SAS® Data Loader 2.4 for Hadoop: User's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS® Data Loader 2.4 for Hadoop: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Data Loader 2.4 for Hadoop: User's Guide

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

Printing 1, January 2016

Contents

What's New

What's New in SAS Data Loader 2.4 for Hadoop

Overview

The main enhancements for SAS Data Loader 2.4 include the following:

Combine Multiple Tables and Merge Matching Rows

Use the new Match-Merge Data directive to combine columns from multiple source tables into a single target table. You can also merge data in specified columns when rows match in two or more source tables. Columns can match across numeric or character data types. Additional features in the directive include the following:

- Rename and copy into the target specified unmatched columns.
- Filter rows from the target using multiple rules combined with an expression, or filter with an expression only.
- Evaluate target rows using expressions, and write the results into new target columns.

To learn more, see "Match-Merge Data".

Execute Multiple Jobs in Series or in Parallel

The new directive named Chain Directives runs two or more saved directives in series or in parallel. One chain directive can contain another chain directive. A serial chain can execute a parallel chain, and a parallel chain can execute a serial chain. An individual directive can appear more than once in a serial directive. Results can be viewed for each directive in a chain as soon as those results become available. To learn more, see "Chain Directives".

Improve Your Data with Clustering and Survivorship

The new Cluster-Survive Data directive uses rules to create clusters of similar rows. Additional rules can be used to construct a survivor row that replaces the cluster of rows in the target. The survivor row combines the best values in the cluster. This directive requires the Apache Spark run-time environment. To learn more, see "Cluster-Survive Data".

Run SQL Programs in Impala or Hive

The directive Run a Hadoop SQL Program replaces the former directive Run a Hive Program. The new directive can use either Impala SQL or HiveQL. In the directive, a Resources box lists the functions that are available in the selected SQL environment. Selecting a function displays syntax help. A click moves the selected function into the SQL program. To learn more, see "Run a Hadoop SQL Program".

Increase Performance Using Spark and Impala

Support for Apache Spark brings massively parallel in-memory processing to the Cleanse Data, Transform Data, and Cluster-Survive directives. Spark-enabled

directives use DataFlux EEL functions in user-written expressions. The Spark-enabled EEL functions replace the default DS2 functions, which support MapReduce. The SAS Data Management Accelerator for Spark manages processing in Hadoop.

Support for Cloudera Impala brings Impala SQL processing to the directives Query or Join, Sort and De-Duplicate, and Run a Hadoop SQL Program. User-written expressions use Impala SQL functions instead of the default HiveQL functions.

Spark and Impala can be enabled by default for new directives. Individual directives can override the default and always use Spark or MapReduce, or Impala or Hive.

Impala is enabled in the Configuration window, in the **Hadoop Configuration** panel. **Host** and **Port** fields specify both the Hive and Impala servers. **Test Connection** buttons validate the host and port entries and confirm the operational status of the servers.

Spark support is enabled by simply selecting that value in the **Preferred runtime target** list box.

Saved directives that are enabled for Spark or Impala are indicated as such in the Run Status directive.

Saved directives that were created before the implementation of Impala and Spark will continue to run as usual using Hive and MapReduce.

To learn more, see “Enable Support for Impala and Spark”.

Cleanup Hadoop Using Table Delete

Hadoop tables can now be selected and deleted in the Source Table and Target Table tasks. To learn more, see “Delete Tables”.

Added Support for Pivotal HD and IBM Big Insights

Support is now available for the Hadoop distributions Pivotal HD and IBM Big Insights. New versions of Cloudera, Hortonworks, and MapR are supported. Kerberos is not supported in combination with MapR or IBM Big Insights. To learn about the supported versions of these distributions, see SAS Data Loader 2.4 for Hadoop: System Requirements.

Enhance the Performance of Profile Jobs

In the Configuration window, the **Profile** panel now configures the number of threads used in Profile Data directives. Also added is the ability to specify the

number of MapReduce jobs that are run by Profile Data directives. For more information, see “Profiles Panel”.

Schedule Jobs Using REST API

A REST API can now be used in a scheduling application to run saved directives. The API can also return a job’s state, results, log file, or error message. Job controls in the API can cancel running jobs and delete job information. To learn more, see the SAS Data Loader for Hadoop: REST API Reference.

Improved Syntax Editing

The **Code** task now displays generated code in a syntax editor rather than a text editor. The **Code** task appears near the **Target Table** task in the directives that generate code.

Apply and Reload Hadoop Configuration Changes

In the Configuration window, the **Hadoop Configuration** panel now includes **Apply** and **Reload** buttons. Click **Apply** to validate your changes. Click **Reload** to restore your previous configuration. For more information, see “Hadoop Configuration Panel”.

Added Support for VirtualBox and VMware Hypervisors

The following hypervisors can now be used with SAS Data Loader for Hadoop: VMware Workstation 12, VMware Workstation 12 Pro, and Oracle VM VirtualBox 5. Hypervisors run the vApp virtual machine on Windows client computers. For more information, see the SAS Data Loader 2.4 for Hadoop: System Requirements.

Directive Names No Longer Use “in Hadoop”

The words “in Hadoop” no longer appear at the end of the directive names. In previous releases, the words appeared in the SAS Data Loader window. The words also appeared in the names of new directives.

1

About SAS Data Loader for Hadoop

What is SAS Data Loader for Hadoop?

SAS Data Loader for Hadoop is a software offering that makes it easier to move, cleanse, and analyze data in Hadoop. It enables business users and data scientists to do self-service data preparation on a Hadoop cluster.

Hadoop is highly efficient at storing and processing large amounts of data. However, moving, cleansing, and analyzing data in Hadoop can be labor-intensive, and these tasks usually require specialized coding skills. As a result, business users and data scientists usually depend on IT personnel to prepare large Hadoop data sets for analysis. This technical overhead makes it harder to turn Hadoop data into useful knowledge.

SAS Data Loader for Hadoop provides a set of “directives” or wizards that help business users and data scientists do the following tasks:

- Copy data to and from Hadoop, using parallel, bulk data transfer.
- Perform data integration, data quality, and data preparation tasks within Hadoop, without writing complex MapReduce code or asking for outside help.
- Minimize data movement for increased scalability, governance, and performance.
- Load data in memory to prepare it for high-performance reporting, visualization, or analytics.

What Does It Help You Do?

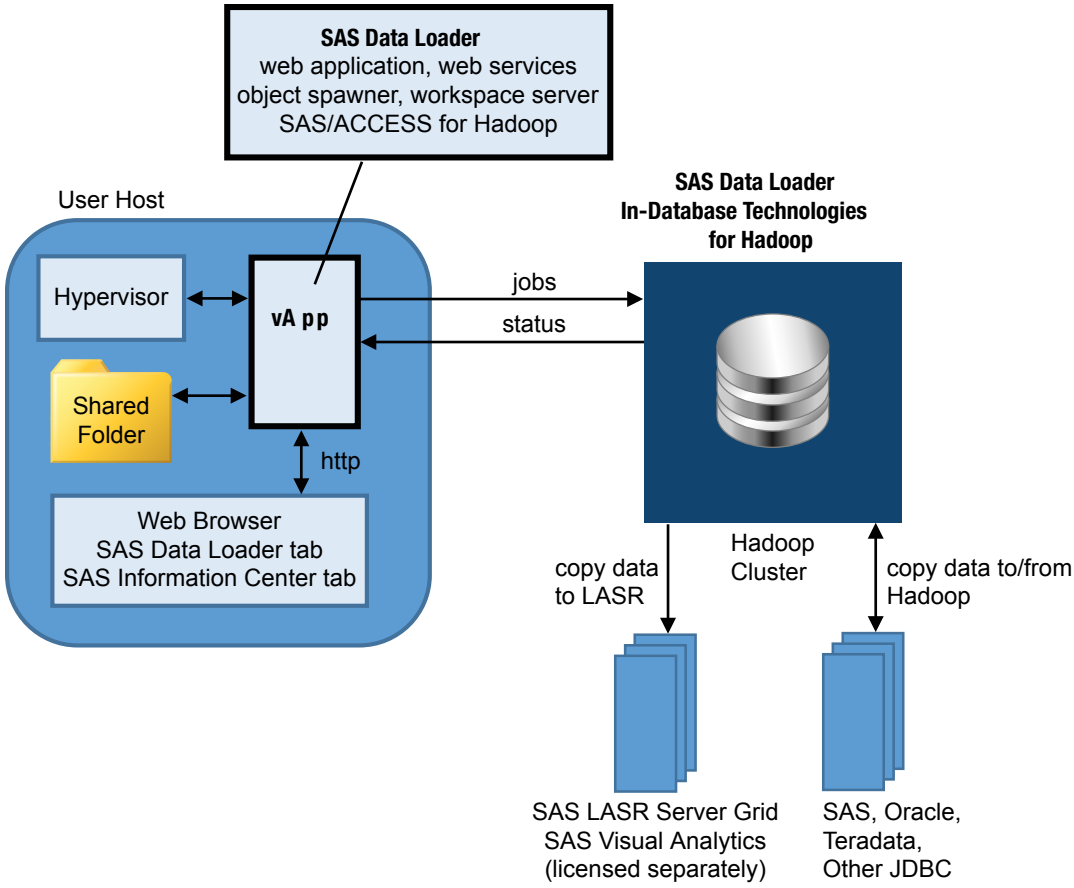
SAS Data Loader enables business users and data scientists to perform tasks such as the following:

Tasks	Description
Copy data to and from Hadoop	Copy relational databases and SAS data sets to and from Hadoop via parallel, bulk data transfer. For more information, see “Copy Data to Hadoop” on page 126, and “Copy Data from Hadoop” on page 144. Import data from delimited text files, such as comma-separated value (CSV) files. For more information, see “Import a File” on page 138.
Transform and transpose data	Transform data by filtering rows, managing columns, and summarizing rows. For more information, see “Transform Data” on page 96. Select columns and transpose or group them. For more information, see “Transpose Data” on page 105.
Cleanse data	Standardize, match, parse, and perform other data quality functions on data in Hadoop. For more information, see “Cleanse Data” on page 30. Use rules and expressions to filter data. For more information, see “About Expressions and the Advanced Editor” on page 48.
Sort or de-duplicate data	Sort data in an existing table and remove duplicate rows from the table. For more information, see “Sort and De-Duplicate Data” on page 89.
Query or join data	Query a table or join multiple tables without knowing SQL. For more information, see “Query or Join Data” on page 81. Run aggregations on selected columns. For more information, see “About the Aggregations in the Summarize Rows Transformation” on page 104. Power users can generate and edit a HiveQL query, or paste and run an existing HiveQL query. For more information, see “Run a Hadoop SQL Program” on page 159.
Profile data and save profile reports	Analyze source columns from one or more tables to determine patterns, uniqueness, and completeness. For more information, see “Profile Data” on page 111. View data profile reports. Add notes to a data profile report to explain a result or ask a question.

Tasks	Description
Run user-written code	Use the Run a SAS Program directive to execute user-written Base SAS code or DS2 code. For more information, see “Run a SAS Program” on page 157. Use the Run a Hive Program directive to execute user-written Hive code. For more information, see “Run a Hadoop SQL Program” on page 159.
Manage and reuse directives	Use directives to guide you through the process of creating and running jobs in Hadoop. View the status of current and previous job executions. For more information, see “Run Status” on page 164. Stop and start directives. Open their logs and generated code files. Run, view, or edit saved directives for reuse. For more information, see “Saved Directives” on page 167.
Load data to SAS LASR Analytic Server	Load specified Hadoop columns in memory onto the SAS LASR Analytic Server for analysis using SAS Visual Analytics or SAS Visual Statistics (licensed separately). For more information, see “Load Data to LASR” on page 155.
Specify global options	Specify server connections, data sources, global options, and other settings for SAS Data Loader. For more information, see “Set Global Options” on page 176.

How Does It Work?

SAS Data Loader for Hadoop is a software offering that includes SAS Data Loader, SAS/ACCESS Interface to Hadoop, SAS In-Database Code Accelerator for Hadoop and SAS Data Quality Accelerator for Hadoop. The following diagram illustrates an installed configuration.



SAS Data Loader for Hadoop runs inside a virtual machine or vApp. The vApp is a complete and isolated operating environment that is accessed through a web browser. Each instance of SAS Data Loader for Hadoop is accessed by a single user. The vApp is started and stopped by a hypervisor application such as VMware Player Pro. SAS Data Loader for Hadoop uses SAS software in the vApp and on the Hadoop cluster to manage data within Hadoop.

The hypervisor provides a web (HTTP) address that you enter into a web browser. The web address opens the SAS Data Loader: Information Center. The Information Center does the following:

- starts SAS Data Loader in a new browser tab.
- provides a Settings window to configure the vApp connection to Hadoop.
- checks for available vApp software updates and installs vApp software updates.

All of the files that are accessed by the vApp reside in the shared folder. The shared folder is the only location on the user host that is accessed by the vApp. The shared folder contains the JDBC drivers needed to connect to external databases, and the Hadoop JAR files that were copied to the client from the Hadoop cluster.

When you create a job using a directive, the web application generates code that is then sent to the Hadoop cluster for execution. When the job is complete, the Hadoop cluster writes data to the target file and delivers log and status information to the vApp. Saved directives are stored in a database within the vApp.

The SAS In-Database Technologies for Hadoop software is deployed to each node in the Hadoop cluster. The in-database technologies consist of the following components:

- SAS Quality Knowledge Base for reference to data cleansing definitions.
- SAS Embedded Process software for code acceleration.
- SAS Data Quality Accelerator software supports SAS DS2 methods that pertain to data cleansing.
- SAS Data Management Accelerator for Spark enables you execute data integration and data quality tasks in Apache Spark on a Hadoop cluster.

How to Get Help for SAS Data Loader for Hadoop

SAS Data Management Support Community

If you need additional help with using SAS Data Loader for Hadoop, the SAS Data Management Community is a great place to find answers. Join the community to ask questions and receive expert online support.

Technical Support

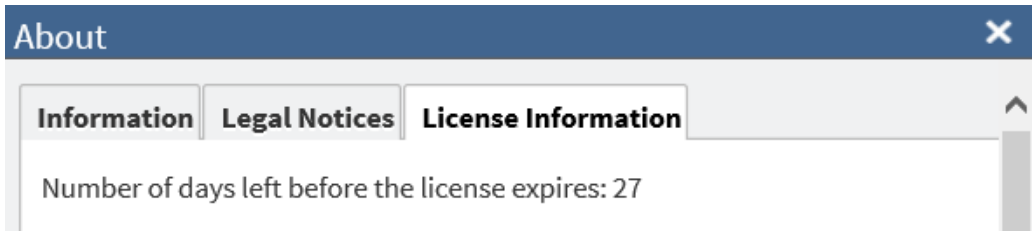
SAS provides customer support through self-help and assisted-help resources. See our Support page for more information about these resources.

Documentation and System Requirements

If you select **Help** from the Help icon at top right of most windows, the SAS Data Loader documentation page is displayed. From the SAS Data Loader for Hadoop documentation page, you can access information about accessibility features and keyboard shortcuts.

If you select **About SAS Data Loader** from the Help icon, you can display version information, supported browsers, legal notices, and license information.

In the About dialog box, the **License Information** tab shows how many days are left before your license expires. For information about renewing your license, see *SAS Data Loader for Hadoop: vApp Deployment Guide*.



The SAS Data Loader for Hadoop product page includes links to additional information, such as technical papers, training resources, and videos that show you how to perform common tasks.

For information about the system requirements, see System Requirements: SAS Data Loader 2.4 for Hadoop.

2

Getting Started

Prerequisites

Review the System Requirements

Because SAS Data Loader runs in a vApp, the computer that runs the vApp must meet the system requirements for the vApp client host environment. For example, you can use only browsers that are supported by the vApp.

Before you start using SAS Data Loader, review the system requirements for the vApp client host environment in System Requirements: SAS Data Loader 2.4 for Hadoop.

Perform Prerequisite Tasks

The following tasks must be completed before you can use SAS Data Loader:

- A Hadoop administrator installs SAS In-Database Technologies for Hadoop across the nodes of a Hadoop cluster. The administrator then provides the vApp installer with site-specific files and settings, including configuration files and JAR files required for the cluster. For MapR deployments, the

administrator provides an additional file that contains user and password information for accessing the MapR cluster. For more information about these tasks, see the "Administrator's Guide for SAS Data Loader for Hadoop" section of the *SAS In-Database Products: Administrator's Guide*.

