



Cloud Data Exchange 2.3 for SAS[®] Viya[®] 3.4: Administrator's Guide 2.3

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *Cloud Data Exchange 2.3 for SAS® Viya® 3.4: Administrator's Guide*. Cary, NC: SAS Institute Inc.

Cloud Data Exchange 2.3 for SAS® Viya® 3.4: Administrator's Guide

Copyright © 2018, 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, NC 27513-2414

March 2024

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.

2.3-P1:dataagentag

Contents

PART 1 Introducing Cloud Data Exchange 1

Chapter 1 • About Cloud Data Exchange	3
Cloud Data Exchange and SAS Data Agent	3
Chapter 2 • Security Concepts	7
Security Features	7
Chapter 3 • Configuring Cloud Data Exchange for CAS	13
Create a CAS Library (Caslib)	13
Remote Connectivity	30
Working with Data Explorer	31
Chapter 4 • Unregister Data Agents	33
SAS Data Preparation Machine: Unregister SAS Data Agent	33

PART 2 SAS Data Agent 37

Chapter 5 • Working with SAS Data Agent	39
Getting Started	39
SAS Data Agent Configuration	45
Removing Data Agent Servers	48
Chapter 6 • Data Administration	51
Working with Data Services	52
Working with DSNs	54
Catalogs and Schemas	55
Command-Line Interface	56
The Command Collection	58
Catalogs	59
Data Services	62
Data Sources	64
Data Source Names	66
Profile	68
Schemas	69
Security (Domains)	70
Security (Credentials)	72
CLI Output	73
CLI Examples	75
Chapter 7 • Server Administration	79
SAS Data Agent Server Administration	79
Authentication and Authorization Services Resource Cache	81
Server Logging Configuration	83

PART 3 Appendices 93

Appendix 1 • CLI Connection Options 95

Appendix 2 • Legal Notices 107

Part 1

Introducing Cloud Data Exchange

<i>Chapter 1</i>	
About Cloud Data Exchange	3
<i>Chapter 2</i>	
Security Concepts	7
<i>Chapter 3</i>	
Configuring Cloud Data Exchange for CAS	13
<i>Chapter 4</i>	
Unregister Data Agents	33

Chapter 1

About Cloud Data Exchange

Cloud Data Exchange and SAS Data Agent	3
About Cloud Data Exchange	3
About SAS Data Agent	3
Command-Line Interface	3
REST APIs	4

Cloud Data Exchange and SAS Data Agent

About Cloud Data Exchange

Cloud Data Exchange (CDE) is a data connection capability for SAS Data Preparation on SAS Viya. Cloud Data Exchange securely copies data from an on-premises data store to a cloud-based instance of SAS Viya for use in SAS Viya applications. CDE performs high-volume data transfer from on-premises to SAS Viya in support of big data. A command-line interface (CLI) is also provided for administration and control of the customer data.

Cloud Data Exchange also allows Cloud Analytic Services (CAS) to read and write data stored on-premises. On-premises can be one or more deployments, all associated with one and the same SAS Viya full stack deployment. SAS Data Agent provides the means to move data to and from the cloud.

With Cloud Data Exchange, a user can create a caslib and access the data through SAS Data Agent. Once you have created a caslib, you can load data to CAS from the source environment and save it back to source environment as well.

About SAS Data Agent

SAS Data Agent is the server that provides the environment to move data. You can deploy SAS Data Agent locally, on-premises, or in a cloud environment. A CLI is used to create definitions such as data services and data source names. The CLI is also used to define security domains for accessing data sources that are located on-premises.

See Working with SAS Data Agent: [Getting Started](#) to begin using SAS Data Agent.

Command-Line Interface

The command-line interface (CLI) provides the primary administrative interface to SAS Data Agent, as follows:

- uses standard representation station transfer (REST) application program interface (API) calls to SAS Viya services
- operates as a plug-in to the sas-admin CLI
- can be updated via the support.sas.com download portal
- uses encrypted communications for all network communications
- uses a standard profile mechanism for storing OAUTH2 credentials and default attributes of the CLI

The CLI commands are organized around collections. Here are a few of the command collections available in the CLI:

- Data Agent servers
- security domains and credentials
- Data Services

The following table lists the supported data sources with their data service names:

Data Source	Data Service Name
SAS data sets	BASE
Apache Hive	HIVE
DB2	DB2
ODBC	ODBC
Oracle	ORACLE
PostgreSQL	POSTGRESQL
Redshift	REDSHIFT
SAP HANA	SAPHANA
SQL Server	SQLSERVER or SQLSVR
Teradata	TERADATA

REST APIs

In addition to the CLI, REST services provide an additional means of administration and configuration of SAS Data Agent. The following REST APIs are included with deployment of SAS Data Agent:

- DataAgentContent
- DataAgentManagement

REST services enables administrators and authorized users to explore data and administer data agent servers. Actions include the following:

- retrieval and update of the server configuration, including the location, purge cache, and shutdown time-out
- retrieval of the data definitions and configuration information for the data services, data source names, catalogs, and schemas
- creation, update, and deletion of the data definitions, including the data services, data source names, catalogs, and schemas

Chapter 2

Security Concepts

Security Features	7
Overview	7
Communication Summary	9
Topology	9
Secured Communications	9
Industry Standard Transport	10
Authentication	10
Data Access Controls	11
Credentials Management	11

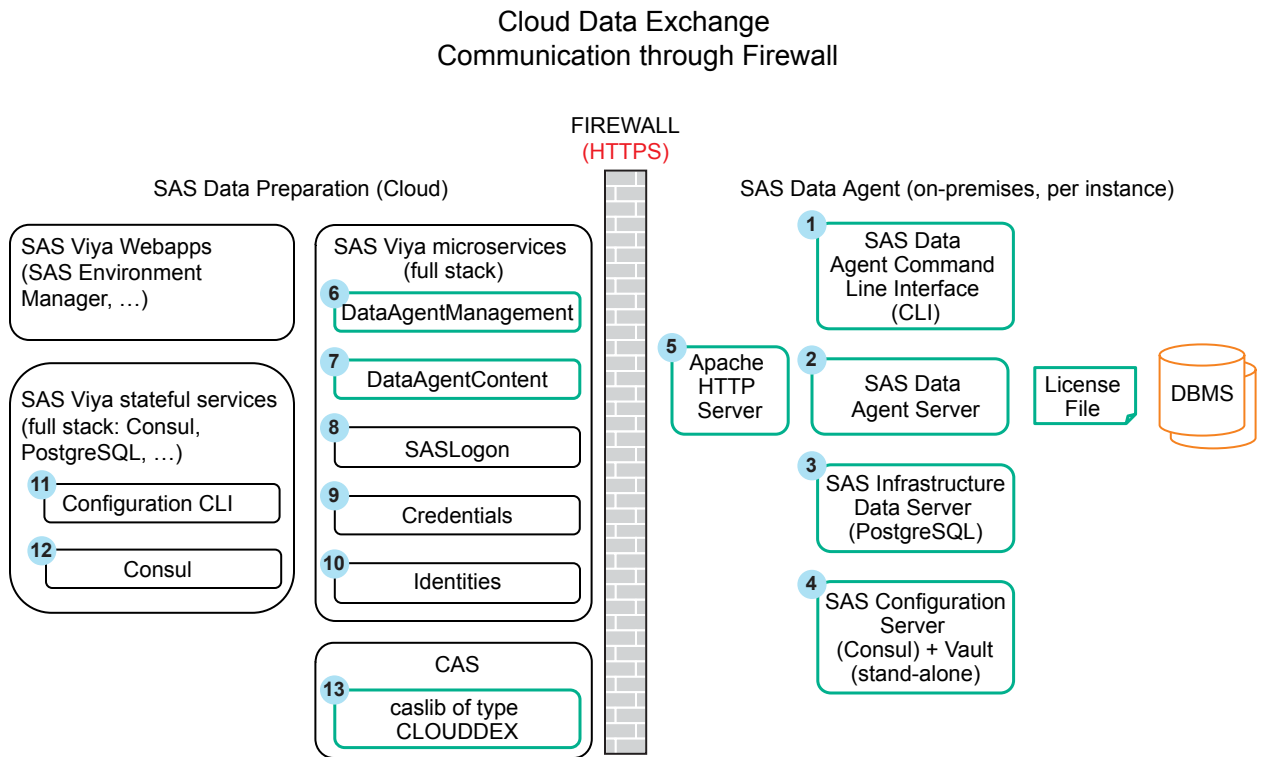
Security Features

Overview

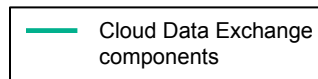
Cloud Data Exchange allows a customer to permit access to databases that are not co-located with a SAS Viya deployment. This is accomplished by using secure standards-based communication coupled with sophisticated authentication and authorization models.

This permits a deployment topology where a portion of Cloud Data Exchange, SAS Data Agent, resides outside of the rest of the SAS Viya deployment. This graphic illustrates the components that support Cloud Data Exchange. This also illustrates the split nature of the deployment. In this case, the description SAS Data Agent refers to the portion of Cloud Data Exchange that resides inside a customer's secured network. SAS Data Preparation in this topology indicates where the remainder of the SAS Viya deployment resides, which could be in a private or public cloud, or directly installed at a customer site but in a different secured network domain.

Figure 2.1 Cloud Data Exchange Communications Summary



Legend



- 1 SAS Data Agent Command Line Interface (CLI)
- 2 SAS Data Agent Server
- 3 SAS Infrastructure Data Server (PostgreSQL)
- 4 SAS Configuration Server (Consul) plus Vault
- 5 Apache HTTP Server
- 6 SAS Viya microservices: DataAgentManagement
- 7 SAS Viya microservices: DataAgentContent
- 8 SAS Viya microservices: SASLogon
- 9 SAS Viya microservices: Credentials
- 10 SAS Viya microservices: Identities
- 11 SAS Viya stateful services: Configuration CLI
- 12 SAS Viya stateful services: Consul
- 13 CAS: caslib and CLOUDEX connector

Communication Summary

The following table summarizes the communication paths represented in the Cloud Data Exchange Communications diagram, illustrating the standards support used at each level.

Table 2.1 *Cloud Data Exchange Communications*

Communication From/To	Description
SAS Data Agent server (2) to PostgreSQL (3) system catalog	SSL encrypted binary protocol with peer certificate validation.
SAS Data Agent server (2) to Consul (4)	HTTPS protocol with peer certificate validation, and token authentication.
SAS Data Agent server (2) to SAS Viya (8, 9, 10)	HTTPS protocol with peer certificate validation, OAUTH2 authentication, and optional forward proxy credentials.
Command Line Interface (1) to SAS Viya (6, 7, 8)	HTTPS protocol with peer certificate validation, OAUTH2 authentication, and optional forward proxy credentials.
CAS Data Connector (13) to SAS Data Agent server (2)	HTTPS protocol with peer certificate validation, OAUTH2 authentication, and optional forward proxy credentials.
SAS Viya (6, 7) to SAS Data Agent Server (2)	JDBC over HTTPS protocol with peer certificate validation, OAUTH2 authentication, optional forward proxy credentials

Topology

Cloud Data Exchange works with your firewall by opening one secured port rather than many, thereby reducing typical data security concerns. An Apache HTTP Server is provided for this purpose. With Cloud Data Exchange, SAS Data Agent can be placed in an organization's perimeter network (DMZ), whereas the data resides on LAN behind a firewall. This allows for monitoring of network traffic and isolation and protection of IT resources.

Secured Communications

Any communication that occurs between SAS Data Agent and SAS Data Preparation components, and any communication that occurs among SAS Data Agent components is performed using industry-standard SSL/TLS encryption. This means that any sensitive data (including credentials) that must be moved between components is protected. The various components depend in part on the customer system's OpenSSL and related libraries. This allows the customer to ensure that Cloud Data Exchange is using the most recent versions of those libraries.

See [Encryption in SAS Viya: Data in Motion](#) for additional information.

Industry Standard Transport

Communication between SAS Data Preparation and SAS Data Agent deployments is accomplished using HTTPS protocols. This supports a variety of network management capabilities that the customer or cloud provider can put in place to measure and control traffic coming to or from the internet.

The HTTPS transport navigates both forward- and reverse-proxy configurations, including proxy systems that require authentication credentials to be permitted to pass through to the internet. This gives the customer flexibility in network configuration and accommodating Cloud Data Exchange into the customer's own security protocols and standards.

The use of HTTP-based protocols also means that Cloud Data Exchange is resilient to transient network failures, such as the loss of a proxy server or router. Cloud Data Exchange can re-negotiate a database operation that was interrupted due to network management events.

Communication within the SAS Data Agent system is accomplished using SSL sockets to encrypt binary communications. The communication between the components supplied by SAS is accomplished using HTTPS protocol as well. Communication between the data agent server component and third-party databases can be encrypted but is typically not based on HTTP protocols.

Authentication

Certificate-Based Authentication

Cloud Data Exchange fully supports certificate-based authentication of each endpoint in a secured HTTPS transport, as well as communication within the SAS Data Agent deployment based on SSL socket protocols. This means that deployment includes exchange of identity certificates between SAS Data Preparation and SAS Data Agent systems, ensuring that spoofing and man-in-the-middle attacks are mitigated when moving sensitive data across the internet.

Cloud Data Exchange can be used with a trusted secure forward proxy if needed, allowing the customer to monitor and control traffic while maintaining secured communications over the public internet.

Token-Based Authentication

In addition to secured SSL communication between SAS Data Agent components, Cloud Data Exchange also uses token-based authentication to protect access to infrastructure components such as HashiCorp Consul and Vault services. These services support configuration, discovery, and secure credentials storage. Vault services are deployed as SAS Secrets Manger in SAS Viya.

These tokens are created as part of the deployment, protected by the native file system, and are accessible only to SAS Data Agent server components.

SASLogon Authentication

The SAS Data Agent server receives tokens from client connect requests. The tokens are authenticated by calling the SASLogon microservice in its associated SAS Data Preparation deployment. Token authentication requires use of an authentication client, which is configured in SAS Data Preparation and shared with SAS Data Agent. This occurs during the post-deployment steps of Cloud Data Exchange.

Data Access Controls

The SAS Data Agent server implements Cloud Data Exchange access controls to databases that reside within the same managed network as the database providers. This includes defining the locations, attributes, and security domains for access to the databases. This information is stored in a system catalog (SYSCAT) located in a PostgreSQL database (provided by SAS) that is deployed with the SAS Data Agent server.

Users can be placed in either the Data Agent Administrators group if they require administration privileges, or Data Agent Power Users group if they require data movement privileges. Users that are not in at least one of these groups, even if they are authenticated users, cannot connect to the data agent server.

In addition to secure access to RDBMS systems, SAS Data Agent also provides secure access to Base SAS data sets. Data sets can be placed under exclusive control of the SAS Data Agent server process, thereby forcing access to the data sets through SAS Data Agent where data access controls are applied.

Credentials Management

SAS Data Agent can be configured to access a database that requires credentials to serve those connections. Through use of a command-line interface, database credentials can be registered in SAS Secrets Manager stored with the SAS Data Agent such that enterprise secrets never leave the confines of the organization. Credentials can be assigned to individuals or groups known in the SAS Data Preparation environment. These credentials can be accessed from SAS Secrets Manager after those individuals are authenticated and pass authorization checks for data access.

Chapter 3

Configuring Cloud Data Exchange for CAS

Create a CAS Library (Caslib)	13
Overview	13
Prerequisites	14
Data Agent CAS Library Options	14
List Tables	18
List Table Columns	19
Load a Table	21
Parallel Loading for Faster Performance	23
Save a Table	26
Delete a Source Table	28
Drop a CAS Table	29
Remote Connectivity	30
Data Set Options	30
Connection String Options	30
Proxy Options	31
Working with Data Explorer	31
Create a Caslib Connection	31

Create a CAS Library (Caslib)

Overview

This topic describes the actions required to move data to and from Cloud Analytic Services (CAS) through Cloud Data Exchange (CDE). With Cloud Data Exchange, you can create a SAS Data Agent caslib within CAS and take advantage of the following actions:

- List tables
- List columns of a table
- Load a table from a data source to CAS
- Save a table to a data source from CAS
- Drop a table in the data source
- Drop a CAS table

Prerequisites

Following are a few prerequisites to address before working with CAS:

- To create a caslib, you must be a member of the SAS Administrators group. If you are a member of the Data Agent Power Users Group, you must obtain permission to create a Global CAS Lib. See “Adjust Caslib Management Privileges” in *SAS Viya Administration: SAS Cloud Analytic Services*.
- Use the CLI to add domains with outbound credentials for database access. See Domain (Credentials) in Data Administration for additional information.
- Register CLOUDEX with the CAS server.

Data Agent CAS Library Options

Load Data Source

Use the CAS action `loadDataSource` to configure a data connector for Cloud Data Exchange (CLOUDEX, alias CDE). The required parameters are:

Name	Type	Description	Default	Required
Data Connector Name	String	Data source type = “CLOUDEX” CDE is an alias for CLOUDEX.	None	Yes

Here is an example:

```
proc cas;
  session mysess;
  action loadDataSource / name="clouddex"; run;
quit;
```

Add a Caslib to Data Agent

After loading the data source, add a caslib using the CAS action `addCasLib`. The options are:

Name	Type	Description	Default	Required
<code>authenticationDomain authDomain</code>	string	Use an authentication domain to retrieve credentials.	none	See Credential Search Order for caslib
<code>casTypes</code>	Enumeric	BASIC (SAS data types) or ALL. For loadTable, both are supported. For saveTable, only BASIC is supported.	BASIC	no
<code>catalog</code>	string	Specifies catalog name for the tables.	default from conopts from caslib	no

Name	Type	Description	Default	Required
charCreateMultiplier	double	Specifies a multiplier value for target column width when creating a table with saveTable action. Use in the event that output fails because column width is insufficient to hold encoded NLS characters.	1	no
charMultiplier	double	Specifies a multiplier value to expand fixed-width character variables that might require transcoding.	1.0	no
conopts	string	Connection options with DSNs created under a SAS Data Agent.	none	yes
dataAgentName	string	SAS Data Agent name registered with SAS Viya Services.	none	Define either the data agent name or server
domainCredsRequired	Boolean	Indicates whether credentials for authentication domain are required. The DomainCredsRequired Option is relevant when the authenticationDomain (authDomain) option is specified. If credentials are searched from the domain, this option flags an error if credentials are not found. Either the domain does not exist, or the user has no credentials in the domain.	TRUE	no
dropTableOnFailedSave	Boolean	Drop the saved table if a failed saveTable action occurs.	TRUE	no
insertbuff	integer	Specifies number of rows to write to the data source in one call during saveTable action.	none	no
metaCatalog	string	Specifies catalog name for the metadata table.	same catalog as the loaded table	no
metaSchema	string	Specifies schema name for the metadata table.	same schema as the loaded table	no
metaTable	string	Specifies table name for the metadata table .	CDEMETA	no

Name	Type	Description	Default	Required
<code>ncharCreateMultiplier</code>	double	Specifies a multiplier value for target column width when creating a table with <code>saveTable</code> action. Use in the event that output fails because column width is insufficient to hold encoded NLS characters.	1	no
<code>ncharMultiplier</code>	double	Specifies a multiplier value to expand fixed-width national character variables that might require transcoding.	3.0	no
<code>numReadNodes</code>	integer	Specifies the number of grid nodes that connect to SAS Data Agent and read data from data sources in parallel during <code>loadTable</code> action	1	0 – all nodes connect
<code>numWriteNodes</code>	integer	Specifies the number of grid nodes that connect to SAS Data Agent and save data to data sources in parallel during <code>saveTable</code> action.	1	0 – all nodes connect <i>Note:</i> This value must be set at 1 when connecting with BASE. Multiple writers cannot connect to a SAS data set.
<code>password</code> <code>pass</code> <code>pwd</code>	string	SAS Data Agent User's password.	none	See Credential Search Order for <code>caslib</code> .
<code>port</code>	integer	Port number for SAS Data Agent.	25141	no
<code>readbuff</code>	integer	Specifies number of rows to read from the data source in one call during <code>loadTable</code> action.	none	no
<code>remoteDriverConnections</code>	string	Options for remote driver such as <code>"LOGIN_TIMEOUT=10;COMPRESS=1;"</code> . See Remote Connectivity for additional connection options.	none	Retrieved from <code>dataAgentContentService</code> if data agent name (<code>dataAgentName</code>) is defined.

