



THE
POWER
TO KNOW.

SAS[®] 9.4 In-Database Products Administrator's Guide

Seventh Edition

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS® 9.4 In-Database Products: Administrator's Guide, Seventh Edition*. Cary, NC: SAS Institute Inc.

SAS® 9.4 In-Database Products: Administrator's Guide, Seventh Edition

Copyright © 2016, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a) and DFAR 227.7202-4 and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

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

June 2016

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

With respect to CENTOS third-party technology included with the vApp ("CENTOS"), CENTOS is open-source software that is used with the Software and is not owned by SAS. Use, copying, distribution, and modification of CENTOS is governed by the CENTOS EULA and the GNU General Public License (GPL) version 2.0. The CENTOS EULA can be found at http://mirror.centos.org/centos/6/os/x86_64/EULA. A copy of the GPL license can be found at <http://www.opensource.org/licenses/gpl-2.0> or can be obtained by writing to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02110-1301 USA. The source code for CENTOS is available at <http://vault.centos.org/>.

With respect to open-vm-tools third party technology included in the vApp ("VMTOOLS"), VMTOOLS is open-source software that is used with the Software and is not owned by SAS. Use, copying, distribution, and modification of VMTOOLS is governed by the GNU General Public License (GPL) version 2.0. A copy of the GPL license can be found at <http://opensource.org/licenses/gpl-2.0> or can be obtained by writing to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02110-1301 USA. The source code for VMTOOLS is available at <http://sourceforge.net/projects/open-vm-tools/>.

With respect to VIRTUALBOX third-party technology included in the vApp ("VIRTUALBOX"), VIRTUALBOX is open-source software that is used with the Software and is not owned by SAS. Use, copying, distribution, and modification of VIRTUALBOX is governed by the GNU General Public License (GPL) version 2.0. A copy of the GPL license can be found at <http://opensource.org/licenses/gpl-2.0> or can be obtained by writing to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02110-1301 USA. The source code for VIRTUALBOX is available at <http://www.virtualbox.org/>.

Contents

<i>What's New in SAS 9.4 In-Database Products: Administrator's Guide</i>	<i>ix</i>
--	-----------

PART 1 Introduction 1

Chapter 1 • Introduction to the Administrator's Guide	3
Overview of SAS In-Database Products	3
What Is Covered in This Document?	4

PART 2 Administrator's Guide for Hadoop (In-Database Deployment Package) 5

Chapter 2 • In-Database Deployment Package for Hadoop	7
Overview of the In-Database Deployment Package for Hadoop	7
Overview of the SAS Embedded Process	8
Prerequisites for Installing the In-Database Deployment Package for Hadoop	8
Backward Compatibility	9
Hadoop Permissions	9
Documentation for Using In-Database Processing in Hadoop	10

Chapter 3 • Deploying the In-Database Deployment Package Using the SAS Deployment Manager	11
When to Deploy the SAS In-Database Deployment Package	
Using the SAS Deployment Manager	11
Prerequisites for Using the SAS Deployment Manager to Deploy the In-Database Deployment Package	12
Hadoop Installation and Configuration Steps Using the SAS Deployment Manager	13
Upgrading from or Reinstalling a Previous Version	14
Using the SAS Deployment Manager to Create the SAS Embedded Process Parcel or Stack	18
Deploying the SAS Embedded Process Parcel on Cloudera	29
Deploying the SAS Embedded Process Stack on Hortonworks	30

Chapter 4 • Deploying the In-Database Deployment Package Manually	35
When to Deploy the SAS In-Database Deployment Package Manually	35
Hadoop Manual Installation and Configuration Steps	36
Upgrading from or Reinstalling a Previous Version	37
Copying the SAS Embedded Process Install Script to the Hadoop Cluster	40
Installing the SAS Embedded Process	41
SASEP-ADMIN.SH Script	43

Chapter 5 • Additional Configuration for the SAS Embedded Process	49
Overview of Additional Configuration Tasks	49
Additional Configuration Needed to Use HCatalog File Formats	50
Additional Configuration for Hortonworks 2.2	52

Additional Configuration for IBM BigInsights	53
Adding the YARN Application CLASSPATH to the Configuration File for MapR Distributions	54
Adjusting the SAS Embedded Process Performance	54
Adding the SAS Embedded Process to Nodes after the Initial Deployment	56

PART 3 Administrator's Guide for SAS Data Loader for Hadoop 57

Chapter 6 • Introduction	59
SAS Data Loader and SAS In-Database Technologies for Hadoop	59
System Requirements	60
Privileges	60
Support for the vApp User	60
Deployment	60
Chapter 7 • Cloudera Manager and Ambari Deployment	61
Getting Started with the Deployment	61
Configure Kerberos	62
Obtain and Extract Zipped Files	63
Deploy Files	63
Edit SAS Hadoop Configuration Properties File	67
Collect Files	68
Configure the Hadoop Cluster	68
Deactivating or Removing Existing Versions	69
Chapter 8 • Standard Deployment	73
Getting Started with the Deployment	73
Configure Kerberos	74
Create Parcels or Stacks	75
Deploy Files	75
Collect Files	75
Configure the Hadoop Cluster	77
Hot Fixes and SAS Notes	77
Chapter 9 • SAS In-Database Deployment Package for Hadoop	79
About the SAS In-Database Deployment Package	79
Getting Started	79
Configure Kerberos	80
Manual Installation	80
Collect Files	81
Install Additional Components	81
Configure the Hadoop Cluster	81
Hot Fixes and SAS Notes	81
Chapter 10 • SAS In-Database Technologies for Data Quality Directives	83
About SAS In-Database Technologies for Data Quality Directives	83
Getting Started	84
Configure Kerberos	84
Manual Installation	85
Install Additional Components	94
Configure the Hadoop Cluster	94

Chapter 11 • SAS Data Management Accelerator for Spark	95
About SAS Data Management Accelerator for Spark	95
Getting Started	95
Configure Kerberos	96
Manual Installation	97
Install Additional Components	103
Configure the Hadoop Cluster	103
Chapter 12 • Configuring the Hadoop Cluster	105
Configuring Components on the Cluster	105
Providing vApp User Configuration Information	108
Chapter 13 • Configuring Kerberos	111
About Kerberos on the Hadoop Cluster	111
Client Configuration	111
Kerberos Configuration	112
Providing vApp User Configuration Information	115
 PART 4 Administrator's Guide for Teradata	 117
Chapter 14 • In-Database Deployment Package for Teradata	119
Prerequisites	119
Overview of the In-Database Deployment Package for Teradata	119
Required Hot Fixes for the SAS In-Database Code Accelerator for Teradata 9.41	121
Teradata Permissions for Publishing Formats and Scoring Models	123
Documentation for Using In-Database Processing in Teradata	124
Chapter 15 • Deploying the SAS Embedded Process: Teradata	125
Teradata Installation and Configuration Steps	125
Upgrading from or Reinstalling a Previous Version	126
Installing the SAS Formats Library and the SAS Embedded Process	129
Controlling the SAS Embedded Process	131
Chapter 16 • SAS Data Quality Accelerator for Teradata	133
Introduction	133
Upgrading from or Re-Installing a Previous Version of the SAS Data Quality Accelerator	134
SAS Data Quality Accelerator and QKB Deployment Steps	134
Obtaining a QKB	135
Understanding Your SAS Data Quality Accelerator Software Installation	135
Packaging the QKB	136
Installing the Package Files with the Teradata Parallel Upgrade Tool	137
Creating and Managing SAS Data Quality Accelerator Stored Procedures in the Teradata Database	138
Creating the Data Quality Stored Procedures	139
Granting Users Authorization to the Data Quality Stored Procedures	139
Validating the Accelerator Installation	140
Troubleshooting the Accelerator Installation	141
Updating and Customizing a QKB	142
Removing the Data Quality Stored Procedures from the Database	143

PART 5 Administrator's Guides for Aster, DB2, Greenplum, Netezza, Oracle, SAP HANA, and SPD Server 145

Chapter 17 • Administrator's Guide for Aster	147
In-Database Deployment Package for Aster	147
Aster Installation and Configuration	148
Validating the Publishing of the SAS_SCORE() and the SAS_PUT() Functions	151
Aster Permissions	151
Documentation for Using In-Database Processing in Aster	151
Chapter 18 • Administrator's Guide for DB2	153
In-Database Deployment Package for DB2	153
Function Publishing Process in DB2	154
DB2 Installation and Configuration	155
Validating the Publishing of SAS_COMPILEUDF and SAS_DELETEUDF Functions and Global Variables	171
DB2 Permissions	172
Documentation for Using In-Database Processing in DB2	173
Chapter 19 • Administrator's Guide for Greenplum	175
In-Database Deployment Package for Greenplum	175
Greenplum Installation and Configuration	177
Validation of Publishing Functions	189
Controlling the SAS Embedded Process	190
Semaphore Requirements When Using the SAS Embedded Process for Greenplum	191
Greenplum Permissions	192
Documentation for Using In-Database Processing in Greenplum	192
Chapter 20 • Administrator's Guide for Netezza	193
In-Database Deployment Package for Netezza	193
Function Publishing Process in Netezza	194
Netezza Installation and Configuration	195
Netezza Permissions	205
Documentation for Using In-Database Processing in Netezza	207
Chapter 21 • Administrator's Guide for Oracle	209
In-Database Deployment Package for Oracle	209
Oracle Installation and Configuration	210
Oracle Permissions	213
Documentation for Using In-Database Processing in Oracle	213
Chapter 22 • Administrator's Guide for SAP HANA	215
In-Database Deployment Package for SAP HANA	215
SAP HANA Installation and Configuration	216
Installing the SASLINK AFL Plugins on the Appliance (HANA SPS08)	219
Installing the SASLINK AFL Plugins on the Appliance (HANA SPS09)	221
Importing the SAS_EP Stored Procedure	222
Auxiliary Wrapper Generator and Eraser Procedures	222
Controlling the SAS Embedded Process	223
Semaphore Requirements When Using the SAS Embedded Process for SAP HANA	224
SAP HANA Permissions	224
Documentation for Using In-Database Processing in SAP HANA	225

Chapter 23 • Administrator's Guide for SPD Server	227
Installation and Configuration Requirements for the SAS Scoring Accelerator for SPD Server	227
Where to Go from Here	227
 PART 6 Configurations for SAS Model Manager	229
Chapter 24 • Configuring SAS Model Manager	231
Preparing a Data Management System for Use with SAS Model Manager	231
Configuring a Database	232
Configuring a Hadoop Distributed File System	235
 Recommended Reading	237
Index	239

What's New in SAS 9.4 In-Database Products: Administrator's Guide

Overview

In SAS 9.4, the following new features and enhancements were added to expand the capabilities of the SAS In-Database products:

- In the January 2016 release of SAS 9.4, the following changes and enhancements were made:
 - The removal of the SAS Embedded Process stack using Ambari has been simplified. The `delete_stack.sh` file now enables you to remove the `ep-config.xml` file, a specific version of the SAS Embedded Process, or all versions of the SAS Embedded Process.
 - A new panel that lists the products being installed by the SAS Deployment Manager has been added. Information that appears on other panels also reflects more closely what is being installed.
 - If you are operating in a Spark environment with the SAS Data Loader for Hadoop, you can deploy the SAS Data Management Accelerator for Spark using the SAS Deployment Manager.
 - You can now use the SAS Deployment Manager with the SAS Data Loader for Hadoop to deploy either the CI or PD Quality Knowledge Base (QKB).
- In the July 2015 release of SAS 9.4, the following changes and enhancements were made:
 - The installation and configuration of the SAS Embedded Process for Teradata has changed. The in-database deployment package is delivered to the client from the SAS Install Depot. The new process has a smaller client footprint and is a faster install process.
 - The installation and configuration of the SAS Embedded Process for Hadoop has changed significantly. For Cloudera and Hortonworks, Cloudera Manager and Ambari are used to install the SAS Embedded Process and the SAS Hadoop MapReduce JAR files. For IBM BigInsights, MapR, and Pivotal HD, the in-database deployment package is delivered to the client from the SAS Install Depot. In addition, the SAS Embedded Process and the SAS Hadoop MapReduce JAR files are installed with one script instead of in two separate scripts. The new process has a smaller client footprint and is a faster install process.
- In the August 2014 release of SAS 9.4, the following changes and enhancements were made:
 - Numerous changes were made to the installation and configuration script for the SAS Embedded Process for Hadoop.

- In the April 2014 release of SAS 9.4, documentation enhancements were made in the following areas:
 - Additional information about the installation and configuration of the SAS Embedded Process for Hadoop was added.
 - Added semaphore requirements when using the SAS Embedded Process for Greenplum.
- In the December 2013 release of SAS 9.4, the following changes and enhancements were made:
 - New Hadoop JAR files are now tied to the version of Apache Hadoop that you are using.
- In the June 2013 release of SAS 9.4, the following changes and enhancements were made:
 - In-database scoring for Netezza has been enhanced by the addition of the SAS Embedded Process. The SAS Embedded Process is a SAS server process that runs within Netezza to read and write data.
 - The Hadoop scripts that install, control, and provide a status of the SAS Embedded Process have changed. There is now just one script, sasep-server.sh, that installs both the SAS Embedded Process and the Hadoop JAR files.

SAS In-Database Code Accelerator

SAS 9.4: Changes and Enhancements

The SAS In-Database Code Accelerator must be licensed at your site.

Greenplum Changes

April 2014 Release of SAS 9.4: Changes and Enhancements

Information about semaphore requirements when using the SAS Embedded Process was added to *SAS In-Database Products: Administrator's Guide*.

SAS 9.4: Changes and Enhancements

There are several changes for Greenplum:

- Version 1.2 of the Greenplum Partner Connector (GPPC) is now available and should be installed if you use SAS Embedded Process 9.4.
- A new script, UninstallSASEPFiles.sh, is available. This script stops and uninstalls the SAS Embedded Process on each database host node.

Hadoop Changes

January 2016 Release of SAS 9.4: Changes and Enhancements

In the January 2016 release of SAS 9.4, the following changes and enhancements were made:

- The removal of the SAS Embedded Process stack using Ambari has been simplified. The delete_stack.sh file now enables you to remove the ep-config.xml file, a specific version of the SAS Embedded Process, or all versions of the SAS Embedded Process.
- A new panel that lists the products being installed by the SAS Deployment Manager has been added. Information that appears on other panels also reflects more closely what is being installed.

July 2015 Release of SAS 9.4: Changes and Enhancements

The installation and configuration of the SAS Embedded Process for Hadoop has changed.

- For Cloudera and Hortonworks, Cloudera Manager and Ambari are used to install the SAS Embedded Process and the SAS Hadoop MapReduce JAR files.
- For IBM BigInsights, MapR, and Pivotal HD, the in-database deployment package is delivered to the client from the SAS Install Depot.
- The SAS Embedded Process and the SAS Hadoop MapReduce JAR files are installed with one script instead of in two separate scripts. The new process has a smaller client footprint and is a faster install process.
- The sasep-servers.sh file has changed names to the sasep-admin.sh file. Some of the scripts arguments are no longer needed and have been deleted. Other arguments have been added.

August 2014 Release of SAS 9.4: Changes and Enhancements

In the August 2014 release of SAS 9.4, the following changes and enhancements were made:

- Instead of manually selecting the Hadoop JAR files to the client machine, the SAS Embedded Process determines which version of the JAR files are required and gathers them into a ZIP file for you to copy to the client machine.
- You now have the option whether to automatically start the SAS Embedded Process when the installation is complete.

April 2014 Release of SAS 9.4: Changes and Enhancements

The documentation about the installation and configuration of the SAS Embedded Process was enhanced.

December 2013 Release of SAS 9.4: Changes and Enhancements

In the December 2013 release of SAS 9.4, the following changes and enhancements were made:

- The trace log messages for the SAS Embedded Process are now stored in the MapReduce job log.
- A new option, *hdfsuser*, is available in the *sasep-servers.sh* script. *hdfsuser* specifies the user ID that has Write access to HDFS root directory.
- The Cloudera JAR files for the SAS Embedded Process have been replaced by a set of Apache JAR files. The new JAR files are based on a release of the Apache Hadoop instead of a particular Hadoop distributor.

SAS 9.4: Changes and Enhancements

The Hadoop scripts that install, control, and provide a status of the SAS Embedded Process have changed. There is now just one script, *sasep-servers.sh*, that installs both the SAS Embedded Process and the Hadoop JAR files. Running this script also enables you to start, stop, and provide a status of the SAS Embedded Process.

Netezza Changes

July 2015 Release of SAS 9.4: Changes and Enhancements

The SAS Embedded Process for Netezza has a new cartridge file that creates the NZRC database.

SAS 9.4: Changes and Enhancements

In-database scoring for Netezza has been enhanced by the addition of the SAS Embedded Process. The SAS Embedded Process is a SAS server process that runs within Netezza to read and write data. The SAS Embedded Process can be used with the SAS Scoring Accelerator for Netezza to run scoring models.

SAP HANA Changes

July 2015 Release of SAS 9.4: Changes and Enhancements

If you are using SAP HANA SPS9, the SAS Embedded Process for SAP HANA must be manually started. For previous versions, the SAS Embedded Process was automatically started by the SASAFL plug-in. In addition, a different procedure must be used to deploy the SASAFL plug-in.

Teradata Changes

July 2015 Release of SAS 9.4: Changes and Enhancements

The installation and configuration of the SAS Embedded Process for Teradata has changed. The in-database deployment package is delivered to the client from the SAS Install Depot. The new process has a smaller client footprint and is a faster install process.

SAS Data Loader for Hadoop Changes

SAS Data Loader 2.4: Changes and Enhancements

In the SAS Data Loader for Hadoop 2.4 release, the following changes and enhancements were made:

- If you are operating in a Spark environment, you can deploy the SAS Data Management Accelerator for Spark using the SAS Deployment Manager.
- You can now use the SAS Deployment Manager to deploy SAS Quality Knowledge Base for Contact Information.
- Support is now available for the Hadoop distributions Pivotal HD (including support for Kerberos security) and IBM Big Insights (without support for Kerberos security.) To learn about the supported versions of these distributions, see [SAS Data Loader 2.4 for Hadoop: System Requirements](#).
- A new ZIP file deployment method is now available in addition to the standard deployment method. You must be using a cluster manager for either Cloudera or Hortonworks to take advantage of this feature. To learn about the supported versions of these managers, see [SAS Data Loader 2.4 for Hadoop: System Requirements](#).

Part 1

Introduction

Chapter 1

***Introduction to the Administrator's Guide* 3**

Chapter 1

Introduction to the Administrator's Guide

Overview of SAS In-Database Products	3
What Is Covered in This Document?	4

Overview of SAS In-Database Products

SAS in-database products integrate SAS solutions, SAS analytic processes, and third-party database management systems. Using SAS in-database technology, you can run scoring models, some SAS procedures, DS2 threaded programs, and formatted SQL queries inside the database. When using conventional processing, all rows of data are returned from the database to SAS. When using SAS in-database technology, processing is done inside the database and thus does not require the transfer of data.

To perform in-database processing, the following SAS products require additional installation and configuration:

- SAS/ACCESS Interface to Aster, SAS/ACCESS Interface to DB2, SAS/ACCESS Interface to Greenplum, SAS/ACCESS Interface to Hadoop, SAS/ACCESS Interface to Netezza, SAS/ACCESS Interface to Oracle, SAS/ACCESS Interface to SAP HANA, and SAS/ACCESS Interface to Teradata

The SAS/ACCESS interfaces to the individual databases include components that are required for both format publishing to the database and for running Base SAS procedures inside the database.

- SAS Scoring Accelerator for Aster, SAS Scoring Accelerator for DB2, SAS Scoring Accelerator for Greenplum, SAS Scoring Accelerator for Hadoop, SAS Scoring Accelerator for Netezza, SAS Scoring Accelerator for Oracle, SAS Scoring Accelerator for SAP HANA, and SAS Scoring Accelerator for Teradata
- SAS In-Database Code Accelerator for Greenplum, SAS In-Database Code Accelerator for Hadoop, and SAS In-Database Code Accelerator for Teradata
- SAS Analytics Accelerator for Teradata
- SAS Data Loader for Hadoop
- SAS Data Quality Accelerator for Teradata
- SAS Model Manager In-Database Scoring Scripts

Note: The SAS Scoring Accelerator for SPD Server does not require any additional installation or configuration.

What Is Covered in This Document?

This document provides detailed instructions for installing and configuring the components that are needed for in-database processing using the SAS/ACCESS Interface, the SAS Scoring Accelerator, the SAS Analytics Accelerator, the SAS Data Loader for Hadoop, the SAS Data Quality Accelerator for Teradata, and the In-Database Code Accelerator. These components are contained in a deployment package that is specific for your database.

The name and version of the in-database deployment packages are as follows:

- SAS Embedded Process for Aster 9.4
- SAS Formats Library for DB2 3.1
- SAS Embedded Process for DB2 9.4
- SAS Formats Library for Greenplum 3.1
- SAS Embedded Process for Greenplum 9.4
- SAS Embedded Process for Hadoop 9.4
- SAS Formats Library for Netezza 3.1
- SAS Embedded Process for Oracle 9.4
- SAS Embedded Process for SAP HANA 9.4
- SAS Formats Library for Teradata 3.1
- SAS Embedded Process for Teradata 9.4

If you want to use SAS Model Manager for in-database scoring with DB2, Greenplum, Hadoop, Netezza, or Teradata, additional configuration tasks are needed. This document provides detailed instructions for configuring a database for use with SAS Model Manager.

This document is intended for the system administrator, the database administrator, or both. It is expected that you work closely with the SAS programmers who use these products.

This document is divided by database management systems.

Note: Administrative tasks for the SAS Analytics Accelerator are currently in the *SAS Analytics Accelerator for Teradata: User's Guide*.

Part 2

Administrator's Guide for Hadoop (In-Database Deployment Package)

<i>Chapter 2</i>	
<i>In-Database Deployment Package for Hadoop</i>	<i>7</i>
<i>Chapter 3</i>	
<i>Deploying the In-Database Deployment Package Using the SAS Deployment Manager</i>	<i>11</i>
<i>Chapter 4</i>	
<i>Deploying the In-Database Deployment Package Manually</i>	<i>35</i>
<i>Chapter 5</i>	
<i>Additional Configuration for the SAS Embedded Process</i>	<i>49</i>

Chapter 2

In-Database Deployment Package for Hadoop

Overview of the In-Database Deployment Package for Hadoop	7
Overview of the SAS Embedded Process	8
Prerequisites for Installing the In-Database Deployment Package for Hadoop	8
Backward Compatibility	9
Hadoop Permissions	9
Documentation for Using In-Database Processing in Hadoop	10

Overview of the In-Database Deployment Package for Hadoop

The in-database deployment package for Hadoop must be installed and configured on your Hadoop cluster before you can perform the following tasks:

- Run a scoring model in Hadoop Distributed File System (HDFS) using the SAS Scoring Accelerator for Hadoop.

For more information about using the scoring publishing macros, see the *SAS In-Database Products: User's Guide*.
- Run DATA step scoring programs in Hadoop.

For more information, see the *SAS In-Database Products: User's Guide*.
- Run DS2 threaded programs in Hadoop using the SAS In-Database Code Accelerator for Hadoop.

For more information, see the *SAS In-Database Products: User's Guide*.
- Perform data quality operations in Hadoop, transform data in Hadoop, and extract transformed data out of Hadoop for analysis in SAS using the SAS Data Loader for Hadoop.

For more information, see *SAS Data Loader for Hadoop: User's Guide*.

Note: If you are installing the SAS Data Loader for Hadoop, you must perform additional steps after you install the in-database deployment package for Hadoop. For more information, see Part 3, “Administrator’s Guide for SAS Data Loader for Hadoop”.
- Read and write data to HDFS in parallel for SAS High-Performance Analytics.

Note: For deployments that use SAS High-Performance Deployment of Hadoop for the co-located data provider, and access SASHDAT tables exclusively, SAS/ACCESS and SAS Embedded Process are not needed.

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*.

Overview of the SAS Embedded Process

The in-database deployment package for Hadoop includes the SAS Embedded Process and the SAS Hadoop MapReduce JAR files. The SAS Embedded Process runs within MapReduce to read and write data. The SAS Embedded Process runs on your Hadoop system where the data lives.

By default, the SAS Embedded Process install script (sasep-admin.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 add the SAS Embedded Process to new nodes at a later time, you can run the sasep-admin.sh script with the **-host <hosts>** option.

For distributions that are running MapReduce 1, the SAS Hadoop MapReduce JAR files are required in the **hadoop/lib** directory. For distributions that are running MapReduce 2, the SAS Hadoop MapReduce JAR files are in the **EPInstallDir/SASEPHome/jars/** directory.

Prerequisites for Installing the In-Database Deployment Package for Hadoop

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

- SAS/ACCESS Interface to Hadoop has been configured.

For more information, see *SAS 9.4 Hadoop Configuration Guide for Base SAS and SAS/ACCESS* at [SAS 9.4 Support for Hadoop](#).

- You have working knowledge of the Hadoop vendor 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 HCatalog, HDFS, Hive, MapReduce, Oozie, Sqoop, and YARN services are running on the Hadoop cluster. The SAS Embedded Process does not necessarily use these services. However, other SAS software that relies on the SAS Embedded Process might use these various services. This ensures that the appropriate JAR files are gathered during the configuration.

- The SAS in-database and high-performance analytic products require a specific version of the Hadoop distribution. For more information, see the SAS Foundation system requirements documentation for your operating environment.
- You have sudo access on the NameNode.
- Your HDFS user has Write permission to the root of HDFS.
- The master node needs to connect to the slave nodes using passwordless SSH. For more information, see the Linux manual pages on ssh-keygen and ssh-copy-id.
- You understand and can verify your security setup.

If your cluster is secured with Kerberos, you need the ability to get a Kerberos ticket. You also need to have knowledge of any additional security policies.

- You have permission to restart the Hadoop MapReduce service.

Backward Compatibility

Starting with the July 2015 release of SAS 9.4, the required location of the SAS Hadoop MapReduce JAR files and whether MapReduce service must be restarted during installation of the in-database deployment package for Hadoop depends on what version of the SAS client is being used.

The following table explains the differences.

Table 2.1 *In-database Deployment Package for Hadoop Backward Compatibility*

SAS Client Version	What Version of MapReduce is Running?	Where Do My SAS Hadoop MapReduce JAR Files Need to be Located?	Is Restart of MapReduce Required?*	Is Use of -link or -linklib Required During Installation?**
9.4M3	MapReduce 2	<code>SASEPHOME/jars</code>	No	No
9.4M3	MapReduce 1	<code>hadoop/lib</code>	Yes	No
9.4M2	MapReduce 2	<code>hadoop/lib</code>	Yes	Yes
9.4M2	MapReduce 1	<code>hadoop/lib</code>	Yes	Yes

* See Step 7 in “Installing the SAS Embedded Process” on page 41.

** See “SASEP-ADMIN.SH Script” on page 43.

Hadoop Permissions

The installation of the in-database deployment package for Hadoop involves writing a configuration file to HDFS and deploying files on all data nodes. These tasks require the following permissions:

- Writing the configuration file requires Write permission to HDFS.
- Deploying files across all nodes requires sudo access.

Documentation for Using In-Database Processing in Hadoop

For information about using in-database processing in Hadoop, see the following publications:

- *SAS In-Database Products: User's Guide*
- High-performance procedures in various SAS publications
- *SAS Data Integration Studio: User's Guide*
- SAS/ACCESS Interface to Hadoop and PROC HDMD in *SAS/ACCESS for Relational Databases: Reference*
- *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*
- *SAS Intelligence Platform: Data Administration Guide*
- PROC HADOOP in *SAS/ACCESS for Relational Databases: Reference*
- FILENAME Statement, Hadoop Access Method in *SAS Statements: Reference*
- *SAS Data Loader for Hadoop: User's Guide*

Chapter 3

Deploying the In-Database Deployment Package Using the SAS Deployment Manager

When to Deploy the SAS In-Database Deployment Package Using the SAS Deployment Manager	11
Prerequisites for Using the SAS Deployment Manager to Deploy the In-Database Deployment Package	12
Hadoop Installation and Configuration Steps Using the SAS Deployment Manager	13
Upgrading from or Reinstalling a Previous Version	14
Upgrading from or Reinstalling from SAS 9.3	14
Upgrading from or Reinstalling from SAS 9.4 before the July 2015 Release of SAS 9.4	14
Upgrading from or Reinstalling from the July 2015 Release of SAS 9.4 and Later	16
Using the SAS Deployment Manager to Create the SAS Embedded Process Parcel or Stack	18
Deploying the SAS Embedded Process Parcel on Cloudera	29
Deploying the SAS Embedded Process Stack on Hortonworks	30
Deploying the SAS Embedded Process Stack for the First Time	30
Deploying a New Version of the SAS Embedded Process Stack	32

When to Deploy the SAS In-Database Deployment Package Using the SAS Deployment Manager

You can use the SAS Deployment Manager to deploy the SAS In-Database Deployment Package when the following conditions are met.

- For Cloudera:
 - You are using Cloudera 5.2 or later. For the latest information, see the SAS Foundation system requirements documentation for your operating environment.
 - Cloudera Manager is installed.
 - Your other SAS software, such as Base SAS and SAS/ACCESS Interface to Hadoop, was installed on a UNIX server.
- For Hortonworks:

- You are using Hortonworks 2.1 or later. For the latest information, see the SAS Foundation system requirements documentation for your operating environment.
- You are using Ambari 2.0 or later.
- Your other SAS software, such as Base SAS and SAS/ACCESS Interface to Hadoop, was installed on a UNIX server.

Otherwise, you should deploy the SAS In-Database deployment package manually. For more information, see [Chapter 4, “Deploying the In-Database Deployment Package Manually,” on page 35](#).

CAUTION:

Once you have chosen a deployment method, you should continue to use that same deployment method when upgrading or redeploying the SAS Embedded Process. Otherwise, the SAS Embedded Process can become unusable. For example, if you use the SAS Deployment Manager to deploy the SAS Embedded Process, you should continue to use the SAS Deployment Manager for upgrades or redeployments. You should not use the manual deployment method to upgrade or redeploy. However, if you do need to change deployment methods, you must first uninstall the SAS Embedded Process using the same method that you used to deploy it. Then you can use the other deployment method to install it.

Prerequisites for Using the SAS Deployment Manager to Deploy the In-Database Deployment Package

The following prerequisites are required before you install and configure the in-database deployment package for Hadoop using the SAS Deployment Manager:

- The SSH user must have passwordless sudo access.
- If your cluster is secured with Kerberos, in addition to having a valid ticket on the client, a Kerberos ticket must be valid on node that is running Hive. This is the node that you specify when using the SAS Deployment Manager.
- If you are using Cloudera, the SSH account must have Write permission to these directories:

```
/opt/cloudera
/opt/cloudera/csd
/opt/cloudera/parcels
```
- You cannot customize the install location of the SAS Embedded Process on the cluster. By default, the SAS Deployment Manager deploys the SAS Embedded Process in the `/opt/cloudera/parcels` directory for Cloudera and the `/opt/sasep_stack` directory for Hortonworks.
- If you are using Cloudera, the Java JAR and GZIP commands must be available.
- If you are using Hortonworks 2.2, you must revise properties in the `mapred-site.xml` file. For more information, see [“Additional Configuration for Hortonworks 2.2” on page 52](#).
- If you are using Hortonworks, the `requiretty` option is enabled, and the SAS Embedded Process is installed using the SAS Deployment Manager, the Ambari

server must be restarted after deployment. Otherwise, the SASEP Service does not appear in the Ambari list of services. It is recommended that you disable the requiretty option until the deployment is complete.

- The following information is required:
 - host name and port of the cluster manager
 - credentials (account name and password) for the Hadoop cluster manager
 - Hive node host name
 - Oozie node host name
 - SSH credentials of the administrator who has access to both Hive and Oozie nodes

Hadoop Installation and Configuration Steps Using the SAS Deployment Manager

To install and configure Hadoop using the SAS Deployment Manager, you must follow and complete these steps:

Step	Description	Where to Go for Information
1	Review these topics:	<ul style="list-style-type: none"> • “Prerequisites for Installing the In-Database Deployment Package for Hadoop” on page 8 • “Backward Compatibility” on page 9 • “When to Deploy the SAS In-Database Deployment Package Using the SAS Deployment Manager” on page 11 • “Prerequisites for Using the SAS Deployment Manager to Deploy the In-Database Deployment Package” on page 12
2	If you have not already done so, configure SAS/ACCESS Interface to Hadoop. One of the key tasks in this step is to configure the Hadoop client files.	“Configuring SAS/ACCESS for Hadoop” in SAS Hadoop Configuration Guide for Base SAS and SAS/ACCESS .
3	If you are upgrading from or reinstalling a previous release, follow these instructions:	“Upgrading from or Reinstalling a Previous Version” on page 14
4	Create the SAS Embedded Process parcel (Cloudera) or stack (Hortonworks).	For more information, see “Using the SAS Deployment Manager to Create the SAS Embedded Process Parcel or Stack” on page 18.
5	Deploy the parcel (Cloudera) or stack (Hortonworks) to the nodes on the cluster.	For more information see “Deploying the SAS Embedded Process Parcel on Cloudera” on page 29 or “Deploying the SAS Embedded Process Stack on Hortonworks” on page 30.

Step	Description	Where to Go for Information
6	Review any additional configuration that might be needed depending on your Hadoop distribution.	For more information, see Chapter 5, “Additional Configuration for the SAS Embedded Process,” on page 49.

Upgrading from or Reinstalling a Previous Version

Upgrading from or Reinstalling from SAS 9.3

To upgrade or reinstall from SAS 9.3, follow these steps:

1. Stop the SAS Embedded Process.

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

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

2. Delete the SAS Embedded Process from all nodes.

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

3. Verify that the `sas.hadoop.ep.distribution-name.jar` files 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
```

4. Continue the installation process.

For more information, see [“Using the SAS Deployment Manager to Create the SAS Embedded Process Parcel or Stack”](#) on page 18.

Upgrading from or Reinstalling from SAS 9.4 before the July 2015 Release of SAS 9.4

Note: SAS Data Loader users: If you want to remove either the Quality Knowledge Base (QKB) or the SAS Data Management Accelerator for Spark, you must remove them before removing the SAS Embedded Process. Removing the SAS Embedded Process removes the scripts that are used to remove these products. For more information, see [“Removing the QKB”](#) on page 92 or [“SASDMP_ADMIN.SH Syntax”](#) on page 100.

To upgrade or reinstall from SAS 9.4 before the July 2015 release of SAS 9.4, follow these steps:

1. Stop the SAS Embedded Process.

```
EPInstallDir/SAS/SASTKInDatabaseServerForHadoop/9.*/bin/sasep-servers.sh
```

```
-stop -hostfile host-list-filename | -host <">host-list<">
```

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

For more information, see the SASEP-SERVERS.SH syntax section of the *SAS In-Database Products: Administrator's Guide* that came with your release.

2. Remove the SAS Embedded Process from all nodes.

```
EPInstallDir/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 the SASEP-SERVERS.SH syntax section of the *SAS In-Database Products: Administrator's Guide* that came with your release.

3. Verify that the sas.hadoop.ep.apache*.jar files 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 manually delete them. Open-source utilities are available that can delete these files across multiple nodes.

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

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

5. Manually remove the SAS Embedded Process directories and files on the node from which you ran the script. Open-source utilities are available that can delete these files across multiple nodes.

The sasep-servers.sh -remove script removes the file everywhere except 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.
```

TIP You can use this command to find the location of any instance of the SAS Embedded Process:

```
ps -ef | grep depserver
```

6. Continue the installation process.

For more information, see [“Using the SAS Deployment Manager to Create the SAS Embedded Process Parcel or Stack”](#) on page 18.

Upgrading from or Reinstalling from the July 2015 Release of SAS 9.4 and Later

Overview

The version number of the parcel or stack is calculated by the SAS Deployment Manager with the actual version of the installed product that you selected to deploy. You cannot deploy a parcel or stack that has the same version number as a parcel or stack that was previously deployed. The SAS Deployment Manager assigns a new version number or you can specify your own.

You can either deactivate the existing parcel or stack or remove it before upgrading or reinstalling. If you want to deactivate the existing parcel or stack, use your cluster manager to deactivate the parcel or stack, and continue with the installation instructions in [“Using the SAS Deployment Manager to Create the SAS Embedded Process Parcel or Stack”](#) on page 18.

If you want to remove the existing parcel or stack, see either [“Removing the SAS Embedded Process Parcel Using Cloudera Manager”](#) on page 16 or [“Removing the SAS Embedded Process Stack Using Ambari”](#) on page 17.

Removing the SAS Embedded Process Parcel Using Cloudera Manager

Note: SAS Data Loader users: If you want to remove either the Quality Knowledge Base (QKB) or the SAS Data Management Accelerator for Spark, you must remove them before removing the SAS Embedded Process. Removing the SAS Embedded Process removes the scripts that are used to remove these products. For more information, see [“Removing the QKB”](#) on page 92 or [“SASDMP_ADMIN.SH Syntax”](#) on page 100.

To remove the SAS Embedded Process Parcel using Cloudera Manager, follow these steps:

1. Start Cloudera Manager.
2. Stop the SAS_EP service:
 - a. On the **Home** page, click the down arrow next to **SASEP** service.
 - b. Under **SAS EPActions**, select **Stop**, and click **Stop**.
 - c. Click **Close**.
3. Delete the **SASEP** service from Cloudera Manager:
 - a. On the **Home** page, click the down arrow next to **SASEP** service.
 - b. Click **Delete**.
 - c. Click **Close**.

The **SASEP** service should not appear on the **Home** ⇒ **Status** tab.

4. Deactivate the **SASEP** parcel:
 - a. Navigate to the **Hosts** ⇒ **Parcels** tab.

- b. Select **Actions** ⇒ **Deactivate**.

You are asked to restart the cluster.

- c. Click **Cancel**.

Note: Restarting the cluster is not required. If you want to restart and a rolling restart is available on your cluster, you can choose to perform a rolling restart instead of a full restart. For instructions about performing a rolling restart, see the Cloudera Manager documentation.

- d. Click **OK** to continue the deactivation.

5. Remove the **SASEP** parcel:

- a. Select **Activate** ⇒ **Remove from Hosts**.

- b. Click **OK** to confirm.

6. Delete the **SASEP** parcel.

7. Select **Distribute** ⇒ **Delete**.

8. Click **OK** to confirm.

This step deletes the parcel files from the `/opt/cloudera/parcel` directory.

9. Manually remove the `ep-config.xml` file:

CAUTION:

If you fail to remove the `ep-config.xml` file, the SAS Embedded Process still appears to be available for use. Any software that uses the SAS Embedded Process fails.

- a. Log on to HDFS.

```
sudo su - root
su - hdfs | hdfs-userid
```

Note: If your cluster is secured with Kerberos, the HDFS user must have a valid Kerberos ticket to access HDFS. This can be done with `kinit`.

- b. Navigate to the `/sas/ep/config/` directory on HDFS.

- c. Locate the `ep-config.xml` file.

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

- d. Delete the directory.

```
hadoop fs -rmr /sas/ep/
```

10. Continue the installation process.

For more information, see [“Using the SAS Deployment Manager to Create the SAS Embedded Process Parcel or Stack”](#) on page 18.

Removing the SAS Embedded Process Stack Using Ambari

Note: SAS Data Loader users: If you want to remove either the Quality Knowledge Base (QKB) or the SAS Data Management Accelerator for Spark, you must remove them before removing the SAS Embedded Process. Removing the SAS Embedded Process removes the scripts that are used to remove these products. For more information, see [“Removing the QKB”](#) on page 92 or [“SASDMP_ADMIN.SH Syntax”](#) on page 100.

Note: You need root or passwordless sudo access to remove the stack.

To remove the SAS Embedded Process stack using Ambari, follow these steps:

1. Navigate to the **`SASHOME/SASHadoopConfigurationLibraries/2.4/Config/Deployment/stacks/sasep`** directory on the client where the SAS software is downloaded and installed.

```
cd SASHOME/SASHadoopConfigurationLibraries/2.4/Config/Deployment/stacks/sasep
```

The `delete_stack.sh` file should be in this directory.

2. Copy the `delete_stack.sh` file to a temporary directory where the cluster manager server is located. Here is an example using secure copy.

```
scp delete_stack.sh user@cluster-manager-host:/mydir
```

3. Use this command to run the delete script.

```
./delete_stack.sh <Ambari-Admin-User-Name>
```

4. Enter the Ambari administrator password at the prompt.

The following message appears.

```
Select one of the options from the below
enter 1 to remove SASEP config file only
enter 2 to remove specific version of SASEP
enter 3 to remove all versions of SASEP
```

5. Choose one of the three options:

- Enter 1 to remove only the `ep-config.xml` file.
- Enter 2 to remove a specific version of the SAS Embedded Process. If you enter 2, a list of all the versions that are available for removal appears. You can then enter any of the versions to be deleted.
- Enter 3 to remove all versions of the SAS Embedded Process.

To complete the removal of the **SASEP SERVICE**, you are prompted to restart the Ambari server.

6. Enter `y` to restart the Ambari server. The **SASEP SERVICE** should not appear.
7. Continue the installation process.

For more information, see [“Using the SAS Deployment Manager to Create the SAS Embedded Process Parcel or Stack”](#) on page 18.

Using the SAS Deployment Manager to Create the SAS Embedded Process Parcel or Stack

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

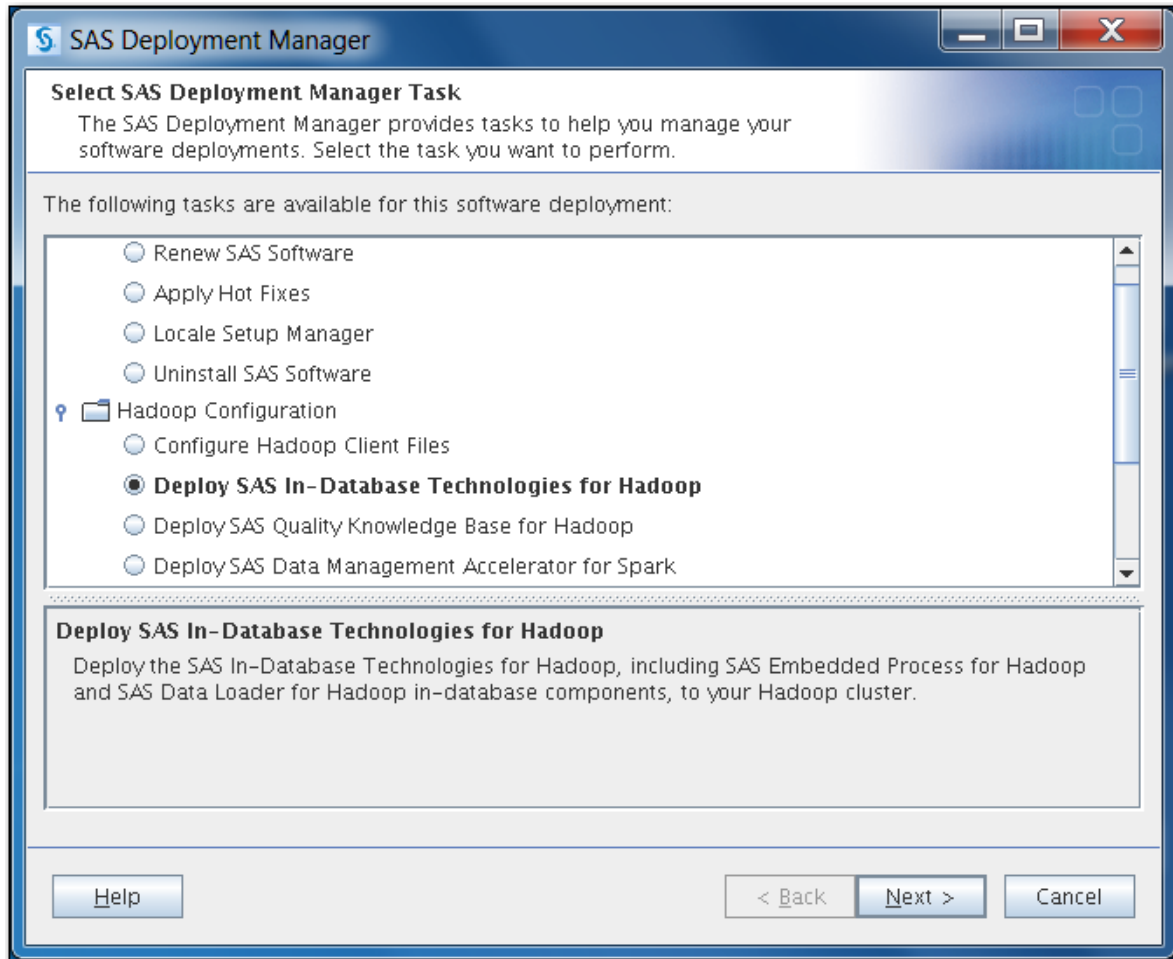
1. Start the SAS Deployment Manager .