- The vApp installer configures a hypervisor (such as VMware Player Pro) and the vApp for SAS Data Loader on each client host. The installer sets up the shared folder for the vApp and adds the Hadoop files that were provided by the Hadoop administrator to the shared folder. The installer also specifies a connection to the Hadoop cluster. SAS Data Loader will later use this connection to access the Hadoop cluster. For more information about these tasks, see *SAS Data Loader for Hadoop: vApp Deployment Guide*.

  To verify that these prerequisites have been met, open SAS Data Loader. For more information, see *SAS Data Loader for Hadoop: vApp Deployment Guide*.

First Tasks in SAS Data Loader

Here are some of the first tasks you can do in SAS Data Loader:

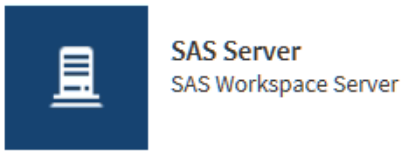
- Open SAS Data Loader. For more information, see *SAS Data Loader for Hadoop: vApp Deployment Guide*.
- Verify that you can connect to the Hadoop cluster. One way to do that is to browse tables on the Hadoop cluster. For more information, see "Browse Tables" on page 26.
- If you do not see the tables that you want to work with on the cluster, ask your Hadoop administrator if they should be there. To use the directives Copy Data to Hadoop and Copy Data from Hadoop, you must copy JDBC drivers from the Hadoop cluster to your shared folder. For more information, see Chapter 6, "Copy Data To and From Hadoop," on page 125.
- If your Hadoop cluster uses the Cloudera Impala SQL environment or the Apache Spark runtime target or both, then your SAS Data Loader directives can benefit from enhanced performance. To take advantage of these technologies, see "Enable Support for Impala and Spark" on page 12.
- If you want to work with SAS LASR Analytic Server, see "Load Data to LASR" on page 155.
- If you want to review the global options for SAS Data Loader, see "Set Global Options" on page 176.
- To review what you can do with this software, see "What Does It Help You Do?" on page 2.

Create and Execute a Job Using SAS Sample Data

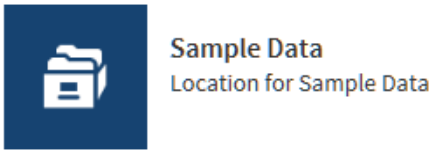
Follow these steps to copy a small SAS sample table into Hadoop and execute a transformation on that data.

1 From the SAS Data Loader directives page, and click the directive Copy Data to Hadoop.

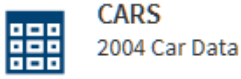
2 In the **Source Table** task, click the **SAS Server** data source.



3 Click **Sample Data**.



4 Click the CARS source table and click **Next**.

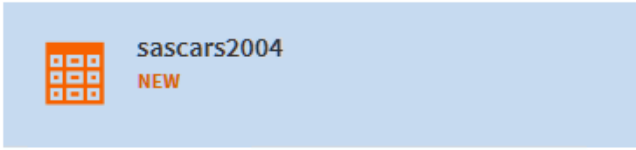


5 In the **Filter** task, click **Next** to include all SAS source rows in the Hadoop target table.

6 In the **Columns** task, click **Next** to accept the existing number and arrangement of columns.

7 In the **Target Table** task, click an appropriate location for the new table.

8 Click New Table... and enter a table name such as SASCars2004.



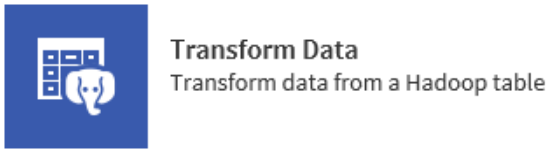
9 In the **Code** tab, browse the generated code, and then click **Next**.

10 In the **Result** task, click **Start copying data**.

11 Click **View Results** to see your new table in Hadoop.

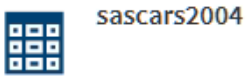
12 To transform your new table, click Back to Directives.

13 In the SAS Data Loader directives page, click **Transform Data**.

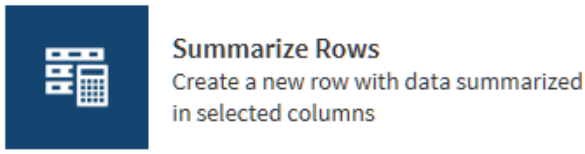


14 In the **Source Table** task, click the data source that you just used to store your new table.

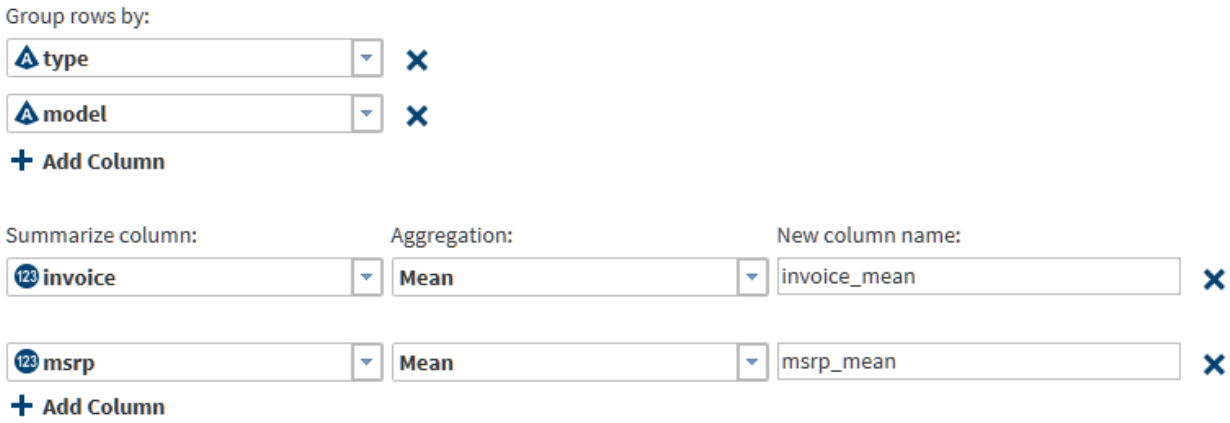
15 Click your new table, and then click **Next**.



16 In the **Transformations** task, select **Summarize Rows**.

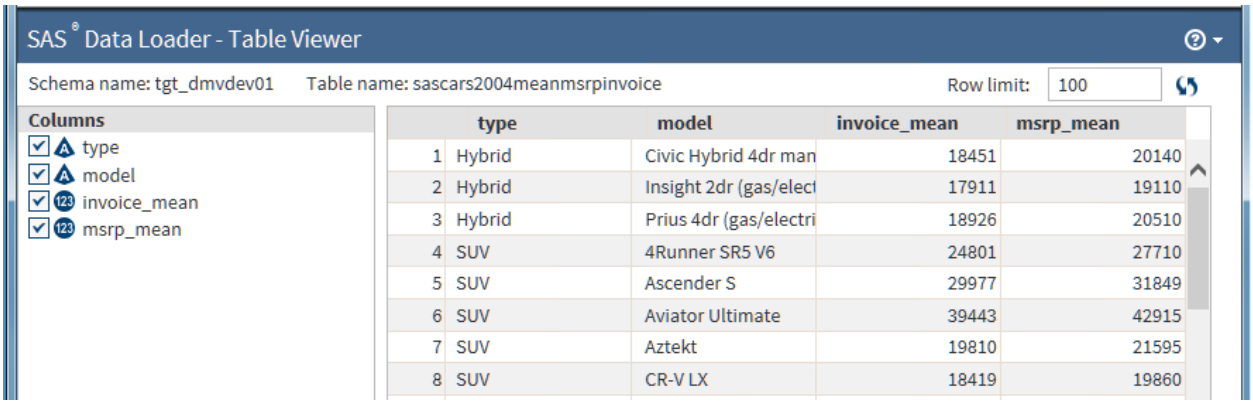


17 Select group-by rows, summaries and aggregations, and then click **Next**.



18 In the **Target Table** task, click the data source that you use for target data, click **New Table...**, enter a table name, and then click **Next**.

19 In the **Result** task, click **Start transforming data**. The job might run for a minute or so. Click **View Results** to see your first transformation in Hadoop using SAS Data Loader. Congratulations!



	type	model	invoice_mean	msrp_mean
1	Hybrid	Civic Hybrid 4dr man	18451	20140
2	Hybrid	Insight 2dr (gas/elect	17911	19110
3	Hybrid	Prius 4dr (gas/electri	18926	20510
4	SUV	4Runner SR5 V6	24801	27710
5	SUV	Ascender S	29977	31849
6	SUV	Aviator Ultimate	39443	42915
7	SUV	Aztekt	19810	21595
8	SUV	CR-V LX	18419	19860

Naming Requirements for Schemas, Tables, and Columns

Follow these requirements to name schemas, tables, and columns:

- Use alphanumeric characters in the names of schemas, tables, and columns.
  - Use underscore characters, but do not use an underscore character as the leading character in the name.
  - Do not use double-byte character sets in names.
  - Do not use Hive quoted identifiers (‘) in column names.
- Limit the length of the names of schemas, tables, and columns to 32 characters. This limit is required in the following directives and transformations:
  - Profile directive
  - Transpose directive
  - Summarize Rows transformation (a task in multiple directives)
- Do not use a DS2 reserved keyword for the name of a column that is the target of any directive that is DS2 based. Using a DS2 reserved keyword for the name of a column that is the target of any DS2-based directive can result in a runtime error.

  These DS2-based directives can be affected:
  - Match-Merge data directive
  - Transform Data directive
  - Transpose Data directive
  - Cleanse Data directive (except the Summarize Rows transformation)
  - Run a SAS program

  For example, if a source column named OTHER is transposed in a Transpose Data directive, a runtime error is generated. OTHER is a DS2 reserved keyword.

  For more information about DS2 keywords, see *SAS 9.4 DS2 Language Reference*.
- For column names, avoid using words that are reserved keywords for your DBMS.

  These directives can be affected:
  - Sort Data directive
  - Query or Join Data directive

  For example, DATE and DATABASE are SQL reserved keywords. If a Sort Data directive has a target table with a column called DATE, the sort fails. For more information about DBMS keywords, see the user’s guide for your DBMS.

TIP If an individual directive has any additional naming or other usage requirements, those requirements are documented in a separate “Usage Notes” section for that directive.

Enable Support for Impala and Spark

Introduction

If your Hadoop cluster uses the Cloudera Impala SQL environment or the Apache Spark runtime target or both, then your SAS Data Loader directives can benefit from enhanced performance. Cloudera Impala SQL and Apache Spark enhance performance using distributed processes and an increased level of in-memory processing.

The following directives support Cloudera Impala SQL:

- Query or Join Data
- Sort and De-Duplicate Data
- Run a Hadoop SQL Program

The following directives support Apache Spark:

- Cleanse Data
- Transform Data
- Cluster-Survive Data (requires Spark)

Note: The only directive that requires Impala or Spark is Cluster-Survive Data, which requires Spark.

Support for Impala and Spark is seen primarily in the code that is generated by the supporting directives. The directives change only in their support of user-written expressions.

User-written expressions can be used to filter source rows from the target or to calculate values for new target columns. When you enable Impala and Spark, you change the functions that can appear in your user-written expressions. In Impala, Impala SQL functions are supported rather than HiveQL functions. In Spark, DataFlux EEL functions are supported rather than SAS DS2 functions. (EEL stands for the Expression Engine Language, and DS2 stands for DATA Step 2.) All supported functions are documented in the Advanced Editor and in syntax reference documents.

When Impala and Spark are enabled, you retain the ability to write and execute new and existing directives in Hive. Continued Hive support is provided because Impala and Spark run in coordination with Hive. Any existing directives that use Hive will continue to run as they have in the past.

For information about the supported versions of Impala and Spark, see *SAS 9.4 Supported Hadoop Distributions*, at https://support.sas.com/resources/thirdpartysupport/v94/hadoop/hadoop-distributions.html.

Prerequisites

Meet the following prerequisites before you enable Impala or Spark:

- Impala and Spark must be fully operational on your Hadoop cluster.
- The Spark features in SAS Data Loader require the installation of the SAS Data Management Accelerator for Spark on your Hadoop cluster. Hadoop administrators can refer to the "SAS Data Management Accelerator for Spark" chapter in the SAS 9.4 In-Database Products: Administrator's Guide.
- The Spark features in SAS Data Loader require HCatalog to be enabled on the Hadoop cluster in order to read data from Hive. Hadoop administrators can refer to the topic "Additional Configuration Needed to Use HCatalog File Format" in the SAS 9.4 In-Database Products: Administrator's Guide.
- For Impala, Cloudera recommends that you install the Cloudera Impala JDBC Driver on your client host. See "In an Environment that Does Not Use Kerberos, Install and Use a Cloudera Impala JDBC Driver".

Enable Impala as the Default SQL Environment

Follow these steps to enable the Impala SQL environment for new instances of the directives that support Impala.

1. Confirm with your administrator that Impala has been installed and configured on the nodes of your Hadoop cluster.
2. Click **More**  and select **Configuration**.
3. In the **Hadoop Configuration** panel of the **Configuration** window, confirm that the **Host** field under **Impala server** contains a server name. If this field is blank, enter the fully qualified network name of the Impala host. Contact your Hadoop Administrator as needed.
4. In the **Port** field under **Impala server**, confirm that a value is specified by default. If this field is blank, obtain the port number for the Impala server from your Hadoop Administrator and enter that number in the field.
5. In the **SQL environment** field, select **Impala**.
6. To confirm your entries, click **Test Connection**.
7. Click **OK** to activate Impala as the default SQL environment.

To restore Hive as the default SQL environment, select **Hive** in the preceding Step 4.

Enable Spark as the Default Runtime Target

Follow these steps to enable Spark as the default runtime target. The default runtime target is applied to all new instances of the directives that support Spark.

Note: The default Hive runtime target does not apply to the directive Cluster-Survive Data, which requires Spark.

1. Confirm with your administrator that Apache Spark has been installed and configured on the nodes of your Hadoop cluster.
2. Click **More**  and select **Configuration**.
3. Click **Preferred runtime target** and select **Spark**.
4. Click **OK**.

To restore MapReduce as the default runtime target, click **MapReduce** in the preceding Step 3.

Override the Impala or Spark Default

In an individual directive, you can override the default setting for the SQL environment or the runtime target. Use the **Settings** menu at the top of the directive to specify an override, or to return to the default setting that is specified in the **Configuration** window.

About Saved Directives and Impala or Spark

Saved directives that were created prior to SAS Data Loader 2.4 for Hadoop continue to run in Hive after you enable Impala and Spark. To run these directives in Impala or Spark, you need to create new directives.

Saved directives that were created in SAS Data Loader 2.4 for Hadoop or later for the HiveQL environment, can be upgraded to use Impala or Spark. To upgrade, follow these steps:

1. Open the saved directive.
2. Click **Settings** and select **Impala** or **Spark**.
3. Replace any user-written expressions in the **Filter** and **Manage Columns** tasks. Replace the existing Hive functions with Impala SQL or the existing SAS DS2 functions with DataFlux EEL functions, as provided in the Advanced Editor.
4. Save and close the directive.

The next time the saved directive is run, new code will be generated for the selected environment.

Usage Notes for Spark

Introduction

As you create and run the directives that support Spark, keep the following subjects in mind.

String Truncation in Spark-Enabled Directives

In directives where Spark is not the preferred runtime target, character columns are truncated based on the value of the field **Maximum length for SAS columns**. This field is available in the **General Preferences** panel of the Configuration window. The default value is 1024 characters. Source columns

with string data types such as VAR and VARCHAR are truncated in SAS when their length exceeds the specified limit. The truncation occurs when SAS reads source columns into memory.

In Spark-enabled directives, the truncation of string columns differs between source columns that return a length, and source columns that do not return a length. Hive releases prior to 0.14.0 do not return a length for VAR and VARCHAR columns.

When Spark is enabled, and when columns **do** return a string length, strings are truncated according to the value of the configuration option EXPRESS_MAX_STRING_LENGTH. The value of the **Maximum length for SAS columns** field is ignored.

When Spark is enabled, and when string columns **do not** return a length, strings are truncated differently. The maximum string length is determined by the lesser value of the configuration option EXPRESS_MAX_STRING_LENGTH or the field **Maximum length for SAS columns**.