Name	Type	Description	Default	Required
schema	string	Specifies schema name for the tables.	default from conopts from caslib	no
server	string	Host name for SAS Data Agent.	none	Define either the data agent name or server.
srcType	string	CLOUDDX is required for SAS Data Agent caslib.	none	yes
traceFile	string	Specifies a name for the trace file.	none	no
traceFlags	string	Specifies flags to use for tracing the database connections.	none	no
uid username	string	SAS Data Agent User ID or user name.	none	See “Credential Search Order” for caslib.
useMaxCharLen	Boolean	Use max character length from the CAS table when creating table for saveTable.	FALSE	no
useMetaTable	Boolean	Use the metadata table for parallel data load during loadTable.	TRUE	no
useMinMaxToSplit	Boolean	Split data among reader nodes automatically with equal increment of $(\text{max} - \text{min}) / \text{numReadNodes}$.	FALSE	no
useNarrowCharacter Types	Boolean	For DBMS like DB2, default database is Unicode, and CHAR/VARCHAR can contain UTF-8. If set to true , it then prevents performance penalty of converting CAS UTF-8 into UCS-2.	FALSE	no

addCaslib Examples

The following example uses the data_agent_name instead of server and port:

```
proc cas;
  session mysess;
  action addCaslib / caslib="dalib_td"
    session=false
    datasource={srcType="clouddex",
                username="dbuser",
```

```

        password="password",
        dataAgentName="dagentsrv-shared-default",
        catalog="GRIDLIB",
        schema="model",
        conopts="dsn=dsn_tera"
    };

run;
quit;

```

This example shows multiple DSNs in the conopts string:

```

proc cas;
  session mysess;
  action addCaslib / caslib="dalib_multidr"
    session=false
    datasource={srcType="cde",
      username="dbuser",
      password="password",
      port="1337",
      server="servername.url", /*host name for Data Agent */
      catalog="GRIDLIB",
      schema="model",
      conopts="( driver=FEDSQL; conopts=((dsn=dadsn_base);
        (dsn=dadsn_tera)) )",
      remoteDriverOptions="COMPRESS=1"
    };

run;
quit;

```

Credential Search Order

Credential search order for caslib follows this sequence:

- Use the user name and password

First, use the user name and password if specified. If not, the system looks for an authentication domain.
- Use Authentication Domain

Second, use the authentication domain if user name and password is not specified. When domainCredsRequired is set to true, an error is returned if no credentials are found for the authDomain. When domainCredsRequired is set to false, and no credentials are found, the system looks for an OAuth token.
- Retrieve OAuth token for CAS user

Third, the system retrieves an OAuth token for the CAS user and uses the token to connect. If neither user name and password or authenticationDomain is specified, the system retrieves an OAuth token for the connection.

List Tables

Use the CAS action `fileInfo` to list tables under a data source. The options are:

Name	Type	Description	Default	Required
<code>uid</code> <code>username</code>	string	SAS Data Agent user name	none	no

Name	Type	Description	Default	Required
<code>pwd</code> <code>password</code> <code>pass</code>	string	SAS Data Agent user's password	none	no
<code>authenticationDomain</code> <code>authDomain</code>	string	Authentication domain for credentials	none	no
<code>domainCredsRequired</code>	Boolean	Indicates whether credentials for authentication domain are required. The <code>DomainCredsRequired</code> Option is relevant when the <code>authenticationDomain</code> (<code>authDomain</code>) option is specified. If credentials are searched from the domain, this option flags an error if credentials are not found. Either the domain does not exist, or the user has no credentials in the domain.	TRUE	no
<code>catalog</code>	string	Specifies catalog name for the tables.	default from conopts from caslib	no
<code>schema</code>	string	Specifies schema name for the tables.	default from conopts from caslib	no
<code>tracefile</code>	string	Specifies a name for the trace file.	none	no
<code>traceFlags</code>	string	Specifies flags to use for tracing the database connections.	none	no

Here is an example:

```
proc cas;
  session mysess;
  action fileInfo / casLib="dalib_td";
run;
quit;
```

List Table Columns

Use the CAS action `columnInfo` to list columns within a table. The options are:

Name	Type	Description	Default	Required
uid username	string	SAS Data Agent user name	none	no
pwd password pass	string	SAS Data Agent user's password	none	no
authentication Domain authDomain	string	Authentication domain for credentials	none	no
domainCredsRequired	Boolean	Indicates whether credentials for authentication domain are required. The DomainCredsRequired Option is relevant when the authenticationDomain (authDomain) option is specified. If credentials are searched from the domain, then this option flags an error if credentials are not found. Either the domain does not exist, or the user has no credentials in the domain.	TRUE	no
catalog	string	Specifies catalog name for the tables.	default from conopts from caslib	no
schema	string	Specifies schema name for the tables.	default from conopts from caslib	no
tracefile	string	Specifies a name for the trace file.	none	no
traceFlags	string	Specifies flags to use for tracing the database connections.	none	no

Here is an example:

```
proc cas;
  session mysess;
  action ColumnInfo / table={name="cars", casLib="dalib_td"};
run;
quit;
```


Load a Table

CAS Action: **LoadTable**

You can use **loadTable** action to load data from a data source to CAS. When you load a table to CAS, the scope of the CAS table could be session or global. If it is a global table, then it exists until you drop the table explicitly. If it is a session table, then it is dropped when the session is terminated.

The options for **LoadTable** are:

Name	Type	Description	Default	Required
uid username	string	SAS Data Agent User name	none	no
pwd password pass	string	SAS Data Agent user's password	none	no
authenticationDomain authDomain	string	Authentication domain for credentials	none	no
domainCredsRequired	Boolean	Indicates whether credentials for authentication domain are required. The DomainCredsRequired Option is relevant when the authenticationDomain (authDomain) option is specified. If credentials are searched from the domain, then this option flags an error if credentials are not found. Either the domain does not exist, or the user has no credentials in the domain.	TRUE	no
catalog	string	Specifies catalog name for the tables.	default from conopts from caslib	no
schema	string	Specifies schema name for the tables.	default from conopts from caslib	no
readbuff	integer	Specifies number of rows to read the data source in one call during loadTable action.	not applicable	no

Name	Type	Description	Default	Required
castTypes	Enumeric	BASIC or ALL. For loadTable, both are supported. For saveTable, only BASIC is supported.	BASIC	no
charMultiplier	double	Specifies a multiplier value to expand fixed-width character variables that might require transcoding.	1.0	no
ncharMultiplier	double	Specifies a multiplier value to expand fixed-width national character variables that might require transcoding.	3.0	no
dbmsWhere	string	Used to filter data in WHERE statement when loading data into CAS.	none	no
numReadNodes	integer	Specifies the number of grid nodes that connect to SAS Data Agent and load data from data sources in parallel during loadTable action.	1	0 = all nodes connect
dataOnReaderNodesOnly	Boolean	Data loaded onto the reader nodes only when numReadNodes > 1 .	FALSE	no
useMetaTable	Boolean	Use the metadata table for parallel data load during loadTable .	TRUE	no
metaCatalog	string	Specifies catalog name for the metadata table.	same catalog as the loaded table	no
metaSchema	string	Specifies schema name for the metadata table.	same schema as the loaded table	no
metaTable	string	Specifies table name for the metadata table.	CDEMETA	no
useMinMaxToSplit	Boolean	Split data among reader nodes automatically with equal increment of $(\text{max} - \text{min}) / \text{numReadNodes}$.	FALSE	no
splitColumn	string	Specifies a column name used to split data for parallel data loading.	none	no

Name	Type	Description	Default	Required
splitRange	string	Specifies increment value starting from minimum value when splitting data among reader nodes.	none	no
partition	string	Specifies list of columns to group the data when loaded to CAS.	none	no
orderBy	string	Specifies list of columns by which to order the data.	none	no
descending	string	Specifies a list of 0s/1s to indicate ascending/ descending order for the orderBy columns.	none	no
traceFile	string	Specifies a name for the trace file.	none	no
traceFlags	string	Specifies flags to use for tracing the database connections.	none	no

Parallel Loading for Faster Performance

Parallel Loading Data to CAS

Multiple grid nodes, or reader nodes, can connect to SAS Data Agent. Each node can fetch data concurrently when **numReadNodes** is greater than **1** (default). Depending on your system configuration, there can be a limit on the number of nodes that can connect simultaneously to SAS Data Agent. Data is distributed to all grid worker nodes when **dataOnReaderNodesOnly** is set to **FALSE**, which is the default. However, data resides on the reader nodes when **dataOnReaderNodes** is set to **TRUE**. There are two ways to split the data fetched by the reader nodes:

1. Create a metadata table by setting **useMetaTable** to **TRUE** (default), or
2. Let the software split the data fetched among the reader nodes by setting **useMinMaxToSplit** to **TRUE** (the default is **FALSE**).

Creating a Metadata Table for Parallel Loading

To control the amount of data that is fetched by each of the reader nodes, define a metadata table that contains the columns shown below:

Table 3.1 Metadata Table for Parallel Loading

Column	Data Type	Length
CATALOG_NAME	CHAR	256

Column	Data Type	Length
SCHEMA_NAME	CHAR	256
TABLE_NAME	CHAR	256
COLUMN_NAME	CHAR	256
HISTOGRAM_ENTRY	integer	4 bytes
HISTOGRAM_BOUNDS	CHAR	256
SPLIT	integer	4 bytes

The CATALOG_NAME, SCHEMA_NAME, and TABLE_NAME columns are used to identify the table to load. COLUMN_NAME defines the column used to split the data to be fetched. The column SPLIT defines the number of data splits to fetch concurrently by multiple reader nodes. For a given SPLIT number, there are as many entries (HISTOGRAM_ENTRY) as (SPLIT - 1) of splits with a bound value (HISTOGRAM_BOUNDS) defined for each entry. An example is shown in the following table.

Table 3.2 Example of Splitting Data for Fetching

CATALOG_NAME	SCHEMA_NAME	TABLE_NAME	COLUMN_NAME	HISTOGRAM_ENTRY	HISTOGRAM_BOUNDS	SPLIT
teralib	model	bigcars	price	1	20000	4
teralib	model	bigcars	price	2	30000	4
teralib	model	bigcars	price	3	40000	4
teralib	model	bigcars	price	1	10000	5
teralib	model	bigcars	price	2	20000	5
teralib	model	bigcars	price	3	30000	5
teralib	model	bigcars	price	4	40000	5

In the example table above, two ways of splitting data are configured: SPLIT=4 or SPLIT=5. When the parameter **numReadNodes** is set to **4**, the configuration of SPLIT=4 is used for four reader nodes to fetch data simultaneously. Nodes 1 through 4 behave differently:

- The first node fetches the data where price < 20000 and all the rows with price being NULL.
- The second node fetches the data where 20000 <= price < 30000.
- The third node fetches the data where 30000 <= price < 40000.
- The fourth node fetches the data where price >= 40000.

An error is returned if the split number does not match the number of reader nodes (`numReadNodes`). For example, if `numReadNodes` were set to 3, an error is returned given that the split number is 4 and 5. The number of reader nodes is reset to the number of worker nodes available if the number of reader nodes defined is more than the number of workers available.

Split Data Programmatically

The software can programmatically split data fetched among the reader nodes if `useMinMaxToSplit` is set to **TRUE**, and `useMetaTable` to **FALSE**. The following series of events takes place:

- If defined, the parameter `splitColumn` is used to split the data fetched. Otherwise, the software looks for the first numeric column and uses that column to split the data that is fetched.
- When `splitRange` is not defined, the software splits the data among reader nodes with equal increments of $(\text{max} - \text{min}) / \text{numReadNodes}$.
- When `splitRange` is defined, each increment fetched by a worker node is defined by the value of `splitRange` beginning with the minimum value of `splitColumn`, which is required when `splitRange` is used.
- The last of the reader nodes, which is the total of `numReadNodes`, fetches the remaining data.

By using `splitRange`, you can leverage splitting data evenly among the reader nodes, especially if the data is not uniformly distributed.

Examples of loadTable Action

```
proc cas;
  session mysess;
  action loadTable / casLib="dalib_td" path="cars" promote=TRUE;
run;
quit;

/* use CDEMETA table to split the data for parallel loading */
proc cas;
  session mysess;
  action loadTable / casLib="dalib_td" path="bigcars" promote=TRUE
    datasourceOptions={numReadNodes=2,
                       metaCatalog="GRIDLIB",
                       metaSchema="model",
                       metaTable="cdemeta"}
    casout={ casLib="dalib_teradoc" } ;
run;
quit;

/* use minmax values to split the table for parallel loading,
where minmax values are being determined by software automatically */
proc cas;
  session mysess;
  action loadTable / casLib="dalib_td" path="bigcars" promote=TRUE
    datasourceOptions={numReadNodes=2,
                       useMetaTable=FALSE,
                       useMinMaxToSplit = TRUE,
                       splitColumn="slice",
                       splitRange = 10000}
run;
quit;
```

```

                                casout={ casLib="dalib_teradoc" } ;
run;
quit;

```

Save a Table

Use the CAS action **saveTable** to save a CAS table to a data source. The options are:

Name	Type	Description	Default	Required
uid username	string	SAS Data Agent User name	none	no
pwd password pass	string	SAS Data Agent user password	none	no
authenticationDomain authDomain	string	Authentication domain for credentials	none	no
domainCredsRequired	Boolean	Indicates whether credentials for authentication domain are required. The DomainCredsRequired Option is relevant when the authenticationDomain (authDomain) option is specified. If credentials are searched from the domain, then this option flags an error if credentials are not found. Either the domain does not exist, or the user has no credentials in the domain.	TRUE	no
catalog	string	Specifies catalog name for the tables.	default from conopts from caslib	no
schema	string	Specifies schema name for the tables.	default from conopts from caslib	no
insertbuff	integer	Specifies number of rows to write to the data source in one call during saveTable action.	not applicable	no
casTypes	Enumeric	BASIC or ALL. For loadTable, both are supported. For saveTable, only BASIC is supported.	BASIC	no

Name	Type	Description	Default	Required
charCreateMultiplier	double	Specifies a multiplier value for target column width when creating a table with saveTable action. Use in the event that output fails because column width is insufficient to hold encoded NLS characters.	1	no
charMultiplier	double	Specifies a multiplier value to expand fixed-width character variables that might require transcoding.	1.0	no
ncharCreateMultiplier	double	Specifies a multiplier value for target column width when creating a table with saveTable action. Use in the event that output fails because column width is insufficient to hold encoded NLS characters.	1	no
ncharMultiplier	double	Specifies a multiplier value to expand fixed-width national character variables that might require transcoding.	3.0	no
useMaxCharLen	Boolean	Use max character length from the CAS table when creating table for saveTable.	FALSE	no
useNarrowCharacterTypes	Boolean	For a DBMS like DB2, default database is a Unicode, and CHAR/VARCHAR can contain UTF-8. If set to true , it then prevents performance penalty of converting the CAS UTF-8 into the UCS-2.	FALSE	no
dbCreateTableOptions	string	Specifies DBMS create table options for the table to be saved to the data source.	none	no

Name	Type	Description	Default	Required
numWriteNodes	integer	Specifies the number of grid nodes that connect to SAS Data Agent and save data to data sources during saveTable action.	1	0 = all nodes connect <i>Note:</i> This value must be set at 1 when connecting with BASE SAS or Teradata. Multiple writers cannot connect to these data sources.
dropTableOnFailedSave	Boolean	Drop the saved table when a failed saveTable action occurs.	TRUE	no
traceFile	string	Specifies a name for the trace file.	none	no
traceFlags	string	Specifies flags to use for tracing the database connections.	none	no

```
proc cas;
  session mysess;
  action save / table={caslib="dalib_td" name="cars" }
    name="cars_save"
    caslib="dalib_teradoc"
    options={catalog="GRIDLIB",
      schema="model",
      dbCreateTableOpts="primary index(make)"}
    replace=1;
run;
quit;
```

Delete a Source Table

Use the CAS action **deleteSource** to delete a table from a data source. The options are:

Name	Type	Description	Default	Required
uid username	string	SAS Data Agent User name	none	no
pwd password pass	string	SAS Data Agent user password	none	no
authentication Domain authDomain	string	Authentication domain for credentials .	none	no

Name	Type	Description	Default	Required
domainCredsRequired	Boolean	Indicates whether credentials for authentication domain are required. The DomainCredsRequired Option is relevant when the authenticationDomain (authDomain) option is specified. If credentials are searched from the domain, then this option flags an error if credentials are not found. Either the domain does not exist, or the user has no credentials in the domain.	yes	FALSE
catalog	string	Specifies catalog name for the tables.	default from conopts from caslib	no
schema	string	Specifies schema name for the tables.	default from conopts from caslib	no
tracefile	string	Specifies a name for the trace file.	none	no
traceFlags	string	Specifies flags to use for tracing the database connections.	none	no

```
proc cas;
  session mysess;
  action deleteSource / source="cars_save" caslib="dalib_td";
run;
```

Drop a CAS Table

The CAS action **dropTable** is used to drop a CAS table to release the resources.

```
proc cas;
  session mysess;
  action dropTable / table="cars" caslib="dalib_td";
run;
quit;
```

Remote Connectivity

Data Set Options

All data set options are sent to SAS Data Agent as is, without modification. The remote driver does not currently implement any data set options.