```
cd /SASHOME/SASDeploymentManager/9.4
./sasdm.sh
```

The **Choose Language** page opens.

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

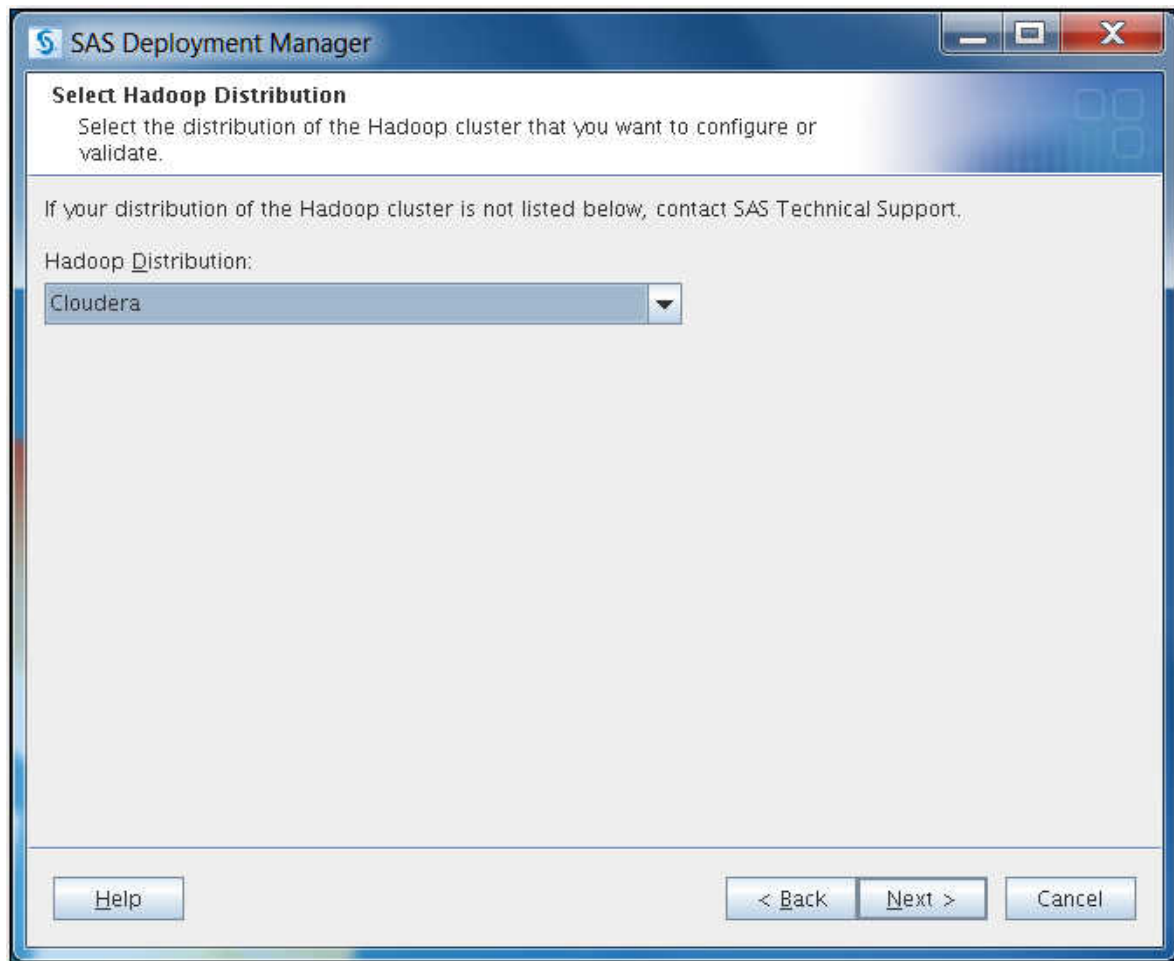
Click **OK**. The **Select SAS Deployment Manager Task** page opens.



3. Under Hadoop Configuration, select **Deploy SAS In-Database Technologies for Hadoop**.

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

Note: If you have licensed and downloaded SAS Data Loader for Hadoop, the SAS Data Loader for Hadoop data quality components are silently deployed at the same time as the SAS Embedded Process for Hadoop.



4. 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**. The **Hadoop Cluster Manager Information** page opens.

The screenshot shows a window titled "SAS Deployment Manager" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there is a section titled "Hadoop Cluster Manager Information" with a subtitle "Specify the host name and port number of your Hadoop cluster manager." Below this, there are two input fields: "Host Name:" and "Port Number:". The "Port Number:" field contains the value "7180". At the bottom of the window, there are three buttons: "Help", "< Back", and "Next >", followed by a "Cancel" button.

5. 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.

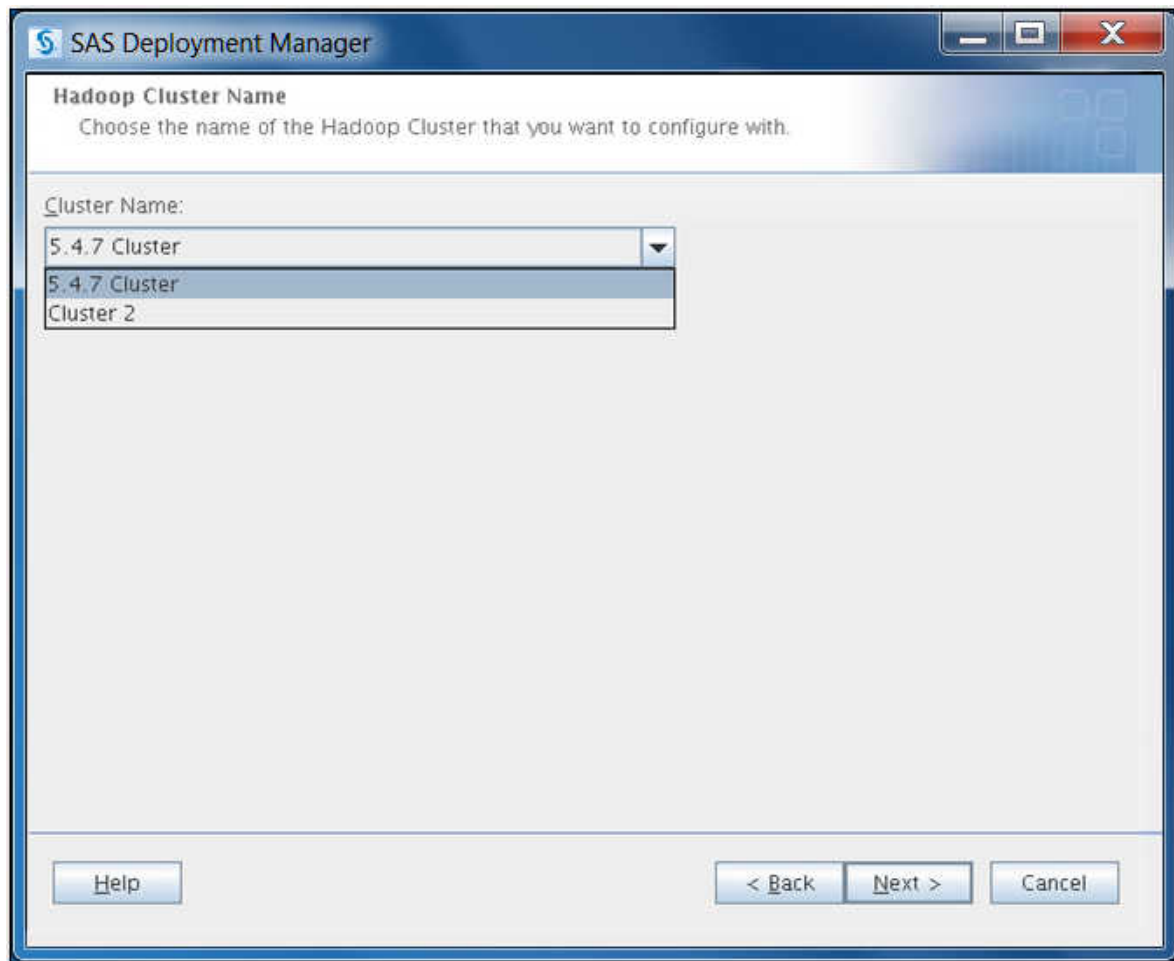
The screenshot shows a window titled "SAS Deployment Manager" with a subtitle "Hadoop Cluster Manager Credentials". The subtitle is followed by the instruction "Specify the credentials of the administrator account for the Hadoop cluster manager." Below this, there are three text input fields labeled "Administrator Account Name:", "Password:", and "Confirm Password:". At the bottom of the window, there are four buttons: "Help", "< Back", "Next >", and "Cancel".

6. 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**.

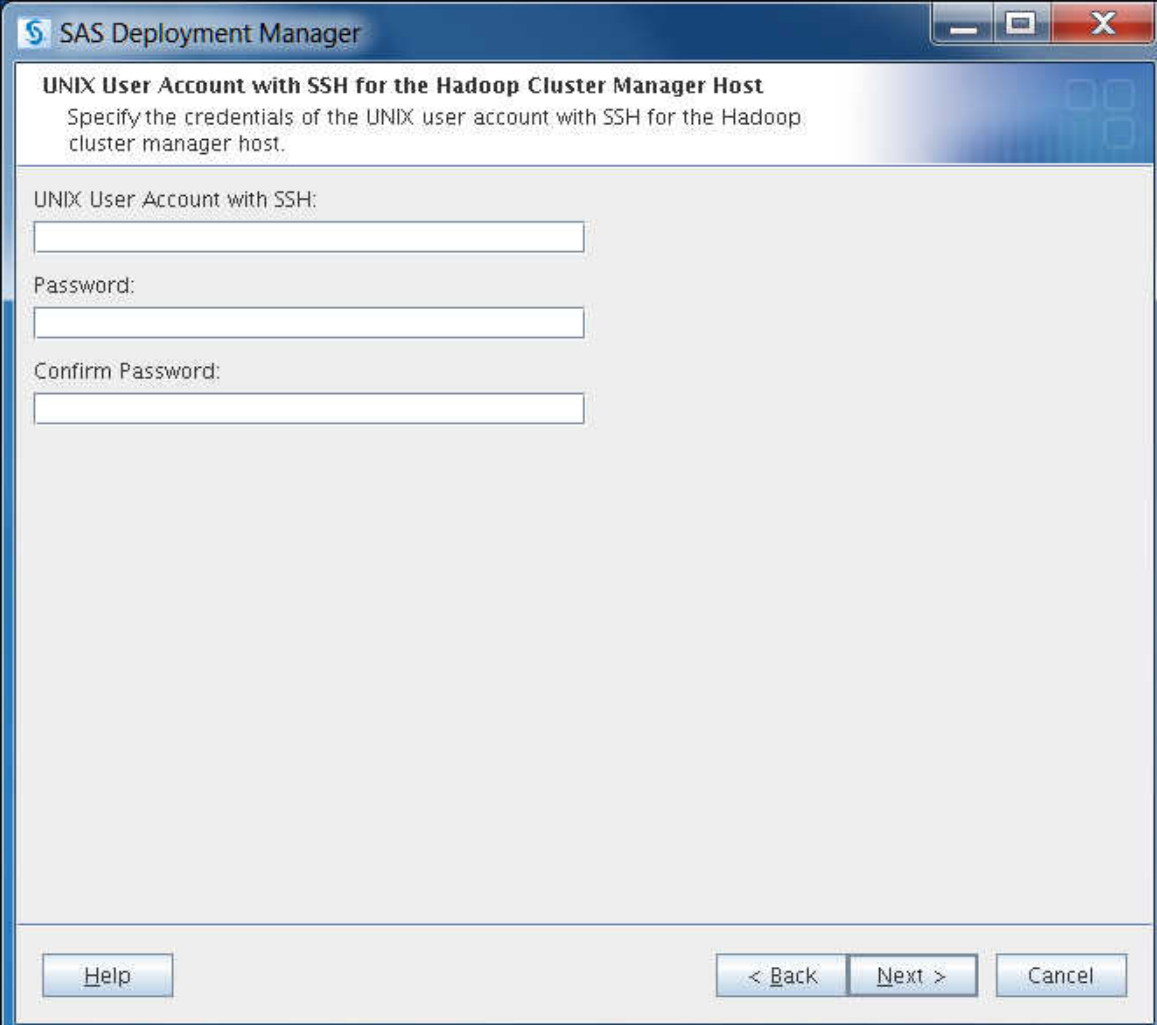
If you are using Cloudera Manager and multiple Hadoop clusters are being managed by the same cluster manager, the **Hadoop Cluster Name** page opens. Continue with [Step 7 on page 23](#).



Otherwise, the **UNIX User Account with SSH for the Hadoop Cluster Manager Host** page opens. Skip to [Step 8 on page 24](#).

7. Select the cluster from the drop-down list.

Click **Next**. The **UNIX User Account with SSH for the Hadoop Cluster Manager Host** page opens.

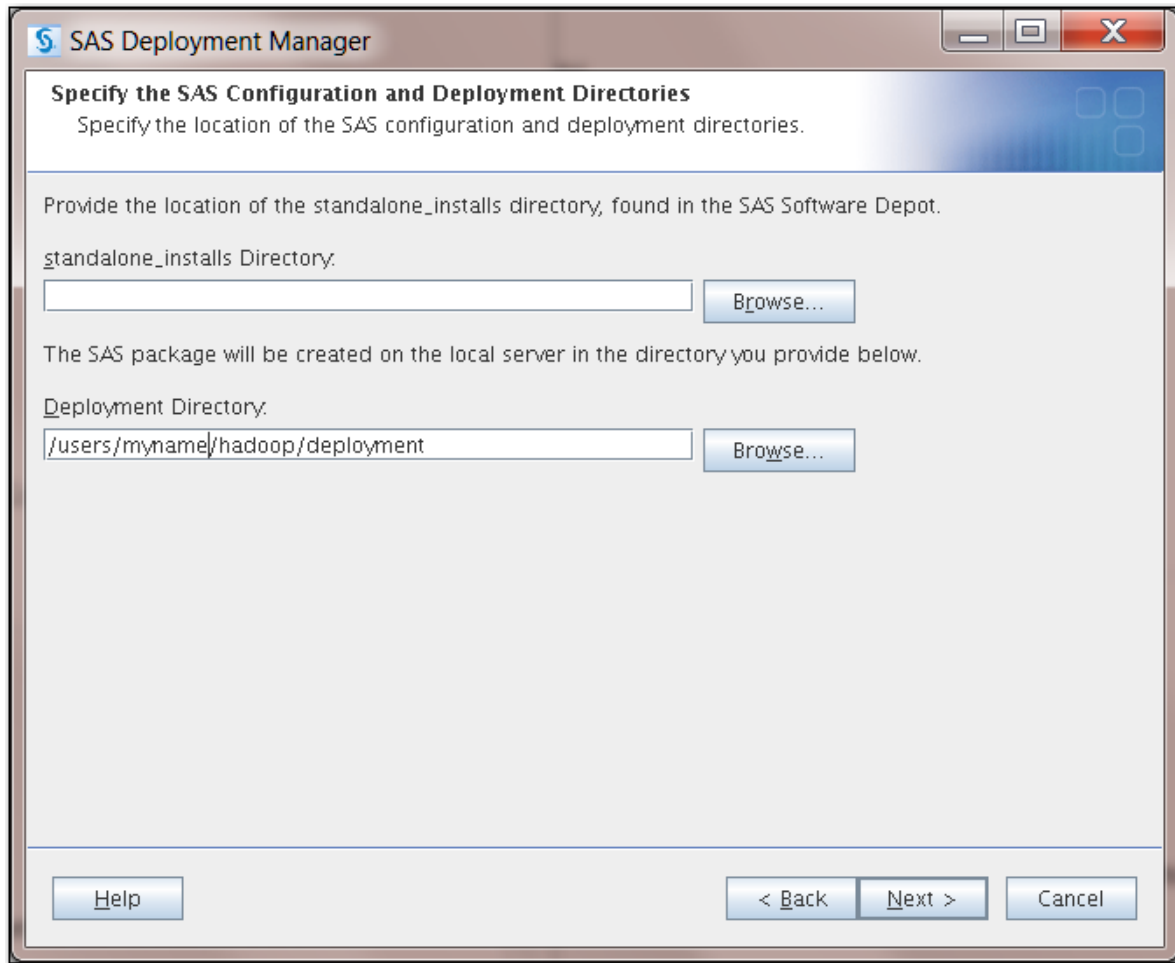


The screenshot shows a window titled "SAS Deployment Manager". Inside, the main heading is "UNIX User Account with SSH for the Hadoop Cluster Manager Host". Below this heading is a sub-instruction: "Specify the credentials of the UNIX user account with SSH for the Hadoop cluster manager host." There are three text input fields: the first is labeled "UNIX User Account with SSH:", the second is labeled "Password:", and the third is labeled "Confirm Password:". At the bottom of the window, there are four buttons: "Help", "< Back", "Next >", and "Cancel".

8. Enter the root SSH account that has access to the cluster manager or enter a non-root SSH account if that account can execute sudo without entering a password.

Note: For Cloudera, the SSH account must have Write permission to the `/opt/cloudera` directory. Otherwise, the deployment completes with errors.

Click **Next**. The **Specify the SAS Configuration and Deployment Directories** page opens.



9. Enter the location of the SAS configuration and deployment directories:
 - a. Enter (or navigate to) the location of the `/standalone_installs` directory. This directory was created when your SAS Software Depot was created by the SAS Download Manager.

CAUTION:

After installation, do not delete your SAS Software Depot `standalone_installs` directory or any of its subdirectories. If hot fixes are made available for your software, they are moved to a subdirectory of the `/standalone_installs/SAS_Core_Embedded_Process_Package_for_Hadoop/` directory. The SAS Deployment Manager requires that both the initial installation files and the hot fix file exist in a subdirectory of the original SAS Software Depot `/standalone_installs/SAS_Core_Embedded_Process_Package_for_Hadoop/` directory.

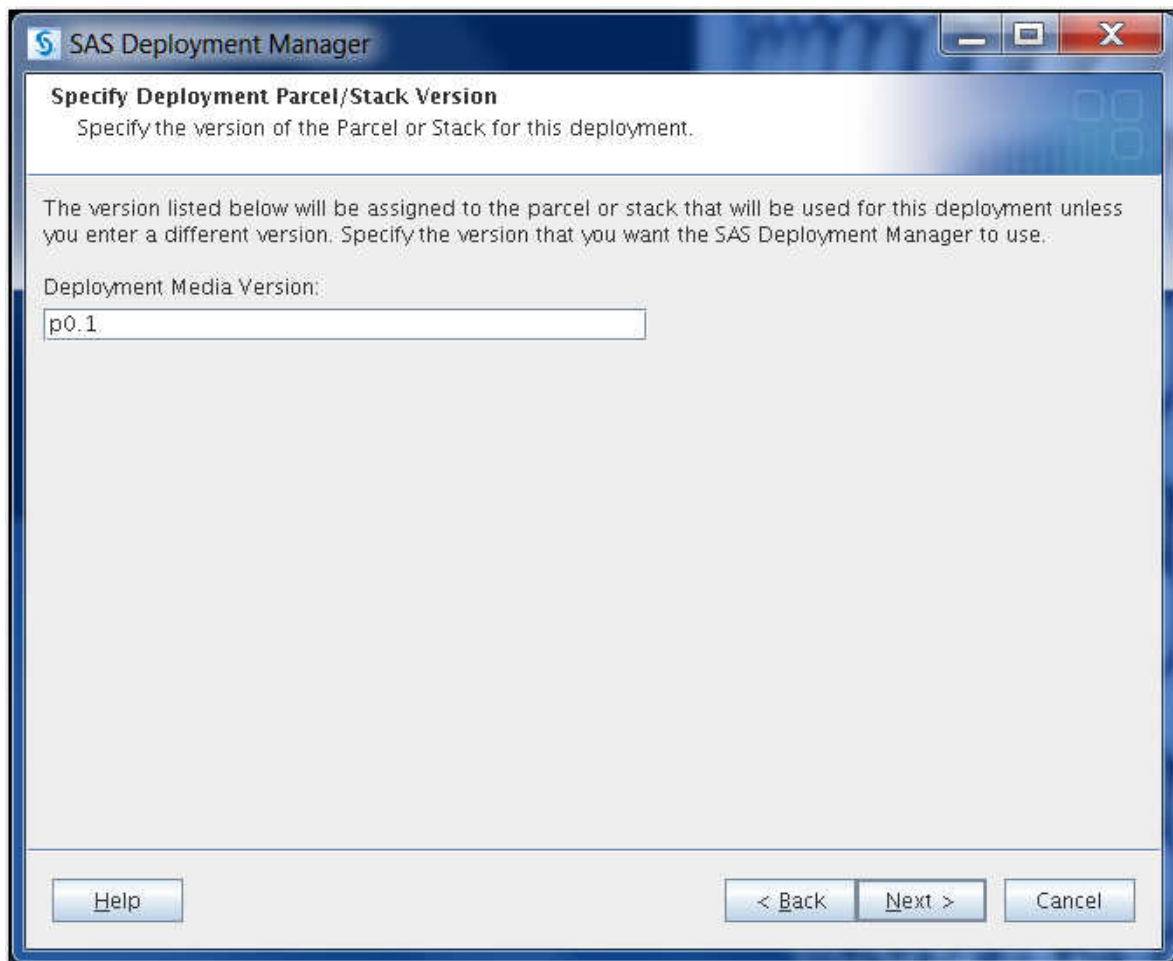
Note: If you have licensed and downloaded SAS Data Loader for Hadoop, the SAS Data Loader for Hadoop data quality components are located in the same directory as the SAS Embedded Process files. The SAS Data Loader for Hadoop files are silently deployed at the same time as the SAS Embedded Process for Hadoop.

- b. Enter (or navigate to) a working directory on the local server where the package or stack is placed. The working directory is removed when the deployment is complete.

Click **Next**. A list of SAS products to deploy is displayed.



Click **OK**. The **Specify Deployment Parcel/Stack Version** page opens.



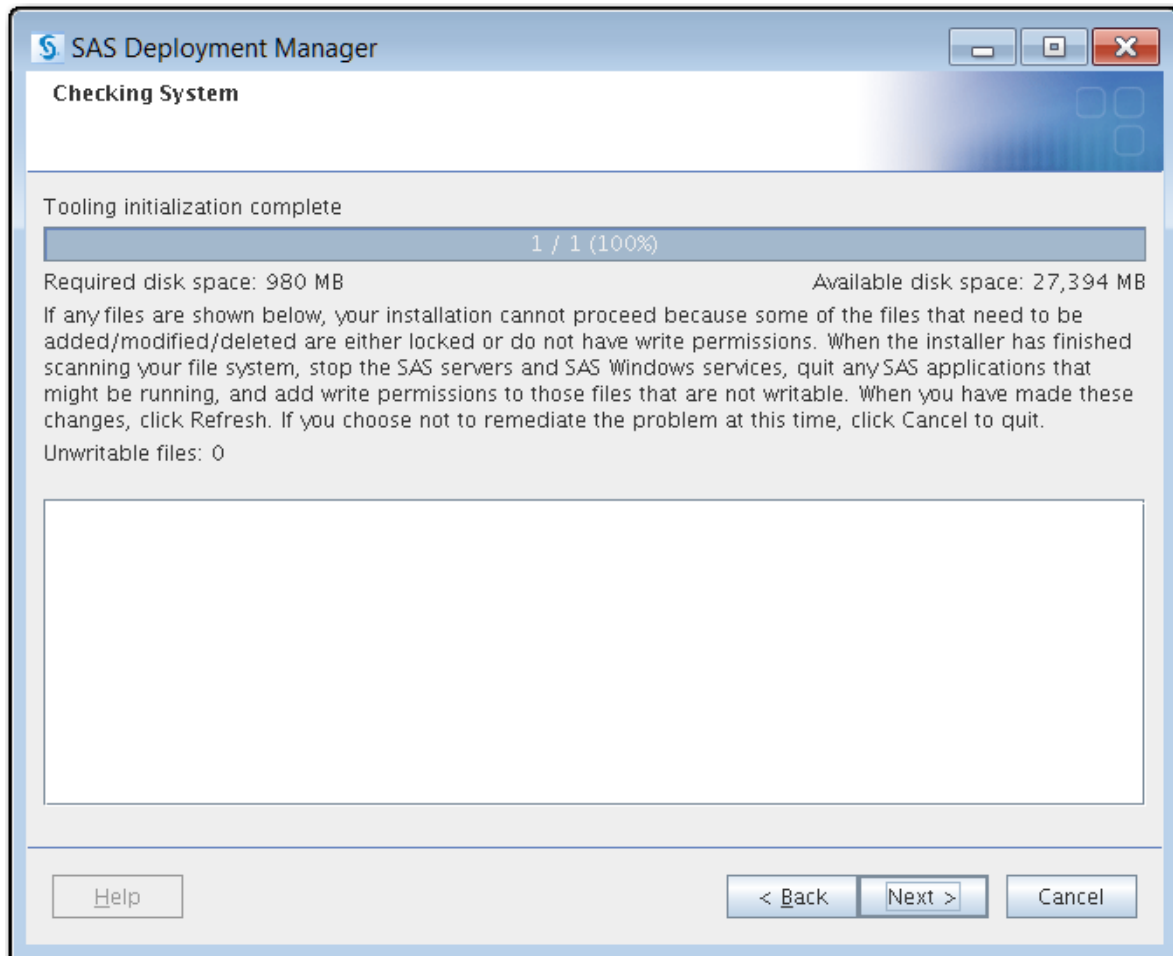
10. The version listed is assigned to the media that is used for deployment unless you enter a different version.

The version number is calculated by the SAS Deployment Manager based on the installed product that you selected to deploy.

Note: You cannot deploy media that has the same version number as media that was previously deployed.

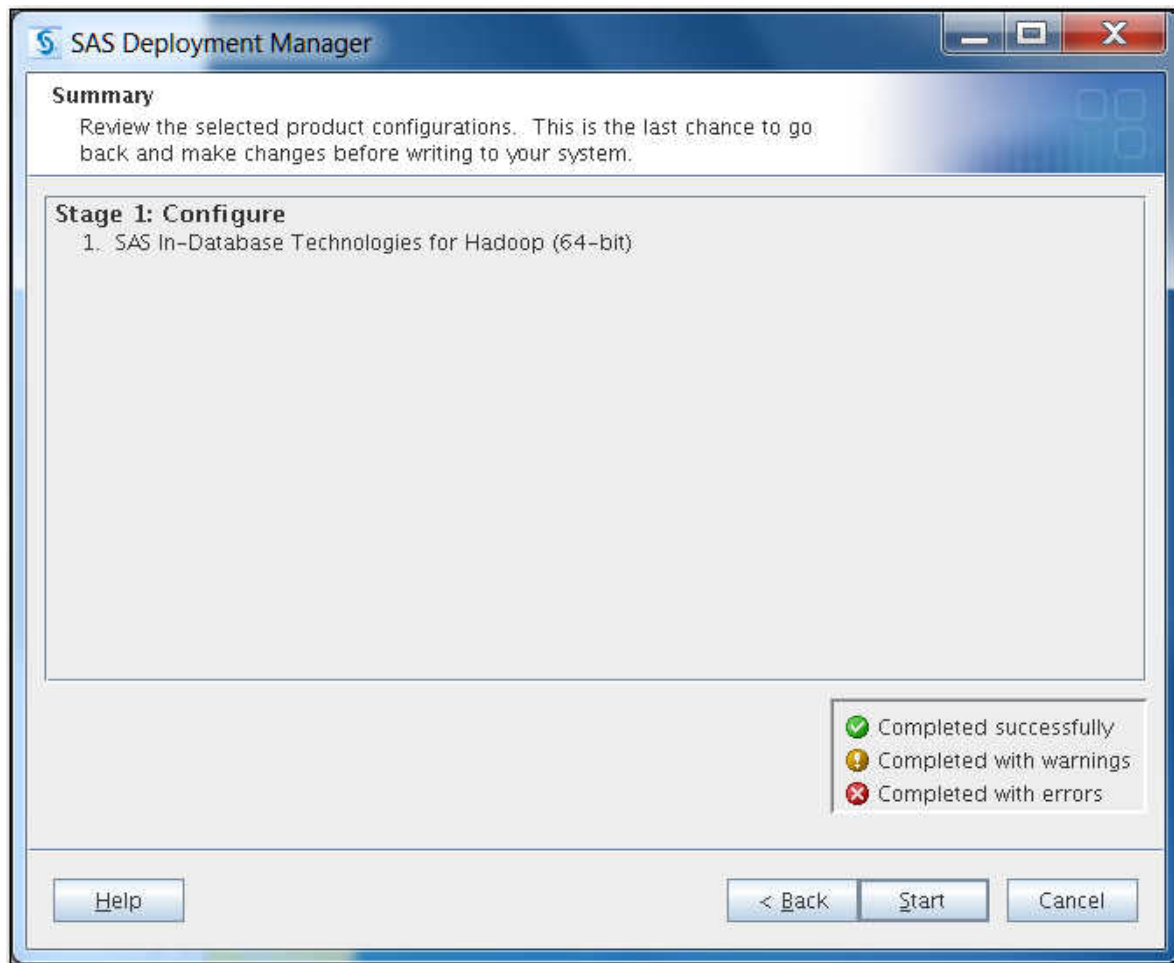
Click **Next**. The **Checking System** page opens, and a check for locked files and Write permissions is performed.

Note: If you are using Hortonworks and the requiretty option is enabled, you receive a warning that you must restart the Ambari server when you deploy the stack.



11. 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.



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

Note: It takes time to complete the configuration. If your cluster is secured with Kerberos, it could take longer.

Note: The product that appears on this page is the SAS product that is associated with the in-database deployment package for Hadoop. This package includes the SAS Embedded Process and possibly other components. Note that a separate license might be required to use the SAS Embedded Process.

If the configuration is successful, the page title changes to **Deployment Complete** and a green check mark is displayed beside SAS In-Database Technologies for Hadoop (64-bit).

Note: Part of the configuration process runs SAS code to validate the environment. A green check mark indicates that the SAS Deployment Manager was able to create the SAS Embedded Process parcel or stack and then verify that the parcel or stack was copied to the cluster manager node.

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

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

A log file is written to the `%HOME/.SASAppData/SASDeploymentWizard` directory on the client machine.

14. Continue the installation process.

For more information, see [“Deploying the SAS Embedded Process Parcel on Cloudera” on page 29](#) or [“Deploying the SAS Embedded Process Stack on Hortonworks” on page 30](#).

Deploying the SAS Embedded Process Parcel on Cloudera

After you run the SAS Deployment Manager to create the SAS Embedded Process parcel, you must distribute and activate the parcel on the cluster. Follow these steps:

Note: More than one SAS Embedded Process parcel can be deployed on your cluster, but only one parcel can be activated at one time. Before activating a new parcel, deactivate the old one.

1. Log on to Cloudera Manager.
2. In Cloudera Manager, choose **Hosts** ⇒ **Parcels**.

The SASEP parcel is located under your cluster. The parcel name is the one from Step 6 in [“Using the SAS Deployment Manager to Create the SAS Embedded Process Parcel or Stack” on page 18](#). An example name is 9.43.p0.1.

3. Click **Distribute** to copy the parcel to all nodes and the SASEPHome directory is created.

Note: If you have licensed and downloaded SAS Data Loader for Hadoop, some SAS Data Loader for Hadoop data quality components are silently deployed at the same time as the SAS Embedded Process for Hadoop. Other configuration is required as noted in step 9.

You can log on to the node and show the contents in the `/opt/cloudera/parcel` directory.

4. Click **Activate**.

This step creates a symbolic link to the SAS Hadoop JAR files.

You are prompted to either restart the cluster or close the window.

5. Click **Close the Window**.

CAUTION:

Do not restart the cluster. Do not click Restart.

6. Use the **Add Service Wizard** page to add **SASEP** as a service on any node where HDFS is a client:
 - a. Navigate to the Cloudera Manager Home.
 - b. Select **Actions** ⇒ **Add a Service**.
 - c. Select the **SASEP** service and click **Continue**.
 - d. Select the dependencies for the SAS Embedded Process service in the **Add Service Wizard** ⇒ **Select the set of dependencies for your new service** page. Click **Continue**.
 - e. Choose a location for the SAS Embedded Process ep-config.xml file in the **Add Service Wizard** ⇒ **Customize Role Assignments** page. Click **Select the set of dependencies for your new service** page. Click **OK**.

The ep-config.xml file is created and added to the HDFS `/sas/ep/config` directory. This task is done in the host that you select.

Note: If your cluster is secured with Kerberos, the host that you select must have a valid ticket for the HDFS user.

- f. After the SAS Embedded Process ep-config.xml file is created, Cloudera Manager starts the SAS Embedded Process service. This step is not required. MapReduce is the only service that is required for the SAS Embedded Process. You must stop the SAS Embedded Process service immediately when the task that adds the SAS Embedded Process is finished. The SAS Embedded Process service no longer needs to be stopped or started.
7. Verify that the ep-config.xml file exists in the `/sas/ep/config` directory of the host that you selected in step 6e.
8. Review any additional configuration that might be needed depending on your Hadoop distribution.

For more information, see [Chapter 5, “Additional Configuration for the SAS Embedded Process,” on page 49](#).

9. Validate the deployment of the SAS Embedded Process by running a program that uses the SAS Embedded Process and the MapReduce service. An example is a scoring program.
10. If you have licensed and downloaded any of the following SAS software, additional configuration is required:
 - SAS Data Loader for Hadoop

For more information, see Part 3, “Administrator’s Guide for SAS Data Loader for Hadoop”.
 - SAS High-Performance Analytics

For more information, see *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*.

Deploying the SAS Embedded Process Stack on Hortonworks

Deploying the SAS Embedded Process Stack for the First Time

After you run the SAS Deployment Manager to create the SAS Embedded Process stack, you must deploy the stack on the cluster. Follow these steps:

Note: If the SAS Embedded Process stack already exists on your cluster, follow the instructions in [“Deploying a New Version of the SAS Embedded Process Stack” on page 32](#).

1. Start the Ambari server and log on.
2. If the requiretty option was enabled when you deployed the SAS Embedded Process, you must restart the Ambari server at this time. Otherwise, skip to Step 3.
 - a. Log on to the cluster.

```
sudo - su
```

- b. Restart the Ambari server.

```
ambari-server restart
```

- c. Start the Ambari server and log on.

3. Click **Actions** and choose + **Add Service**.

The **Add Service Wizard** page appears.

4. Select **Choose Services**.

The **Choose Services** panel appears.

5. In the **Choose Services** panel, select **SASEP SERVICE**. Click **Next**.

The **Assign Slaves and Clients** panel appears.

Note: You should always select NAMENODE as one of the clients and NAMENODE should have these two client components installed: HDFS_CLIENT and HCAT_CLIENT.

Note: If you have licensed and downloaded SAS Data Loader for Hadoop, some SAS Data Loader for Hadoop data quality components are silently deployed at the same time as the SAS Embedded Process for Hadoop. Other configuration is required as noted in step 11.

6. In the **Assign Slaves and Clients** panel, select items under **Client** where you want the stack to be deployed.

The **Customize Services** panel appears.

The SASEP stack is listed under **activated_version**. The stack name is the one from Step 6 in [“Using the SAS Deployment Manager to Create the SAS Embedded Process Parcel or Stack”](#) on page 18. An example name is 9.43.s0.1.

7. Do not change any settings on the **Customize Services** panel. Click **Next**.

If your cluster is secured with Kerberos, the **Configure Identities** panel appears. Enter your Kerberos credentials in the **admin_principal** and **admin_password** text boxes.

The **Review** panel appears.

8. Review the information about the panel. If everything is correct, click **Deploy**.

The **Install, Start, and Test** panel appears. When the SAS Embedded Process stack is installed on all nodes, click **Next**.

The **Summary** panel appears.

9. Click **Complete**. The SAS Embedded Process stack is now installed on all nodes of the cluster.

You should now be able to see **SASEP SERVICE** on the Ambari dashboard.

10. Verify that the ep-config.xml file exists in the **/sas/ep/config** directory.

11. Review any additional configuration that might be needed depending on your Hadoop distribution.

For more information, see [Chapter 5, “Additional Configuration for the SAS Embedded Process,”](#) on page 49.

12. Validate the deployment of the SAS Embedded Process by running a program that uses the SAS Embedded Process and the MapReduce service. An example is a scoring program.

13. If you have licensed and downloaded any of the following SAS software, additional configuration is required:

- SAS Data Loader for Hadoop

For more information, see Part 3, “Administrator’s Guide for SAS Data Loader for Hadoop”.

- SAS High-Performance Analytics

For more information, see *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*.

Deploying a New Version of the SAS Embedded Process Stack

More than one SAS Embedded Process stack can be deployed on your cluster, but only one stack can be activated at one time. After you run the SAS Deployment Manager to create the SAS Embedded Process stack, follow these steps to deploy an additional SAS Embedded Process stack when one already exists on your cluster.

1. Restart the Ambari server and log on to the Ambari manager.
2. Select **SASEP SERVICE**.

In the **Services** panel, a restart symbol appears next to **SASEP SERVICE**. The **Configs** tab indicates that a restart is required.

3. Click **Restart**.
4. Click **Restart All**.

After the service is restarted, the previous version of the SAS Embedded Process still appears in the **activated_version** text box on the **Configs** tab. All deployed versions of the SAS Embedded Process stack should appear in the **sasep_allversions** text box.

5. Refresh the browser.

The new version of the SAS Embedded Process should now appear as the **activated_version** text box on the **Configs** tab.

If, at any time, you want to activate another version of the SAS Embedded Process stack, follow these steps:

1. Enter the version number in the **activated_version** text box on the **Configs** tab.
2. Click **Save**.
3. Add a note describing your action (for example, “Changed from version 9.43.s01.1 to 9.43.s01.2”), and click **Next**.
4. Click **Restart**.
5. Click **Restart All**.
6. Refresh Ambari.

The new service is activated.

7. Review any additional configuration that might be needed depending on your Hadoop distribution.

For more information, see [Chapter 5, “Additional Configuration for the SAS Embedded Process,”](#) on page 49.

8. Validate the deployment of the SAS Embedded Process by running a program that uses the SAS Embedded Process and the MapReduce service. An example is a scoring program.
9. If you have licensed and downloaded any of the following SAS software, additional configuration is required:
 - SAS Data Loader for Hadoop
For more information, see Part 3, “Administrator’s Guide for SAS Data Loader for Hadoop”.
 - SAS High-Performance Analytics
For more information, see *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*.

Chapter 4

Deploying the In-Database Deployment Package Manually

When to Deploy the SAS In-Database Deployment Package Manually	35
Hadoop Manual Installation and Configuration Steps	36
Upgrading from or Reinstalling a Previous Version	37
Upgrading from or Reinstalling from SAS 9.3	37
Upgrading from or Reinstalling from SAS 9.4 before the July 2015 Release of SAS 9.4	37
Upgrading from or Reinstalling from the July 2015 Release of SAS 9.4 or Later	39
Copying the SAS Embedded Process Install Script to the Hadoop Cluster	40
Creating the SAS Embedded Process Directory	40
Copying the SAS Embedded Process Install Script	40
Installing the SAS Embedded Process	41
SASEP-ADMIN.SH Script	43
Overview of the SASEP-ADMIN.SH Script	43
SASEP-ADMIN.SH Syntax	44

When to Deploy the SAS In-Database Deployment Package Manually

You should deploy the SAS In-Database deployment package manually in the following instances:

- Your Hadoop distribution is IBM BigInsights, Pivotal HD, or MapR.
- Your Hadoop distribution is Cloudera and any of the following is true:
 - Cloudera Manager is not installed.
 - You are not using Cloudera 5.2 or later.
 - Your other SAS software, such as Base SAS and SAS/ACCESS Interface to Hadoop, was installed on Windows. The SAS Deployment Manager cannot be used on a Windows client to install the SAS In-Database deployment package.
- Your Hadoop distribution is Hortonworks and any of the following are true:
 - Ambari is not installed or you are using Ambari 1.7.
 - You are not using Hortonworks 2.1 or later.

- Your other SAS software, such as Base SAS and SAS/ACCESS Interface to Hadoop, was installed on Windows. The SAS Deployment Manager cannot be used on a Windows client to install the SAS in-database deployment package.

For more information, see [Chapter 3, “Deploying the In-Database Deployment Package Using the SAS Deployment Manager,”](#) on page 11.

CAUTION:

Once you have chosen a deployment method, you should continue to use that same deployment method when upgrading or redeploying the SAS Embedded Process. Otherwise, the SAS Embedded Process can become unusable. For example, if you use the SAS Deployment Manager to deploy the SAS Embedded Process, you should continue to use the SAS Deployment Manager for upgrades or redeployments. You should not use the manual deployment method to upgrade or redeploy. However, if you do need to change deployment methods, you must first uninstall the SAS Embedded Process using the same method that you used to deploy it. Then you can use the other deployment method to install it.

Hadoop Manual Installation and Configuration Steps

To install and configure Hadoop manually, you must follow and complete these steps:

Step	Description	Where to Go for Information
1	Review these topics:	<ul style="list-style-type: none"> • “Prerequisites for Installing the In-Database Deployment Package for Hadoop” on page 8 • “Backward Compatibility” on page 9 • “When to Deploy the SAS In-Database Deployment Package Manually” on page 35
2	If you have not already done so, configure SAS/ACCESS Interface to Hadoop. One of the key tasks in this step is to configure the Hadoop client files.	“Configuring SAS/ACCESS for Hadoop” in SAS Hadoop Configuration Guide for Base SAS and SAS/ACCESS .
3	If you are upgrading from or reinstalling a previous release, follow these instructions:	“Upgrading from or Reinstalling a Previous Version” on page 37
4	Copy the in-database deployment package install script (sepcorehadp) to the Hadoop master node (the NameNode).	“Copying the SAS Embedded Process Install Script to the Hadoop Cluster” on page 40 <i>Note:</i> In the July 2015 release of SAS 9.4, the in-database deployment package install script changed name from tkindbsrv to sepcorehadp. The SAS Embedded Process and the SAS Hadoop MapReduce JAR files are now included in the same script. The SAS Embedded Process is the core technology of the in-database deployment package.
5	Install the SAS Embedded Process.	“Installing the SAS Embedded Process” on page 41

Step	Description	Where to Go for Information
6	Review any additional configuration that might be needed depending on your Hadoop distribution.	Chapter 5, “Additional Configuration for the SAS Embedded Process,” on page 49
7 (Optional)	If you are installing the SAS Data Loader for Hadoop, you must perform additional steps after you install the SAS Embedded Process.	Part 3, “Administrator’s Guide for SAS Data Loader for Hadoop”
8 (Optional)	If you are installing the SAS High-Performance Analytics environment, you must perform additional steps after you install the SAS Embedded Process.	<i>SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide</i>

Upgrading from or Reinstalling a Previous Version

Upgrading from or Reinstalling from SAS 9.3

To upgrade or reinstall from SAS 9.3, follow these steps:

1. Stop the Hadoop SAS Embedded Process using the 9.3 `sasep-stop.all.sh` script.

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

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

2. Delete the Hadoop SAS Embedded Process from all nodes.

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

3. Verify that the `sas.hadoop.ep.distribution-name.jar` files 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
```

4. Restart the MapReduce service to clear the SAS Hadoop MapReduce JAR files from the cache.
5. Continue the installation process.

For more information, see [“Copying the SAS Embedded Process Install Script to the Hadoop Cluster” on page 40](#).

Upgrading from or Reinstalling from SAS 9.4 before the July 2015 Release of SAS 9.4

CAUTION:

If you are using SAS Data Loader, you should remove the QKB and the SAS Data Management Accelerator for Spark from the Hadoop nodes before removing the SAS Embedded Process. For more information, see “[Removing the QKB](#)” on page 92 or “[SASDMP_ADMIN.SH Syntax](#)” on page 100.

To upgrade or reinstall from a version of SAS 9.4 before the July 2015 release of SAS 9.4, follow these steps:

1. Stop the Hadoop SAS Embedded Process using the 9.4 `sasep-servers.sh -stop` script.

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

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

For more information, see the `SASEP-SERVERS.SH` syntax section of the *SAS In-Database Products: Administrator's Guide* that came with your release.

2. Remove the SAS Embedded Process from all nodes.

```
EPInstallDir/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 the `SASEP-SERVERS.SH` syntax section of the *SAS In-Database Products: Administrator's Guide* that came with your release.

3. Restart the MapReduce service to clear the SAS Hadoop MapReduce JAR files from the cache.
4. Verify that all files associated with the SAS Embedded Process have been removed.

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

- a. Verify that the `sas.hadoop.ep.apache*.jar` files 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
```

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

- c. Manually remove the SAS Embedded Process directories and files on the master node (*EPInstallDir*) from which you ran the script.

The `sasep-servers.sh -remove` script removes the file everywhere except 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.
```

TIP You can use this command to find the location of any instance of the SAS Embedded Process:

```
ps -ef | grep depserver
```

5. Continue the installation process.

For more information, see [“Copying the SAS Embedded Process Install Script to the Hadoop Cluster” on page 40](#).

Upgrading from or Reinstalling from the July 2015 Release of SAS 9.4 or Later

CAUTION:

If you are using SAS Data Loader, you should remove the QKB from the Hadoop nodes before removing the SAS Embedded Process. The QKB is removed by running the QKBPUK script. For more information, see [“Removing the QKB” on page 92](#).

To upgrade or reinstall from the July 2015 release of SAS 9.4 or later, follow these steps:

1. Locate the sasep-admin.sh file.

This file is in the **EPInstallDir/sasexe/SASEPHome/bin** directory. **EPInstallDir** is where you installed the SAS Embedded Process.

One way to find the **EPInstallDir** directory is to look at the sas.ep.classpath property in the ep-config.xml file. The ep-config.xml file is located on HDFS in the **/sas/ep/config/** directory.

- a. Enter this Hadoop command to read the ep-config.xml file on HDFS.

```
hadoop fs -cat /sas/ep/config/ep-config.xml
```

- b. Search for the sas.ep.classpath property.
- c. Copy the directory path.

The path should be **EPInstallDir/sasexe/SASEPHome/** where **EPInstallDir** is where you installed the SAS Embedded Process.

- d. Navigate to the **EPInstallDir/sasexe/SASEPHome/bin** directory.

2. Run sasep-admin.sh -remove script.

This script removes the SAS Embedded Process from the data nodes.

3. Run this command to remove the SASEPHome directories from the master node.

```
rm -rf SASEPHome
```

4. Continue the installation process.

For more information, see [“Copying the SAS Embedded Process Install Script to the Hadoop Cluster” on page 40](#).

Copying the SAS Embedded Process Install Script to the Hadoop Cluster

Creating the SAS Embedded Process Directory

Create a new directory on the Hadoop master node 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. We do not recommend that you use existing system directories such as `/opt` or `/usr`. This new directory is referred to as *EPInstallDir* throughout this section.

Copying the SAS Embedded Process Install Script

The SAS Embedded Process install script is contained in a self-extracting archive file named `sepcorehadp-9.43000-1.sh`. This file is contained in a ZIP file that is put in a directory in your SAS Software Depot.

Using a method of your choice, transfer the ZIP file to the *EPInstallDir* on your Hadoop master node.

1. Navigate to the `YourSASDepot/standalone_installs` directory.

This directory was created when your SAS Software Depot was created by the SAS Download Manager.

2. Locate the `en_sasexe.zip` file. The `en_sasexe.zip` file is located in the following directory: `YourSASDepot/standalone_installs/SAS_Core_Embedded_Process_Package_for_Hadoop/9_43/Hadoop_on_Linux_x64/`.

The `sepcorehadp-9.43000-1.sh` file is included in this ZIP file.

3. Log on to the cluster using SSH with sudo access.

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

4. Copy the `en_sasexe.zip` file from the client to the *EPInstallDir* on the cluster. This example uses secure copy.

```
scp en_sasexe.zip username@hdpclus1: /EPInstallDir
```

Note: The location where you transfer the `en_sasexe.zip` file becomes the SAS Embedded Process home and is referred to as *EPInstallDir* throughout this section.

CAUTION:

After installation, do not delete your SAS Software Depot `standalone_installs` directory or any of its subdirectories. If hot fixes are made available for your software, they are moved to a subdirectory of the `/standalone_installs/SAS_Core_Embedded_Process_Package_for_Hadoop/` directory. The SAS Deployment Manager requires that both the initial installation files and the hot fix file exist in a subdirectory of the original SAS Software Depot /

`standalone_installs/
SAS_Core_Embedded_Process_Package_for_Hadoop/` directory.

Installing the SAS Embedded Process

To install the SAS Embedded Process and SAS Hadoop MapReduce JAR files, 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 9](#).

1. Navigate to the location on your Hadoop master node where you copied the `en_sasexe.zip` file.

```
cd /EPInstallDir
```

For more information, see Step 4 in [“Copying the SAS Embedded Process Install Script” on page 40](#).

2. Ensure that both the `EPInstallDir` folder and the `en_sasexe.zip` file have Read, Write, and Execute permissions (**`chmod 777 -R`**).

3. Unzip the `en_sasexe.zip` file.

```
unzip en_sasexe.zip
```

After the file is unzipped, a **`sasexe`** directory is created in the same location as the `en_sasexe.zip` file. The `sepcorehadoop-9.43000-1.sh` file is in the **`sasexe`** directory.

```
EPInstallDir/sasexe/sepcorehadoop-9.43000-1.sh
```

4. Use the following command to unpack the `sepcorehadoop-9.43000-1.sh` file.

```
./sepcorehadoop-9.43000-1.sh
```

After this script is run and the files are unpacked, the script creates the following directory structure where `EPInstallDir` is the location on the master node from Step 2.

```
EPInstallDir/sasexe/SASEPHome
```

```
EPInstallDir/sasexe/sepcorehadoop-9.43000-1.sh
```

Note: During the install process, the `sepcorehadoop-9.43000-1.sh` is copied to all data nodes. Do not remove or move this file from the **`EPInstallDir/sasexe`** directory.

The **`SASEPHome`** directory structure should look like this.

```
EPInstallDir/sasexe/SASEPHome/bin
```

```
EPInstallDir/sasexe/SASEPHome/misc
```

```
EPInstallDir/sasexe/SASEPHome/sasexe
```

```
EPInstallDir/sasexe/SASEPHome/utilities
```

```
EPInstallDir/sasexe/SASEPHome/jars
```

The **`EPInstallDir/sasexe/SASEPHome/jars`** directory contains the SAS Hadoop MapReduce JAR files.

```
EPInstallDir/sasexe/SASEPHome/jars/sas.hadoop.ep.apache023.jar
```

```
EPInstallDir/sasexe/SASEPHome/jars/sas.hadoop.ep.apache023.nls.jar
```

```
EPInstallDir/sasexe/SASEPHome/jars/sas.hadoop.ep.apache121.jar
```

```
EPInstallDir/sasexe/SASEPHome/jars/sas.hadoop.ep.apache121.nls.jar
```

```
EPInstallDir/sasexe/SASEPHome/jars/sas.hadoop.ep.apache205.jar
```

```
EPInstallDir/sasexe/SASEPHome/jars/sas.hadoop.ep.apache205.nls.jar
```

The **EPInstallDir/sasexe/SASEPHome/bin** directory should look similar to this.

```
EPInstallDir/sasexe/SASEPHome/bin/sasep-admin.sh
```

5. Use the sasep-admin.sh script to deploy the SAS Embedded Process installation across all nodes.

This is when the sepcorehadoop-9.43000-1.sh file is copied to all data nodes.

TIP Many options are available for installing the SAS Embedded Process. We recommend that you review the script syntax before running it. For more information, see “[SASEP-ADMIN.SH Script](#)” on page 43.

Note: If your cluster is secured with Kerberos, complete both steps a and b. If your cluster is not secured with Kerberos, complete only step b.

- a. If your cluster is secured with Kerberos, the HDFS user must have a valid Kerberos ticket to access HDFS. This can be done with kinit.

```
sudo su - root
su - hdfs | hdfs-userid
kinit -kt location of keytab file user for which you are requesting a ticket
exit
```

Note: For all Hadoop distributions except MapR, the default HDFS user is **hdfs**. For MapR distributions, the default HDFS user is **mapr**. You can specify a different user ID with the **-hdfsuser** argument when you run the **sasep-admin.sh -add** script.

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

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

Valid starting    Expires          Service principal
06/20/15 09:51:26 06/27/15 09:51:26 krbtgt/HOST.COMPANY.COM@HOST.COMPANY.COM
        renew until 06/22/15 09:51:26
```

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

```
cd EPInstallDir/sasexe/SASEPHome/bin/
./sasep-admin.sh -add
```

Note: The sasep-admin.sh script must be run from the **EPInstallDir/sasexe/SASEPHome/bin/** location.

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-ADMIN.SH Script](#)” on page 43.

Note: By default, the SAS Embedded Process install script (sasep-admin.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 add the SAS Embedded Process to new nodes at a later time, you should run the sasep-admin.sh script with the **-host <hosts>** option.

6. Verify that the SAS Embedded Process is installed by running the sasep-admin.sh script with the **-check** option.

```
cd EPInstallDir/sasexe/SASEPHome/bin/
./sasep-admin.sh -check
```

This command checks if the SAS Embedded Process is installed on all data nodes.

Note: The sasep-admin.sh -check script does not run successfully if the SAS Embedded Process is not installed.

7. If your distribution is running MapReduce 1 or your SAS client is running on the second maintenance release for SAS 9.4, follow these steps. Otherwise, skip to Step 8.

Note: For more information, see [“Backward Compatibility” on page 9](#).

- a. Verify that the sas.hadoop.ep.apache*.jar files are now in the **hadoop/lib** directory.

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
```

- b. Restart the Hadoop MapReduce service.

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

Note: It is preferable to restart the service by using Cloudera Manager or Ambari (for Hortonworks), if available.

8. Verify that the configuration file, ep-config.xml, was written to the HDFS file system.

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

Note: If your cluster is secured with Kerberos, you need a valid Kerberos ticket to access HDFS. If not, you can use the WebHDFS browser.

Note: The **/sas/ep/config** directory is created automatically when you run the install script. If you used the -epconfig or -genconfig to specify a non-default location, use that location to find the ep-config.xml file.

SASEP-ADMIN.SH Script

Overview of the SASEP-ADMIN.SH Script

The sasep-admin.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.
- Check if the SAS Embedded Process is installed correctly.
- Generate a SAS Embedded Process configuration file and write the file to an HDFS location.

- Create a SAS Hadoop MapReduce JAR file symbolic link in the **hadoop/lib** directory.
- Create a HADOOP_JARS.zip file. This ZIP file contains all required client JAR files.
- Write the installation output to a log file.
- Display all live data nodes on the cluster.
- Display the Hadoop configuration environment.

Note: The `sasep-admin.sh` script must be run from the **EPInstallDir/sasexe/SASEPHome/bin** directory.

Note: You must have sudo access on the master node only to run the `sasep-admin.sh` script. You must also have SSH set up in such a way that the master node can passwordless SSH to all data nodes on the cluster where the SAS Embedded Process is installed.

SASEP-ADMIN.SH Syntax

sasep-admin.sh

```
-add <-link><-epconfig config-filename > <-maxscp number-of-copies>
    <-hostfile host-list-filename | -host <">host-list<">>
    <-hdfsuser user-id> <-log filename>
```

sasep-admin.sh

```
-remove <-epconfig config-filename > <-hostfile host-list-filename | -host <">host-list<">>
    <-hdfsuser user-id> <-log filename>
```

sasep-admin.sh

```
<-genconfig config-filename <-force>>
<-getjars>
<-linklib | -unlinklib>
<-check> <-hostfile host-list-filename | -host <">host-list<">>
<-env>
<-hadoopversion >
<-log filename>
<-nodelist>
<-version >
<-hotfix >
```

Arguments

-add

installs the SAS Embedded Process.

Tip If at a later time you add nodes to the cluster, 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 45](#)

-link

forces the creation of SAS Hadoop MapReduce JAR files symbolic links in the **hadoop/lib** folder during the installation of the SAS Embedded Process.

Restriction	This argument should be used only for backward compatibility (that is, when you install the July 2015 release of SAS 9.4 of the SAS Embedded Process on a client that runs the second maintenance release of SAS 9.4).
Requirement	If you use this argument, you must restart the MapReduce service, the YARN service, or both after the SAS Embedded Process is installed.
Interactions	Use this argument in conjunction with the <code>-add</code> argument to force the creation of the symbolic links. Use the <code>-linklib</code> argument after the SAS Embedded Process is already installed to create the symbolic links.
See	“Backward Compatibility” on page 9 “-linklib” on page 47

-epconfig *config-filename*

generates the SAS Embedded Process configuration file in the specified location.

Default	If the <code>-epconfig</code> argument is not specified, the install script will create the SAS Embedded Process configuration file in the default location <code>/sas/ep/config/ep-config.xml</code> .
Requirement	If the <code>-epconfig</code> argument is not specified a configuration file location must be provided. If you choose a non-default location, you must set the <code>sas.ep.config.file</code> property in the <code>mapred-site.xml</code> file that is on your client machine to the non-default location.
Interaction	Use the <code>-epconfig</code> argument in conjunction with the <code>-add</code> or <code>-remove</code> argument to specify the HDFS location of the configuration file.
Tip	Use the <code>-epconfig</code> argument only if you decide to create the configuration file in a non-default location.
See	“-genconfig <i>config-filename</i> -force” on page 47

-maxscp *number-of-copies*

specifies the maximum number of parallel copies between the master and data nodes.

Default	10
Interaction	Use this argument in conjunction with the <code>-add</code> argument.

-hostfile *host-list-filename*

specifies the full path of a file that contains the list of hosts where the SAS Embedded Process is installed or removed.

Default	The <code>sasep-admin.sh</code> script discovers the cluster topology and uses the retrieved list of data nodes.
Interaction	Use the <code>-hostfile</code> argument in conjunction with the <code>-add</code> when new nodes are added to the cluster.
Tip	You can also assign a host list filename to a UNIX variable, <code>sas_ephosts_file</code> .

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

See [“-hdfsuser user-id” on page 46](#)

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

-host <">host-list<">

specifies the target host or host list where the SAS Embedded Process is installed or removed.

Default The sasep-admin.sh script discovers 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.

Interaction Use the -host argument in conjunction with the -add when new nodes are added to the cluster.

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 46](#)

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

-hdfsuser user-id

specifies the user ID that has Write access to HDFS root directory.

Defaults hdfs for Cloudera, Hortonworks, Pivotal HD, and IBM BigInsights
 mapr for MapR

Interaction Use the -hdfsuser argument in conjunction with the -add or -remove argument to change or remove the HDFS user ID.

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

-log filename

writes the installation output to the specified filename.

Interaction Use the -log argument in conjunction with the -add or -remove argument to write or remove the installation output file.

-remove

removes the SAS Embedded Process.

CAUTION:

If you are using SAS Data Loader, you should remove the QKB and the SAS Data Management Accelerator for Spark from the Hadoop nodes before removing the SAS Embedded Process. For more information, see [“Removing the QKB” on page 92](#) or [“SASDMP_ADMIN.SH Syntax” on page 100](#).

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 45](#)

-genconfig *config-filename* <-force>

generates a new SAS Embedded Process configuration file in the specified location.

Requirement There is no default location associated with the `-genconfig` argument. If you specify the `-genconfig` argument, you must provide a location. If the provided location already exists, you can overwrite it by specifying the `-force` argument. The SAS Embedded Process reads its configuration from the default location `/sas/ep/config/ep-config.xml`. If you decide to generate a new configuration file in a non-default location, you must set the `sas.ep.config.file` property in the `mapred-site.xml` file that is on your client machine to the non-default location.

Interaction Use the `-genconfig` argument to generate a new SAS Embedded Process configuration file when you upgrade to a new version of your Hadoop distribution.

Tip This argument generates an updated `ep-config.xml` file. Use the `-force` argument to overwrite the existing configuration file.

See [“-epconfig *config-filename*” on page 45](#)

-getjars

creates a `HADOOP_JARS.zip` file in the `EPInstall dir/SASEPHome/bin` directory. This ZIP file contains all required client JAR files.

Restrictions This argument is not supported for MapR distributions.

The `-getjars` argument is for use only with TKGrid and High-Performance Analytics. It does not gather all of the JAR files that are required for full functionality of SAS software that requires the use of the SAS Embedded Process. Most of the JAR files that are required for full functionality of SAS software are gathered when you install SAS/ACCESS Interface to Hadoop. For more information, see *SAS Hadoop Configuration Guide for Base SAS and SAS/ACCESS* at <http://support.sas.com/resources/thirdpartysupport/v94/hadoop/>.

Note In the July 2015 release of SAS 9.4, the `SAS_HADOOP_JAR_PATH` environment variable has replaced the need for copying the Hadoop JAR files to the client machine with the exception of High-Performance Analytics. The `SAS_HADOOP_JAR_PATH` environment variable is usually set when you install SAS/ACCESS Interface to Hadoop.

Tip You can move this ZIP file to your client machine and unpack it. If you want to replace the existing JAR files, move it to the same directory where you previously unpacked the existing JAR files.

-linklib

creates SAS Hadoop MapReduce JAR file symbolic links in the `hadoop/lib` folder.

Restriction	This argument should be used only for backward compatibility (that is, when you install the July 2015 release of SAS 9.4 of the SAS Embedded Process on a client that runs the second maintenance release of SAS 9.4).
Requirement	If you use this argument, you must restart the MapReduce service, the YARN service, or both after the SAS Embedded Process is installed.
Interaction	Use the <code>-linklib</code> argument after the SAS Embedded Process is already installed to create the symbolic links. Use the <code>-link</code> argument in conjunction with the <code>-add</code> argument to force the creation of the symbolic links.
See	“Backward Compatibility” on page 9 “-link” on page 44

-unlinklib

removes SAS Hadoop MapReduce JAR file symbolic links in the `hadoop/lib` folder.

Restriction	This argument should be used only for backward compatibility (that is, when you install the July 2015 release of SAS 9.4 of the SAS Embedded Process on a client that runs the second maintenance release of SAS 9.4).
Requirement	If you use this argument, you must restart the MapReduce service, the YARN service, or both after the SAS Embedded Process is installed.
See	“Backward Compatibility” on page 9

-check

checks if the SAS Embedded Process is installed correctly on all data nodes.

-env

displays the Hadoop configuration environment.

-hadoopversion

displays the Hadoop version information for the cluster.

-nodelist

displays all live DataNodes on the cluster.

-version

displays the version of the SAS Embedded Process that is installed.

-hotfix

distributes a hot fix package.

Requirements	Hot fixes must be installed using the same user ID who performed the initial software installation.
---------------------	---

Hot fixes should be installed following the installation instructions provided by SAS Technical Support.

Chapter 5

Additional Configuration for the SAS Embedded Process

Overview of Additional Configuration Tasks	49
Additional Configuration Needed to Use HCatalog File Formats	50
Overview of HCatalog File Types	50
Prerequisites for HCatalog Support	50
SAS Client Configuration	50
SAS Server-Side Configuration	51
Additional Configuration for Cloudera 5.4 or IBM BigInsights 4.0	52
Additional Configuration for Hortonworks 2.2	52
Additional Configuration for IBM BigInsights	53
Additional Configuration for IBM BigInsights 3.0	53
Additional Configuration for IBM BigInsights 4.1	53
Adding the YARN Application CLASSPATH to the Configuration File for MapR Distributions	54
Adjusting the SAS Embedded Process Performance	54
Overview of the ep-config.xml File	54
Changing the Trace Level	55
Specifying the Number of MapReduce Tasks	55
Specifying the Amount of Memory That the SAS Embedded Process Uses	56
Adding the SAS Embedded Process to Nodes after the Initial Deployment	56

Overview of Additional Configuration Tasks

After you have installed the SAS Embedded Process either manually or by using the SAS Deployment Manager, the following additional configuration tasks must be performed:

- [“Additional Configuration Needed to Use HCatalog File Formats” on page 50.](#)
- [“Additional Configuration for Hortonworks 2.2” on page 52.](#)
- [“Additional Configuration for IBM BigInsights” on page 53.](#)
- [“Adding the YARN Application CLASSPATH to the Configuration File for MapR Distributions” on page 54.](#)
- [“Adjusting the SAS Embedded Process Performance” on page 54.](#)

- “Adding the SAS Embedded Process to Nodes after the Initial Deployment” on page 56.

Additional Configuration Needed to Use HCatalog File Formats

Overview of HCatalog File Types

HCatalog is a table management layer that presents a relational view of data in the HDFS to applications within the Hadoop ecosystem. With HCatalog, data structures that are registered in the Hive metastore, including SAS data, can be accessed through standard MapReduce code and Pig. HCatalog is part of Apache Hive.

The SAS Embedded Process for Hadoop uses HCatalog to process the following complex, non-delimited file formats: Avro, ORC, Parquet, and RCFile.

Prerequisites for HCatalog Support

If you plan to access complex, non-delimited file types such as Avro or Parquet, you must perform these additional prerequisites:

- Hive and HCatalog must be installed on all nodes of the cluster.
- HCatalog support depends on the version of Hive that is running on your Hadoop distribution. See the following table for more information.

Note: For MapR distributions, Hive 0.13.0 build: 1501 or later must be installed for access to any HCatalog file type.

File Type	Required Hive Version
Avro	0.14
ORC	0.11
Parquet	0.13
RCFile	0.6

SAS Client Configuration

Note: If you used the SAS Deployment Manager to install the SAS Embedded Process, these configuration tasks are not necessary. It was completed using the SAS Deployment Manager.

The following additional configuration tasks must be performed:

- The hive-site.xml configuration file must be in the SAS_HADOOP_CONFIG_PATH.
- The following Hive or HCatalog JAR files must be in the SAS_HADOOP_JAR_PATH.


```
hive-hcatalog-core-*.jar
hive-webhcat-java-client-*.jar
jdo-api*.jar
```

- If you are using MapR, the following Hive or HCatalog JAR files must be in the SAS_HADOOP_JAR_PATH.