The default value of the EXPRESS_MAX_STRING_LENGTH configuration option is 5 MB. To specify a different value, ask your Hadoop administrator to update the app.cfg file on each node that runs the SAS Data Management Accelerator for Spark. In those files, add or update the label/value pair for EXPRESS_MAX_STRING_LENGTH.

Note: The value of EXPRESS_MAX_STRING_LENGTH also specifies the maximum amount of memory that is allocated for the underlying expression. For this reason, Hadoop administrators should be judicious when changing the default value.

VAR and VARCHAR columns that do not return a length are converted to the STRING type in the target so that they can receive a default length. To retain the original column types, use the Manage Columns task in the directive. In Manage Columns, the type of the target column needs to be VAR or VARCHAR and a length specification is required.

Spark Date Error

When Spark is the runtime environment, and if you run a Hive release earlier than 1.2, then dates starting with January 1, 1970 and older might be incorrect. To learn more, see https://issues.apache.org/jira/browse/HIVE-10178.

Hive Views Cannot Be Source Tables

When Spark is the preferred runtime environment, Hive views cannot be used as source tables.

Parquet Cannot Be Specified as a Format for Target Tables

When Spark is the preferred runtime environment, the Parquet table format cannot be selected for target tables. Parquet source tables are supported.

Spark Bin Directory Required in the Hadoop PATH

Spark support requires the addition of the Spark bin directory to the PATH environment variable in each Hadoop node. If your Spark-enabled directives fail early, contact your Hadoop administrator to research this issue.

Most Hadoop distributions include the Spark bin directory in /usr/bin, which resolves the issue. Your configuration might differ.

In the MapR distribution of Hadoop, the Spark bin directory is not included in the PATH variable by default. To resolve the issue in MapR, your Hadoop administrator can add a line to yarn-env.sh on each node manager node. The following example illustrates a typical addition to yarn-env.sh:

```
/* In MapR 5.0, using Spark 1.3.1 */
export PATH=$PATH:/opt/mapr/spark/spark-1.3.1/bin
```

Usage Notes for Impala

Introduction

To create and run the directives that support Impala, install a JDBC driver and avoid metadata synchronizaton errors between Impala and Hive.

To Use Impala in a Kerberos Environment, Use the Hive JDBC Driver

If your site uses Cloudera Impala and Kerberos authentication, then use the Hive JDBC driver that is currently in use on your Hadoop cluster. The Hive JDBC driver is normally installed during the deployment of the vApp, as described in the *SAS Data Loader for Hadoop: vApp Deployment Guide*.

Do not install any of the available Cloudera Impala JDBC drivers in ***vApp-path*\SASWorkspace\JDBCdrivers**.

In an Environment that Does Not Use Kerberos, Install and Use a Cloudera Impala JDBC Driver

To use Impala in environments that do not use Kerberos, Cloudera recommends that you install a Cloudera Impala JDBC driver.

By default, if any of the following drivers are found in ***vApp-path*\SASWorkspace\JDBCdrivers**, then SAS Data Loader uses the latest:

- com.cloudera.impala.jdbc3.Driver
- com.cloudera.impala.jdbc4.Driver
- com.cloudera.impala.jdbc41.Driver

To override the default behavior and specify a Cloudera Impala JDBC driver, follow these steps:

1 In a text editor, open or create the file **data-loader-site.xml** in the folder ***vApp-path*\SASWorkspace\conf**.

Enter the following text:

```
<?xml version="1.0" encoding="UTF-8"?>

<configuration>

    <property>

        <name>cloudera.impala.driver.class</name>

        <value>com.cloudera.impala.jdbc4.Driver</value>

    </property>
```

```
</configuration>
```

2 As needed, replace the contents of the `value` tag above with the name of your JDBC driver class.

3 Save the XML file and restart the vApp to enable the use of your required JDBC driver.

Information about the Cloudera Impala JDBC drivers is provided at http://www.cloudera.com/content/cloudera/en/documentation/cloudera-impala/latest/topics/impala_jdbc.html. The download page for the JDBC drivers is http://www.cloudera.com/content/cloudera/en/downloads/connectors/impala/jdbc/impala-jdbc-v2-5-24.html.

Avoid Metadata Errors between Impala and Hive

To avoid metadata errors, avoid using a table in Hive and then using that same table soon thereafter in Impala (or in Impala and then in Hive). One place that could generate synchronization errors is a serial chain directive. One directive can use a table as a target and the next directive can use the same table as a source.

Overview of the vApp for SAS Data Loader

The SAS Data Loader for Hadoop web application runs inside a virtual machine or vApp. The vApp is started and managed by a hypervisor application. SAS Data Loader runs with a number of available third-party hypervisors. One of the available hypervisors is called VMware Player Pro.

The web application in the vApp communicates with SAS software on the Hadoop cluster to manage data within Hadoop.

The topics in this section review basic tasks, such as starting and stopping the vApp.

The *SAS Data Loader for Hadoop: vApp Deployment Guide* is a complete reference to tasks that can be performed in the vApp, such as the following:

- Configure the vApp and SAS Data Loader after installing your SAS software.
- Migrate from previous releases of SAS Data Loader for Hadoop.
- Troubleshoot the vApp start process.
- Update your vApp software.
- If the version of Hadoop on your cluster has changed, change the version of Hadoop that is specified in the vApp to match.
- If the security settings for your cluster have changed, change the security settings in the vApp to match.
- Enable logging inside the vApp.
- Manage your SAS license.

You can access *SAS Data Loader for Hadoop: vApp Deployment Guide* from the SAS Data Loader documentation page.

3

About the Directive Interface

Using the Directives Page

In the top-level web page for SAS Data Loader, you can browse and select directives. You can also select the following menus and icons:

Configuration 
opens the Configuration window, with separate panels for configuring Hadoop connections, external database connections, SAS LASR Analytic Server connections, and several categories of user preferences. Some of these configurations are set during installation. You can also add a new database connection or add a connection to an instance of the SAS LASR Analytic Server software. For more information, see “Set Global Options” on page 176.

Back Up Directives 
performs a backup of your saved directives. For more information about this option, see “Back Up Directives” on page 175.

Help 
displays the SAS Data Loader documentation page on the SAS support website. Also displays version information, supported browsers, legal notices, and license information.

Viewing Data Sources and Tables

Overview

For most directives in SAS Data Loader, data sources are Hive schemas that contain one or more tables. Data sources are defined in Hive by your Hadoop administrator. If you do not see the data source or table that you need, contact your Hadoop administrator. If needed, the administrator can add a new Hive schema and set appropriate user permissions to read and write data.

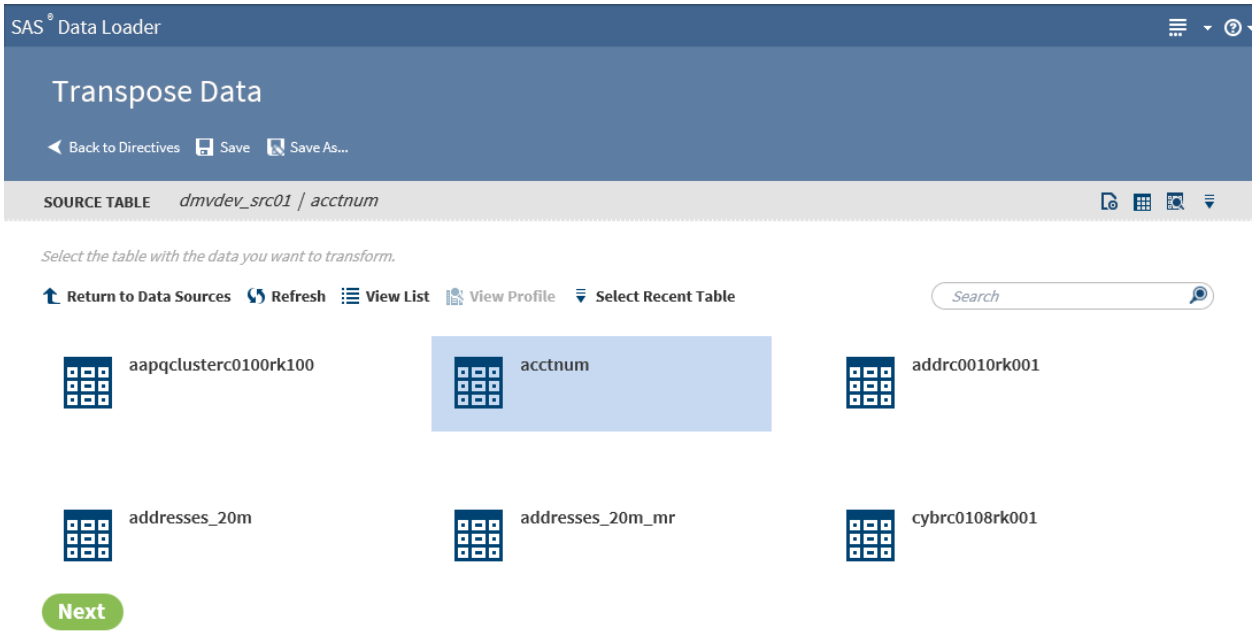
In some cases, data sources are not based on Hive schemas. For example, data sources for the Copy Data to Hadoop directive are RDBMS connections. Data sources for the Import a File directive are delimited files that are stored in the shared folder of the vApp.

When you open a directive to create a job that runs in Hadoop, you select a data source and a source table that is contained within that data source. If the directive produces output tables, you then select a data source and a target table at the end of the directive.

To protect your data, target tables do not overwrite source tables. Target tables are not required to be new tables each time you run your job. You can overwrite target tables that you created in previous job runs.

As the data is processed in each task in the job, you can view a sample on page 23 of the data that is produced in each task.

A typical Source Table task includes a graphical view of the tables in the selected data source.



SAS Table Viewer icon

Click to open the selected table in the SAS Table Viewer, which provides column information and sample data for the table.

View Data Sample icon

Click to display the first 100 rows of source data, as that data has been transformed up to that point in the job.

View List and **View Grid**

Click the View List icon to display data sources or tables as a list. When you view tables, the list format displays the table name and description, along with the dates on which the table was last profiled and last modified.

Note: The last modified date is displayed only when the **Identify each table as "new" when created or modified** setting is selected on the **General Preferences** panel of the Configuration window. For more information, see "General Preferences Panel" on page 179.

Click the View Grid icon to display data sources or tables in a grid.

View Profile

Click to view profile information for the selected table. If a profile exists for a table, PROFILED appears beneath the table name.

Return to Data Sources

Click to select a source table from another data source.

Select Recent Table

Click to choose from a list of recently used tables. If you select a table from a different data source, the source table information is adjusted accordingly. The table that you selected is automatically highlighted.

Enter text in the search field to filter the list of data sources or tables. The search feature filters according to name when applied to data sources and according to name and description when applied to tables.

Click to return to the top of the page when viewing a long list of data sources or tables.

TIP If you frequently work with the same data source across multiple directives, you can have SAS Data Loader select the most recently used schema automatically. This can help you select source tables and target tables more quickly. To enable this feature, click  , select **Configuration**, and complete the following steps:

1. Click **General Preferences**.
2. Select **Automatically select the most recently selected hive schema**.

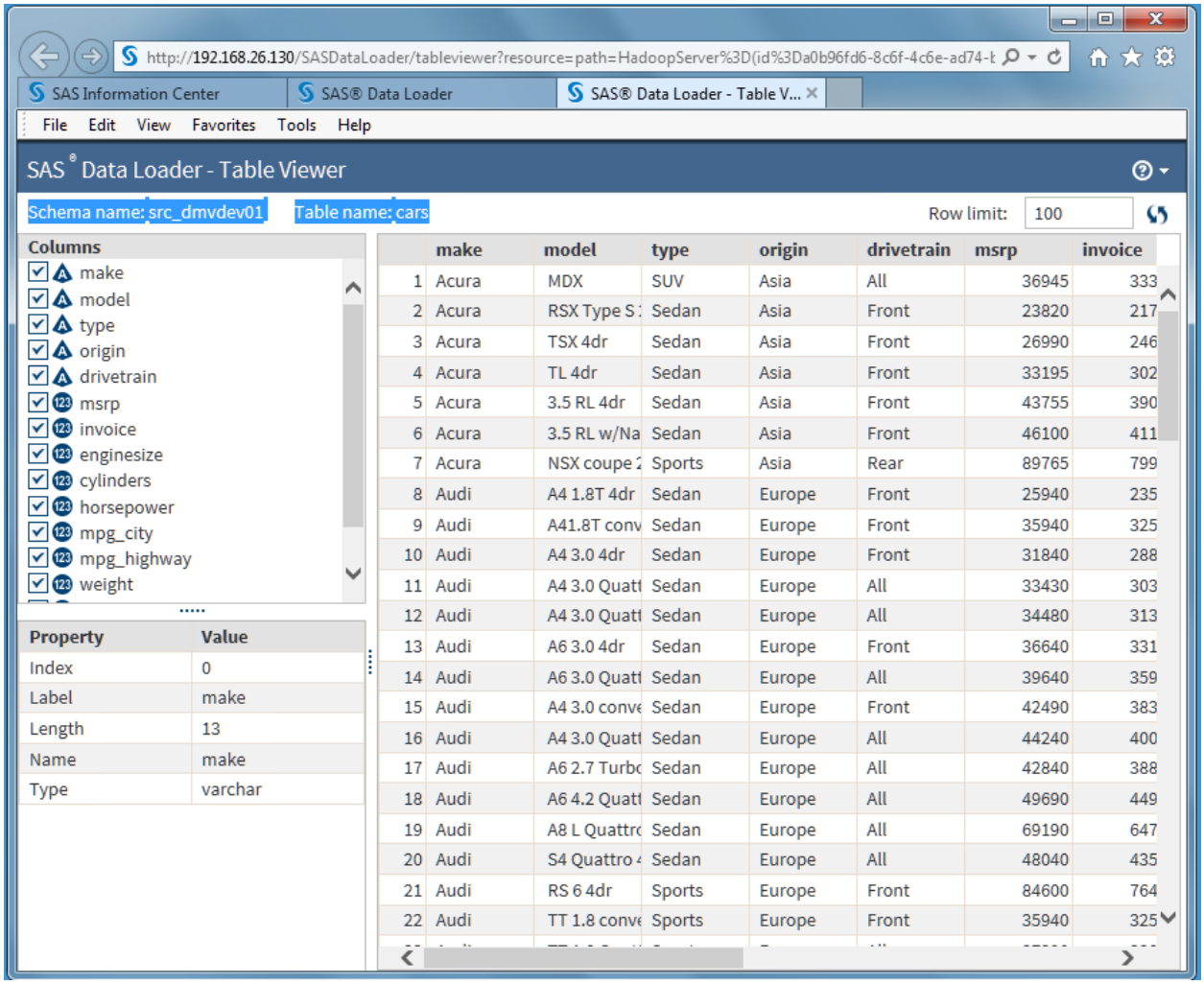
About the SAS Table Viewer

How It Works

The SAS Table Viewer displays sample data and column information for a selected table. The viewer is available when you select source or target tables or

when you view results or status. The SAS Table Viewer opens in a separate tab in the browser, so you can continue to reference that information while working with directives.

To open the viewer, click the **Open the selected table in the table viewer** icon .



In the viewer, you can click a column name to display the properties of that column. You can also clear the check box next to the column name to temporarily remove that column from the sample data view.

To change the number of sample rows that are displayed, change the value of the **Row Limit** field.

To refresh the sample data after a directive has operated on that table, click the **Refresh** icon .

Column properties are defined as follows:

Index
: Column number.

Label
: A shortened version of the column name that can be added to the data values for that column. If a label is not assigned, then the column name is used as the label.

Length
: The size of the table cell (or variable value) in bytes.

Name
: Column name.

Type
: The type of the data in the column.

For information about data types and data conversions in SAS and Hadoop, see the chapter SAS/ACCESS Interface to Hadoop in the document *SAS/ACCESS Interface to Relational Databases: Reference*.

Usage Notes

- When viewing a SQL Server table, the following numeric data types are displayed in the **Columns** list with a character data type: datetime (datetime_col), money (money_col), smallmoney (smallmoney_col), numeric (numeric_col), and real (real_col).
- Viewing the source and target tables of transformations can show differences in decimal values. The source columns show no decimal values, and the target shows full double-precision values. This difference exists in the display only. In the Hadoop file system HDFS, the values are the same.

About the Sample Table Viewer

In directives that list tables for selection, you can click the **View a data sample** icon  to display a subset of the source data, as that data has been transformed up to that point in the job. This gives you a preview of your data before you run your job against the full source table in Hadoop.

