **Settings** dialog box. The dialog box contains certain fields for which you must provide values to the vApp user. These fields are as follows:

Table 6.1 *Settings Fields*

Field	Value
Hostname	The host name that you create for Kerberos security.. See "Client Configuration".
User id for host log in	The normal logon option ID for the user. .
Realm for user id	The name of the Kerberos realm or AD domain against which the user authenticates.
krb5 configuration	The location of the Kerberos configuration file.
Host keytab	The location of the keytab generated for the host SPN. See "vApp".
SAS server keytab	The location of the keytab generated for the SAS server SPN. See "vApp".
HTTP keytab	The location of the keytab generated for the HTTP SPN. See "vApp".
Local JCE security policy jar	The location of the local Java Cryptography Extension files. See http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html for more information.

Field	Value
US JCE security policy jar	The location of the U.S. Java Cryptography Extension files. See http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html for more information.

You must provide the krb5 configuration, keytab, and JCE files and notify the user of their locations.

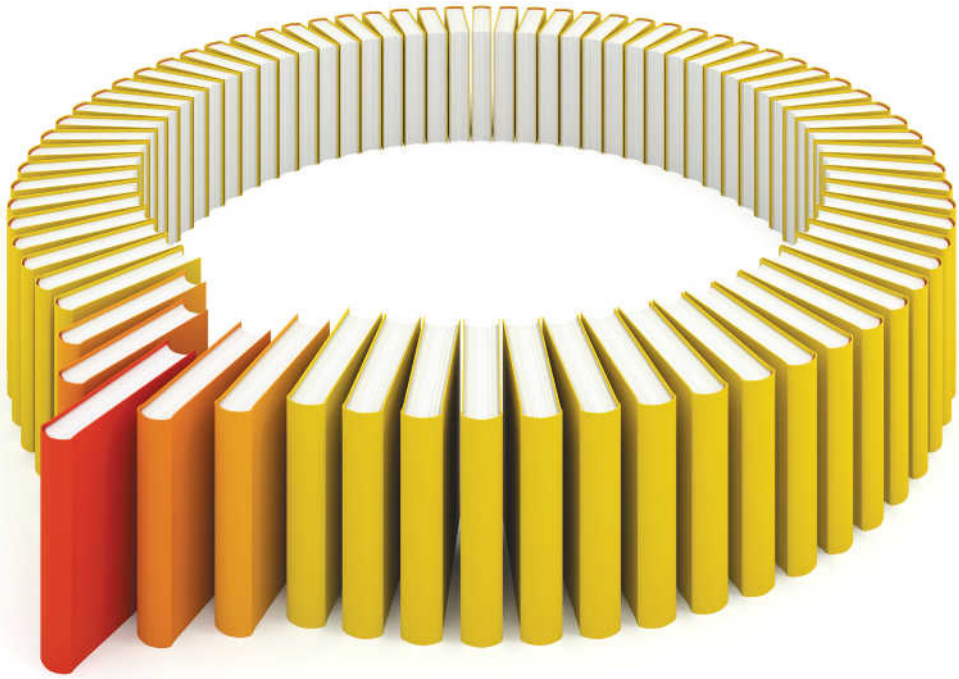
Recommended Reading

- *SAS Data Loader for Hadoop: User's Guide*
- *SAS Data Loader for Hadoop: vApp Deployment Guide*
- *SAS In-Database Products: Administrator's Guide*
- *SAS Hadoop Configuration Guide for Base SAS and SAS/ACCESS*
- *The Little SAS Book: A Primer, Fifth Edition*
- *Introduction to SAS and Hadoop Course Notes*

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



Gain Greater Insight into Your SAS® Software with SAS Books.

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

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