```
hive-hcatalog-hbase-storage-handler-0.13.0-mapr-1408.jar
hive-hcatalog-server-extensions-0.13.0-mapr-1408.jar
hive-hcatalog-pig-adapter-0.13.0-mapr-1408.jar
datanucleus-api-jdo-3.2.6.jar
datanucleus-core-3.2.10.jar
datanucleus-rdbms-3.2.9.jar
```
- To access Avro file types, the avro-1.7.4.jar file must be added to the SAS_HADOOP_JAR_PATH environment variable.
- To access Parquet file types with Cloudera 5.1, the parquet-hadoop-bundle.jar file must be added to the directory defined in the SAS_HADOOP_JAR_PATH environment variable.
- If your distribution is running Hive 0.12, the jersey-client-1.9.jar must be added to the SAS_HADOOP_JAR_PATH environment variable.

For more information about the SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables, see the *SAS Hadoop Configuration Guide for Base SAS and SAS/ACCESS*.

SAS Server-Side Configuration

If your distribution is running MapReduce 2 and YARN, the SAS Embedded Process installation automatically sets the HCatalog CLASSPATH in the ep-config.xml file. Otherwise, you must manually include the HCatalog JAR files in either the MapReduce 2 library or the Hadoop CLASSPATH. For Hadoop distributions that run with MapReduce 1, you must also manually add the HCatalog CLASSPATH to the MapReduce CLASSPATH.

Here is an example for a Cloudera distribution.

```
<property>
  <name>mapreduce.application.classpath</name>
  <value>/EPInstallDir/SASEPHome/jars/sas.hadoop.ep.apache205.jar,/EPInstallDir/
/SASEPHome/jars/sas.hadoop.ep.apache205.nls.jar,/opt/cloudera/parcels/
CDH-5.2.0-1.cdh5.2.0.p0.36/bin/./lib/hive/lib/*,
/opt/cloudera/parcels/CDH-5.2.0-1.cdh5.2.0.p0.36/lib/hive-hcatalog/libexec/
../share/hcatalog/*,/opt/cloudera/parcels/CDH-5.2.0-1.cdh5.2.0.p0.36/
lib/hive-hcatalog/libexec/./share/hcatalog/storage-handlers/hbase/lib/*,
$HADOOP_MAPRED_HOME/*,$HADOOP_MAPRED_HOME/lib/*,$MR2_CLASSPATH</value>
</property>
```

Here is an example for a Hortonworks distribution.

```
<property>
  <name>mapreduce.application.classpath</name>
  <value>/EPInstallDir/SASEPHome/jars/sas.hadoop.ep.apache205.jar,/SASEPHome/
jars/sas.hadoop.ep.apache205.nls.jar,/usr/lib/hive-hcatalog/libexec/
../share/hcatalog/*,/usr/lib/hive-hcatalog/libexec/./share/hcatalog/
storage-handlers/hbase/lib/*,/usr/lib/hive/lib/*,$HADOOP_MAPRED_HOME/
```

```
share/hadoop/mapreduce/*,$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/
lib/*</value>
</property>
```

Additional Configuration for Cloudera 5.4 or IBM BigInsights 4.0

If you are using Cloudera 5.4 or IBM Big Insights 4.0 with HCatalog sources, you must add the HADOOP_HOME environment variable in the Windows environment. An example of the value for this option is **HADOOP_HOME=c:\hadoop**.

The directory should contain a subdirectory named **bin** which must contain the winutils.exe for your distribution. Please contact your distribution vendor for a copy of the winutils.exe file.

Additional Configuration for Hortonworks 2.2

If you are installing the SAS Embedded Process on Hortonworks 2.2, you must manually revise the following properties in the mapred-site.xml property file on the SAS client side. Otherwise, an error occurs when you submit a program to Hadoop.

Use the **hadoop version** command to determine the exact version number of your distribution to use in place of **\${hdp.version}**. This example assumes that the current version is 2.2.0.0-2041.

mapreduce.application.framework.path

Change

```
/hdp/apps/${hdp.version}/mapreduce/mapreduce.tar.gz#mr-framework
```

to

```
/hdp/apps/2.2.0.0-2041/mapreduce/mapreduce.tar.gz#yarn
```

mapreduce.application.classpath

Change

```
$PWD/mr-framework/hadoop/share/hadoop/mapreduce/*:$PWD/mr-framework/
hadoop/share/hadoop/mapreduce/lib/*:$PWD/mr-framework/hadoop/share/
hadoop/common/*:$PWD/mr-framework/hadoop/share/hadoop/common/lib/*:$PWD/
mr-framework/hadoop/share/hadoop/yarn/*:$PWD/mr-framework/hadoop/share/hadoop/
yarn/lib/*:$PWD/mr-framework/hadoop/share/hadoop/hdfs/*:$PWD/mr-framework/
hadoop/share/hadoop/hdfs/lib/*:/usr/hdp/${hdp.version}/hadoop/lib/
hadoop-lzo-0.6.0.${hdp.version}.jar:/etc/hadoop/conf/secure
```

to

```
/usr/hdp/2.2.0.0-2041/hadoop-mapreduce/*:/usr/hdp/2.2.0.0-2041/hadoop-mapreduce/
lib/*:/usr/hdp/2.2.0.0-2041/hadoop/*:/usr/hdp/2.2.0.0-2041/hadoop/lib/
*/usr/hdp/2.2.0.0-2041/hadoop-yarn/*:/usr/hdp/2.2.0.0-2041/hadoop-yarn/lib/
*/usr/hdp/2.2.0.0-2041/hadoop-hdfs/*:/usr/hdp/2.2.0.0-2041/hadoop-hdfs/lib/
*/usr/hdp/2.2.0.0-2041/hadoop/lib/hadoop-lzo-0.6.0.2.2.0.0-2041.jar:/etc/
hadoop/conf/secure
```

yarn.app.mapreduce.am.admin-command-opts

Change

```
-Dhdp.version=${hdp.version}
```

to

```
-Dhdp.version=2.2.0.0-2041
```

yarn.app.mapreduce.am.command-opts

Change

```
-Xmx410m -Dhdp.version=${hdp.version}
```

to

```
-Xmx410m -Dhdp.version=2.2.0.0-2041
```

Note: If you upgrade your Hortonworks distribution and the version changes, you need to make this update again.

Additional Configuration for IBM BigInsights

Additional Configuration for IBM BigInsights 3.0

If you are installing the SAS Embedded Process on IBM BigInsights 3.0, you must revise the `hadoop.job.history.user.location` property in the `core-site.xml` file that is in the `SAS_HADOOP_CONFIG_PATH` to a value other than the output directory. Otherwise, loading data into the Hive table fails. Here is an example where the output directory is set to `/tmp`.

```
<property>
  <name>hadoop.job.history.user.location</name>
  <value>/tmp</value>
</property>
```

Additional Configuration for IBM BigInsights 4.1

If you are installing the SAS Embedded Process on IBM BigInsights 4.1, you must manually revise the following properties in the `mapred-site.xml` property file on the SAS client side. Otherwise, an error occurs when you submit a program to Hadoop.

The IBM BigInsights 4.1 `mapred-site.xml` property file contains numerous values with the parameter `${iop.version}`, including `mapreduce.application.classpath`. You must change `${iop.version}` to the actual cluster version. This example assumes that the current version is 4.1.0.0 and changes the `mapreduce.admin.user.env` property.

Change

```
<property>
  <name>mapreduce.admin.user.env</name>
  <value>/LD_LIBRARY_PATH=/usr/iop/${iop.version}/hadoop/lib/native</value>
</property>
```

to

```
<property>
  <name>mapreduce.admin.user.env</name>
  <value>/LD_LIBRARY_PATH=/usr/iop/4.1.0.0/hadoop/lib/native</value>
</property>
```

Adding the YARN Application CLASSPATH to the Configuration File for MapR Distributions

Two main configuration properties specify the application CLASSPATH: `yarn.application.classpath` and `mapreduce.application.classpath`. If you do not specify the YARN application CLASSPATH, MapR takes the default CLASSPATH. However, if you specify the MapReduce application CLASSPATH, the YARN application CLASSPATH is ignored. The SAS Embedded Process for Hadoop requires both the MapReduce application CLASSPATH and the YARN application CLASSPATH.

To ensure the existence of the YARN application CLASSPATH, you must manually add the YARN application CLASSPATH to the `yarn-site.xml` file. Without the manual definition in the configuration file, the MapReduce application master fails to start a container.

The default YARN application CLASSPATH for Linux is:

```
$HADOOP_CONF_DIR,
$HADOOP_COMMON_HOME/share/hadoop/common/*,
$HADOOP_COMMON_HOME/share/hadoop/common/lib/*,
$HADOOP_HDFS_HOME/share/hadoop/hdfs/*,
$HADOOP_HDFS_HOME/share/hadoop/hdfs/lib/*,
$HADOOP_YARN_HOME/share/hadoop/yarn/*,
$HADOOP_YARN_HOME/share/hadoop/yarn/lib/*
```

The default YARN application CLASSPATH for Windows is:

```
%HADOOP_CONF_DIR%,
%HADOOP_COMMON_HOME%/share/hadoop/common/*,
%HADOOP_COMMON_HOME%/share/hadoop/common/lib/*,
%HADOOP_HDFS_HOME%/share/hadoop/hdfs/*,
%HADOOP_HDFS_HOME%/share/hadoop/hdfs/lib/*,
%HADOOP_YARN_HOME%/share/hadoop/yarn/*,
%HADOOP_YARN_HOME%/share/hadoop/yarn/lib/*
```

Note: On MapR, the YARN application CLASSPATH does not resolve the symbols or variables specified in the paths (`$HADOOP_HDFS_HOME`, and so on).

Adjusting the SAS Embedded Process Performance

Overview of the *ep-config.xml* File

You can adjust how the SAS Embedded Process runs by changing properties in the `ep-config.xml` file.

The `ep-config.xml` file is created when you install the SAS Embedded Process. By default, the file is located in the `/sas/ep/config/ep-config.xml` directory.

You can change property values that enable you to perform the following tasks:

- change trace levels

For more information, see [“Changing the Trace Level” on page 55](#).

- specify the number of SAS Embedded Process MapReduce 1 tasks per node

For more information, see [“Specifying the Number of MapReduce Tasks” on page 55](#).

- specify the maximum amount of memory in bytes that the SAS Embedded Process is allowed to use

For more information, see [“Specifying the Amount of Memory That the SAS Embedded Process Uses” on page 56](#).

Changing the Trace Level

You can modify the level of tracing by changing the value of the `sas.ep.server.trace.level` property in the `ep-config.xml` file. The default value is 4 (TRACE_NOTE).

```
<property>
  <name>sas.ep.server.trace.level</name>
  <value>trace-level</value>
</property>
```

The *trace-level* represents the level of trace that is produced by the SAS Embedded Process. *trace-level* can be one of the following values:

```
0
  TRACE_OFF
1
  TRACE_FATAL
2
  TRACE_ERROR
3
  TRACE_WARN
4
  TRACE_NOTE
5
  TRACE_INFO
10
  TRACE_ALL
```

Note: Tracing requires that an `/opt/SAS` directory to exist on every node of the cluster when the SAS Embedded Process is installed. If the folder does not exist or does not have Write permission, the SAS Embedded Process job fails.

Specifying the Number of MapReduce Tasks

You can specify the number of SAS Embedded Process MapReduce Tasks per node by changing the `sas.ep.superreader.tasks.per.node` property in the `ep-config.xml` file. The default number of tasks is 6.

```
<property>
  <name>sas.ep.superreader.tasks.per.node</name>
  <value>number-of-tasks</value>
</property>
```

Specifying the Amount of Memory That the SAS Embedded Process Uses

You can specify the amount of memory in bytes that the SAS Embedded Process is allowed to use with MapReduce 1 by changing the `sas.ep.max.memory` property in the `ep-config.xml` file. The default value is 2147483647 bytes.

```
<property>
  <name>sas.ep.max.memory</name>
  <value>number-of-bytes</value>
</property>
```

Note: This property is valid only for Hadoop distributions that are running MapReduce 1.

If your Hadoop distribution is running MapReduce 2, this value does not supersede the YARN maximum memory per task. Adjust the YARN container limit to change the amount of memory that the SAS Embedded Process is allowed to use.

Adding the SAS Embedded Process to Nodes after the Initial Deployment

After the initial deployment of the SAS Embedded Process, additional nodes might be added to your cluster or nodes might need to be replaced. In these instances, you can install the SAS Embedded Process on the new nodes.

Follow these steps:

1. Log on to HDFS.

```
sudo su - root
su - hdfs | hdfs-userid
```

Note: If your cluster is secured with Kerberos, the HDFS user must have a Kerberos ticket to access HDFS. This can be done with `kinit`.

2. Navigate to the `/sas/ep/config/` directory on HDFS.
3. Remove the `ep-config.xml` file from HDFS.

```
cd /sas/ep/config/
hadoop fs -rm ep-config.xml
```

4. Run the `sasep-admin.sh` script and specify the nodes on which you want to install the SAS Embedded Process.

```
cd EPInstallDir/SASEPHome/bin/
./sasep-admin.sh -add -hostfile host-list-filename | -host <">host-list<">
```

Part 3

Administrator's Guide for SAS Data Loader for Hadoop

<i>Chapter 6</i>	
Introduction	<i>59</i>
<i>Chapter 7</i>	
Cloudera Manager and Ambari Deployment	<i>61</i>
<i>Chapter 8</i>	
Standard Deployment	<i>73</i>
<i>Chapter 9</i>	
SAS In-Database Deployment Package for Hadoop	<i>79</i>
<i>Chapter 10</i>	
SAS In-Database Technologies for Data Quality Directives	<i>83</i>
<i>Chapter 11</i>	
SAS Data Management Accelerator for Spark	<i>95</i>
<i>Chapter 12</i>	
Configuring the Hadoop Cluster	<i>105</i>
<i>Chapter 13</i>	
Configuring Kerberos	<i>111</i>

Chapter 6

Introduction

SAS Data Loader and SAS In-Database Technologies for Hadoop	59
About SAS In-Database Technologies for Hadoop	59
SAS In-Database Deployment Package	59
SAS Data Quality Accelerator and SAS Quality Knowledge Base	59
SAS Data Management Accelerator for Spark	60
System Requirements	60
Privileges	60
Support for the vApp User	60
Deployment	60

SAS Data Loader and SAS In-Database Technologies for Hadoop

About SAS In-Database Technologies for Hadoop

SAS In-Database Technologies for Hadoop supports the Hadoop operations of SAS Data Loader for Hadoop. SAS Data Loader for Hadoop is web-client software that is installed as a vApp and is run on a virtual machine. The following products are included in SAS In-Database Technologies for Hadoop: SAS In-Database Deployment Package, SAS Data Quality Accelerator, SAS Quality Knowledge Base, and SAS Data Management Accelerator for Spark.

SAS In-Database Deployment Package

The SAS In-Database Deployment Package includes the SAS Embedded Process and the SAS Hadoop MapReduce JAR files. The SAS Embedded Process runs within MapReduce to read and write data. You must deploy the SAS In-Database Deployment Package. Deploying and configuring the SAS In-Database Deployment Package needs to be done only once for each Hadoop cluster.

SAS Data Quality Accelerator and SAS Quality Knowledge Base

The data quality directives in SAS Data Loader for Hadoop are supported by SAS Data Quality Accelerator and the SAS Quality Knowledge Base (QKB). Both are required

components for SAS Data Loader for Hadoop and are included in SAS In-Database Technologies for Hadoop. The QKB is a collection of files that store data and logic to support data management operations. A QKB is specific to a locale, that is, to a country and language. SAS Data Loader for Hadoop data quality directives reference the QKB when performing data quality operations on your data. It is recommended that you periodically update the QKB. For more information, see [“Updating and Customizing the QKB” on page 89](#).

SAS Data Management Accelerator for Spark

Spark is a processing engine that is compatible with Hadoop data. SAS Data Management Accelerator for Spark runs data integration and data quality tasks in a Spark environment. These tasks include mapping columns, summarizing columns, performing data quality tasks such as clustering and survivorship, and standardization of data. Deploy SAS Data Management Accelerator for Spark only if Spark is available on the cluster.

System Requirements

You can review the system requirements for the SAS Data Loader offering at the following location:

<http://support.sas.com/documentation/installcenter/en/ikdmdhdpvofrsr/68979/PDF/default/sreq.pdf>

Privileges

The Hadoop administrator installing SAS In-Database Technologies for Hadoop must have `sudo` or root privileges on the Hadoop cluster.

Support for the vApp User

You must configure the Hadoop cluster and provide certain values to the vApp user. For specific information about what you must provide, see [“Providing vApp User Configuration Information” on page 108](#) and [“Providing vApp User Configuration Information” on page 115](#).

Deployment

The procedure for installing and deploying SAS In-Database Technologies for Hadoop depends on which distribution you have downloaded. For specific instructions, see [Chapter 7, “Cloudera Manager and Ambari Deployment,” on page 61](#) or [Chapter 8, “Standard Deployment,” on page 73](#).

Chapter 7

Cloudera Manager and Ambari Deployment

Getting Started with the Deployment	61
About the Deployment	61
Before Deployment	62
Overview of Deployment Steps	62
Configure Kerberos	62
Obtain and Extract Zipped Files	63
Deploy Files	63
Cloudera Manager	63
Ambari	65
Edit SAS Hadoop Configuration Properties File	67
Collect Files	68
Configure the Hadoop Cluster	68
Deactivating or Removing Existing Versions	69
About Deactivating and Removing	69
Cloudera Manager	69
Ambari	70

Getting Started with the Deployment

About the Deployment

This chapter describes deployment of SAS Data Loader 2.4 for Hadoop Cloudera and SAS Data Loader 2.4 for Hadoop Hortonworks. SAS sends an email to a contact person at your business or organization. This email specifies whether the product is SAS Data Loader 2.4 for Hadoop Cloudera or SAS Data Loader 2.4 for Hadoop Hortonworks and includes instructions for downloading a ZIP file. The ZIP file contains all product files that are required for installation of SAS In-Database Technologies for Hadoop on the Hadoop cluster. The contact person is responsible for making the ZIP file available to you.

The individual components of SAS In-Database Technologies for Hadoop are described in the following chapters: [Chapter 9, “SAS In-Database Deployment Package for Hadoop,”](#) on page 79, [Chapter 10, “SAS In-Database Technologies for Data Quality](#)

[Directives,” on page 83](#), and [Chapter 11, “SAS Data Management Accelerator for Spark,” on page 95](#).

Note: For further specific information about SAS In-Database Technologies for Hadoop, see [Chapter 6, “Introduction,” on page 59](#).

Before Deployment

If you are installing a new version or reinstalling a previous version of SAS In-Database Technologies for Hadoop, you must deactivate or remove other existing SAS In-Database Technologies for Hadoop parcels or stacks after installing the new one. More than one parcel or stack can be deployed on your cluster, but only one parcel can be activated at a time. See [“Deactivating or Removing Existing Versions” on page 69](#).

Overview of Deployment Steps

1. Configure Kerberos, if appropriate, and then provide required configuration values to the vApp user.
2. Identify a Windows server in a shared network location that is accessible to vApp users.
3. Review the Hadoop Environment topic in the [system requirements](#) for SAS Data Loader 2.4.
4. Obtain the ZIP file.
5. Extract zipped files.
6. Deploy services using Cloudera Manager or Ambari.
7. Edit the Hadoop configuration file.
8. Collect required files from the Hadoop cluster.
9. Make the required vApp directory available on the Windows server in the shared network location.
10. Configure the Hadoop cluster, and then provide required configuration values to the vApp user.

Note: If you switch to a different distribution of Hadoop after the initial installation of SAS In-Database Technologies for Hadoop, you must reinstall and reconfigure SAS In-Database Technologies for Hadoop on the new Hadoop cluster.

Configure Kerberos

If you are using Kerberos, you must have all valid tickets in place on the cluster. When deploying SAS In-Database Technologies for Hadoop, the HDFS user must have a valid ticket. See [Chapter 13, “Configuring Kerberos,” on page 111](#). Provide the necessary configuration values to the vApp user.

Obtain and Extract Zipped Files

Download the ZIP file to a Linux machine that is accessible to the NameNode of the Hadoop cluster. Extract the contents of the file. Depending on whether your Hadoop distribution is Cloudera or Hortonworks, you see one of the following file structures under `\products\package_name`.

Figure 7.1 Cloudera Manager File Structure

```
*
|-Admin
|---bin
|-User
|---SASWorkspace
|----hadoop
|-cdhmanager
|---dmspark
|---indatabase
|---qkb
|-data
|-etc
|-lib
```

Figure 7.2 Ambari File Structure

```
*
|-Admin
|---bin
|-User
|---SASWorkspace
|----hadoop
|-ambari
|---dmspark
|---indatabase
|---qkb
|-bin
|---dmspark
|---indatabase
|---qkb
|-data
|-etc
|-lib
```

Deploy Files

The following deployment steps assume that the extracted ZIP files are located directly on the Cloudera Manager or Ambari server or that you have placed the extracted ZIP files on a network location that is accessible to Cloudera Manager or the Ambari server.

Cloudera Manager

Creating Parcels

Navigate to the **Admin** directory and execute the following:

```
./bin/create_dl_parcel.sh -s pathname/Admin/cdhmanager -t pathname/Admin/parcels -v distro
```

where *pathname* is the location of the unzipped file structure and *distro* is one of the following Linux distributions: redhat5, redhat6, suse11x, ubuntu10, ubuntu12, ubuntu14, debian6, or debian7. You can also enter `-v all` to specify all distributions.

Deploying the Services to Cloudera

You must deploy SAS In-Database Deployment Package (SASEP) and SAS Quality Knowledge Base (SASQKB). Deploy SAS Data Management Accelerator for Spark (SASDMSPARK) only if Spark is available on the cluster. It is recommended that you periodically update the QKB. For more information, see [“Updating and Customizing the QKB” on page 89](#).

1. Copy the following Custom Service Descriptor (CSD) files to the Cloudera Manager host, where *pathname* is the path to the unzipped files:

```
cp pathname/Admin/cdhmanager/indatabase/SASEP-9.43.jar /opt/cloudera/csd
cp pathname/Admin/cdhmanager/qkb/SASQKB-26.jar /opt/cloudera/csd
cp pathname/Admin/cdhmanager/dmspark/SASDMSPARK-2.4.jar /opt/cloudera/csd
```

2. Copy the following parcels to the Cloudera Manager host, where *pathname* is the path to the unzipped files:

```
cp pathname/Admin/cdhmanager/indatabase/SASEP-9.43.pdl.24-el6.parcel* /opt/cloudera/parcel-repo
cp pathname/Admin/cdhmanager/qkb/SASQKB-26.pdl.24-el6.parcel* /opt/cloudera/parcel-repo
cp pathname/Admin/cdhmanager/dmspark/SASDMSPARK-2.4.pdl.24-el6.parcel* /opt/cloudera/parcel-repo
```

3. On the Cloudera Manager host, change the ownership permissions on each of the following files:

```
chown cloudera-scm:cloudera-scm /opt/cloudera/csd/SASEP-9.43.jar
chown cloudera-scm:cloudera-scm /opt/cloudera/csd/SASQKB-26.jar
chown cloudera-scm:cloudera-scm /opt/cloudera/csd/SASDMSPARK-2.4.jar
chown cloudera-scm:cloudera-scm /opt/cloudera/parcel-repo/SASEP-9.43.pdl.24-el6.parcel*
chown cloudera-scm:cloudera-scm /opt/cloudera/parcel-repo/SASQKB-26.pdl.24-el6.parcel*
chown cloudera-scm:cloudera-scm /opt/cloudera/parcel-repo/SASDMSPARK-2.4.pdl.24-el6.parcel*
```

Note: If installing all three services, you can condense these commands as follows:

```
chown cloudera-scm:cloudera-scm /opt/cloudera/csd/SAS*
chown cloudera-scm:cloudera-scm /opt/cloudera/parcel-repo/SAS*
```

4. Restart the Cloudera Manager server by running the following command:

```
sudo service cloudera-scm-server restart
```

5. Log on to Cloudera Manager.
6. Activate each of the three parcels.

Note: The following steps are iterative. For example, you must activate SASEP before activating SASQKB before activating SASDMSPARK.

Select **Hosts** ⇨ **Parcels**. The parcels are located under your cluster. Complete the following for each parcel:

- a. Click **Distribute** to copy the parcel to all nodes.
- b. Click **Activate**. You are prompted either to restart the cluster or close the window.
- c. When prompted, click **Close**.

CAUTION:

Do not restart the cluster.

7. Add each of the three services. This creates files in HDFS.

Note:

- The following steps are iterative. For example, you must add the SASEP service before adding SASQKB before adding SASDMSPARK.

- After adding a service, do not proceed to add another service without stopping the service that you have just added. If you proceed to add another service while any of the other services are running, an error might be returned.

Complete the following for each service:

- Navigate to Cloudera Manager Home.
- In Cloudera Manager, select the drop-down arrow next to the name of the cluster, and then select **Add a Service**. The Add Service Wizard appears.
- Select the service and click **Continue**.
- Select the dependencies for the service in the **Add Service Wizard** ⇒ **Select the set of dependencies for your new service** page. Click **Continue**.

Note: The dependencies are automatically selected for this service.

- Select a node for the service in the **Add Service Wizard** ⇒ **Customize Role Assignments** page. Click **OK**, and then click **Continue**.

A file is added to HDFS for each of the services as follows:

- SASEP: `/sas/ep/config/ep-config.xml`
- SASQKB: `/sas/qkb/default.idx`
- SASDMSPARK: `/sas/ep/config/dmp-config.xml`

- Click **Continue**, and then click **Finish**.

Note: If the services that you have just deployed are started, navigate to Cloudera Manager Home and stop them.

Ambari

Deploying the Services to Hortonworks

You must deploy SAS In-Database Deployment Package (SASEP) and SAS Quality Knowledge Base (SASQKB). Deploy SAS Data Management Accelerator for Spark (SASDMSPARK) only if Spark is available on the cluster. It is recommended that you periodically update the QKB. For more information, see [“Updating and Customizing the QKB” on page 89](#).

Note: You must complete the following steps on the Ambari Server host as the root user or as a user with **sudo** access

- Copy the following files to the Ambari host, where *pathname* is the path to the unzipped files:

```
cp pathname/Admin/ambari/indatabase/SASEPINSTALL.gz /var/lib/ambari-server/resources
cp pathname/Admin/ambari/qkb/QKBINSTALL.gz /var/lib/ambari-server/resources
cp pathname/Admin/ambari/dmspark/SASDMSPARKINSTALL.gz /var/lib/ambari-server/resources
```

- Execute the following command:

```
cd /var/lib/ambari-server/resources
```

- Extract the contents of the following files:

```
tar -xvf SASEPINSTALL.gz
tar -xvf SASDMSPARKINSTALL.gz
```

Note: You do not need to extract QKBINSTALL.gz. During the deployment process, this file is extracted to each of the nodes in the cluster.

4. Copy and extract the following files to the Ambari host, where *pathname* is the path to the unzipped files:

```
cp pathname/Admin/ambari/indatabase/stacks.gz /var/lib/ambari-server/resources/stacks/HDP/2.0.6/services
cd /var/lib/ambari-server/resources/stacks/HDP/2.0.6/services
tar -xvf stacks.gz
rm stacks.gz
```

```
cp pathname/Admin/ambari/qkb/stacks.gz /var/lib/ambari-server/resources/stacks/HDP/2.0.6/services
cd /var/lib/ambari-server/resources/stacks/HDP/2.0.6/services
tar -xvf stacks.gz
rm stacks.gz
```

```
cp pathname/Admin/ambari/dmspark/stacks.gz /var/lib/ambari-server/resources/stacks/HDP/2.0.6/services
cd /var/lib/ambari-server/resources/stacks/HDP/2.0.6/services
tar -xvf stacks.gz
rm stacks.gz
```

5. Restart the Ambari server by running the following command:

```
sudo ambari-server restart
```

6. Log on to Ambari.

7. Deploy the services:

- a. Click **Actions** and select + **Add Service**.

The **Add Service Wizard** page and the **Choose Services** panel appear.

- b. In the **Choose Services** panel, select **SASEP SERVICE**, **SAS QKB**, and **SASDMSPARK**. Click **Next**.

The **Assign Slaves and Clients** panel appears.

- c. In the **Assign Slaves and Clients** panel, select the NameNode, HDFS_CLIENT, and HCAT_CLIENT under **Client** where you want the stack to be deployed.

The **Customize Services** panel appears.

The SASQKB, SASDMSPARK, and SASEP SERVICE stacks are listed.

- d. Do not change any settings on the **Customize Services** panel. Click **Next**.

Note: If your cluster is secured with Kerberos, the **Configure Identities** panel appears. Enter your Kerberos credentials in the **admin_principal** and **admin_password** text boxes.

If your cluster is secured with Kerberos, the **Configure Identities** panel appears. Enter your Kerberos credentials in the **admin_principal** and **admin_password** text boxes. Click **Next**.

The **Review** panel appears.

- e. Review the information about the panel. If everything is correct, click **Deploy**.

The **Install, Start, and Test** panel appears. When the stack is installed on all nodes, click **Next**.

The **Summary** panel appears.

- f. Click **Complete**. The stacks are now installed on all nodes of the cluster.

SASEP SERVICE, SASQKB, and SASDMSHARK are displayed on the Ambari dashboard.

- g. After deploying all of the services, verify that the following files exist in the Hadoop file system:
 - SASEP: `/sas/ep/config/ep-config.xml`
 - SASQKB: `/sas/qkb/default.idx`
 - SASDMSHARK: `/sas/ep/config/dmp-config.xml`

Edit SAS Hadoop Configuration Properties File

In the unzipped file structure, you must edit the file **Admin/etc/sas_hadoop_config.properties** to supply certain information that cannot be obtained automatically. Optional settings also exist that you might want to enable.

For the following section:

```
hadoop.client.config.filepath=<replace with full path>/User/SASWorkspace/hadoop/conf
hadoop.client.jar.filepath=<replace with full path>/User/SASWorkspace/hadoop/lib
hadoop.client.repository.path=<replace with full path>/User/SASWorkspace/hadoop/repository/
hadoop.client.configfile.repository=<replace with full path>/User/SASWorkspace/hadoop/repository
```

Replace *<replace with full path>* with the full path to the location where the ZIP file was unzipped.

For the following section:

```
hadoop.cluster.manager.hostname=
hadoop.cluster.manager.port=
hadoop.cluster.hivenode.admin.account=
hadoop.cluster.manager.admin.account=
```

Set **hadoop.cluster.manager.hostname** to the value of the host where either Cloudera Manager or Ambari is running.

Set **hadoop.cluster.manager.port** to the value of the port on which Cloudera Manager or Ambari is listening. Default values are provided.

Set **hadoop.cluster.hivenode.admin.account** to the value of a valid account on the machine on which the Hive2 service is running.

Set **hadoop.cluster.manager.admin.account** to the value of a valid Cloudera Manager or Ambari account.

For the following section:

```
hadoop.client.sasconfig.logfile.path=logs
hadoop.client.sasconfig.logfile.name=logs/sashadoopconfig/sashadoopconfig.log
hadoop.client.config.log.level=0
```

The default values of **logs** and **sashadoopconfig.log** create the directory **Admin/logs** and the filename **sashadoopconfig.log**, respectively. Both of these values can be changed if you prefer.

You can set the value of **hadoop.client.config.log.level** to 3 to increase the amount of information logged.

Note: If your distribution is secured with Kerberos,

- set `hadoop.cluster.hivenode.credential.type=kerberos`
- set `hadoop.client.config.log.level=3`

If you use Cloudera Manager and it manages multiple clusters, provide the name of the cluster to use for the value of `hadoop.cluster.manager.clustername=`.

Collect Files

Certain files must be collected from the Hadoop cluster and made available to the vApp user.

In the unzipped file structure, navigate to the **Admin** directory and run the following command:

```
./bin/hadoop_extract.sh
```

You are asked for two passwords:

- The cluster password is the password to the cluster manager administrative interface that corresponds to the `hadoop.cluster.manager.admin.account` name entered in the `sas_hadoop_config.properties` file.
- The hive password is the password for the SSH user account that is allowed to connect to the cluster that corresponds to the `hadoop.cluster.hivenode.admin.account` name entered in the `sas_hadoop_config.properties` file.

The script `hadoop_extract.sh` collects necessary files from the Hadoop cluster and stores them in two folders in the unzipped file structure:

pathname/User/SASWorkspace/hadoop/conf

pathname/User/SASWorkspace/hadoop/lib

Any collection issues are documented in the logs in **pathname/Admin/logs**. The script creates a backup of the original `sas_hadoop_config.properties` file.

Copy the complete **User** directory to a directory on a Windows server to which all vApp users have READ access. Inform all vApp users about the location of the **User** directory, which they must copy to their vApp client machines.

Configure the Hadoop Cluster

Complete configuration of the Hadoop cluster as described in [Chapter 12, “Configuring the Hadoop Cluster,” on page 105](#). Provide the necessary configuration values to the vApp user.

Review any additional configuration that might be needed for the SAS Embedded Process, which is part of the In-Database Deployment Package. This is Hadoop distribution dependent. For more information, see [Chapter 5, “Additional Configuration for the SAS Embedded Process,” on page 49](#).

Deactivating or Removing Existing Versions

About Deactivating and Removing

If you are installing a new version or reinstalling a previous version of SAS In-Database Technologies for Hadoop, you must deactivate or remove other existing parcels or stacks after installing the new one. You can have more than one parcel or stack for a particular product on the cluster, but only one can be active. At a minimum, you deactivate parcels or stacks that you do not want to use. Optionally, you can remove them from the cluster after deactivation.

Cloudera Manager

Example Names

Deactivation and removal of the parcels for SAS In-Database Deployment Package, SAS Quality Knowledge Base, and SAS Data Management Accelerator for Spark each follow the same procedure. The parcel names are SASEP, SASQKB, and SASDMSPARK, respectively. These names are represented in the following procedures by *parcel_name*. The configuration filenames for the SAS In-Database Deployment Package and SAS Data Management Accelerator for Spark are ep-config.xml and dmp-config.xml, respectively. The index filename for SAS Quality Knowledge Base is default.idx. These filenames are represented in the following procedures by *file_name*.

Deactivating

To deactivate a parcel using Cloudera Manager, follow these steps:

1. Log on to Cloudera Manager.
2. If running, stop any of the *parcel_name* services:
 - a. On the Home page, click the down arrow next to *parcel_name* service.
 - b. Under *parcel_name* Actions, select Stop, and then click **Stop**.
3. Delete the *parcel_name* service from Cloudera Manager:

Note: If you are deleting more than one service, delete all services before proceeding to the step of deactivation.

 - a. On the Home page, click the down arrow next to *parcel_name* service.
 - b. Click **Delete**. The *parcel_name* service no longer appears on the **Home** ⇒ **Status** tab.
4. Deactivate the *parcel_name* parcel:
 - a. Navigate to the **Hosts** ⇒ **Parcels** tab.
 - b. For *parcel_name*, select **Actions** ⇒ **Deactivate**. You are prompted either to restart the cluster or close the window.
 - c. When prompted, click **Close**.

CAUTION:

Do not restart the cluster.

- d. Click **OK** to continue the deactivation.

Removing

After deactivating the parcel, follow these steps to remove it:

1. Remove the *parcel_name* parcel:
 - a. For *parcel_name*, select **Activate** ⇒ **Remove from Hosts**.
 - b. Click **OK** to confirm.
2. For *parcel_name*, select **Distribute** ⇒ **Delete**.
3. Click **OK** to confirm.

This step deletes the parcel files from the `/opt/cloudera/parcel` directory.

4. Manually remove the *file_name* file:

- a. Log on to HDFS.

```
sudo su - root
su - hdfs | hdfs-userid
```

Note: If your cluster is secured with Kerberos, the HDFS user must have a valid Kerberos ticket to access HDFS. This can be done with kinit.

- b. Navigate to the appropriate directory on HDFS.
 - The directory for SASEP and SASDMSPARK is `/sas/ep/config/`
 - The directory for SASQKB is `/sas/qkb/`
- c. Delete the *file_name* file.

Ambari**Example Names**

Deactivation and removal of the stacks for SAS In-Database Deployment Package, SAS Quality Knowledge Base, and SAS Data Management Accelerator for Spark each follow the same procedure. The stack names are SASEP, SASQKB, and SASDMSPARK, respectively. These names are represented in the following procedures by *stack_service*. The configuration filenames for the SAS In-Database Deployment Package and SAS Data Management Accelerator for Spark are `ep-config.xml` and `dmp-config.xml`, respectively. The index filename for SAS Quality Knowledge Base is `default.idx`. These filenames are represented in the following procedures by *file_name*.

Deactivating

You deactivate a stack by activating another stack.

To deactivate a stack using Ambari, follow these steps:

1. Log on to the Ambari manager. All deployed versions of the *stack_service* stack appear in the left pane of the Home page under the **allversions** text box.
2. Select the *stack_service* stack that you want to activate.
3. Enter the version number of the stack that you want to activate in the **activated_version** text box on the **Configs** tab.
4. Click **Save**.
5. Optionally, add a note describing your action, and then click **OK**.

6. If you are deactivating more than one stack, finish all deactivation tasks before restarting services.
7. Click **Restart** to restart the *stack_service* after you have deactivated all the stacks.
8. Click **Restart All Affected**. The affected services are restarted.
9. The new stack is activated, leaving the previous stack deactivated.
10. If you have deactivated additional stacks, select them and restart all affected services. The new stacks are activated, leaving the previous stacks deactivated.

Removing

Note: Root or passwordless sudo access is required to remove the stack.

After deactivating the stack, follow these steps to remove it:

1. Navigate to the appropriate **Admin/bin/stack** directory, where *stack* represents either **indatabase**, **qkb**, or **dmspark**. These directories are on the Linux machine where SAS In-Database Technologies for Hadoop is downloaded and unzipped.

A `delete_stack.sh` file is in each *stack* directory.

2. Copy the `delete_stack.sh` file to a temporary directory where the cluster manager server is located. Here is an example using secure copy.

```
scp delete_stack.sh user@cluster-manager-host:/mytemp
```

3. Use this command to run the delete script.

```
./delete_stack.sh <Ambari-Admin-User-Name>
```

4. Enter the Ambari administrator password at the prompt.

A message appears that offers options for removal.

5. Enter one of the options:

- Enter 1 to remove only the *file_name* file.
- Enter 2 to remove a specific version of *stack_service*.
- Enter 3 to remove all versions of *stack_service*.

You are prompted to restart the Ambari server to complete the removal of the **SASEP SERVICE**.

6. Enter *y* to restart the Ambari server. The *stack_service* no longer appears.

Chapter 8

Standard Deployment

Getting Started with the Deployment	73
About the Deployment	73
Before Deployment	74
Overview of Deployment Steps	74
Configure Kerberos	74
Create Parcels or Stacks	75
Deploy Files	75
Deploying the Parcels on Cloudera	75
Deploying the Stacks on Hortonworks	75
Collect Files	75
About Collecting Files	75
Configure the Files	76
Inform the vApp Users	77
Configure the Hadoop Cluster	77
Hot Fixes and SAS Notes	77
SAS Notes for SAS Data Loader for Hadoop	77
Required Hot Fixes for Data Loader 2.4	78

Getting Started with the Deployment

About the Deployment

This chapter describes standard deployment of SAS In-Database Technologies for Hadoop. SAS sends an email to a contact person at your business or organization. This email includes instructions for downloading SAS In-Database Technologies for Hadoop to the SAS Software Depot. After downloading the software, you can use the SAS Deployment Manager to deploy it. The conditions under which you use the SAS Deployment Manager and the prerequisites for using it are described in [“When to Deploy the SAS In-Database Deployment Package Using the SAS Deployment Manager”](#) on page 11 and [“Prerequisites for Using the SAS Deployment Manager to Deploy the In-Database Deployment Package”](#) on page 12. Although these topics discuss the SAS Deployment Manager in relation to deploying the SAS In-Database Deployment Package for Hadoop, the prerequisites are the same for the deployment of any component.

If your site does not meet the conditions for using the SAS Deployment Manager, you can deploy the SAS In-Database Technologies for Hadoop components manually. The individual components of SAS In-Database Technologies for Hadoop and the processes for manual installation are discussed in the following chapters: [Chapter 9, “SAS In-Database Deployment Package for Hadoop,”](#) on page 79, [Chapter 10, “SAS In-Database Technologies for Data Quality Directives,”](#) on page 83, and [Chapter 11, “SAS Data Management Accelerator for Spark,”](#) on page 95.

Note: For further specific information about SAS In-Database Technologies for Hadoop, see [Chapter 6, “Introduction,”](#) on page 59.

Before Deployment

If you are installing a new version or reinstalling a previous version of SAS In-Database Technologies for Hadoop, you must deactivate or remove other existing SAS In-Database Technologies for Hadoop parcels or stacks after installing the new one. More than one parcel or stack can be deployed on your cluster, but only one parcel can be activated at a time. See [“Deactivating or Removing Existing Versions”](#) on page 69.

Overview of Deployment Steps

Here are the tasks to be completed during deployment:

1. Configure Kerberos, if appropriate, and then provide required configuration values to the vApp user.
2. Identify a Windows server in a shared network location that is accessible to vApp users.
3. Review the Hadoop Environment topic from the [system requirements](#) for SAS Data Loader 2.4.
4. Create parcels or stacks using SAS Deployment Manager.
5. Deploy services using Cloudera Manager or Ambari.
6. Collect required files from the Hadoop cluster.
7. Make the required vApp directories available on the Windows server in the shared network location.
8. Configure the Hadoop cluster, and then provide required configuration values to the vApp user.
9. Check for SAS Notes and hot fixes that might be available.

Note: If you switch to a different distribution of Hadoop after the initial installation of SAS In-Database Technologies for Hadoop, you must reinstall and reconfigure SAS In-Database Technologies for Hadoop on the new Hadoop cluster.

Configure Kerberos

If you are using Kerberos, you must have all valid tickets in place on the cluster. When deploying SAS In-Database Technologies for Hadoop, the HDFS user must have a valid ticket. See [Chapter 13, “Configuring Kerberos,”](#) on page 111. Provide the necessary configuration values to the vApp user.

Create Parcels or Stacks

To use the SAS Deployment Manager to create the parcel or stack for each SAS In-Database Technologies for Hadoop component, follow the steps described in [“Using the SAS Deployment Manager to Create the SAS Embedded Process Parcel or Stack”](#) on page 18.

After creating the parcel or stack for the SAS Embedded Process, complete the following steps:

1. Return to the **Select SAS Deployment Manager Task** page.
2. Repeat the parcel or stack creation process after selecting **SAS Quality Knowledge Base for Hadoop**. This installs the SAS QKB for Contact Information. It is recommended that you periodically update the QKB. For more information, see [“Updating and Customizing the QKB”](#) on page 89.
3. If you are using Spark, return to the **Select SAS Deployment Manager Task** page.
4. Repeat the parcel or stack creation process after selecting **SAS Data Management Accelerator for Spark**.

Deploy Files

Deploying the Parcels on Cloudera

After you run the SAS Deployment Manager to create the parcels, you must distribute and activate the parcels on the cluster. For this procedure, start with [Step 5 on page 64](#) under [“Deploying the Services to Cloudera”](#) on page 64.

Deploying the Stacks on Hortonworks

After you run the SAS Deployment Manager to create the stacks, you must distribute and activate the stacks on the cluster. For this procedure, start with [Step 6 on page 66](#) under [“Deploying the Services to Hortonworks”](#) on page 65.

Collect Files

About Collecting Files

Certain files must be collected from the Hadoop cluster and made available to the vApp user.

Complete configuration of SAS/ACCESS Interface to Hadoop, as described in *SAS Hadoop Configuration Guide for Base SAS and SAS/ACCESS*. This process collects necessary files in the appropriate folders on the Hadoop cluster:

`installation_path/conf`

installation_path/lib

The conf folder contains the required XML and JSON files for the vApp client. The lib folder contains the required JAR files.

Configure the Files

Copying

The conf and lib folders must be copied to a directory on a Windows server to which all vApp users have READ access.

Edit inventory.json

If the Oozie, Spark, or Impala services are running on the Hadoop Cluster, you must edit the appropriate section of the **conf/inventory.json** file on the Windows server to reflect this. For any service that is available, the “available” parameter must be set to “true.” In addition, the Impala service must specify a host and port, and the Oozie service must specify a URL.

The following example specifies all three services as available:

```
"impala":{
  "available":"true",
  "port": "21050",
  "hosts":["machine1.domain.com","machine2.domain.com"]
},
"spark":{
  "available":"true"
},
"oozie":{
  "available":"true",
  "url":"http://machine1.domain.com:11000/oozie"
},
```

For MapR Users

For MapR deployments only, you must manually create a file named **mapr-user.json** (case-sensitive) that specifies user information that is required by the SAS Data Loader for Hadoop vApp in order for the vApp to interact with the Hadoop cluster. You must supply a user name, user ID, and group ID in this file. The user name must be a valid user on the MapR cluster.

Note: You must add this file to the conf directory that was copied to the Windows server.

To configure user IDs, follow these steps:

1. Create one User ID for each vApp user.
2. Create UNIX user IDs on all nodes of the cluster and assign them to a group.
3. Create the **mapr-user.json** file containing user ID information. You can obtain this information by logging on to a cluster node and running the ID command. You might create a file similar to the following:

```
{
  "user_name"      : "myuser",
  "user_id"        : "2133",
  "user_group_id"  : "2133",
  "take_ownership" : "true"
```

```
}
```

4. Copy `mapr-user.json` to the `conf` directory on the Windows server from which the vApp users copy the `conf` and `lib` directories.

Note: To log on to the MapR Hadoop cluster with a different valid user ID, you must edit the information in the `mapr-user.json` file and in the **User ID** field of the SAS Data Loader for Hadoop Configuration dialog box. See “[User ID](#)” on page 108.

5. Create a user home directory and Hadoop staging directory in MapRFS. The user home directory is `/user/myuser`. The Hadoop staging directory is controlled by the setting `yarn.app.mapreduce.am.staging-dir` in `mapred-site.xml` and defaults to `/user/myuser`.

6. Change the permissions and owner of `/user/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 MapRFS directory (used for Oozie jobs)
 - Read, Write, and Delete permission for tables in Hive
7. SAS Data Loader for Hadoop uses HiveServer2 as its source of tabular data. Ensure that the UNIX user has appropriate permissions on mapRFS for the locations of the Hive tables on which the user is permitted to operate.

Inform the vApp Users

Inform the vApp users that they can copy the `conf` and `lib` folders from the Windows server to the shared folder `SASWorkspace\hadoop` on all active instances of the vApp client. These folders are required for the vApp to connect to Hadoop successfully.

Configure the Hadoop Cluster

Complete configuration of the Hadoop cluster as described in [Chapter 12, “Configuring the Hadoop Cluster,”](#) on page 105. Provide the necessary configuration values to the vApp user.

Review any additional configuration that might be needed for the SAS Embedded Process, which part of the In-Database Deployment Package. This is Hadoop distribution dependent. For more information, see [Chapter 5, “Additional Configuration for the SAS Embedded Process,”](#) on page 49.

Hot Fixes and SAS Notes

SAS Notes for SAS Data Loader for Hadoop

After installing the SAS In-Database Deployment Package for Hadoop, check SAS Notes for any specific issues. For more information, see [Samples & SAS Notes](#).

Required Hot Fixes for Data Loader 2.4

Hot fix V68002 is required for the SAS Embedded Process that is used by SAS Data Loader for Hadoop 2.4. This hot fix must be installed after the SAS In-Database Deployment Package for Hadoop is installed. This hot fix is installed on the Hadoop cluster.

CAUTION:

This hot fix is required only if you used the Standard deployment method. If you used the ZIP file deployment, this hot fix is automatically included in your software deployment.

Note: The specific hot fix that you need to deploy depends on your Hadoop distribution:

- If you used the SAS Deployment Manager to install the SAS In-Database Deployment Package for Hadoop on your Cloudera cluster, use hot fix V68002, which is located here:http://ftp.sas.com/techsup/download/hotfix/HF2/V/V68/V68002/xx/hdl/V68002hl_cloudera.html.
- If you used the SAS Deployment Manager to install the SAS In-Database Deployment Package for Hadoop on your Hortonworks cluster, use hot fix V68002, which is located here:http://ftp.sas.com/techsup/download/hotfix/HF2/V/V68/V68002/xx/hdl/V68002hl_hortonworks.html.

Chapter 9

SAS In-Database Deployment Package for Hadoop

About the SAS In-Database Deployment Package	79
Getting Started	79
About Deployment	79
Before Deployment	80
Overview of Deployment Steps	80
Configure Kerberos	80
Manual Installation	80
Collect Files	81
Install Additional Components	81
Configure the Hadoop Cluster	81
Hot Fixes and SAS Notes	81
SAS Notes for SAS Data Loader for Hadoop	81
Required Hot Fixes for Data Loader 2.4	81

About the SAS In-Database Deployment Package

The SAS In-Database Deployment Package includes the SAS Embedded Process and the SAS Hadoop MapReduce JAR files. The SAS Embedded Process runs within MapReduce to read and write data. You must deploy the SAS In-Database Deployment Package. Deploying and configuring the SAS In-Database Deployment Package needs to be done only once for each Hadoop cluster.

Getting Started

About Deployment

This chapter describes manual deployment of the SAS In-Database Deployment Package. SAS sends an email to a contact person at your business or organization. This email includes instructions for downloading your software to the SAS Software Depot. After downloading the software, you can install it manually. The conditions under which

you install manually are described in [“When to Deploy the SAS In-Database Deployment Package Manually”](#) on page 35.

Before Deployment

If you are installing a new version or reinstalling a previous version of the SAS In-Database Deployment Package, you must first remove the current version. For this procedure, see [“Upgrading from or Reinstalling a Previous Version”](#) on page 37.

Overview of Deployment Steps

Here are the tasks to be completed during deployment:

1. Configure Kerberos, if appropriate, and then provide required configuration values to the vApp user.
2. Identify a Windows server in a shared network location that is accessible to vApp users.
3. Review the Hadoop Environment topic from the [system requirements](#) for SAS Data Loader 2.4.
4. Install the SAS In-Database Deployment Package.
5. Collect required files from the Hadoop cluster.
6. Make the required vApp directories available on the Windows server in the shared network location.
7. Install additional components, if necessary.
8. Configure the Hadoop cluster, and then provide required configuration values to the vApp user.
9. Check for SAS Notes and hot fixes that might be available.

Note: If you switch to a different distribution of Hadoop after the initial installation of SAS In-Database Technologies for Hadoop, you must reinstall and reconfigure SAS In-Database Technologies for Hadoop on the new Hadoop cluster.

Configure Kerberos

If you are using Kerberos, you must have all valid tickets in place on the cluster. When deploying SAS In-Database Technologies for Hadoop, the HDFS user must have a valid ticket. See [Chapter 13, “Configuring Kerberos,”](#) on page 111. Provide the necessary configuration values to the vApp user.

Manual Installation

For the manual installation procedure, see [Chapter 4, “Deploying the In-Database Deployment Package Manually,”](#) beginning with [“Copying the SAS Embedded Process Install Script to the Hadoop Cluster”](#) on page 40.

Collect Files

Certain files must be collected from the Hadoop cluster and made available to the vApp user. For a description of this process, see [“Collect Files” on page 75](#).

Install Additional Components

You must install the SAS In-Database Technologies for Data Quality Directives and, optionally, SAS Data Management Accelerator for Spark if you have not already done so. For more information, see [Chapter 10, “SAS In-Database Technologies for Data Quality Directives,” on page 83](#) and [Chapter 11, “SAS Data Management Accelerator for Spark,” on page 95](#).

Configure the Hadoop Cluster

Complete configuration of the Hadoop cluster as described in [Chapter 12, “Configuring the Hadoop Cluster,” on page 105](#). Provide the necessary configuration values to the vApp user.

Review any additional configuration that might be needed for the SAS Embedded Process, which is part of the In-Database Deployment Package. This is Hadoop distribution dependent. For more information, see [Chapter 5, “Additional Configuration for the SAS Embedded Process,” on page 49](#).

Hot Fixes and SAS Notes

SAS Notes for SAS Data Loader for Hadoop

After installing the SAS In-Database Deployment Package for Hadoop, check SAS Notes for any specific issues. For more information, see [Samples & SAS Notes](#).

Required Hot Fixes for Data Loader 2.4

Hot fix V68002 is required for the SAS Embedded Process that is used by SAS Data Loader for Hadoop 2.4. This hot fix must be installed after the SAS In-Database Deployment Package for Hadoop is installed. This hot fix is installed on the Hadoop cluster.

CAUTION:

This hot fix is required only if you used the manual deployment method.

If you used the manual deployment method to install the on either Cloudera or Hortonworks, use hot fix V68002, which is located here: <http://ftp.sas.com/techsup/download/hotfix/HF2/V/V68/V68002/xx/hdl/V68002hl.html>.

Chapter 10

SAS In-Database Technologies for Data Quality Directives

About SAS In-Database Technologies for Data Quality Directives	83
Getting Started	84
About Deployment	84
Before Deployment	84
Overview of Deployment Steps	84
Configure Kerberos	84
Manual Installation	85
Copying the Scripts	85
SAS Data Quality Accelerator	86
SAS Quality Knowledge Base (QKB)	88
Install Additional Components	94
Configure the Hadoop Cluster	94

About SAS In-Database Technologies for Data Quality Directives

The data quality directives in SAS Data Loader for Hadoop are supported by SAS Data Quality Accelerator and the SAS Quality Knowledge Base (QKB). SAS Data Quality Accelerator is a required component for SAS Data Loader for Hadoop and is included in SAS In-Database Technologies for Hadoop. The QKB, either the SAS QKB for Contact Information or the SAS QKB for Product Data, is a collection of files that store data and logic to support data management operations. A QKB is specific to a locale, that is, to a country and language. SAS Data Loader for Hadoop data quality directives reference the QKB when performing data quality operations on your data. It is recommended that you periodically update the QKB. For more information, see [“Updating and Customizing the QKB” on page 89](#).

Both the SAS Data Quality Accelerator and the SAS Quality Knowledge Base must be deployed in the Hadoop cluster.

Getting Started

About Deployment

This chapter describes manual deployment of SAS Data Quality Accelerator and the SAS QKB. SAS sends an email to a contact person at your business or organization. This email includes instructions for downloading your software to the SAS Software Depot. After downloading the software, you can install it manually. The conditions under which you install manually are described in [“When to Deploy the SAS In-Database Deployment Package Manually” on page 35](#). Although this is a description of the conditions for the SAS In-Database deployment package, they are also valid concerning SAS Data Quality Accelerator and the SAS QKB.

Before Deployment

If you are installing a new version or reinstalling a previous version of SAS Data Quality Accelerator or the SAS QKB, you must first remove the current version. For this procedure, see [“Removing SAS Data Quality Accelerator” on page 86](#) or [“Removing the QKB” on page 92](#).

Overview of Deployment Steps

Here are the tasks to be completed during deployment:

1. Configure Kerberos, if appropriate, and then provide required configuration values to the vApp user.
2. Identify a Windows server in a shared network location that is accessible to vApp users.
3. Review the Hadoop Environment topic from the [system requirements](#) for SAS Data Loader 2.4.
4. Install the SAS In-Database Technologies for Data Quality Directives.
5. Install additional components, if necessary.
6. Configure the Hadoop cluster, and then provide required configuration values to the vApp user.

Note: If you switch to a different distribution of Hadoop after the initial installation of SAS In-Database Technologies for Hadoop, you must reinstall and reconfigure SAS In-Database Technologies for Hadoop on the new Hadoop cluster.

Configure Kerberos

If you are using Kerberos, you must have all valid tickets in place on the cluster. When deploying SAS In-Database Technologies for Hadoop, the HDFS user must have a valid ticket. See [Chapter 13, “Configuring Kerberos,” on page 111](#). Provide the necessary configuration values to the vApp user.

Manual Installation

Copying the Scripts

The SAS Data Quality Accelerator and SAS QKB scripts are contained in a self-extracting archive file named `sepdqacchadp-2.70000-1.sh`. This file is contained in a ZIP file that is located in a directory in your SAS Software Depot. This ZIP file must be copied to the *EPInstallDir* that was created during the installation of the SAS In-Database Deployment Package, as described in [“Creating the SAS Embedded Process Directory” on page 40](#).

To copy the ZIP file to the *EPInstallDir* on your Hadoop master node, follow these steps:

1. Navigate to the **YourSASDepot/standalone_installs** directory.
This directory was created when your SAS Software Depot was created by the SAS Download Manager.
2. Locate the `en_sasexe.zip` file. This file is in the following directory:
**YourSASDepot/standalone_installs/
SAS_Data_Quality_Accelerator_Embedded_Process_Package_for_Hadoop/2_7/Hadoop_on_Linux_x64.**
The `sepdqacchadp-2.70000-1.sh` file is included in this ZIP file.
3. Unzip the ZIP file on the client.