In the data sample, you can click **Refresh** to display the latest data or click **X** to close the data sample.

Working with the Code Editor

You can edit and save changes to the code that is generated by directives. There are two ways to access code:

- by using the code editor from the **Code** task within a directive

  Note: Some directives, such as Transform Data and Cleanse Data, do not include a **Code** task.

- by downloading the code from a directive's **Result** task or from the Run Status directive. After downloading the code, you can work with it in a third-party text editor on your local machine.

The code editor is intended to be used only to implement advanced features. In normal use, there is no need to edit code. The code editor is a good way to see what will be running, but making changes can be problematic. If you make changes in the directive interface after you edit code, then your edits are lost when the code is regenerated. Also, your code edits are not reflected in the directive interface, which further complicates updates to edited code.

In addition to the code editor, SAS Data Loader provides two directives for user-written code. For more information, see Chapter 7, "Run User-Written Programs," on page 157.

4

Manage Data

Overview of Data Management Directives

The data management directives support combinations of queries, summarizations, joins, transformations, sorts, filters, column management, and de-duplication. Data quality transformations include standardization, parsing, match code generation, and identification analysis, combined with available filtering and column management to reduce the size of target tables.

In addition to these directives, the **Delete Table** action is a feature that enables you to delete a table from the data source.

Browse Tables

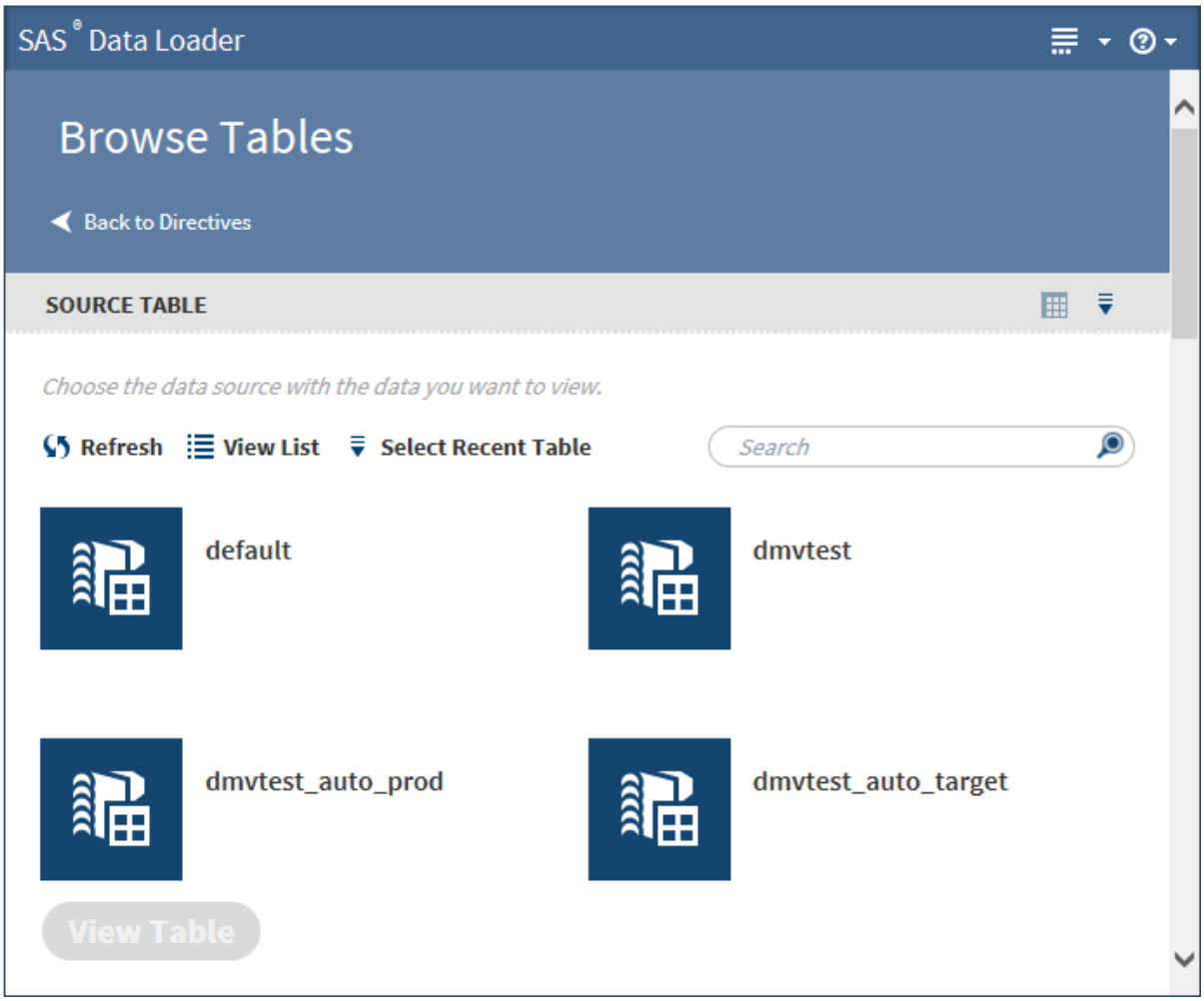
Introduction

Use the Browse Tables directive to browse a list of the tables in a data source that is available on the Hadoop cluster. You can also view the contents of a table in the Table Viewer. With the Browse Tables directive, you can examine the data on quickly and conveniently before you begin working with the data in other directives.

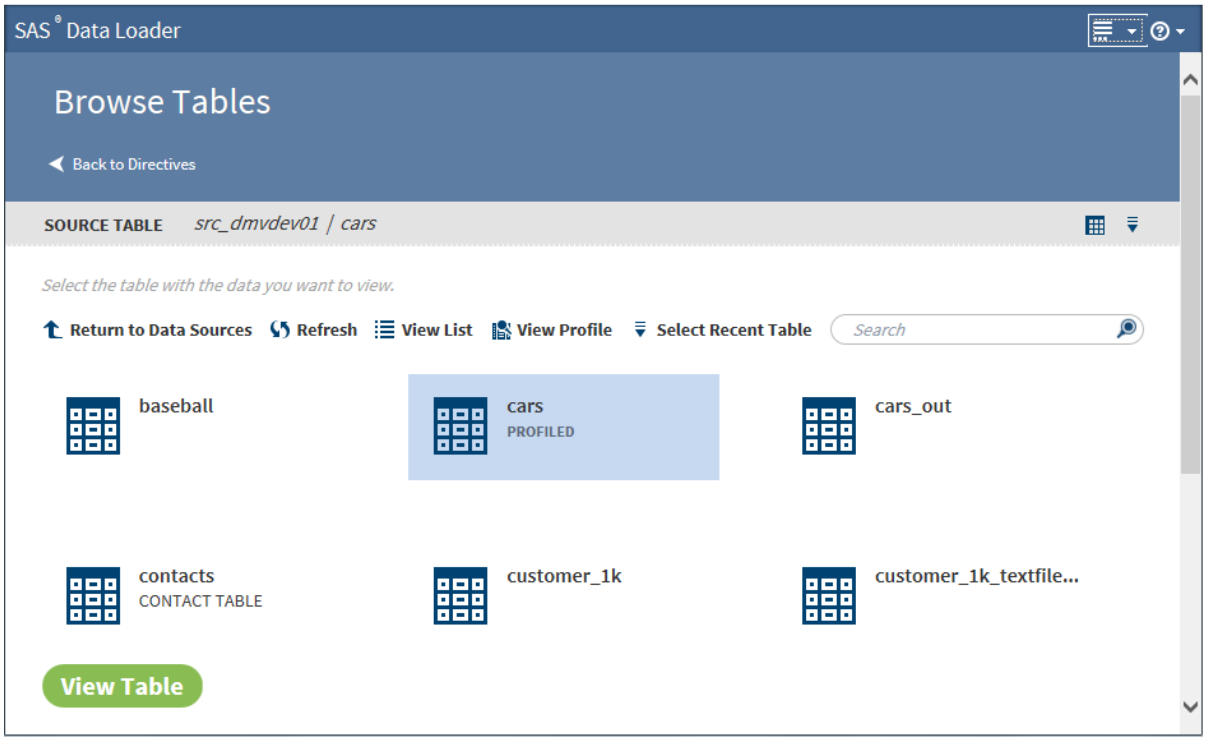
Example

Follow these steps to view the data in a table:

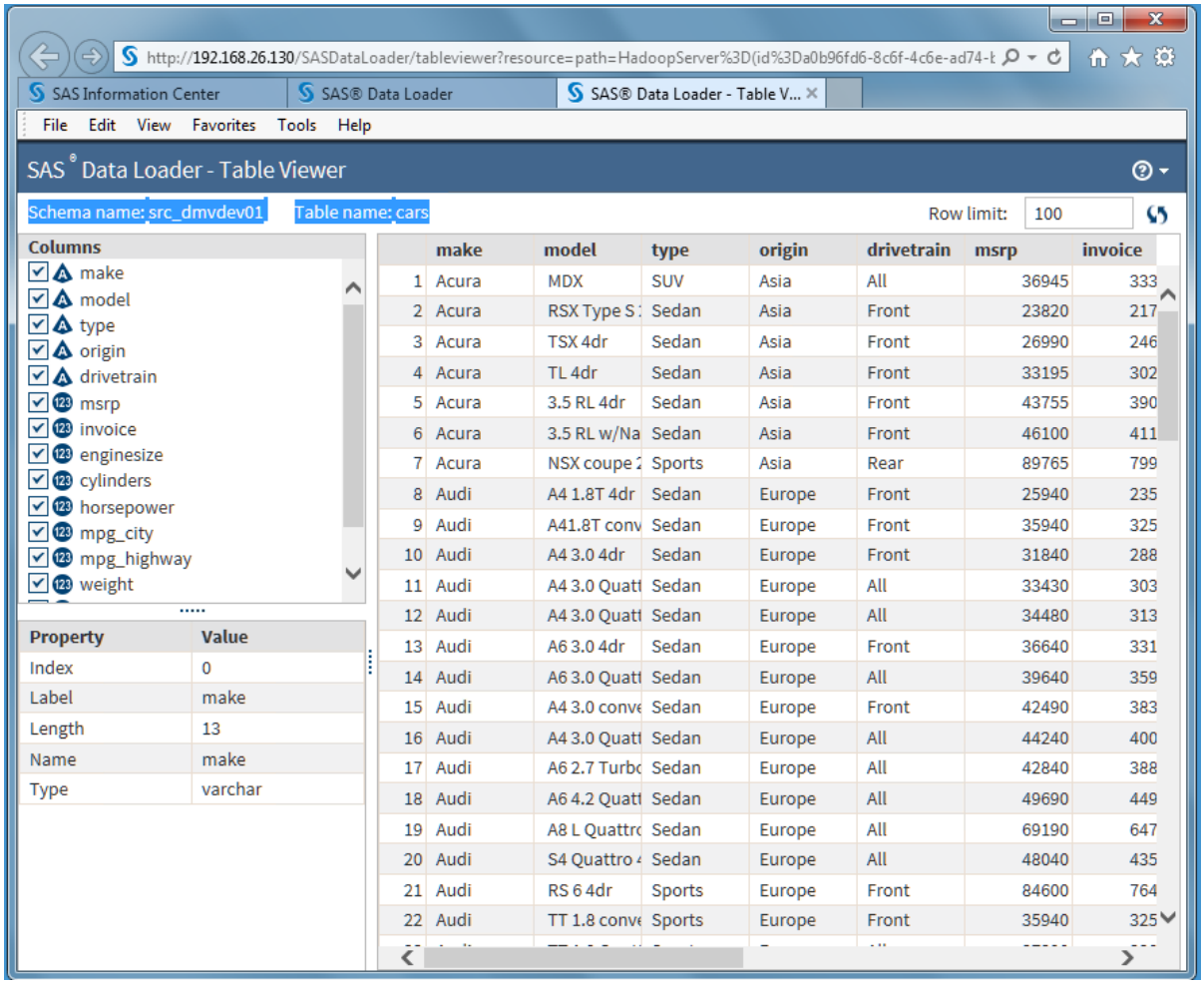
1 On the SAS Data Loader directives page, click **Browse Tables**. The **Source Table** task is displayed. For more information about data sources and tables, see “Viewing Data Sources and Tables” on page 20.



2 Click a data source to display its tables, and select the table that you want to view.



3 Click **View Table**.



	make	model	type	origin	drivetrain	msrp	invoice
1	Acura	MDX	SUV	Asia	All	36945	333
2	Acura	RSX Type S	Sedan	Asia	Front	23820	217
3	Acura	TSX 4dr	Sedan	Asia	Front	26990	246
4	Acura	TL 4dr	Sedan	Asia	Front	33195	302
5	Acura	3.5 RL 4dr	Sedan	Asia	Front	43755	390
6	Acura	3.5 RL w/Na	Sedan	Asia	Front	46100	411
7	Acura	NSX coupe 2	Sports	Asia	Rear	89765	799
8	Audi	A4 1.8T 4dr	Sedan	Europe	Front	25940	235
9	Audi	A41.8T conv	Sedan	Europe	Front	35940	325
10	Audi	A4 3.0 4dr	Sedan	Europe	Front	31840	288
11	Audi	A4 3.0 Quatt	Sedan	Europe	All	33430	303
12	Audi	A4 3.0 Quatt	Sedan	Europe	All	34480	313
13	Audi	A6 3.0 4dr	Sedan	Europe	Front	36640	331
14	Audi	A6 3.0 Quatt	Sedan	Europe	All	39640	359
15	Audi	A4 3.0 conve	Sedan	Europe	Front	42490	383
16	Audi	A4 3.0 Quatt	Sedan	Europe	All	44240	400
17	Audi	A6 2.7 Turbo	Sedan	Europe	All	42840	388
18	Audi	A6 4.2 Quatt	Sedan	Europe	All	49690	449
19	Audi	A8 L Quattro	Sedan	Europe	All	69190	647
20	Audi	S4 Quattro 4	Sedan	Europe	All	48040	435
21	Audi	RS 6 4dr	Sports	Europe	Front	84600	764
22	Audi	TT 1.8 conve	Sports	Europe	Front	35940	325

TIP Because the SAS Table Viewer appears in a separate browser tab, you can view the contents of multiple tables at the same time. For each additional table, just return to the Browse Tables directive in the **SAS Data Loader** tab and repeat the previous steps.

Delete Tables

Introduction

Unlike other features described in this chapter, the **Delete Table** action is not a data management directive. It is an action that is available in many directives.

The **Delete Table** action is available on the **Action** menu  for any directive that has a **Source Table** task. You can select one source table at a time in the data source (in either the grid view or the list view) and use the **Delete Table** action to delete the selected table.

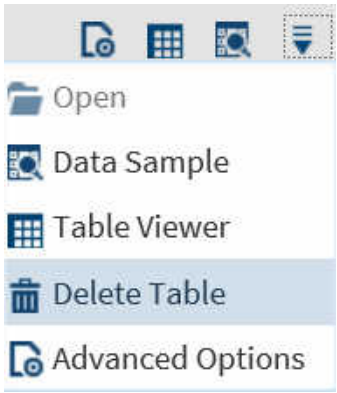
Deleting a Table

To delete a table:

1 On a **Source Table** task, select the table that you want to delete.

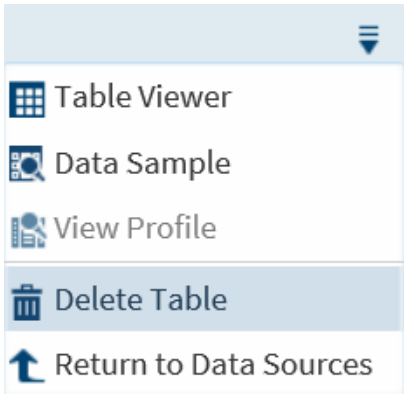
2 Access the **Action** menu in one of two ways:

- In either the grid view or the list view, click the **Action** menu  on the menu bar and select the **Delete Table** action.



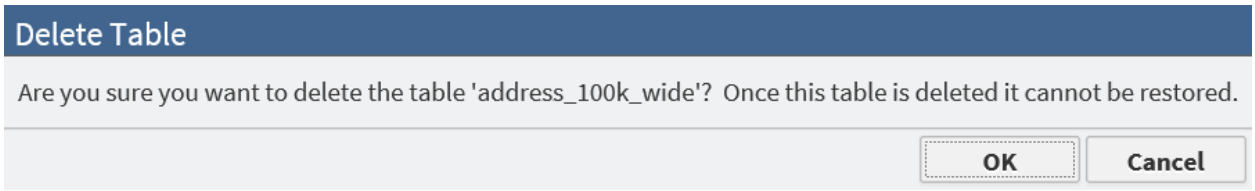
- In the list view only, click the **Action** menu at the end of the **Description** cell for the selected table.