Connection String Options

SERVER= <hostname>

The host name of the SAS Data Agent to connect to. (Required)

PORT= <port-number>

Specifies the port number for SAS Data Agent. (Required)

CONOPTS= <(connection-string)>

The connection string to pass to SAS Data Agent for the underlying connection.

UID= <user-name>

The user name to use when connecting to SAS Data Agent.

PWD= <user-password>

The password to use when connecting to SAS Data Agent.

TOKEN= <OAuth-token>

The OAuth token to use when connecting to SAS Data Agent.

CON_TIMEOUT | CTO= <connection-timeout-in-seconds>

This option defaults the attribute `TKTS_ATTR_CONNECTION_TIMEOUT`, which specifies the number of seconds the REMFS driver should wait when performing a non-Statement call (except for `DriverConnect`, which uses `LOGIN_TIMEOUT`). The default is 0.

LOGIN_TIMEOUT | LTO= <login-timeout-in-seconds>

This option defaults the attribute `TKTS_ATTR_LOGIN_TIMEOUT`, which specifies the number of seconds the REMFS driver should wait when performing a `TKTSDriverConnect` call. The default is 0.

SESSION_TIMEOUT | IDLE_TIMEOUT= <session-timeout-in-seconds>

If the network connection between the client and the server closes, the server keeps the session alive for the specified time, waiting for the client to reconnect. The value 0 means use the default specified on the server. The value 1 means clean up immediately. The default configured on the server is used if a time-out is not specified, and might also be limited by a separate maximum configured on the server.

COMPRESS= <0-6>

Specifies the amount of data compression to apply. 0=None, 1=Fastest, and 6=Smallest. Values larger than 6 are treated as Smallest. The default is 0.

SSL | TLS= <(ssl-option=ssl-value;ssl-option=ssl-value;...)>

SSL Options.

PROXY= <(proxy-option=proxy-value;proxy-option=proxy-value;...)>

Proxy server options. See Proxy Options for specific names and option values. The default is `none`. This indicates the connection is made directly to the FSNET interface of SAS Data Agent.

Proxy Options

SERVER= <proxy-hostname>

The host name of the HTTP proxy to use. (Required if the PROXY= connection option is specified.)

PORT= <port-number>

The numeric port number for the proxy. (Required if the PROXY= connection option is specified)

UID= <username> ¹

The user name to use when connecting to the proxy server.

PWD= <password> ¹

The password to use when connecting to the proxy server.

Working with Data Explorer

Create a Caslib Connection

A caslib is an in-memory space to hold tables, access control lists, and data source information. One property of a caslib is scope. A caslib can have one of two scopes: session scope or global scope. After you are finished creating a caslib for SAS Data Agent, you can create a database connection with SAS Data Explorer, accessed through SAS Environment Manager.

In SAS Environment Manager, select  **Data** from the navigation menu to begin. Use these tasks to create a connection for Cloud Data Exchange:


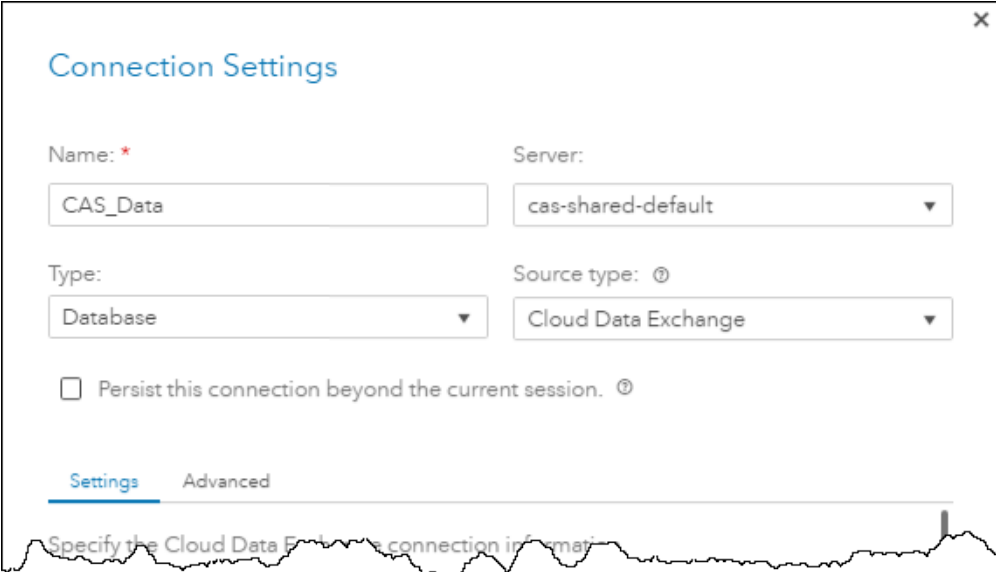
1. Select **Data Sources**
2. Click **Connect**  to display the Connection Settings dialog box.

Figure 3.1 Connection Settings Dialog Box



Connection Settings

Name: * Server:

Type: Source type: ⓘ

Persist this connection beyond the current session. ⓘ

Settings Advanced

Specify the Cloud Data Exchange connection information

3. Enter a name for the caslib in the **Name** field.
4. Select a CAS server in the **Server** field.
5. Select **Database** in the **connection type** field.
6. Select **Cloud Data Exchange** in the **Source type** field.
7. Select **Persist this connection beyond the current session** if it applies.

Note: If you select the **Persist this connection beyond the current session** check box, and you specify a connection, the system creates a global caslib. Your login must have the privilege required to create a global caslib, or the connection will fail. If this option is not selected, the system creates a session caslib.

8. Enter the following under the **Settings** tab:
 - User ID and Password (optional)
 - Data Agent DSN
 - Select a server at **Use registered Data Agent**.

Note: You can also enter a server name and port to identify a server.

9. (Optional) Select **Test Connection**.

Note: Verify that the message ‘The connection was successful’ is displayed in the top banner.

10. Select **Save**.

Use the **Advanced** tab to add additional connection information such as domain, number of read nodes, and number of write nodes, and so on. Each value represents a name/value pair. There is no need to enclose in quotation marks unless the option is nested within another option. Select **Add Parameter** if you run out of fields.

For additional information, see [“Understanding SAS Data Explorer” in SAS Data Explorer: User’s Guide..](#)

Chapter 4

Unregister Data Agents

SAS Data Preparation Machine: Unregister SAS Data Agent	33
Remove Data Agent Registration	33
Argument Details	33
Remove Consul Key	34
Rebuild the Truststore	34
Delete the Proxy Certificate	34

SAS Data Preparation Machine: Unregister SAS Data Agent

Remove Data Agent Registration

Use the script `da_unreg_server.sh` to remove a registered data agent from the CAS controller (Data Preparation deployment). The script is located in this directory path: `/opt/sas/viya/home/bin/`. If SAS Data Preparation is located on multiple machines, the `da_unreg_server.sh` script is located on the CAS controller. Logged on as `sas user`, run the script on the SAS Data Preparation machine for each data agent registration that you want to remove. You can also remove an entire tenant by including the option `--removeoauthclient Y`.

Note: If you want to remove data agents but keep at least 1 active registration, do not include the `--removeoauthclient Y` option.

For a single-tenant deployment, specify:

```
--customerid
--sas-endpoint
```

For a multitenant deployment, specify:

```
--tenantid
--provider-endpoint
--tenant-endpoint
```

Argument Details

The script detects any required options that are missing and displays all required options along with their values. A blank value indicates that a value needs to be specified.

```

--removeoauthclient <Y|N>
  Y specifies that the OAuth client be removed as well. By default, the script does not
  remove the OAuth client registration.

--sasadministratoruser <SAS-Viya-admin-user>
  SAS Viya administration user. Required option.

--sasadministratorpassword <admin-user-password>
  SAS Viya administration user password. Required option.

--registeredservername server-name
  name of the server to be removed.

```

Here is an example that removes registration for a single data agent server:

```

/opt/sas/viya/home/bin/da_unreg_server.sh --tenantid ted1
--registeredservername secondary_data_agent
--provider-endpoint https://provider.home.com:443
--tenant-endpoint https://mysite.provider.home.com:443
--sasadministratoruser adminuser
--sasadministratorpassword adminpwd

```

Remove Consul Key

After the registration has been removed, you can remove the associated key from Consul. Here are the commands:

Note: This task is optional.

```

source /opt/sas/viya/config/consul.conf

export CONSUL_HTTP_SSL=true

/opt/sas/viya/home/bin/sas-bootstrap-config --token-file
/opt/sas/viya/config/etc/SASSecurityCertificateFramework/tokens/consul/default/
client.token kv delete "cacerts/op1"

```

where `op1` is a name given to the HTTP certificate at the time of deployment.

Note: Multiple lines are used for the command above to improve readability. You should enter the command on a single line.

Rebuild the Truststore

1. Run this command from the Ansible controller machine: `cd sas_viya_playbook.`
2. Rebuild the truststore, without the cloud certificate, using the Ansible-playbook: `ansible-playbook utility/rebuild-trust-stores.yml.`

Delete the Proxy Certificate

Delete the HTTP certificate file copied from the on-premises machine during post-installation Step 1 for cloud:

```
/tmp/httpproxy-certificate-ca.crt
```

where *httpproxy-certificate* is the name of the HTTP proxy certificate that was copied to the SAS Data Preparation machine. The certificate might have been copied to the `/tmp` directory as suggested in the "[SAS Data Agent 2.3 for Linux: Deployment Guide](#)".

Important: Remove only the certificate that correlates to the data agent that you are removing.

Part 2

SAS Data Agent

<i>Chapter 5</i>	
Working with SAS Data Agent	39
<i>Chapter 6</i>	
Data Administration	51
<i>Chapter 7</i>	
Server Administration	79

Chapter 5

Working with SAS Data Agent

Getting Started	39
Before You Begin	39
SAS Data Agent Server Initialization	39
Add Users to Groups	40
Set the SSL_CERT_FILE Environment Variable	40
Download the CLI Executable	41
Prepare Your Data Sources	42
The Caslib Data Connector	44
SAS Data Agent Configuration	45
Authentication Service Cache	45
Environment Variables	45
Locale Support	48
Removing Data Agent Servers	48
Suspending Connectivity to SAS Data Preparation	48
Stop (Shut Down) the SAS Data Agent Server	48
Uninstall SAS Viya	48
Clean up Residual Items	49

Getting Started

Before You Begin

Prior to using SAS Data Agent server, ensure that you have completed all of the deployment and post-installation tasks. See the [SAS Data Agent 2.3 for Linux: Deployment Guide](#) for additional information. Here is a summary of administrative tasks for SAS Data Agent.

SAS Data Agent Server Initialization

Server Start-Up

SAS Data Agent starts after the server boots up. On start-up of the server, SAS Data Agent reads PostgreSQL credentials from SAS Secrets Manager and establishes a persistent connection to the PostgreSQL Server. After connection is made, SAS Data Agent initializes the system catalog database. The system catalog contains configuration

metadata. Configuration metadata includes a list of existing data services, DSNs, privileges, and other information generated as a result of configuring SAS Data Agent.

Add Users to Groups

Data Agent Administrators and Data Agent Power Users are the default groups created during deployment. If other group names were specified during deployment, the applicable group names are represented by `DA_ADMIN_GRP` and `DA_POWER_GRP` in the `/etc/sysconfig/sas/sas-viya-dagentsrv-default` configuration file. If you are using non-default names for groups, the following lines should reflect the group names configured during deployment:

```
DA_ADMIN_GRP="your-Data-Agent-Administrators-group"
DA_POWER_GRP="your-Data-Agent-Power-Users-group"
```

SAS Data Agent also supports SAS Administrator groups.

The following table shows the default permissions granted to the administrators and power users groups:

Table 5.1 SAS Data Agent Groups and Permissions

SAS Administrators	SAS Data Agent Administrators (DA_ADMIN_GRP)	SAS Data Agent Power Users (DA_POWER_GRP)
Create domains.	Create domains.	Add credentials for themselves to a domain.
Add credentials for other users on their behalf.	Create data services, add credentials for other users with CLI (for example, catalogs and schemas).	Connect with the DSNs and services created by the administrator.
Create caslib.	Create dsns.	Fetch data.
		Write back data.

Determine the users that should go into each group and populate the groups using SAS Environment Manager. The information displayed for users and groups comes from your organization's directory service (such as LDAP or Microsoft Active Directory). Because this information is managed by your identity provider, it is displayed as read-only data in SAS Environment Manager. See [Add or Remove Custom Group Members](#) for additional information.

Set the `SSL_CERT_FILE` Environment Variable

If your environment is enabled for Transport Layer Security (TLS), you must set the `SSL_CERT_FILE` environment variable to one of the following: to the path location of the `trustedcerts.pem` file (if using the SAS default truststore) or the path location of your site-signed certificate (if using an internal truststore).

- the path location of the `trustedcerts.pem` file (if using the SAS default truststore)
- the path location of your site-signed certificate (if using an internal truststore)

Set the `SSL_CERT_FILE` environment variable to the following value:

Truststore	Operating System	Path
SAS default truststore	Linux	/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.pem
internal truststore	Linux or Windows	<i>path_to_certificate</i>

Here is an example of how to set the environment variable in a Linux environment:

```
export SSL_CERT_FILE=/opt/sas/viya/config/etc/SASSecurityCertificateFramework/cacerts/trustedcerts.pem
```

Download the CLI Executable

You can download the admin CLI directly from the SAS Support website, and install the plug-ins that you need. This is the recommended approach for running the admin CLI from client machines. The most current plug-ins are available with the downloaded version of the admin CLI. Here are the steps to download the admin CLI and install the plug-ins:

1. Go to the [SAS Support / Downloads and Hot Fixes](#) site, and download the appropriate file to your machine.
2. Prepare the files for your environment:

- a. Expand the file to a directory from which you plan to run the CLI:

```
tar -xvzf /CLI-directory/sas-admin-cli-1.1.2-download-linux.tgz
```

- b. From the directory, make sure that the execution permission is set:

```
chmod +x sas-admin
```

CAUTION:

Do not place the admin CLI download file in the following directory: **/opt/sas/viya/home/bin**.

3. Navigate to the directory location where you saved the CLI file. You must run the CLI commands directly from this directory, or you can add this directory to your system path to make the sas-admin CLI available to all CLI users.
4. Create a profile and sign in.
5. Install the CLI plug-ins by running these commands:

- a. Displays the commands that are available to the admin CLI plug-ins command:

```
sas-admin plugins
```

- b. Displays the admin CLI plug-ins that are currently installed:

```
sas-admin plugins list
```

- c. Lists the plug-ins in the SAS repository that are available for installing:

```
sas-admin plugins list-repo-plugins
```

- d. Installs a plug-in from the SAS repository:

```
sas-admin plugins install --repo SAS plugin-name
```

Note: The plug-ins are installed in the home directory of the user who ran the **sas-admin plugins** command: **home-directory/.sas/admin-plugins**.

Prepare Your Data Sources

The *dagentsrv* CLI

The SAS Data Agent administrator uses a CLI to create definitions such as data services and data source names. The CLI is used to define security domains for accessing data sources that are located on-premises. The CLI is located in `/opt/sas/viya/home/bin/sas-admin`. See [dagentsrv CLI](#) for details.

Create Data Services

Before creating a caslib, you must create a data service that describes the connection to your data source. Data services form the foundation for connectivity to a data source (for example, the server where a data source is located is stored in a data service). When a data service is created, a Data Source Name (DSN) with the same name is created automatically. See “[Working with Data Services](#)” for additional information about data services.

The following examples demonstrate the necessary steps to create access for your data sources. After your data services are created, you are ready to configure a caslib for Cloud Data Exchange.

Example: Provisioning Access to Oracle Data

Use these steps to provision access for an Oracle database. The first step is to create a user profile that sets the location of the SAS Viya machine. Use the `sas-admin` CLI to create a profile.

1. Use the `sas-admin` CLI to create a profile.

```
[centos@datahub ~]$ /opt/sas/viya/home/bin/sas-admin profile init
Enter configuration options:

Service Endpoint> https://viya.dmmdev.sashq-d.openstack.sas.com:443

Output type (text|json|fulljson)> text

Enable ANSI colored output (y/n)?> y
Saved 'Default' profile to /home/centos/.sas/config.json.
```

2. Log on to SAS Viya, which grants access to SAS Viya services:

```
[centos@datahub ~]$ /opt/sas/viya/home/bin/sas-admin auth login
Enter credentials for https://viya.dev.openstack.abc.com:443:

Userid> dbusers

Password>
Login succeeded. Token saved.
```

- Determine whether a server is available and configured for use. Since there is only one server, you do not need to specify the server name on each command.

```
[centos@datahub ~]$ /opt/sas/viya/home/bin/sas-admin dagentsrv data-agents list
```

Name	Host	Port
dagentsrv-shared-default	datahub.dev.openstack.abc.com	443

```
Attributes
contextroot=dagentsrv-shared-default-private
```

- Create a domain that can handle securing access to an Oracle instance. You will also add credentials to the domain so that SAS Data Agent knows how to connect to Oracle when a request comes in for user “dbusers”.
- Confirm that the credentials were created. The **ShadowKey** is the unique ID of the password, which was stored on the data agent machine on your behalf, for use by SAS Data Agent.

Note: In this example, replace username “bobsmith” and password “a3b3ub35” with a user and password known to your Oracle system.

```
[centos@datahub ~]$ /opt/sas/viya/home/bin/sas-admin dagentsrv security domains
create --domain OracleUsers
```

```
[centos@datahub ~]$ /opt/sas/viya/home/bin/sas-admin dagentsrv security credentials
create --domain OracleUsers --identity dbusers --username bobsmith --password a3b3ub35
```

```
[centos@datahub ~]$ /opt/sas/viya/home/bin/sas-admin dagentsrv security credentials
list --domain OracleUsers
```

Identity	Type	Username	ShadowKey
dbusers	user	bobsmith	f3213876-5e0d-4981-b360-8d9ea8fd5d34

- Create an Oracle data service and include the following items:
 - Name the data service (which is one way a SAS Data Preparation user of Cloud Data Exchange might identify this data).
 - Describe the Oracle path, which defines where the server is located.
 - Provide the security domain previously created,
 - Add a catalog name that can be used to reference the data source.

```
[centos@datahub ~]$ /opt/sas/viya/home/bin/sas-admin dagentsrv data-services
create oracle --name HRData --path hr_server --domain OracleUsers --catalog HRCAT
```

- You can use this data source and catalog to see what schemas are already defined in the Oracle database. This operation not only shows you information about the configuration for Oracle, but also verifies that SAS Data Agent can successfully access Oracle using the information provided in the data service.

```
[centos@datahub ~]$ /opt/sas/viya/home/bin/sas-admin dagentsrv data-sources
catalogs list --data-source HRData --catalog HRCAT
```

```
Catalog
HRCAT
```

Now you are ready to configure a SAS Viya caslib that uses Cloud Data Exchange to access your Oracle database identified by the data source name of “HRData”.