```
unzip en_sasexe.zip
```


The ZIP file contains one file: `sepdqacchadp-2.70000-1.sh`.
4. Copy the `sepdqacchadp-2.70000-1.sh` file to the *EPInstallDir* directory on the Hadoop master node (NameNode). The following example uses secure copy:

```
scp sepdqacchadp-2.70000-1.sh username@hdpclus1:/EPInstallDir
```
5. Log on to the Hadoop NameNode as root. Then, execute the following command from the *EPInstallDir* directory:

```
./sepdqacchadp-2.70000-1.sh
```

This command creates the following files in *EPInstallDir/sasexe/SASEPHome/bin* of the Hadoop NameNode:

- `dq_install.sh`
- `dq_uninstall.sh`
- `dq_env.sh`
- `qkb_push.sh`

The `dq_install.sh` script enables you to deploy SAS Data Quality Accelerator files to the cluster nodes. See [“Installing SAS Data Quality Accelerator” on page 86](#).

The `dq_uninstall.sh` script enables you to remove SAS Data Quality Accelerator files from the cluster nodes. See [“Removing SAS Data Quality Accelerator” on page 86](#).

The `dq_env.sh` script is a utility script that is used by the other scripts.

The `qkb_push.sh` script enables you to deploy or remove the QKB to or from the cluster nodes. Before you can use `qkb_push.sh`, you must copy a QKB to the Hadoop master node. See [“Installing the QKB” on page 91](#).

SAS Data Quality Accelerator

Installing SAS Data Quality Accelerator

To deploy SAS Data Quality Accelerator binaries to the cluster, run `dq_install.sh`. You must run `dq_install.sh` as the root user.

Run `dq_install.sh` as follows:

```
cd EPInstallDir/sasexe/SASEPHome/bin
./dq_install.sh
```

The `dq_install.sh` file automatically discovers all nodes of the cluster by default and deploys the SAS Data Quality Accelerator files to those nodes. Use the `-h` or `-f` arguments to specify deploying the files to a specific node or group of nodes.

By default, `dq_install.sh` does not list the names of the host nodes to which it deploys the files. To create such a list, include the `-v` argument in the command.

For information about supported arguments, see [“DQ_INSTALL.SH and DQ_UNINSTALL.SH Syntax” on page 87](#).

The `dq_install.sh` script creates the following files on each node on which it is executed:

```
EPInstallDir/bin/dq_install.sh
EPInstallDir/bin/dq_install.sh
EPInstallDir/bin/qkb_push.sh
EPInstallDir/bin/dq_env.sh
EPInstallDir/jars/sas.tools.qkb.hadoop.jar
EPInstallDir/sasexe/tkeblufn.so
EPInstallDir/sasexe/t0w7zt.so
EPInstallDir/sasexe/t0w7zh.so
EPInstallDir/sasexe/t0w7ko.so
EPInstallDir/sasexe/t0w7ja.so
EPInstallDir/sasexe/t0w7fr.so
EPInstallDir/sasexe/t0w7en.so
EPInstallDir/sasexe/d2dqtokens.so
EPInstallDir/sasexe/d2dqlocales.so
EPInstallDir/sasexe/d2dqdefns.so
EPInstallDir/sasexe/d2dq.so
```

Verify that these files have been copied to the nodes.

Removing SAS Data Quality Accelerator

To remove SAS Data Quality Accelerator binaries from the cluster, run `dq_uninstall.sh`. You must run `dq_uninstall.sh` as the root user.

Note:

- If you are removing the QKB, you must do so before removing the binaries. Removing the binaries removes the `qkb_push.sh` file that is used to remove the QKB. Running `dq_uninstall.sh` does not remove the QKB from the cluster. Instructions for removing the QKB are found in [“Removing the QKB” on page 92](#).

- This step is not necessary for Cloudera and Hortonworks distributions where SAS In-Database Technologies for Hadoop were installed through the SAS Deployment Manager.

Run `dq_uninstall.sh` as follows:

```
cd $EPInstallDir/sasexe/SASEPHome/bin
./dq_uninstall.sh
```

The `dq_uninstall.sh` file automatically discovers all nodes of the cluster by default and removes the SAS Data Quality Accelerator files from those nodes. Use the `-h` or `-f` arguments to specify removing the files from a specific node or group of nodes.

By default, `dq_uninstall.sh` does not list the names of the host nodes from which it removes the files. To create such a list, include the `-v` argument in the command.

For information about supported arguments, see “[DQ_INSTALL.SH and DQ_UNINSTALL.SH Syntax](#)” on page 87.

DQ_INSTALL.SH and DQ_UNINSTALL.SH Syntax

dq_install.sh

`<-?>`

`<-l logfile>`

`<-f hostfile>`

`<-h hostname>`

`<-v >`

dq_uninstall.sh

`<-?>`

`<-l logfile>`

`<-f hostfile>`

`<-h hostname>`

`<-v >`

Arguments

-?

prints usage information.

-l logfile

directs status information to the specified log file instead of to standard output.

-f hostfile

specifies the full path of a file that contains the list of hosts where SAS Data Quality Accelerator is installed or removed.

Default	The script discovers the cluster topology and uses the retrieved list of data nodes.
----------------	--

Note	The <code>-f</code> and <code>-h</code> arguments are mutually exclusive.
-------------	---

Example	<code>-f /etc/hadoop/conf/slaves</code>
----------------	---

-h hostname <hostname>

specifies the target host or host list where SAS Data Quality Accelerator is installed or removed.

Default	The script discovers the cluster topology and uses the retrieved list of data nodes.
Requirement	If you specify more than one host, the host names must be separated by spaces.
Note	The -f and -h arguments are mutually exclusive.
Tip	Use the -host argument when new nodes are added to the cluster.
Example	<pre>-h server1 server2 server3 -h bluesvr</pre>

-v
specifies verbose output, which lists the names of the nodes on which the script ran.

SAS Quality Knowledge Base (QKB)

Obtaining a QKB

You can obtain a QKB in one of the following ways:

- Run the SAS Deployment Wizard, which is part of your SAS order. In the Select Products to Install dialog box, select the check box for SAS Quality Knowledge Base. This installs the SAS QKB for Contact Information.

Note:

- This option applies only to the SAS QKB for Contact Information. For step-by-step guidance on installing a QKB using the SAS Deployment Wizard, see [SAS Quality Knowledge Base for Contact Information: Installation and Configuration Guide](#).
- If you did not select the check box for SAS Quality Knowledge Base when you initially ran the SAS Deployment Wizard, you can run it again. In the Select Products to Install dialog box, deselect all check boxes, and then select the check box for SAS Quality Knowledge Base.
- Download a QKB from the SAS downloads site. You can download either the SAS QKB for Contact Information or the SAS QKB for Product Data.

Note: You must have a SAS profile for this option.

1. Open the [SAS Downloads site](#).
 2. Select the appropriate QKB.
 3. When prompted, log on to your SAS profile or create a new profile.
 4. Complete downloading and installing the QKB.
- Copy a QKB that you already use with other SAS software in your enterprise.
For more information, see [“Copying the QKB to the Hadoop NameNode”](#) on page 89.

After your initial deployment, periodically update the QKB in your Hadoop cluster to ensure that you are using the latest QKB updates provided by SAS.

Updating and Customizing the QKB

SAS provides regular updates to the QKB. It is recommended that you update your QKB each time that a new one is released. For a listing of the latest enhancements to the QKB, see the What's New document 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. After obtaining the new QKB, copy it to the Hadoop NameNode (See [“Copying the QKB to the Hadoop NameNode” on page 89](#)) and use the same steps that you would to deploy a standard QKB.

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 on licensing DataFlux Data Management Studio.

If you want to customize your QKB, it is recommended that you customize your QKB on a local workstation, and then copy the customized QKB to the Hadoop NameNode for deployment. When updates to the QKB are required, merge your customizations into an updated QKB locally, and copy the updated, customized QKB to the Hadoop NameNode (See [“Copying the QKB to the Hadoop NameNode” on page 89](#)) for deployment. This enables you to deploy a customized QKB to the Hadoop cluster using the same steps that you would to deploy a standard QKB. Copying your customized QKB from a local workstation also means that you have a backup of the QKB on your local workstation. See 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.

Kerberos Security Requirements

A Kerberos ticket (TGT) is required to deploy the QKB in a Kerberos environment.

To create the ticket, follow these steps:

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

The following is an example of commands used to obtain the ticket.

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

Copying the QKB to the Hadoop NameNode

After you have obtained a QKB (see [“SAS Quality Knowledge Base \(QKB\)” on page 88](#)), you must copy it to the Hadoop NameNode. Copy the QKB to a temporary staging area, such as `/tmp/qkbstage`.

You can copy the QKB to the Hadoop NameNode by using a file transfer command like FTP or SCP, or by mounting the file system where the QKB is located on the Hadoop NameNode. You must copy the complete QKB directory structure.

SAS installation tools typically create a QKB in the following locations, where *qkb_product* is the QKB product name and *qkb_version* is the QKB version number:

- Windows 7: **C:\ProgramData\SAS\QKB\qkb_product\qkb_version.**

For example:

```
C:\ProgramData\SAS\QKB\CI\26
```

Note: ProgramData is a hidden location.

- UNIX and Linux: **/opt/sas/qkb/qkb_product/qkb_version.**

For example:

```
/opt/sas/qkb/ci/26
```

The following example shows how you might copy a QKB that exists on a Linux system to the Hadoop NameNode. The example uses secure copy with the **-r** argument to recursively copy the specified directory and its subdirectories.

- Assume that **hmaster456** is the host name of the Hadoop NameNode.
- The target location on the NameNode is **/tmp/qkbstage**

To copy the QKB from the client desktop, issue the command:

```
scp -r /opt/sas/qkb/ci/26 hmaster456:/tmp/qkbstage
```

Overview of the QKB_PUSH.SH Script

The `qkb_push.sh` script enables you to perform the following actions.

- Install or remove SAS QKB files on a single node or a group of nodes.
- Generate a SAS QKB index file and write the file to an HDFS location.
- Write the installation or removal output to a log file.

The `qkb_push.sh` file is created in the **EPInstallDir/sasexe/SASEPHome/bin** directory. You must execute `qkb_push.sh` from this directory.

Note: If you used SAS Deployment Manager or the zip file method of deploying SAS In-Database Technologies for Hadoop, the `qkb_push.sh` file is located in the **EPInstallDir/SASEPHome/bin** directory.

You can also use `qkb_push.sh` to deploy updated versions of the QKB. For more information, see [“Updating and Customizing the QKB” on page 89](#).

You suppress index creation or perform only index creation by using the **-i** and **-x** arguments. If users have a problem viewing QKB definitions from within SAS Data Loader, you might want to re-create the index file.

Note: Only one QKB and one index file are supported in the Hadoop framework at a time. For example, you cannot have a QKB for Contact Information and a QKB for Product Data in the Hadoop framework at the same time. Subsequent QKB and index pushes replace prior ones, unless you are pushing a QKB that is an earlier version than the one installed or has a different name. In these cases, you must remove the old QKB from the cluster before deploying the new one.

The QKB source directory is copied to the fixed location **/opt/qkb/default** on each node. The QKB index file is created in the **/sas/qkb** directory in HDFS. If a QKB or QKB index file already exists in the target location, the new QKB or QKB index file overwrites it.

Installing the QKB

Installing the QKB on the Hadoop cluster nodes performs the following two tasks:

- copies the specified QKB directory to a fixed location (`/opt/qkb/default`) on each of the Hadoop nodes.

Note: Each Hadoop node requires approximately 8 GB of disk space for the QKB.

- 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.

Note: Creating the index file requires special permissions in a Kerberos security environment. These permissions must be configured before deploying the QKB. See [“Kerberos Security Requirements” on page 89](#).

To deploy the QKB to the cluster, run `qkb_push.sh`. You must run `qkb_push.sh` as the root user.

Run `qkb_push.sh` as follows:

```
cd EPInstallDir/sasexe/SASEPHome/bin
./qkb_push.sh qkb_path
```

where `qkb_path` is the name of the directory on the NameNode to which you copied the QKB. For example, you might use the following:

```
./qkb_push.sh /tmp/qkbstage/version
```

The `qkb_push.sh` script automatically discovers all nodes of the cluster by default and deploys the QKB to those nodes. Use the `-h` or `-f` arguments to specify deploying the files to a specific node or group of nodes.

By default, `qkb_push.sh` does not list the names of the host nodes to which it deploys the files. To create such a list, include the `-v` argument in the command. If a name other than the default was configured for the HDFS or MAPR user name, include the `-s` argument in the command.

For information about supported arguments, see [“QKB_PUSH.SH Syntax” on page 92](#).

The `qkb_push.sh` script creates the following directories and files on each node on which it is executed:

```
EPInstallDir/opt/qkb/default/chopinfo
opt/qkb/default/dfx.meta
opt/qkb/default/grammar
opt/qkb/default/inst.meta
opt/qkb/default/locale
opt/qkb/default/phonetx
opt/qkb/default/regexlib
opt/qkb/default/scheme
opt/qkb/default/upgrade.40
opt/qkb/default/vocab
```

Verify that these directories and files have been copied to the nodes.

Check that the `default.idx` file was created in HDFS or MAPR by issuing the command:

```
hadoop fs -ls /sas/qkb
```

Removing the QKB

The QKB can be removed from the Hadoop cluster by executing the `qkb_push.sh` executable file with the `-r` argument. You must have root access to execute `qkb_push.sh`.

Note: If you are removing the entire in-database deployment, you must remove the QKB first.

Run `qkb_push.sh` as follows:

```
cd EPInstallDir/sasexe/SASEPHome/bin
./qkb_push.sh -r
```

The `-r` argument automatically discovers all nodes of the cluster by default and removes the QKB files from those nodes. Use the `-h` or `-f` arguments to specify removing the files from a specific node or group of nodes.

Note: The QKB index file is not removed from HDFS when the `-h` or `-f` argument is specified with `-r`.

By default, the `-r` argument does not list the names of the host nodes from which it removes the files. To create such a list, include the `-v` argument in the command.

For information about supported arguments, see [“QKB_PUSH.SH Syntax” on page 92](#).

QKB_PUSH.SH Syntax

qkb_push.sh <arguments> *qkb_path*

<-?>

<-l logfile>

<-f hostfile>

<-h hostname>

<-v >

<-s user-id>

<-i >

<-x >

<-r >

Arguments

-?

prints usage information.

-l logfile

directs status information to the specified log file instead of to standard output.

-f hostfile

specifies the full path of a file that contains the list of hosts where the QKB is installed or removed.

Default	The <code>qkb_push.sh</code> script discovers the cluster topology and uses the retrieved list of data nodes.
----------------	---

Interaction	Use the <code>-f</code> argument in conjunction with the <code>-r</code> argument to remove the QKB from specific nodes.
--------------------	--

Note	The <code>-f</code> and <code>-h</code> arguments are mutually exclusive.
-------------	---

See	“-r” on page 93
------------	---------------------------------

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

-h *hostname* < *hostname*>

specifies the target host or host list where the QKB is installed or removed.

Default The `qkb_push.sh` script discovers the cluster topology and uses the retrieved list of data nodes.

Requirement If you specify more than one host, the host names must be separated by spaces.

Interaction Use the `-h` argument in conjunction with the `-r` argument to remove the QKB from specific nodes.

Note The `-f` and `-h` arguments are mutually exclusive.

Tip Use the `-host` argument when new nodes are added to the cluster

See [“-r” on page 93](#)

Example `-h server1 server2 server3`
 `-h bluesvr`

-v

specifies verbose output, which lists the names of the nodes on which the script ran.

-s *user-id*

specifies the user ID that has Write access to the HDFS root directory when the default user name is not used.

Defaults `hdfs` for all Hadoop distributions except MapR
 `mapr` for MapR

-i

creates and pushes the QKB index only.

-x

suppresses QKB index creation.

-r

removes the QKB from the Hadoop nodes and it removes the QKB index file from HDFS.

Default The `-r` argument discovers the cluster topology and uses the retrieved list of data nodes.

Interaction You can specify the hosts from which you want to remove the QKB by using the `-f` or `-h` arguments. The `-f` and `-h` arguments are mutually exclusive.

Note The QKB index file is not removed from HDFS when the `-h` or `-f` argument is specified in conjunction with `-r`.

See [“-f *hostfile*” on page 92](#)

[“-h *hostname hostname*” on page 93](#)

Example `-r -h server1 server2 server3`

```
-r -f /etc/hadoop/conf/slaves  
-r -l logfile
```

Install Additional Components

You must install the In-Database Deployment Package and, optionally, SAS Data Management Accelerator for Spark if you have not already done so. For more information, see [Chapter 9, “SAS In-Database Deployment Package for Hadoop,”](#) on [page 79](#) and [Chapter 11, “SAS Data Management Accelerator for Spark,”](#) on [page 95](#).

Configure the Hadoop Cluster

Complete configuration of the Hadoop cluster as described in [Chapter 12, “Configuring the Hadoop Cluster,”](#) on [page 105](#). Provide the necessary configuration values to the vApp user.

Chapter 11

SAS Data Management Accelerator for Spark

About SAS Data Management Accelerator for Spark	95
Getting Started	95
About Deployment	95
Before Deploying	96
Overview of Deployment Steps	96
Configure Kerberos	96
Manual Installation	97
Creating the SAS Data Management Accelerator for Spark Directory	97
Copying the SAS Data Management Accelerator for Spark Install Script	97
Installing SAS Data Management Accelerator for Spark	97
Overview of the SASDMP_ADMIN.SH Script	100
SASDMP_ADMIN.SH Syntax	100
Install Additional Components	103
Configure the Hadoop Cluster	103

About SAS Data Management Accelerator for Spark

Spark is a processing engine that is compatible with Hadoop data. SAS Data Management Accelerator for Spark runs data integration and data quality tasks in a Spark environment. These tasks include mapping columns, summarizing columns, performing data quality tasks such as clustering and survivorship, and standardization of data.

Getting Started

About Deployment

This chapter describes manual deployment of SAS Data Management Accelerator for Spark. SAS sends an email to a contact person at your business or organization. This email includes instructions for downloading your software to the SAS Software Depot.

After downloading the software, you can install it manually. The conditions under which you install manually are described in [“When to Deploy the SAS In-Database Deployment Package Manually” on page 35](#). Although this is a description of the conditions for the SAS In-Database deployment package, they are also valid concerning SAS Data Management Accelerator for Spark.

Note: Deploy SAS Data Management Accelerator for Spark only if Spark is available on the cluster.

Before Deploying

If you are installing a new version or reinstalling a previous version of SAS Data Management Accelerator for Spark, you must first remove the current version. For this procedure, see [“-remove -keepconfig” on page 102](#) under [“SASDMP_ADMIN.SH Syntax” on page 100](#).

Overview of Deployment Steps

Here are the tasks to be completed during deployment:

1. Configure Kerberos, if appropriate, and then provide required configuration values to the vApp user.
2. Identify a Windows server in a shared network location that is accessible to vApp users.
3. Review the Hadoop Environment topic from the [system requirements](#) for SAS Data Loader 2.4.
4. Install SAS Data Management Accelerator for Spark.
5. Install additional components, if necessary.
6. Configure the Hadoop cluster, and then provide required configuration values to the vApp user.

Note: If you switch to a different distribution of Hadoop after the initial installation of SAS In-Database Technologies for Hadoop, you must reinstall and reconfigure SAS In-Database Technologies for Hadoop on the new Hadoop cluster.

Configure Kerberos

If you are using Kerberos, you must have all valid tickets in place on the cluster. When deploying SAS In-Database Technologies for Hadoop, the HDFS user must have a valid ticket. See [Chapter 13, “Configuring Kerberos,” on page 111](#). Provide the necessary configuration values to the vApp user.

Manual Installation

Creating the SAS Data Management Accelerator for Spark Directory

Create a new directory on the Hadoop master node that is not part of an existing directory structure, such as `/sasdmp`.

This path is created on each node in the Hadoop cluster during the SAS Data Management Accelerator for Spark installation. Do not use existing system directories such as `/opt` or `/usr`. This new directory is referred to as *DMPInstallDir* throughout this section.

Copying the SAS Data Management Accelerator for Spark Install Script

The SAS Data Management Accelerator for Spark install script is contained in a self-extracting archive file named `sasdmp_admin.sh`. This file is contained in a ZIP file that is located in a directory in your SAS Software Depot.

To copy the ZIP file to the *DMPInstallDir* on your Hadoop master node, follow these steps:

1. Navigate to the `YourSASDepot/standalone_installs` directory.

This directory was created when your SAS Software Depot was created by the SAS Download Manager.

2. Locate the `en_sasexe.zip` file. This file is in the following directory:
`YourSASDepot/standalone_installs/
 SAS_Data_Management_Accelerator_for_Spark/2_4/
 Hadoop_on_Linux_x64`.

The `sasdmp_admin.sh` file is included in this ZIP file.

3. Log on to the cluster using SSH with sudo access.

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

4. Copy the `en_sasexe.zip` file from the client to the *DMPInstallDir* on the cluster. The following example uses secure copy:

```
scp en_sasexe.zip username@hdpclus1: /DMPInstallDir
```

Note: The *DMPInstallDir* location becomes the SAS Data Management Accelerator for Spark home.

Installing SAS Data Management Accelerator for Spark

To install SAS Data Management Accelerator for Spark, follow these steps:

Note: Permissions are required to install SAS Data Management Accelerator for Spark. For more information, see [“Hadoop Permissions” on page 9](#).

1. Navigate to the location on your Hadoop master node where you copied the `en_sasexe.zip` file.

```
cd /DMPInstallDir
```

2. Ensure that both the *DMPInstallDir* folder and the *en_sasexe.zip* file have Read, Write, and Execute permissions (**chmod 777**).
3. Unzip the *en_sasexe.zip* file.

```
unzip en_sasexe.zip
```

After the file is unzipped, a **sasexe** directory is created in the same location as the *en_sasexe.zip* file. The *dmsprkhdp-2.40000-1.sh* file is located in the **sasexe** directory.

```
DMPInstallDir/sasexe/dmsprkhdp-2.40000-1.sh
```

4. Use the following command to unpack the *dmsprkhdp-2.40000-1.sh* file.

```
./dmsprkhdp-2.40000-1.sh
```

After this script is run and the files are unpacked, the script creates the following directory structure:

```
DMPInstallDir/sasexe/SASDMPHome
```

```
DMPInstallDir/sasexe/dmsprkhdp-2.40000-1.sh
```

Note: During the install process, the *dmsprkhdp-2.40000-1.sh* is copied to all data nodes. Do not remove or move this file from the *DMPInstallDir/sasexe* directory.

The SASDMPHome directory structure looks like this.

```
DMPInstallDir/sasexe/SASDMPHome/bin
```

```
DMPInstallDir/sasexe/SASDMPHome/dat
```

```
DMPInstallDir/sasexe/SASDMPHome/etc
```

```
DMPInstallDir/sasexe/SASDMPHome/lib
```

```
DMPInstallDir/sasexe/SASDMPHome/share
```

```
DMPInstallDir/sasexe/SASDMPHome/var
```

The *DMPInstallDir/sasexe/SASDMPHome/bin* directory looks like this.

```
DMPInstallDir/sasexe/SASDMPHome/bin/dfwsvc
```

```
DMPInstallDir/sasexe/SASDMPHome/bin/dfxver
```

```
DMPInstallDir/sasexe/SASDMPHome/bin/dfxver.bin
```

```
DMPInstallDir/sasexe/SASDMPHome/bin/sasdm_admin.sh
```

```
DMPInstallDir/sasexe/SASDMPHome/bin/settings.sh
```

```
DMPInstallDir/sasexe/SASDMPHome/bin/dmpsvc
```

5. Use the *sasdm_admin.sh* script to deploy the SAS Data Management Accelerator for Spark installation across all nodes.

TIP Many options are available for installing SAS Data Management Accelerator for Spark. Review the script syntax before running it. For more information, see [“Overview of the SASDMP_ADMIN.SH Script” on page 100](#).

Note: If your cluster is secured with Kerberos, complete both steps a and b. If your cluster is not secured with Kerberos, complete only step b.

- a. If your cluster is secured with Kerberos, the HDFS user must have a valid Kerberos ticket to access HDFS. This can be done with *kinit*.

```
sudo su - root
```

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

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

```
exit
```


Note: For all Hadoop distributions except MapR, the default HDFS user is **hdfs**. For MapR distributions, the default MapR superuser is **mapr**. You can specify a different user ID with the **-hdfsuser** argument when you run the **bin/sasdmp_admin.sh -add** script.

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

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

Valid starting    Expires          Service principal
06/20/15 09:51:26 06/27/15 09:51:26 krbtgt/HOST.COMPANY.COM@HOST.COMPANY.COM
renew until 06/22/15 09:51:26
```

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

```
cd DMPInstallDir/SASDMPHome/
bin/sasdmp_admin.sh -genconfig
bin/sasdmp_admin.sh -add
```

TIP Many options are available when installing SAS Data Management Accelerator for Spark. Review the script syntax before running it. For more information, see [“Overview of the SASDMP_ADMIN.SH Script” on page 100](#).

Note: By default, the SAS Data Management Accelerator for Spark install script (**sasdmp_admin.sh**) discovers the cluster topology and installs SAS Data Management Accelerator for Spark 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 add SAS Data Management Accelerator for Spark to new nodes at a later time, you should run the **sasdmp_admin.sh** script with the **-host <hosts>** option.

6. Verify that SAS Data Management Accelerator for Spark is installed by running the **sasdmp_admin.sh** script with the **-check** option.

```
cd DMPInstallDir/SASDMPHome/bin/
bin/sasdmp_admin.sh -check
```

This command checks whether SAS Data Management Accelerator for Spark is installed on all data nodes.

Note: The **sasdmp_admin.sh -check** script does not run successfully if SAS Data Management Accelerator for Spark is not installed.

7. Verify that the configuration file, **dmp-config.xml**, was written to the HDFS file system.

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

Note: If your cluster is secured with Kerberos, you need a valid Kerberos ticket to access HDFS. If not, you can use the WebHDFS browser.

Note: The **/sas/ep/config** directory is created automatically when you run the install script. If you used **-dmpconfig** or **-genconfig** to specify a non-default location, use that location to find the **dmp-config.xml** file.

Overview of the SASDMP_ADMIN.SH Script

The sasdmp_admin.sh script enables you to perform the following actions.

- Install or uninstall SAS Data Management Accelerator for Spark on a single node or a group of nodes.
- Check if SAS Data Management Accelerator for Spark is installed correctly.
- Generate a SAS Data Management Accelerator for Spark configuration file and write the file to an HDFS location.
- Write the installation output to a log file.
- Display all live data nodes on the cluster.
- Display the Hadoop configuration environment.

Note: You must have sudo access on the master node only to run the sasdmp_admin.sh script. You must also have SSH set up in such a way that the master node can passwordless SSH to all data nodes on the cluster where SAS Data Management Accelerator for Spark is installed.

SASDMP_ADMIN.SH Syntax

sasdmp_admin.sh

```
-add <-dmpconfig config-filename> <-maxscp number-of-copies>
    <-hostfile host-list-filename | -host <">host-list<">>
    <-hdfsuser user-id> <-log filename>
```

sasdmp_admin.sh

```
-remove <-dmpconfig config-filename> <-hostfile host-list-filename | -host <">host-list<">>
    <-hdfsuser user-id> <-log filename><-keepconfig>
```

sasdmp_admin.sh

```
<-genconfig config-filename <-force>>
<-check> <-hostfile host-list-filename | -host <">host-list<">>
<-env>
<-hadoopversion>
<-hotfix>
<-log filename>
<-nodelist>
<-sparkversion>
<-validate>
<-version>
```

Arguments

-add

installs SAS Data Management Accelerator for Spark.

Tip If at a later time you add nodes to the cluster, you can specify the hosts on which you want to install SAS Data Management Accelerator for Spark by using the -hostfile or -host option. The -hostfile and -host options are mutually exclusive.

See [-hostfile and -host option on page 101](#)

-dmpconfig *config-filename*

generates the SAS Data Management Accelerator for Spark configuration file in the specified location.

Default /sas/ep/config/dmp-config.xml

Interaction Use the -dmpconfig argument in conjunction with the -add or -remove argument to specify the HDFS location of the configuration file. Use the -genconfig argument when you upgrade to a new version of your Hadoop distribution.

Tip Use the -dmpconfig argument to create the configuration file in a non-default location.

See [“-genconfig *config-filename* -force” on page 102](#)

-maxscp *number-of-copies*

specifies the maximum number of parallel copies between the master and data nodes.

Default 10

Interaction Use this argument in conjunction with the -add argument.

-hostfile *host-list-filename*

specifies the full path of a file that contains the list of hosts where SAS Data Management Accelerator for Spark is installed or removed.

Default The sasdmp_admin.sh script discovers the cluster topology and uses the retrieved list of data nodes.

Interaction Use the -hostfile argument in conjunction with the -add when new nodes are added to the cluster.

Tip You can also assign a host list filename to a UNIX variable, **SASEP_HOSTS_FILE**.
export SASEP_HOSTS_FILE=/etc/hadoop/conf/slaves

See [“-hdfsuser *user-id*” on page 102](#)

Example -hostfile /etc/hadoop/conf/slaves

-host <">*host-list*<">

specifies the target host or host list where SAS Data Management Accelerator for Spark is installed or removed.

Default The sasdmp_admin.sh script discovers 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.

Interaction Use the -host argument in conjunction with the -add when new nodes are added to the cluster.

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

```
export SASEP_HOSTS="server1 server2 server3"
```

See [“-hdfsuser *user-id*” on page 102](#)

Example

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

-hdfsuser *user-id*

specifies the user ID that has Write access to the HDFS root directory.

Defaults hdfs for Cloudera, Hortonworks, Pivotal HD, and IBM BigInsights
mapr for MapR

Interaction Use the -hdfsuser argument in conjunction with the -add or -remove argument to change or remove the HDFS user ID.

Note The user ID is used to copy the SAS Data Management Accelerator for Spark configuration files to HDFS.

-log *filename*

writes the installation output to the specified filename.

Interaction Use the -log argument in conjunction with the -add or -remove argument to write or remove the installation output file.

-remove <-keepconfig>

removes SAS Data Management Accelerator for Spark.

Tips You can specify the hosts for which you want to remove SAS Data Management Accelerator for Spark by using the -hostfile or -host option. The -hostfile or -host options are mutually exclusive.

This argument removes the generated dmp-config.xml file. Use the -keepconfig argument to retain the existing configuration file.

See [-hostfile and -host option on page 101](#)

-genconfig *config-filename* <-force>

generates a new SAS Data Management Accelerator for Spark configuration file in the specified location.

Default /sas/ep/config/dmp-config.xml

Interaction Use the -dmpconfig argument in conjunction with the -add or -remove argument to specify the HDFS location of the configuration file. Use the -genconfig argument when you upgrade to a new version of your Hadoop distribution.

Tip This argument generates an updated dmp-config.xml file. Use the -force argument to overwrite the existing configuration file.

See [“-dmpconfig *config-filename*” on page 101](#)

-check

checks if SAS Data Management Accelerator for Spark is installed correctly on all data nodes.

- env**
displays the Hadoop configuration environment.
- hadoopversion**
displays the Hadoop version information for the cluster.
- hotfix**
installs a hotfix on an existing SAS Data Management Accelerator for Spark installation.
- nodelist**
displays all live DataNodes on the cluster.
- sparkversion**
displays the Spark version information for the cluster.
- validate**
validates the install by executing simple Spark and MapReduce jobs.
- version**
displays the version of SAS Data Management Accelerator for Spark that is installed.

Install Additional Components

You must install the In-Database Deployment Package and SAS In-Database Technologies for Data Quality Directives if you have not already done so. For more information, see [Chapter 9, “SAS In-Database Deployment Package for Hadoop,”](#) on page 79 and [Chapter 10, “SAS In-Database Technologies for Data Quality Directives,”](#) on page 83.

Configure the Hadoop Cluster

Complete configuration of the Hadoop cluster as described in [Chapter 12, “Configuring the Hadoop Cluster,”](#) on page 105. Provide the necessary configuration values to the vApp user.

Chapter 12

Configuring the Hadoop Cluster

Configuring Components on the Cluster	105
Overview	105
SQOOP and OOZIE	105
JDBC Drivers	106
Spark Bin Directory Required in the Hadoop PATH	107
User IDs	107
Configuration Values	108
Providing vApp User Configuration Information	108

Configuring Components on the Cluster

Overview

After deploying the in-database deployment package, you must configure several components and settings on the Hadoop cluster in order for SAS Data Loader for Hadoop to operate correctly. These components and settings are explained in the following topics:

- [“SQOOP and OOZIE” on page 105](#)
- [“JDBC Drivers” on page 106](#)
- [“Spark Bin Directory Required in the Hadoop PATH” on page 107](#)
- [“User IDs” on page 107](#)
- [“Configuration Values” on page 108](#)

SQOOP and OOZIE

Your Hadoop cluster must be configured to use OOZIE scripts.

Note: Ensure that Oozie 4.0 or later is installed. You must add the following as entries in the list for the `oozie.service.SchemaService.wf.ext.schemas` property:

- `sqoop-action-0.4.xsd`
- `hive-action-0.3.xsd`
- `oozie-workflow-0.4.xsd`

- shell-action-0.3.xsd (for Spark submission)

JDBC Drivers

SAS Data Loader for Hadoop leverages the SQOOP and OOZIE components installed with the Hadoop cluster to move data to and from a DBMS. The SAS Data Loader for Hadoop vApp client also accesses 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 or drivers required by the DBMSs that users need to access.

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

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

Table 12.1 JDBC Files

Database	Required Files
Oracle	ojdbc6.jar
Teradata	tdgssconfig.jar and terajdbc4.jar <i>Note:</i> You must also download the Teradata connector JAR file that is matched to your cluster distribution, if available.

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.

The default directories in the Hadoop file system are as follows:

- Hortonworks, Pivotal HD, IBM BigInsights Hadoop clusters: `/user/oozie/share/lib/lib_version/sqoop`
- Cloudera Hadoop clusters: `/user/oozie/share/lib/sharelibversion/sqoop`
- MapR Hadoop clusters: `/oozie/share/lib/sqoop`

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 refresh sharelib, as follows:

```
oozie admin -oozie oozie_url -sharelibupdate
```

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.

Spark Bin Directory Required in the Hadoop PATH

SAS Data Loader for Hadoop supports the Apache Spark cluster computing framework. Spark support requires the addition of the Spark bin directory to the PATH environment variable on each Hadoop node.

Most Hadoop distributions include the Spark bin directory in **/usr/bin**. In some distributions, such as MapR, the Spark bin directory is not included by default in the PATH variable. You must add a line to `yarn-env.sh` on each nodemanager node. The following example illustrates a typical addition to `yarn-env.sh`:

```
/* In MapR 5.0, using Spark 1.3.1 */
export PATH=$PATH:/opt/mapr/spark/spark-1.3.1/bin
```

You can use the command **echo \$PATH** to verify that the path has been added.

User IDs

Kerberos

If your installation uses Kerberos authentication, see [Chapter 13, “Configuring Kerberos,”](#) on page 111.

UNIX User Accounts and Home Directories

You must create one or more user IDs and enable certain permissions for the SAS Data Loader for Hadoop vApp user.

Note: MapR users must create a special user ID file. For more information, see [“For MapR Users”](#) on page 76.

To configure user IDs, follow these steps:

1. Choose one of the following options for user IDs:
 - Create one user ID that any vApp user can use for login.

Note: Do not use the super user, which is typically `hdfs`.
 - Create an individual user ID for each vApp user.
 - Map the user ID to a user principal for clusters using Kerberos.
2. Create UNIX user IDs on all nodes of the cluster and assign them to a group.
3. Create a user home directory and Hadoop staging directory in HDFS. The user home directory is **/user/myuser**. The Hadoop staging directory is controlled by the setting `yarn.app.mapreduce.am.staging-dir` in `mapred-site.xml` and defaults to **/user/myuser**.
4. Change the permissions and owner of **/user/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

Configuration Values

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

Host

specifies the full host name of the machine on the cluster running the HiveServer2 server.

Port

specifies the number of the HiveServer2 server port on your Hadoop cluster. For most distributions, the default is 10000.

User ID

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

Note:

- For Cloudera and Hortonworks user IDs, see [“UNIX User Accounts and Home Directories” on page 107](#).
- For MapR user IDs, the user ID information is supplied through the `mapr-user.json` file. For more information, see [“For MapR Users” on page 76](#).

Password

if your enterprise uses LDAP, you must supply the vApp user with the LDAP password. This field must be blank otherwise.

Oozie URL

specifies the Oozie base URL. The URL is the property `oozie.base.url` in the file `oozie-site.xml`. The URL is similar to the following example: `http://host_name:port_number/oozie/`.

Although the Oozie web UI at this URL does not have to be enabled for Data Loader to function, it is useful for monitoring and debugging Oozie jobs. Confirm that the Oozie Web UI is enabled before providing it to the vApp user. Consult your cluster documentation for more information.

Providing vApp User Configuration Information

The configuration components and information that the Hadoop administrator must supply to the vApp user are summarized in the following tables:

Table 12.2 Configuration Components and Information

Component	Location of Description
JDBC drivers	See “JDBC Drivers” on page 106 .
User IDs	See “User IDs” on page 107 .

The SAS Data Loader for Hadoop vApp that runs on the client machine contains both Settings and Configuration dialog boxes. For more information about these dialog boxes, see the *SAS Data Loader for Hadoop: vApp Deployment Guide*.

The Configuration dialog box contains certain fields for which you must provide values to the vApp user. These fields are as follows:

Table 12.3 Configuration Fields

Field	Location of Description
Host	See “Host” on page 108.
Port	See “Port” on page 108.
User ID	See “User ID” on page 108.
Password	See “Password” on page 108.
Oozie URL	See “Oozie URL” on page 108.

Chapter 13

Configuring Kerberos

About Kerberos on the Hadoop Cluster	111
Client Configuration	111
Kerberos Configuration	112
Overview	112
vApp	112
Hadoop	114
SAS LASR Analytic Server	115
Providing vApp User Configuration Information	115

About Kerberos on the Hadoop Cluster

If your enterprise uses Kerberos security, you must have all valid tickets in place on the cluster. When SAS In-Database Technologies for Hadoop is deployed, the HDFS user must have a valid ticket.

Note:

- For all Hadoop distributions except MapR, the default HDFS user is **hdfs**. For MapR distributions, the default HDFS user is **mapr**.
- If you set a maximum lifetime for Kerberos tickets, ensure that the person deploying SAS In-Database Technologies for Hadoop is aware of the expiration date of the ticket.

After configuring Kerberos, provide the necessary configuration values to the vApp user. See [“Providing vApp User Configuration Information” on page 115](#).

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

Client Configuration

Certain configuration must take place on the client machine that hosts the vApp. For example, the hosts file on the client machine must be modified to include the host name

that is used to access SAS Data Loader for Hadoop. This host name must be the same host name that is used to generate keytabs for Kerberos, as described in [“Kerberos Configuration” on page 112](#). For more information, see the *SAS Data Loader for Hadoop: vApp Deployment Guide*.

Kerberos Configuration

Overview

The Kerberos topology contains multiple tiers. They 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 less 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. Finally, it is possible to configure an all-MIT environment using the MIT Kerberos for Windows libraries to authenticate the client. 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 middle tier to the SAS Object Spawner.
- `HTTP/hostname@KRBREALM`. This allows single sign-on with the tc Server and the SASLogon web application.

Protecting the vApp with MIT Kerberos

When protecting the vApp using MIT Kerberos, the client machine must be configured to acquire tickets for the vApp from the correct realm. For more information, see the *SAS Data Loader for Hadoop: vApp Deployment Guide*. You must provide the name of the KDC server to the person configuring the client machine.

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 `dltest1.vapps.zzz.com` issue the following commands:

```
$ kadmin -p user2/admin -kt /home/user2/user2_admin.keytab
kadmin: addprinc -randkey +ok_as_delegate host/dltest1.vapps.zzz.com
kadmin: ktadd -k $hostname/host.dltest1.keytab host/dltest1.vapps.zzz.com
kadmin: addprinc -randkey +ok_as_delegate SAS/dltest1.vapps.zzz.com
kadmin: ktadd -k $hostname/SAS.dltest1.keytab SAS/dltest1.vapps.zzz.com
kadmin: addprinc -randkey +ok_as_delegate HTTP/dltest1.vapps.zzz.com
kadmin: ktadd -k $hostname/HTTP.dltest1.keytab HTTP/dltest1.vapps.zzz.com
```

Note: You must enable the `ok_as_delegate` flag to allow ticket delegation in the middle tier.

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 then follow these steps:

1. Create Managed Service Accounts:
 - a. Launch the Server Manager on the domain controller:
 - b. Select **Server Manager** ⇒ **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/dltest1.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/dltest1`.
 - h. Deselect **User must change password at next logon** and the select **Password never expires**.
 - i. Click **Finish**.
 - j. Repeat the previous steps for the SAS and HTTP service accounts.
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 `dltest1.vapps.zzz.com` and `dltest1`:


```
> setspn -A host/dltest1.vapps.zzz.com host_dltest1
> setspn -A SAS/dltest1.vapps.zzz.com SAS_dltest1
> setspn -A HTTP/dltest1.vapps.zzz.com HTTP_dltest1
```
3. Authorize ticket delegation:
 - a. Launch the Server Manager on the domain controller.
 - b. Select **Server Manager** ⇒ **Tools** ⇒ **Active Directory Users and Computers**.
 - c. Select **View** ⇒ **Advanced Features**.
 - d. Select `host/<vapp> user`. Right-click, and then select **Properties**.
 - e. Select the **Delegation** tab.

- f. Select **Trust this user for delegation to any service (Kerberos only)**, and then click **Apply**.
 - g. Navigate to the **Attribute Editor** tab
 - h. On the **Attribute Editor** tab, locate the **msDS-KeyVersionNumber** attribute. Record this number. Click **OK**.
 - i. Repeat the previous steps to authorize ticket delegation for the SAS and HTTP users.
4. Create keytabs for each SPN. For UNIX, continue with this step. For Windows, skip to [Step 5 on page 114](#).
 - a. 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, **dltest1.vapps.zzz.com**, **AD.ZZZ.COM**, **Psword**, and **-k 2 -e arcfour-hmac** are used for these values:

```

ktutil
ktutil: addent -password -p host/dltest1.vapps.zzz.com@AD.ZZZ.COM -k 2 -e arcfour-hmac
Psword for host/dltest1.vapps.zzz.com@AD.ZZZ.COM :
ktutil: addent -password -p host/dltest1.vapps.zzz.com@AD.ZZZ.COM -k 2 -e aes128-cts-hmac-sha1-96
Psword for host/dltest1.host.zzz.com@AD.ZZZ.COM :
ktutil: addent -password -p host/dltest1.vapps.zzz.com@AD.ZZZ.COM -k 2 -e aes256-cts-hmac-sha1-96
Psword for host/dltest1.vapps.zzz.com@AD.ZZZ.COM :
ktutil: wkt host.dltest1.keytab
ktutil: quit

```

- b. Repeat the previous steps to create the SAS and HTTP keytabs.
5. To create keytabs for each SPN on Windows, follow these steps:
 - a. 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, **dltest1.vapps.zzz.com**, **AD.ZZZ.COM**, and **Psword** are used for these values:

```

ktpass.exe -princ host/dltest1.vapps.zzz.com@AD.ZZZ.COM -mapUser user@fully.qualified.domain -pass "Psword"
-pType KRB5_NT_PRINCIPAL -out dltest1-host.keytab -crypto All

```

- b. Repeat the previous steps to create the SAS and HTTP keytabs.
6. Provide the keytabs to the vApp user.

Hadoop

Overview

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

Ensure that the following setting is correct on your cluster:

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


Configure Kerberos Trusts

If the Kerberos environment includes users or services authenticated by a realm other than the default realm of the cluster, you must configure the cluster to interpret principals from the trusted realm. This is the case when the cluster is protected by MIT Kerberos and the client is protected by Active Directory.

Cloudera

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

Other Distributions

When the cluster is protected by MIT Kerberos, you must set the properties `hadoop.security.auth_to_local` and `oozie.authentication.kerberos.name.rules` as follows:

```
RULE: [1:$1@$0] (. *@\QAD_DOMAIN_REALM\E$) s/@\QAD_DOMAIN_REALM\E$//
RULE: [2:$1@$0] (. *@\QAD_DOMAIN_REALM\E$) s/@\QAD_DOMAIN_REALM\E$//
RULE: [1:$1@$0] (. *@\QMIT_DOMAIN_REALM\E$) s/@\QMIT_DOMAIN_REALM\E$//
RULE: [2:$1@$0] (. *@\QMIT_DOMAIN_REALM\E$) s/@\QMIT_DOMAIN_REALM\E$//
DEFAULT
```

An example of RULE 1 and RULE 2 for `AD_DOMAIN_REALM` is as follows:

```
RULE: [1:$1@$0] (. *@\QDAFFY_KRB5.COM\E$) s/@\QDAFFY_KRB5.COM\E$//
RULE: [2:$1@$0] (. *@\QDAFFY_KRB5.COM\E$) s/@\QDAFFY_KRB5.COM\E$//
DEFAULT
```

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.

For more information about the SAS LASR Analytic Server administrator, see “LASR Analytic Servers Panel” in the *SAS Data Loader for Hadoop: User’s Guide*.

Providing vApp User Configuration Information

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. For more information about the Settings dialog box, see the *SAS Data Loader for Hadoop: vApp Deployment Guide*. The dialog box contains certain fields for which the Hadoop administrator must provide values to the vApp user. These fields are as follows:

Table 13.1 Settings Fields

Field	Value
Host name	The host name that you create for Kerberos security. See “Client Configuration” on page 111 .
User ID	The normal logon ID for the user.
Kerberos Realm	The name of the Kerberos realm or AD domain against which the user authenticates.
Kerberos configuration file	The location of the Kerberos configuration file.
Host keytab file	The location of the keytab generated for the host SPN. See “vApp” on page 112 .
SAS server keytab file	The location of the keytab generated for the SAS server SPN. See “vApp” on page 112 .
HTTP keytab file	The location of the keytab generated for the HTTP SPN. See “vApp” on page 112 .

You must provide the Kerberos configuration and keytab files to the user.

Part 4

Administrator's Guide for Teradata

Chapter 14

***In-Database Deployment Package for Teradata* 119**

Chapter 15

***Deploying the SAS Embedded Process: Teradata* 125**

Chapter 16

***SAS Data Quality Accelerator for Teradata* 133**

Chapter 14

In-Database Deployment Package for Teradata

Prerequisites	119
Overview of the In-Database Deployment Package for Teradata	119
Required Hot Fixes for the SAS In-Database Code Accelerator for Teradata 9.41	121
Hot Fixes That Are Installed on the SAS Host Operating System	121
Hot Fixes That Are Installed on the Teradata Cluster	123
Teradata Permissions for Publishing Formats and Scoring Models	123
Documentation for Using In-Database Processing in Teradata	124

Prerequisites

SAS Foundation and the SAS/ACCESS Interface to Teradata must be installed before you install and configure the in-database deployment package for Teradata.

The SAS in-database and high-performance analytic products require a specific version of the Teradata client and server environment. For more information, see the SAS Foundation system requirements documentation for your operating environment.

If you are using Teradata 13.10, 14.00, or 14.10, you must run DIPGLOP from the Teradata DIP utility before you install the SAS Embedded Process. DIPGLOP installs the DBCEXTENSION.ServerControl procedure. This procedure is used to stop and shut down the SAS Embedded Process. DIPGLOP is not required for Teradata 15.00 or later.

The SAS Embedded Process installation requires approximately 200MB of disk space in the /opt file system on each Teradata TPA node.

Overview of the In-Database Deployment Package for Teradata

This section describes how to install and configure the in-database deployment package for Teradata (SAS Formats Library for Teradata and SAS Embedded Process). The in-database deployment packages for Teradata must be installed and configured before you can perform the following tasks:

- Use the %INDTD_PUBLISH_FORMATS format publishing macro to publish the SAS_PUT() function and to publish user-defined formats as format functions inside the database.

For more information about using the format publishing macros, see the *SAS In-Database Products: User's Guide*.

- Use the %INDTD_PUBLISH_MODEL scoring publishing macro to publish scoring model files or functions inside the database.

For more information about using the scoring publishing macros, see the *SAS In-Database Products: User's Guide*.

- Use the SAS In-Database Code Accelerator for Teradata to execute DS2 thread programs in parallel inside the database.

For more information, see the *SAS DS2 Language Reference*.

- Perform data quality operations in Teradata using the SAS Data Quality Accelerator for Teradata.

For more information, see *SAS Data Quality Accelerator for Teradata: User's Guide*.

Note: If you are installing the SAS Data Quality Accelerator for Teradata, you must perform additional steps after you install the SAS Embedded Process. For more information, see [Chapter 16, “SAS Data Quality Accelerator for Teradata,” on page 133](#).

- Run SAS High-Performance Analytics when the analytics cluster is using a parallel connection with a remote Teradata data appliance. The SAS Embedded Process, which resides on the data appliance, is used to provide high-speed parallel data transfer between the data appliance and the analytics environment where it is processed.

For more information, see the *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*.

The in-database deployment package for Teradata includes the SAS formats library and the SAS Embedded Process.

The SAS formats library is a run-time library that is installed on your Teradata system. This installation is done so that the SAS scoring model functions or the SAS_PUT() function can access the routines within the run-time library. The SAS formats library contains the formats that are supplied by SAS.

Note: The SAS formats library is not required by the SAS Data Quality Accelerator for Teradata.

The SAS Embedded Process is a SAS server process that runs within Teradata to read and write data. The SAS Embedded Process contains macros, run-time libraries, and other software that is installed on your Teradata system.

Note: If you are performing a system expansion where additional nodes are being added, the version of the SAS formats library and the SAS Embedded Process on the new database nodes must be the same as the version that is being used on already existing nodes.

Note: In addition to the in-database deployment package for Teradata, a set of SAS Embedded Process functions must be installed in the Teradata database. The SAS Embedded Process functions package is downloadable from Teradata. For more information, see [“Installing the SAS Embedded Process Support Functions” on page 131](#).

Required Hot Fixes for the SAS In-Database Code Accelerator for Teradata 9.41

Hot Fixes That Are Installed on the SAS Host Operating System

The following hot fixes are required for the SAS In-Database Code Accelerator for Teradata. These hot fixes must be installed after the SAS In-Database Deployment Package is installed.

These hot fixes are installed on your SAS host operating system:

V68001

For installation instructions, choose one of these links depending on your operating environment:

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V68/V68001/xx/mvs/V68001os.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V68/V68001/xx/win/V68001wn.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V68/V68001/xx/wx6/V68001x6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V68/V68001/xx/s64/V68001s6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V68/V68001/xx/r64/V68001r6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V68/V68001/xx/h6i/V68001hx.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V68/V68001/xx/lax/V68001la.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V68/V68001/xx/sax/V68001sx.html>

V87001

For installation instructions, choose one of these links depending on your operating environment:

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V87/V87001/xx/mvs/V87001os.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V87/V87001/xx/win/V87001wn.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V87/V87001/xx/wx6/V87001x6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V87/V87001/xx/s64/V87001s6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V87/V87001/xx/r64/V87001r6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V87/V87001/xx/h6i/V87001hx.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V87/V87001/xx/lax/V87001la.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V87/V87001/xx/sax/V87001sx.html>

V91001

For installation instructions, choose one of these links depending on your operating environment:

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V91/V91001/xx/mvs/V91001os.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V91/V91001/xx/win/V91001wn.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V91/V91001/xx/wx6/V91001x6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V91/V91001/xx/s64/V91001s6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V91/V91001/xx/r64/V91001r6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V91/V91001/xx/h6i/V91001hx.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V91/V91001/xx/lax/V91001la.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V91/V91001/xx/sax/V91001sx.html>

V61001

For installation instructions, choose one of these links depending on your operating environment:

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V61/V61001/xx/mvs/V61001os.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V61/V61001/xx/win/V61001wn.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V61/V61001/xx/wx6/V61001x6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V61/V61001/xx/s64/V61001s6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V61/V61001/xx/r64/V61001r6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V61/V61001/xx/h6i/V61001hx.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V61/V61001/xx/lax/V61001la.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V61/V61001/xx/sax/V61001sx.html>

V95001

For installation instructions, choose one of these links depending on your operating environment:

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V95/V95001/xx/win/V95001wn.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V95/V95001/xx/wx6/V95001x6.html>

<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V95/V95001/xx/s64/V95001s6.html>
<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V95/V95001/xx/r64/V95001r6.html>
<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V95/V95001/xx/h6i/V95001hx.html>
<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V95/V95001/xx/lax/V95001la.html>
<http://ftp.sas.com/techsup/download/hotfix/HF2/V/V95/V95001/xx/sax/V95001sx.html>

Hot Fixes That Are Installed on the Teradata Cluster

The following hot fix is required for the SAS In-Database Code Accelerator for Teradata. This hot fix must be installed after the SAS In-Database Deployment Package is installed.

This hot fix is installed on the Teradata cluster:

V68003

For installation instructions, see <http://ftp.sas.com/techsup/download/hotfix/HF2/V/V68/V68003/xx/tdl/V68003tl.html>.

Teradata Permissions for Publishing Formats and Scoring Models

Because functions are associated with a database, the functions inherit the access rights of that database. It might be useful to create a separate shared database for the SAS scoring functions or the SAS_PUT() function so that access rights can be customized as needed.

You must grant the following permissions to any user who runs the scoring or format publishing macros:

```
CREATE FUNCTION ON database TO userid
DROP FUNCTION ON database TO userid
EXECUTE FUNCTION ON database TO userid
ALTER FUNCTION ON database TO userid
```

If you use the SAS Embedded Process to run your scoring model, you must grant the following permissions:

```
SELECT, CREATE TABLE, INSERT ON database TO userid
EXECUTE PROCEDURE ON SAS_SYSFNLIB TO userid
EXECUTE FUNCTION ON SAS_SYSFNLIB TO userid
EXECUTE FUNCTION ON SYSLIB.MonitorVirtualConfig TO userid
```

Note: If you plan to use SAS Model Manager with the SAS Scoring Accelerator for in-database scoring, additional permissions are required. For more information, see [Chapter 24, “Configuring SAS Model Manager,” on page 231](#).