Select the **Delete Table** action.



Note: Although you can select multiple tables in the list view, you can delete only one table at a time.

3 When the confirmation dialog box appears, click **Cancel** to cancel the deletion.



CAUTION! You cannot undo a table deletion.

Click **OK** to confirm the deletion of the table.

Usage Notes

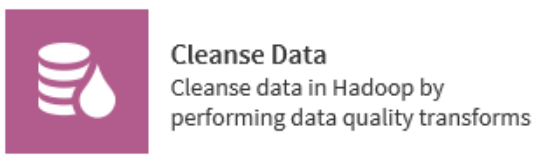
If you attempt to delete a table in HDFS, and you do not have the appropriate privileges in HDFS to complete that action, the Hive metadata about the table is deleted, but the table might remain after the delete.

Cloudera 5.4 does not support the **Delete Table** action.

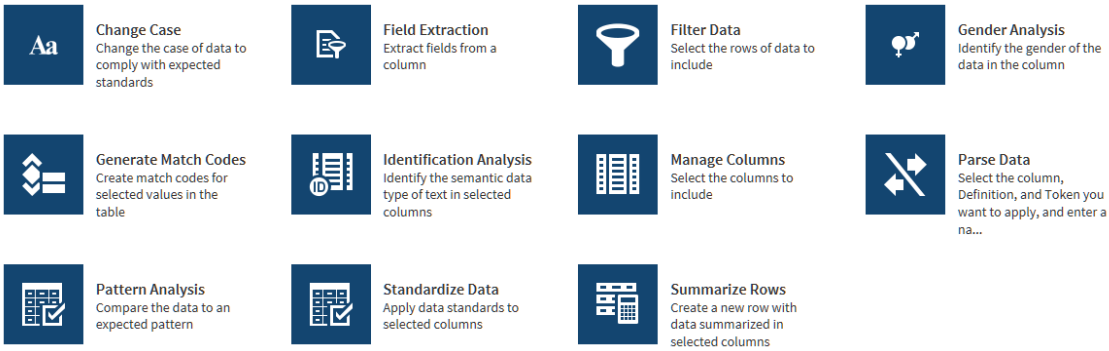
You can use the **Delete Table** action to delete a view as well. However, if you delete a view, the underlying table that contains the data in the view is not deleted. If you want to delete a table, you must explicitly delete it using the **Delete Table** action.

Cleanse Data

Introduction



Use the Cleanse Data directive to create jobs that improve the quality of your Hadoop data. Your jobs can combine any of the data quality transformations in any order. When you run your job, the transformations will be executed in the order in which you defined them.



About Locales, Definitions, and the Quality Knowledge Base

Most of the data quality transformations ask you to select a source column, a locale, and a definition. A *locale* represents a distinct alphabetical language, combined with a specified regional usage of that language. For example, the English, United States locale applies only to that region. The locale English, England addresses different usage or data content for the same alphabetic language.

A locale consists of a collection of *definitions*. Definitions tell SAS how to cleanse data. For example, the Street Address definition for the English, United States locale describes the structure of the first part of an American mailing address. In the locale Spanish, Mexico, the Street Address definition accommodates differences in mailing address structure as well as the differences in language and alphabet.

Locales and definitions make up a *SAS Quality Knowledge Base*. A Quality Knowledge Base is deployed on your Hadoop cluster. When you run a data cleansing job in Hadoop, the SAS software on your cluster accesses the Quality Knowledge Base to transform your data.

In SAS Data Loader you specify a default locale, which should match the typical locale of your source data. The default locale is selected in the **QKB** panel of the Configuration window, as described in “QKB Panel” on page 189. You can override the default locale in any of the data quality transformations. The override applies only to the current transformation.

To learn more about the Quality Knowledge Base, refer to the related document titles in “Recommended Reading” on page 201.

To learn about the output that is generated by a given definition, refer to the online Help for the SAS Quality Knowledge Base, in the topic Global Definitions.

Enable the Spark Runtime Target

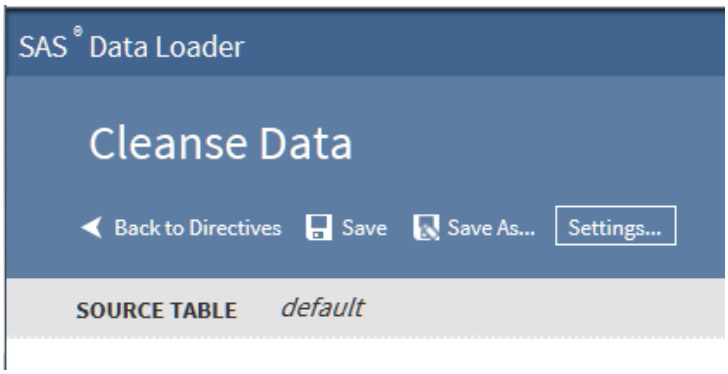
Support for the Apache Spark runtime target is enabled in the **Hadoop Configuration** panel of the **Configuration** window. When Spark is enabled, new instances of the following directives use Spark by default:

- Cleanse Data
- Transform Data
- Cluster-Survive Data

The default runtime target can be overridden using the **Settings** menu. To learn more about runtime targets, see "Enable Support for Impala and Spark".

Note: Changing the default runtime target does not change the runtime target for saved directives. Saved directives continue to run with their existing runtime target unless they are opened, reconfigured, and saved.

Note: Enabling Spark changes the truncation of character columns, as described in the "Usage Notes for Spark".



Select a Source Table

When you use a data cleansing directive to create and run a job, you begin by selecting a source table.

Follow these steps to select a source table:

1. Scroll through the list or grid of data sources or schemas, and then click the data source (also known as a schema) that contains your source table. Or you can click **Select a Recent Table** and quickly choose from that list.
2. If you opened a data source, click the source table and then click **Next**.

   Note: To explore the contents of source tables, click a table and click **Data Sample**, **Table Viewer**, or (if available) View Profile.

   Note: To override the default maximum length of SAS character columns, click a source table and click **Edit Advanced Options**. If your directive uses the Spark runtime target (click **Settings** to check), then see "String Truncation in Spark-Enabled Directives".
3. In the **Transformation** task, click a data cleansing transformation to begin building your job.

Select a Data Cleansing Transformation

In a new job, after you select a source table, you click a data cleansing transformation. Click below to find usage information for your selected transformation:

Filter Data Transformation

Use the Filter Data transformation at the beginning of a job to decrease the number of rows that will be processed in subsequent transformations. The filter is specified as a user-written expression. The expression uses SAS DS2 functions and the MapReduce runtime environment, or DataFlux Expression Engine Language functions (EEL functions) and the Spark runtime environment. For a given source row, if the filter evaluates to true, then the row is included in the target table.

Follow these steps to use the Filter Data transformation:

1. If this is the first transformation in a new job, select a source table.
2. In the **Transformation** task, click **Filter Data**.

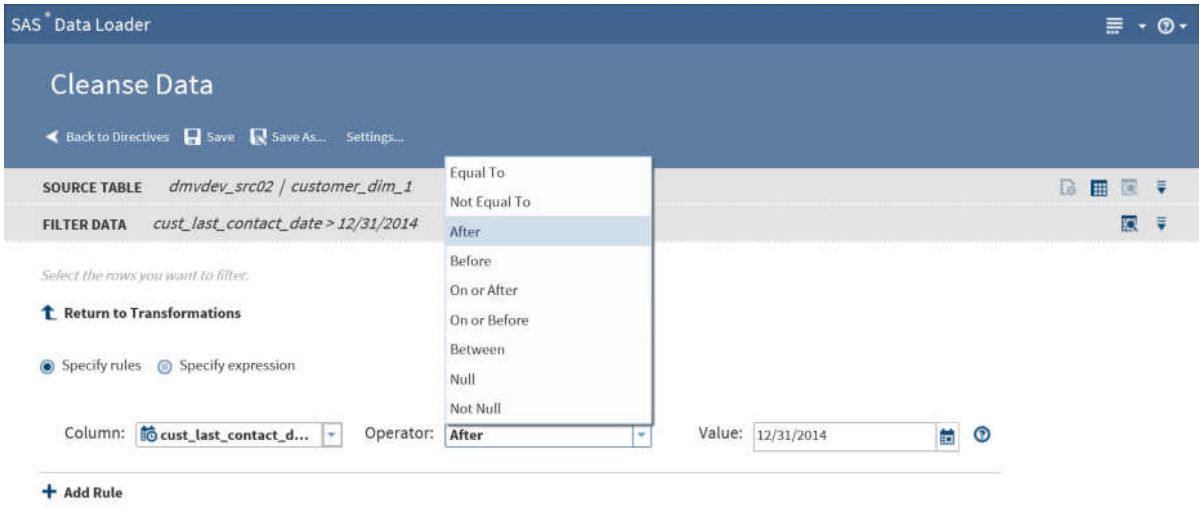
3. In the **Filter Data** transformation, choose one of the following:
   a. To filter rows using one or more rules, click **Specify rules** and proceed to the next step. You can specify multiple rules and apply them using logical AND and OR operators.
   b. To filter rows with a user-written expression, click **Specify expression** and go to Step 5.

TIP If the table that you selected has been profiled, an ellipsis button (...) appears next to the filter value selection. Click that button to view profile results while building your filters. For more information about generating profile reports for tables, see “Profile Data” on page 111.

4. To filter rows by specifying one or more rules, follow these steps:

a Click **Select a column** and choose the source column that forms the basis of your rule.

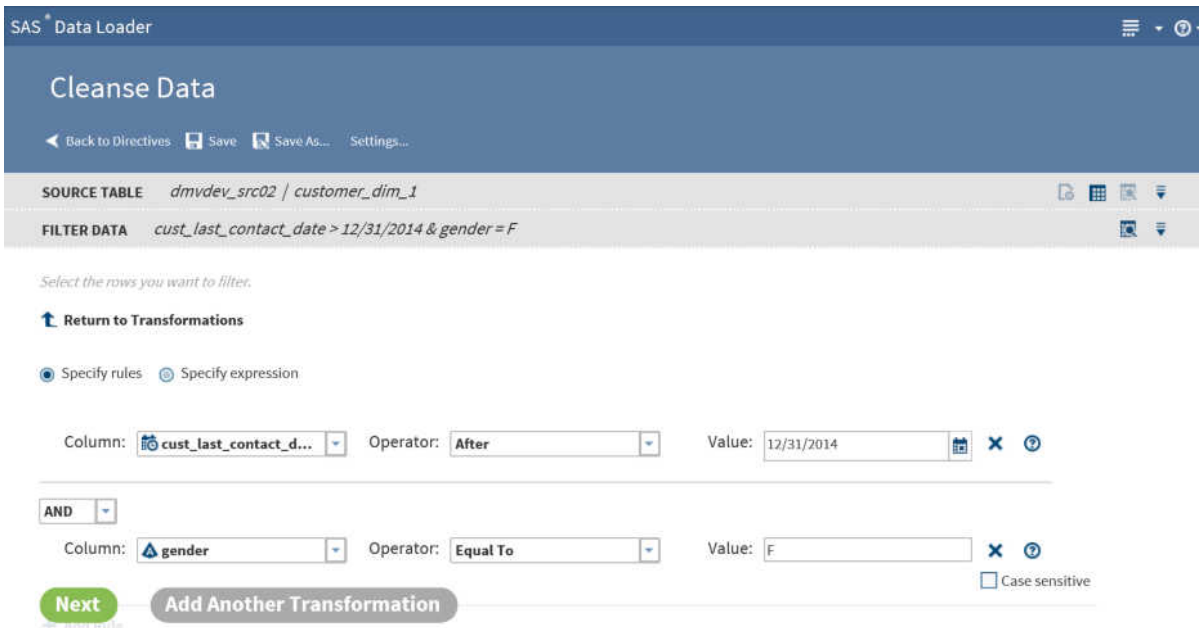
b Click and select a logical **Operator**. The operators that are available depend on the type of the data in the source column. For example, the following image shows the operators that are available for the date/time data type:



c In the **Value** field, add the source column value that completes the expression. In the preceding example, the rule can be read as “Filter from the target all source rows with a last contact date after December 31, 2014.”

d Click **Add Rule** to add another rule. Select a different column, operator, and value.

e To filter rows when either the new rule or the preceding rule are true, change the **AND** condition to **OR**.



f When your rules are complete, go to Step 6.

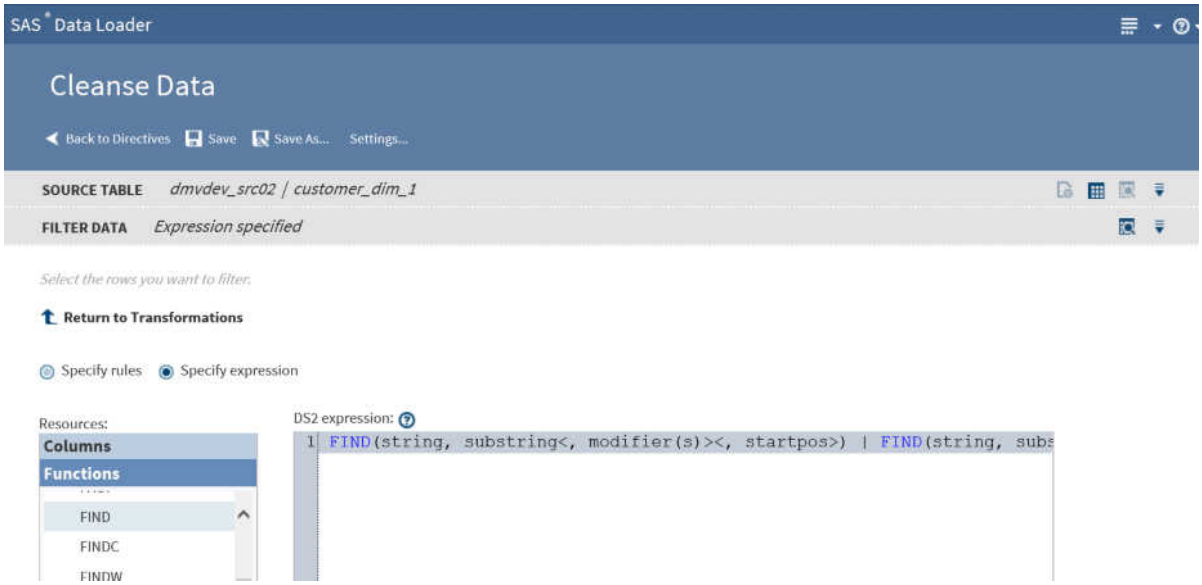
5 To filter rows using an expression, follow these steps:

a In the expression text box, enter or paste your expression.

Your expression can use either SAS DS2 functions (with the MapReduce runtime target,) or DataFlux EEL functions (with the Spark runtime target). Click **Settings** to display the selected runtime target. To learn more about runtime targets, see "Enable Support for Impala and Spark".

To learn the requirements for expressions, see "Develop Expressions for Directives" .

b To add a function to your expression, click **Functions** in the **Resources** box, expand a category, select a function, and click .



To add column names to your expression, position the cursor in the expression text box, click **Columns** in the **Resources** box, click a source column, and then click .

6 When your rules or expression are complete, click **Next** to select a target table and run your job.

To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 32.

Change Case Transformation

Use the Change Case transformation to standardize the casing of selected character columns. You can convert to ALL UPPERCASE, all lowercase, or Initial Capital Letters (or Proper Case).

Follow these steps to use the Change Case transformation:

1 If this is the first transformation in a new job, select a source table.

2 In the **Transformation** task, click **Change Case**.

3 In the **Change Case** transformation, accept or change the default **Locale**. The selected locale needs to reflect the language and region that applies to the content in the source table.

4 Click to **Select a Column**.

5 Select a **Type** of casing for the selected column.

6 Select the case **Definition** that best fits the content of your column. For the source column cust_full_name, and for **Proper** casing, you would select the case definition **Proper (Name)**.