Example: Provisioning Access to SAS Data Sets

You can also include SAS data sets as data stored on your SAS Data Agent, to make available on the network with Cloud Data Exchange.

1. SAS Data Agent comes with a Base SAS data service already configured, so you just need to create a catalog and define a schema before accessing the data.

```
[centos@datahub ~]$ /opt/sas/viya/home/bin/sas-admin dagentsrv catalogs
create base --name BASECAT
```

```
[centos@datahub ~]$ /opt/sas/viya/home/bin/sas-admin dagentsrv schemas
create base --catalog BASECAT --name SCH1 --primary-path "/data/sales/sasdata/"
```

2. After specifying a catalog and schema name for Base SAS data sets, you can access the contents of the directory (`/data/sales/sasdata`) using fully qualified SQL names. Use this command to list the tables found in this directory:

```
[centos@datahub ~]$ /opt/sas/viya/home/bin/sas-admin dagentsrv data-sources
tables list --data-source BASE --catalog BASECAT --schema SCH1
```

All the Base SAS data sets located in the directory `/data/sales/sasdata` are now listed, showing that the data is available for access in a Cloud Data Exchange caslib.

The Caslib Data Connector

After creating data services, you can set up a Cloud Data Exchange data connector and [create a caslib](#) as shown in section 2. Here are the tasks involved:

- Configure a [CLOUDEX](#) Data Connector.
- [Add a caslib](#) to SAS Data Agent.
- [Create a CAS connection](#) with SAS Data Explorer.

The tasks for creating a caslib and connection are outlined in Part 1: Cloud Data Exchange [Create a CAS Library](#).

SAS Data Agent Configuration

Authentication Service Cache

If you need to disable Authentication Service caching, or to alter the expiration time of cached items, you can do so in the `da_serv.xml` configuration file. This file is located in the following directory: `/opt/sas/viya/config/etc/dagentsrv/default/`

1. Open `da_serv.xml` for editing.
2. Locate this section: `<Option name="Command"> ALTER SERVER {OPTIONS(XSET CACHE(NAME AS, TIMEOUT 300))} </Option>`.
3. Change the TIMEOUT value from 300 (5 minutes) to a different time 900 (15 minutes): `<Option name="Command"> ALTER SERVER {OPTIONS(XSET CACHE(NAME AS, TIMEOUT 900))} </Option>`.
4. Save the configuration file when you are finished, and restart SAS Data Agent for this change to take effect..
5. Run the following command on the machine where your data agent server is hosted:

```
/etc/init.d/sas-viya-dagentsrv-default restart
```

, where *default* is the name of the data agent server.

Environment Variables

Proxy Settings

If you need to set or change proxy server settings for SAS Data Agent, use the configuration file `sas-dagentsrv` located in the following directory: `/opt/sas/viya/config/etc/sysconfig/dagentsrv/default/`. Add or update the following environment variables with your proxy configurations:

```
export DA_SERVICES_PROXY_SERVER=my.proxy.server.com
export DA_SERVICES_PROXY_PORT=port-number
export DA_SERVICES_PROXY_USER=username
export DA_SERVICES_PROXY_PASS=password
```

When these configurations are set, each value is placed into the `da_entities.dtd` file at start-up of SAS Data Agent. The corresponding entities are:

```
<!ENTITY cfg.services.proxy.server    "">
<!ENTITY cfg.services.proxy.port     "">
<!ENTITY cfg.services.proxy.user     "">
<!ENTITY cfg.services.proxy.pass     "">
```

SAS Environment

These environment variables are registered during deployment but can be overwritten if needed:

```
SASDEPLOYID    - default value "viya"
SASINSTANCE    - default value "default"
```

```

SASROOT           - default value "/opt/sas/${SASDEPLOYID}"
SASHOME           - default value "/opt/sas/${SASDEPLOYID}/home"
SASCONFIG         - default value "/opt/sas/${SASDEPLOYID}/config"
SASLOGROOT        - default value "${SASCONFIG}/var/log"
SASUSER           - default value "sas"
SASGROUP          - default value "sas"
SASLICENSEFILE    - default value "setinit.txt"

```

PostgreSQL Database Server

PGHOST
default value "localhost"

PGPORT
default value "5431"

Data Agent Specific

DA_PROD_NAME
default value "dagentsrv"

DA_INSTANCE
default value "default"

DA_TENANT_ID
Specify the tenant ID. Also used to specify the --customerid for a non-tenancy environment. The default value is "shared".

DA_FSNET_PORT
default value 25141

DA_IOM_PORT
default value 26301

CONSUL_HTTP_ADDR
default value "https://localhost:8501"

CONSUL_TOKEN_FILE
default value "\${TOKENS_PATH}/dagentsrv/\${SASINSTANCE}/client.token"

CONSUL_TOKEN
default value \$(cat "\${CONSUL_TOKEN_FILE}")

VAULT_HTTP_ADDR
default value "https://localhost:8200"

VAULT_TOKEN_FILE
default value "\${TOKENS_PATH}/dagentsrv/\${SASINSTANCE}/vault.token"

DA_SERVICES_HOST
default value ""

DA_SERVICES_PORT
default value "443"

DA_SERVICES_HTTPTRANSPORT
default value "https://"

DA_HOSTNAME
default value "\$(hostname -f)"

DA_HADOOP_JAR_PATH
default value "\${SASHOME}"

DA_HADOOP_CONFIG_PATH

default value "\${SASCONFIG}"

DA_TKJNI_JREOPTIONS

default value "(-Djava.class.path=&cfg.home.install/./SASFoundation/lib
/base/base-tkjni.jar -Dlog4j.configuration=file: &cfg.home.install/./
/SASFoundation/misc/tkjavasas.log4j.properties -Djavax.security.auth.
useSubjectCredsOnly=false -Djava.security.auth.login.config=&cfg.home.install/./
/SASFoundation/misc/tkjava/sas.login.config -Djava.security.policy=&cfg.home
.install/./SASFoundation/misc/tkjava/sas.policy -Djavax.net.ssl.trustStore=
&cfg.home.install/././config/etc/SASSecurityCertificateFramework/cacerts
/trustedcerts.jks -Dsas.ext.config=&cfg.home.install/./SASFoundation/misc
/tkjava/sas.java.ext.config -DPFS_TEMPLATE=&cfg.home.install/./SASFoundation
/misc/tkjava/rpfstpt.xml -Dfile.encoding=UTF-8 -Xms256m -Xmx1024m)"

DA_ADMIN_GRP

default value "Data Agent Administrators"

DA_POWER_GRP

default value "Data Agent Power Users"

DA_PG_SYSCAT_DB

default value "DA_SYSCAT_\${DA_INSTANCE}"

DA_PG_SQLLOG_DB

default value "DA_SQL_LOG_\${DA_INSTANCE}"

DA_TRUSTED_CERTS

default value "\${SASCONFIG}/etc/SASSecurityCertificateFramework/cacerts
/trustedcerts.pem"

DASSLCALISTLOC

default value \${DA_TRUSTED_CERTS}

DASSLREQCERT

default value "DEMAND"

DASSLPVTKEYLOC

default value "\${SASCONFIG}/etc/SASSecurityCertificateFramework/private
/\${DA_PROD_NAME}/default/\${hostname -f}.key"

DASSLCERTLOC

default value "\${SASCONFIG}/etc/SASSecurityCertificateFramework/tls/certs
/\${DA_PROD_NAME}/default/\${hostname -f}.cert"

DA_SITE_NAME

default "private"

DA_CONSUL_TOKEN_FILE

default ".././../././config/etc/SASSecurityCertificateFramework/tokens
/consul/\${SASINSTANCE}/client.token"

DA_VAULT_TOKEN_FILE

default ".././../././config/etc/SASSecurityCertificateFramework/tokens
/dagentsrv/\${DA_INSTANCE}/vault.token"

DA_CLIENT_SECRET_KEY

default dagentsrv/dagentsrv-<tenantid>/login/password

DASSLPVTKEYPASS

default "cat \${SASCONFIG}/etc/SASSecurityCertificateFramework/private/
\${DA_PROD_NAME}/default/dagentsrv_encryption_\${DA_INSTANCE}.key"

Locale Support

Cloud Data Exchange supports the English, United States of America (en_US) locale. The following table outlines the character representations and format used for output. There are no deviations from these formats:

Character Type	Format
Number	<code>dddd.dfffffff</code>
Date	<code>yyyy-mm-dd</code>
Time	<code>hh:mm:ss</code>
Timestamp	<code>yyyy-mm-dd hh:mm:ss [.fffffff]</code>

Note: You should configure database drivers and clients to match this behavior to ensure that conversions are handled correctly.

Removing Data Agent Servers

Suspending Connectivity to SAS Data Preparation

If you no longer require a SAS Data Agent server, you can suspend connectivity to the server or remove it altogether. There are two actions that you can take to prevent connectivity to the SAS Data Preparation machine:

- Stop the data agent to suspend connectivity temporarily.
- Uninstall the data agent to remove the server's connection permanently. When you uninstall a server, the associated certificates are also removed.

Stop (Shut Down) the SAS Data Agent Server

Run the following command on the SAS Data Agent server that you want to stop:

- For Red Hat Enterprise Linux 6.x:


```
sudo service sas-viya-dagentsrv-default stop
```
- For Red Hat Enterprise Linux 7.x and SUSE Linux:


```
sudo systemctl stop sas-viya-dagentsrv-default
```

Uninstall SAS Viya

To completely remove a SAS Data Agent server, follow the uninstall instructions here: [SAS Data Agent 2.3 for Linux: Deployment Guide: Uninstalling SAS Viya](#). The uninstall removes the Data Agent server and all associated software. After the data agent server has been uninstalled, you can update the registration from the SAS Data

Preparation machine. See “[SAS Data Preparation Machine: Unregister SAS Data Agent](#)” for details.

Clean up Residual Items

If no longer needed, remove the **Data Agent Administrators** and **Data Agent Power Users** groups that were created during deployment. These are the default group names suggested at installation.

You can remove the groups through SAS Environment Manager or by using the **identities** CLI to delete the groups.

- In SAS Environment Manager, see "[Users and Groups](#)" to remove the Data Agent Administrators and Power Users groups.
- Using the identities CLI, issue this command to remove groups:

```
/opt/sas/viya/home/bin/sas-admin identities  
delete-group --id "your-Data-Agent-Administrators-group-ID"
```

See “[CLI Examples: Identities](#)” in *SAS Viya Administration: Using the Command-Line Interfaces* for additional information.

Chapter 6

Data Administration

Working with Data Services	52
Overview of Data Services	52
Data Source Security	53
Creating a Data Service	53
Working with DSNs	54
Overview of DSNs	54
Configuring DSNs	54
Catalogs and Schemas	55
Working with Catalogs and Schemas	55
Catalog Name Mapping	55
List Catalogs	56
Command-Line Interface	56
Overview	56
Preliminary Instructions	56
Invoking the CLI	57
Connection Options	58
The Command Collection	58
Command Line	58
Invoking Help	58
Catalogs	59
Description	59
Command Usage	59
List Catalogs	60
Create Catalog	60
Alter Catalog	61
Delete Catalog	61
Data Services	62
Description	62
Command Usage	62
List Data Services	62
Create Data Services	62
Delete Data Services	63
Data Sources	64
Description	64
Command Usage	64
List Data Sources	64
Test Data Source	64

List Data Source Catalogs	65
List Data Source Schemas	65
List Data Source Tables	65
Data Source Names	66
Description	66
Command Usage	66
List DSNs	66
Create Standard DSN	66
Add Schema	67
Delete DSN	67
Profile	68
Description	68
Command Usage	68
Create a New Profile	68
Show Default Profile Contents	68
Set Default URL	68
Set Default Output	68
Schemas	69
Description	69
Command Usage	69
List Schemas	69
Create Schemas	69
Security (Domains)	70
Description	70
Command Usage	70
Create Domain	70
Delete Domain	71
List Domains	71
Show Domains	71
Security (Credentials)	72
Description	72
Command Usage	72
Create New Credential	72
Update Credentials: Set Password	72
Delete Credentials	73
Clean Vault	73
CLI Output	73
Output Formatting Options	73
Examples	74
CLI Examples	75

Working with Data Services

Overview of Data Services

To access data, the administrator must create and configure a data service. Data services contain connection information and driver specifics to connect with data sources such as Oracle or Base SAS data sets.

Data services contain information that identifies the location of tables residing in your data source. If a data source does not support native catalogs, SAS Data Agent enables you to define a logical catalog name to use as an SQL identifier. This allows unique identification of each data source when performing heterogeneous operations.

Data services that require logins must be associated with a domain. When users connect to the data source through a data source name (DSN), the domain name is used to retrieve user credentials associated with that data service. The credentials are then passed along to the back-end database.

Data services can also contain optional information to control driver behavior, such as locking semantics and tracing. Data services form the foundation for connectivity to a data source, and you can assign privileges that control user access to the data. However, relational databases provide authorization that limits the operations that can be performed on the data. SAS Data Agent does not override authorizations that are defined on a third-party database. Authorizations and permissions defined on a third-party database overrule permissions and privileges that are set on SAS Data Agent.

Data Source Security

Data source security, such as authorization, cannot be bypassed by a SAS Data Agent data service. If an Oracle administrator denies privileges to a user on table T1, then that user will always be denied access to table T1, no matter what privileges are set in SAS Data Agent.

Creating a Data Service

Overview

Use the SAS Data Agent Command Line Interface (CLI) to create or delete a data service. A new DSN is automatically generated each time a new data service is created. This is a standard DSN given the same name as the data service. However, if the DSN name already exists on the server, it will not be created. If the data service is renamed, the DSN name remains unchanged.

By default, a BASE data service is created the first time that Cloud Data Exchange is started. Only one BASE data service can exist in a SAS Data Agent installation, and it cannot be modified or deleted.

Example: Create a Data Service

Use the [Data Services command set](#) in the CLI to create a data service. Here is an example:

```
sas-admin dagentsrv data-services create oracle --name MyOracle [connection options]
```

To see the connection options that are available for Oracle, type `--help` after 'Oracle' and a list of options is displayed. You can also reference Appendix 1 to see a list of connection options that are sorted by option name. You can create a data service for these data types:

db2	Create a DB2 data service definition
federation-server,	Create a data service definition to communicate with
fedsrv	Federation Server
hive	Create an Apache Hive data service definition
odbc	Create an ODBC data service definition
oracle	Create an ORACLE data service definition

postgres	Create a PostgreSQL data service definition
redshift	Create a RedShift data service definition
saphana, hana	Create a SAP HANA data service definition
sqlserver, sqlsvr	Create SQL Server (via ODBC) data service definition
teradata	Create a Teradata data service definition

See the following topics for additional information that pertains to creating a data service.

Native Catalog Support

When creating a data service for a data source that supports native catalogs, and using the `--register-all` option, the server attempts to connect to the database to acquire a list of catalogs. Credentials are required to secure the connection. If the connection cannot be made, creation of the data service fails. The same requirement for pre-registered credentials applies when creating a data service to ODBC with native catalog support, or for any data services that support native catalogs.

Identifier Case Sensitivity

When creating an ODBC data service, the server must query the data source to acquire its identifier case sensitivity property. The identifier case sensitivity property is used to create security entries in the server's system tables and is stored with the data service.

Database Login Prerequisite

Due to the requirement for a database connection described above, the following database login prerequisite applies to data service types SQLSERVER, ODBC and ODBC_FED. These actions must be completed before creating the data service.

1. A database login must be registered in the domain that will be associated with the new data service.
2. The domain must be registered.

Working with DSNs

Overview of DSNs

A data source name (DSN) is a resource that provides connection information for data sources that are accessed through SAS Data Agent. A DSN references a specific data source to which it connects and defines the connection information so that a user can access data. A DSN must be associated with a data service. DSNs differ from data services in that they specify how a user accesses data. A data service defines the location of the data (for example, specifying a server where the data source resides).

Configuring DSNs

Example: Create a DSN

Use the [Create Data Source Name](#) command set to create a DSN. This is an example of the CLI command for a DSN:

```
sas-admin dagentsrv dsns create oracle --name my-DSN
--data-service Oracle-data-service[connection options]
```

For a complete list of options associated with DSNs, see the *Data Source Names* command set in the CLI.

Catalogs and Schemas

Working with Catalogs and Schemas

A catalog is a named collection of logically related schemas. The catalog is the first-level (top) grouping mechanism in a data organization hierarchy that qualifies schemas. SAS Data Agent requires that catalogs are registered in its system tables. Catalog and schema information typically comes from the data source, but there are exceptions:

- The BASE data service requires that its catalogs and schemas are registered manually in the system tables. See [Create Catalogs](#) and [Create Schemas](#) for syntax and examples. Each catalog must have at least one associated schema; otherwise, the data service is not configured properly and you will be unable to connect to the BASE data service.
- Some relational databases do not support the concept of a catalog (i.e., no “native” catalog). For these data sources, a logical catalog name is specified as part of the data service definition. This logical catalog is then registered in the system tables. The logical catalog must be unique among all data services.

Specify the `--register` option when creating a data service for data sources that support native catalogs. This automatically registers all catalogs in the system tables. If a catalog name already exists under a different data service, it will not be registered. Here, you would manually create a catalog with a unique name under the data service, and use the `--native-catalog` option to assign its native catalog name as the name from the data source. See [Create Data Services](#) and [Create Catalogs](#) for syntax and examples.

Catalog Name Mapping

All catalog names must be unique within SAS Data Agent server. If your database supports native catalogs, you can use catalog mapping to avoid duplication errors. The connection will default to exposing native catalog names unless you specify a logical name for the catalog and map it to the native catalog name in the database. Use the `catalogs create` command to provide a different logical catalog name to a native catalog name in the database. Consider this scenario:

1. Create a data service for SQL Server and register all catalogs.
2. Create a second data service for another SQL Server database. Some catalogs may overlap between the 2 SQL Server databases, so you cannot register all catalogs. Use the `--native-catalogs` option to register only the catalogs that do not conflict.
3. Use the `catalogs create` command to create catalog mappings for catalogs that conflict with existing catalogs, for example,

```
sas-admin dagentsrv catalogs create sqlserver --data-service data_service_name
--name logical_catalog_name --native-catalog native_catalog_name [command options]
```

List Catalogs

Use the catalogs list command and define options to filter your search. For example, if you specify data source, only catalogs associated with that data source are listed. If you do not specify `--data source`, then all catalogs from all available data sources are listed. The command is:

```
sas-admin dagentsrv data-sources catalogs list [--data-source dataSourceName]
  [filters] [sorting]
```

You can filter the name or data source with other options, such as `--data source contains`. You can sort based on name or data source. You can also use `--start` and `--limit` to control the starting place and number of rows in the result set.