Documentation for Using In-Database Processing in Teradata

- *SAS In-Database Products: User's Guide*
- *SAS DS2 Language Reference*
- *SAS Data Quality Accelerator for Teradata: User's Guide*

Chapter 15

Deploying the SAS Embedded Process: Teradata

Teradata Installation and Configuration Steps	125
Upgrading from or Reinstalling a Previous Version	126
Upgrading from or Reinstalling Versions That Were Installed before the July 2015 Release of SAS 9.4	126
Upgrading from or Reinstalling Versions That Were Installed after the July 2015 Release of SAS 9.4	128
Installing the SAS Formats Library and the SAS Embedded Process	129
Moving the SAS Formats Library and the SAS Embedded Process Packages to the Server Machine	129
Installing the SAS Formats Library and the SAS Embedded Process with the Teradata Parallel Upgrade Tool	130
Installing the SAS Embedded Process Support Functions	131
Controlling the SAS Embedded Process	131

Teradata Installation and Configuration Steps

1. If you are upgrading from or reinstalling a previous version, follow the instructions in [“Upgrading from or Reinstalling a Previous Version”](#) on page 126.
2. Install the in-database deployment package.
For more information, see [“Installing the SAS Formats Library and the SAS Embedded Process”](#) on page 129.
3. Install the SAS Embedded Process support functions.
For more information, see [“Installing the SAS Embedded Process Support Functions”](#) on page 131.
4. If you have licensed the SAS In-Database Code Accelerator for Teradata version 9.41, you must download and install some required hot fixes. For more information, see [“Required Hot Fixes for the SAS In-Database Code Accelerator for Teradata 9.41”](#) on page 121.

Note: If you are using any of the following SAS Software, additional configuration is needed:

- If you plan to use SAS Model Manager with the SAS Scoring Accelerator for in-database scoring, perform the additional configuration tasks provided in [Chapter 24, “Configuring SAS Model Manager,”](#) on page 231.
- If you plan to use the SAS Data Quality Accelerator for Teradata, perform the additional configuration tasks provided in [Chapter 16, “SAS Data Quality Accelerator for Teradata,”](#) on page 133.
- If you plan to use the SAS High-Performance Analytics environment, perform the additional configuration tasks provided in *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*.

Upgrading from or Reinstalling a Previous Version

Upgrading from or Reinstalling Versions That Were Installed before the July 2015 Release of SAS 9.4

To upgrade from or reinstall a previous version of the SAS Formats Library, the SAS Embedded Process, or both, follow these steps:

1. Check the current installed version of the SAS formats library.

How you do this depends on the version of the SAS formats library.

- If a SAS 9.2 version of the formats library is currently installed, run this command:

```
psh "rpm -q -a" | grep jazxfbrs
```

If a previous version is installed, a result similar to this is displayed. The version number might be different.

```
jazxfbrs-9.2-1.9
```

- If a SAS 9.3 or SAS 9.4 version of the formats library is currently installed, run this command:

```
psh "rpm -q -a" | grep acc
```

If a previous version is installed, a result similar to this is displayed. The version number might be different.

```
accelterafmt-3.1-1.x86_64
```

If the library is not installed on the Teradata nodes, no output is displayed. You can continue with the installation steps in [“Installing the SAS Formats Library and the SAS Embedded Process”](#) on page 129.

2. Run this command to check the current installed version of the SAS Embedded Process.

```
psh "rpm -qa | grep tkindbsrv"
```

If a previous version is installed, a result similar to this is displayed. The version number might be different.

```
tkindbsrv-9.42_M1-2.x86_64
```

If the SAS Embedded Process is not installed on the Teradata nodes, no output is displayed. You can continue with the installation steps in [“Installing the SAS Formats Library and the SAS Embedded Process” on page 129](#).

3. If a version of the SAS formats library, the SAS Embedded Process, or both is being installed that has a name that is different from the library that was previously installed, then follow these steps. An example would be one of these:

- accelerafmt-3.1-1 replacing jazxfbrs-9.2-1.6
- sepcoretera-4.3000-1 replacing tkindbsrv-9.42_M1-2

- a. If you are upgrading from or reinstalling the SAS Formats Library, shut down the Teradata database.

```
tpareset -y -x shutdown_comment
```

This step is required because an older version of the SAS formats library might be loaded in a currently running SAS query.

Note: If you are upgrading or reinstalling only the SAS Embedded Process (tkindbsrv.rpm file), you do not need to shut down the database. You do need to shut down the SAS Embedded Process. For more information about how to shut down the SAS Embedded Process, see [“Controlling the SAS Embedded Process” on page 131](#).

- b. Confirm that the database is shut down.

```
pdestate -a
```

DOWN/HARDSTOP is displayed if the database is shut down.

- c. If the SAS Data Quality Accelerator for Teradata is installed, you must uninstall it before you uninstall the SAS Embedded Process. For more information, see [“Upgrading from or Re-Installing a Previous Version of the SAS Data Quality Accelerator” on page 134](#).

- d. Remove the old version of the in-database deployment package before you install the updated version.

- To remove the packages from all nodes concurrently, run this command:

```
psh "rpm -e package-name"
```

package-name is either jazxfbrs.9.version, accelerafmt-version, or tkindbsrv-version.

For example, to remove **jazxfbrs**, run the command **psh "rpm -e jazxfbrs-9.2-1.6"**.

- To remove the package from each node, run this command on each node:

```
rpm -e package-name
```

package-name is either jazxfbrs.9.version, accelerafmt-version, or tkindbsrv-version.

4. (Optional) To confirm removal of the package before installing the new package, run this command:

```
psh "rpm -q package-name"
```

package-name is either jazxfbrs.9.version, accelerafmt-version, or tkindbsrv-version.

The SAS Formats Library or the SAS Embedded Process should not appear on any node.

5. Continue with the installation steps in [“Installing the SAS Formats Library and the SAS Embedded Process” on page 129](#).

Upgrading from or Reinstalling Versions That Were Installed after the July 2015 Release of SAS 9.4

To upgrade from or reinstall a previous version of the SAS Formats Library, the SAS Embedded Process, or both, follow these steps:

1. Run this command to check the current installed version of the SAS formats library.

```
psh "rpm -q -a" | grep acc
```

If a previous version is installed, a result similar to this is displayed. The version number might be different.

```
accelterafmt-3.1-1.x86_64
```

If the library is not installed on the Teradata nodes, no output is displayed. You can continue with the installation steps in [“Installing the SAS Formats Library and the SAS Embedded Process” on page 129](#).

2. Run this command to check the current installed version of the SAS Embedded Process.

```
psh "rpm -qa | grep sepcoretera"
```

If a previous version is installed, a result similar to this is displayed. The version number might be different.

```
sepcoretera-4.3000-1.x86_64
```

If the SAS Embedded Process is not installed on the Teradata nodes, no output is displayed. You can continue with the installation steps in [“Installing the SAS Formats Library and the SAS Embedded Process” on page 129](#).

3. If a version of the SAS formats library, the SAS Embedded Process, or both is being installed, and has a name that is different from the library that was previously installed, then follow these steps. An example is one of these:

- accelterafmt-3.1-1 replacing jazxfbrs-9.2-1.6
- sepcoretera-4.3000-*version1* replacing sepcoretera-4.3000-*version2*

- a. If you are upgrading from or reinstalling the SAS Formats Library, shut down the Teradata database.

```
tpareset -y -x shutdown_comment
```

This step is required because an older version of the SAS formats library might be loaded in a currently running SAS query.

Note: If you are upgrading or reinstalling only the SAS Embedded Process (tkindbsrv.rpm file), you do not need to shut down the database. You do need to shut down the SAS Embedded Process. For more information about how to shut down the SAS Embedded Process, see [“Controlling the SAS Embedded Process” on page 131](#).

- b. Confirm that the database is shut down.

```
pdestate -a
```

DOWN/HARDSTOP is displayed if the database is shut down.

- c. Remove the old version before you install the updated version of the in-database deployment package.

- To remove the packages from all nodes concurrently, run this command:

```
psh "rpm -e package-name"
```

package-name is either *accelerafmt-version* or *sepcoretera-4.30000-version*.

For example, to remove **sepcoretera**, run the command **psh "rpm -e sepcoretera-4.3000-1"**.

- To remove the package from each node, run this command on each node:

```
rpm -e package-name
```

package-name is either *accelerafmt-version* or *sepcoretera-4.30000-version*.

4. (Optional) To confirm removal of the package before installing the new package, run this command:

```
psh "rpm -q package-name"
```

package-name is either *accelerafmt-version* or *sepcoretera-9.43000-version*.

The SAS Formats Library or the SAS Embedded Process should not appear on any node.

5. Continue with the installation steps in [“Installing the SAS Formats Library and the SAS Embedded Process” on page 129](#).

Installing the SAS Formats Library and the SAS Embedded Process

Moving the SAS Formats Library and the SAS Embedded Process Packages to the Server Machine

1. Locate the SAS Formats Library for Teradata deployment package file, *accelerafmt-3.1-n.x86_64.rpm*. *n* is a number that indicates the latest version of the file. If this is the initial installation, *n* has a value of 1.

The *accelerafmt-3.1-n.x86_64.rpm* file is located in the ***SAS-installation-directory/SASFormatsLibraryforTeradata/3.1/TeradataonLinux/*** directory.

Note: The SAS formats library is not required by the SAS Data Quality Accelerator for Teradata.

2. Move the package file to your Teradata database server in a location where it is both Read and Write accessible. You need to move this package file to the server machine in accordance with procedures used at your site. Here is an example using secure copy.

```
scp accelerafmt-3.1-n.x86_64.rpm root@teramach1:/sasdir/18MAR15
```

This package file is readable by the Teradata Parallel Upgrade Tool.

3. Locate the SAS Embedded Process deployment package file, *sepcoretera-9.43000-n.x86_64.rpm*. *n* is a number that indicates the latest version of the file. Follow these steps :

- a. Navigate to the **YourSASDepot/standalone_installs** directory. This directory was created when you created your SAS Software Depot.
- b. Locate the **en_sasexe.zip** file. The **en_sasexe.zip** file is located in the **YourSASDepot/ standalone_installs/ SAS_Core_Embedded_Process_Package_for_Teradata/9_43/ Teradata_on_Linux/** directory.

The **sepcoretera-9.43000-n.x86_64.rpm** file is included in this ZIP file.

- c. Copy the **en_sasexe.zip** file to a temporary directory on the server machine. You need to move this package file to the server machine in accordance with procedures used at your site. Here is an example using secure copy.

```
scp en_sasexe.zip root@teramach1:/SomeTempDir
```

- d. Log on to the cluster and navigate to the temporary directory in Step 3c.
- e. Unzip **en_sasexe.zip**.

After the file is unzipped, a **sasexe** directory is created in the same location as the **en_sasexe.zip** file. The **sepcoretera-9.43000-n.x86_64.rpm** should be in the **/SomeTempDir/sasexe** directory.

4. Copy the **sepcoretera-9.43000-n.x86_64.rpm** file to the same location on the server as the **accelterafmt-3.1-n.x86_64.rpm** file in Step 2.

You need to move this package file to the server machine in accordance with procedures used at your site. Here is an example using secure copy.

```
scp sepcoretera-9.43000-n.x86_64.rpm root@teramach1:/sasdir/18MAR15
```

This package file is readable by the Teradata Parallel Upgrade Tool.

Installing the SAS Formats Library and the SAS Embedded Process with the Teradata Parallel Upgrade Tool

This installation should be performed by a Teradata systems administrator in collaboration with Teradata Customer Services. A Teradata Change Control is required when a package is added to the Teradata server. Teradata Customer Services has developed change control procedures for installing the SAS in-database deployment package.

The steps assume full knowledge of the Teradata Parallel Upgrade Tool and your environment. For more information about using the Teradata Parallel Upgrade Tool, see the *Parallel Upgrade Tool (PUT) Reference* which is at the Teradata Online Publications site, located at <http://www.info.teradata.com/GenSrch/eOnLine-Srch.cfm>. On this page, search for “Parallel Upgrade Tool” and download the appropriate document for your system.

The following steps explain the basic steps to install the SAS formats library package by using the Teradata Parallel Upgrade Tool.

Note: The Teradata Parallel Upgrade Tool prompts are subject to change as Teradata enhances its software.

1. Locate the SAS Formats Library and the SAS Embedded Process packages on your server machine. They must be in a location where they can be accessed from at least one of the Teradata nodes. For more information, see “[Moving the SAS Formats Library and the SAS Embedded Process Packages to the Server Machine](#)” on page 129.

2. Start the Teradata Parallel Upgrade Tool.
3. Be sure to select all Teradata TPA nodes for installation, including Hot Stand-By nodes.
4. If Teradata Version Migration and Fallback (VM&F) is installed, you might be prompted whether to use VM&F or not. If you are prompted, choose Non-VM&F installation.
5. If the installation is successful, `accelterfmt-3.1-n` or `sepcoretera-9.43000-n.x86_64` is displayed. *n* is a number that indicates the latest version of the file.

Alternatively, you can manually verify that the installation is successful by running these commands from the shell prompt.

```
psh "rpm -q -a" | grep accelterafmt
psh "rpm -q -a" | grep sepcoretera
```

Installing the SAS Embedded Process Support Functions

The SAS Embedded Process support function package (sasepfunc) includes stored procedures that generate SQL to interface with the SAS Embedded Process and functions that load the SAS program and other run-time control information into shared memory. The SAS Embedded Process support functions setup script creates the SAS_SYSFNLIB database and the SAS Embedded Process interface fast path functions in TD_SYSFNLIB.

The SAS Embedded Process support function package is available from the Teradata Software Server. For access to the package that includes the installation instructions, contact your local Teradata account representative or the Teradata consultant supporting your SAS and Teradata integration activities.

CAUTION:

If you are using Teradata 15, you must drop the SAS_SYSFNLIB.SASEP_VERSION function to disable the Teradata Table Operator (SASTblOp). Otherwise, your output can contain missing rows or incorrect results. To drop the function, enter the following command: **drop function SAS_SYSFNLIB.SASEP_VERSION**. This issue is fixed in Teradata maintenance release 15.00.04.

Note: If you are using SAS Data Quality Accelerator v2.7, you must contact your Teradata representative to get access to version 15.00-8 or higher of the SAS Embedded Process support functions (sasepfunc-15.00-8).

Controlling the SAS Embedded Process

The SAS Embedded Process starts when a query is submitted. The SAS Embedded Process continues to run until it is manually stopped or the database is shutdown. You might want to disable or shutdown the SAS Embedded Process without shutting down the database.

The following commands control the SAS Embedded Process.

Action performed	Command (by Teradata version)
Provides the status of the SAS Embedded Process.	CALL DBCEXTENSION.SERVERCONTROL ('status', :A); * CALL DBCEXTENSION.SERVERCONTROL ('SAS', 'status', :A); ** CALL SQLJ.SERVERCONTROL ('SAS', 'status', :A); ***
Shuts down the SAS Embedded Process. <i>Note:</i> You cannot shut down until all queries are complete.	CALL DBCEXTENSION.SERVERCONTROL ('shutdown', :A); * CALL DBCEXTENSION.SERVERCONTROL ('SAS', 'shutdown', :A); ** CALL SQLJ.SERVERCONTROL ('SAS', 'shutdown', :A); ***
Stops new queries from being started. Queries that are currently running continue to run until they are complete.	CALL DBCEXTENSION.SERVERCONTROL ('disable', :A); * CALL DBCEXTENSION.SERVERCONTROL ('SAS', 'disable', :A); ** CALL SQLJ.SERVERCONTROL ('SAS', 'disable', :A); ***
Enables new queries to start running.	CALL DBCEXTENSION.SERVERCONTROL ('enable', :A); * CALL DBCEXTENSION.SERVERCONTROL ('SAS', 'enable', :A); ** CALL SQLJ.SERVERCONTROL ('SAS', 'enable', :A); ***

* For Teradata 13.10 and 14.00 only. Note that the Cmd parameter (for example, 'status') must be lowercase.

** For Teradata 14.10 only. Note that the LanguageName parameter, 'SAS', is required and must be uppercase. The Cmd parameter (for example, 'status'), must be lowercase.

*** For Teradata 15 only. Note that the LanguageName parameter, 'SAS', is required and must be uppercase. The Cmd parameter (for example, 'status'), must be lowercase.

Chapter 16

SAS Data Quality Accelerator for Teradata

Introduction	133
Upgrading from or Re-Installing a Previous Version of the SAS Data Quality Accelerator	134
SAS Data Quality Accelerator and QKB Deployment Steps	134
Obtaining a QKB	135
Understanding Your SAS Data Quality Accelerator Software Installation	135
Packaging the QKB	136
Installing the Package Files with the Teradata Parallel Upgrade Tool	137
Creating and Managing SAS Data Quality Accelerator Stored Procedures in the Teradata Database	138
Overview	138
Creating the Data Quality Stored Procedures	139
Granting Users Authorization to the Data Quality Stored Procedures	139
Validating the Accelerator Installation	140
Troubleshooting the Accelerator Installation	141
Updating and Customizing a QKB	142
Removing the Data Quality Stored Procedures from the Database	143

Introduction

In order to use SAS data cleansing functionality inside the Teradata database, the following products must be installed in addition to the SAS In-Database Technologies for Teradata (SAS Embedded Process):

- SAS Data Quality Accelerator for Teradata
- a SAS Quality Knowledge Base (QKB)

SAS Data Quality Accelerator for Teradata contains shell scripts that enable you to create and manage data quality stored procedures within the Teradata database. In addition, it contains a shell script that enables you to package the QKB for deployment inside the Teradata database.

The 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.

Each Teradata node needs approximately 200 MB of disk space in the `/opt` file system for the SAS Embedded Process and approximately 8 GB for the QKB.

Upgrading from or Re-Installing a Previous Version of the SAS Data Quality Accelerator

If you are upgrading from an earlier version of the SAS Data Quality Accelerator for Teradata or reinstalling SAS Data Quality Accelerator 2.7 for Teradata, you must remove the current set of data quality stored procedures from the Teradata database before creating new ones. These steps must be performed before removing or re-installing the SAS in-database deployment package for Teradata (SAS Embedded Process).

To remove SAS Data Quality Accelerator 2.6 for Teradata or earlier, follow these steps:

1. SAS Data Quality Accelerator provides the `dq_uninstall.sh` shell script for removing the data quality stored procedures from the Teradata database. Run the `dq_uninstall.sh` script. For instructions, see [“Removing the Data Quality Stored Procedures from the Database” on page 143](#).
2. Remove the SAS Embedded Process following the steps in [“Upgrading from or Reinstalling a Previous Version” on page 126](#).

To remove SAS Data Quality Accelerator 2.7 for Teradata, follow these steps:

1. Run `dq_uninstall.sh` to remove the stored procedures, if you’ve already created them.
2. Run the following commands to first locate and then remove the SAS Data Quality Accelerator package from the Teradata database:

```
rpm -q -a | grep sepdqacctera
rpm -e package-name
```

Specify the output of the `rpm -q -a` command as the *package-name*. These commands remove the SAS Data Quality Accelerator binaries and shell scripts from the Teradata database.

3. (Optional) Remove the SAS Embedded Process following the steps in [“Upgrading from or Reinstalling a Previous Version” on page 126](#).

It is not necessary to remove the QKB when upgrading or re-installing software. QKB deployment steps automatically overwrite an older version of the QKB when you install a new one.

SAS Data Quality Accelerator and QKB Deployment Steps

To install SAS Data Quality Accelerator 2.7 for Teradata and a QKB, follow these steps:

Note: Before performing these steps, you must have installed the SAS Embedded Process as described in [Chapter 15, “Deploying the SAS Embedded Process”](#):

Teradata,” on page 125. SAS Data Quality Accelerator 2.7 for Teradata requires sepcoretera-9.43000-1 or later.

1. Obtain a QKB.
2. Obtain the SAS Data Quality Accelerator deployment package, sepdqaccta, and qkb_pack script. qkb_pack is a shell script for packaging the QKB. See “Understanding Your SAS Data Quality Accelerator Software Installation” on page 135.
3. Package the QKB into an .rpm file.
4. Deploy the sepdqaccta and sasqkb packages in the Teradata database with the Teradata Parallel Upgrade Tool.
5. Run the dq_install.sh script to create the data quality stored procedures in the Teradata database.
6. Run the dq_grant.sh script to grant users authorization to run the stored procedures.
7. Validate the deployment.

Obtaining a QKB

You can obtain a QKB in one of the following ways:

- Run the SAS Deployment Wizard. In the Select Products to Install dialog box, select the check box for SAS Quality Knowledge Base for your order. This installs the SAS QKB for Contact Information.

Note: This option applies only to the SAS QKB for Contact Information. For step-by-step guidance on installing a QKB using the SAS Deployment Wizard, see the *SAS Quality Knowledge Base for Contact Information: Installation and Configuration Guide* on the [SAS Documentation](#) site.

- Download a QKB from the [SAS Downloads site](#). You can select the SAS QKB for Product Data or SAS QKB for Contact Information.

Select a QKB, and then follow the installation instructions in the Readme file for your operating environment. To open the Readme, you must have a SAS profile. When prompted, you can log on or create a new profile.

- Copy a QKB that you already use with other SAS software in your enterprise. Contact your system administrator for its location.

After your initial deployment, you might want to periodically update the QKB in your Teradata database to make sure that you are using the latest QKB updates provided by SAS. For more information, see “Updating and Customizing a QKB” on page 142.

Understanding Your SAS Data Quality Accelerator Software Installation

The SAS Data Quality Accelerator for Teradata software is delivered in two pieces.

- In-database components are contained in a package file that is delivered in a ZIP file in the `YourSASDepot/standalone_installs/`

SAS_Data_Quality_Accelerator_Embedded_Process_Package_for_Teradata/2_7/Teradata_on_Linux/ directory of the computer on which the SAS depot was installed. The ZIP file is named `en_sasexe.zip`. The package file that it contains is named `sepdqacctera-2.70000-1.x86_64.rpm`. It is not necessary to run the SAS Deployment Wizard to get access to this package. To access the package file:

1. Unzip the `en_sasexe.zip` file.
 2. Put the `sepdqacctera` package file on your Teradata database server in a location where it is available for both reading and writing. The package file must be readable by the Teradata Parallel Upgrade Tool. You need to move this package file to the server machine in accordance with procedures used at your site.
- A script for packaging the QKB is provided in the `<SASHome>` directory of your SAS installation. This script was created by the SAS Deployment Wizard when you installed the SAS In-Database Technologies for Teradata. For more information about this script, see “Packaging the QKB” on page 136.

We recommend that you run the SAS Deployment Wizard and follow the steps for packaging your QKB before attempting to install the `sepdqacctera` package in the Teradata database. That way, you can deploy the QKB package and the `sepdqacctera` package at the same time.

Packaging the QKB

Before a QKB can be deployed in the Teradata database, you must package it into a `.rpm` file. A `.rpm` file is a file that is suitable for installation on Linux systems that use RPM package management software. SAS Data Quality Accelerator for Teradata provides the `qkb_pack` script to package the QKB into a `.rpm`.

Windows and UNIX versions of `qkb_pack` are available. You must run the version that is appropriate for the host environment in which your QKB is installed.

`qkb_pack` is created in the following directories by the SAS Deployment Wizard:

Windows

```
<SASHome>\SASDataQualityAcceleratorforTeradata\2.7\dqacctera\sasmisc
```

UNIX

```
<SASHome>/SASDataQualityAcceleratorforTeradata/2.7/install/pgm
```

You must execute `qkb_pack` from the `<SASHome>` location.

Here is the syntax for executing `qkb_pack`:

Windows:

```
qkb_pack.cmd qkb-dir out-dir
```

UNIX:

```
./qkb_pack.sh qkb-dir out-dir
```

qkb-dir

specify the path to the QKB. Use the name of the QKB’s root directory. Typically, the root directory is found at the following locations:

Windows 7/Windows 8: C:\ProgramData\SAS\QKB\product\version

UNIX: /opt/sas/qkb/share

Note: On Windows systems, QKB information exists in two locations: in **C:\Program Data** and in **C:\Program Files**. For the `qkb_pack` command, you must specify the **C:\Program Data** location.

out-dir

specify the directory where you want the package file to be created.

Here's an example of a command that you might execute to package a SAS QKB for Contact Information that resides on a Windows computer:

```
cd c:\Program Files\SASHome\SASDataQualityAcceleratorforTeradata\2.7\dqacctera\sasmisc
qkb_pack.cmd c:\ProgramData\SAS\SASQualityKnowledgeBase\CI\25 c:\temp\
```

The package file that is created in `C:\temp\` will have a name in the form:

```
sasqkb_product-version-timestamp.noarch.rpm
```

product

is a two-character product code for the QKB, such as `CI` (for Contact Information) or `PD` (for Product Data).

version

is the version number of the QKB.

timestamp

is a UNIX datetime value that indicates when `qkb_pack` was invoked. A UNIX datetime value is stored as the number of seconds since January 1, 1970.

noarch

indicates the package file is platform-independent.

Here is an example of an output filename representing the QKB for Contact Information 25:

```
sasqkb_ci-25.0-1367606747659.noarch.rpm
```

After running `qkb_pack`, put the `sasqkb` package file on your Teradata database server in a location where it is available for both reading and writing. The package file must be readable by the Teradata Parallel Upgrade Tool. You need to move this package file to the server machine in accordance with procedures used at your site.

Follow the steps in “[Installing the Package Files with the Teradata Parallel Upgrade Tool](#)” on [page 137](#) to deploy both the `sasqkb` and `sepdqacctera` package files in the Teradata database.

Installing the Package Files with the Teradata Parallel Upgrade Tool

This installation should be performed by a Teradata systems administrator in collaboration with Teradata Customer Services. A Teradata Change Control is required when a package is added to the Teradata server. Teradata Customer Services has developed change control procedures for installing the SAS in-database deployment package.

The steps assume full knowledge of the Teradata Parallel Upgrade Tool and your environment. For more information about using the Teradata Parallel Upgrade Tool, see the Parallel Upgrade Tool (PUT) Reference, which is at the Teradata Online Publications site located at <http://www.info.teradata.com/GenSrch/eOnLine-Srch.cfm>. On this page, search for “Parallel Upgrade Tool” and download the appropriate document for your system.

The following section explains the basic steps to install the sasqkb and sepdqacctera package files using the Teradata Parallel Upgrade Tool.

Note: It is not necessary to stop and restart the Teradata database when you install a QKB. However, if the SAS Embedded Process is running, you must stop it and then re-start it after the QKB is installed. It is also necessary to stop and restart the SAS Embedded Process for QKB updates. See [“Controlling the SAS Embedded Process” on page 131](#) for information about stopping and restarting the embedded process.

1. Start the Teradata Parallel Upgrade Tool.
2. Be sure to select all Teradata TPA nodes for installation, including Hot Stand-By nodes.
3. If Teradata Version Migration and Fallback (VM&F) is installed, you might be prompted whether to use VM&F. If you are prompted, choose Non-VM&F installation.

If the installation is successful, sepdqacctera-2.70000-*n* is displayed. *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.

Alternatively, you can manually verify that the sepdqacctera installation was successful by running these commands from the shell prompt on one of the Teradata nodes.

```
psh "rpm -q -a" | grep sepdqacctera
psh "rpm -q -a" | grep sasqkb
```

If the installations were successful, these commands return the version numbers of sepdqacctera and sasqkb packages, respectively. Failure to return an output indicates that a library of that name could not be found.

The QKB is installed in the `/opt/qkb/default` directory of each Teradata node.

Creating and Managing SAS Data Quality Accelerator Stored Procedures in the Teradata Database

Overview

SAS data quality functionality is provided in the Teradata database as Teradata stored procedures. The sepdqacctera package installs three scripts in the Teradata database in addition to deploying SAS Data Quality Accelerator binaries:

- a stored procedure creation script named dq_install.sh
- a stored procedure removal script named dq_uninstall.sh
- a user authorization script named dq_grant.sh

The scripts are created in the `/opt/SAS/SASTKInDatabaseServer/9.4/TeradataonLinux/install/pgm` directory of the Teradata database server.

Run the dq_install.sh shell script to create the stored procedures. For more information, see [“Creating the Data Quality Stored Procedures” on page 139](#). Then, run dq_grant.sh to grant users access to the stored procedures. See [“Granting Users Authorization to the Data Quality Stored Procedures” on page 139](#).

Finally, see [“Validating the Accelerator Installation” on page 140](#). If you have problems, see [“Troubleshooting the Accelerator Installation” on page 141](#).

For information about dq_uninstall.sh, see [“Removing the Data Quality Stored Procedures from the Database” on page 143](#).

The dq_install.sh, dq_uninstall.sh, and dq_grant.sh shell scripts must be run as the root user.

Creating the Data Quality Stored Procedures

The data quality stored procedures are created in the Teradata database by running the dq_install.sh shell script. The dq_install.sh script is located in the `/opt/SAS/SASTKInDatabaseServer/9.4/TeradataonLinux/install/pgm` directory of the Teradata database server.

The dq_install.sh script requires modification before it can be run. The Teradata administrator must edit the shell script to specify the site-specific Teradata server name and DBC user logon credentials for the DBC_PASS=, DBC_SRVR=, and DBC_USER= variables.

Running dq_install.sh puts the data quality stored procedures into the SAS_SYSFNLIB database and enables the accelerator functionality.

Here is the syntax for executing dq_install.sh:

```
./dq_install.sh <-l log-path>
```

log-path

specifies an alternative name and location for the dq_install.sh log. When this parameter is omitted, the script creates a file named dq_install.log in the current directory.

Granting Users Authorization to the Data Quality Stored Procedures

The dq_grant.sh shell script is provided to enable the Teradata system administrator to grant users authorization to the data quality stored procedures. The dq_grant.sh script is located in the `/opt/SAS/SASTKInDatabaseServer/9.4/TeradataonLinux/install/pgm` directory of the Teradata database server. Before running the dq_grant.sh script, the Teradata administrator must edit it to specify the site-specific Teradata server name and DBC user logon credentials for the DBC_SRVR=, DBC_USER=, and DBC_PASS= variables. The user name specified in DBC_USER= and DBC_PASS= must have grant authority in the database.

Here is the syntax for executing dq_grant.sh:

```
./dq_grant.sh <-l log-path> user-name
```

log-path

specifies an alternative name and location for the dq_grant.sh log. When this parameter is omitted, the script creates a file named dq_grant.log in the current directory.

user-name

is the user name to which permission is being granted. The target user account must already exist in the Teradata database.

The authorizations granted by dq_grant.sh augment existing authorizations that the target user account already has in the Teradata database.

After you have installed the sepcoretera, sepdqacctera, and sasqkb package files and run the dq_install.sh and dq_grant.sh scripts, the installation of the SAS Data Quality Accelerator for Teradata is complete.

Validating the Accelerator Installation

Here is a simple BTEQ program that can be used to verify that the SAS Data Quality Accelerator for Teradata is operational.

The code first lists the locales that are installed in the QKB. Then it creates a table named Dqacctest and executes the DQ_GENDER() stored procedure on the table. Before running the example, substitute a real value for the *output_table_1*, *output_table_2*, and *locale* variables throughout the program. For *locale*, use one of the values returned by the DQ_LIST_LOCALES() stored procedure. This example assumes that the SAS Data Quality Accelerator for Teradata is using the QKB for Contact Information.

The example also sets the SAS Data Quality Accelerator DQ_OVERWRITE_TABLE option to create temporary output tables in the SAS Data Quality Accelerator session. If you run the example again in the same SAS Data Quality Accelerator session, the new output tables overwrite any existing output tables and the output tables are automatically discarded at the end of the session. The DROP TABLE statement removes table Dqacctest from your database.

```
call sas_sysfnlib.dq_list_locales('mydb.output_table_1');
select * from mydb.output_table_1;

call sas_sysfnlib.dq_set_option('DQ_OVERWRITE_TABLE', '1');

create table mydb.dqacctest (id_num integer, name varchar(64))
    unique primary index(id_num);

insert into mydb.dqacctest (id_num, name) values (1, 'John Smith');
insert into mydb.dqacctest (id_num, name) values (2, 'Mary Jones');

call sas_sysfnlib.dq_gender('Name', 'mydb.dqacctest', 'name', 'id_num',
    'mydb.output_table_2', 'locale');

select gender from mydb.output_table_2;
drop table mydb.dqacctest;
```

If the request was successful, the SELECT statement produces an output table that contains this:

```
Gender
-----
M
F
```

Troubleshooting the Accelerator Installation

Q. I ran the sample code and the output tables were not created in my user schema. What now?

A. The stored procedures can fail if one or more of the following are true:

- The request specifies an output location to which the user does not have Write permission. Verify that you have access to the database that is specified in the *output_table* parameters.
- The data quality stored procedures are not installed correctly. Verify that the stored procedures are in the SAS_SYSFNLIB database by executing the following command in BTEQ:

```
select TableName from dbc.tables where databasename='SAS_SYSFNLIB'
and tablename like 'dq_%';
```

The command should return a list similar to the following list (This is not a complete list.):

```
TableName
-----
dq_set_qkb
dq_match_parsed
dqi_drop_view_if_exists
dqi_get_option_default
dq_debug
dq_propercase
dqi_tbl_dbname
dqi_drop_tbl_if_exists
dq_set_option
dqt_error
dq_standardize
dq_standardize_parsed
dq_debug2
dqi_invoke_table
dq_lowercase
dq_set_locale
dq_extract
dq_uppercase
dq_list_bindings
dqi_replace_tags
dq_list_defns
dqi_call_ep
dqi_get_bool_option
dqi_gen_toktxt
dqt_codegen
dq_match
dq_parse
dqt_trace
dq_pattern
dqi_clear_tok_tbls
dqt_tokname_tmp
dq_format
```

```

dq_list_locales
dqi_invoke_scalar
dqi_invoke_prepared
dq_bind_token
dq_gender

```

If the procedures are absent, run the `dq_install.sh` script again, making sure you are logged in as Teradata system administrator.

- Permission to the data quality stored procedures is not granted correctly. Verify that the target user name submitted to the `dq_grant.sh` script is a valid user account in the Teradata database. Verify that the database server and granter information in the `dq_grant.sh` shell script is correct.
- The QKB is not in the correct location. Look for subdirectories similar to the following in the `/opt/qkb/default` directory on the Teradata nodes: **chopinfo**, **grammar**, **locale**, **phonetx**, **regexlib**, **scheme**, and **vocab**.
- Your SQL request does not use the Teradata dialect. The stored procedures are invoked with the `CALL` keyword from any product that supports the Teradata SQL dialect. When you submit the data quality stored procedures in the SAS SQL procedure using explicit pass-through, the database connection is made in ANSI mode by default. You must specify the `MODE=` option to switch to Teradata mode. Consult the SAS/ACCESS Interface to Teradata documentation for more information about the `MODE=` option. Consult appropriate documentation for how to set Teradata mode in other client programs.

Updating and Customizing a QKB

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, see “What’s New in SAS Quality Knowledge Base.” The What’s New document is available on the SAS Quality Knowledge Base (QKB) 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. To deploy a new QKB, follow the steps in “[Packaging the QKB](#)” on page 136 and “[Installing the Package Files with the Teradata Parallel Upgrade Tool](#)” on page 137. The accelerator supports one QKB in the Teradata database.

The standard definitions in the QKB are sufficient for performing most data quality operations. However, you can use the Customize feature of DataFlux Data Management Studio to modify the QKB definitions to meet specific needs.

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 Teradata database for deployment. When updates to the QKB are required, merge your customizations into an updated QKB locally, and copy the updated, customized QKB to the Teradata node. This enables you to deploy a customized QKB to the Teradata database using the same steps that you would use to deploy a standard QKB. Copying your customized QKB from a local workstation into your cluster also means you will have a backup of the QKB on your local workstation. See 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.

Removing the Data Quality Stored Procedures from the Database

Note: Stop the embedded process by using the instructions at [“Controlling the SAS Embedded Process” on page 131](#) before following these steps. Stopping the SAS Embedded Process ensures that none of the accelerator files are locked when dq_uninstall.sh attempts to remove them.

The accelerator provides the dq_uninstall.sh shell script for removing the data quality stored procedures from the Teradata database. The dq_uninstall.sh script is located in the `/opt/SAS/SASTKInDatabaseServer/9.4/TeradataonLinux/install/pgm` directory of the Teradata database server.

The dq_uninstall.sh script requires modification before it can be run. The Teradata administrator must edit the shell script to specify the site-specific Teradata server name and DBC user logon credentials for the DBC_PASS=, DBC_SRVR=, and DBC_USER= variables.

Here is the syntax for executing dq_uninstall.sh:

```
./dq_uninstall.sh <-l log-path>
```

log-path

specifies an alternative name and location for the dq_uninstall.sh log. When this parameter is omitted, the script creates a file named dq_uninstall.log in the current directory.

Running dq_uninstall.sh disables the SAS Data Quality Accelerator for Teradata functionality and removes the data quality stored procedures from the database. The dq_uninstall.sh script does not remove the QKB or the SAS Embedded Process from the Teradata nodes. Follow whatever procedure is appropriate at your site for removing the QKB. See [“Upgrading from or Reinstalling a Previous Version” on page 126](#) for information about how to uninstall the SAS Embedded Process from the Teradata database. The dq_grant.sh script also does not remove permissions that were granted by dq_grant.sh. You need to remove the permissions in accordance with the procedures used at your site.

Part 5

Administrator's Guides for Aster, DB2, Greenplum, Netezza, Oracle, SAP HANA, and SPD Server

<i>Chapter 17</i>	
Administrator's Guide for Aster	147
<i>Chapter 18</i>	
Administrator's Guide for DB2	153
<i>Chapter 19</i>	
Administrator's Guide for Greenplum	175
<i>Chapter 20</i>	
Administrator's Guide for Netezza	193
<i>Chapter 21</i>	
Administrator's Guide for Oracle	209
<i>Chapter 22</i>	
Administrator's Guide for SAP HANA	215
<i>Chapter 23</i>	
Administrator's Guide for SPD Server	227

Chapter 17

Administrator's Guide for Aster

In-Database Deployment Package for Aster	147
Prerequisites	147
Overview of the In-Database Deployment Package for Aster	147
Aster Installation and Configuration	148
Aster Installation and Configuration Steps	148
Upgrading from or Reinstalling a Previous Version	148
Installing the In-Database Deployment Package Binary Files for Aster	148
Validating the Publishing of the SAS_SCORE() and the SAS_PUT() Functions	151
Aster Permissions	151
Documentation for Using In-Database Processing in Aster	151

In-Database Deployment Package for Aster

Prerequisites

SAS Foundation and the SAS/ACCESS Interface to Aster must be installed before you install and configure the in-database deployment package for Aster.

The SAS Scoring Accelerator for Aster requires a specific version of the Aster client and server environment. For more information, see the SAS Foundation system requirements documentation for your operating environment.

Overview of the In-Database Deployment Package for Aster

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

The in-database deployment package for Aster must be installed and configured before you can use the %INDAC_PUBLISH_MODEL scoring publishing macro to create scoring files inside the database and the %INDAC_PUBLISH_FORMATS format publishing macro to create user-defined format files.

For more information about using the scoring and format publishing macros, see the *SAS In-Database Products: User's Guide*.

The in-database deployment package for Aster includes the SAS Embedded Process. The SAS Embedded Process is a SAS server process that runs within Aster to read and write data. The SAS Embedded Process contains macros, run-time libraries, and other software that is installed on your Aster system so that the SAS_SCORE() and the SAS_PUT() functions can access the routines within its run-time libraries.

Aster Installation and Configuration

Aster 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 148 before installing the in-database deployment package.
2. Install the in-database deployment package.

For more information, see [“Installing the In-Database Deployment Package Binary Files for Aster”](#) on page 148.

Upgrading from or Reinstalling a Previous Version

Follow these steps to upgrade from or reinstall a previous release.

1. Log on to the queen node.
2. Move to the partner directory.
3. If a SAS directory exists in the partner directory, enter this command to remove an existing installation from the queen.

```
ssh -l root name-or-ip-of-queen-node
```

```
cd /home/beehive/partner
```

```
rm -rf SAS
```

If you want to perform a clean install, enter these commands to remove the SAS directory from all the workers.

```
location=/home/beehive/partner/SAS/
for ip in `cat /home/beehive/cluster-management/hosts | grep node |
  awk '{print $3}'`; \
do \
  echo $ip; \
  ssh $ip "rm -r $location"; \
done
rm -rf $location;
```

Installing the In-Database Deployment Package Binary Files for Aster

The in-database deployment package binary files for Aster are contained in a self-extracting archive file named `tkindbsrv-9.43-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. The self-extracting archive file is located

in the *SAS-installation-directory/SASTKInDatabaseServer/9.4/AsternClusteronLinuxx64/* directory.

To install the in-database deployment package binary files for Aster, you need root privileges for the queen node. Once you are logged in to the queen node as root, you need to create a directory in which to put *tkindbsrv-9.43-n_lax.sh*, execute *tkindbsrv-9.43-n_lax.sh*, and install the *SAS_SCORE()* and the *SAS_PUT()* SQL/MR functions.

Enter these commands to install the SAS System Libraries and the binary files:

1. Change the directory to the location of the self-extracting archive file.

```
cd SAS-installation-directory/SASTKInDatabaseServer/9.4/AsternClusteronLinuxx64/
```

2. Log on to the queen node.

```
ssh -l root name-or-ip-of-queen-node
```

3. Move to the parent of the partner directory.

```
cd /home/beehive/
```

4. Create a partner directory if it does not already exist.

```
mkdir partner
```

5. Move to the partner directory.

```
cd partner
```

6. From the SAS client machine, use Secure File Transfer Protocol (SFTP) to transfer the self-extracting archive file to the partner directory.

- a. Using a method of your choice, start the SFTP client.

Here is an example of starting SFTP from a command line.

```
sftp root@name-or-ip-of-queen-node:/home/beehive/partner
```

- b. At the SFTP prompt, enter this command to transfer the self-extracting archive file.

```
put tkindbsrv-9.43-n_lax.sh
```

7. (Optional) If your SFTP client does not copy the executable attribute from the client machine to the server, change the EXECUTE permission on the self-extracting archive file.

```
chmod +x tkindbsrv-9.43-n_lax.sh
```

8. Unpack the self-extracting archive file in the partner directory.

```
./tkindbsrv-9.43-n_lax.sh
```

Note: You might need to add permissions for execution on this file. If so, do a **chmod +x** command on this file.

This installs the SAS Embedded Process on the queen node. When Aster synchronizes the beehive, the files are copied to all the nodes. This can take a long time.

9. (Optional) There are two methods to copy the files to the nodes right away. You can do either of the following.

- Run this code to manually move the files across all nodes on the beehive by using secure copy and SSH.

```
location=/home/beehive/partner/
```

```

cd $location
for ip in `cat /home/beehive/cluster-management/hosts |
    grep node | awk '{print $3}'`; \
do \
echo $ip; \
scp -r SAS root@$ip":$location"; \
done

```

- Run this command to synchronize the beehive and restart the database.

```
/home/beehive/bin/utils/primitives/UpgradeNCluster.py -u
```

10. Change to the directory where SAS is installed.

```
cd /home/beehive/partner/SAS/SASTKInDatabaseServerForAster/9.43/sasexe
```

11. Install the SAS_SCORE(), SAS_PUT(), and other SQL/MR functions.

- a. Start the ACT tool.

```
/home/beehive/clients/act -U db_superuser -w db_superuser-password
-d database-to-install-sas_score-into

```

- b. (Optional) If this is not the first time you have installed the in-database deployment package for Aster, it is recommended that you remove the existing SQL/MR functions before installing the new ones. To do so, enter the following commands.

```

\remove sas_score.tk.so
\remove sas_put.tk.so
\remove sas_row.tk.so
\remove sas_partition.tk.so

```

- c. Enter the following commands to install the new SQL/MR functions. The SQL/MR functions need to be installed under the PUBLIC schema.

```

\install sas_score.tk.so
\install sas_put.tk.so
\install sas_row.tk.so
\install sas_partition.tk.so

```

12. Exit the ACT tool.

```
\q
```

13. Verify the existence and current date of the tkast-runInCluster and tkeastrmr.so files. These two binary files are needed by the SAS SQL/MR functions.

```

for ip in \
`cat /home/beehive/cluster-management/hosts | grep node | awk '{print $3}'`; \
do \
echo $ip; \
ssh $ip "ls -al /home/beehive/partner/SAS/SASTKInDatabaseServerForAster/
9.43/sasexe/tkeastrmr.so"; \
ssh $ip "ls -al /home/beehive/partner/SAS/SASTKInDatabaseServerForAster/
9.43/utilities/bin/tkast-runInCluster"; \
done

```

Validating the Publishing of the SAS_SCORE() and the SAS_PUT() Functions

To validate that the SAS_SCORE() and the SAS_PUT() functions were installed, run the `\df` command in the Aster Client or use any of the following views:

- `nc_all_sqlmr_funcs`, where `all` returns all functions on the system
- `nc_user_sqlmr_funcs`, where `user` returns all functions that are owned by or granted to the user
- `nc_user_owned_sqlmr_funcs`, where `user_owned` returns all functions that are owned by the user

Aster Permissions

The person who installs the in-database deployment package binary files in Aster needs root privileges for the queen node. This permission is most likely, but not necessarily, needed by the Aster system administrator.

For Aster 4.5, no permissions are needed by the person who runs the scoring or format publishing macros, because all functions and files are published to the PUBLIC schema.

For Aster 4.6 or later, the following schema permissions are needed by the person who runs the scoring and format publishing macros, because all functions and files can be published to a specific schema.

USAGE permission

```
GRANT USAGE ON SCHEMA yourschemaname TO youruserid;
```

INSTALL FILE permission

```
GRANT INSTALL FILE ON SCHEMA yourschemaname TO youruserid;
```

CREATE permission

```
GRANT CREATE ON SCHEMA yourschemaname TO youruserid;
```

EXECUTE permission

```
GRANT EXECUTE ON FUNCTION PUBLIC.SAS_SCORE TO youruserid;
```

```
GRANT EXECUTE ON FUNCTION PUBLIC.SAS_PUT TO youruserid;
```

```
GRANT EXECUTE ON FUNCTION PUBLIC.SAS_ROW TO youruserid;
```

```
GRANT EXECUTE ON FUNCTION PUBLIC.SAS_PARTITION TO youruserid;
```

Documentation for Using In-Database Processing in Aster

For information about how to publish SAS formats and scoring models, see the *SAS In-Database Products: User's Guide*, located at <http://support.sas.com/documentation/onlinedoc/indbtech/index.html>

Chapter 18

Administrator's Guide for DB2

In-Database Deployment Package for DB2	153
Prerequisites	153
Overview of the In-Database Deployment Package for DB2	153
Function Publishing Process in DB2	154
DB2 Installation and Configuration	155
DB2 Installation and Configuration Steps	155
Upgrading from or Reinstalling a Previous Version	155
Installing the SAS Formats Library, Binary Files, and SAS Embedded Process ..	158
Running the %INDB2_PUBLISH_COMPILEUDF Macro	164
Running the %INDB2_PUBLISH_DELETEUDF Macro	168
Validating the Publishing of SAS_COMPILEUDF and SAS_DELETEUDF Functions and Global Variables	171
DB2 Permissions	172
Documentation for Using In-Database Processing in DB2	173

In-Database Deployment Package for DB2

Prerequisites

SAS Foundation and the SAS/ACCESS Interface to DB2 must be installed before you install and configure the in-database deployment package for DB2.

The SAS Scoring Accelerator for DB2 requires a specific version of the DB2 client and server environment. For more information, see the SAS Foundation system requirements documentation for your operating environment.

Overview of the In-Database Deployment Package for DB2

This section describes how to install and configure the in-database deployment package for DB2 (SAS Formats Library for DB2 and SAS Embedded Process).

The in-database deployment package for DB2 must be installed and configured before you can perform the following tasks:

- Use the %INDB2_PUBLISH_FORMATS format publishing macro to create or publish the SAS_PUT() function and to create or publish user-defined formats as format functions inside the database.
- Use the %INDB2_PUBLISH_MODEL scoring publishing macro to create scoring model functions inside the database.

For more information about using the format and scoring publishing macros, see the *SAS In-Database Products: User's Guide*.

The in-database deployment package for DB2 contains the SAS formats library and the precompiled binary files for two additional utility functions. The package also contains the SAS Embedded Process.

The SAS formats library is a run-time library that is installed on your DB2 system so that the SAS scoring model functions and the SAS_PUT() function created in DB2 can access the routines within the run-time library. The SAS formats library contains the formats that are supplied by SAS.

The two publishing macros, %INDB2_PUBLISH_COMPILEUDF and %INDB2_PUBLISH_DELETEUDF, register utility functions in the database. The utility functions are called by the format and scoring publishing macros. You must run these two macros before you run the format and scoring publishing macros.

The SAS Embedded Process is a SAS server process that runs within DB2 to read and write data. The SAS Embedded Process contains macros, run-time libraries, and other software that is installed on your DB2 system so that the SAS scoring files created in DB2 can access the routines within the SAS Embedded Process's run-time libraries.

Function Publishing Process in DB2

To publish scoring model functions and the SAS_PUT() function on a DB2 server, the publishing macros perform the following tasks:

- Create and transfer the files to the DB2 environment.
- Compile those source files into object files using the appropriate compiler for that system.
- Link with the SAS formats library.

After that, the publishing macros register the format and scoring model functions in DB2 with those object files. If an existing format or scoring model function is replaced, the publishing macros remove the obsolete object file upon successful compilation and publication of the new format or scoring model functions.

The publishing macros use a SAS FILENAME SFTP statement to transfer the format or scoring source files to the DB2 server. An SFTP statement offers a secure method of user validation and data transfer. The SAS FILENAME SFTP statement dynamically launches an SFTP or PSFTP executable, which creates an SSH client process that creates a secure connection to an OpenSSH Server. All conversation across this connection is encrypted, from user authentication to the data transfers.

Currently, only the OpenSSH client and server on UNIX that supports protocol level SSH-2 and the PUTTY client on WINDOWS are supported. For more information about setting up the SSH software to enable the SAS SFTP to work, please see *Setting Up SSH Client Software in UNIX and Windows Environments for Use with the SFTP Access Method in SAS 9.2, SAS 9.3, and SAS 9.4*, located at <http://support.sas.com/techsup/technote/ts800.pdf>.

Note: This process is valid only when using publishing formats and scoring functions. It is not applicable to the SAS Embedded Process. If you use the SAS Embedded Process, the scoring publishing macro creates the scoring files and uses the SAS/ACCESS Interface to DB2 to insert the scoring files into a model table.

DB2 Installation and Configuration

DB2 Installation and Configuration Steps

1. If you are upgrading from or reinstalling a previous version, follow the instructions in [“Upgrading from or Reinstalling a Previous Version”](#) on page 155.
2. Verify that you can use PSFTP from Windows to UNIX without being prompted for a password or cache.

To do this, enter the following commands from the PSFTP prompt, where *userid* is the user ID that you want to log on as and *machinename* is the machine to which you want to log on.

```
psftp> open userid@machinename
psftp> ls
```

3. Install the SAS formats library, the binary files for the SAS_COMPILEUDF and SAS_DELETEUDF functions, and the SAS Embedded Process.

For more information, see [“Installing the SAS Formats Library, Binary Files, and SAS Embedded Process”](#) on page 158.

4. Run the %INDB2_PUBLISH_COMPILEUDF macro to create the SAS_COMPILEUDF function.

For more information, see [“Running the %INDB2_PUBLISH_COMPILEUDF Macro”](#) on page 164.

5. Run the %INDB2_PUBLISH_DELETEUDF macro to create the SAS_DELETEUDF function.

For more information, see [“Running the %INDB2_PUBLISH_DELETEUDF Macro”](#) on page 168.

6. If you plan to use SAS Model Manager with the SAS Scoring Accelerator for in-database scoring, perform the additional configuration tasks provided in [Chapter 24](#), [“Configuring SAS Model Manager,”](#) on page 231.

Upgrading from or Reinstalling a Previous Version

Overview of Upgrading from or Reinstalling a Previous Version

You can upgrade from or reinstall a previous version of the SAS Formats Library and binary files, the SAS Embedded Process, or both. See the following topics:

- If you want to upgrade or reinstall a previous version of the SAS Formats Library, binary files, and the SAS Embedded Process, see [“Upgrading from or Reinstalling the SAS Formats Library, Binary Files, and the SAS Embedded Process”](#) on page 156.

- If you want to upgrade or reinstall only the SAS Embedded Process, see [“Upgrading from or Reinstalling the SAS Embedded Process” on page 157](#).

Upgrading from or Reinstalling the SAS Formats Library, Binary Files, and the SAS Embedded Process

To upgrade from or reinstall a previous version of the SAS Formats Library, binary files, and the SAS Embedded Process, follow these steps.

Note: These steps also apply if you want to upgrade from or reinstall only the SAS Formats Library and binary files. If you want to upgrade from or reinstall only the SAS Embedded Process, see [“Upgrading from or Reinstalling the SAS Embedded Process” on page 157](#).

1. Drop the SAS_COMPILEUDF and SAS_DELETEUDF functions by running the %INDB2_PUBLISH_COMPILEUDF and %INDB2_PUBLISH_DELETEUDF macros with ACTION=DROP.

Here is an example.

```
%let indconn = user=abcd password=xxxx database=indbdb server=indbsvr;
%indb2_publish_compileudf(action=drop, db2path=/db2/9.4_M2/sqllib,
    compiler_path=/usr/vac/bin);
%indb2_publish_deleteudf(action=drop);
```

2. Confirm that the SAS_COMPILEUDF and SAS_DELETEUDF functions were dropped.

Here is an example.

```
proc sql noerrorstop;
    connect to db2 (user=abcd password=xxxx database=indbdb);
    select * from connection to db2 (
        select cast(funcname as char(40)),
            cast(definer as char(20)) from syscat.functions
            where funcschema='SASLIB' );
quit;
```

If you are upgrading from or reinstalling only the SAS Formats Library and the binary files, skip to Step 6.

3. Enter the following command to see whether the SAS Embedded Process is running.

```
$ps -ef | grep db2sasep
```

If the SAS Embedded Process is running, results similar to this are displayed.

```
ps -ef | grep db2sasep
db2v9 23265382 20840668 0 Oct 06 - 4:03 db2sasep
db2v9 27983990 16646196 1 08:24:09 pts/10 0:00 grep db2sasep
```

4. Stop the DB2 SAS Embedded Process using DB2IDA command.

Use this command to stop the SAS Embedded Process.

```
$db2ida -provider sas -stop
```

If the SAS Embedded Process is still running, an error occurs. Enter this command to force the SAS Embedded Process to stop.

```
$db2ida -provider sas -stopforce
```

For more information about the DB2IDA command, see [“Controlling the SAS Embedded Process for DB2” on page 163](#).