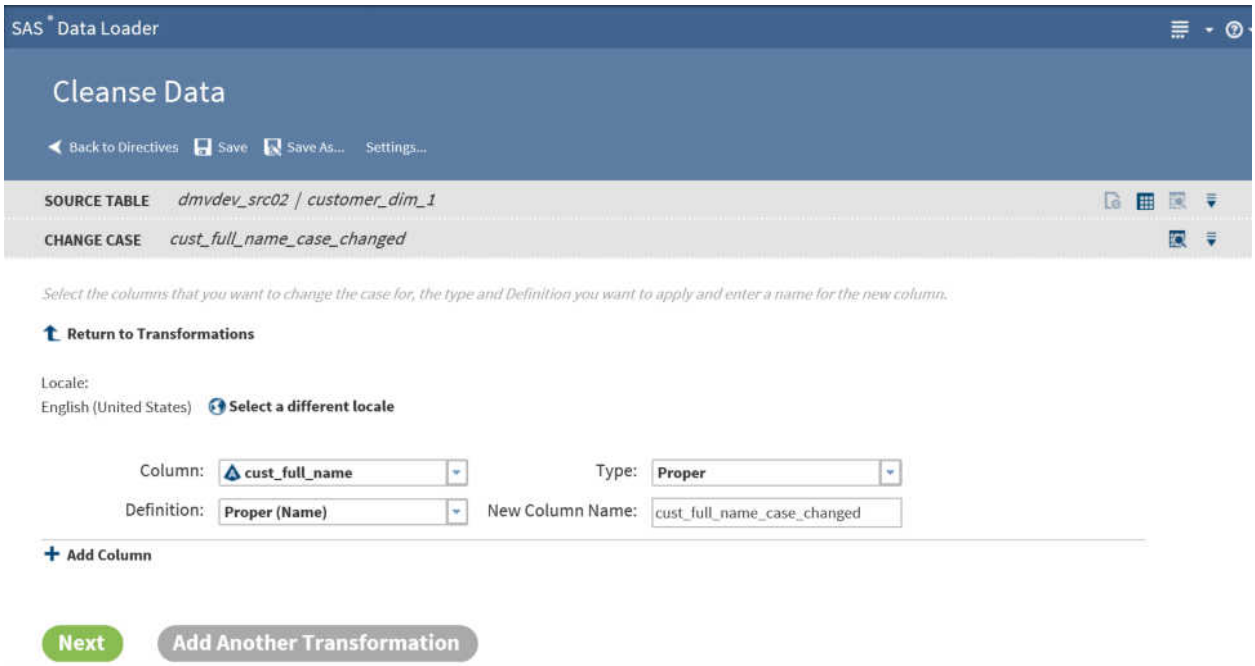
   The case definition is part of the SAS Quality Knowledge Base that is installed on your Hadoop cluster. The case definition determines how case changes are applied to your data, based on your data content and selected locale.

7 Accept or change the default value in the field **New Column Name**.

8 Click **Add Column** + to define another case change.

9 Click **Next** to select a target table and run your job.

   To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 32.



Field Extraction Transformation

Use the Field Extraction transformation to copy tokens from a source column to new columns in the target. Tokens represent types of content that can be extracted using an extraction definition. The available extraction definitions provide locale-specific information that enables extraction.

Follow these steps to use the Field Extraction transformation:

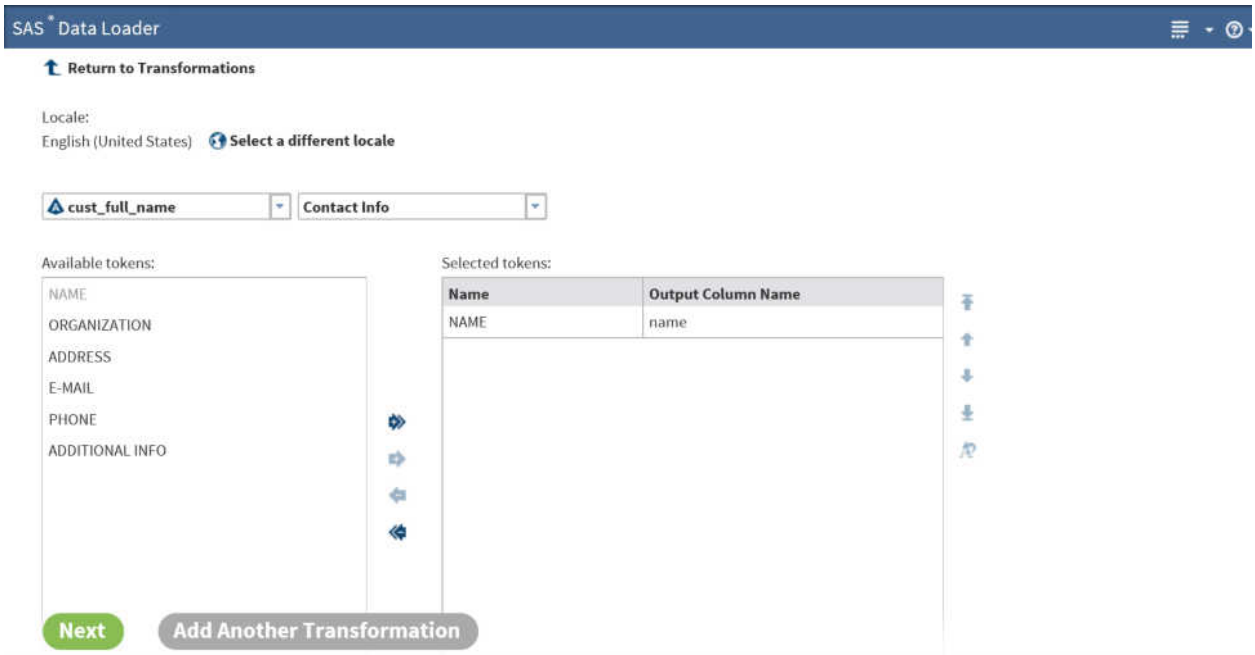
1. If this is the first transformation in a new job, select a source table.
2. In the **Transformation** task, click **Field Extraction**.

3. In the **Field Extraction** transformation, accept or change the default **Locale**.
4. Click **Column** and select a column from which you want to copy data to the target.
5. Click **Definition** and select the set of Field Extraction definitions that best fit your source data. Typical available selections include **Contact Info** and **Product Data**. The list of tokens that appear after you make your selection will show if you selected the appropriate definition.

   The tokens that you select are used to parse each source row and extract values of the specified type.
6. Click one or more tokens that you want to extract from the selected column and click . The tokens and default new column names appear in **Selected Tokens**.

   To select all tokens, click .
7. To change the default column name, click on the name in **Output Column Name**.
8. To reorder the columns in the target, click a row in **Selected tokens** and then click the up and down icons to the right of **Selected tokens**. The top row in **Selected tokens** specifies the first row in the target.
9. Click **Next** to select a target table and run your job.

   To add another data cleansing transformation, click **Add Another Transformation** and see “Select a Data Cleansing Transformation” on page 32.



Parse Data Transformation

Use the Parse Data transformation to extract tokens from a source column and add the token to a new column. A token is a meaningful subset of a data value that provides a basis for analysis. For example, for a column that contains phone numbers, you could extract the area code token and insert that value in a new column. You could then analyze the source table by grouping rows by area code.

Follow these steps to learn how to use the Parse Data transformation:

1 If this is the first transformation in a new job, select a source table.

2 In the **Transformation** task, click **Parse Data**.

3 In the **Parse Data** transformation, click **Select a column** and select a source column from the list.

4 Click the **Definition** field and click the definition that you will apply to the selected column.

5 In the **Available tokens** list, click the token that you will copy out to a new target column.

6 Click the right plus arrow to apply the token to a new column. You can change the suggested **Output Column Name**.

7 At this point you can choose other tokens to add to other new columns in the target table.

8 If you have multiple tokens, you can arrange the target columns using the up and down arrow icons.

9 To remove a token column, select it and click the minus arrow icon 

10 Click **Next** to select a target table and run your job.

To add another data cleansing transformation, click **Add Another Transformation** and see “Select a Data Cleansing Transformation” on page 32.

Standardization Transformation

Follow these steps with your own data to learn how to use the Standardization transformation. This example creates a job that standardizes a column of state names in a table of customer data.

1 If this is the first transformation in your job, select a source table.

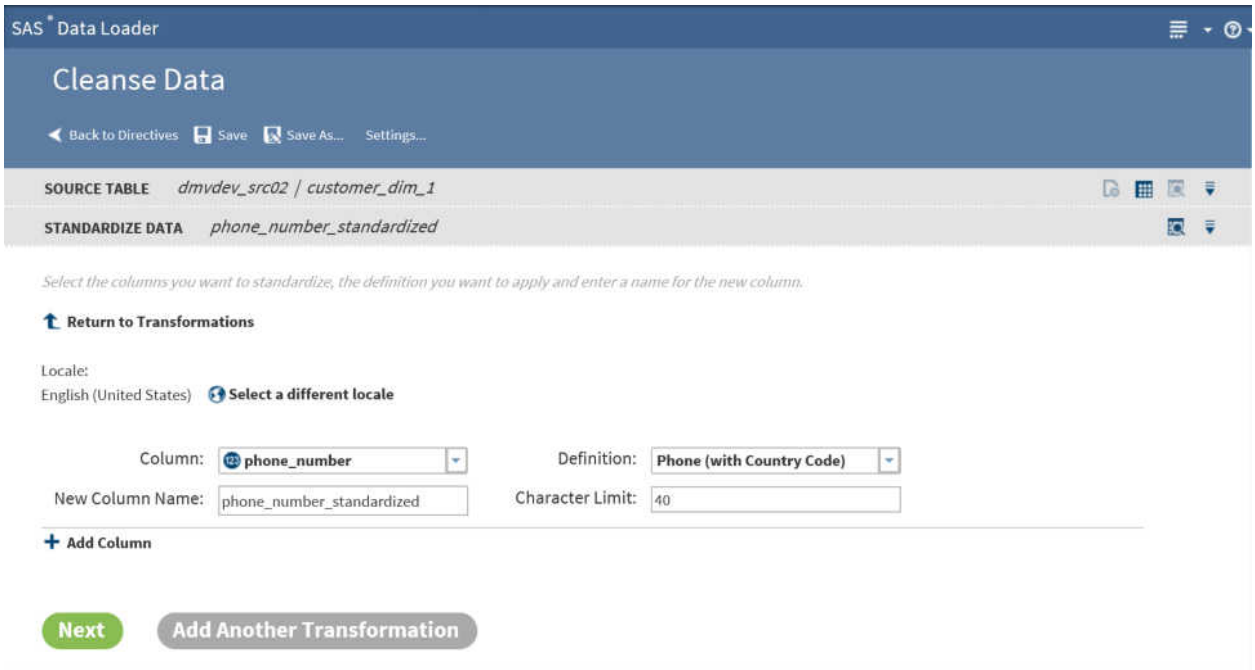
2 In the **Transformation** task, click **Standardize Data**.

3 In the **Standardize Data** transformation, click **Select a Column** and select the column from the list.

4 Click **Select a Definition** and select the standardization definition to be applied to the selected column. Standardization definitions are available for certain character strings and numeric values. Also, standardization definitions are available for generic actions that are independent of content, such as Space Removal and Multiple Space Collapse. To learn about the standardization definitions, Standardization Definitions in the online Help for the SAS Quality Knowledge Base.

5 Standardized values are applied to a new column in the target. You can change the default name of the new column by clicking **New column name**.

6 To save space or truncate long values, you can change the **Character limit** from its default value of 256.



7 The standardization transformation is now completely defined. By default, the target table contains both the original source column and the new standardized column. If you would prefer to remove the source column in the target, or make other changes to target columns, add a Manage Columns transformation toward the end of your job.

Click **Next** to select a target table and run your job.

To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 32.

Pattern Analysis Transformation

The Pattern Analysis transformation reads a source row and generates a corresponding pattern value in the target. The content of the pattern value describes the content of the data. For example, character pattern analysis generate patterns that show if each character is uppercase, lowercase, or numeric.

The patterns form the basis for structural analysis. For example, you can apply a Filter transformation to the output of a pattern analysis. The filter can exclude the expected pattern and write to the target the rows that are structurally invalid.

Follow these steps to use the Pattern Analysis transformation:

1 If this is the first transformation in your job, select a source table.

2 In the **Transformation** task, click **Pattern Analysis**.

3 In the **Pattern Analysis** task, accept or change the default **Locale**. The selected locale needs to reflect the language and region that applies to the content in the source table.

4 Click **Select a column** and click the column that you want to analyze.

5 Click **Definition** and select a pattern analysis definition.

Character
: Generates patterns that represent the types of each character in the source. `A` indicates uppercase, `a` lowercase, `9` numbers, and `*` other (punctuation, and so on). Blanks in the source are replicated as blanks in the pattern. Example: the source value `1 877-846-Flux` generates the pattern `9 999*999*Aaaa`.

Character (Script Identification)
: Generates patterns that identify the Unicode character set of each character in the source. Eleven or more character sets can be detected, including Latin, Arabic, Kanji/Han, Katakana, Cyrillic, and Numeric. Uppercase and lowercase are detected for at least three character sets. Example: `(7F, SAS Institute)スズキイチロウ` generates `*9L* LLL Llllllll*アアアアアアア`.

  Note: The full mapping of pattern characters to Unicode character sets is provided in the Pattern Analysis Definitions in the online Help for the Contact Information Quality Knowledge Base.

Word
: Generates patterns that represent the types of words in the source. `A` represents alphabetic words, `9` numeric, `M` mixed, and `*` other. Example: `216 E 116th St` generates `9 A M A`.

Word (Script Identification)
: Generates patterns that represent the Unicode character set of each word in the source. Eleven or more character sets can be detected, including Latin, Arabic, Kanji/Han, Katakana, Cyrillic, and Numeric. `W` indicates a potentially invalid word that contains multiple character sets. Example: `(7F, SAS Institute)スズキイチロウ` generates `*9L* L L*ア`.

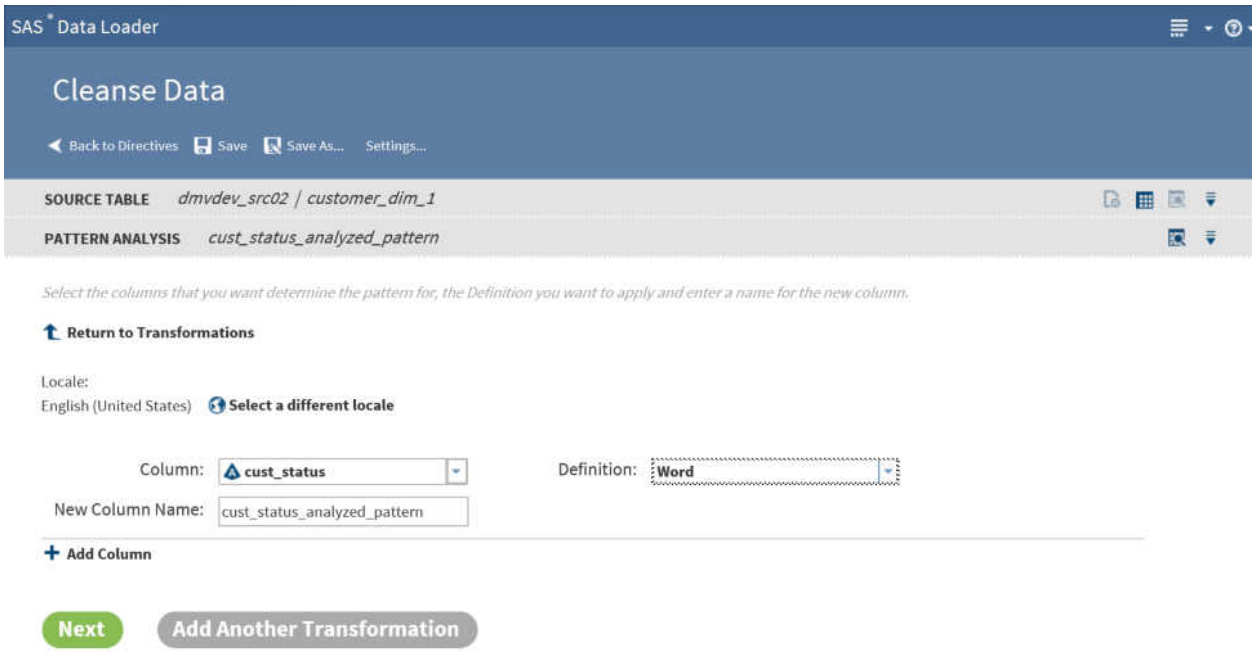
6 Review and update the default **New Column Name**.