Command-Line Interface

Overview

The command-line interface (CLI) for interacting with SAS Data Agent servers is `dagentsrv`. The CLI is an executable that operates on Windows 64-bit systems, Linux, and macOS systems. The tool is written for administrators and provides a shell- or console-based interface for communicating with a SAS Data Agent driver.

The `dagentsrv` uses a REST infrastructure to communicate with SAS Data Agent. It supports commands to create, list, and delete data services. It also provides the capability to view definitions for data sources, catalogs, and schemas.

All the information needed to process user requests are expressed through command-line options or stored in the user's SAS Viya profile. You can append `--help` to any command to see usage information for the command (or partial command) you have entered.

Before using the CLI, you must accomplish these tasks:

1. Set environment variables.
2. Set the base URL.
3. Specify a default server (optional).
4. Obtain login credentials.

Preliminary Instructions

Set the `SSL_CERT_FILE` Environment Variable

Before using the CLI, you must set the `SSL_CERT_FILE` environment variable. See ["Getting Started"](#) for additional information to set this environment variable.

Set the Base URL

Nearly all commands in `dagentsrv` require support from a REST interface. Therefore, the CLI must be given the base URL that describes where the endpoints are located. This is typically a server name and port, expressed as a URL. You can specify the base URL by using one of these methods:

- **Explicitly on each command line:** Specify the endpoint used as the base URL with a global option `--sas-endpoint` placed on the command line immediately after the `sas-admin` program name. This is expressed as an HTTPS URL, such as `https://hr-servers.mega.com:443`.
- **As an environment variable:** Specify the endpoint as an environment variable, `SAS_SERVICES_ENDPOINT`. If you do not specify a command-line global value, `dagentsrv` looks for this environment variable to determine the base URL. Here is an example: `export SAS_SERVICES_ENDPOINT="https://hr-servers.mega.com:443"`
- **In your SAS profile:** You can persist the definition of the default endpoint in your profile, which is stored in the `.sas subdirectory...` of your home directory. Here is the command:

```
sas-admin profile set-endpoint URL-specification
```

Specify a Default Server

A majority of commands in `dagentsrv` require that you specify the name of the Data Agent server where the commands will operate. You can specify this with the `--data-agent` option in any command that requires a designated server name. You can also set a default server name that is used whenever a command requires a server specification and the `--data-agent` option is not specified. Use this command to set a server name in the default profile:

```
sas-admin dagentsrv servers set-default --data-agent server-name
```

If a server is not specified on the command line or in the profile, the CLI queries the SAS Viya infrastructure to determine whether more than one data agent server is defined. If only one server has been registered, and cannot be assumed, it is used as the default server. If there are multiple servers, an error is reported indicating that a server name must be specified.

Login Credentials

SAS Viya

All commands that work with the SAS Viya REST interfaces use OAUTH2 login credentials to create a token which provides authentication and authorization for each service. This token must be acquired by logging in to SAS Viya before executing additional commands. Each additional command reuses the OAUTH2 token received. When the token expires, you must acquire a new token. Use this command to acquire a new token:

```
sas-admin auth login [--username user-name --password password]
```

If the optional `username` and `password` values are not supplied on the command line, you will be prompted for them. Once you have successfully authenticated, the token is stored in the SAS Viya default profile and is reused for each subsequent command.

Invoking the CLI

Before invoking the CLI, you might want to update your path environment variable to reflect the location of your command-line tools. Invoke the CLI using the `sas-admin` command, found at `/opt/sas/viya/home/bin/` and specify `dagentsrv` as the plug-in.

Connection Options

You can specify connection options for commands that create a data service, catalog, schema, or DSN. These are described in the commands that use **[options]**. These are CLI options used to construct elements of a connection string. For example, to specify a user name, password, and PATH for an Oracle data service, you might use the following:

```
sas-admin dagentsrv services create oracle --name myora
      --database-user myID
      --database-pwd myPass
      --path myoraPath
```

The CLI option names do not always match the associated connection string option name. To conform to CLI requirements and avoid ambiguous names, see Appendix 1 – [Connection Options](#) for a list of available connection options.

You can also use the **--help** option with any create command to see the list of options that are available. The available options depend on the associated data service type.

Note: In this command (as with all examples that follow) the command text is shown wrapped for ease of reading. However, the options must be either specified on a single command line or your shell's continuation character must be used.

The Command Collection

Command Line

Here is a basic command line with the necessary parameters:

```
sas-admin [global-options] dagentsrv <collection-type> <command>
      [--command options]
```

Where **[global-options]** represent commands that impact operation of the CLI, such as **output**, **profile**, and **help**. The **<collection-type>** parameter describes the class of objects that you want to work with, such as **servers**, **data-sources**, **catalogs**, **schemas**, or **tables**. A command is preceded with two hyphens if it requires additional information, for example, **--catalog DBCAT1**.

TIP When interacting with the CLIs, keep in mind that the commands are case sensitive and should always be lowercase.

Note: Use double quotation marks to group strings with special characters or whitespace characters. Within the double quotations, use single quotation marks for values with whitespace characters.

Invoking Help

Append the global option **--help** to any command that you have entered to see a list of valid arguments for that command. Where you invoke help determines the information that is returned. For example, using **data-services create** to create a service type, enter **--help** after the **create** command, to see a list of valid data source types:

```
sas-admin dagentsrv data-services create --help
```

The CLI returns the following information:

```
USAGE: sas-admin dagentsrv data-services create command [command options]
COMMANDS:
  db2unxpc, db2           Create a data service definition for DB2 (Unix and PC)
  federation-server, fedsvr  Create a data service to communicate with Fed Server
  help, h                 Shows a list of commands or help for one command.
  hive                     Create an Apache Hive data service definition
  odbc                     Create an ODBC data service definition
  oracle                   Create an ORACLE data service definition
  postgres                 Create a POSTGRESQL data service definition
  redshift                  Create a RedShift data service definition
  saphana, hana            Create a SAP HANA data service definition
  sqlserver, sqlsvr        Create SQL Server (via ODBC) data service definition
  teradata                  Create a Teradata data service definition
```

If you invoke help after the **create type** command, options are displayed for the specific data service type that you are creating. For example, you are creating a data service for Oracle, but are unsure what connection options are available. Help is invoked on the command line after your data source type, Oracle:

```
sas-admin dagentsrv data-services create oracle --help
```

The CLI returns the following:

```
USAGE: sas-admin dagentsrv data-services create oracle [command options]
OPTIONS:
  --case-sensitive-columns  If present, column names case sensitive
  --case-sensitive-objects  If present, object names case sensitive
  --catalog                  Specify the name of the logical catalog for
                             this data service
  --ct-preserve              Preserve result set column types, can be STRICT,
                             SAFE, FORCE, or FORCE_COL_SIZE
  --data-agent, -d          Specify the name of the Data Agent
  --database-password        Specify the password used to connect to the database
  --database-user            Specify the user ID used to connect to the database
```

Note: This is not a complete option list. Invoke help from the CLI to see all of the command options that are available for Oracle.

Catalogs

Description

The **catalogs** command collection manages catalogs registered with SAS Data Agent.

Command Usage

```
sas-admin dagentsrv catalogs list|create|alter|delete [command options]
```

List Catalogs

Command

```
sas-admin dagentsrv catalogs list
```

Description

The **catalogs list** command displays the catalog names and the data service associated with the catalog. This might not include all catalogs available to the underlying data services. An optional native catalog, which is used internally by the data service, might be specified.

Table 6.1 Catalog Output Fields

Field	Description
dataServiceName	The name of the data service, such as “BASE” or “ADMIN”.
name	The name of the catalog.
nativeCatalogName	The native catalog name used by the service. This can be empty.
options	Additional connection options that are used with this catalog, if any. If there are no options, this field is empty.

Create Catalog

Command

```
sas-admin dagentsrv catalogs create base|sqlserver --name catalog [command options]
```

Description

This command creates a catalog registration for driver type BASE or SQLSERVER. The driver type must be specified after the CREATE command to indicate which driver class the catalog is associated with. You must also specify the name of the catalog with the **--name** option, and the associated data service with the **--data-service** option. You can specify the native catalog name or connection options with additional command options.

Here are the options for **create catalogs**:

```
base          Create a logical catalog for the BASE data service
help, h      Shows a list of commands or help for one command.
sqlserver    Create a logical catalog that maps to a SQLServer native catalog
```

Create a logical catalog for BASE

```
sas-admin dagentsrv catalogs create base [command options]
```

Here are the options for **catalogs create base**:


```

--data-agent, -d    Specify the name of the Data Agent
--driver "base"    Database driver to use (defaults to associated data service type)
--name             Specify the name of the catalog to create
--schema          Specify the name of the associated schema

```

Note: You are not required to specify a data service for Base SAS because there is only one data service for this driver.

Create a logical catalog for SQLserver

```
sas-dagentsrv catalogs create sqlserver [command options]
```

Here are the options for **catalogs create sqlserver**:

```

--data-agent, -d    Specify the name of the Data Agent
--data-service      Specify the name of the data service under which this
                   catalog is created
--name             Specify the name of the catalog to create
--native-catalog    Specify the native catalog linked to this logical catalog

```

Alter Catalog

Command

```
sas-admin dagentsrv catalogs alter --name catalog-name [command options]
```

Description

Use the **catalogs alter** command in cases where catalog registration generates catalog entries in SAS Data Agent that need to be amended to support catalog mapping. You can rename the registered catalog and also add the native catalog name in a single operation when you use **alter**. The options are:

```

--native-catalog    Specify the optional native catalog name
                   associated with this catalog
--rename-to         Specify the optional new name of the catalog

```

Delete Catalog

Command

```
sas-admin dagentsrv catalogs delete --name catalog-name [--force] [--cascade]
```

Description

Use **catalogs delete** to delete a catalog registration. Use the **--name** option to specify the name of the catalog to delete. Here are additional options for deleting catalogs:

--force

Use **--force** to force the command to complete successfully, meaning that an error is not reported if the named catalog is not found or cannot be deleted.

--cascade

Use the **--cascade** option to delete all logical schema definitions when the catalog is deleted.

Data Services

Description

The **data-services** command collection operates on data services residing on SAS Data Agent servers. You can shorten the data-services command to **services**. This collection uses the *dataAgentMgmt REST API*.

Command Usage

```
sas-admin dagentsrv data-services list|create|delete [command options]
```

List Data Services

Command

```
sas-admin dagentsrv data-services list
```

Description

The **data-services list** command displays data service names and their type.

Table 6.2 Data Services Output Fields

Field	Description
domain	The named security domain for the service to use.
name	The name of the data service.
options	Connection options to be used by this data service.
type	The type of the data service, such as DB2 (UNIX and PC), BASE, or Oracle.

Create Data Services

Command

```
sas-admin dagentsrv data-services create type --name service-name [connection options]
```

Description

Use **data-services create** to create a new data service definition. The create command must be followed by the **type** of data service that you are creating. Here is a list of the supported types for data services:

```
db2unxpc, db2 Create a data service definition for DB2 (Unix and PC)
```

fedsvr	Create a data service definition for Federation Server
hive	Create an Apache Hive data service definition
odbc	Create an ODBC data service definition
oracle	Create an ORACLE data service definition
postgres	Create a POSTGRES data service definition
redshift	Create a RedShift data service definition
saphana, hana	Create a SAP HANA data service definition
sqlserver, sqlsvr	Create SQL Server (via ODBC) data service definition
teradata	Create a Teradata data service definition

BASE is also a valid data service type, but you cannot create a new BASE data service in SAS Data Agent. A default BASE service is automatically created at server start-up.

The data service must be given a name with the **--name** option, followed by any required connection options that the specific data service type requires. For example, for an Oracle type you must specify a catalog name, which is automatically created for the Oracle service.

Register Catalogs

As an option, you can register native catalog names for ODBC or SQL Server. Specify registration for a list of catalogs with **--native-catalogs**, or use **register-all** to register all catalogs that are visible to the connection. If both commands are used, **register-all** takes precedence. Here is an example:

```
sas-admin dagentsrv data-services create sqlsvr --name MySQLSvr
--native-catalogs "cat1,cat2"
sas-admin dagentsrv data-services create sqlsvr ... --register-all
```

Delete Data Services

Command

```
sas-admin dagentsrv data-services delete --name service-name [--cascade] [--force]
```

Description

The **delete data-services** command deletes an existing data service definition. There is no text or JSON output for this form of the command. Here are additional options for deleting a data service:

--cascade

Direct SAS Data Agent to also delete any dependent data source names, catalogs, or schemas that reference the service being deleted. If you do not specify this and there are any dependent objects for the service, the command returns an error.

--force

Forces the command to complete successfully, meaning that an error is not reported if the named catalog is not found or cannot be deleted.

Data Sources

Description

The **data-sources** command collection operates on data sources that reside on SAS Data Agent servers. You can abbreviate the **data-sources** command as **ds**. Not to be confused with **data source names**, this command collection describes either a data service or a data source name and is for information purposes only. To create a DSN, use the [data source names](#) command collection.

Command Usage

```
sas-admin dagentsrv data-sources list|test|catalogs list|schemas list|
tables list [command options]
```

List Data Sources

Command

```
sas-admin dagentsrv data-sources list [--dsn] [--data-service]
```

Description

The **data-sources list** command lists available data sources. For each data source, the output contains the name of the data source, a unique ID, and an indicator to show whether it is a data source name or a service name. You can use the **--dsn** and **--data-service** options to limit the output to only data source names or data services.

The output is a table of data sources and an indication of whether the data source references a data service or a data source name (DSN). If you specify the **--output fulljson** option, an additional field that contains the unique data source ID is included. This is needed because a data source name can occur twice; once for a data service, and again for a DSN.

Table 6.3 Data Sources Output Fields

Field	Description
id	The unique name of the data source used internally.
isDataService	Boolean indicating if this is a data service (as opposed to a DSN).
Name	The name of the data service.

Test Data Source

Command

```
sas-admin dagentsrv data-sources test --name dsName
```

Description

Use the **data-sources test** command to test a connection to an underlying data provider. The name of the data source, which can be either a data service or a data source name, must be given. The command-line tool establishes a connection to the data provider (such as an Oracle or PostgreSQL database) that supports the data source and reports any errors in the connection. Administrators can use this to verify that the specification of the data source and handling of credentials is correct.

List Data Source Catalogs**Command**

```
sas-admin dagentsrv data-sources catalogs list --data-source dsName
```

Description

The **data-sources catalogs list** command lists catalogs associated with a specific data source. This can include catalogs that have been registered with a data agent, as well as catalogs known only to the data source itself. This is not the same as the **list** command of the catalogs command collection, which reports only catalogs registered to SAS Data Agent.

List Data Source Schemas**Command**

```
sas-admin dagentsrv data-sources schemas list --data-source dsName
--catalog catName
```

Description

The **data-sources list** command shows schemas that are associated with a specific data source, by requesting that the underlying data provider list its schemas. This is not the same as the list command of the schemas collection, which reports only schemas registered to SAS Data Agent. You must specify the catalog name in the data provider that qualifies the schemas to be listed.

List Data Source Tables**Command**

```
sas-admin dagentsrv data-sources tables list --data-source dsName --catalog catName
--schema schName
```

Description

This command lists tables for a given data source. You must specify the **--data source**, **--catalog**, and **--schema** options to identify the location to search for available tables.

Data Source Names

Description

The **data-source-names** command collection operates on DSNs registered on a SAS Data Agent server. You can abbreviate the **data-source-names** command as **dsns**. The abbreviated form is used in the commands that follow.

Command Usage

```
sas-admin dagentsrv data-source-names|dsns add-schema|create|list|delete
[command options]
```

List DSNs

Command

```
sas-admin dagentsrv data-source-names list
```

Description

This command lists available data source names. The output is a table of data source names (DSNs) that include the name and type of data service that the DSNs are configured to use.

Table 6.4 Data Source Names List

Field	Description
description	Optional text description of the DSN
name	The name of the DSN
owner	The name of the owner of the DSN
ownerType	Whether the owner describes a “role” or a “group”
service	The name of the data-service used by this DSN
type	Type of data service that the DSN is associated with (for example, DB2 [UNIX and PC], Oracle).

Create Standard DSN

Command

```
sas-admin dagentsrv dsns create type --name dsnName --data-service svcName
[connection options]
```

Description

This command, with an alias of **dsns**, creates a new data source name. The command includes the data service type that the data source name is associated with, such as Base SAS, DB2 (UNIX and PC), or PostgreSQL. A DSN must be created under an existing data service, specified with the **--data-service** option. This is followed by additional connection options specific to the DSN being created. Here are the command options for **dsns**:

```
--catalog           Specify the catalog to be used with this data source name
--data-service, --ds  Specify the name of the associated data service
--description       Specify a quoted string documenting this data source name
--name              Specify the name of the data source name to create
```

The data service types associated with creating a standard DSN are:

```
base                Create a data source name for the BASE data service
db2unxpc, db2       Create a data source name for a DB2 (Unix and PC) data service
fedsvr              Create a data source name for a Federation Server data service
hive                Create a data source name for an Apache HIVE data service
odbc                Create a data source name for an ODBC data service
oracle              Create a data source name for an Oracle data service
postgres            Create a data source name for a PostgreSQL data service
redshift            Create a data source name for a RedShift data service
saphana, hana       Create a data source name for a SAP HANA data service
sqlserver           Create a data source name for a SQL Server data service
teradata            Create a data source name for a Teradata data service
```

Add Schema**Command**

```
sas-admin dagetnsrv dsns add-schema --name schema-name [command options]
```

Description

The **dsns add-schema** command is for DSNs associated with a BASE data service. A DSN for a BASE data service can have multiple schemas, each representing different file system locations containing Base SAS data sets. Because you can add only one schema when you originally create a DSN, you can use the add-schema command to add additional schemas to the DSN. This command accepts the same options used to [create a schema](#).

Delete DSN**Command**

```
sas-admin dagentsrv dsns delete --name dsnName [--force]
```

Description

The **dsns delete** command is used to delete an existing DSN. You must specify the name of the DSN to be deleted. An error is returned if the DSN does not exist or if you do not have permission to delete the resource. Specify the **--force** option if you want the command to complete without error even if the DSN is not found or cannot be deleted.

Profile

Description

Use the **profile** command to view or modify persistent SAS profile data. This is the same profile used by all SAS command line interfaces in a SAS Viya deployment. You can see the full set by using the **--help** option after the profile command.

Command Usage

```
sas-admin profile init|set-endpoint|set-output [command options]
```

Create a New Profile

Command

```
sas-admin profile init
```

Description

This command creates a new empty profile if one does not already exist. This should be done only once with the first SAS command-line interface tool that you use. Subsequent command-line tool invocations might use the same default profile.

Show Default Profile Contents

Command

```
sas-admin profile show
```

Description

This command displays the contents of the default profile.