5. Remove the SAS directory that contain the SAS Embedded Process binary files from the DB2 instance path.

Enter these commands to move to the *db2instancepath* directory and remove the SAS directory. *db2instancepath* is the path to the SAS Embedded Process binary files in the DB2 instance.

```
$ cd db2instancepath
$ rm -fr SAS
```

6. Stop the DB2 instance.

- a. Log on to the DB2 server and enter this command to determine whether there are any users connected to the instance.

```
$db2 list applications
```

- b. If any users are connected, enter these commands to force them off before the instance is stopped and clear any background processes.

```
$db2 force applications all
$db2 terminate
```

- c. Enter this command to stop the DB2 instance.

```
$db2stop
```

7. Remove the SAS directory from the DB2 instance path. Enter these commands to move to the *db2instancepath/sqllib/function* directory and remove the SAS directory. *db2instancepath/sqllib/function* is the path to the SAS_COMPILEUDF and SAS_DELETEUDF functions in the DB2 instance.

```
$ cd db2instancepath/sqllib/function
$ rm -fr SAS
```

Upgrading from or Reinstalling the SAS Embedded Process

To upgrade from or reinstall a previous version of the SAS Embedded Process, follow these steps.

Note: These steps are for upgrading from or reinstalling only the SAS Embedded Process. If you want to upgrade from or reinstall the SAS Formats Library and binary files or both the SAS Formats Library and binary files and the SAS Embedded Process, you must follow the steps in [“Upgrading from or Reinstalling the SAS Formats Library, Binary Files, and the SAS Embedded Process”](#) on page 156.

1. Enter the following command to see whether the SAS Embedded Process is running.

```
$ps -ef | grep db2sasep
```

If the SAS Embedded Process is running, results similar to this are displayed.

```
ps -ef | grep db2sasep
db2v9 23265382 20840668 0 Oct 06 - 4:03 db2sasep
db2v9 27983990 16646196 1 08:24:09 pts/10 0:00 grep db2sasep
```

2. Enter the following command to determine whether there are any users connected to the instance.

```
$db2 list applications
```

3. Stop the DB2 SAS Embedded Process using DB2IDA command.

Note: If you are upgrading or reinstalling the SAS Embedded Process (tkindbsrv*.sh file), you do not need to shut down the database. The DB2IDA command enables you to upgrade or reinstall only the SAS Embedded Process components without

impacting clients already connected to the database. For more information about the DB2IDA command, see [“Controlling the SAS Embedded Process for DB2” on page 163](#).

Use this command to stop the SAS Embedded Process.

```
$db2ida -provider sas -stop
```

If the SAS Embedded Process is still running, an error occurs. Enter this command to force the SAS Embedded Process to stop.

```
$db2ida -provider sas -stopforce
```

4. Remove the SAS directory that contain the SAS Embedded Process binary files from the DB2 instance path.

Enter these commands to move to the *db2instancepath* directory and remove the SAS directory. *db2instancepath* is the path to the SAS Embedded Process binary files in the DB2 instance.

```
$ cd db2instancepath
$ rm -fr SAS
```

Installing the SAS Formats Library, Binary Files, and SAS Embedded Process

Move the Files to DB2

There are two self-extracting archive files (.sh files) that need to be moved to DB2. You can use PSFTP, SFTP, or FTP to transfer the self-extracting archive files to the DB2 server to be unpacked and compiled.

- The first self-extracting archive file contains the SAS formats library and the binary files for the SAS_COMPILEUDF and SAS_DELETEUDF functions. You need these files when you want to use scoring functions to run your scoring model and when publishing SAS formats.

This self-extracting archive file is located in the *SAS-installation-directory/SASFormatsLibraryforDB2/3.1/DB2on<AIX | Linux64>/* directory.

Choose the self-extracting archive files based on the UNIX platform that your DB2 server runs on. *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.

- AIX: *acceldb2fmt-3.1-n_r64.sh*
- Linux(x86_64): *acceldb2fmt-3.1-n_lax.sh*

The file does not have to be downloaded to a specific location. However, you need to note where it is downloaded so that it can be executed as the DB2 instance owner at a later time. It is recommended that you put the *acceldb2fmt* file somewhere other than the DB2 home directory tree.

- The second self-extracting archive file contains the SAS Embedded Process. You need these files if you want to use the SAS Embedded Process to run your scoring model.

Note: The SAS Embedded Process might require a later release of DB2 than function-based scoring. Please refer to the SAS system requirements documentation.

This self-extracting archive file is located in the ***SAS-installation-directory/SASTKInDatabaseServer/9.4/DB2on<AIX | Linuxx64>/.***

Choose the self-extracting archive files based on the UNIX platform that your DB2 server runs on. *n* is a number that indicates the latest version of the file.

- AIX: tkindbsrv-9.43-*n*_r64.sh
- Linux(x86_64): tkindbsrv-9.43-*n*_lax.sh

You must put the tkindbsrv file in the instance owner's home directory.

List the directory in UNIX to verify that the files have been moved.

Unpack the SAS Formats Library and Binary Files

After the acceldb2fmt-3.1-*n*_lax.sh or acceldb2fmt-3.1-*n*_r64.sh self-extracting archive file is transferred to the DB2 machine, follow these steps to unpack the file. *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.

1. Log on as the user who owns the DB2 instance from a secured shell, such as SSH.
2. Change to the directory where you put the acceldb2fmt file.

```
$ cd path_to_sh_file
```

path_to_sh_file is the location to which you copied the self-extracting archive file.

3. If necessary, change permissions on the file to enable you to execute the script and write to the directory.

```
$ chmod +x acceldb2fmt-3.1-n_r64.sh
```

Note: AIX is the platform that is being used as an example for all the steps in this topic.

4. If there are previous self-extracting archive files in the SAS directory, you must either rename or remove the directory. These are examples of the commands that you would use.

```
$mv SAS to SAS_OLD /* rename SAS directory */
```

```
$rm -fr SAS /* remove SAS directory */
```

5. Use the following commands to unpack the appropriate self-extracting archive file.

```
$ ./sh_file
```

sh_file is either acceldb2fmt-3.1-*n*_lax.sh or acceldb2fmt-3.1-*n*_r64.sh depending on your platform.

After this script is run and the files are unpacked, a SAS tree is built in the current directory. The content of the target directories should be similar to the following, depending on your operating system. Part of the directory path is shaded to emphasize the different target directories that are used.

```
/path_to_sh_file/SAS/SASFormatsLibraryForDB2/3.1-n/bin/
    InstallAccelDB2Fmt.sh
/path_to_sh_file/SAS/SASFormatsLibraryForDB2/3.1-n/bin/CopySASFiles.sh
/path_to_sh_file/SAS/SASFormatsLibraryForDB2/3.1-n/lib/SAS_CompileUDF
/path_to_sh_file/SAS/SASFormatsLibraryForDB2/3.1-n/lib/SAS_DeleteUDF
/path_to_sh_file/SAS/SASFormatsLibraryForDB2/3.1-n/lib/libjazxfbrs.so
/path_to_sh_file/SAS/SASFormatsLibraryForDB2/3.1 ->3.1-n
```

6. Use the following command to place the files in the DB2 instance:

```
$ path_to_sh_file/SAS/SASFormatsLibraryForDB2/3.1-n/bin/
  CopySASFiles.sh db2instancepath/sqlllib
```

db2instancepath/sqlllib is the path to the **sqlllib** directory of the DB2 instance that you want to use.

After this script is run and the files are copied, the target directory should look similar to this.

```
db2instancepath/sqlllib/function/SAS/SAS_CompileUDF
db2instancepath/sqlllib/function/SAS/SAS_DeleteUDF
db2instancepath/sqlllib/function/SAS/libjazzfbrs.so
```

Note: If the SAS_CompileUDF, SAS_DeleteUDF, and libjazzfbrs.so files currently exist under the target directory, you must rename the existing files before you run the CopySASFiles.sh script. Otherwise, the CopySASFiles.sh script does not work, and you get a "Text file is busy" message for each of the three files.

7. Use the DB2SET command to tell DB2 where to find the 64-bit formats library.

```
$ db2set DB2LIBPATH=db2instancepath/sqlllib/function/SAS
```

db2instancepath/sqlllib is the path to the **sqlllib** directory of the DB2 instance that you want to use.

The DB2 instance owner must run this command for it to be successful. Note that this is similar to setting a UNIX system environment variable using the UNIX EXPORT or SETENV commands. DB2SET registers the environment variable within DB2 only for the specified database server.

8. To verify that DB2LIBPATH was set appropriately, run the DB2SET command without any parameters.

```
$ db2set
```

The results should be similar to this one if it was set correctly.

```
DB2LIBPATH=db2instancepath/sqlllib/function/SAS
```

Unpack the SAS Embedded Process Files

After the tkindbsrv-9.43-*n*_aix.sh or tkindbsrv-9.43-*n*_r64.sh self-extracting archive file has been transferred to the DB2 machine, follow these steps to unpack the file. *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.

1. Log on as the user who owns the DB2 instance from a secured shell, such as SSH.
2. Change to the directory where you put the tkindbsrv file.

```
$ cd path_to_sh_file
```

path_to_sh_file is the location to which you copied the self-extracting archive file. This must be the instance owner home directory.

3. If necessary, change permissions on the file to enable you to execute the script and write to the directory.

```
$ chmod +x tkindbsrv-9.43-n_aix.sh
```

4. If there are previous self-extracting archive files in the SAS directory, you must either rename or remove the directory. These are examples of the commands that you would use.

```
$mv SAS to SAS_OLD /* rename SAS directory */
$rm -fr SAS /* remove SAS directory */
```

5. Use the following commands to unpack the appropriate self-extracting archive file.

```
$ ./sh_file
```

sh_file is either tkindbsrv-9.43-n_lax.sh or tkindbsrv-9.43-n_r64.sh depending on your platform.

After this script is run and the files are unpacked, a SAS tree is built in the current directory. The target directories should be similar to the following, depending on your operating system. Part of the directory path is shaded to emphasize the different target directories that are used.

```
/db2instancepath/SAS/SASTKInDatabaseServerForDB2/9.43-n/bin
/db2instancepath/SAS/SASTKInDatabaseServerForDB2/9.43-n/misc
/db2instancepath/SAS/SASTKInDatabaseServerForDB2/9.43-n/sasexe
/db2instancepath/SAS/SASTKInDatabaseServerForDB2/9.43-n/utilities
```

6. Use the DB2SET command to enable the SAS Embedded Process in DB2 and to tell the SAS Embedded Process where to find the SAS Embedded Process library files.

```
$ dbset DB2_SAS_SETTINGS="ENABLE_SAS_EP:true;
LIBRARY_PATH:/db2instancepath/SAS/SASTKInDatabaseServerForDB2/9.43-n/sasexe"
```

The DB2 instance owner must run this command for it to be successful. Note that this is similar to setting a UNIX system environment variable using the UNIX EXPORT for SETENV commands. DB2SET registers the environment variable within DB2 only for the default database instance.

For more information about all of the arguments that can be used with the DB2SET command for the SAS Embedded Process, see [“DB2SET Command Syntax for the SAS Embedded Process” on page 162](#).

7. To verify that the SAS Embedded Process is set appropriately, run the DB2SET command without any parameters.

```
$ db2set
```

The path should be similar to this one if it was set correctly. Note that the DB2LIBPATH that was set when you installed the SAS Formats Library and binary files is also listed.

```
DB2_SAS_SETTINGS=ENABLE_SAS_EP:true
LIBRARY_PATH:/db2instancepath/SAS/SASTKInDatabaseServerForDB2/9.43-n/sasexe
DB2LIBPATH=/db2instancepath/sql/lib/function/SAS
```

8. Stop the database manager instance if it is not stopped already.

```
$ db2stop
```

A message indicating that the stop was successful displays.

If the database manager instance cannot be stopped because application programs are still connected to databases, use the FORCE APPLICATION command to disconnect all users, use the TERMINATE command to clear any background processes, and then use the DB2STOP command.

```
$ db2 list applications
$ db2 force applications all
$ db2 terminate
$ db2stop
```

9. (AIX only) Clear the cache.

```
$ su root
$ slibclean
```

```
$ exit
```

10. Restart the database manager instance.

```
$ db2start
```

11. Verify that the SAS Embedded Process started.

```
$ ps -ef | grep db2sasep
```

If the SAS Embedded Process was started, lines similar to the following are displayed.

```
ps -ef | grep db2sasep
db2v9 23265382 20840668 0 Oct 06 - 4:03 db2sasep
db2v9 27983990 16646196 1 08:24:09 pts/10 0:00 grep db2sasep
```

In the DB2 instance, you can also verify if the SAS Embedded Process log file was created in the DB2 instance's diagnostic directory.

```
$ cd instance-home/sqlllib/db2dump
$ ls -al sasep0.log
```

DB2SET Command Syntax for the SAS Embedded Process

The syntax for the DB2SET command is shown below.

```
DB2SET DB2_SAS_SETTINGS="
ENABLE_SAS_EP:TRUE | FALSE;
<LIBRARY_PATH:path>
<COMM_BUFFER_SZ:size>;
<COMM_TIMEOUT:timeout>;
<RESTART_RETRIES:number-of-tries>;
<DIAGPATH:path>;
<DIAGLEVEL:level-number>;"
```

Arguments

ENABLE_SAS_EP:TRUE | FALSE

specifies whether the SAS Embedded Process is started with the DB2 instance.

Default FALSE

LIBRARY_PATH:path

specifies the path from which the SAS Embedded Process library is loaded.

Requirement The path must be fully qualified.

COMM_BUFFER_SZ:size

specifies the size in 4K pages of the shared memory buffer that is used for communication sessions between DB2 and SAS.

Default ASLHEAPSZ dbm configuration value

Range 1–32767

Requirement size must be an integer value.

COMM_TIMEOUT:timeout

specifies a value in seconds that DB2 uses to determine whether the SAS Embedded Process is non-responsive when DB2 and SAS are exchanging control messages.

Default 600 seconds

Note If the time-out value is exceeded, DB2 forces the SAS Embedded Process to stop in order for it to be re-spawned.

RESTART_RETRIES:*number-of-tries*

specifies the number of times that DB2 attempts to re-spawn the SAS Embedded Process after DB2 has detected that the SAS Embedded Process has terminated abnormally.

Default 10

Range 1–100

Requirement *number-of-tries* must be an integer value.

Note When DB2 detects that the SAS Embedded Process has terminated abnormally, DB2 immediately attempts to re-spawn it. This argument limits the number of times that DB2 attempts to re-spawn the SAS Embedded Process. Once the retry count is exceeded, DB2 waits 15 minutes before trying to re-spawn it again.

DIAGPATH:*path*

specifies the path that indicates where the SAS Embedded Process diagnostic logs are written.

Default DIAGPATH dbm configuration value

Requirement The path must be fully qualified.

DIAGLEVEL:*level-number*

specifies the minimum severity level of messages that are captured in the SAS Embedded Process diagnostic logs. The levels are defined as follows.

- 1 SEVERE
- 2 ERROR
- 3 WARNING
- 4 INFORMATIONAL

Default DIAGLEVEL dbm configuration value

Range 1–4

Controlling the SAS Embedded Process for DB2

The SAS Embedded Process starts when a query is submitted. The SAS Embedded Process continues to run until it is manually stopped or the database is shut down.

The DB2IDA command is a utility that is installed with the DB2 server to control the SAS Embedded Process. The DB2IDA command enables you to manually stop and restart the SAS Embedded Process without shutting down the database. You might use the DB2IDA command to upgrade or reinstall the SAS Embedded Process library or correct an erroneous library path.

Note: DB2IDA requires IBM Fixpack 6 or later.

The DB2IDA command has the following parameters:

-provider sas

specifies the provider that is targeted by the command. The only provider that is supported is "sas".

-start

starts the SAS Embedded Process on the DB2 instance if the SAS Embedded Process is not currently running.

If the SAS Embedded Process is running, this command has no effect.

Note: Once the SAS Embedded Process is started, the normal re-spawn logic in DB2 applies if the SAS Embedded Process is abnormally terminated.

-stop

stops the SAS Embedded Process if it is safe to do so.

If the SAS Embedded Process is stopped, this command has no effect.

If any queries are currently running on the SAS Embedded Process, the **db2ida -stop** command fails and indicates that the SAS Embedded Process is in use and could not be stopped.

Note: DB2 does not attempt to re-spawn the SAS Embedded Process once it has been stopped with the **db2ida -stop** command.

-stopforce

forces the SAS Embedded Process to shut down regardless of whether there are any queries currently running on it.

If the SAS Embedded Process is stopped, this command has no effect.

If any queries are currently running on the SAS Embedded Process, those queries receive errors.

Note: DB2 does not attempt to re-spawn the SAS Embedded Process once it has been stopped with the **db2ida -stopforce** command.

Here are some examples of the DB2IDA command:

```
db2ida -provider sas -stopforce
```

```
db2ida -provider sas -start
```

Running the %INDB2_PUBLISH_COMPILEUDF Macro

Overview of the %INDB2_PUBLISH_COMPILEUDF Macro

The %INDB2_PUBLISH_COMPILEUDF macro publishes the following components to the SASLIB schema in a DB2 database:

- SAS_COMPILEUDF function

The SAS_COMPILEUDF function facilitates the %INDB2_PUBLISH_FORMATS format publishing macro and the %INDB2_PUBLISH_MODEL scoring publishing macro when you use scoring functions to run the scoring model. The SAS_COMPILEUDF function performs the following tasks:

- compiles the format and scoring model source files into object files. This compilation occurs through the SQL interface using an appropriate compiler for the system.
- links with the SAS formats library that is needed for format and scoring model publishing.

- copies the object files to the `db2instancepath/sqlllib/function/SAS` directory. You specify the value of `db2instancepath` in the `%INDB2_PUBLISH_COMPILEUDF` macro syntax.
- `SASUDF_DB2PATH` and `SASUDF_COMPILER_PATH` global variables

The `SASUDF_DB2PATH` and the `SASUDF_COMPILER_PATH` global variables are used when you publish the format and scoring model functions.

You have to run the `%INDB2_PUBLISH_COMPILEUDF` macro only one time in a given database.

The `SAS_COMPILEUDF` function must be published before you run the `%INDB2_PUBLISH_DELETEUDF` macro, the `%INDB2_PUBLISH_FORMATS` macro, and the `%INDB2_PUBLISH_MODEL` macro. Otherwise, these macros fail.

Note: To publish the `SAS_COMPILEUDF` function, you must have the appropriate DB2 user permissions to create and execute this function in the SASLIB schema and in the specified database. For more information, see [“DB2 Permissions” on page 172](#).

`%INDB2_PUBLISH_COMPILEUDF` Macro Run Process

To run the `%INDB2_PUBLISH_COMPILEUDF` macro, follow these steps:

1. Create a SASLIB schema in the database where the `SAS_COMPILEUDF` function is to be published.

The SASLIB schema is used when publishing the `%INDB2_PUBLISH_COMPILEUDF` macro for DB2 in-database processing.

You specify that database in the `DATABASE` argument of the `%INDB2_PUBLISH_COMPILEUDF` macro. For more information, see [“`%INDB2_PUBLISH_COMPILEUDF` Macro Syntax” on page 167](#).

The SASLIB schema contains the `SAS_COMPILEUDF` and `SAS_DELETEUDF` functions and the `SASUDF_DB2PATH` and `SASUDF_COMPILER_PATH` global variables.

2. Start SAS and submit the following command in the Enhanced Editor or Program Editor:

```
%let indconn = server=yourserver user=youruserid password=yourpwd
               database=yourdb schema=saslib;
```

For more information, see the [“INDCONN Macro Variable” on page 165](#).

3. Run the `%INDB2_PUBLISH_COMPILEUDF` macro. For more information, see [“`%INDB2_PUBLISH_COMPILEUDF` Macro Syntax” on page 167](#).

You can verify that the `SAS_COMPILEUDF` function and global variables have been published successfully. For more information, see [“Validating the Publishing of `SAS_COMPILEUDF` and `SAS_DELETEUDF` Functions and Global Variables” on page 171](#).

After the `SAS_COMPILEUDF` function is published, run the `%INDB2_PUBLISH_DELETEUDF` publishing macro to create the `SAS_DELETEUDF` function. For more information, see [“Running the `%INDB2_PUBLISH_DELETEUDF` Macro” on page 168](#).

`INDCONN` Macro Variable

The `INDCONN` macro variable provides the credentials to make a connection to DB2. You must specify the server, user, password, and database information to access the

machine on which you have installed the DB2 database. You must assign the INDCONN macro variable before the %INDB2_PUBLISH_COMPILEUDF macro is invoked.

The value of the INDCONN macro variable for the %INDB2_PUBLISH_COMPILEUDF macro has this format.

```
SERVER=server USER=userid PASSWORD=password
DATABASE=database <SCHEMA=SASLIB>
```

SERVER=*server*

specifies the DB2 server name or the IP address of the server host. If the server name contains spaces or nonalphanumeric characters, enclose the server name in quotation marks.

Requirement The name must be consistent with how the host name was cached when PSFTP *server* was run from the command window. If the full server name was cached, you must use the full server name in the SERVER argument. If the short server name was cached, you must use the short server name. For example, if the long name, *disk3295.unx.comp.com*, is used when PSFTP was run, then *server=disk3295.unx.comp.com* must be specified. If the short name, *disk3295*, was used, then *server=disk3295* must be specified. For more information, see “DB2 Installation and Configuration Steps” on page 155.

USER=*userid*

specifies the DB2 user name (also called the user ID) that is used to connect to the database. If the user name contains spaces or nonalphanumeric characters, enclose the user name in quotation marks.

PASSWORD=*password*

specifies the password that is associated with your DB2 user ID. If the password contains spaces or nonalphabetic characters, enclose the password in quotation marks.

Tip Use only PASSWORD=, PASS=, or PW= for the password argument. PWD= is not supported and causes an error.

DATABASE=*database*

specifies the DB2 database that contains the tables and views that you want to access. If the database name contains spaces or nonalphanumeric characters, enclose the database name in quotation marks.

Requirement The SAS_COMPILEUDF function is created as a Unicode function. If the database is not a Unicode database, then the alternate collating sequence must be configured to use **identity_16bit**.

SCHEMA=SASLIB

specifies SASLIB as the schema name.

Default	SASLIB
----------------	--------

Restriction	The SAS_COMPILEUDF function and the two global variables (SASUDF_DB2PATH and SASUDF_COMPILER_PATH) are published to the SASLIB schema in the specified database. If a value other than SASLIB is used, it is ignored.
--------------------	---

Requirement	The SASLIB schema must be created before publishing the SAS_COMPILEUDF and SAS_DELETEUDF functions.
--------------------	---

%INDB2_PUBLISH_COMPILEUDF Macro Syntax**%INDB2_PUBLISH_COMPILEUDF**

```
(DB2PATH=db2instancepath/sqllib
, COMPILER_PATH=compiler-path-directory
<, DATABASE=database-name>
<, ACTION=CREATE | REPLACE | DROP>
<, OBJNAME=object-file-name>
<, OUTDIR=diagnostic-output-directory>
);
```

Arguments**DB2PATH=*db2instancepath/sqllib***

specifies the parent directory that contains the **function/SAS** subdirectory, where all the object files are stored and defines the SASUDF_DB2PATH global variable that is used when publishing the format and scoring model functions.

Interaction *db2instancepath* should be the same path as the path that was specified during the installation of the SAS_COMPILEUDF binary file. For more information, see Step 3 in “[Unpack the SAS Formats Library and Binary Files](#)” on page 159.

Tip The SASUDF_DB2PATH global variable is defined in the SASLIB schema under the specified database name.

COMPILER_PATH=*compiler-path-directory*

specifies the path to the location of the compiler that compiles the source files and defines the SASUDF_COMPILER_PATH global variable that is used when publishing the format and scoring model functions.

Tip The SASUDF_COMPILER_PATH global variable is defined in the SASLIB schema under the specified database name. The XLC compiler should be used for AIX, and the GGG compiler should be used for Linux.

DATABASE=*database-name*

specifies the name of a DB2 database to which the SAS_COMPILEUDF function is published.

Interaction: The database that you specify in the DATABASE= argument takes precedence over the database that you specify in the INDCONN macro variable. For more information, see “[%INDB2_PUBLISH_COMPILEUDF Macro Run Process](#)” on page 165.

ACTION=CREATE | REPLACE | DROP

specifies that the macro performs one of the following actions:

CREATE

creates a new SAS_COMPILEUDF function.

REPLACE

overwrites the current SAS_COMPILEUDF function, if a SAS_COMPILEUDF function by the same name is already registered, or creates a new SAS_COMPILEUDF function if one is not registered.

DROP

causes the SAS_COMPILEUDF function to be dropped from the DB2 database.

Default CREATE

Tip If the SAS_COMPILEUDF function was published previously and you now specify ACTION=CREATE, you receive warning messages from DB2. If the SAS_COMPILEUDF function was published previously and you specify ACTION=REPLACE, no warnings are issued.

OBJNAME=object-file-name

specifies the object filename that the publishing macro uses to register the SAS_COMPILEUDF function. The object filename is a file system reference to a specific object file, and the value entered for OBJNAME must match the name as it exists in the file system. For example, SAS_CompileUDF is mixed case.

Default SAS_CompileUDF

Interaction If the SAS_COMPILEUDF function is updated, you might want to rename the object file to avoid stopping and restarting the database. If so, the SAS_COMPILEUDF function needs to be reregistered with the new object filename.

OUTDIR=diagnostic-output-directory

specifies a directory that contains diagnostic files.

Tip Files that are produced include an event log that contains detailed information about the success or failure of the publishing process.

Running the %INDB2_PUBLISH_DELETEUDF Macro

Overview of the %INDB2_PUBLISH_DELETEUDF Macro

The %INDB2_PUBLISH_DELETEUDF macro publishes the SAS_DELETEUDF function in the SASLIB schema of a DB2 database. The SAS_DELETEUDF function facilitates the %INDB2_PUBLISH_FORMATS format publishing macro and the %INDB2_PUBLISH_MODEL scoring publishing macro. The SAS_DELETEUDF function removes existing object files when the format or scoring publishing macro registers new ones by the same name.

You have to run the %INDB2_PUBLISH_DELETEUDF macro only one time in a given database.

The SAS_COMPILEUDF function must be published before you run the %INDB2_PUBLISH_DELETEUDF macro, the %INDB2_PUBLISH_FORMATS macro, and the %INDB2_PUBLISH_MODEL macro. Otherwise, these macros fail.

Note: To publish the SAS_DELETEUDF function, you must have the appropriate DB2 user permissions to create and execute this function in the SASLIB schema and specified database. For more information, see [“DB2 Permissions” on page 172](#).

%INDB2_PUBLISH_DELETEUDF Macro Run Process

To run the %INDB2_PUBLISH_DELETEUDF macro, follow these steps:

1. Ensure that you have created a SASLIB schema in the database where the SAS_DELETEUDF function is to be published.

Use the SASLIB schema when publishing the %INDB2_PUBLISH_DELETEUDF macro for DB2 in-database processing.

The SASLIB schema should have been created before you ran the %INDB2_PUBLISH_COMPILEUDF macro to create the SAS_COMPILEUDF

function. The SASLIB schema contains the SAS_COMPILEUDF and SAS_DELETEUDF functions and the SASUDF_DB2PATH and SASUDF_COMPILER_PATH global variables.

The SAS_COMPILEUDF function must be published before you run the %INDB2_PUBLISH_DELETEUDF macro. The SAS_COMPILEUDF and SAS_DELETEUDF functions must be published to the SASLIB schema in the same database. For more information about creating the SASLIB schema, see [“%INDB2_PUBLISH_COMPILEUDF Macro Run Process” on page 165](#).

2. Start SAS and submit the following command in the Enhanced Editor or Program Editor.

```
%let indconn = server=yourserver user=youruserid password=yourpwd
               database=yourdb schema=saslib;
```

For more information, see the [“INDCONN Macro Variable” on page 169](#).

3. Run the %INDB2_PUBLISH_DELETEUDF macro. For more information, see [“%INDB2_PUBLISH_DELETEUDF Macro Syntax” on page 170](#).

You can verify that the function has been published successfully. For more information, see [“Validating the Publishing of SAS_COMPILEUDF and SAS_DELETEUDF Functions and Global Variables” on page 171](#).

After the SAS_DELETEUDF function is published, the %INDB2_PUBLISH_FORMATS and the %INDB2_PUBLISH_MODEL macros can be run to publish the format and scoring model functions.

INDCONN Macro Variable

The INDCONN macro variable provides the credentials to make a connection to DB2. You must specify the server, user, password, and database information to access the machine on which you have installed the DB2 database. You must assign the INDCONN macro variable before the %INDB2_PUBLISH_DELETEUDF macro is invoked.

The value of the INDCONN macro variable for the %INDB2_PUBLISH_DELETEUDF macro has this format.

```
SERVER=server USER=userid PASSWORD=password
DATABASE=database <SCHEMA=SASLIB>
```

SERVER=server

specifies the DB2 server name or the IP address of the server host. If the server name contains spaces or nonalphanumeric characters, enclose the server name in quotation marks.

Requirement The name must be consistent with how the host name was cached when PSFTP *server* was run from the command window. If the full server name was cached, use the full server name in the SERVER argument. If the short server name was cached, use the short server name. For example, if the long name, *disk3295.unx.comp.com*, is used when PSFTP was run, then *server=disk3295.unx.comp.com* must be specified. If the short name, *disk3295*, was used, then *server=disk3295* must be specified. For more information, see [“DB2 Installation and Configuration Steps” on page 155](#).

USER=userid

specifies the DB2 user name (also called the user ID) that is used to connect to the database. If the user name contains spaces or nonalphanumeric characters, enclose the user name in quotation marks.

PASSWORD=*password*

specifies the password that is associated with your DB2 user ID. If the password contains spaces or nonalphabetic characters, enclose the password in quotation marks.

Tip Use only PASSWORD=, PASS=, or PW= for the password argument. PWD= is not supported and causes errors.

DATABASE=*database*

specifies the DB2 database that contains the tables and views that you want to access. If the database name contains spaces or nonalphanumeric characters, enclose the database name in quotation marks.

SCHEMA=SASLIB

specifies SASLIB as the schema name.

Default	SASLIB
Restriction	The SAS_DELETEUDF function is published to the SASLIB schema in the specified database. If a value other than SASLIB is used, it is ignored.
Requirement	Create the SASLIB schema before publishing the SAS_COMPILEUDF and SAS_DELETEUDF functions.

%INDB2_PUBLISH_DELETEUDF Macro Syntax**%INDB2_PUBLISH_DELETEUDF**

```
(<DATABASE=database-name>
<, ACTION=CREATE | REPLACE | DROP>
<, OUTDIR=diagnostic-output-directory>
);
```

Arguments**DATABASE=*database-name***

specifies the name of a DB2 database to which the SAS_DELETEUDF function is published.

Interaction The database that you specify in the DATABASE argument takes precedence over the database that you specify in the INDCONN macro variable. For more information, see [“Running the %INDB2_PUBLISH_DELETEUDF Macro” on page 168](#).

ACTION=CREATE | REPLACE | DROP

specifies that the macro performs one of the following actions:

CREATE

creates a new SAS_DELETEUDF function.

REPLACE

overwrites the current SAS_DELETEUDF function, if a SAS_DELETEUDF function by the same name is already registered, or creates a new SAS_DELETEUDF function if one is not registered.

DROP

causes the SAS_DELETEUDF function to be dropped from the DB2 database.

Default CREATE

Tip If the SAS_DELETEUDF function was published previously and you specify ACTION=CREATE, you receive warning messages from DB2. If the SAS_DELETEUDF function was published previously and you specify ACTION=REPLACE, no warnings are issued.

OUTDIR=diagnostic-output-directory

specifies a directory that contains diagnostic files.

Tip Files that are produced include an event log that contains detailed information about the success or failure of the publishing process.

Validating the Publishing of SAS_COMPILEUDF and SAS_DELETEUDF Functions and Global Variables

To validate that the SAS_COMPILEUDF and SAS_DELETEUDF functions and global variables are created properly, follow these steps.

1. Connect to your DB2 database using Command Line Processor (CLP).
2. Enter the following command to verify that the SASUDF_COMPILER_PATH global variable was published.

```
values(saslib.sasudf_compiler_path)
```

You should receive a result similar to one of the following.

```
/usr/vac/bin      /* on AIX */
/usr/bin          /* on Linux */
```

3. Enter the following command to verify that the SASUDF_DB2PATH global variable was published.

```
values(saslib.sasudf_db2path)
```

You should receive a result similar to the following.

```
/users/db2v9/sqllib
```

In this example, **/users/db2v9** is the value of *db2instancepath* that was specified during installation and **/users/db2v9/sqllib** is also where the SAS_COMPILEUDF function was published.

4. Enter the following command to verify that the SAS_COMPILEUDF and SAS_DELETEUDF functions were published.

```
select funcname, implementation from syscat.functions where
       funcschema='SASLIB'
```

You should receive a result similar to the following.

```
FUNCNAME                                IMPLEMENTATION
-----
SAS_DELETEUDF
/users/db2v9/sqllib/function/SAS/SAS_DeleteUDF!SAS_DeleteUDF
SAS_COMPILEUDF
/users/db2v9/sqllib/function/SAS/SAS_CompileUDF!SAS_CompileUDF
```

DB2 Permissions

There are two sets of permissions involved with the in-database software.

- The first set of permissions is needed by the person who publishes the SAS_COMPILEUDF and SAS_DELETEUDF functions and creates the SASUDF_COMPILER_PATH and SASUDF_DB2PATH global variables.

These permissions must be granted before the %INDB2_PUBLISH_COMPILEUDF and %INDB2_PUBLISH_DELETEUDF macros are run. Without these permissions, running these macros fails.

The following table summarizes the permissions that are needed by the person who publishes the functions and creates the global variables.

Permission Needed	Authority Required to Grant Permission	Examples
CREATEIN permission for the SASLIB schema in which the SAS_COMPILEUDF and SAS_DELETEUDF functions are published and the SASUDF_COMPILER_PATH and SASUDF_DB2PATH global variables are defined	System Administrator or Database Administrator <i>Note:</i> If you have SYSADM or DBADM authority or are the DB2 instance owner, then you have these permissions. Otherwise, contact your database administrator to obtain these permissions.	GRANT CREATEIN ON SCHEMA SASLIB TO <i>compiledetelepublisheruserid</i>
CREATE_EXTERNAL_ROUTINE permission to the database in which the SAS_COMPILEUDF and SAS_DELETEUDF functions are published		GRANT CREATE_EXTERNAL_ROUTINE ON DATABASE TO <i>compiledetelepublisheruserid</i>

- The second set of permissions is needed by the person who publishes the format or scoring model functions. The person who publishes the format or scoring model functions is not necessarily the same person who publishes the SAS_COMPILEUDF and SAS_DELETEUDF functions and creates the SASUDF_COMPILER_PATH and SASUDF_DB2PATH global variables. These permissions are most likely needed by the format publishing or scoring model developer. Without these permissions, the publishing of the format or scoring model functions fails.

Note: Permissions must be granted for every format or scoring model publisher and for each database that the format or scoring model publishing uses. Therefore, you might need to grant these permissions multiple times.

Note: If you are using the SAS Embedded Process to run your scoring functions, only the CREATE TABLE permission is needed.

After the DB2 permissions have been set appropriately, the format or scoring publishing macro should be called to register the formats or scoring model functions.

The following table summarizes the permissions that are needed by the person who publishes the format or scoring model functions.

Permission Needed	Authority Required to Grant Permission	Examples
EXECUTE permission for functions that have been published. This enables the person who publishes the formats or scoring model functions to execute the SAS_COMPILEUDF and SAS_DELETEUDF functions.	System Administrator or Database Administrator <i>Note:</i> If you have SYSADM or DBADM authority, then you have these permissions. Otherwise, contact your database administrator to obtain these permissions.	GRANT EXECUTE ON FUNCTION SASLIB.* TO <i>scoringorfmtpublisherid</i>
CREATE_EXTERNAL_ROUTINE permission to the database to create format or scoring model functions		GRANT CREATE_EXTERNAL_ROUTINE ON DATABASE TO <i>scoringorfmtpublisherid</i>
CREATE_NOT_FENCED_ROUTINE permission to create format or scoring model functions that are not fenced		GRANT CREATE_NOT_FENCED_ROUTINE ON DATABASE TO <i>scoringorfmtpublisherid</i>
CREATEIN permission for the schema in which the format or scoring model functions are published if the default schema (SASLIB) is not used		GRANT CREATEIN ON SCHEMA <i>scoringschema</i> TO <i>scoringorfmtpublisherid</i>
CREATE TABLE permission to create the model table used in with scoring and the SAS Embedded Process		GRANT CREATETAB TO <i>scoringpublisherSEPid</i>
READ permission to read the SASUDF_COMPILER_PATH and SASUDF_DB2PATH global variables <i>Note:</i> The person who ran the %INDB2_PUBLISH_COMPILEUDF macro has these READ permissions and does not need to grant them to himself or herself again.	Person who ran the %INDB2_PUBLISH_COMPILEUDF macro <i>Note:</i> For security reasons, only the user who created these variables has the permission to grant READ permission to other users. This is true even for the user with administrator permissions such as the DB2 instance owner.	GRANT READ ON VARIABLE SASLIB.SASUDF_DB2PATH TO <i>scoringorfmtpublisherid</i> GRANT READ ON VARIABLE SASLIB.SASUDF_COMPILER_PATH TO <i>scoringorfmtpublisherid</i>

Note: If you plan to use SAS Model Manager with the SAS Scoring Accelerator for in-database scoring, additional permissions are required. For more information, see [Chapter 24, “Configuring SAS Model Manager,” on page 231](#).

Documentation for Using In-Database Processing in DB2

For information about how to publish SAS formats or scoring models, see the *SAS In-Database Products: User's Guide*, located at <http://support.sas.com/documentation/onlinedoc/indbtech/index.html>.

Chapter 19

Administrator's Guide for Greenplum

In-Database Deployment Package for Greenplum	175
Prerequisites	175
Overview of the In-Database Deployment Package for Greenplum	176
Greenplum Installation and Configuration	177
Greenplum Installation and Configuration Steps	177
Upgrading from or Reinstalling a Previous Version	177
Installing the SAS Formats Library, Binary Files, and SAS Embedded Process ..	178
Running the %INDGP_PUBLISH_COMPILEUDF Macro	182
Running the %INDGP_PUBLISH_COMPILEUDF_EP Macro	186
Validation of Publishing Functions	189
Validating the Publishing of the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF Functions	189
Validating the Publishing of the SAS_EP Function	190
Controlling the SAS Embedded Process	190
Semaphore Requirements When Using the SAS Embedded Process for Greenplum	191
Greenplum Permissions	192
Documentation for Using In-Database Processing in Greenplum	192

In-Database Deployment Package for Greenplum

Prerequisites

SAS Foundation and the SAS/ACCESS Interface to Greenplum must be installed before you install and configure the in-database deployment package for Greenplum.

The SAS Scoring Accelerator for Greenplum requires a specific version of the Greenplum client and server environment and the Greenplum Partner Connector (GPPC) API. For more information, see the SAS Foundation system requirements documentation for your operating environment.

Overview of the In-Database Deployment Package for Greenplum

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

The in-database deployment package for Greenplum must be installed and configured before you can perform the following tasks:

- Use the %INDGP_PUBLISH_FORMATS format publishing macro to create or publish the SAS_PUT() function and to create or publish user-defined formats as format functions inside the database.
- Use the %INDGP_PUBLISH_MODEL scoring publishing macro to create scoring files or functions inside the database.
- Use the SAS In-Database Code Accelerator for Greenplum to execute DS2 thread programs in parallel inside the database.

For more information, see the *SAS DS2 Language Reference*.

- Run SAS High-Performance Analytics when the analytics cluster is co-located with the Greenplum data appliance or when the analytics cluster is using a parallel connection with a remote Greenplum data appliance. The SAS Embedded Process, which resides on the data appliance, is used to provide high-speed parallel data transfer between the data appliance and the analytics environment where it is processed.

For more information, see the *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*.

For more information about using the format and scoring publishing macros, see the *SAS In-Database Products: User's Guide*.

The in-database deployment package for Greenplum contains the SAS formats library and precompiled binary files for the utility functions. The package also contains the SAS Embedded Process.

The SAS formats library is a run-time library that is installed on your Greenplum system. This installation is done so that the SAS scoring model functions and the SAS_PUT() function created in Greenplum can access the routines within the run-time library. The SAS formats library contains the formats that are supplied by SAS.

The %INDGP_PUBLISH_COMPILEUDF macro registers utility functions in the database. The utility functions are called by the format and scoring publishing macros: %INDGP_PUBLISH_FORMATS and %INDGP_PUBLISH_MODEL. You must run the %INDGP_PUBLISH_COMPILEUDF macro before you run the format and scoring publishing macros.

The SAS Embedded Process is a SAS server process that runs within Greenplum to read and write data. The SAS Embedded Process contains the

%INDGP_PUBLISH_COMPILEUDF_EP macro, run-time libraries, and other software that is installed on your Greenplum system. The

%INDGP_PUBLISH_COMPILEUDF_EP macro defines the SAS_EP table function to the Greenplum database. You use the SAS_EP table function to produce scoring models after you run the %INDGP_PUBLISH_MODEL macro to create the SAS scoring files.

The SAS Embedded Process accesses the SAS scoring files when a scoring operation is performed. You also use the SAS_EP table function for other SAS software that requires it, such as SAS High-Performance Analytics.

Greenplum Installation and Configuration

Greenplum 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 177](#) before installing the in-database deployment package.
2. Install the SAS formats library, the binary files, and the SAS Embedded Process.
For more information, see [“Installing the SAS Formats Library, Binary Files, and SAS Embedded Process” on page 178](#).
3. Run the %INDGP_PUBLISH_COMPILEUDF macro if you want to publish formats or use scoring functions to run a scoring model. Run the %INDGP_PUBLISH_COMPILEUDF_EP macro if you want to use the SAS Embedded Process to run a scoring model or other SAS software that requires it.
For more information, see [“Running the %INDGP_PUBLISH_COMPILEUDF Macro” on page 182](#) or [“Running the %INDGP_PUBLISH_COMPILEUDF_EP Macro” on page 186](#).
4. If you plan to use SAS Model Manager with the SAS Scoring Accelerator for in-database scoring, perform the additional configuration tasks provided in [Chapter 24, “Configuring SAS Model Manager,” on page 231](#).

Note: If you are installing the SAS High-Performance Analytics environment, there are additional steps to be performed 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

Upgrading or Reinstalling the 9.3 SAS Formats Library and SAS Embedded Process

To upgrade from or reinstall the SAS 9.3 version, follow these steps:

1. Delete the `full-path-to-pkglibdir/SAS` directory that contains the SAS Formats Library and the SAS Embedded Process.

Note: You can use the following command to determine the `full-path-to-pkglibdir` directory.

```
pg_config --pkglibdir
```

If you did not perform the Greenplum install, you cannot run the `pg_config --pkglibdir` command. The `pg_config --pkglibdir` command must be run by the person who performed the Greenplum installation.

CAUTION:

If you delete the SAS directory, all the scoring models that you published using scoring functions and all user-defined formats that you published are deleted. If you previously published scoring models using scoring functions or if you previously published user-defined formats, you must republish your scoring

models and formats. If you used the SAS Embedded Process to publish scoring models, the scoring models are not deleted.

It is a best practice to delete the SAS directory when you upgrade from a previous version or reinstall a previous version. Doing so ensures that you get the latest version of both the SAS Formats Library and the SAS Embedded Process.

2. Continue the installation instructions in “Installing the SAS Formats Library, Binary Files, and SAS Embedded Process” on page 178.

Upgrading or Reinstalling the 9.4 SAS Formats Library and SAS Embedded Process

To upgrade from or reinstall the SAS 9.4 version, follow these steps. If you upgrade or install the SAS Formats Library and the SAS Embedded Process in this manner, you do not delete any scoring models or formats that were previously published.

1. Log on to the Greenplum master node as a superuser.
2. Run the UninstallSASEPFiles.sh file.

```
./UninstallSASEPFiles.sh
```

This script stops the SAS Embedded Process on each database host node. The script deletes the `/SAS/SASTKInDatabaseServerForGreenplum` directory and all its contents from each database host node.

The UninstallSASEPFiles.sh file is in the `path_to_sh_file` directory where you copied the `tkindbsrv-9.43-n_lax.sh` self-extracting archive file.

CAUTION:

The timing option must be off for the UninstallSASEPFiles.sh scripts to work. Put `\timing off` in your `.psqlrc` file before running this script.

3. Move to the directory where the SAS Formats Library is installed.

The directory path is `full-path-to-pkglibdir/SAS/`.

Note: You can use the following command to determine the `full-path-to-pkglibdir` directory.

```
pg_config --pkglibdir
```

If you did not perform the Greenplum install, you cannot run the `pg_config --pkglibdir` command. The `pg_config --pkglibdir` command must be run by the person who performed the Greenplum install.

4. Delete the `libjazzfbrs.so` and `sas_compileudf.so` files.
5. In addition to deleting the `libjazzfbrs.so` and `sas_compileudf.so` files on the master node, you must log on to each host node and delete the files on these nodes.
6. Continue the installation instructions in “Installing the SAS Formats Library, Binary Files, and SAS Embedded Process” on page 178.

Installing the SAS Formats Library, Binary Files, and SAS Embedded Process

Moving and Installing the SAS Formats Library and Binary Files

The SAS formats library and the binary files for the publishing macros are contained in a self-extracting archive file. The self-extracting archive file is located in the `SAS-`

installation-directory/SASFormatsLibraryforGreenplum/3.1/GreenplumonLinux64/ directory.

To move and unpack the self-extracting archive file, follow these steps:

1. Using a method of your choice, transfer the `accelgplmfmt-3.1-n_lax.sh` file to your Greenplum master node. *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 file does not have to be downloaded to a specific location. However, you should note where the file is downloaded so that it can be executed at a later time.

2. After the `accelgplmfmt-3.1-n_lax.sh` has been transferred, log on to the Greenplum master node as a superuser.
3. Move to the directory where the self-extracting archive file was downloaded.
4. Use the following command at the UNIX prompt to unpack the self-extracting archive file:

```
./accelgplmfmt-3.1-n_lax.sh
```

Note: If you receive a “permissions denied” message, check the permissions on the `accelgplmfmt-3.1-n_lax.sh` file. This file must have EXECUTE permissions to run.

After the script runs and the files are unpacked, the content of the target directories should look similar to these where *path_to_sh_file* is the location to which you copied the self-extracting archive file.

```
/path_to_sh_file/SAS/SASFormatsLibraryForGreenplum/3.1-1/bin/
  InstallAccelGplmFmt.sh
/path_to_sh_file/SAS/SASFormatsLibraryForGreenplum/3.1-1/bin/
  CopySASFiles.sh
/path_to_sh_file/SAS/SASFormatsLibraryForGreenplum/3.1-1/lib/
  SAS_CompileUDF.so
/path_to_sh_file/SAS/SASFormatsLibraryForGreenplum/3.1-1/lib/
  libjazzfbrs.so
```

5. Use the following command to place the files in Greenplum:

```
./path_to_sh_file/SAS/SASFormatsLibraryForGreenplum/3.1-1/bin/
  CopySASFiles.sh
```

CAUTION:

The timing option must be off for the CopySASFiles.sh script to work. Put `\timing off` in your `.psqlrc` file before running this script.

This command replaces all previous versions of the `libjazzfbrs.so` file.

All the SAS object files are stored under ***full-path-to-pkglibdir/SAS***. The files are copied to the master node and each of the segment nodes.

Note: You can use the following command to determine the ***full-path-to-pkglibdir*** directory:

```
pg_config --pkglibdir
```

If you did not perform the Greenplum install, you cannot run the `pg_config --pkglibdir` command. The `pg_config --pkglibdir` command must be run by the person who performed the Greenplum install.

Note: If you add new nodes at a later date, you must copy all the binary files to the new nodes. For more information, see Step 6.

6. (Optional) If you add new nodes to the Greenplum master node after the initial installation of the SAS formats library and publishing macro, you must copy all the binaries in the **full-path-to-pkglibdir/SAS** directory to the new nodes using a method of your choice such as **scp /SAS**. The binary files include SAS_CompileUDF.so, libjazzfbrs.so, and the binary files for the already published functions.

Moving and Installing the SAS Embedded Process

The SAS Embedded Process is contained in a self-extracting archive file. The self-extracting archive file is located in the **SAS-installation-directory/SASTKInDatabaseServer/9.4/GreenplumonLinux64** directory.

To move and unpack the self-extracting archive file, follow these steps:

1. Using a method of your choice, transfer the **tkindbsrv-9.43-*n*_lax.sh** file to your Greenplum master node. *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 file does not have to be downloaded to a specific location. However, you need to note where it is downloaded so that it can be executed at a later time.

2. After the **tkindbsrv-9.43-*n*_lax.sh** has been transferred, log on to the Greenplum master node as a superuser.
3. Move to the directory where the self-extracting archive file was downloaded.
4. Use the following command at the UNIX prompt to unpack the self-extracting archive file.

```
./tkindbsrv-9.43-n_lax.sh
```

Note: If you receive a “permissions denied” message, check the permissions on the **tkindbsrv-9.43-*n*_lax.sh** file. This file must have EXECUTE permissions to run.

After the script runs and the files are unpacked, the contents of the target directories should look similar to these. *path_to_sh_file* is the location to which you copied the self-extracting archive file in Step 1.

```
/path_to_sh_file/InstallSASEPFiles.sh
/path_to_sh_file/UninstallSASEPFiles.sh
/path_to_sh_file/StartupSASEP.sh
/path_to_sh_file/ShutdownSASEP.sh
/path_to_sh_file/ShowSASEPStatus.sh
/path_to_sh_file/SAS/SASTKInDatabaseServerForGreenplum/9.43/admin
/path_to_sh_file/SAS/SASTKInDatabaseServerForGreenplum/9.43/bin
/path_to_sh_file/SAS/SASTKInDatabaseServerForGreenplum/9.43/logs
/path_to_sh_file/SAS/SASTKInDatabaseServerForGreenplum/9.43/misc
/path_to_sh_file/SAS/SASTKInDatabaseServerForGreenplum/9.43/sasexe
/path_to_sh_file/SAS/SASTKInDatabaseServerForGreenplum/9.43/utilities
```

Note: In addition to the **/path_to_sh_file/** directory, all of the .sh files are also placed in the **/path_to_sh_file/SAS/SASTKInDatabaseServerForGreenplum/9.43/admin** directory.

The **InstallSASEPFiles.sh** file installs the SAS Embedded Process. The next step explains how to run this file. The **StartupSASEP.sh** and **ShutdownSASEP.sh** files enable you to manually start and stop the SAS Embedded Process. For more information about running these two files, see [“Controlling the SAS Embedded Process” on page 190](#).

The `UninstallSASEPFiles.sh` file uninstalls the SAS Embedded Process. The `ShowEPFilesStatus.sh` file shows the status of the SAS Embedded Process on each host.

5. Use the following commands at the UNIX prompt to install the SAS Embedded Process on the master node.

The `InstallSASEPFiles.sh` file must be run from the `/path_to_sh_file/` directory.

```
cd /path_to_sh_file/
./InstallSASEPFiles.sh <-quiet>
```

CAUTION:

The timing option must be off for the `InstallSASEPFiles.sh` script to work.

Put `\timing off` in your `.psqlrc` file before running this script.

Note: `-verbose` is on by default and enables you to see all messages generated during the installation process. Specify `-quiet` to suppress messages.

The installation deploys the SAS Embedded Process to all the host nodes automatically.

The installation also creates a `full-path-to-pkglibdir/SAS` directory. This directory is created on the master node and each host node.

The installation also copies the SAS directories and files from Step 4 across every node.

The contents of the `full-path-to-pkglibdir/SAS/SASTKInDatabaseServerForGreenplum` directory should look similar to these.

```
full-path-to-pkglibdir/SAS/SASTKInDatabaseServerForGreenplum/
  9.43/admin
full-path-to-pkglibdir/SAS/SASTKInDatabaseServerForGreenplum/
  9.43/bin
full-path-to-pkglibdir/SAS/SASTKInDatabaseServerForGreenplum/
  9.43/logs
full-path-to-pkglibdir/SAS/SASTKInDatabaseServerForGreenplum/
  9.43/misc
full-path-to-pkglibdir/SAS/SASTKInDatabaseServerForGreenplum/
  9.43/sasexe
full-path-to-pkglibdir/SAS/SASTKInDatabaseServerForGreenplum/
  9.43/utilities
```

Note: You can use the following command to determine the `full-path-to-pkglibdir` directory:

```
pg_config --pkglibdir
```

If you did not perform the Greenplum install, you cannot run the `pg_config --pkglibdir` command. The `pg_config --pkglibdir` command must be run by the person who performed the Greenplum install.

This is an example of a SAS directory.

```
usr/local/greenplum-db-4.2.3.0/lib/postgresql/SAS
```

Running the %INDGP_PUBLISH_COMPILEUDF Macro

Overview of the %INDGP_PUBLISH_COMPILEUDF Macro

Use the %INDGP_PUBLISH_COMPILEUDF macro if you want to use scoring functions to run scoring models.

Note: Use the %INDGP_PUBLISH_COMPILEUDF_EP macro if you need to use the SAS Embedded Process. For more information, see [“Running the %INDGP_PUBLISH_COMPILEUDF_EP Macro”](#) on page 186.

The %INDGP_PUBLISH_COMPILEUDF macro publishes the following functions to the SASLIB schema in a Greenplum database:

- SAS_COMPILEUDF function

This function facilitates the %INDGP_PUBLISH_FORMATS format publishing macro and the %INDGP_PUBLISH_MODEL scoring publishing macro. The SAS_COMPILEUDF function performs the following tasks:

- compiles the format and scoring model source files into object files. This compilation occurs through the SQL interface using an appropriate compiler for the system.
- links with the SAS formats library.
- copies the object files to the *full-path-to-pkglibdir/SAS* directory. All the SAS object files are stored under *full-path-to-pkglibdir/SAS*.

Note: You can use the following command to determine the *full-path-to-pkglibdir* directory:

```
pg_config --pkglibdir
```

If you did not perform the Greenplum install, you cannot run the `pg_config --pkglibdir` command. The `pg_config --pkglibdir` command must be run by the person who performed the Greenplum install.

- Three utility functions that are used when the scoring publishing macro transfers source files from the client to the host:
 - SAS_COPYUDF function

This function copies the shared libraries to the *full-path-to-pkglibdir/SAS* path on the whole database array including the master and all segments.
 - SAS_DIRECTORYUDF function

This function creates and removes a temporary directory that holds the source files on the server.
 - SAS_DEHEXUDF function

This function converts the files from hexadecimal back to text after the files are exported on the host.

You have to run the %INDGP_PUBLISH_COMPILEUDF macro only one time in each database.

Note: The SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions must be published before you run the %INDGP_PUBLISH_FORMATS or the %INDGP_PUBLISH_MODEL macro. Otherwise, these macros fail.

Note: To publish the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions, you must have superuser permissions to create and execute these functions in the SASLIB schema and in the specified database.

%INDGP_PUBLISH_COMPILEUDF Macro Run Process

To run the %INDGP_PUBLISH_COMPILEUDF macro, follow these steps:

Note: To publish the SAS_COMPILEUDF function, you must have superuser permissions to create and execute this function in the SASLIB schema and in the specified database.

1. Create a SASLIB schema in the database where the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions are published.

You must use “SASLIB” as the schema name for Greenplum in-database processing to work correctly.

You specify that database in the DATABASE argument of the %INDGP_PUBLISH_COMPILEUDF macro. For more information, see [“%INDGP_PUBLISH_COMPILEUDF Macro Syntax” on page 185](#).