7 Review and update as needed the default **New Column Name**.

8 To generate patterns for other columns, click + Add Column.

9 Click **Next** to select a target table and run your job.

To add another data cleansing transformation, click **Add Another Transformation** and see “Select a Data Cleansing Transformation” on page 32.



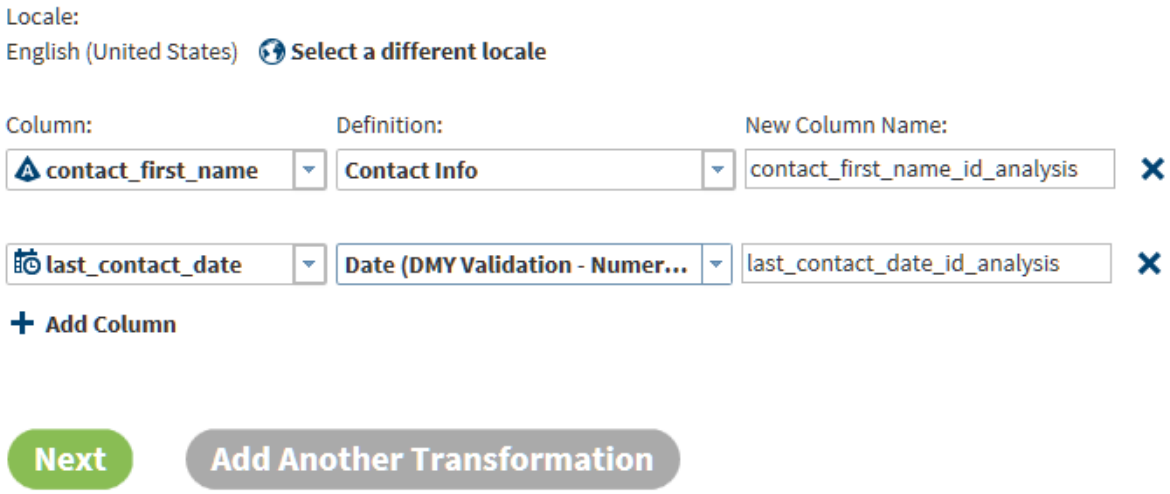
Identification Analysis Transformation

Use the Identification Analysis transformation to report on the type of the content in a given column. The content types that can be detected include contact information, dates, email, field names, offensive content, and phone numbers. The result of the analysis is added to a new column in the target table. You can analyze one column for multiple content types, and you can analyze multiple columns in the source table.

Follow these steps to use the Identification Analysis transformation:

1. If this is the first transformation in your job, select a source table.
2. In the **Transformation** task, click **Identification Analysis**.

3. In the **Identification Analysis** transformation, click **Select a Column**, and then select a column for analysis.
4. Click **Select a Definition** and choose the content type that you want to apply to the source column.
5. In the **New Column Name** field, a name is suggested for the column that will be added to the target. The new column will contain the results of the identification analysis. Click in the text field for New Column Name to change the suggested column name.
6. To analyze another column, or to analyze the same column with a different definition, click **Add Column**.



7 Click **Next** to select a target table and run your job.

To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 32.

Gender Analysis Transformation

The Gender Analysis transformation analyzes columns of names and generates columns that indicate the probable gender of the names.

Follow these steps to use the Gender Analysis transformation:

1 If this is the first transformation in your job, select a source table.

2 In the **Transformation** task, click **Gender Analysis**.

3 In the **Gender Analysis** transformation, review and update the default **Locale** as needed to ensure that the locale matches the content of your source data.

4 Click **Select a Column** and click the column of name data in your source table.

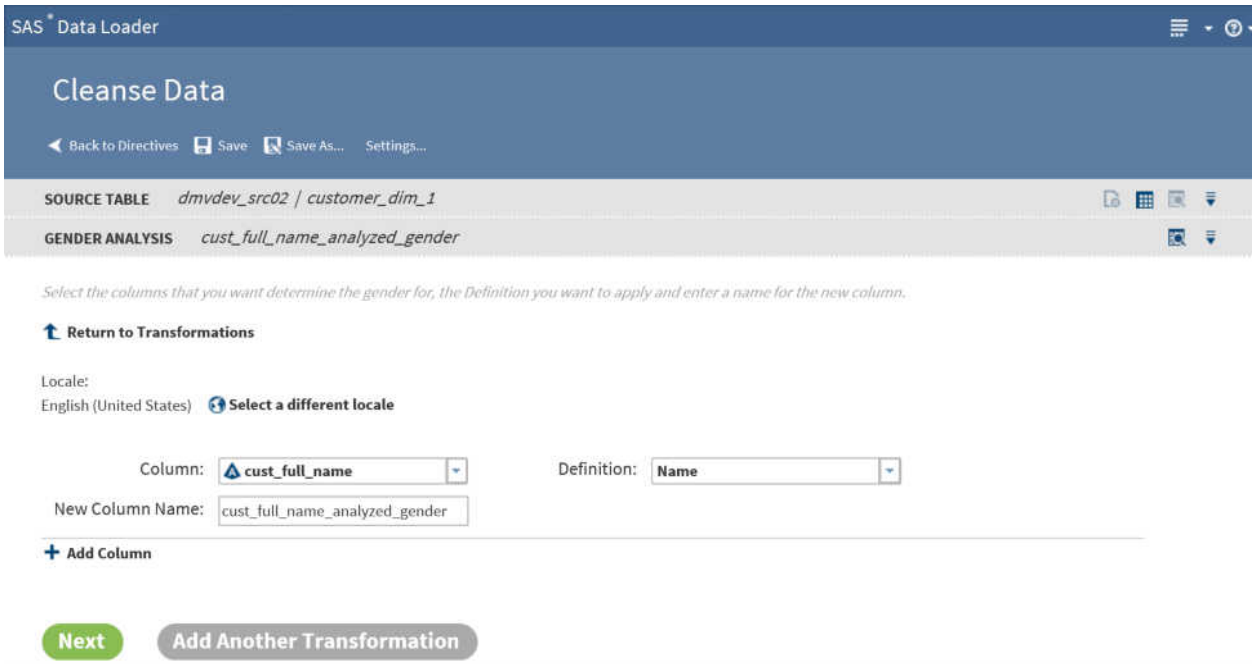
5 Click **Definition** and click **Name**.

6 To analyze a second column of name data, click + Add Column.

7 Review and update as needed the default **New Column Name**.

8 Click **Next** to select a target table and run your job.

To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 32.



Generate Match Codes Transformation

The Generate Match Codes transformation generates match codes for specified columns. The generated match codes are then added to new columns in the target table. The match codes are generated based on a definition and a sensitivity. The definition specifies the type of the content in the column. The sensitivity determines the degree of exactitude that is required in order for two data values to be declared a match. Higher sensitivity values specify that data values must be more similar to be declared a match. Lower sensitivity values enable matching with less similarity. The level of sensitivity is reflected in the length and complexity of the match codes.

Match codes can be used to find columns that contain similar data. For example, you can generate match codes for name and address columns, and then compare the match codes to detect duplicates.

Follow these steps to use the Generate Match Codes transformation:

1 If this is the first transformation in your job, select a source table.

2 In the **Transformation** task, click **Generate Match Codes**.

3 In the **Generate Match Codes** transformation, click **Select a Column** and then click the column for which you want to generate match codes.

4 Click **Select a Definition** and then click the definition that you want to use to generate match codes.

5 To change the default sensitivity value, click the **Sensitivity** field and select a new value. Lower sensitivity numbers give you more matches (identical match codes) and perhaps more matching errors. Higher sensitivity numbers produce the same match code only when data values are nearly identical.

6 Click **Next** to select a target table and run your job.

To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 32.

Manage Columns Transformation

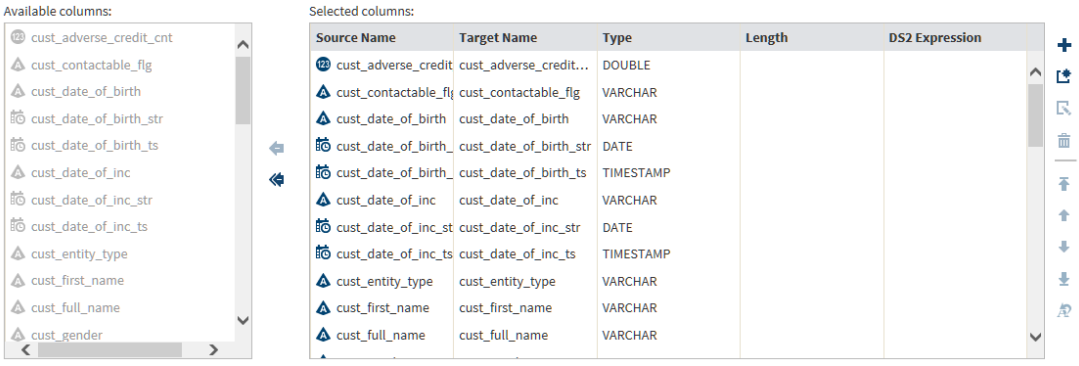
Use the Manage Columns transformation to remove, reorder, rename, change type, and change the length of the columns in the target table. You can also use Manage Columns to add generated data to new columns or to modify or replace data in existing columns. The new or changed data is generated by user-written expressions.

Follow these steps to learn how to use the Manage Columns transformation:

1 If this is the first transformation in your job, select a source table.

2 In the **Transformation** task, click **Manage Columns**.

3 In the **Manage Columns** transformation, columns are listed in order of appearance. The top column is the first or leftmost column.



Note the arrow icons between **Available columns** and **Selected columns**. To remove a column from the target, click the column name on the right and click the top arrow. To move all columns out of the target, click the double-arrow icon. After you remove a column, arrows will appear so that you can move columns from Available to Selected.

Initially, all columns are selected for the target table, including all of the new columns that you added in prior transformations.

4 Locate the icons on the right side of **Selected columns**. These icons provide the following functions:

- Add a new column and enter or paste an expression into the **Expression** column.

  Your expression can use either SAS DS2 functions (with the MapReduce runtime target,) or DataFlux EEL functions (with the Spark runtime target.) Click **Settings** to display the selected runtime target. To learn more about runtime targets, see "Enable Support for Impala and Spark".

  To learn the requirements for expressions, see "Develop Expressions for Directives".

- Add a new column and specify an expression using the Advanced Editor window. To learn how to use the Advanced Editor, see "About Expressions and the Advanced Editor".
- Develop an expression using the Advanced Editor.
- Remove the selected column from the target table. Removed columns appear in **Available columns**.
- Move the selected column to the first column position in the target.
- Move the selected column one position to the left in the target.
- Move the selected column one position to the right.
- Move the selected column to the last column position in the target (rightmost.)
- Change the name of the selected target column.

5 Click **Next** to select a target table and run your job.

To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 32.

Summarize Rows Transformation

Use the Summarize Rows transformation to add summarized numeric values to your target table. To generate summaries, you first group rows by one or more columns. Then you select the columns that you want to summarize for each group and subgroup. The method of summarization is known as an aggregation. The number of aggregations depends on the column data type. Numeric columns have 13 available aggregations.

Follow these steps to learn how to use the Summarize Rows transformation:

1 If this is the first transformation in your job, select a source table.

2 In the **Transformation** task, click **Summarize Rows**.

3 In the **Summarize Rows** transformation, click the **Group rows by** field and choose the first column that you want to use to group rows for summarization. In the target table, rows with the same values in the selected column appear together, along with their summary values in new columns.

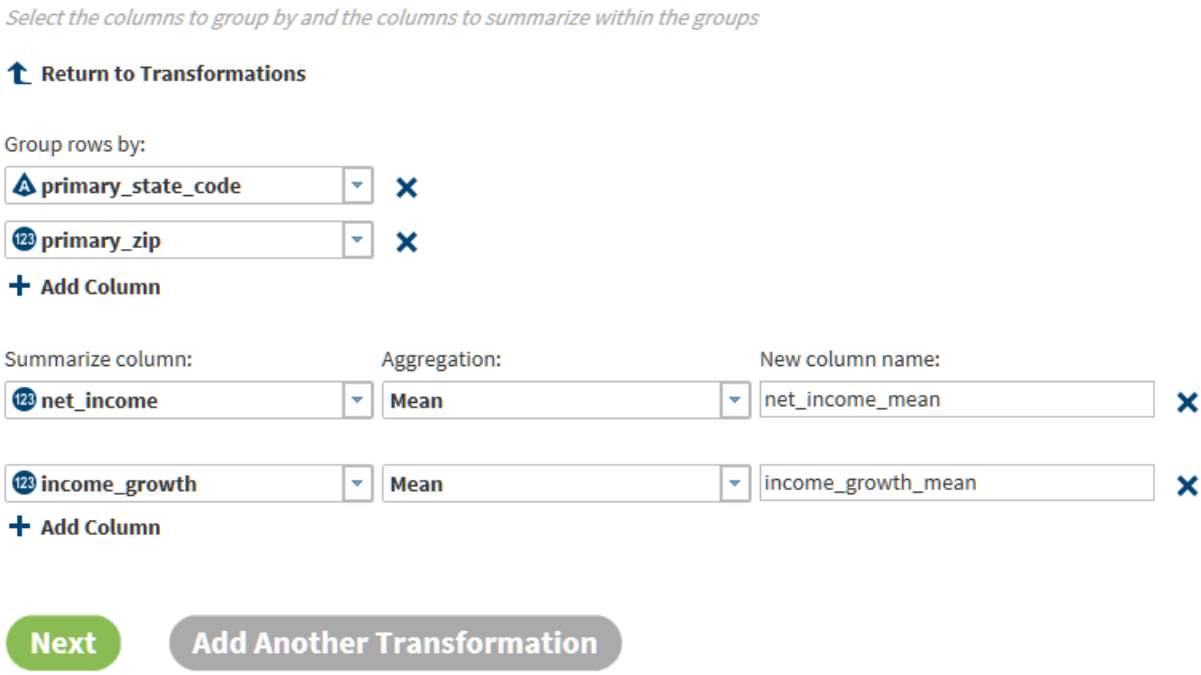
4 To further subset the initial set of groups, and to generate a second set of summary values, click **Add Column**. Select a second column. Add additional groups as needed.

5 Click **Summarize column** and select the first numeric column that you want to summarize.

6 Click **Aggregation** and select the aggregation that you would like to provide for the selected column.

7 To change the suggested name for the new column that will contain the aggregation values for each group, click **New Column Name**.

8 To add a second aggregation, click **Add Column**.



9 Click **Next** to select a target table and run your job.

To add another data cleansing transformation, click **Add Another Transformation** and see “Select a Data Cleansing Transformation” on page 32.

Select a Target Table and Run Your Job

After you click **Next**, follow these steps to select a target table and complete your data cleansing job:

1 In the **Target Table** task:

To select an existing target table (and completely overwrite any existing content), click the data source, click an existing target table, and then click Next. Or, you can click Select Recent Table and choose from a list of your recent targets.

To create a new target table, click a data source, click **New Table** New Table..., and specify the table name in the New Table window. A new table of that name appears in the grid or list.

To explore the contents of target tables, click a table and click **Data Sample**, Table Viewer, or (if available) View Profile.

To view or change the target table format or the Hive storage location, click.

2 With a target table highlighted in the list or grid, click **Next**.

3 In the **Result** task, click **Save** or **Save As** to save your job, and then click **Start Transforming Data**.

4 When the job is complete, you can view the results, the log file, and the code that ran in Hadoop.

About Expressions and the Advanced Editor

The **Advanced Editor** window enables you to develop expressions. The editor is available in the Filter and Manage Columns transformations. In those transformations, user-written expressions can filter source rows from the target, modify or replace existing column data, and generate new data for new target columns.

Expressions are written using SAS DS2 functions in the MapReduce runtime environment, or DataFlux EEL functions in the Spark runtime environment. You can determine or specify the runtime environment using the **Settings** menu at the top of the directive. For more information about runtime environments, see "Enable Support for Impala and Spark".

In the Advanced Editor, you can click **Save and New** to save your expression and apply its return value to a new target column.

In the Advanced Editor, the **Resources** list box displays the categories of the available DS2 or EEL functions. Within the categories, each function displays a short description and syntax help. To add a function to your expression, click the listing and click.

The **Resources** list box also lists column names. To add a column name to your expression, click the cursor in the expression text box, click the column name in **Resources**, and then click.

To learn the requirements of expressions, see “Develop Expressions for Directives” .