Set Default URL

Command

```
sas-admin profile set-endpoint url-specification
```

Description

This command enables you to specify the URL of the SAS Viya cloud deployment to be used on all dagentsrv CLI commands. Therefore, you do not need to specify the **--sas-endpoint** global option on each command.

Set Default Output

Command

```
sas-admin profile set-output text | json | fulljson
```


Description

Use **set-output** to set the default output format of CLI commands. The formats are:

text

Text is human-readable text, usually in columnar format if multiple data items are presented.

json | fulljson

The formats json and fulljson result in command output expressed as a JSON object. The fulljson command displays the output using a documented REST API media type when possible. The json output typically contains only the items that were explicitly requested by the command.

Schemas

Description

The **schemas** command collection manages schema information registered on the SAS Data Agent server. This operation applies only to schemas contained within catalogs for the BASE data service.

Command Usage

```
sas-admin dagentsrv schemas list|create|delete [command options]
```

List Schemas**Command**

```
sas-admin dagentsrv schemas list
```

Description

This command lists schemas that are registered with SAS Data Agent. This might not be a list of all available schemas defined in the underlying data provider.

Create Schemas**Command**

```
sas-admin dagentsrv schemas create base --name schemaName
--catalog catalogName [command options]
```

Note: You can also specify a full path to represent the schema name. Doing so identifies the existing catalog with the new schema. Here is an example:

```
sas-admin dagentsrv schemas create base --name BCAT1.MYSCHEMA
```

. This is the same as specifying **--name MYSCHEMA --catalog BCAT1**. In either case, the catalog name must already exist and be associated with the BASE data service. If you do not use a full path, then you must specify the **--catalog** option.

Description

The **base** ‘type’ implies that the schema is used with a BASE data service, of which there can be only one. However, you must specify the parent catalog, which must also be associated with the BASE service.

```
base          Create a new BASE schema
```

Security (Domains)

Description

Use the **security domains** command collection to create and delete a domain. You can also show and list domains. To create security credentials within the domain, use the [Security Credentials](#) command collection.

Important: The security domains commands must be run on the same machine as the SAS Data Agent deployment. These commands interact directly with SAS Data Agent without going through SAS Viya to manage credentials that, by design, are safeguarded in SAS Secrets Manager, on-premises, and are never sent to the cloud. Therefore, as a security measure, the Data Agent administrator must have a unique login and associated home directory, on the machine where SAS Data Agent is deployed. Account uniqueness is required because the authentication information used to log on with the CLI is stored in the account’s home directory on the deployment machine. Using a shared account is not recommended because other users could use the account to run administrative CLI commands, bypassing the required authentication.

If you have more than one SAS Data Agent registered, you must identify which data agent to use with the **--data-agent** option. The CLI needs information from SAS Viya to access information about the server to complete this command.

Command Usage

```
sas-admin dagentsrv security domains create|delete|list|show [command options]
```

Create Domain**Command Usage**

```
sas-admin dagentsrv security domains create [command options]
```

Description

Use **domains create** to create a domain. These commands must be run after authenticating to SAS Viya using an identity with SAS Administrator privileges. The CLI needs information from the registration of the server to complete this command.

The command options are as follows:

```
--domain      Create a new security domain for SAS Data Agent
--force       Create a new domain even if it overwrites an existing domain
--help, h     Shows a list of commands or help for one command.
```

Delete Domain

Command Usage

```
sas-admin dagentsrv security domains delete [command options]
```

Description

Use **domains d** command to delete a domain. These commands must be executed after authenticating to SAS Viya using an identity with SAS Administrator privileges. The CLI needs information from SAS Viya about the data Agent server to complete this command.

The command options are as follows:

```
--cascade      If specified, delete any credentials in the domain as well.
--domain       Specify the id of the domain to delete
```

List Domains

Command Usage

```
sas-admin dagentsrv security domains list
```

Description

This command lists domains that are registered with SAS Data Agent.

Show Domains

Command Usage

```
sas-admin dagentsrv security domains show --domain domain-name
```

Description

Use the **show** command to view information related to a specific domain name. If you have more than one SAS Data Agent registered, you must use the **--data-agent** option to identify which data agent to use.

In this example, the specified domain name, DOM1, returns the following information:

```
$ sas-admin dagentsrv security domains show --domain DOM1
Item          Value
Version       1
ID            DOM1
Description    PostgreSQL domain
Type          shadow
Created       2018-07-23T17:20:30.898Z
Modified      2018-07-23T17:22:12.096Z
Created By    dagentsrv-shared
Modified By   dagentsrv-shared
```

Security (Credentials)

Description

The security credentials command collection is used to create local credentials for SAS Data Agent associated by key with sans-password credentials created in the SAS Viya infrastructure. SAS Viya services are still used for authentication. However, when credentials are issued for connection to the customer databases local to the data agent, the credentials are extracted by key from the SAS Secrets Manager (vault) since they are not persisted in the SAS Viya Cloud.

Important: When using the `security credentials` command set, the Data Agent CLI must be run on the same machine that SAS Data Agent runs on. The CLI requires direct connection to the data agent server to ensure that the credentials are never exposed on the internet.

After an administrator creates a domain, individual users are allowed to specify their own credentials, thereby keeping them secret from all other users, including the SAS Data Agent administrator. These users must have their own account on the SAS Data Agent deployment machine.

Command Usage

```
sas-admin dagentsrv security credentials create|set-password|delete|clean-vault
[command options]
```

Create New Credential

Command

```
sas-admin dagentsrv security credentials create [command options]
```

Description

Use this command to create a local credential linked to the SAS Viya credentials service. If the password is not specified on the command line, the CLI prompts you for the password value. The command options are as follows:

<code>--domain</code>	Specify the domain for the credential
<code>--identity</code>	Specify the identity ID for the credential
<code>--identity-type</code>	Specify the identity type - user or group. User is the default.
<code>--password</code>	Specify the password for the credential
<code>--username</code>	Specify the username for the credential
<code>--force</code>	Specify that existing credentials be overwritten

Update Credentials: Set Password

Command

```
sas-admin dagentsrv security credentials set-password [command options]
```

Description

The `set-password` command enables you to modify the password for an existing credential. The credential must be in the domain, and must be associated with the identity of the user currently logged on. You must know the current password to be able to change it. The command options are as follows:

```
--data-agent, -d    Specify the name of the server
--domain            Specify the domain for the credential
--password         Specify the new password for the credential
```

Delete Credentials**Command**

```
sas-admin dagentsrv security credentials delete [command options]
```

Description

You can delete a credential for the identity of the user who is currently logged on. If you are an administrator, you can delete the credentials for another user by specifying the `--identity` option. The command options are as follows:

```
--domain    Specify the domain for the credential
--identity  Allows you to delete another users credential
            if you have administrator privileges.
            If you don't specify an identity, the CLI assumes
            the identity you're logged on as.
```

Clean Vault**Command**

```
sas-admin dagentsrv security credentials clean-vault --domain domain-name
```

Description

This command is used to remove SAS Data Agent database passwords from the SAS Secrets Manager. Use this step when removing a SAS Data Agent deployment.

CLI Output**Output Formatting Options**

You can specify a desired output format on any command using the `--output` option as a global option after `sas-admin`. You can choose any of these outputs: `text`, `json`, or `fulljson`. The `fulljson` option is the same as the `json` option, but `fulljson` has a more complete description of the underlying data structures, which correspond to the published REST media types. You can also add any of these formatting options to control the nature of the output for any commands that display tables of output, such as lists of servers, services, catalogs, and schemas:

Table 6.5 Output Formatting Options

Option	Description
<code>--limit number-of-rows</code>	Limits the number of rows returned. If not specified, the default is 30 rows.
<code>--start first-row-number</code>	Specifies the starting row of the output. If not specified, the output starts on the first row, which is the default.
<code>--sort-by [~]column-name[, [~]column-name...]</code>	Specify one or more columns on which to sort the output. If not specified, the data is not sorted. You can specify a tilde (~) character before the column name to sort in descending order instead of the default ascending order. Separate the column names with a comma if specifying more than one column name.
<code>--name "string"</code>	Filter the output by restricting which rows are processed by using the <code>--name</code> option that displays only rows that match the specific named object.

Examples

Here is an example command requesting a list of available SAS Data Agent servers in a JSON array object:

```
sas-admin dagentsrv --output json servers list
```

You can also set a default output format in the profile that is used for any command that does not explicitly specify an output format. See the **profile** command for more information.

The command shown above produces a list of available SAS Data Agent servers in a JSON array object. Here is an example of the output:

```
{
  "items": [
    { "host": "172.17.0.53",
      "name": "HRserver",
      "port": 26301 },
    {
      "host": "172.17.0.58",
      "name": "PayrollServer",
      "port": 26301 } ] }
```

This is the output produced when there are two available Data Agent servers named HRserver and PayrollServer.

Note that the array of objects contains three fields. These fields correspond to the columns of the equivalent text output format, as in the following text output:

Name	Host	Port
HRserver	172.17.0.53	26301
PayrollServer	172.17.0.58	26301

As a standard, output that produces a table in text output mode produces a JSON array, with each element of the array being a nested structure that contains the columns of data. The field name is like the column names in the text output, but it can vary due to the requirements of being a valid JSON tag name.

CLI Examples

The following examples assume that you have already signed in to SAS Viya at the command line. See [“Command-Line Interface: Preliminary Instructions” in SAS Viya Administration: Using the Command-Line Interfaces.](#)

Example: List all of the catalogs that are registered with SAS Data Agent and connected to a data source.

```
sas-admin dagentsrv catalogs list
```

Example: Display a table of service names and their service type.

```
sas-admin dagentsrv data-services list
```

Example: List the schemas that are registered with the SAS Data Agent.

```
sas-admin dagentsrv schemas list
```

Example: List the domains that are registered with the SAS Data Agent.

```
sas-admin dagentsrv security domains list
```

Example: Create a data service for a connection to an Oracle database with the SAS Data Agent and verify the connection.

- 1 `sas-admin dagentsrv security domains create --domain ORACLE_ODBC_DOMAIN`
- 2 `sas-admin dagentsrv security credentials create --domain ORACLE_ODBC_DOMAIN --identity username --username database-username --password database-password`
- 3 `sas-admin dagentsrv data-services create oracle --name ORACLE_ODBC_SERVICE --domain ORACLE_ODBC_DOMAIN --driver odbc --dsn tktsora`
- 4 `sas-admin dagentsrv data-sources test --name ORACLE_ODBC_SERVICE`
- 5 `sas-admin dagentsrv data-sources schemas list --data-source ORACLE_ODBC_SERVICE --catalog ORACLE_ODBC_SERVICE`
- 6 `sas-admin dagentsrv data-sources tables list --schema MODEL --data-source ORACLE_ODBC_SERVICE --catalog ORACLE_ODBC_SERVICE`
- 7 `sas-admin dagentsrv security domains delete --domain ORACLE_ODBC_DOMAIN --cascade`

```
8 sas-admin dagentsrv data-services delete --name ORACLE_ODBC_SERVICE --cascade
```

- 1 Create a domain for the connection.
- 2 Create the credentials for the connection.
- 3 Create a data service definition for an Oracle data base. The name of the service is `ORACLE_ODBC_SERVICE`. Since the `catalog` option is not specified, a catalog is created with the same name as the data service.
- 4 Test the connection to the Oracle database.
- 5 List the schemas that are registered with SAS Data Agent in the specified catalog and associated data service.
- 6 List the tables for the data service.
- 7 Delete the domain that you created. Specify the `cascade` option to delete any credentials in the domain.
- 8 Delete the data service definition that you created. Specify the `cascade` option to delete any dependent data source names, catalogs or schemas.

Example: Create a data service for a connection to a BASE database with the SAS Data Agent and verify the connection.

```
1 sas-admin dagentsrv catalogs create base --name BASE_CAT
```

```
2 sas-admin dagentsrv schemas create base --name BASE_SCHEMA --driver BASE
--catalog BASE_CAT --primary-path /location-of-tables
```

```
3 sas-admin dagentsrv data-sources test --name BASE
```

```
4 sas-admin dagentsrv data-sources catalogs list --data-source BASE
```

```
5 sas-admin dagentsrv data-sources schemas list --catalog BASE_CAT
--data-source BASE
```

```
6 sas-admin dagentsrv data-sources tables list --schema BASE_SCHEMA
--catalog BASE_CAT --data-source BASE
```

```
7 sas-admin dagentsrv catalogs delete --name BASE_CAT --cascade
```

- 1 Create a catalog registration for a BASE database driver.
- 2 Create a schema for a BASE data service and specify the catalog that you just created
- 3 Test the connection to the BASE database.
- 4 List the catalogs that are associated with the SAS Data Agent to verify that the catalog that you created is there.
- 5 List the schemas that are associated with the SAS Data Agent to verify that the schema that you created is there.
- 6 List the tables for the data service.
- 7 Delete the catalog that you created. Specify the `cascade` option to delete all logical schema definitions.

Example: Create a data service for a connection to an SQLServer database with the SAS Data Agent and verify the connection.

```
1 sas-admin dagentsrv security domains create --domain SQLSVR_DOMAIN
```



```
2 sas-admin dagentsrv security credentials create --domain QLSVR_DOMAIN
  --identity username --username database-username --password database-password
```

```
3 sas-admin dagentsrv data-services create sqlserver --name SQL_SERVICE
  --driver odbc --dsn tktssql --domain QLSVR_DOMAIN --register-all
```

```
4 sas-admin dagentsrv data-sources test --name SQL_SERVICE
```

```
5 sas-admin dagentsrv data-sources catalogs list --data-source SQL_SERVICE
```

```
6 sas-admin dagentsrv data-sources schemas list --data-source SQL_SERVICE
  --catalog catalog-name
```

```
7 sas-admin dagentsrv data-sources tables list --schema sys
  --data-source SQL_SERVICE --catalog catalog-name
```

```
8 sas-admin dagentsrv security domains delete --domain QLSVR_DOMAIN --cascade
```

```
9 sas-admin dagentsrv data-services delete --name SQL_SERVICE --cascade
```

- 1 Create a domain for the connection.
- 2 Create the credentials for the connection.
- 3 Create a data service definition for an SQLServer database. Register all catalogs that are visible to the connection.
- 4 Test the connection to the SQLServer database.
- 5 List the catalogs that are associated with the SAS Data Agent and the associated data service.
- 6 List the schemas that are associated with the SAS Data Agent and associated data service.
- 7 List the tables for the data service.
- 8 Delete the domain that you created. Specify the **cascade** option to delete any credentials in the domain.
- 9 Delete the data service definition that you created. Specify the **cascade** option to delete any dependent data source names, catalogs or schemas.

See Also

- “Command-Line Interface: Overview” in *SAS Viya Administration: Using the Command-Line Interfaces*
- Chapter 6, “Data Administration,”.

Chapter 7

Server Administration

SAS Data Agent Server Administration	79
Site and Database Backup	79
Apply a New or Renewal License	80
Restart the SAS Data Agent Server	80
Stop (Shut Down) the SAS Data Agent Server	81
Restart Microservices	81
Authentication and Authorization Services Resource Cache	81
Overview	81
Managing Named Server Caches	81
Managing Cache Configuration Properties	82
Cache Properties	83
Server Logging Configuration	83
Introduction	83
Initial Logging Configuration	84
FSNET Loggers	86
SQL Loggers	90
Logging Thresholds	90
Modifying the Server Logging Configuration	91
Adding a Logging Appender	91
Trace Log	91

SAS Data Agent Server Administration

Site and Database Backup

SAS Viya Backup and Restore

SAS Data Agent cloud components are handled with the SAS Viya backup agents. The backup and restore function of SAS Viya automatically discovers information about the SAS Viya deployment and backs up critical configuration and user content from the SAS Viya deployment. You can use the SAS Backup Manager application or command-line interface (CLI) utility to back up and restore the content. See [SAS Viya Administration: Backup and Restore](#) for tasks and additional information.

Data Agent Database (PostgreSQL)

The PostgreSQL database is your transactional database, also referred to as SYSCAT (System Catalog). It is very important to back up this data regularly. There are three different approaches for backing up PostgreSQL data:

- SQL dump
- file system-level backup
- continuous archiving and point-in-time recovery (PITR)

These backup approaches are explained in PostgreSQL documentation at <https://www.postgresql.org/docs/>. A copy of the PostgreSQL utilities is located in this directory: `/opt/sas/viya/home/bin` of your data agent server.

Apache Web Server

The Apache server is your data agent gateway. Server configuration should be backed up on a regular basis or added to your regular backup schedule.

Consul

Use the following commands to take a backup snapshot of consul and restore if needed.

Consul Backup

```
export CONSUL_HTTP_ADDR=localhost:8501
export CONSUL_HTTP_SSL=true
export CONSUL_CACERT=/opt/sas/viya/config/etc/SASSecurityCertificateFramework/
  tls/certs/consul/default/da-sasrbh-onprem.dmmdev.sashq-d.openstack.sas.com.crt
export CONSUL_HTTP_TOKEN=$(cat /opt/sas/viya/config/etc/
  SASSecurityCertificateFramework/tokens/consul/default/management.token)
/opt/sas/viya/home/bin/consul snapshot save backup.snap
```

Consul Restore

```
/opt/sas/viya/home/bin/consul snapshot restore backup.snap
```

Apply a New or Renewal License

The SAS software license is a `setinit.txt` file located in `opt/sas/viya/config/etc/dagentsrv/default` is delivered in your Software Order Email (SOE). The `.txt` file in the SOE uses a format similar to this:

```
SASViyaV0300_09N2L3_Linux_x86-64.txt.
```

Use the following steps to apply a license for your location:

1. Copy your existing **setinit** license file to a directory that is accessible, should you need to retrieve it.
2. Rename the new license to **setinit.txt** as you copy it to `opt/sas/viya/config/etc/dagentsrv/default`.
3. Restart the SAS Data Agent server.

Restart the SAS Data Agent Server

On the SAS Data Agent machine, restart SAS Data Agent server by running the following command:

- For Red Hat Enterprise Linux 6.x:

```
sudo service sas-viya-dagentsrv-default restart
```

- For Red Hat Enterprise Linux 7.x and SUSE Linux:

```
sudo systemctl restart sas-viya-dagentsrv-default
```

The start-up script registers the SAS Data Agent server in Consul and ensures that the Apache HTTP server is aware of the SAS Data Agent server.

Stop (Shut Down) the SAS Data Agent Server

In the event that you wish to stop the SAS Data Agent server, run the following command:

- For Red Hat Enterprise Linux 6.x:

```
sudo service sas-viya-dagentsrv-default stop
```

- For Red Hat Enterprise Linux 7.x and SUSE Linux:

```
sudo systemctl stop sas-viya-dagentsrv-default
```

Restart Microservices

Typically, microservices start when the SAS Viya cloud server is booted or restarted. Use the following commands in the cloud server in the event that microservices do not start automatically:

- For Red Hat Enterprise Linux 6.x:

```
sudo service sas-viya-dagentcont-default restart
sudo service sas-viya-dagentmgmt-default restart
```

- For Red Hat Enterprise Linux 7.x and SUSE Linux:

```
sudo systemctl restart sas-viya-dagentcont-default
sudo systemctl restart sas-viya-dagentmgmt-default
```

Authentication and Authorization Services Resource Cache

Overview

Authentication and authorization metadata that is used frequently can be cached and periodically refreshed while the information is needed or purged when no longer needed. Resource caching can help improve performance by reducing the number of calls needed from Cloud Data Exchange to the Identities and Credentials microservices.