The SASLIB schema contains the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions.

2. Start SAS 9.4 and submit the following command in the Enhanced Editor or Program Editor:

```
%let indconn = user=youruserid password=yourpwd dsn=yourdsn;
/* You can use server=yourserver database=yourdb instead of dsn=yourdsn */
```

For more information, see the [“INDCONN Macro Variable” on page 183](#).

3. Run the %INDGP_PUBLISH_COMPILEUDF macro.

For more information, see [“%INDGP_PUBLISH_COMPILEUDF Macro Syntax” on page 185](#).

You can verify that the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions have been published successfully. For more information, see [“Validating the Publishing of the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF Functions” on page 189](#).

INDCONN Macro Variable

The INDCONN macro variable provides the credentials to make a connection to Greenplum. You must specify the user, password, and either the DSN or server and database information to access the machine on which you have installed the Greenplum database. You must assign the INDCONN macro variable before the %INDGP_PUBLISH_COMPILEUDF macro is invoked.

The value of the INDCONN macro variable for the %INDGP_PUBLISH_COMPILEUDF macro has one of these formats:

```
USER=<'>userid<'> PASSWORD=<'>password<'> DSN=<'>dsnname<'>
<PORT=<'>port-number<'>>
```

```
USER=<'>userid<'> PASSWORD=<'>password<'> SERVER=<'>server<'>
DATABASE=<'>database<'> <PORT=<'>port-number<'>>
```

USER=<'>userid<'>

specifies the Greenplum user name (also called the user ID) that is used to connect to the database. If the user name contains spaces or nonalphanumeric characters, enclose the user name in quotation marks.

PASSWORD=<'>password<'>

specifies the password that is associated with your Greenplum user ID. If the password contains spaces or nonalphabetic characters, enclose the password in quotation marks.

Tip Use only PASSWORD=, PASS=, or PW= for the password argument. PWD= is not supported and causes an error.

DSN=<'>datasource<'>

specifies the configured Greenplum ODBC data source to which you want to connect. If the DSN name contains spaces or nonalphabetic characters, enclose the DSN name in quotation marks.

Requirement You must specify either the DSN= argument or the SERVER= and DATABASE= arguments in the INDCONN macro variable.

SERVER=<'>server<'>

specifies the Greenplum server name or the IP address of the server host. If the server name contains spaces or nonalphanumeric characters, enclose the server name in quotation marks.

Requirement You must specify either the DSN= argument or the SERVER= and DATABASE= arguments in the INDCONN macro variable.

DATABASE=<'>database<'>

specifies the Greenplum database that contains the tables and views that you want to access. If the database name contains spaces or nonalphanumeric characters, enclose the database name in quotation marks.

Requirement You must specify either the DSN= argument or the SERVER= and DATABASE= arguments in the INDCONN macro variable.

PORT=<'>port-number<'>

specifies the psql port number.

Default 5432

Requirement The server-side installer uses psql, and psql default port is 5432. If you want to use another port, you must have the UNIX or database administrator change the psql port.

Note: The SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions are published to the SASLIB schema in the specified database. The SASLIB schema must be created before publishing the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions.

%INDGP_PUBLISH_COMPILEUDF Macro Syntax**%INDGP_PUBLISH_COMPILEUDF**

```
(OBJPATH=full-path-to-pkglibdir/SAS
  <, DATABASE=database-name>
  <, ACTION=CREATE | REPLACE | DROP>
  <, OUTDIR=diagnostic-output-directory>
);
```

Arguments**OBJPATH=*full-path-to-pkglibdir*/SAS**

specifies the parent directory where all the object files are stored.

Tip The *full-path-to-pkglibdir* directory was created during installation of the self-extracting archive file. You can use the following command to determine the *full-path-to-pkglibdir* directory:

```
pg_config --pkglibdir
```

If you did not perform the Greenplum install, you cannot run the **pg_config --pkglibdir** command. The **pg_config --pkglibdir** command must be run by the person who performed the Greenplum install.

DATABASE=*database-name*

specifies the name of a Greenplum database to which the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions are published.

Restriction If you specify DSN= in the INDCONN macro variable, do not use the DATABASE argument.

ACTION=CREATE | REPLACE | DROP

specifies that the macro performs one of the following actions:

CREATE

creates a new SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF function.

REPLACE

overwrites the current SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions, if a function by the same name is already registered, or creates a new SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF function if one is not registered.

Requirement If you are upgrading from or reinstalling the SAS Formats Library, run the %INDGP_PUBLISH_COMPILEUDF macro with ACTION=REPLACE. The CopySASFiles.sh install script replaces existing versions of most files. However, you need to replace the existing SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions after you run the CopySASFiles.sh install script. For more information, see [“Upgrading from or Reinstalling a Previous Version” on page 177](#) and [“Moving and Installing the SAS Formats Library and Binary Files” on page 178](#).

DROP

causes the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions to be dropped from the Greenplum database.

Default CREATE

Tip If the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions were published previously and you specify ACTION=CREATE, you receive warning messages that the functions already exist and you are prompted to use REPLACE. If the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions were published previously and you specify ACTION=REPLACE, no warnings are issued.

OUTDIR=*diagnostic-output-directory*

specifies a directory that contains diagnostic files.

Tip Files that are produced include an event log that contains detailed information about the success or failure of the publishing process.

Running the %INDGP_PUBLISH_COMPILEUDF_EP Macro

Overview of the %INDGP_PUBLISH_COMPILEUDF_EP Macro

Use the %INDGP_PUBLISH_COMPILEUDF_EP macro if you want to use the SAS Embedded Process to run scoring models or other SAS software that requires it.

Note: Use the %INDGP_PUBLISH_COMPILEUDF macro if you want to use scoring functions to run scoring models. For more information, see [“Running the %INDGP_PUBLISH_COMPILEUDF Macro” on page 182](#).

The %INDGP_PUBLISH_COMPILEUDF_EP macro registers the SAS_EP table function in the database.

You have to run the %INDGP_PUBLISH_COMPILEUDF_EP macro only one time in each database where scoring models are published.

The %INDGP_PUBLISH_COMPILEUDF_EP macro must be run before you use the SAS_EP function in an SQL query.

Note: To publish the SAS_EP function, you must have superuser permissions to create and execute this function in the specified schema and database.

%INDGP_PUBLISH_COMPILEUDF_EP Macro Run Process

To run the %INDGP_PUBLISH_COMPILEUDF_EP macro, follow these steps:

Note: To publish the SAS_EP function, you must have superuser permissions to create and execute this function in the specified schema and database.

1. Create a schema in the database where the SAS_EP function is published.

Note: You must publish the SAS_EP function to a schema that is in your schema search path.

You specify the schema and database in the INDCONN macro variable. For more information, see [“INDCONN Macro Variable” on page 187](#).

2. Start SAS 9.4 and submit the following command in the Enhanced Editor or Program Editor:

```
%let indconn = user=youruserid password=yourpwd dsn=yourdsn <schema=yourschema>;
/* You can use server=yourserver database=yourdb instead of dsn=yourdsn */
```

For more information, see the [“INDCONN Macro Variable” on page 187](#).

3. Run the %INDGP_PUBLISH_COMPILEUDF_EP macro. For more information, see “%INDGP_PUBLISH_COMPILEUDF_EP Macro Syntax” on page 188.

You can verify that the SAS_EP function has been published successfully. For more information, see “Validating the Publishing of the SAS_EP Function” on page 190.

INDCONN Macro Variable

The INDCONN macro variable provides the credentials to make a connection to Greenplum. You must specify the user, password, and either the DSN or server and database information to access the machine on which you have installed the Greenplum database. You must assign the INDCONN macro variable before the %INDGP_PUBLISH_COMPILEUDF_EP macro is invoked.

The value of the INDCONN macro variable for the %INDGP_PUBLISH_COMPILEUDF_EP macro has one of these formats:

```
USER=<'>userid<'> PASSWORD=<'>password<'> DSN=<'>dsname <'>
<SCHEMA=<'>schema<'>> <PORT=<'>port-number<'>>
```

```
USER=<'>userid<'> PASSWORD=<'>password<'> SERVER=<'>server<'>
DATABASE=<'>database<'> <SCHEMA=<'>schema<'>>
<PORT=<'>port-number<'>>
```

USER=<'>userid<'>

specifies the Greenplum user name (also called the user ID) that is used to connect to the database. If the user name contains spaces or nonalphanumeric characters, enclose the user name in quotation marks.

PASSWORD=<'>password<'>

specifies the password that is associated with your Greenplum user ID. If the password contains spaces or nonalphabetic characters, enclose the password in quotation marks.

Tip Use only PASSWORD=, PASS=, or PW= for the password argument. PWD= is not supported and causes an error.

DSN=<'>datasource<'>

specifies the configured Greenplum ODBC data source to which you want to connect. If the DSN name contains spaces or nonalphabetic characters, enclose the DSN name in quotation marks.

Requirement You must specify either the DSN= argument or the SERVER= and DATABASE= arguments in the INDCONN macro variable.

SERVER=<'>server<'>

specifies the Greenplum server name or the IP address of the server host. If the server name contains spaces or nonalphanumeric characters, enclose the server name in quotation marks.

Requirement You must specify either the DSN= argument or the SERVER= and DATABASE= arguments in the INDCONN macro variable.

DATABASE=<'>database<'>

specifies the Greenplum database that contains the tables and views that you want to access. If the database name contains spaces or nonalphanumeric characters, enclose the database name in quotation marks.

Requirement You must specify either the DSN= argument or the SERVER= and DATABASE= arguments in the INDCONN macro variable.

SCHEMA=<'>schema<'>

specifies the name of the schema where the SAS_EP function is defined.

Default SASLIB

Requirements You must create the schema in the database before you run the %INDGP_PUBLISH_COMPILEUDF_EP macro.

You must publish the SAS_EP function to a schema that is in your schema search path.

PORT=<'>port-number<'>

specifies the psql port number.

Default 5432

Requirement The server-side installer uses psql, and psql default port is 5432. If you want to use another port, you must have the UNIX or database administrator change the psql port.

%INDGP_PUBLISH_COMPILEUDF_EP Macro Syntax

%INDGP_PUBLISH_COMPILEUDF_EP

```
(<OBJPATH=full-path-to-pkglibdir/SAS>
<, DATABASE=database-name>
<, ACTION=CREATE | REPLACE | DROP>
<, OUTDIR=diagnostic-output-directory>
);
```

Arguments

OBJPATH=full-path-to-pkglibdir/SAS

specifies the parent directory where all the object files are stored.

Tip The *full-path-to-pkglibdir* directory was created during installation of the InstallSASEP.sh self-extracting archive file. You can use the following command to determine the *full-path-to-pkglibdir* directory:

```
pg_config --pkglibdir
```

If you did not perform the Greenplum install, you cannot run the **pg_config --pkglibdir** command. The **pg_config --pkglibdir** command must be run by the person who performed the Greenplum install.

DATABASE=database-name

specifies the name of a Greenplum database where the SAS_EP function is defined.

Restriction If you specify DSN= in the INDCONN macro variable, do not use the DATABASE argument.

ACTION=CREATE | REPLACE | DROP

specifies that the macro performs one of the following actions:

CREATE

creates a new SAS_EP function.

REPLACE

overwrites the current SAS_EP function, if a function by the same name is already registered, or creates a new SAS_EP function if one is not registered.

Requirement If you are upgrading from or reinstalling the SAS Embedded Process, run the %INDGP_PUBLISH_COMPILEUDF_EP macro with ACTION=REPLACE. The InstallSASEPFiles.sh install script replaces existing versions of most files. However, you need to replace the existing SAS_EP function after you run the InstallSASEPFiles.sh install script. For more information, see [“Upgrading from or Reinstalling a Previous Version” on page 177](#) and [“Moving and Installing the SAS Embedded Process” on page 180](#).

DROP

causes the SAS_EP function to be dropped from the Greenplum database.

Default CREATE

Tip If the SAS_EP function was defined previously and you specify ACTION=CREATE, you receive warning messages that the functions already exist and you are prompted to use REPLACE. If the SAS_EP function was defined previously and you specify ACTION=REPLACE, no warnings are issued.

OUTDIR=diagnostic-output-directory

specifies a directory that contains diagnostic files.

Tip Files that are produced include an event log that contains detailed information about the success or failure of the publishing process.

Validation of Publishing Functions

Validating the Publishing of the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF Functions

To validate that the SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, and SAS_DEHEXUDF functions are registered properly under the SASLIB schema in the specified database, follow these steps.

1. Use psql to connect to the database.

```
psql -d databasename
```

You should receive the following prompt.

```
databasename=#
```

2. At the prompt, enter the following command.

```
select prosrc from pg_proc f, pg_namespace s where f.pronamespace=s.oid
and upper(s.nspname)='SASLIB';
```

You should receive a result similar to the following:

```
SAS_CompileUDF
SAS_CopyUDF
SAS_DirectoryUDF
SAS_DehexUDF
```

Validating the Publishing of the SAS_EP Function

To validate that the SAS_EP function is registered properly under the specified schema in the specified database, follow these steps.

1. Use `psql` to connect to the database.

```
psql -d databasename
```

You should receive the following prompt.

```
databasename=#
```

2. At the prompt, enter the following command.

```
select prosrc, probin from pg_catalog.pg_proc where proname = 'sas_ep';
```

You should receive a result similar to the following:

```
SAS_EP | $libdir/SAS/sasep_tablefunc.so
```

3. Exit `psql`.

```
\q
```

Controlling the SAS Embedded Process

The SAS Embedded Process starts when a query is submitted using the SAS_EP function. It continues to run until it is manually stopped or the database is shut down.

Note: Starting and stopping the SAS Embedded Process has implications for all scoring model publishers.

Note: Manually starting and stopping the SAS Embedded Process requires superuser permissions and must be done from the Greenplum master node.

When the SAS Embedded Process is installed, the `ShutdownSASEP.sh` and `StartupSASEP.sh` scripts are installed in the following directory. For more information about these files, see [“Moving and Installing the SAS Embedded Process” on page 180](#).

```
/path_to_sh_file/SAS/SASTKInDatabaseServerForGreenplum/9.43
```

Use the following command to shut down the SAS Embedded Process.

```
/path_to_sh_file/SAS/SASTKInDatabaseServerForGreenplum/9.4e/ShutdownSASEP.sh
<-quiet>
```

When invoked from the master node, `ShutdownSASEP.sh` shuts down the SAS Embedded Process on each database node. The `-verbose` option is on by default and provides a status of the shutdown operations as they occur. You can specify the `-quiet` option to suppress messages. This script should not be used as part of the normal operation. It is designed to be used to shut down the SAS Embedded Process prior to a database upgrade or re-install.

Use the following command to start the SAS Embedded Process.

```
/path_to_sh_file/SAS/SASTKInDatabaseServerForGreenplum/9.43/StartupSASEP.sh
<-quiet>
```

When invoked from the master node, StartupSASEP.sh manually starts the SAS Embedded Process on each database node. The *-verbose* option is on by default and provides a status of the installation as it occurs. You can specify the *-quiet* option to suppress messages. This script should not be used as part of the normal operation. It is designed to be used to manually start the SAS Embedded Process and only after consultation with SAS Technical Support.

CAUTION:

The timing option must be off for any of the .sh scripts to work. Put `\timing off` in your `.psqlrc` file before running these scripts.

Semaphore Requirements When Using the SAS Embedded Process for Greenplum

Each time a query using a SAS_EP table function is invoked to execute a score, it requests a set of semaphore arrays (sometimes referred to as semaphore "sets") from the operating system. The SAS Embedded Process releases the semaphore arrays back to the operating system after scoring is complete.

The number of semaphore arrays required for a given SAS Embedded Process execution is a function of the number of Greenplum database segments that are engaged for the query. The Greenplum system determines the number of segments to engage as part of its query plan based on a number of factors, including the data distribution across the appliance.

The SAS Embedded Process requires five semaphore arrays per database segment that is engaged. The maximum number of semaphore arrays required per database host node per SAS Embedded Process execution can be determined by the following formula:

$$\text{maximum_number_semaphore_arrays} = 5 * \text{number_database_segments}$$

Here is an example. On a full-rack Greenplum appliance configured with 16 host nodes and six database segment servers per node, a maximum of 30 ($5 * 6$) semaphore arrays are required on each host node per concurrent SAS Embedded Process execution of a score. If the requirement is to support the concurrent execution by the SAS Embedded Process of 10 scores, then the SAS Embedded Process requires a maximum of 300 ($5 * 6 * 10$) semaphore arrays on each host node.

SAS recommends that you configure the semaphore array limit on the Greenplum appliance to support twice the limit that is configured by default on the appliance. For example, if the default limit is 2048, double the default limit to 4096.

Note: The semaphore limit discussed here is the limit on the number of "semaphore arrays", where each semaphore array is allocated with an application-specified number of semaphores. For the SAS Embedded Process, the limit on the number of semaphore arrays is distinct from the limit on the "maximum number of semaphores system wide". The SAS Embedded Process requests semaphore arrays with two or fewer semaphores in each array. The limit on the maximum semaphores system wide should not need to be increased. The Linux `$ ipcs -sl` command output shows the typical default semaphore-related limits set on a Greenplum appliance:

```
----- Semaphore Limits -----
max number of arrays = 2048
max semaphores per array = 250
```

```
max semaphores system wide = 512000
max ops per semop call = 100
semaphore max value = 32767
```

Greenplum Permissions

To publish the utility (SAS_COMPILEUDF, SAS_COPYUDF, SAS_DIRECTORYUDF, SAS_DEHEXUDF, SAS_EP), format, and scoring model functions, Greenplum requires that you have superuser permissions to create and execute these functions in the SASLIB (or other specified) schema and in the specified database.

In addition to Greenplum superuser permissions, you must have CREATE TABLE permission to create a model table when using the SAS Embedded Process.

If you plan to use SAS Model Manager with the SAS Scoring Accelerator for in-database scoring, additional permissions are required. For more information, see [Chapter 24, “Configuring SAS Model Manager,”](#) on page 231.

Documentation for Using In-Database Processing in Greenplum

For information about how to publish SAS formats and scoring models, see the *SAS In-Database Products: User's Guide*, located at <http://support.sas.com/documentation/onlinedoc/indbtech/index.html>.

For information about how to use the SAS In-Database Code Accelerator, see the *SAS DS2 Language Reference*, located at <http://support.sas.com/documentation/onlinedoc/base/index.html>.

Chapter 20

Administrator's Guide for Netezza

In-Database Deployment Package for Netezza	193
Prerequisites	193
Overview of the In-Database Deployment Package for Netezza	193
Function Publishing Process in Netezza	194
Netezza Installation and Configuration	195
Netezza Installation and Configuration Steps	195
Upgrading from or Reinstalling a Previous Version	195
Installing the SAS Formats Library, Binary Files, and the SAS Embedded Process	197
Running the %INDNZ_PUBLISH_JAZLIB Macro	199
Running the %INDNZ_PUBLISH_COMPILEUDF Macro	202
Netezza Permissions	205
Documentation for Using In-Database Processing in Netezza	207

In-Database Deployment Package for Netezza

Prerequisites

SAS Foundation and the SAS/ACCESS Interface to Netezza must be installed before you install and configure the in-database deployment package for Netezza.

The SAS Scoring Accelerator for Netezza and the SAS Embedded Process require a specific version of the Netezza client and server environment. For more information, see the SAS Foundation system requirements documentation for your operating environment.

Overview of the In-Database Deployment Package for Netezza

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

The in-database deployment package for Netezza must be installed and configured before you can perform the following tasks:

- Use the %INDNZ_PUBLISH_FORMATS format publishing macro to create or publish the SAS_PUT() function and to create or publish user-defined formats as format functions inside the database.
- Use the %INDNZ_PUBLISH_MODEL scoring publishing macro to create scoring model functions inside the database.

For more information about using the format and scoring publishing macros, see the *SAS In-Database Products: User's Guide*.

The in-database deployment package for Netezza contains the SAS formats library, two pre-compiled binaries for utility functions, and the SAS Embedded Process.

The SAS formats library is a run-time library that is installed on your Netezza system. This installation is made so that the SAS scoring model functions and the SAS_PUT() function can access the routines within the run-time library. The SAS formats library contains the formats that are supplied by SAS.

The %INDNZ_PUBLISH_JAZLIB macro registers the SAS formats library. The %INDNZ_PUBLISH_COMPILEUDF macro registers a utility function in the database. The utility function is then called by the format and scoring publishing macros. You must run these two macros before you run the format and scoring publishing macros.

The SAS Embedded Process is a SAS server process that runs within Netezza to read and write data. The SAS Embedded Process contains macros, run-time libraries, and other software that is installed on your Netezza system. These installations are done so that the SAS scoring files created in Netezza can access routines within the SAS Embedded Process run-time libraries.

Function Publishing Process in Netezza

To publish the SAS scoring model functions, the SAS_PUT() function, and format functions on Netezza systems, the format and scoring publishing macros perform the following tasks:

- Create and transfer the files, using the Netezza External Table interface, to the Netezza server.

Using the Netezza External Table interface, the source files are loaded from the client to a database table through remote ODBC. The source files are then exported to files (external table objects) on the host. Before transfer, each source file is divided into 32K blocks and converted to hexadecimal values to avoid problems with special characters, such as line feed or quotation marks. After the files are exported to the host, the source files are converted back to text.

- Compile those source files into object files using a Netezza compiler.
- Link with the SAS formats library.
- Register those object files with the Netezza system.

Note: This process is valid only when using publishing formats and scoring functions. It is not applicable to the SAS Embedded Process. If you use the SAS Embedded Process, the scoring publishing macro creates the scoring files and uses the SAS/ACCESS Interface to Netezza to insert the scoring files into a model table.

Netezza Installation and Configuration

Netezza Installation and Configuration Steps

1. If you are upgrading from or reinstalling a previous version, follow the instructions in [“Upgrading from or Reinstalling a Previous Version”](#) on page 195.
2. Install the in-database deployment package.
For more information, see [“Installing the SAS Formats Library, Binary Files, and the SAS Embedded Process”](#) on page 197.
3. Run the %INDNZ_PUBLISH_JAZLIB macro to publish the SAS formats library as an object.
For more information, see [“Running the %INDNZ_PUBLISH_JAZLIB Macro”](#) on page 199.
4. Run the %INDNZ_PUBLISH_COMPILEUDF macro.
For more information, see [“Running the %INDNZ_PUBLISH_COMPILEUDF Macro”](#) on page 202.
5. If you plan to use SAS Model Manager with the SAS Scoring Accelerator for in-database scoring, perform the additional configuration tasks in [Chapter 24, “Configuring SAS Model Manager,”](#) on page 231.

Upgrading from or Reinstalling a Previous Version

Overview of Upgrading from or Reinstalling a Previous Version

You can upgrade from or reinstall a previous version of the SAS Formats Library and binary files, the SAS Embedded Process, or both. See the following topics:

- If you want to upgrade or reinstall a previous version of the SAS Formats Library and binary files, see [“Upgrading from or Reinstalling the SAS Formats Library and Binary Files”](#) on page 195.
- If you want to upgrade or reinstall a previous version of the SAS Embedded Process, see [“Upgrading from or Reinstalling the SAS Embedded Process”](#) on page 196.

Upgrading from or Reinstalling the SAS Formats Library and Binary Files

To upgrade from or reinstall a previous version of the SAS Formats Library and binary files, follow these steps.

Note: These steps apply if you want to upgrade from or reinstall only the SAS Formats Library and binary files. If you want to upgrade from or reinstall the SAS Embedded Process, see [“Upgrading from or Reinstalling the SAS Embedded Process”](#) on page 196.

1. Run the %INDNZ_PUBLISH_JAZLIB macro with ACTION=DROP to remove the SAS formats library as an object.
For more information, see [“Running the %INDNZ_PUBLISH_JAZLIB Macro”](#) on page 199.

2. Run the %INDNZ_PUBLISH_COMPILEUDF macro with ACTION=DROP to remove the SAS_COMPILEUDF, SAS_DIRECTORYUDF, and SAS_HEXTOTEXTUDF functions.

For more information, see [“Running the %INDNZ_PUBLISH_COMPILEUDF Macro” on page 202](#).

3. Navigate to the /nz/extensions/SAS directory and delete the SASFormatsLibraryForNetezza directory.

Note: Under the SAS directory, the installer for the SAS Formats Library and binary files and the SAS Embedded Process installer both create a directory under the SAS directory. These directories are named SASFormatsLibraryForNetezza and SASTKInDatabaseServerForNetezza, respectively. If you delete everything under the SAS directory, the SAS Embedded Process, the SAS Formats Library, and the binary files are removed. If you want to remove only one, then you must leave the other directory.

4. If you are also upgrading from or reinstalling the SAS Embedded Process, continue the installation instructions in [“Upgrading from or Reinstalling the SAS Embedded Process” on page 196](#). Otherwise, continue the installation instructions in [“Installing the SAS Formats Library, Binary Files, and the SAS Embedded Process” on page 197](#).

Upgrading from or Reinstalling the SAS Embedded Process

To upgrade from or reinstall a previous version of the SAS Embedded Process, follow these steps.

Note: These steps are for upgrading from or reinstalling only the SAS Embedded Process. If you want to upgrade from or reinstall the SAS Formats Library and binary files, you must follow the steps in [“Upgrading from or Reinstalling the SAS Formats Library and Binary Files” on page 195](#).

1. Check the current installed version of the SAS Embedded Process.

```
nzcm --installed
```

2. Enter these commands to unregister and uninstall the SAS Embedded Process.

```
nzcm -u SASTKInDatabaseServerForNetezza
nzcm -e SASTKInDatabaseServerForNetezza
```

3. Navigate to the /nz/extensions/SASTKInDatabaseServerForNetezza directory and verify that the directory is empty.

Note: Under the SAS directory, the installer for the SAS Formats Library and binary files and the SAS Embedded Process installer both create a directory under the SAS directory. These directories are named SASFormatsLibraryForNetezza and SASTKInDatabaseServerForNetezza, respectively. If you delete everything under the SAS directory, the SAS Embedded Process, the SAS Formats Library, and the binary files are removed. If you want to remove only one, then you must leave the other directory.

4. Continue the installation instructions in [“Installing the SAS Formats Library, Binary Files, and the SAS Embedded Process” on page 197](#).

Installing the SAS Formats Library, Binary Files, and the SAS Embedded Process

Moving and Installing the SAS Formats Library and Binary Files

The SAS formats library and the binary files for the SAS_COMPILEUDF function are contained in a self-extracting archive file. The self-extracting archive file is located in the **SAS-installation-directory/SASFormatsLibraryForNetezza/3.1/Netezza32bitTwinFin/** directory.

To move and unpack the self-extracting archive file, follow these steps:

1. Using a method of your choice, transfer the `accelnetzfnt-3.1-n_lax.sh` to your Netezza system.

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.
2. After the `accelnetzfnt-3.1-n_lax.sh` file has been transferred to the Netezza machine, log on as the user who owns the Netezza software (usually the “nz” ID).
3. Use the following commands at the UNIX prompt to unpack the self-extracting archive file.

```
mkdir -p /nz/extensions
chmod 755 /nz/extensions
cd /nz/extensions
chmod 755 path_to_sh_file/accelnetzfnt-3.1-n_lax.sh
path_to_sh_file/accelnetzfnt-3.1-n_lax.sh
```

path_to_sh_file is the location to which you copied the self-extracting archive file in Step 1.

After the script runs and the files are unpacked, the target directories should look similar to these.

```
/nz/extensions/SAS/SASFormatsLibraryForNetezza/3.1-n/bin/InstallAccelNetzFmt.sh
/nz/extensions/SAS/SASFormatsLibraryForNetezza/3.1-n/lib/SAS_CompileUDF.o_spu10
/nz/extensions/SAS/SASFormatsLibraryForNetezza/3.1-n/lib/SAS_CompileUDF.o_x86
/nz/extensions/SAS/SASFormatsLibraryForNetezza/3.1-n/lib/libjazxfbrs_spu10.so
/nz/extensions/SAS/SASFormatsLibraryForNetezza/3.1-n/lib/libjazxfbrs_x86.so
```

There also is a symbolic link such that ***/nz/extensions/SAS/SASFormatsLibraryForNetezza/3.1*** points to the latest version.

Moving and Installing the SAS Embedded Process

The SAS Embedded Process is contained in a self-extracting archive file. The self-extracting archive file is located in the **SAS-installation-directory/SASTKInDatabaseServer/9.4/Netezza64bitTwinFin/** directory.

To move and unpack the self-extracting archive file to create a Netezza cartridge file, follow these steps:

1. Using a method of your choice, transfer the `tkindbsrv-9.43-n_lax.sh` to any directory on the Netezza host machine.

n is a number that indicates the latest version of the file.
2. After the `tkindbsrv-9.43-n_lax.sh` file has been transferred to the Netezza, log on as the user who owns the Netezza appliance (usually the “nz” ID).

3. If you have a database named SAS_EP, you should rename it.

When you unpack the self-extracting archive file, a SAS_EP database that contains SAS Embedded Process function is created. The creation of the SAS_EP database overwrites any existing database that is named SAS_EP.

4. Unpack the self-extracting archive file and create a Netezza cartridge file.

- a. Change to the directory where you put the tkindbsrv.sh file.

```
cd path_to_sh_file
```

path_to_sh_file is the location to which you copied the self-extracting archive file in Step 1.

- b. Use the following command at the UNIX prompt to unpack the self-extracting archive file.

```
./tkindbsrv-9.43-n_lax.sh
```

After the script runs, the tkindbsrv-9.43-n_lax.sh file goes away, and the SASTKInDatabaseServerForNetezza-9.4.1.n.nzc Netezza cartridge file is created in its place.

5. Use these **nzcm** commands to install and register the sas_ep cartridge.

```
nzcm -i sas_ep
nzcm -r sas_ep
```

Note: The sas_ep cartridge creates the NZRC database. The NZRC database contains remote controller functions that are required by the SAS Embedded Process. The sas_ep cartridge is available on the Netezza website. For access to the sas_ep cartridge, contact your local Netezza representative.

6. Use these **nzcm** commands to install and register the SAS Embedded Process.

```
nzcm -i SASTKInDatabaseServerForNetezza-9.43.0.n.nzc
nzcm -r SASTKInDatabaseServerForNetezza
```

Note: The installation of the SAS Embedded Process is dependent on the sas_ep cartridge that is supplied by Netezza.

For more NZCM commands, see “[NZCM Commands for the SAS Embedded Process](#)” on page 198.

NZCM Commands for the SAS Embedded Process

The following table lists and describes the NZCM commands that you can use with the SAS Embedded Process.

Command	Action performed
nzcm -help	Displays help for NZCM commands
nzcm --installed nzcm - i	Displays the filename (SASTKInDatabaseServerForNetezza) and the version number that is installed
nzcm --registered nzcm - r	Displays the filename (SASTKInDatabaseServerForNetezza) and the version number that is registered

Command	Action performed
<pre>nzcm --unregister SASTKInDatabaseServerForNetezza nzcm -u SASTKInDatabaseServerForNetezza</pre>	Unregisters the SAS Embedded Process
<pre>nzcm --unregister sas_ep nzcm -u sas_ep</pre>	Unregisters the sas_ep cartridge <i>Note:</i> The sas_ep cartridge is installed only once. It does not need to be unregistered or uninstalled when the SAS Embedded Process is upgraded or reinstalled. The sas_ep cartridge needs to be unregistered and uninstalled only when Netezza changes the cartridge version.
<pre>nzcm --uninstall SASTKInDatabaseServerForNetezza nzcm -e SASTKInDatabaseServerForNetezza</pre>	Uninstalls the SAS Embedded Process
<pre>nzcm --uninstall sas_ep nzcm -e sas_ep</pre>	Uninstalls the sas_ep cartridge <i>Note:</i> The sas_ep cartridge is installed only once. It does not need to be unregistered or uninstalled when the SAS Embedded Process is upgraded or reinstalled. The sas_ep cartridge needs to be unregistered and uninstalled only when Netezza changes the cartridge version.
<pre>nzcm --install SASTKInDatabaseServerForNetezza-9.43.0.n.nzc nzcm -i SASTKInDatabaseServerForNetezza-9.43.0.n.nzc</pre>	Installs the SAS Embedded Process
<pre>nzcm --install sas_ep nzcm -i sas_ep</pre>	Installs the sas_ep cartridge
<pre>nzcm --register SASTKInDatabaseServerForNetezza nzcm -r SASTKInDatabaseServerForNetezza</pre>	Registers the SAS Embedded Process
<pre>nzcm --register sas_ep nzcm -register sas_ep</pre>	Registers the sas_ep cartridge

Running the %INDNZ_PUBLISH_JAZLIB Macro

Overview of Publishing the SAS Formats Library

The SAS formats library is a shared library and must be published and registered as an object in the Netezza database. The library is linked to the scoring and format publishing macros through a DEPENDENCIES statement when the scoring model functions or formats are created.

You must run the %INDNZ_PUBLISH_JAZLIB macro to publish and register the SAS formats library. The %INDNZ_PUBLISH_JAZLIB macro publishes and registers the SAS formats library in the database as the **sas_jazlib** object.

%INDNZ_PUBLISH_JAZLIB Macro Run Process

To run the %INDNZ_PUBLISH_JAZLIB macro, follow these steps:

1. Start SAS and submit the following command in the Enhanced Editor or Program Editor:

```
%let indconn=SERVER=yourservername USER=youruserid PW=yourpwd DB=database;
```

For more information, see the “[INDCONN Macro Variable](#)” on page 200.

2. Run the %INDNZ_PUBLISH_JAZLIB macro. For more information, see “[%INDNZ_PUBLISH_JAZLIB Macro Syntax](#)” on page 201.

INDCONN Macro Variable

The INDCONN macro variable is used to provide credentials to connect to Netezza. You must specify server, user, password, and database information to access the machine on which you have installed the Netezza data warehouse. You must assign the INDCONN macro variable before the %INDNZ_PUBLISH_JAZLIB macro is invoked.

The value of the INDCONN macro variable for the %INDNZ_PUBLISH_JAZLIB macro has this format:

```
SERVER=<'>server<'> USER=<'>userid<'> PASSWORD=<'>password<'>
DATABASE=<'>database<'> SCHEMA=<'>schema-name<'>
```

SERVER=<'>server<'>

specifies the server name or IP address of the server to which you want to connect. This server accesses the database that contains the tables and views that you want to access. If the server name contains spaces or nonalphanumeric characters, enclose the server name in quotation marks.

USER=<'>userid<'>

specifies the Netezza user name (also called the user ID) that you use to connect to your database. If the user name contains spaces or nonalphanumeric characters, enclose the user name in quotation marks.

PASSWORD=<'>password<'>

specifies the password that is associated with your Netezza user name. If the password contains spaces or nonalphanumeric characters, enclose the password in quotation marks.

Tip Use only PASSWORD=, PASS=, or PW= for the password argument. PWD= is not supported and causes an error.

DATABASE=<'>database<'>

specifies the name of the database on the server that contains the tables and views that you want to access. If the database name contains spaces or nonalphanumeric characters, enclose the database name in quotation marks.

Interaction The database that is specified by the %INDNZ_PUBLISH_JAZLIB macro's DATABASE= argument takes precedence over the database that you specify in the INDCONN macro variable. If you do not specify a value for DATABASE= in either the INDCONN macro variable or the %INDNZ_PUBLISH_JAZLIB macro, the default value of SASLIB is used. For more information, see “[%INDNZ_PUBLISH_JAZLIB Macro Syntax](#)” on page 201.

Tip The object name for the SAS formats library is **sas_jazlib**.

SCHEMA=<'>schema-name<'>

specifies the name of the schema where the SAS formats library is published.

Restriction This argument is supported only on Netezza v7.0.3 or later.

Interaction The schema that is specified by the %INDNZ_PUBLISH_JAZLIB macro's DBSCHEMA= argument takes precedence over the schema that you specify in the INDCONN macro variable. If you do not specify a schema in the DBSCHEMA= argument or the INDCONN macro variable, the default schema for the target database is used.

%INDNZ_PUBLISH_JAZLIB Macro Syntax

%INDNZ_PUBLISH_JAZLIB

```
( <DATABASE=database>
  <, DBSCHEMA=schema-name>
  <, ACTION=CREATE | REPLACE | DROP>
  <, OUTDIR=diagnostic-output-directory>
);
```

Arguments

DATABASE=database

specifies the name of a Netezza database to which the SAS formats library is published as the **sas_jazlib** object.

Default SASLIB

Interaction The database that is specified by the DATABASE= argument takes precedence over the database that you specify in the INDCONN macro variable.

Tip The object name for the SAS formats library is **sas_jazlib**.

DBSCHEMA=schema-name

specifies the name of a Netezza schema to which the SAS formats library is published.

Restrictions This argument is supported only on Netezza v7.0.3 or later.

This argument is supported only on Netezza v7.0.3 or later.

Interaction The schema that is specified by the DBSCHEMA= argument takes precedence over the schema that you specify in the INDCONN macro variable. If you do not specify a schema in the DBSCHEMA= argument or the INDCONN macro variable, the default schema for the target database is used.

ACTION=CREATE | REPLACE | DROP

specifies that the macro performs one of the following actions:

CREATE

creates a new SAS formats library.

REPLACE

overwrites the current SAS formats library, if a SAS formats library by the same name is already registered, or creates a new SAS formats library if one is not registered.

DROP

causes the SAS formats library to be dropped from the Netezza database.

Default CREATE

Tip If the SAS formats library was published previously and you specify ACTION=CREATE, you receive warning messages that the library already exists. You are prompted to use REPLACE. If you specify ACTION=DROP and the SAS formats library does not exist, you receive an error message.

OUTDIR=*diagnostic-output-directory*

specifies a directory that contains diagnostic files.

Tip Files that are produced include an event log that contains detailed information about the success or failure of the publishing process.

Running the %INDNZ_PUBLISH_COMPILEUDF Macro

Overview of the %INDNZ_PUBLISH_COMPILEUDF Macro

The %INDNZ_PUBLISH_COMPILEUDF macro creates three functions:

- **SAS_COMPILEUDF.** This function facilitates the scoring and format publishing macros. The SAS_COMPILEUDF function compiles the scoring model and format source files into object files. This compilation uses a Netezza compiler and occurs through the SQL interface.
- **SAS_DIRECTORYUDF and SAS_HEXTOTEXTUDF.** These functions are used when the scoring and format publishing macros transfer source files from the client to the host using the Netezza External Tables interface. SAS_DIRECTORYUDF creates and deletes temporary directories on the host. SAS_HEXTOTEXTUDF converts the files from hexadecimal back to text after the files are exported on the host. For more information about the file transfer process, see [“Function Publishing Process in Netezza” on page 194](#).

You have to run the %INDNZ_PUBLISH_COMPILEUDF macro only one time.

The SAS_COMPILEUDF, SAS_DIRECTORYUDF, and SAS_HEXTOTEXTUDF functions must be published before the %INDNZ_PUBLISH_FORMATS or %INDNZ_PUBLISH_MODEL macros are run. Otherwise, these macros fail.

Note: To publish the SAS_COMPILEUDF, SAS_DIRECTORYUDF, and SAS_HEXTOTEXTUDF functions, you must have the appropriate Netezza user permissions to create these functions in either the SASLIB database (default) or in the database that is used in lieu of SASLIB. For more information, see [“Netezza Permissions” on page 205](#).

%INDNZ_PUBLISH_COMPILEUDF Macro Run Process

To run the %INDNZ_PUBLISH_COMPILEUDF macro to publish the SAS_COMPILEUDF, SAS_DIRECTORYUDF, and SAS_HEXTOTEXTUDF functions, follow these steps:

1. Create either a SASLIB database or a database to be used in lieu of the SASLIB database.

This database is where the SAS_COMPILEUDF, SAS_DIRECTORYUDF, and SAS_HEXTOTEXTUDF functions are published. You specify this database in the DATABASE argument of the %INDNZ_PUBLISH_COMPILEUDF macro. For more information about how to specify the database that is used in lieu of SASLIB, see [“%INDNZ_PUBLISH_COMPILEUDF Macro Run Process” on page 202](#).

2. Start SAS and submit the following command in the Enhanced Editor or Program Editor.

```
%let indconn = server=yourserver user=youruserid password=yourpwd
database=database;
```

For more information, see the “[INDCONN Macro Variable](#)” on page 203.

3. Run the %INDNZ_PUBLISH_COMPILEUDF macro. For more information, see “[%INDNZ_PUBLISH_COMPILEUDF Macro Syntax](#)” on page 204.

After the SAS_COMPILEUDF function is published, the model or format publishing macros can be run to publish the scoring model or format functions.

INDCONN Macro Variable

The INDCONN macro variable provides the credentials to make a connection to Netezza. You must specify the server, user, password, and database information to access the machine on which you have installed the Netezza database. You must assign the INDCONN macro variable before the %INDNZ_PUBLISH_COMPILEUDF macro is invoked.

The value of the INDCONN macro variable for the %INDNZ_PUBLISH_COMPILEUDF macro has this format.

```
SERVER=<'>server<'> USER=<'>userid<'> PASSWORD=<'>password<'>
DATABASE=SASLIB | <'>database<'> SCHEMA=<'>schema-name<'>
```

SERVER=<'>server<'>

specifies the server name or IP address of the server to which you want to connect. This server accesses the database that contains the tables and views that you want to access. If the server name contains spaces or nonalphanumeric characters, enclose the server name in quotation marks.

USER=<'>userid<'>

specifies the Netezza user name (also called the user ID) that you use to connect to your database. If the user name contains spaces or nonalphanumeric characters, enclose the user name in quotation marks.

PASSWORD=<'>password<'>

specifies the password that is associated with your Netezza user name. If the password contains spaces or nonalphanumeric characters, enclose the password in quotation marks.

Tip Use only PASSWORD=, PASS=, or PW= for the password argument. PWD= is not supported and causes an error.

DATABASE=SASLIB | <'>database<'>

specifies the name of the database on the server that contains the tables and views that you want to access. If the database name contains spaces or nonalphanumeric characters, enclose the database name in quotation marks.

Default	SASLIB
----------------	--------

Interactions	The database that is specified by the %INDNZ_PUBLISH_COMPILEUDF macro's DATABASE= argument takes precedence over the database that you specify in the INDCONN macro variable. If you do not specify a value for DATABASE= in either the INDCONN macro variable or the %INDNZ_PUBLISH_COMPILEUDF macro, the default value of
---------------------	---

SASLIB is used. For more information, see [“%INDNZ_PUBLISH_COMPILEUDF Macro Syntax” on page 204](#).

If the SAS_COMPILEUDF function is published in a database other than SASLIB, then that database name should be used instead of SASLIB for the DBCOMPILE argument in the %INDNZ_PUBLISH_FORMATS and %INDNZ_PUBLISH_MODEL macros. Otherwise, the %INDNZ_PUBLISH_FORMATS and %INDNZ_PUBLISH_MODEL macros fail when calling the SAS_COMPILEUDF function during the publishing process. If a database name is not specified, the default is SASLIB. For documentation on the %INDNZ_PUBLISH_FORMATS and %INDNZ_PUBLISH_MODEL macros, see [“Documentation for Using In-Database Processing in Netezza” on page 207](#).

SCHEMA=<'>schema-name<'>

specifies the name of the schema where the SAS_COMPILEUDF, SAS_DIRECTORYUDF, and SAS_HEXTOTEXTUDF functions are published.

Restriction This argument is supported only on Netezza v7.0.3 or later.

Interaction The schema that is specified by the %INDNZ_PUBLISH_COMPILEUDF macro's DBSCHEMA= argument takes precedence over the schema that you specify in the INDCONN macro variable. If you do not specify a schema in the DBSCHEMA= argument or the INDCONN macro variable, the default schema for the target database is used.

%INDNZ_PUBLISH_COMPILEUDF Macro Syntax

%INDNZ_PUBLISH_COMPILEUDF

```
(<DATABASE=database-name>
<, DBSCHEMA=schema-name>
<, ACTION=CREATE | REPLACE | DROP>
<, OUTDIR=diagnostic-output-directory>
);
```

Arguments

DATABASE=database-name

specifies the name of a Netezza database to which the SAS_COMPILEUDF is published.

Default SASLIB

Interaction The database that is specified by the DATABASE= argument takes precedence over the database that you specify in the INDCONN macro variable. For more information, see [“INDCONN Macro Variable” on page 203](#).

DBSCHEMA=schema-name

specifies the name of a Netezza schema to which the SAS_COMPILEUDF function is published.

Restriction This argument is supported only on Netezza v7.0.3 or later.

Interaction The schema that is specified by the DBSCHEMA= argument takes precedence over the schema that you specify in the INDCONN macro

variable. If you do not specify a schema in the DBSCHEMA= argument or the INDCONN macro variable, the default schema for the target database is used.

ACTION=CREATE | REPLACE | DROP

specifies that the macro performs one of the following actions:

CREATE

creates a new SAS_COMPILEUDF function.

REPLACE

overwrites the current SAS_COMPILEUDF function, if a SAS_COMPILEUDF function by the same name is already registered, or creates a new SAS_COMPILEUDF function if one is not registered.

DROP

causes the SAS_COMPILEUDF function to be dropped from the Netezza database.

Default CREATE

Tip If the SAS_COMPILEUDF function was published previously and you specify ACTION=CREATE, you receive warning messages that the function already exists and be prompted to use REPLACE. If you specify ACTION=DROP and the SAS_COMPILEUDF function does not exist, you receive an error message.

OUTDIR=*diagnostic-output-directory*

specifies a directory that contains diagnostic files.

Tip Files that are produced include an event log that contains detailed information about the success or failure of the publishing process.

Netezza Permissions

There are three sets of permissions involved with the in-database software.

- The first set of permissions is needed by the person who publishes the SAS formats library and the SAS_COMPILEUDF, SAS_DIRECTORYUDF, and SAS_HEXTOTEXTUDF functions. These permissions must be granted before the %INDNZ_PUBLISH_JAZLIB and %INDNZ_PUBLISH_COMPILEUDF macros are run. Without these permissions, running these macros fails.

The following table summarizes the permissions that are needed by the person who publishes the formats library and the functions.

Permission Needed	Authority Required to Grant Permission	Examples
CREATE LIBRARY permission to run the %INDNZ_PUBLISH_JAZLIB macro that publishes the SAS formats library (sas_jazlib object)	System Administrator or Database Administrator <i>Note:</i> If you have SYSADM or DBADM authority, then you have these permissions. Otherwise, contact your database administrator to obtain these permissions.	GRANT CREATE LIBRARY TO <i>fmtlibpublisherid</i>
CREATE FUNCTION permission to run the %INDNZ_PUBLISH_COMPILEUDF macro that publishes the SAS_COMPILEUDF, SAS_DIRECTORYUDF, and the SAS_HEXTOTEXTUDF functions		GRANT CREATE FUNCTION TO <i>compileudfpublisherid</i>

- The second set of permissions is needed by the person who runs the format publishing macro, %INDNZ_PUBLISH_FORMATS, or the scoring publishing macro, %INDNZ_PUBLISH_MODEL. The person who runs these macros is not necessarily the same person who runs the %INDNZ_PUBLISH_JAZLIB and %INDNZ_PUBLISH_COMPILEUDF macros. These permissions are most likely needed by the format publishing or scoring model developer. Without these permissions, the publishing of the scoring model functions and the SAS_PUT() function and formats fails.

Note: Permissions must be granted for every format and scoring model publisher and for each database that the format and scoring model publishing uses. Therefore, you might need to grant these permissions multiple times. After the Netezza permissions are set appropriately, the format and scoring publishing macros can be run.

Note: When permissions are granted to specific functions, the correct signature, including the sizes for numeric and string data types, must be specified.

The following table summarizes the permissions that are needed by the person who runs the format or scoring publishing macro.

Permission Needed	Authority Required to Grant Permission	Examples
EXECUTE permission for the SAS Formats Library	System Administrator or Database Administrator	GRANT EXECUTE ON SAS_JAZLIB TO <i>scoringorfmtpublisherid</i>
EXECUTE permission for the SAS_COMPILEUDF function	<i>Note:</i> If you have SYSADM or DBADM authority, then you have these permissions. Otherwise, contact your database administrator to obtain these permissions.	GRANT EXECUTE ON SAS_COMPILEUDF TO <i>scoringorfmtpublisherid</i>
EXECUTE permission for the SAS_DIRECTORYUDF function		GRANT EXECUTE ON SAS_DIRECTORYUDF TO <i>scoringorfmtpublisherid</i>
EXECUTE permission for the SAS_HEXTOTEXTUDF function		GRANT EXECUTE ON SAS_HEXTOTEXTUDF TO <i>scoringorfmtpublisherid</i>
CREATE FUNCTION, CREATE TABLE, CREATE TEMP TABLE, and CREATE EXTERNAL TABLE permissions to run the format and scoring publishing macros		GRANT CREATE FUNCTION TO <i>scoringorfmtpublisherid</i> GRANT CREATE TABLE TO <i>scoringorfmtpublisherid</i> GRANT CREATE TEMP TABLE TO <i>scoringorfmtpublisherid</i> GRANT CREATE EXTERNAL TABLE TO <i>scoringorfmtpublisherid</i> GRANT UNFENCED TO <i>scoringorfmtpublisherid</i>

- The third set of permissions is needed by the person who runs the SAS Embedded Process to create scoring files.

The SAS Embedded Process has a dependency on the IBM Netezza Analytics (INZA) utility. You must grant the user and database permissions using these commands.

```
/nz/export/ae/utilities/bin/create_inza_db_user.sh user-name database-name
/nz/export/ae/utilities/bin/create_inza_db.sh database-name
```

Note: If you plan to use SAS Model Manager with the SAS Scoring Accelerator for in-database scoring, additional permissions are required. For more information, see [Chapter 24, “Configuring SAS Model Manager,” on page 231](#).

Documentation for Using In-Database Processing in Netezza

For information about how to publish SAS formats, the SAS_PUT() function, and scoring models, see the *SAS In-Database Products: User's Guide*, located at <http://support.sas.com/documentation/onlinedoc/indbtech/index.html>.

Chapter 21

Administrator's Guide for Oracle

In-Database Deployment Package for Oracle	209
Prerequisites	209
Overview of the In-Database Package for Oracle	209
Oracle Installation and Configuration	210
Installing and Configuring Oracle	210
Upgrading from or Reinstalling a Previous Version	210
Installing the In-Database Deployment Package for Oracle	211
Creating Users and Objects for the SAS Embedded Process	212
Oracle Permissions	213
Documentation for Using In-Database Processing in Oracle	213

In-Database Deployment Package for Oracle

Prerequisites

SAS Foundation and the SAS/ACCESS Interface to Oracle must be installed before you install and configure the in-database deployment package for Oracle.

The SAS Scoring Accelerator for Oracle requires a specific version of the Oracle client and server environment. For more information, see the SAS Foundation system requirements documentation for your operating environment.

Overview of the In-Database Package for Oracle

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

The in-database deployment package for Oracle must be installed and configured before you perform the following tasks:

- Use the %INDOR_PUBLISH_MODEL scoring publishing macro to create scoring files inside the database.
- Run SAS High-Performance Analytics when the analytics cluster is using a parallel connection with a remote Oracle Exadata appliance. The SAS Embedded Process, which resides on the data appliance, is used to provide high-speed parallel data

transfer between the data appliance and the analytics environment where it is processed.

For more information, see the *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*.

For more information about using the scoring publishing macros, see the *SAS In-Database Products: User's Guide*.

The in-database deployment package for Oracle includes the SAS Embedded Process. The SAS Embedded Process is a SAS server process that runs within Oracle to read and write data. The SAS Embedded Process contains macros, run-time libraries, and other software that are installed on your Oracle system. The software is installed so that the SAS scoring files created in Oracle can access the routines within the SAS Embedded Process's run-time libraries.

Oracle Installation and Configuration

Installing and Configuring Oracle

To install and configure Oracle, follow these 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 210](#) before installing the in-database deployment package.
2. Install the in-database deployment package.

For more information, see [“Installing the In-Database Deployment Package for Oracle” on page 211](#).

3. Create the required users and objects in the Oracle server.

For more information, see [“Creating Users and Objects for the SAS Embedded Process” on page 212](#).

4. If you plan to use SAS Model Manager with the SAS Scoring Accelerator for in-database scoring, perform the additional configuration tasks provided in [Chapter 24, “Configuring SAS Model Manager,” on page 231](#).

Note: If you are installing the SAS High-Performance Analytics environment, there are additional steps to be performed 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

You can upgrade from or reinstall a previous version of the SAS Embedded Process. Before installing the In-Database Deployment Package for Oracle, have the database administrator (DBA) notify the user community that there will be an upgrade of the SAS Embedded Process. The DBA should then alter the availability of the database by restricting access, or by bringing the database down. Then, follow the steps outlined in [“Installing the In-Database Deployment Package for Oracle” on page 211](#).

Installing the In-Database Deployment Package for Oracle

Overview

The in-database deployment package for Oracle is contained in a self-extracting archive file named `tkindbsrv-9.43-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.

The self-extracting archive file is located in the `SAS-installation-directory/SASTKInDatabaseServer/9.4/OracleDatabaseonLinuxx64/` directory.

Move the SAS Embedded Process Package to the Oracle Server

To move and copy the Oracle in-database deployment package, follow these steps:

1. Using a method of your choice (for example, PSFTP, SFTP, SCP, or FTP), move the `tkindbsrv-9.43-n_lax.sh` file to directory of your choice. It is suggested that you create a SAS directory under your home directory. An example is `/u01/pochohome/SAS`.
2. Copy the `tkindbsrv-9.43-n_lax.sh` file onto each of the RAC nodes using a method of your choice (for example, DCLI, SFTP, SCP, or FTP).

Note: This might not be necessary. For RAC environments with a shared Oracle Home, you can also use one of these methods:

- Copy the extracted directories from a single node.
- Copy the self-extracting archive file to a directory common to all the nodes.
- If the file system is not a database file system (DBFS), extract the file in one location for the whole appliance.

Unpack the SAS Embedded Process Files

For each node, log on as the owner user for the Oracle software using a secured shell, such as SSH. Follow these steps:

1. Change to the directory where the `tkindbsrv-9.43-n_lax.sh` file is located.
2. If necessary, change permissions on the file to enable you to execute the script and write to the directory.

```
chmod +x tkindbsrv-9.43-n_lax.sh
```

3. Use this command to unpack the self-extracting archive file.

```
./tkindbsrv-9.43-n_lax.sh
```

After this script is run and the files are unpacked, a SAS tree is built in the current directory. The content of the target directories should be similar to the following, depending on the path to your self-extracting archive file. Part of the directory path is shaded to emphasize the different target directories that are used.

```
/path_to_sh_file/SAS/SASTKInDatabaseServerForOracle/9.43/bin
/path_to_sh_file/SAS/SASTKInDatabaseServerForOracle/9.43/misc
/path_to_sh_file/SAS/SASTKInDatabaseServerForOracle/9.43/sasexe
/path_to_sh_file/SAS/SASTKInDatabaseServerForOracle/9.43/utilities
/path_to_sh_file/SAS/SASTKInDatabaseServerForOracle/9.43/admin
/path_to_sh_file/SAS/SASTKInDatabaseServerForOracle/9.43/logs
```

4. On non-shared Oracle home systems, update the contents of the \$ORACLE_HOME/hs/admin/extproc.ora file on each node. On shared Oracle home systems, you can update the file in one location that is accessible by all nodes.