Managing Named Server Caches

Cloud Data Exchange maintains several internal resource caches, which are designed to improve the performance of potentially expensive operations. Among the cached resources are user and group identity information. This information is required in authorization enforcement and multi-tiered authentication.

SAS Data Agent can cache resources that are related to authentication, reducing roundtrips to the Identities and Credentials microservices. Several of these caches are periodically repopulated as SAS Data Agent captures information from SASLogon microservices during the authentication process. The cache names prefixed with AS represent resources consumed by the internal authentication services run-time services. By default, these resources are not cached.

Cloud Data Exchange can also cache privilege information, reducing internal queries to various system tables related to privileges, thereby improving the rendering of authorization enforcement decisions. The authorization cache is periodically updated as Cloud Data Exchange performs authorization enforcement and processes permissions and various commands. The authorization cache is named **Authorization** and is configured at maximum level by default.

The following cache namespace table describes the information cached under each name.

Cache Name	Description
AS	All authentication service (AS) cached resources
AS.Name	Name to identifier mappings
AS.Name.Subjects	User name to authentication service identifier cache
AS.Name.Groups	Group name to authentication service identifier cache
AS.Subject	Per user cache resources
AS.Subject.Groups	User group memberships cache
AS.Subject.Principals	User-owned principals cache
AS.List	Directory listings
AS.List.Subjects	User listings cache
AS.List.Groups	Group listings cache
Authorization	Privileges cache

Managing Cache Configuration Properties

Common cache management operations are administered in the `da_serv.xml` configuration file. The configured time-out default is set to 5 minutes. To modify the default value, edit the `da_serv.xml` file and modify the TIMEOUT value to your desired option.

```
<Option name="Command">
  ALTER SERVER {OPTIONS( XSET CACHE(NAME AS, TIMEOUT 300))}
</Option>
```

Cache Properties

TIMEOUT *timeout*

All caches support the TIMEOUT option. The value for TIMEOUT specifies the length of time, in seconds, that a resource can be cached before being considered stale and marked for on-demand refresh. When a resource becomes stale, it is typically refreshed and reached on its next access. Here are the default TIMEOUT values associated with each of the caches:

NAME	Default TIMEOUT Value
Authorization	-1 (infinite)
All others	0 (not applicable – not cached)

The TIMEOUT property can be restored to a default several ways once it is explicitly configured. In the following scenario, the configured TIMEOUT values for result set caching are as follows:

PURGE | **FLUSH**

Specifies that the cache named should be refreshed. Associated resources are reacquired and cached on next access and can be flushed immediately. This option is not persisted, and its use does not affect existing properties already configured for the named cache. All caches support the FLUSH option.

LEVEL *level*

Controls the caching granularity of the named cache. This property applies to the Authorization cache only. Valid values are as follows:

ALL OBJECT	Cache privileges for columns, tables, and all higher level securables. This is the default privilege caching level.
CONTAINER	Cache privileges for schemas and all higher level securables.
NONE OFF	Turn off all privilege caching.

Server Logging Configuration

Introduction

The SAS logging facility is a framework that categorizes, filters, and writes log messages in SAS server and SAS programming environments. In the server environment, the logging facility logs messages based on predefined message categories, such as Admin for administrative messages, App for application messages, and Perf for performance messages. The logging facility also enables message filtering based on the following thresholds: TRACE, DEBUG, INFO, WARN, ERROR, and FATAL.

The `da_log4sas.xml` configuration file controls the destination, contents, and formats of the logging facility log for Cloud Data Exchange. You can change logging levels dynamically without stopping the server.

Initial Logging Configuration

The default logging facility configuration for Cloud Data Exchange includes a definition for the RollingFileAppender. The appender routes events to a rolling log file.

The rolling log file is configured as follows:

- A new log is created when the date changes and when a new server process is started.
- Events are written by using a layout that includes the current date, current time, logging level, process ID, the user identity that is associated with the event, and a message.
- The name of the rolling log file follows this convention:

```
da_%S{hostname}_%d_%S{pid}.log
```

where **%S{hostname}** is the hostname, **%d** is the date, and **%S{pid}** is the process ID number (PID) for Cloud Data Exchange.

- The rolling log files are placed in the `var/log/dagentsrv/default` directory.
- When a new rolling log file is created, a heading is written to the file. The heading identifies the server's host machine, operating system, and server start-up command.

The following table lists the loggers that reference the RollingFileAppender:

Log Name	Logging Level	Description
Admin	Info	Processes log events that are relevant to system administrators or computer operators.
App	Info	Processes log events that are related to specific applications. For example, OLAP servers, stored process servers, and workspace servers use loggers that are named <code>App.class.interface.method</code> to record method calls that are issued to the server.
App.Server	Info	Server top-level object run-time and interface events.
	Debug	Method call and return events.
	Trace	Method parameters.
App.Session	Info	Session object run-time and interface events.
	Debug	Method call and return events.
	Trace	Method parameters.

Log Name	Logging Level	Description
App.Connection	Info	Connection object run-time and interface events.
	Debug	Method call and return events.
	Trace	Method parameters.
App.Statement	Info	Statement object run-time and interface events.
	Debug	Method call and return events.
	Trace	Method parameters.
App.Program	Info	General application independent events including errors and warnings from arbitrary services or the OS.
Audit	Info	Processes log events to be used for auditing. These events include updates to public objects, user access to SAS libraries, accepted and rejected user authentication requests, and administration of users, groups, and access controls.
Audit Authentication	Info	Authentication provider events.
Audit Table	Info	Server-specific events.
Audit Table Connection	Info	Audit events related to server connections, including connection pooling and dynamic connections.
Audit.Table.Security	Info	Server authorization events.
Audit.Table.Security.Provider	Info	Detailed authorization services run-time events relating to user identity management and access control logic and enforcement decisions.
Logging	Error	SAS logging facility configuration and run-time events.
Logging.Appender	Error	Appender-specific configuration and run-time events.
Logging.Appender.DB	Error	DB Appender-specific events used in SQL Logging.
Cradle	Info	General server process framework services, start-up, and termination events.
Perf	Error	Processes log events that are related to system performance.

Log Name	Logging Level	Description
Perf.ARM	Error	Application Response Measurement performance events.
Perf.ARM.IOM.Session	Error	Session API performance events.
Perf.ARM.IOM.Environment	Error	Environment API performance events.
Perf.ARM.IOM.Connection	Error	Connection API performance events.
Perf.ARM.IOM.Statement	Error	Statement API performance events.
Perf.ARM.SQLServices	Warn	Local SQL services performance events.
<root>	Error	All events produced from SAS Data Agent.

FSNET Loggers

FSNET is used to configure remote connectivity for the Cloud Data Exchange data connector. To retrieve logging information for FSNET, add the logger App.tk.FSNet to the da_log4sas.xml configuration file:

```
<logger name="App.tk.FSNet">
  <level value="Info"/>
</logger>
```

Table 7.1 FSNET Logging

Log Name	Logging Level	Description
App.tk.FSNet	Info	Loading and Unloading the FSNet extension.
	Warn	Any warning SQLSTATE diagnostic records added by the FSNet layer without a server context.
	Error	Any error SQLSTATE diagnostic records added by the FSNet layer without a server context.

Log Name	Logging Level	Description
App.tk.FSNet.Server	Info	FSNet Server configuration options. Start or FSNet Server. CreateSession or Session with MD5'd UUIDs. Any info SQLSTATE diagnostic records added by the FSNet layer with a server context.
	Warn	Any warning SQLSTATE diagnostic records added by the FSNet layer with a server context.
	Error	Any unknown or invalid commands. Any error SQLSTATE diagnostic records added by the FSNet layer with a server context.
	Trace	The input arguments (as best determined) of any unknown or invalid commands. The response to any unknown or invalid commands . CreateSession or FreeSession arguments.
App.tk.FSNet.Environment	Warn	Any warning SQLSTATE diagnostic records added by the FSNet layer with an environment context.
	Error	Any error SQLSTATE diagnostic records added by the FSNet layer with an environment context.
	Debug	All FSNet requests with an environment context (MD5'd UUID and CMD only). All FSNet responses with an environment context (MD5'd UUID and return code only).
	Trace	All FSNet request args with an environment context (with MD5'd UUID). All FSNet response arguments with an environment context (with MD5'd UUID).

Log Name	Logging Level	Description
App.tk.FSNet.Connection	Warn	Any warning SQLSTATE diagnostic records added by the FSNet layer with a connection context.
	Error	Any error SQLSTATE diagnostic records added by the FSNet layer with a connection context.
	Debug	All FSNet requests with a connection context (MD5'd UUID and CMD only).
		All FSNet responses with a connection context (MD5'd UUID and return code only).
Trace	All FSNet request arguments with a connection context (with MD5'd UUID).	
	All FSNet response arguments with a connection context (with MD5'd UUID).	
App.tk.FSNet.Statement	Warn	Any warning SQLSTATE diagnostic records added by the FSNet layer with a statement context.
	Error	Any error SQLSTATE diagnostic records added by the FSNet layer with a statement context.
	Debug	All FSNet requests with a statement context (MD5'd UUID and CMD only).
		All FSNet responses with a statement context (MD5'd UUID and return code only).
Trace	All FSNet request arguments with a statement context (with MD5'd UUID).	
	All FSNet response arguments with a statement context (with MD5'd UUID).	
App.tk.FSNet.Statement.SQL	Debug	MD5'd statement UUID and SQL statement for Prepare and ExecDirect.
	Debug	MD5'd statement UUID and return code for Prepare and ExecDirect.

Log Name	Logging Level	Description
App.tk.FSNet.Socket	Info	Loading or Unloading the extension.
		Starting or Stopping the FSNet Socket server.
		Client and server socket configuration options along with connection open and close.
		Socket and port of incoming client (if provided).
		Any info SQLSTATE diagnostic records added by the socket layer.
	Warn	Socket read or write failed due to closed socket.
		Any warning SQLSTATE diagnostic records added by the socket layer.
	Error	Any unexpected return code that results in a closed connection.
		Any unexpected socket failures or error SQLSTATE diag reps added by the socket layer.
	Debug	Any object created or destroyed.
Whenever a socket is opened or closed.		
Some higher level internal calls (arguments are not dumped).		
Trace	SSL connection options.	
	Additional connection options set internally.	
	Additional internal calls (including some arguments and state information plus successful calls to internal functions).	
	Cookie information.	
	Socket data interpretation (for example, headers, chunk information, and so on).	
App.tk.FSNet.Socket.Read	Trace	Data buffers read from socket.
App.tk.FSNet.Socket.Write	Trace	Data buffers written to socket.

Note: This list covers logging as part of the FSNet service but it does not include log information from the underlying server. As an example: If a call on the server returns an error with diagnostic records, the errors are not logged as part of `App.tk.FSNet.Statement` (except for the return code in `Debug` and the response argument array in `Trace`). If an error occurs as part of FSNet processing (like an out of memory error allocating a buffer), it is logged as part of `App.tk.FSNet`.

SQL Loggers

Reserved Loggers for SQL

The following loggers, which are unique to Cloud Data Exchange, are based on the Audit and App loggers referenced above.

Logger	Logging Level	Description
<code>Audit.SQLFPkg.package-name</code>	Debug Trace	Used to log security-related events.
<code>App.SQLFPkg.package-name</code>	Debug Trace	Used to log API, logic run-time events.

Logging Thresholds

The SAS logging facility provides six thresholds: TRACE, DEBUG, INFO, WARN, ERROR, and FATAL. Thresholds are used to ignore log events that are lower than a particular level, or to filter messages so that only a single message level is logged. The SQL function loggers use DEBUG and TRACE only.

When a log event occurs, up to three levels of filtering can take place:

1. filtering log events by comparing the log event level to the log event's logger level
2. filtering log events by comparing the log event level to the appender's threshold
3. filtering log events by comparing the log event level to the threshold that is specified in the filter definition, which is a part of the appender configuration

In the first two cases, if the log event level is lower than the logger or appender threshold, the logging facility ignores the log event. Otherwise, processing of the log event continues.

In the third case, the log event level is compared to the filter threshold. If there is a match, the log event can be either accepted or denied. If there is no match, the filtering process continues to the next filter in the filtering policy.

The logging levels, from the lowest to the highest, are as follows:

Level	Description
TRACE	Produces the most detailed information about your application. This level is primarily used by SAS Technical Support or development.

Level	Description
DEBUG	Produces detailed information that you use to debug your application. This level is primarily used by SAS Technical Support or development.
INFO	Provides information that highlights the progress of an application.
WARN	Provides messages that identify potentially harmful situations.
ERROR	Provides messages that indicate that errors have occurred. The application might continue to run.
FATAL	Provides messages that indicate that severe errors have occurred. These errors might cause the application to end.

Note: The logging level must be enclosed in quotation marks.

By default, appenders do not have a threshold, but a threshold can be configured. When set, all log events that have a level lower than the threshold are ignored by the appender.

Modifying the Server Logging Configuration

You can modify the logging facility configuration for Cloud Data Exchange by modifying the `da_log4sas.xml` file located in the `/etc/dagentsrv/default` directory of the configuration path. Before modifying the file, be sure to make a backup copy.

Here are some examples of changes that you might want to make:

- Configure `RollingFileAppender` to use a different log filename, to roll over log files more or less frequently, or to roll over log files based on file size rather than date.
- Specify additional appenders.
- Use filters to limit the events that are written to an appender.
- Configure a different message layout for an appender.

Adding a Logging Appender

To capture additional information, you can add a logger level to the `da_log4sas.xml` configuration file.

Trace Log

By tracing each internal API routine that is called by the application, a trace log records transactions that can be used for debugging connection and processing issues. By default, tracing is not activated for server logging. You should not activate tracing unless you are instructed to do so by SAS Technical Support.

Tracing can be activated by using the following methods:

- connection string options
- server start-up options
- data service connection arguments

- system options

When you activate tracing, you also specify the physical location where the transaction records are saved. Because Cloud Data Exchange supports one root-file, trace-log directory and multiple subdirectories, you can group trace logs if necessary.

Part 3

Appendices

<i>Appendix 1</i>	
CLI Connection Options	95
<i>Appendix 2</i>	
Legal Notices	107

Appendix 1

CLI Connection Options

These connection options are for use within the data agent CLI connection options argument: **sas-admin dagentsrv data-services create Oracle --name MyOra [connection options]**. A connection option is represented by a **keyword=value** pair, where the equal sign (=) connects each keyword and its value. A connection string consists of a series of **keyword=value** pairs separated by commas. Follow these guidelines when constructing a connection string:

- Use double quotation marks when grouping values that contain special characters or spaces (white space).
- If you are using a **--conopts** string, any keyword or value that contains special characters, such as spaces or commas, should be enclosed in single quotation marks. Here is an example:

```
--conopts "Driver='{ODBC SQL Server}',server='tcp:mydata.abc.com,1433'"
```

Here is an example of a connection that uses a **--conopts** string: **/opt/sas/viya/home/bin/**

```
sas-admin dagentsrv services create sqlserver --name SQL_SERVICE
--domain SQLSVR --conopts "Driver='{SAS Institute, Inc 7.1 SQL Server Wire
Protocol}',Server='server-name.domain.com,1433',Database=TEST"
```

This table describes connection option names, with the corresponding connection string value, and a brief description.

Keyword	CLI Option Name /Description
ACCESS	--access=readonly temp Specify access to a BASE table; can be READONLY or TEMP. DRIVER: BASE
ACCOUNT	--account=account_number Specify an optional account number used to charge your Teradata session. DRIVER: TERADATA
ALLOW_UNQUOTED_NAMES	--allow-unquoted-names=Y N Specifies whether to enclose table and column names in quotation marks. Tables and columns are quoted when this option is set at NO (default). If set to YES, the driver does not automatically add quotation marks to table and column names if they are not specified. DRIVER: POSTGRES, REDSHIFT

Keyword	CLI Option Name /Description
AUTHENTICATION MODE	<p>--auth-mode=default kerberos</p> <p>Specify authentication mode: DEFAULT or KERBEROS. If using Kerberos for the authentication mode, you must specify the Hive principal host name using the HIVE_PRINCIPAL connection option.</p> <p>DRIVER: HIVE</p>
CLIENT_ENCODING	<p>--client-encoding encoding value</p> <p>Specify the character set encoding value for the client data.</p> <p>DRIVER: HIVE, DB2, ODBC, POSTGRES, MSSQLSVR, TERADATA</p> <p><i>Note:</i> See ENCODING for the Base SAS driver.</p>
COMPRESS	<p>--compress=NO YES (CHAR) BINARY</p> <p>Controls the compression of rows in created SAS data sets. Compress data, can be NO, YES, or BINARY:</p> <p>NO Specifies that the rows in a newly created SAS data set are uncompressed (fixed-length records). NO is the default.</p> <p>YES (CHAR) Specifies that the rows in a newly created SAS data set are compressed (variable-length records) by using RLE (Run Length Encoding). RLE compresses rows by reducing repeated consecutive characters (including blanks) to two- or three-byte representations. Use this compression algorithm for character data.</p> <p>BINARY Specifies that the rows in a newly created SAS data set are compressed (variable-length records) by using RDC (Ross Data Compression). RDC combines run-length encoding and sliding-window compression to compress the file. This method is highly effective for compressing medium to large (several hundred bytes or larger) blocks of binary data (numeric columns). Because the compression function operates on a single record at a time, the record length must be several hundred bytes or larger for effective compression.</p> <p>DRIVER: BASE</p>
CONOPTS	<p>--conopts=(ODBC-compliant connection-string)</p> <p>Specify additional subordinate connection options as "name=value,..." pairs.</p> <p>DRIVER: FEDSVR, ODBC, POSTGRES, REDSHIFT</p>