- a. Make a backup of the current extproc.ora file.
- b. Add the following settings to the file making sure to override any previous settings.

```
SET EXTPROC_DLLS=ANY
SET EPPATH=/path_to_sh_file/SAS/SASTKInDatabaseServerForOracle/9.43/
SET TKPATH=/path_to_sh_file/SAS/SASTKInDatabaseServerForOracle/9.43/sasexe
```

Note: Ask your DBA if the ORACLE_HOME environment variable is not set.

5. On non-shared Oracle home systems, update the contents of the \$ORACLE_HOME/network/admin/sqlnet.ora file on each node. On shared Oracle home systems, you can update the file in one location that is accessible by all nodes.

- a. Make a backup of the current sqlnet.ora file. If the file does not exist, create one.
- b. Add the following setting to the file.

```
DIAG_ADR_ENABLED=OFF
```

Creating Users and Objects for the SAS Embedded Process

After the In-Database Deployment Package for Oracle is installed, the DBA must create the users and grant user privileges. The DBA needs to perform these tasks before the SAS administrator can create the objects for the Oracle server. The users and objects are required for the SAS Embedded Process to work.

Note: SQLPLUS or an equivalent SQL tool can be used to submit the SQL statements in this topic.

1. Create a SASADMIN user.

To create the user accounts for Oracle, the DBA must perform the following steps:

- a. Change the directory to `/path_to_sh_file/SAS/SASTKInDatabaseServerForOracle/9.43/admin`.
- b. Connect as SYS, using the following command:

```
sqlplus sys/<password> as sysdba
```

- c. Create and grant user privileges for the SASADMIN user.

Here is an example of how to create a SASADMIN user.

```
CREATE USER SASADMIN IDENTIFIED BY <password>
  DEFAULT TABLESPACE <tablespace-name>
  TEMPORARY TABLESPACE <tablespace-name>;
GRANT UNLIMITED TABLESPACE TO SASADMIN;
```

- d. Submit the following SQL script to grant the required privileges to the SASADMIN user.

```
SQL>@sasadmin_grant_privs.sql
```

- e. Log off from the SQLPLUS session using “Quit” or close your SQL tool.

2. Create the necessary database objects.

To create the objects and the SASEPFUNC table function that are needed to run the scoring model, the SAS administrator (SASADMIN) must perform the following steps:

- a. Change the current directory to `/path_to_sh_file/SAS/SASTKInDatabaseServerForOracle/9.43/admin` (if you are not already there).
- b. Connect as SASADMIN, using the following command:
- c. Submit the following SQL statement:

```
sqlplus sasadmin/<password>
```

```
@create_sasepfunc.sql;
```

Note: You can ignore the following errors:

ORA-00942: table or view does not exist

ORA-01432: public synonym to be dropped does not exist

Oracle Permissions

The person who runs the %INDOR_CREATE_MODELTABLE needs CREATE permission to create the model table. Here is an example.

```
GRANT CREATE TABLE TO userid
```

The person who runs the %INDOR_PUBLISH_MODEL macro needs INSERT permission to load data into the model table. This permission must be granted after the model table is created. Here is an example.

```
GRANT INSERT ON modeltablename TO userid
```

Note: The RESOURCE user privilege that was granted in the previous topic includes the permissions for CREATE, DELETE, DROP, and INSERT.

If you plan to use SAS Model Manager with the SAS Scoring Accelerator for in-database scoring, additional permissions are required. For more information, see [Chapter 24, “Configuring SAS Model Manager,”](#) on page 231.

Documentation for Using In-Database Processing in Oracle

For information about how to publish SAS scoring models, see the *SAS In-Database Products: User's Guide*, located at <http://support.sas.com/documentation/onlinedoc/indbtech/index.html>.

Chapter 22

Administrator's Guide for SAP HANA

In-Database Deployment Package for SAP HANA	215
Prerequisites	215
Overview of the In-Database Deployment Package for SAP HANA	216
SAP HANA Installation and Configuration	216
Installing and Configuring SAP HANA	216
Upgrading or Reinstalling a Previous Version	217
Installing the In-Database Deployment Package for SAP HANA	218
Installing the SASLINK AFL Plugins on the Appliance (HANA SPS08)	219
Installing the SASLINK AFL Plugins on the Appliance (HANA SPS09)	221
Importing the SAS_EP Stored Procedure	222
Auxiliary Wrapper Generator and Eraser Procedures	222
SAP HANA SPS08	222
SAP HANA SPS09	223
Controlling the SAS Embedded Process	223
Semaphore Requirements When Using the SAS Embedded Process for SAP HANA	224
SAP HANA Permissions	224
Documentation for Using In-Database Processing in SAP HANA	225

In-Database Deployment Package for SAP HANA

Prerequisites

SAS Foundation and the SAS/ACCESS Interface to SAP HANA must be installed before you install and configure the in-database deployment package for SAP HANA.

The SAS Scoring Accelerator for SAP HANA and the SAS Embedded Process require a specific version of the SAP HANA client and server environment. For more information, see the SAS Foundation system requirements documentation for your operating environment.

Overview of the In-Database Deployment Package for SAP HANA

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

The in-database deployment package for SAP HANA must be installed and configured before you can perform the following tasks:

- Use the %INDHN_PUBLISH_MODEL scoring publishing macro to create scoring files inside the database.
- Run SAS High-Performance Analytics when the analytics cluster is using a parallel connection with a remote SAP HANA appliance. The SAS Embedded Process, which resides on the data appliance, is used to provide high-speed parallel data transfer between the data appliance and the analytics environment where it is processed.

For more information, see the *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*.

For more information about using the scoring publishing macros, see the *SAS In-Database Products: User's Guide*.

The SAS Embedded Process is a SAS server process that runs within SAP HANA to read and write data. The SAS Embedded Process contains macros, run-time libraries, and other software that is installed on your SAP HANA system. These installations are done so that the SAS scoring files created in SAP HANA can access routines within the SAS Embedded Process run-time libraries.

SAP HANA Installation and Configuration

Installing and Configuring SAP HANA

To install and configure SAP HANA, follow these steps:

1. Review the permissions required for installation.

For more information, see [“SAP HANA Permissions” on page 224](#).

2. Review the number of semaphore arrays configured for the SAP HANA server.

It is recommended that the SAP HANA server that runs the SAS Embedded Process be configured with a minimum of 1024 to 2048 semaphore arrays. For more information, see [“Semaphore Requirements When Using the SAS Embedded Process for SAP HANA” on page 224](#).

3. Enable the SAP HANA Script Server process as SYSTEM in the SAP HANA Studio.

The SAP HANA script server process must be enabled to run in the HANA instance. The script server process can be started while the SAP HANA database is already running.

To start the Script Server, follow these steps:

- a. Open the **Configuration** tab page in the SAP HANA Studio.
- b. Expand the daemon.ini configuration file.

- c. Expand the **scriptserver** section.
- d. Change the **instances** parameter from 0 to 1 at the system level.
A value of 1 indicates you have enabled the server.

Note: For more information, see SAP Note 1650957.

- 4. If you are upgrading from or reinstalling a previous release, follow the instructions in [“Upgrading or Reinstalling a Previous Version” on page 217](#) before installing the in-database deployment package.
- 5. Install the SAS Embedded Process.

For more information, see [“Installing the In-Database Deployment Package for SAP HANA” on page 218](#).

- 6. Install the SASLINK Application Function Library.

For more information, see [“Installing the SASLINK AFL Plugins on the Appliance \(HANA SPS08\)” on page 219](#) or [“Installing the SASLINK AFL Plugins on the Appliance \(HANA SPS08\)” on page 219](#).

- 7. Import the SAS_EP Stored Procedure.

For more information, see [“Importing the SAS_EP Stored Procedure” on page 222](#).

- 8. Verify that the Auxiliary Wrapper Generator and Eraser Procedures are installed in the SAP HANA catalog.

For more information, see [“Auxiliary Wrapper Generator and Eraser Procedures” on page 222](#).

- 9. Start the SAS Embedded Process.

- a. Log on to the SAP HANA server as the database administrator or change the user to the database administrator.

You can use one of these commands.

```
su - SIDadm
ssh SIDadm@server-name
```

- b. Navigate to the directory that contains the StartupSASEP.sh script.

```
cd /EPInstallDir/StartupSASEP.sh
```

- c. Run the StartupSASEP.sh script.

```
./StartupSASEP.sh
```

- 10. If you plan to use SAS Model Manager with the SAS Scoring Accelerator for in-database scoring, perform the additional configuration tasks provided in [Chapter 24](#), [“Configuring SAS Model Manager,” on page 231](#).

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 or Reinstalling a Previous Version

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

- 1. Log on to the SAP HANA system as root.

You can use `su` or `sudo` to become the root authority.

2. Run the `UninstallSASEPFiles.sh` file.

```
./UninstallSASEPFiles.sh
```

The `UninstallSASEPFiles.sh` file is in the `/EPInstallDir/` where you copied the `tkindbsrv-9.43-n_lax.sh` self-extracting archive file.

This script stops the SAS Embedded Process on the server. The script deletes the `/SAS/SASTKInDatabaseServerForSAPHANA` directory and all its contents.

Note: You can find the location of `EPInstallDir` by using the following command:

```
ls -l /usr/local/bin/tkhnmain
```

3. Reinstall the SAS Embedded Process.

For more information, see [“Installing the In-Database Deployment Package for SAP HANA” on page 218](#).

Installing the In-Database Deployment Package for SAP HANA

The SAS Embedded Process is contained in a self-extracting archive file. The self-extracting archive file is located in the `SAS-installation-directory/SASTKInDatabaseServer/9.4/SAPHANAonLinuxx64/` directory.

To install the self-extracting archive file, follow these steps:

1. Using a method of your choice, transfer the `tkindbsrv-9.43-n_lax.sh` file to the target SAS Embedded Process directory on the SAP HANA appliance.

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.

This example uses secure copy, and `/EPInstallDir/` is the location where you want to install the SAS Embedded Process.

```
scp tkindbsrv-9.43-n_lax.sh username@hana: /EPInstallDir/
```

Note: The `EPInstallDir` directory requires Read and Execute permissions for the database administrator.

2. After the `tkindbsrv-9.43-n_lax.sh` has been transferred, log on to the SAP HANA server as the “owner” of the SAS Embedded Process installation directory.

```
ssh sas-owner@server-name
```

3. Navigate to the directory where the self-extracting archive file was downloaded in Step 1.

```
cd /EPInstallDir
```

4. Use the following command at the UNIX prompt to unpack the self-extracting archive file.

```
./tkindbsrv-9.43-n_lax.sh
```

Note: If you receive a “permissions denied” message, check the permissions on the `tkindbsrv-9.43-n_lax.sh` file. This file must have Execute permissions to run.

After the script runs and the files are unpacked, the content of the target directories should look similar to these. Directories and files of interest are shaded.

```
/EPInstallDir/afl_wrapper_eraser.sql
```



```

/EPInstallDir/afl_wrapper_generator.sql
/EPInstallDir/InstallSASEPFiles.sh
/EPInstallDir/manifest
/EPInstallDir/mit_unzip.log
/EPInstallDir/saslink.lst
/EPInstallDir/saslink_area.pkg
/EPInstallDir/SAS
/EPInstallDir/SAS_EP_sas.com.tgz
/EPInstallDir/sas_saslink_installer.tgz
/EPInstallDir/ShowSASEPStatus.sh
/EPInstallDir/ShutdownSASEP.sh
/EPInstallDir/StartupSASEP.sh
/EPInstallDir/UninstallSASEPFiles.sh
/EPInstallDir/tkindbsrv-9.43-n_lax.sh

```

Note that a SAS directory is created where the EP files are installed. The contents of the **/EPInstallDir/SAS/** directories should look similar to these.

```

/EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/admin
/EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/bin
/EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/logs
/EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/sasexe
/EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/utilities

```

The InstallSASEPFiles.sh file installs the SAS Embedded Process. The next step explains how to run this file.

The UninstallSASEPFiles.sh file uninstalls the SAS Embedded Process. The ShowSASEPStatus.sh file shows the status of the SAS Embedded Process on each instance. The StartupSASEP.sh and ShutdownSASEP.sh files enable you to manually start and stop the SAS Embedded Process. For more information about running these two files, see [“Controlling the SAS Embedded Process” on page 223](#).

5. Use the following command at the UNIX prompt to install the SAS Embedded Process.

```
./InstallSASEPFiles.sh
```

Note: To execute this script you need root authority. Either use the su command to become the root or use the sudo command to execute this script to install the SAS Embedded Process.

Note: `-verbose` is on by default and enables you to see all messages generated during the installation process. Specify `-quiet` to suppress messages.

Installing the SASLINK AFL Plugins on the Appliance (HANA SPS08)

The SASLINK Application Function Library (AFL) files are included with the server side components. These files must be copied to the SASLINK AFL plugins directory on the SAP HANA server.

Note: The *SID* referenced in these instructions is the SAP HANA system identifier (for example, **HDB**).

To install the SASLINK AFL plugins on the appliance (HANA SPS08), follow these steps:

1. If it does not exist, create a plugins directory in the \$DIR_SYSEXE directory.

- a. Log on to the SAP HANA server as the root authority.

You can use one of these commands.

```
su - root
sudo su -
```

- b. If it does not exist, create the plugins directory.

```
mkdir -p /usr/sap/SID/SYS/exe/hdb/plugins
chown SIDadm:sapsys /usr/sap/SID/SYS/exe/hdb/plugins
chmod 750 /usr/sap/SID/SYS/exe/hdb/plugins
exit
```

2. Use one of these commands to change the user to the database administrator.

```
su - SIDadm
ssh SIDadm@server-name
```

3. Stop the SAP HANA database if it is running.

```
HDB stop
```

4. If it does not exist, create the SASLINK AFL plugins directory.

```
cdexe
cd -P ..
mkdir -p plugins/sas_afl_sdk_saslink_1.00.1.0.0_1
cdexe
mkdir -p plugins
cd -P plugins
ln -s ../../plugins/sas_afl_sdk_saslink_1.00.1.0.0_1 sas_afl_sdk_saslink
```

5. Copy the SASLINK AFL files from the **/EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/sasexe** and **/EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/admin** directories to the SASLINK AFL plugins directory.

```
cdexe
cd plugins/sas_afl_sdk_saslink
cp /EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/
sasexe/libaflsaslink.so .
cp /EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/
admin/saslink.lst .
cp /EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/
admin/saslink_area.pkg .
cp /EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/
admin/manifest .
```

Note: You can find the location of *EPInstallDir* by using the following command:

```
ls -l /usr/local/bin
```

6. Restart the SAP HANA database.

```
HDB start
```

Installing the SASLINK AFL Plugins on the Appliance (HANA SPS09)

The SASLINK Application Function Library (AFL) files are included in an installer that is packaged as a tarball (TAR file) and that is provided when the SAS Embedded Process self-extracting archive file is unpacked.

Note: The *SID* referenced in these instructions is the SAP HANA system identifier (for example, **HDB**).

To install the SASLINK AFL plugins on the appliance (HANA SPS09), follow these steps:

1. Log on to the SAP HANA server as the database administrator or change the user to the database administrator.

You can use one of these commands.

```
su - SIDadm
ssh SIDadm@server-name
```

2. If the SAS Embedded Process is running, run the ShutdownSASEP.sh script to stop the process.

```
/EPInstallDir/ShutdownSASEP.sh
```

Alternatively, you can shutdown the SAS Embedded Process by removing its PID file.

```
rm /var/tmp/tkhnmain.pid
```

3. Stop the SAP HANA database if it is running.

```
HDB stop
```

4. Use one of these commands to change the user to the root authority.

```
su - root
sudo su -
```

5. Copy the TAR file to the **/tmp** directory.

```
cp /EPInstallDir/sas_saslink_install.tgz /tmp
```

6. Unpack the TAR file.

```
cd /tmp
tar -xvzf sas_saslink_install.tgz
```

7. Run the HANA install utility from the directory where the TAR file was unpacked. Specify the system ID of the HANA instance when prompted by the install utility.

```
cd /tmp/sas_saslink_installer/installer
./hdbinst
```

8. Use one of these commands to change the user back to the database administrator or change the user to the database administrator.

```
su - SIDadm
exit
```

9. Restart the SAP HANA database.

HDB start

Importing the SAS_EP Stored Procedure

The SAS_EP Stored Procedure is used by the %INDHN_RUN_MODEL macro to run the scoring model.

The SAS_EP stored procedure is contained in a delivery unit named SAS_EP_sas.com.tgz. The SAS_EP_sas.com.tgz package was installed in the **EPInstallDir** directory when the tkindsrv-9.43-n_lax.sh file was unpacked.

Note: You can find the location of **EPInstallDir** by using the following command:

```
ls -l /usr/local/bin
```

To import the delivery unit into SAP HANA, follow these steps:

Note: Permissions and roles are required to import the procedure package. For more information, see “[SAP HANA Permissions](#)” on page 224.

1. Navigate to the **EPInstallDir** directory.
2. Copy the SAS_EP_sas.com.tgz package to a client machine on which the SAP HANA Studio client is installed.
3. Import the delivery unit.

There are several methods of importing the .tgz file. Examples are SAP HANA Studio or the Lifecycle Manager. To import the delivery unit using SAP HANA Studio, follow these steps:

- a. Ensure that you have a connection to the target SAP HANA back end from your local SAP HANA Studio.
- b. Select **File** ⇒ **Import**.
- c. Select **SAP HANA Content** ⇒ **Delivery Unit** and click **Next**.
- d. Select the target system and click **Next**.
- e. In the Import Through Delivery Unit window, select the **Client** check box and select the SAS_EP_sas.com.tgz file.
- f. Select the **Overwrite inactive versions** and **Activate object** check boxes.

The list of objects is displayed under **Object import simulation**.

- g. Click **Finish** to import the delivery unit.

Auxiliary Wrapper Generator and Eraser Procedures

SAP HANA SPS08

Operation of the SASLINK AFL and the SAS Embedded Process requires that the SAP HANA AFL_WRAPPER_GENERATOR and AFL_WRAPPER_ERASER procedures

are installed in the SAP HANA catalog. If the procedures are not already installed in the SAP HANA catalogs, then copies of these procedures can be found in the install directory on the server.

```
/EPInstallDir/afl_wrapper_generator.sql
/EPInstallDir/afl_wrapper_eraser.sql
```

Note: You can find the location of *EPInstallDir* by using the following command:

```
ls -l /usr/local/bin
```

To install these procedures, you must execute the SQL scripts, using the SAP HANA Studio, as the SAP HANA user SYSTEM. For more information, see the SAP HANA Predictive Analysis Library (PAL) document.

CAUTION:

If a procedure has already been installed, executing the SQL script causes an error. If you encounter an error, see your SAP HANA database administrator.

SAP HANA SPS09

Operation of the SASLINK AFL and the SAS Embedded Process requires wrapper generator and eraser procedures that are already installed in the SAP HANA catalog on the server. There is no need to manually install these procedures.

However, an additional permission, `AFLPM_CREATOR_ERASER_EXECUTE`, is required. For more information, see [“SAP HANA Permissions” on page 224](#).

Controlling the SAS Embedded Process

The SAS Embedded Process starts when you run the `StartupSASEP.sh` script. It continues to run until it is manually stopped or the database is shut down.

Note: Starting and stopping the SAS Embedded Process has implications for all scoring model publishers.

Note: Manually starting and stopping the SAS Embedded Process requires HANA database administrator user permissions.

When the SAS Embedded Process is installed, the `ShutdownSASEP.sh` and `StartupSASEP.sh` scripts are installed in the following directory. For more information about these files, see [“Installing the In-Database Deployment Package for SAP HANA” on page 218](#).

```
/EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/
```

Note: You can find the location of *EPInstallDir* by using the following command:

```
ls -l /usr/local/bin
```

Use the following command to start the SAS Embedded Process.

```
/EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/StartupSASEP.sh
```

Note: The `-verbose` option is on by default and provides a status of the start-up operations as they occur. You can specify the `-quiet` option to suppress messages.

`ShutdownSASEP.sh` shuts down the SAS Embedded Process. It is designed to be used to shut down the SAS Embedded Process prior to a database upgrade or re-install. This script should not be used as part of the normal operation.

Use the following command to shut down the SAS Embedded Process.

```
/EPInstallDir/SAS/SASTKInDatabaseServerForSAPHANA/9.43/ShutdownSASEP.sh
```

Note: The `-verbose` option is on by default and provides a status of the shutdown operations as they occur. You can specify the `-quiet` option to suppress messages.

Semaphore Requirements When Using the SAS Embedded Process for SAP HANA

Each time a query using the SAS_EP stored procedure is invoked to execute a score, it requests a set of semaphore arrays (sometimes referred to as semaphore "sets") from the operating system. The SAS Embedded Process releases the semaphore arrays back to the operating system after scoring is complete.

The SAP HANA server that runs the SAS Embedded Process should be configured with a minimum of 1024 to 2048 semaphore arrays.

Note: The semaphore limit on the "maximum number of arrays" is distinct from the semaphore limit on the "maximum number of semaphores system wide". The Linux `ipcs -sl` command shows the typical default semaphore-related limits set on SAP HANA:

```
----- Semaphore Limits -----
max number of arrays = 2048
max semaphores per array = 250
max semaphores system wide = 512000
max ops per semop call = 100
semaphore max value = 32767
```

SAP HANA Permissions

The following permissions are needed by the person who installs the in-database deployment package:

Note: Some of the permissions listed below cannot be granted until the Auxiliary Wrapper Generator and Eraser Procedures are installed. For more information, see ["Auxiliary Wrapper Generator and Eraser Procedures" on page 222](#).

Task	Permission Needed
Unpack the self-extracting archive file	owner of the SAS Embedded Process install directory. The SAS Embedded Process install directory must have permissions that allow Read and Execute permission by the database administrator user.
Install or uninstall the SAS Embedded Process (run InstallSASEPFiles.sh or UninstallSASEPFiles.sh script)	root authority

Task	Permission Needed
Import the SAS_EP procedure package	A user on the SAP HANA server that has at least the CONTENT_ADMIN role or its equivalent
Install AFL plugins (requires starting and stopping the database)	root authority and database administrator
Install an auxiliary procedure generator and eraser	SYSTEM user

The following permissions are needed by the person who runs the scoring models. Without these permissions, the publishing of the scoring models fails:

SAP HANA SPS08

- EXECUTE ON SYSTEM.afl_wrapper_generator to *userid* | *role*;
- EXECUTE ON SYSTEM.afl_wrapper_eraser to *userid* | *role*;
- AFL__SYS_AFL_SASLINK_AREA_EXECUTE to *userid* | *role*;

SAP HANA SPS09:

- AFLPM_CREATOR_ERASER_EXECUTE to *userid* | *role*;
- EXECUTE, SELECT, INSERT, UPDATE, and DELETE on the schema that is used for scoring

In addition, the roles of **sas.ep::User** and **AFL__SYS_AFL_SASLINK_AREA_EXECUTE** must be assigned to any user who wants to perform in-database processing. The **sas.ep::User** role is created when you import the SAS_EP stored procedure. The **AFL__SYS_AFL_SASLINK_AREA_EXECUTE** role is created when the AFL wrapper generator is created.

Note: If you plan to use SAS Model Manager with the SAS Scoring Accelerator for in-database scoring, additional permissions are required. For more information, see [Chapter 24, “Configuring SAS Model Manager,” on page 231](#).

Documentation for Using In-Database Processing in SAP HANA

For information about using in-database processing in SAP HANA, see the *SAS In-Database Products: User's Guide*, located at <http://support.sas.com/documentation/onlinedoc/indbtech/index.html>.

Chapter 23

Administrator's Guide for SPD Server

Installation and Configuration Requirements for the SAS	
Scoring Accelerator for SPD Server	227
Prerequisites	227
SPD Server Permissions	227
Where to Go from Here	227

Installation and Configuration Requirements for the SAS Scoring Accelerator for SPD Server

Prerequisites

The SAS Scoring Accelerator for SPD Server requires SAS Scalable Performance Data Server 5.1 and SAS 9.4.

If you have a model that was produced by SAS Enterprise Miner, an active SPD Server, and a license for the SAS Scoring Accelerator for SPD Server, you have everything that you need to run scoring models in the SPD Server. Installation of an in-database deployment package is not required.

SPD Server Permissions

You must have permissions for the domains that you specify in the INDCONN and INDDATA macro variables when you execute the publish and run macros.

You also need regular Read, Write, and Alter permissions when writing files to the OUTDIR directory in the %INDSP_RUN_MODEL macro.

Where to Go from Here

For more information about using the SAS Scoring Accelerator for SPD Server, see the *SAS In-Database Products: User's Guide*.

Part 6

Configurations for SAS Model Manager

Chapter 24

Configuring SAS Model Manager 231

Chapter 24

Configuring SAS Model Manager

Preparing a Data Management System for Use with SAS Model Manager	231
Prerequisites	231
Overview of Preparing a Data Management System for Use with SAS Model Manager	232
Configuring a Database	232
SAS Embedded Process Publish Method	232
Scoring Function Publish Method	233
Finding the JDBC JAR Files	234
Configuring a Hadoop Distributed File System	235

Preparing a Data Management System for Use with SAS Model Manager

Prerequisites

SAS Foundation, SAS/ACCESS, and the in-database deployment package for the database must be installed and configured before you can prepare a data management system (database or file system) for use with SAS Model Manager. For more information, see the chapter for your type of database or file system in this guide. Here are the databases and file systems that can be used with SAS Model Manager:

- [DB2](#)
- [Greenplum](#)
- [Hadoop](#)
- [Netezza](#)
- [Oracle](#)
- [SAP HANA](#)
- [Teradata](#)

Overview of Preparing a Data Management System for Use with SAS Model Manager

Additional configuration steps are required to prepare a data management system (database or file system) for publishing and scoring in SAS Model Manager if you plan to use the scoring function (MECHANISM=STATIC) publish method or the SAS Embedded Process (MECHANISM=EP) publish method. If you want to store the scoring function metadata tables in the database, then the SAS Model Manager In-Database Scoring Scripts product must be installed before the database administrator (DBA) can prepare a database for use with SAS Model Manager.

During the installation and configuration of SAS 9.4 products, the SAS Model Manager In-Database Scoring Scripts product is installed on the middle-tier server or another server tier if it is included in the custom plan file.

The location of the SAS installation directory is specified by the user. Here is the default installation location for the SAS Model Manager In-Database Scoring Scripts product on a Microsoft Windows server: `C:\Program Files\SASHome\SASModelManagerInDatabaseScoringScripts`

The script installation directory includes a directory that specifies the version of SAS Model Manager (currently 14.1). The files and subdirectories that are needed to prepare a database for use by SAS Model Manager are located in the version directory. The `Utilities` subdirectory contains two SQL scripts for each type of database: a Create Tables script and a Drop Tables script. The DBA needs these SQL scripts to create the tables needed by the SAS Model Manager to publish scoring functions.

Note: The database tables store SAS Model Manager metadata about scoring functions.

Configuring a Database

SAS Embedded Process Publish Method

To enable users to publish scoring model files to a database from SAS Model Manager using the SAS Embedded Process, follow these steps:

1. Create a separate database where the tables can be stored.
2. Set the user access permissions for the database.
 - a. GRANT CREATE, DROP, EXECUTE, and ALTER permissions for functions and procedures.

For more information about permissions for the specific databases, see the following topics:

- [“DB2 Permissions” on page 172](#)
- [“Greenplum Permissions” on page 192](#)
- [“Netezza Permissions” on page 205](#)
- [“Oracle Permissions” on page 213](#)
- [“SAP HANA Permissions” on page 224](#)
- [“Teradata Permissions for Publishing Formats and Scoring Models” on page 123](#)

- b. GRANT CREATE and DROP permissions for tables. With these permissions, users can validate the scoring results when publishing a scoring model files using SAS Model Manager.
- c. Run the database-specific macro to create a table in the database to store the published model scoring files. The value of the MODELTABLE= argument in the macro should match the specification of the In-Database Options for SAS Model Manager in SAS Management Console. For more information, see [In-Database Options](#).

If the **Use model manager table** option is set to **No**, then the model-table-name should be `sas_model_table`. Otherwise, it should be `sas_mdlmgr_ep`.

Here is an example of the create model table macro for Teradata:

```
%INDTD_CREATE_MODELTABLE(DATABASE=database-name, MODELTABLE=model-table-name,
ACTION=CREATE) ;
```

For more information about creating a table for a specific database, see the *SAS In-Database Products: User's Guide*.

Scoring Function Publish Method

To enable users to publish scoring functions to a database from SAS Model Manager, follow these steps:

1. Create a separate database where the tables can be stored.
2. Set the user access permissions for the database.
 - a. GRANT CREATE, DROP, EXECUTE, and ALTER permissions for functions and procedures.

For more information about permissions for the specific databases, see the following topics:

 - [“DB2 Permissions” on page 172](#)
 - [“Greenplum Permissions” on page 192](#)
 - [“Netezza Permissions” on page 205](#)
 - [“Teradata Permissions for Publishing Formats and Scoring Models” on page 123](#)
 - b. GRANT CREATE and DROP permissions for tables. With these permissions, users can validate the scoring results when publishing a scoring function using SAS Model Manager.
 - c. GRANT SELECT, INSERT, UPDATE, and DELETE permissions for SAS Model Manager metadata tables.
 - d. GRANT SELECT permission for the following views to validate the scoring function names:
 - syscat.functions for DB2
 - pg_catalog.pg_proc for Greenplum
 - dbc.functions for Teradata
 - _v_function for Netezza

Note: If scoring input tables, scoring output tables, or views exist in another database, then the user needs appropriate permissions to access those tables or views.

3. Navigate to the `\sasinstalldir\SASModelManagerInDatabaseScoringScripts\14.1\Utilities` directory to find the Create Tables and Drop Tables scripts for your database. Then, follow these steps:
 - a. Verify the statements that are specified in the Create Tables script. Here are the names of the scripts for each type of database:
 - DB2 SQL scripts: `createTablesDB2.sql` and `dropTablesDB2.sql`
 - Greenplum SQL scripts: `createTablesGreenplum.sql` and `dropTablesGreenplum.sql`
 - Netezza SQL scripts: `createTablesNetezza.sql` and `dropTablesNetezza.sql`
 - Teradata SQL scripts: `createTablesTD.sql` and `dropTablesTD.sql`
 - b. Execute the Create Tables script for a specific type of database.
4. Download the JDBC driver JAR files and place them in the `\lib` directory on the web application server where the SAS Model Manager web application is deployed.

The default directory paths for the SAS Web Application Server are the following:

single server install and configuration

`\sasconfigdir\Lev#\Web\WebAppServer\SASServer1_1\lib`

This is an example of the directory path: `C:\SAS\Config\Lev1\Web\WebAppServer\SASServer1_1\lib`

multiple server install and configuration

`\sasconfigdir\Lev#\Web\WebAppServer\SASServer11_1\lib`

This is an example of the directory path: `C:\SAS\Config\Lev1\Web\WebAppServer\SASServer11_1\lib`

Note: You must have Write permission to place the JDBC driver JAR files in the `\lib` directory. Otherwise, you can have the server administrator download them for you.

For more information, see [“Finding the JDBC JAR Files” on page 234](#).

5. Restart the SAS servers on the web application server.

Finding the JDBC JAR Files

The DB2 JDBC JAR files are `db2jcc.jar` and `db2jcc_license_cu.jar`. The DB2 JDBC JAR files can be found on the server on which the database client was installed. For example, the default location for Windows is `C:\Program Files\IBM\SQLLIB\java`.

The Greenplum database uses the standard PostgreSQL database drivers. The PostgreSQL JDBC JAR file can be found on the PostgreSQL – JDBC Driver site at <https://jdbc.postgresql.org/download.html>. An example of a JDBC driver name is `postgresql-9.2-1002.jdbc4.jar`.

The Netezza JDBC JAR file is `nzjdbc.jar`. The Netezza JDBC JAR file can be found on the server on which the database client was installed. For example, the default location for Windows is `C:\JDBC`.

The Teradata JDBC JAR files are `terajdbc4.jar` and `tdgssconfig.jar`. The Teradata JDBC JAR files can be found on the Teradata website at <http://www.teradata.com>. Select **Support** ⇒ **Downloads** ⇒ **Developer Downloads**, and then click **JDBC Driver** in the table.

For more information about the database versions that are supported, see the SAS Foundation system requirements documentation for your operating environment.

Configuring a Hadoop Distributed File System

To enable users to publish scoring model files to a Hadoop Distributed File System (HDFS) from SAS Model Manager using the SAS Embedded Process, follow these steps:

1. Create an HDFS directory where the model files can be stored.

Note: The path to this directory is used when a user publishes a model from the SAS Model Manager user interface to Hadoop.

2. Grant users Write access permission to the HDFS directory. For more information, see “[Hadoop Permissions](#)” on page 9.
3. Add this line of code to the `autoexec_usermods.sas` file that is located in the Windows directory `\SAS-configuration-directory\Lev#\SASApp\WorkspaceServer\`:

```
%let HADOOP_Auth = Kerberos or blank;
```

UNIX Specifics

The location of the `autoexec_usermods.sas` file for UNIX is `/SAS-configuration-directory/Lev#/SASApp/WorkspaceServer/`.

If your Hadoop server is configured with Kerberos, set the `HADOOP_Auth` variable to Kerberos. Otherwise, leave it blank.

4. (Optional) If you want users to be able to copy the publish code and execute it using Base SAS, then this line of code must be added to the `sasv9.cfg` file that is located in the Windows directory `\SASHome\SASFoundation\9.4\`:

```
-AUTOEXEC '\SAS-configuration-directory\Lev#\SASApp\WorkspaceServer\autoexec_usermods.sas'
```

UNIX Specifics

The location of the `sasv9.cfg` file for UNIX is `/SASHome/SASFoundation/9.4/`.

5. (Optional) If your Hadoop distribution is using Kerberos authentication, each user must have a valid Kerberos ticket to access SAS Model Manager. However, users that are authenticated by Kerberos cannot write the publish results files to the SAS Content Server when publishing a model because they have not supplied a password to SAS Model Manager. Therefore, additional post-installation configuration steps are needed so that users can publish models to a Hadoop Distributed File System (HDFS) from SAS Model Manager. For more information, see *SAS Model Manager: Administrator's Guide*.

Recommended Reading

Here is the recommended reading list for this title:

- *SAS/ACCESS for Relational Databases: Reference*
- *SAS Data Loader for Hadoop: User's Guide*
- *SAS Data Quality Accelerator for Teradata: User's Guide*
- *SAS DS2 Language Reference*
- *SAS Hadoop Configuration Guide for Base SAS and SAS/ACCESS*
- *SAS High-Performance Analytics Infrastructure: Installation and Configuration Guide*
- *SAS In-Database Products: User's Guide*
- *SAS Model Manager: Administrator's Guide*

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

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-0025
Fax: 1-919-677-4444
Email: sasbook@sas.com
Web address: sas.com/store/books

Index

Special Characters

%INDAC_PUBLISH_FORMATS macro
 147
 %INDAC_PUBLISH_MODEL macro
 147
 %INDB2_PUBLISH_COMPILEUDF
 macro 164
 running 165
 syntax 167
 %INDB2_PUBLISH_DELETEUDF
 macro 168
 running 168
 syntax 170
 %INDB2_PUBLISH_FORMATS macro
 153
 %INDB2_PUBLISH_MODEL macro
 153
 %INDGP_PUBLISH_COMPILEUDF
 macro 182
 running 183
 syntax 185
 %INDGP_PUBLISH_COMPILEUDF_E
 P macro 186
 running 186
 syntax 188
 %INDGP_PUBLISH_FORMATS macro
 176
 %INDGP_PUBLISH_MODEL macro
 176
 %INDHN_PUBLISH_MODEL macro
 216
 %INDNZ_PUBLISH_COMPILEUDF
 macro 202
 running 202
 syntax 204
 %INDNZ_PUBLISH_FORMATS macro
 193
 %INDNZ_PUBLISH_JAZLIB macro 199
 running 199
 syntax 201
 %INDNZ_PUBLISH_MODEL macro
 193
 %INDOR_PUBLISH_MODEL macro
 209

%INDTD_PUBLISH_FORMATS macro
 120
 %INDTD_PUBLISH_MODEL macro
 120

A

ACTION= argument
 %INDB2_PUBLISH_COMPILEUDF
 macro 167
 %INDB2_PUBLISH_DELETEUDF
 macro 170
 %INDGP_PUBLISH_COMPILEUDF
 macro 185
 %INDGP_PUBLISH_COMPILEUDF_
 EP macro 188
 %INDNZ_PUBLISH_COMPILEUDF
 macro 205
 %INDNZ_PUBLISH_JAZLIB macro
 201
 AFL_WRAPPER_ERASER procedure
 222
 AFL_WRAPPER_GENERATOR
 procedure 222
 Ambari
 deploy SAS Embedded Process stack
 18, 30
 remove SAS Embedded Process stack
 17
 Aster
 documentation for publishing formats
 and scoring models 151
 in-database deployment package 147
 installation and configuration 148
 permissions 151
 SAS Embedded Process 147
 SAS/ACCESS Interface 147
 SQL/MR functions 148
 authorization for stored procedures 139

B

binary files
 for Aster 148

- for DB2 functions 159
- for Greenplum functions 178
- for Netezza functions 197

BTEQ 140

C

Cloudera

- Hadoop installation and configuration
 - using the SAS Deployment Manager 11
- manual Hadoop installation and configuration 35

Cloudera Manager

- deploy SAS Embedded Process parcel 18, 29
- remove SAS Embedded Process stack 16

cluster, configuring 105

COMPILER_PATH= argument

- %INDB2_PUBLISH_COMPILEUDF macro 167

configuration

- Aster 147
- DB2 155
- Greenplum 177
- HDFS for Model Manager 235
- IBM BigInsights 35
- MapR 35
- Model Manager database 232
- Netezza 195
- Oracle 209
- Pivotal HD 35
- SAP HANA 216
- SPD Server 227
- Teradata 125

customizing the QKB 142

D

Data Loader system requirements 60

data quality stored procedures

- See [stored procedures](#)

DATABASE= argument

- %INDB2_PUBLISH_COMPILEUDF macro 167
- %INDB2_PUBLISH_DELETEUDF macro 170
- %INDGP_PUBLISH_COMPILEUDF macro 185
- %INDGP_PUBLISH_COMPILEUDF_EP macro 188
- %INDNZ_PUBLISH_COMPILEUDF macro 204
- %INDNZ_PUBLISH_JAZLIB macro 201

DataFlux Data Management Studio

- customizing the QKB 142

DB2

- documentation for publishing formats or scoring models 173
- function publishing process 154
- installation and configuration 155
- JDBC Driver 234
- permissions 172
- preparing for SAS Model Manager use 231
- SAS Embedded Process 153
- SAS/ACCESS Interface 153
- unpacking self-extracting archive files 159, 160

DB2IDA command 163

DB2PATH= argument

- %INDB2_PUBLISH_COMPILEUDF macro 167

DB2SET command 160

- syntax 162

DBSCHEMA= argument

- %INDNZ_PUBLISH_COMPILEUDF macro 204
- %INDNZ_PUBLISH_JAZLIB macro 201

deactivating existing versions 69

deploying files

- standard deployment 75
- zip file deployment 63

deployment

- standard 73
- zip file 61

documentaion

- for in-database processing in Hadoop 10

documentation

- for in-database processing in SAP HANA 225
- for publishing formats and scoring models in Aster 151
- for publishing formats and scoring models in DB2 173
- for publishing formats and scoring models in Greenplum 192
- for publishing formats and scoring models in Netezza 207
- for publishing formats and scoring models in Teradata 124
- for publishing scoring models in Oracle 213

dq_grant.sh script 138, 139

dq_install.sh script 138, 139

dq_uninstall script 138

dq_uninstall.sh script 143

drivers, JDBC 106

E

end-user support 108, 115

F

formats library

DB2 installation 158
Greenplum installation 178
Netezza installation 197
Teradata installation 129

function publishing process

DB2 154
Netezza 194

functions

SAS_COMPILEUDF (DB2) 158, 164, 171
SAS_COMPILEUDF (Greenplum) 178, 182, 189
SAS_COMPILEUDF (Netezza) 197, 202
SAS_COPYUDF (Greenplum) 189
SAS_DEHEXUDF (Greenplum) 189
SAS_DELETEUDF (DB2) 158, 168, 171
SAS_DIRECTORYUDF (Greenplum) 189
SAS_DIRECTORYUDF (Netezza) 202
SAS_EP (Greenplum) 190
SAS_HEXTOTEXTUDF (Netezza) 202
SAS_PUT() (Aster) 147
SAS_PUT() (DB2) 153
SAS_PUT() (Greenplum) 176
SAS_PUT() (Netezza) 193, 194
SAS_PUT() (Teradata) 120
SAS_SCORE() (Aster) 148
SQL/MR (Aster) 148

G

global variables

See [variables](#)

Greenplum

documentation for publishing formats and scoring models 192
in-database deployment package 175
installation and configuration 177
JDBC Driver 234
permissions 192
preparing for SAS Model Manager use 231
SAS Embedded Process 190
SAS/ACCESS Interface 175
semaphore requirements 191
unpacking self-extracting archive files 178

H

Hadoop

backward compatibility 9
configuring HDFS using Model Manager 235
in-database deployment package 8
installation and configuration for IBM BigInsights 35
installation and configuration for MapR 35
installation and configuration for Pivotal HD 35
overview of configuration steps using the SAS Deployment Manager 13
overview of manual configuration steps 36
permissions 9
preparing for SAS Model Manager use 231
SAS/ACCESS Interface 8
unpacking self-extracting archive files 41

HCatalog

prerequisites 50
SAS client configuration 50
SAS Embedded Process configuration 50
SAS server-side configuration 51

Hortonworks

additional configuration 52
Hadoop installation and configuration using the SAS Deployment Manager 11
manual Hadoop installation and configuration 35

hot fixes 77, 81

I

IBM BigInsights

additional configuration 53
Hadoop installation and configuration 35

IDs, user 107

in-database deployment package for Aster overview 147

prerequisites 147

in-database deployment package for DB2 overview 153

prerequisites 153

in-database deployment package for Greenplum

overview 176

prerequisites 175

in-database deployment package for Hadoop

- overview 7
- prerequisites 8
- in-database deployment package for Netezza
 - overview 193
 - prerequisites 193
- in-database deployment package for Oracle
 - overview 209
 - prerequisites 209
- in-database deployment package for SAP HANA
 - overview 216
 - prerequisites 215
- in-database deployment package for Teradata
 - overview 119
 - prerequisites 119
- INDCONN macro variable 165, 169, 183, 187, 200, 203
- installation
 - Aster 147
 - DB2 155
 - Greenplum 177
 - IBM BigInsights 35
 - MapR 35
 - Netezza 195
 - Oracle 209
 - Pivotal HD 35
 - SAP HANA 216
 - SAS Embedded Process (Aster) 147
 - SAS Embedded Process (DB2) 154, 158
 - SAS Embedded Process (Greenplum) 176, 178
 - SAS Embedded Process (Hadoop) 7, 41
 - SAS Embedded Process (Netezza) 194, 197
 - SAS Embedded Process (Oracle) 209
 - SAS Embedded Process (SAP HANA) 218
 - SAS Embedded Process (Teradata) 120
 - SAS formats library 130, 158, 178, 197
 - SAS Hadoop MapReduce JAR files 41
 - scripts 138
 - SPD Server 227
 - Teradata 125
 - troubleshooting 141
 - verifying 140
- installation, manual
 - SAS Data Management Accelerator for Spark 97
 - SAS Data Quality Accelerator 85
 - SAS In-Database Deployment Package 80
 - SAS QKB 85

J

- JDBC Driver
 - DB2 234
 - Greenplum 234
 - Netezza 234
 - Teradata 235
- JDBC drivers 106
- JDBC JAR file locations 234

K

- kerberos
 - configuring 62, 74, 80, 84, 96, 112

M

- macro variables
 - See variables
- macros
 - %INDAC_PUBLISH_FORMATS 147
 - %INDAC_PUBLISH_MODEL 147
 - %INDB2_PUBLISH_COMPILEUDF 165, 167
 - %INDB2_PUBLISH_DELETEUDF 168, 170
 - %INDB2_PUBLISH_FORMATS 153
 - %INDB2_PUBLISH_MODEL 153
 - %INDGP_PUBLISH_COMPILEUDF 182, 185
 - %INDGP_PUBLISH_COMPILEUDF_EP 188
 - %INDGP_PUBLISH_FORMATS 176
 - %INDGP_PUBLISH_MODEL 176
 - %INDHN_PUBLISH_MODEL 216
 - %INDNZ_PUBLISH_COMPILEUDF 202, 204
 - %INDNZ_PUBLISH_FORMATS 193
 - %INDNZ_PUBLISH_JAZLIB 199, 201
 - %INDNZ_PUBLISH_MODEL 193
 - %INDOR_PUBLISH_MODEL 209
 - %INDTD_PUBLISH_FORMATS 120
 - %INDTD_PUBLISH_MODEL 120
- manual installation
 - SAS Data Management Accelerator for Spark 97
 - SAS Data Quality Accelerator 85
 - SAS In-Database Deployment Package 80
 - SAS QKB 85
- MapR
 - additional configuration 54
 - Hadoop installation and configuration 35
 - YARN application CLASSPATH 54
- Model Manager

- configuration 231
- configuring a database 232
- configuring HDFS 235
- creating tables 233
- JDBC Driver 234

N

Netezza

- documentation for publishing formats and scoring models 207
- function publishing process 194
- in-database deployment package 193
- installation and configuration 195
- JDBC Driver 234
- permissions 205
- preparing for SAS Model Manager use 231
- publishing SAS formats library 199
- SAS Embedded Process 193
- sas_ep cartridge 198
- SAS/ACCESS Interface 193

O

OBJNAME= argument

- %INDB2_PUBLISH_COMPILEUDF macro 168

OBJPATH= argument

- %INDGP_PUBLISH_COMPILEUDF macro 185
- %INDGP_PUBLISH_COMPILEUDF_EP macro 188

OOZIE 105

Oracle

- documentation for publishing formats and scoring models 213
- in-database deployment package 209
- permissions 213
- preparing for SAS Model Manager use 231
- SAS Embedded Process 209
- SAS/ACCESS Interface 209

OUTDIR= argument

- %INDB2_PUBLISH_COMPILEUDF macro 168
- %INDB2_PUBLISH_DELETEUDF macro 171
- %INDGP_PUBLISH_COMPILEUDF macro 186
- %INDGP_PUBLISH_COMPILEUDF_EP macro 189
- %INDNZ_PUBLISH_COMPILEUDF macro 205
- %INDNZ_PUBLISH_JAZLIB macro 202

P

parcels, creating 75

permissions

- for Aster 151
- for DB2 172
- for Greenplum 192
- for Hadoop 9
- for Netezza 205
- for Oracle 213
- for SAP HANA 224
- for SPD Server 227
- for Teradata 123

Pivotal

- Hadoop installation and configuration 35

PSFTP (DB2) 155

publishing

- Aster permissions 151
- DB2 permissions 172
- functions in DB2 154
- functions in Netezza 194
- Greenplum permissions 192
- Hadoop permissions 9
- Netezza permissions 205
- Oracle permissions 213
- SAP HANA permissions 224
- SPD Server permissions 227
- Teradata permissions 123

Q

QKB 83

- about 59
- customizing 142
- packaging for deployment 136
- updates 142

qkb_pack script 136

R

reinstalling a previous version

- Aster 148
- DB2 155
- Greenplum 177
- Hadoop 37
- Netezza 195
- Oracle 210
- SAP HANA 217
- Teradata 126

removing existing versions

- SAS Deployment Manager 69

removing stored procedures 143

requirements, Data Loader system 60

RPM file (Teradata) 129

S**SAP HANA**

AFL_WRAPPER_ERASER procedure 222

AFL_WRAPPER_GENERATOR procedure 222

documentation for in-database processing 225

in-database deployment package 215

installation and configuration 216

permissions 224

SAS Embedded Process 216, 223

SAS/ACCESS Interface 215

semaphore requirements 224

unpacking self-extracting archive files 218

SAS Data Management Accelerator for Spark 60, 95

SAS Data Quality Accelerator 59, 83

SAS Deployment Manager

using to deploy Hadoop in-database deployment package 11

using to deploy the Teradata in-database deployment package 125

SAS Embedded Process

adding to nodes after initial installation 56

adjusting performance 54

Aster 147

check status (DB2) 163

check status (Teradata) 131

configuration for HCatalog file formats 50

controlling (DB2) 163

controlling (Greenplum) 190

controlling (Hadoop) 43

controlling (SAP HANA) 223

controlling (Teradata) 131

DB2 153

disable or enable (DB2) 163

disable or enable (Teradata) 131

Greenplum 190

Hadoop 8

Netezza 193, 197

Oracle 209

overview 8

SAP HANA 216, 223

shutdown (DB2) 163

shutdown (Teradata) 131

support functions (Teradata) 131

Teradata 119

upgrading from a previous version (Aster) 148

upgrading from a previous version (DB2) 155

upgrading from a previous version (Netezza) 195

upgrading from a previous version (Oracle) 210

upgrading from a previous version (Teradata) 126

SAS FILENAME SFTP statement (DB2) 154

SAS formats library

DB2 158

Greenplum 178

Netezza 197, 199

Teradata 129

upgrading from a previous version (Greenplum) 177

upgrading from a previous versions (DB2) 155

upgrading from a previous versions (Netezza) 195

upgrading from a previous versions (Teradata) 126

SAS Foundation 8, 119, 147, 175, 193, 209, 215

SAS Hadoop MapReduce JAR files 41

SAS In-Database Deployment Package 59, 79

SAS In-Database products 4

SAS Quality Knowledge Base 83
about 59

SAS_COMPILEUDF function

actions for DB2 164

actions for Greenplum 182

actions for Netezza 202

binary files for DB2 158

binary files for Greenplum 178

binary files for Netezza 197

validating publication for DB2 171

validating publication for Greenplum 189

SAS_COPYUDF function 182

validating publication for Greenplum 189

SAS_DEHEXUDF function 182

validating publication for Greenplum 189

SAS_DELETEUDF function

actions for DB2 168

binary files for DB2 158

validating publication for DB2 171

SAS_DIRECTORYUDF function 182, 202

validating publication for Greenplum 189

sas_ep cartridge 198

SAS_EP function

- validating publication for Greenplum 190
- SAS_HEXTOTEXTUDF function 202
- SAS_PUT() function
 - Aster 147
 - DB2 154
 - Greenplum 176
 - Netezza 194
 - Teradata 120
- SAS_SCORE() function
 - publishing 148
 - validating publication for Aster 151
- SAS_SYSFNLIB (Teradata) 131
- SAS/ACCESS Interface to Aster 147
- SAS/ACCESS Interface to Greenplum 175
- SAS/ACCESS Interface to Hadoop 8
- SAS/ACCESS Interface to Netezza 193
- SAS/ACCESS Interface to Oracle 209
- SAS/ACCESS Interface to SAP HANA 215
- SAS/ACCESS Interface to Teradata 119
- sasep-admin.sh script
 - overview 43
 - syntax 44
- sasepfunc function package 131
- SASLIB database (Netezza) 202
- SASLIB schema
 - DB2 165, 168
 - Greenplum 183
- SASUDF_COMPILER_PATH global variable 165
- SASUDF_DB2PATH global variable 165
- scoring functions in SAS Model Manager 232
- scripts for installation 138
- self-extracting archive files
 - unpacking for Aster 148
 - unpacking for DB2 159, 160
 - unpacking for Greenplum 178
 - unpacking for Hadoop 41
 - unpacking for SAP HANA 218
- semaphore requirements
 - Greenplum 191
 - SAP HANA 224
- SFTP statement 154
- SPD Server
 - in-database deployment package 227
 - permissions 227
- SQL/MR functions (Aster) 148
- SQOOP 105
- SSH software (DB2) 154
- stacks, creating 75
- standard deployment 73
- stored procedures
 - creating 139

- removing from database 143

T

- tables
 - creating for SAS Model Manager 233
- Teradata
 - BTEQ 140
 - documentation for publishing formats and scoring models 124
 - in-database deployment package 119
 - installation and configuration 125
 - JDBC Driver 235
 - Parallel Upgrade Tool 130
 - permissions 123
 - preparing for SAS Model Manager use 231
 - SAS Embedded Process 119
 - SAS Embedded Process support functions 131
 - SAS/ACCESS Interface 119
 - sasepfunc function package 131
 - troubleshooting installation 141

U

- unpacking self-extracting archive files
 - for Aster 148
 - for DB2 159, 160
 - for Greenplum 178
 - for Hadoop 41
 - for SAP HANA 218
- upgrading from a previous version
 - Aster 148
 - DB2 155
 - Greenplum 177
 - Hadoop 37
 - Netezza 195
 - Oracle 210
 - SAP HANA 217
 - Teradata 126
- user authorization for stored procedures 139
- user IDs 107

V

- validating publication of functions and variables for DB2 171
- validating publication of functions for Aster 151
- validating publication of functions for Greenplum 189, 190
- variables
 - INDCONN macro variable 165, 169, 183, 187, 200, 203

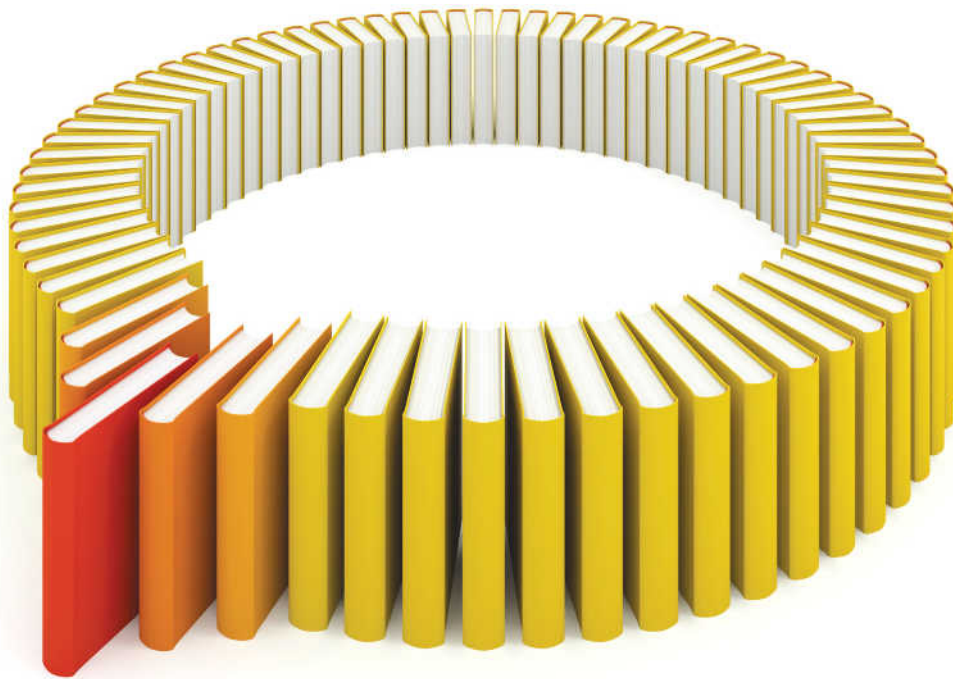
SASUDF_COMPILER_PATH global
variable [165](#)
SASUDF_DB2PATH global variable
[165](#)
verifying installation [140](#)

Y

YARN application CLASSPATH for
MapR [54](#)

Z

zip file deployment [61](#)



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.


THE POWER TO KNOW.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613