Keyword	CLI Option Name /Description
CT-PRESERVE	<p>--ct-preserve=STRICT SAFE FORCE FORCE_COL_SIZE</p> <p>Allows users to control how data types are mapped. Note that data type mapping is disabled when CT_PRESERVE is set to STRICT. If the requested type does not exist on the target database, an error is returned. The options are as follows:</p> <ul style="list-style-type: none"> • STRICT: The requested type must exist in the target database. No type promotion occurs. If the type does not exist, an error is returned. • SAFE: Target data types are upscaled only if they do not result in a loss of precision or scale. When character encodings are changed, the new column size is recalculated to ensure all characters can be stored in the new encoding. • FORCE: This is the default for all drivers. The best corresponding target data type is chosen, even if it could potentially result in a loss of precision or scale. When character encodings are changed, the new column size is recalculated to ensure all characters can be stored in the new encoding. • FORCE_COL_SIZE: This option is the same as FORCE, except that the column size for the new encoding is the same as the original encoding. This option can be used to avoid column size creep. However, the resulting column might be too large or too small for the target data. <p>DRIVER: ALL</p>
DATABASE	<p>--database-name name</p> <p>Identifies the database to which you want to connect.</p> <p>DRIVER: DB2, MSSQLSVR, TERADATA, REDSHIFT, POSTGRES, SAPHANA</p>
DATABASE PASSWORD PWD	<p>--database-password password</p> <p>Specify the password used to connect to the database.</p> <p>DRIVER: ODBC, REDSHIFT, MSSQLSVR (PASSWORD)</p> <p>DRIVER: HIVE, SAPHANA, ORACLE, POSTGRES, DB2 (PWD)</p>
DATABASE USER: UID USER	<p>--database-user UID USER user-ID</p> <p>Specify the user ID that is used to connect to the database. Apache HIVE and Oracle: UID, All others: USER</p> <p>DRIVER: HIVE, DB2, BASE, FEDSVR, ODBC, ORACLE, POSTGRES, REDSHIFT, SAPHANA, MSSQLSVR, TERADATA</p>
DBMAX_TEXT	<p>--dbmax-text 32767</p> <p>Specifies the length for a string data type. The maximum length is 2 gigabytes. The default is 32767.</p> <p>DRIVER: HIVE</p>

Keyword	CLI Option Name /Description
DEFAULT_ATTR	<p data-bbox="472 239 922 264">--default-attr "name=value,..."</p> <p data-bbox="472 285 1366 338">Used to specify connection handle or statement handle attributes supported for initial connect-time configuration, where name=value corresponds to any of the following options:</p> <ul data-bbox="472 359 1366 800" style="list-style-type: none"> <li data-bbox="472 359 1366 411">• cursor=0 1 2: Connection handle option. This option controls the driver's use of client side result set cursors. 2 is the default. <ul data-bbox="762 443 1366 800" style="list-style-type: none"> <li data-bbox="762 443 1366 537">0 Causes the driver to use client-side static cursor emulation if a scrollable cursor is requested but the database server cannot provide one. <li data-bbox="762 569 1366 663">1 Causes the driver to always use client-side static cursor emulation if a scrollable cursor is requested. The database server's native cursor is not used. <li data-bbox="762 695 1366 800">2 (Default) Causes the driver to never use client-side static cursor emulation if a scrollable cursor is requested. The database server's native cursor is used if available. Otherwise, the cursor is forward-only. <p data-bbox="504 821 991 846">EXAMPLE: DEFAULT_ATTR=(CURSOR=2)</p> <ul data-bbox="472 863 1366 915" style="list-style-type: none"> <li data-bbox="472 863 1366 915">• USE_EVP=0 1: Statement handle option. This option optimizes the driver for large result sets. The possible values are 0 (OFF) or 1 (ON), which is the default. <p data-bbox="504 936 991 961">EXAMPLE: DEFAULT_ATTR=(USE_EVP=0)</p> <ul data-bbox="472 978 1366 1062" style="list-style-type: none"> <li data-bbox="472 978 1366 1062">• XCODE_WARN=0 1 2: Statement handle option. Used to warn on character transcoding errors that occur during row input or output operations. Possible values are 0 (returns an error), 1 (returns a warning), or 2 (ignore transaction errors). 0 is the default. <p data-bbox="504 1083 1038 1108">EXAMPLE: DEFAULT_ATTR=(XCODE_WARN=1)</p> <p data-bbox="472 1125 1318 1178">DRIVER: HIVE, DB2, BASE, FEDSVR, ODBC, ORACLE, POSTGRES, REDSHIFT, SAPHANA, MSSQLSVR, TERADATA</p>

Keyword	CLI Option Name /Description
DEFAULT_CURSOR_TYPE	<p>--default-cursor-type FORWARD_ONLY KEYSET_DRIVEN DYNAMIC STATIC</p> <p>Specifies a valid default cursor type for new statements. The valid options are:</p> <p>FORWARD_ONLY Specifies a non-scrollable cursor that moves only forward through the result set. Forward-only cursors are dynamic in that all changes are detected as the current row is processed. If an application does not require scrolling, the forward-only cursor retrieves data quickly, with the least amount of overhead processing.</p> <p>KEYSET_DRIVEN Specifies a scrollable cursor that detects changes that are made to the values of rows in the result set. However, that does not always detect changes to deletion of rows and changes to the order of rows in the result set. A keyset-driven cursor is based on row keys, which are used to determine the order and set of rows that are included in the result set. As the cursor scrolls the result set, it uses the keys to retrieve the most recent values in the table.</p> <p>DYNAMIC Specifies a scrollable cursor that detects changes that are made to the rows in the result set. All INSERT, UPDATE, and DELETE statements that are made by all users are visible through the cursor. The dynamic cursor is good for an application that must detect all concurrent updates that are made by other users.</p> <p>STATIC Specifies a scrollable cursor that displays the result set as it existed when the cursor was first opened. The static cursor provides forward and backward scrolling. If the application does not need to detect changes but requires scrolling, the static cursor is a good choice.</p> <p>DRIVER: ODBC, REDSHIFT, SAPHANA, MSSQLSVR</p> <p><i>Note:</i> The application can still override this value, but if the application does not explicitly set a cursor type, this value will be in effect</p>
DM_UNICODE	<p>--dm-unicode UNICODE-value</p> <p>Specifies the Unicode setting for the ODBC NDriver Manager. The default is UTF-8.</p> <p>DRIVER: ODBC, MSSQLSVR</p>
DRIVER	<p>--driver driver-name</p> <p>Specifies which database driver to use (defaults to associated data service type).</p> <p>DB2 FEDSVR HIVE ODBC ORACLE POSTGRES REDSHIFT SAPHANA MSSQLSVR TERADATA</p>

Keyword	CLI Option Name /Description
DRIVER_TRACE	<p>--driver-trace API SQL ALL</p> <p>Specify optional additional tracing information, which logs transaction records to an external file that can be used for debugging purposes. The tracing levels are:</p> <p>ALL Activates all trace levels.</p> <p>API Specifies that API method calls be sent to the trace log. This option is most useful if you are having a problem and need to send a trace log to Technical Support for troubleshooting.</p> <p>DRIVER Specifies that driver-specific information be sent to the trace log.</p> <p>SQL Specifies that SQL statements that are sent to the database management system (DBMS) be sent to the trace log. Tracing information is DBMS specific, but most drivers log SQL statements such as SELECT and COMMIT.</p> <p><i>Note:</i> If you activate tracing, you must also specify the location of the trace log with DRIVER_TRACEFILE.</p> <p>DRIVER: HIVE, DB2, BASE, FEDSVR, ODBC, ORACLE, POSTGRES, REDSHIFT, SAPHANA, MSSQLSVR, TERADATA</p>
DRIVER_TRACEFILE	<p>--driver-trace-file file-name</p> <p>Specify the name of the text file for the trace log. Include the filename and extension in single or double quotation marks. For example: --driver-trace-file 'mytrace.log'</p> <p>DRIVER: HIVE, DB2, BASE, FEDSVR, ODBC, ORACLE, POSTGRES, REDSHIFT, SAPHANA, MSSQLSVR, TERADATA</p>
DRIVER_TRACEOPTIONS	<p>--driver-trace-options APPEND THREADSTAMP TIMESTAMP</p> <p>Specifies options in order to control formatting and other properties for the trace log:</p> <p>APPEND Adds trace information to the end of an existing trace log. The contents of the file are not overwritten.</p> <p>THREADSTAMP Prepends each line of the trace log with a thread identification.</p> <p>TIMESTAMP Prepends each line of the trace log with a time stamp.</p> <p>DRIVER: HIVE, DB2, BASE, FEDSVR, ODBC, ORACLE, POSTGRES, REDSHIFT, SAPHANA, MSSQLSVR, TERADATA</p>
DSN	<p>--dsn data-source-name</p> <p>Specify the data source name (DSN). For PostgreSQL use PTG_DSN. For ODBC and SQL Server, use ODBC_DSN.</p> <p>DRIVER: FEDSVR, DSN ODBC and MSSQLSVR, use ODBC_DSN POSTGRES use PTG_DSN</p>

Keyword	CLI Option Name /Description
ENABLE_MARS	<p>--enable-mars YES NO</p> <p>Enables or disables the use of multiple active result sets (MARS) on SQL Server.</p> <p>DRIVER: MSSQLSVR, ODBC</p>
ENCODING	<p>--client-encoding encoding-value</p> <p>Specify name of character set encoding of client data.</p> <p>DRIVER: BASE</p>
HDFS_TEMPDIR	<p>--hdfs-tempdir path to temporary</p> <p>Specifies the HDFS directory used for temporary read and write data.</p> <p>DRIVER: HIVE</p>
HD_CONFIG	<p>--hd-config path-to-hadoop-configuration-file</p> <p>Specifies the name and path for the Hadoop cluster configuration file.</p> <p>DRIVER: HIVE</p>
HIVE_PRINCIPAL	<p>--hive-principal principal-string</p> <p>Specify the Hive principal string.</p> <p>DRIVER: HIVE</p>
LOCKTABLE	<p>--locktable SHARED EXCLUSIVE</p> <p>Places exclusive or shared locks on SAS data sets. You can lock tables only if you are the owner or have been granted the necessary privilege. The default value is SHARED.</p> <p>SHARED</p> <p>Locks tables in shared mode, allowing other users or processes to read data from the tables, but preventing other users from updating.</p> <p>EXCLUSIVE</p> <p>Locks tables exclusively, preventing other users from accessing any table that you open</p> <p>DRIVER: BASE</p>
LOGIN_TIMEOUT	<p>--login-timeout #secs</p> <p>Specifies the number of seconds before a non-responsive connection fails.</p> <p>DRIVER: HIVE</p>
MAXPARMSIZE	<p>--maxparmsize size-in-bytes</p> <p>Specify the maximum byte size of an SQL parameter. Specifies the maximum byte limit for parameter bindings for variable length data types (VARCHAR, CHAR, VARBINARY, BINARY). Use this connection option if the number of required parameters exceeds the driver's limit of 64,256 bytes. The default value is 8K (8192 bytes). Alias: MPS</p> <p>DRIVER: TERADATA</p>
MAX_BINARY_LEN	<p>--max-binary-len value</p> <p>Specifies a value, in bytes, that limits the length of long binary fields (LONG VARBINARY). Unlike other databases, PostgreSQL does not have a size limit for long binary fields. The default is 1048576.</p> <p>DRIVER: POSTGRES, REDSHIFT</p>

Keyword	CLI Option Name /Description
MAX_CHAR_LEN	<p>--max-char-len value</p> <p>Specifies a value that limits the length of character fields (CHAR and VARCHAR). The default is 2000.</p> <p>DRIVER: POSTGRES, REDSHIFT</p>
MAX_TEXT_LEN	<p>--max-text-len value</p> <p>Specifies a value that limits the length of long character fields (LONG VARCHAR). The default is 409500.</p> <p>DRIVER: POSTGRES, REDSHIFT</p>
ORA_ENCODING	<p>--ora-encoding UNICODE</p> <p>Specify if strings returned from Oracle are in UNICODE format. The valid value is UNICODE.</p> <p>DRIVER: ORACLE</p>
ORNUMERIC	<p>--ornumeric NO YES</p> <p>Specifies how numbers read from or inserted into the Oracle NUMBER column are treated. This option defaults to YES so that a NUMBER column with precision or scale is described as TKTS_NUMERIC. This option can be specified as both a connection option and a table option. When specified as both connection and table option, the table option value overrides the connection option.</p> <ul style="list-style-type: none"> • NO indicates that the numbers are treated as TKTS_DOUBLE values. They might not have precision beyond 14 digits. • YES indicates that non-integer values with explicit precision are treated as TKTS_NUMERIC values. This is the default setting. <p>DRIVER: ORACLE</p>
PATH	<p>--path tktsora</p> <p>Specifies the Oracle connect identifier as defined in tnsnames.ora or other naming method. A connect identifier can be a net service name or a database service name that resolves to a connect descriptor.</p> <p>DRIVER: ORACLE</p>
PATH_BIND	<p>--path-bind CONNECT ACCESS</p> <p>Specifies when and how schemas are validated during connection. CONNECT validates the entire connection string at the time of connection and returns an error if one or more schemas is invalid. ACCESS validates schemas when they are accessed so that processing continues regardless of errors in the schema portion of the connection string.</p> <p>DRIVER: BASE</p>
PORT	<p>--port port_number</p> <p>Specifies the port number of the Federation Server HOST that you are connecting to. There is no default port number associated with this option. Therefore, PORT must be specified.</p> <p>DRIVER: FEDSVR</p>

Keyword	CLI Option Name /Description
PRIMARY PATH	<p>--primary-path path-to-SAS-library</p> <p>A SAS library is a group of SAS files that are stored in the same directory.</p> <p>DRIVER: BASE</p>
PROPERTIES	<p>--properties</p> <p>Specify the JDBC properties to override default properties.</p> <p>DRIVER: HIVE</p>
PROTOCOL	<p>--protocol=bridge</p> <p>Specifies the protocol for the connection. At this time, BRIDGE is the only option and is the default.</p> <p>DRIVER=FEDSVR</p>
ROLE	<p>--role=security-role</p> <p>Specify a security role for the session.</p> <p>DRIVER=TERADATA</p>
SCHEMA	<p>--schema=schema-name</p> <p>Specify the name of the associated schema.</p> <p>DRIVER=ODBC, POSTGRES, MSSQLSVR</p>
(SCHEMA) NAME	<p>--name=schema-name</p> <p>Specifies an arbitrary identifier for an SQL schema. The schema identifier is an alias for the physical location of the SAS library, which is much like the Base SAS libref. A schema name must be a valid SAS name and can be up to 32 characters long.</p> <p>DRIVER=BASE</p>
SCHEMA (ATTRIBUTES)	<p>--schema="name=value,..."</p> <p>Specifies schema attributes that are specific to a SAS data set. A schema is a data container object that groups tables. The schema contains a name, which is unique within the catalog that qualifies table names.</p> <p>DRIVER=BASE</p>
SERVER	<p>--server=address</p> <p>Specify the address of the database server.</p> <p>DRIVER=HIVE, FEDSVR, POSTGRES, REDSHIFT, TERADATA, SAPHANA</p>
SERVICE	<p>--service</p> <p>Specify the UNIX or Windows service name to use for the port.</p> <p>DRIVER=FEDSVR</p>

Keyword	CLI Option Name /Description
SSL CREATE SELF SIGNED CERTIFICATE	<p>--sslcreateselfsignedcertificate=YES NO</p> <p>Specifies if a self-signed certificate is created if the keystore cannot be found. If set to YES, a self-signed certificate is created in the event that the keystore is not found. If this option is not specified, the driver uses the default, which is NO.</p> <p>DRIVER=SAPHANA</p>
SSL CRYPTO PROVIDER	<p>--sslcryptoprovider=SAPCRYPTO OPENSLL MSCRYPTO</p> <p>Specifies the cryptographic library provider for SSL connectivity.</p> <p>DRIVER=SAPHANA</p>
SSL HOST NAME IN CERTIFICATE	<p>--sslhostnameincertificate='host-name'</p> <p>Specifies the host name to use for validation. Use this host name when validating a server's certificate using SSLVALIDATECERTIFICATE.</p> <p>DRIVER=SAPHANA</p>
SSL KEY STORE	<p>--sslkeystore='path-to-the-keystore-file'</p> <p>Specifies the path to the keystore file that contains the server's private key. If a value is not specified, the ODBC driver uses the default <code>\$HOME/.ssl/key.pem</code>.</p> <p>DRIVER=SAPHANA</p>
SSL TRUSTSTORE	<p>--ssltruststore='path-to-the-truststore-file'</p> <p>Specifies the path to the truststore file that contains the server's certificate. If a value is not specified, the ODBC driver uses the default <code>\$HOME/.ssl/trust.pem</code>.</p> <p><i>Note:</i> Leave this option empty if you are using the mscrypto cryptographic library.</p> <p>DRIVER=SAPHANA</p>
SSL VALIDATE CERTIFICATE	<p>--sslvalidatecertificate=YES NO</p> <p>Set this option to YES to validate the server's certificate. The default is NO. If this option is not specified, the ODBC driver uses the default and does not validate certificates.</p> <p>DRIVER=SAPHANA</p>
STRIP BLANKS	<p>--strip-blanks=YES NO</p> <p>Specify if blanks are removed from end of strings.</p> <p>DRIVER=POSTGRES, REDSHIFT</p>
SUBPROTOCOL	<p>--subprotocol=HIVE HIVE2</p> <p>Specifies whether you are connecting to a Hive service or a HiveServer2 (Hive2) service. The default is Hive2.</p> <p>DRIVER=HIVE</p>
TIME TYPE	<p>--time-type=YES NO</p> <p>Specifies the format used for date types. YES is the default behavior, which returns the date type as DATES. When NO is specified, date types are formatted as DOUBLES.</p> <p>DRIVER=BASE</p>

Keyword	CLI Option Name /Description
URI	--uri=proxy-address Specifies a proxy with a URI instead of using the PROXYLIST option. If using both the URI and PROXYLIST options, URI takes precedence and overrides PROXYLIST. DRIVER=FEDSVR
USE CACHED CATALOG	--use-cached-catalog YES NO Specify if Oracle should use cached catalog (value YES) versus recompile catalog (value NO) DRIVER: ORACLE
USER PRINCIPAL	--user-principal Specify the User principal string for HDFS and JDBC paths using JAAS to perform a doAs for the given Kerberos user principal. DRIVER: HIVE

Appendix 2

Legal Notices

Microsoft Copyright Disclosure

Microsoft®, Windows, NT, SQL Server, and Access, are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle Copyright Disclosure

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates.

Red Hat Copyright Disclosure

Red Hat® Enterprise Linux®, and Red Hat Fedora™ are registered trademarks of Red Hat, Inc. in the United States and other countries.

SAS Copyright Disclosure

Portions of this software and documentation are copyrighted by SAS® Institute Inc., Cary, NC, USA, 2009. All Rights Reserved.

Sun Microsystems Copyright Disclosure

Java™ is a trademark of Sun Microsystems, Inc. in the U.S. or other countries.

TomTom Copyright Disclosure

© 2006-2015 TomTom. All rights reserved. This material is proprietary and the subject of copyright protection, database right protection, and other intellectual property rights owned by TomTom or its suppliers. The use of this material is subject to the terms of a license agreement. Any unauthorized copying or disclosure of this material will lead to criminal and civil liabilities.