Cluster-Survive Data

Introduction

The Cluster-Survive Data directive enables you to use rules to cluster similar records together and to use survivorship rules to produce a unique surviving record from that cluster. If you want to do cluster analysis only, you are not required to run the survivorship processing.

Note: Before running the Cluster-Survive directive, it is a best practice to use the data quality transformations available through the Cleanse Data directive. These transformations standardize or compute match codes from values in the records, which can improve the clustering results.

Prerequisites

To run Cluster-Survive Data directives in Hadoop, Apache Spark must be installed and configured on your Hadoop cluster. For information about the supported versions of Spark, see *SAS 9.4 Supported Hadoop Distributions*, at https://support.sas.com/resources/thirdpartysupport/v94/hadoop/hadoop-distributions.html. See also the “Usage Notes for Spark” .

The SAS Data Management Accelerator for Spark must also be deployed on your Hadoop cluster, as described in the *SAS In-Database Products: Administrator’s Guide*.

To learn more about subjects such as string truncation in Spark, see “Usage Notes for Spark”.

About Clustering and Survivorship

Clustering

Clustering helps identify unique entities after you determine which fields can be used to identify related records.

Using a set of cluster rules, the clustering process partitions a set of records into related sets of entities based on the rules. Each set is assigned a unique cluster ID, which is added as a field to the records. Cluster rules are expressed as sets of fields that have equal values. If the values for two records match for the fields specified in the rule, the records are considered matched and are assigned to the same cluster.

The simplest example of a cluster rule uses a single column. Rows in the table that have the same value for a field in this column are gathered into a set. In the following example, if the cluster rule specifies only the NAME column, then the result yields two sets:

Table 4.1 NAME Result

name	email1	email2	
Alice	alice@alice.net	alice@alice.com	SET 1
Alice	(null)	Alice	
Bob	bob@bob.com		SET 2
Bob	robert@robert.com		

To define identity such that both the NAME and EMAIL1 columns are considered, you can use the AND Boolean operator to construct a NAME and EMAIL1 rule, which yields four sets:

Table 4.2 NAME and EMAIL1 Result

name	email1	email2	
Alice	alice@alice.net	alice@alice.com	SET 1
Alice	(null)	alice@alice.net	SET 2
Bob	bob@bob.com		SET 3
Bob	robert@robert.com		SET 4

In the previous example, the two Alice records are separated into two sets because the values for EMAIL1 differ. You might want these two records to be identified as one set because alice@alice.net is a common address for both records, but it appears in two different columns. If you want to combine these records together in a set, you must specify an advanced cluster rule using parentheses and the OR Boolean operator to construct the rule NAME and (EMAIL1 or EMAIL2):

Table 4.3 NAME and (EMAIL1 or EMAIL2) Result

name	email1	email2	
Alice	alice@alice.net	alice@alice.com	SET 1
Alice	(null)	alice@alice.net	
Bob	bob@bob.com		SET 2
Bob	robert@robert.com		SET 3

When you use the OR operator this way, matching values in the NAME column and matching values in either the EMAIL1 or EMAIL2 columns cause the records to be grouped into one set.

It is possible to use the OR operator without parentheses. For example, the rule EMAIL1 or EMAIL2 still returns a single set for the Alice records. However, when the rule uses AND as well as OR operators, an order of precedence applies. Without specifying the parentheses, the previous rule would be evaluated as (NAME and EMAIL1) or EMAIL2, which yields very different results.

Multiple cluster rules can be specified, and are all used when grouping records together.

Survivorship

The clustering process identifies a set of records that are logically related, after which you can use survivorship rules to create a survivor record. The survivor record is considered the best single representative of the clustered data. Survivorship processing uses rules for base record selection and column selection. For example, after unique records are identified and clustered, you might find multiple addresses listed for a particular individual. The rules have identified these records as similar, but you must determine which address should be used as the standard.

Note: Leading and trailing spaces are trimmed from fields when evaluating survivorship rules.

Survivor Base Record Rules

Survivor base record rules are used to select a single record as the template for the surviving record. The rules apply only to records that share a common identity as determined by the cluster rules. The template can be further modified by applying survivor column rules that update individual field values from the records in the set.

The base record is determined by applying operators to column values. For example, if your record contains a date field indicating when it was last updated, you can use a rule such as Max(UPDATED) to select the most recent update to use as your base record:

Table 4.4 *Max(UPDATED)*

Name	Street	City	UPDATED
Alice	101 Main Street	Albuquerque	June 26, 2015
Alice	205 North Avenue	Tucson	May 15, 2012
Alice	PO Box 12081	Tucson	January 8, 2012

Alternatively, you could use the most frequent value for city. For example, you could use Highest Occurrence (CITY) in your rule, which would yield two possibilities for the base record:

Table 4.5 Highest Occurrence (CITY)

Name	Street	CITY	Updated
Alice	Alice	Albuquerque	June 26, 2015
Alice	205 North Avenue	Tucson	May 15, 2012
Alice	PO Box 12081	Tucson	January 8, 2012

You can have only one base record, so you must either refine the first rule or provide another rule that selects a unique record.

If you want to refine the results, you can specify another rule that is linked to the first rule by using the **Apply to previous rule output** setting. When this option has been selected, only the records that match the first rule are provided to the second. For example, if the first rule is Highest Occurrence(CITY) and the second rule is Max(UPDATED), the base record represents the last known address for Alice in Tucson:

Table 4.6 Highest Occurrence(CITY) and Max(UPDATED)

Name	Street	CITY	UPDATED
Alice	101 Main Street	Albuquerque	June 26, 2015
Alice	205 North Avenue	Tucson	May 15, 2012
Alice	PO Box 12081	Tucson	January 8, 2012

Sometimes, chaining rules can result in no records being selected, even though at some point in the process a unique record was selected. For example, if you added another chained rule that required the UPDATED field to be greater than January 1st, 2013, no records would be selected. Even though a unique record was selected by the second rule in the chain, it does not survive the third rule. Use the **Stop processing rules after the first rule yields a single record** option to stop processing when a unique record has been identified.

Alternatively, you can create another rule, or chain of rules, that uses a different strategy to select a base record. If the **Apply to previous rule output** option is not selected for a rule, then the complete group of matched records is provided for the rule to process.

Survivor Column Rules

Survivor column rules apply field-level edits to the survivor base record, where field values are selected from the clustered group of records. This is useful when integrating data from different sources where the fields might be absent or less reliable.

For example, given the following data, and assuming that the first record has been selected as the base record, you might want to incorporate the email address from the second record into the base record:

Table 4.7 *First Row Base Record*

Name	Street	City	Updated	Email
Alice	101 Main Street	Albuquerque	June 26, 2015	(null)
Alice	205 North Avenue	Tucson	May 15, 2012	alice@alice.net

To do this, you would build one or more rules that identify the desired field values and copy them into the base record. In this example, a rule such as Not Null(EMAIL) selects the second record. Specifying the EMAIL field as the field selection then causes the (null) value to be updated:

Table 4.8 *Not Null(EMAIL) Result*

Name	Street	City	Updated	Email
Alice	101 Main Street	Albuquerque	June 26, 2015	alice@alice.net

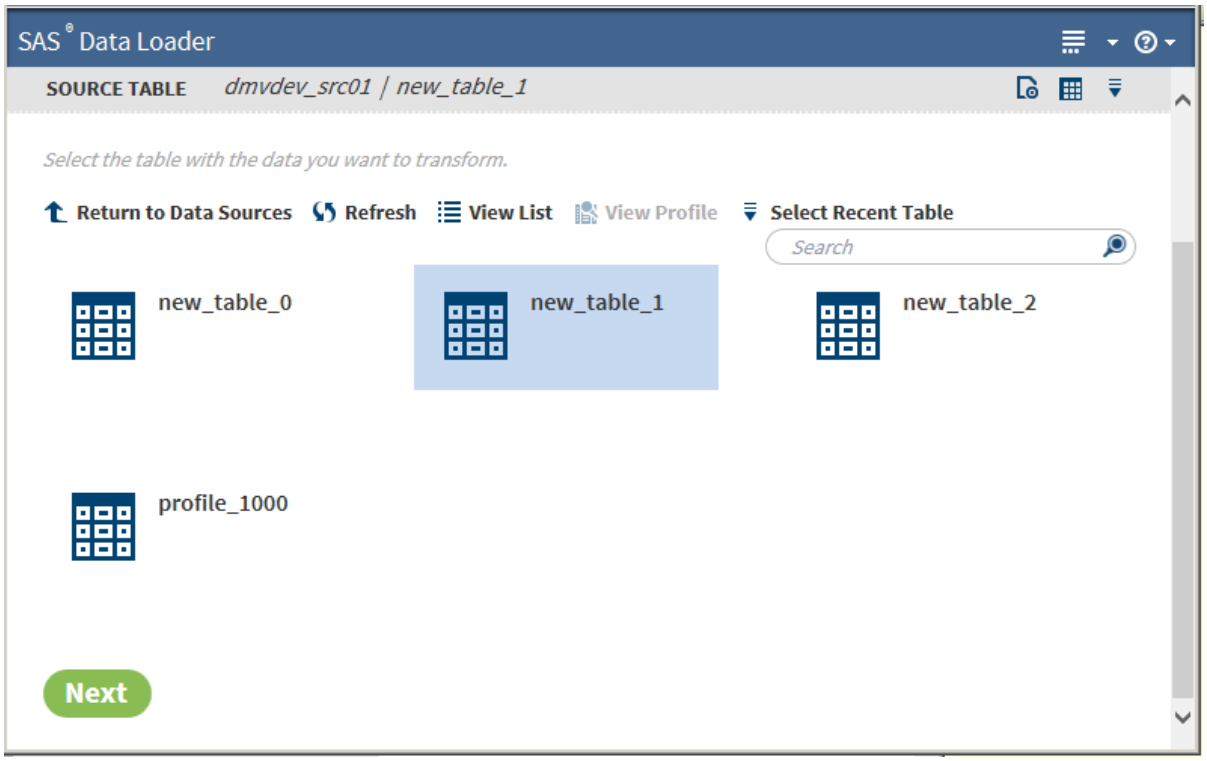
Note: The survivor column rules modify a copy of the base record, so that the original record in a cluster group remains unchanged.

The fields used in the rule are not required to be related to the fields specified in the field selection list. For example, you could use the Updated field to identify the record, and then copy the Street and City field values to the survivor without including Updated in the field selection list.

Example

Follow these steps to use the Cluster-Survive directive:

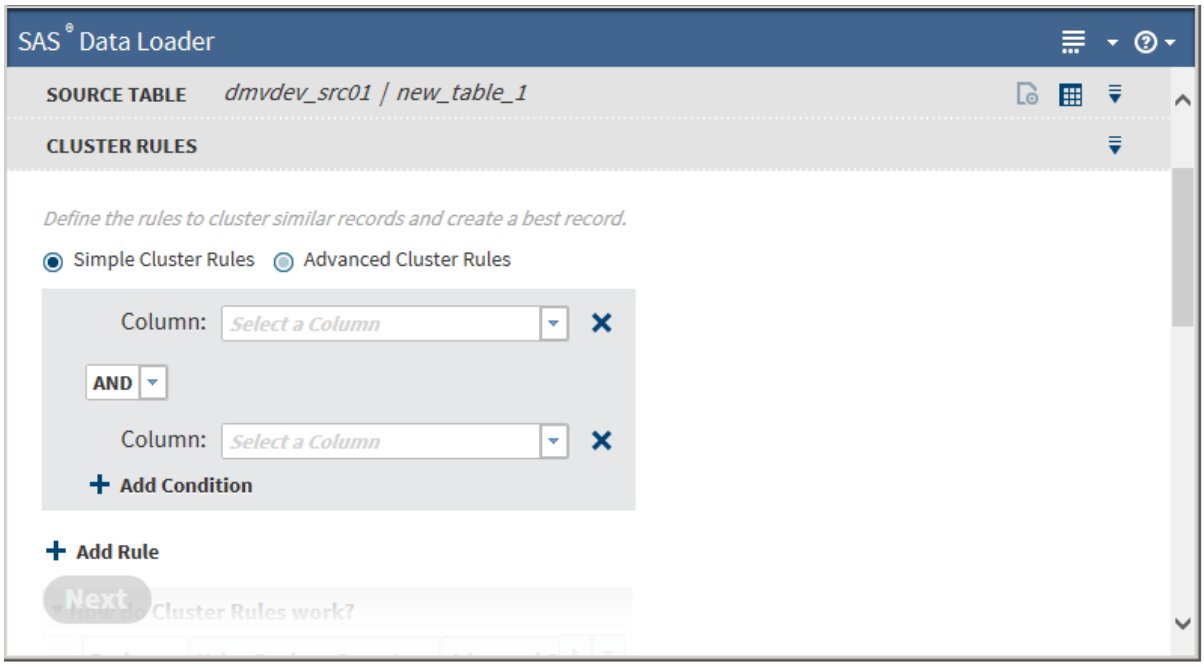
1 On the SAS Data Loader directives page, click **Cluster-Survive**. The Source Table task is displayed:



For more information about data sources and tables, see "Viewing Data Sources and Tables" on page 20.

2 Select a table. To explore the contents of source tables, click to open the table viewer or, if available, to open the profile.

After selecting a table, click **Next**. The Cluster Rules task is displayed:



3

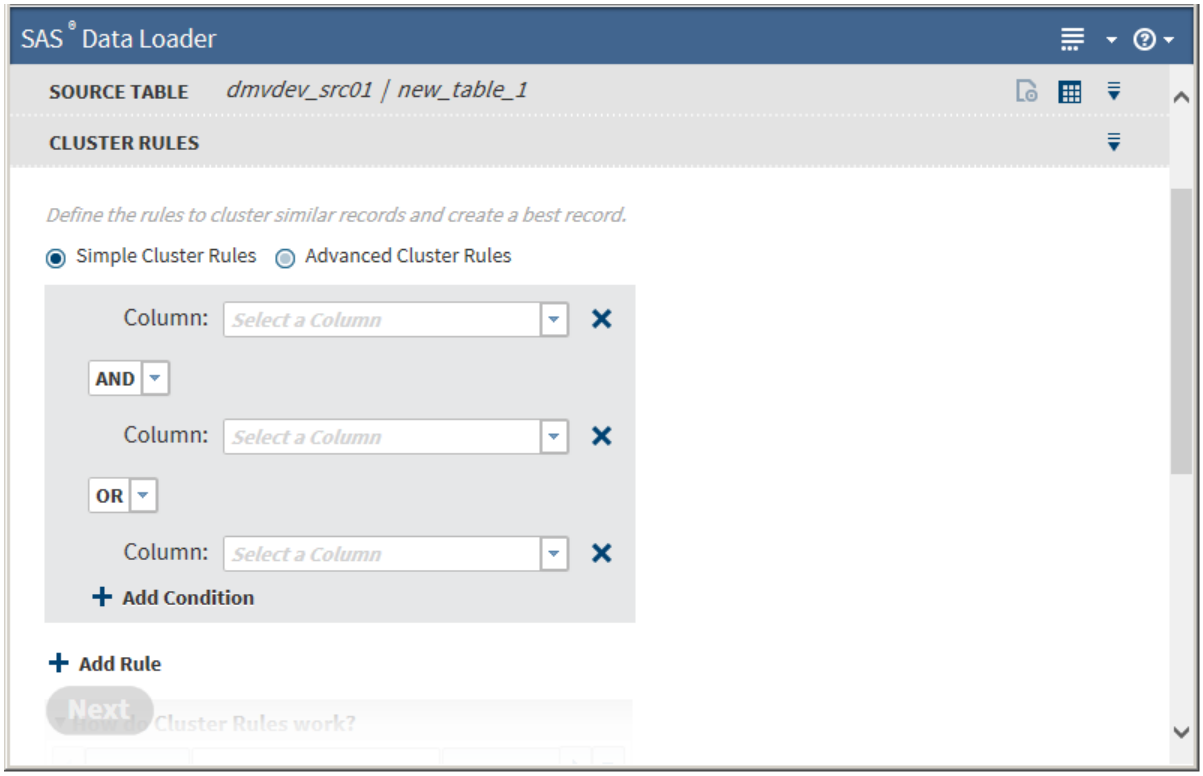
Cluster rules are sets of column names from the source table joined by AND| OR conditions. The initial Cluster Rules task defaults to **Simple Cluster Rules**, which does not allow the specification of parentheses.

You must specify a minimum of one rule (one column). Clicking ✕ next to a column field removes that field.

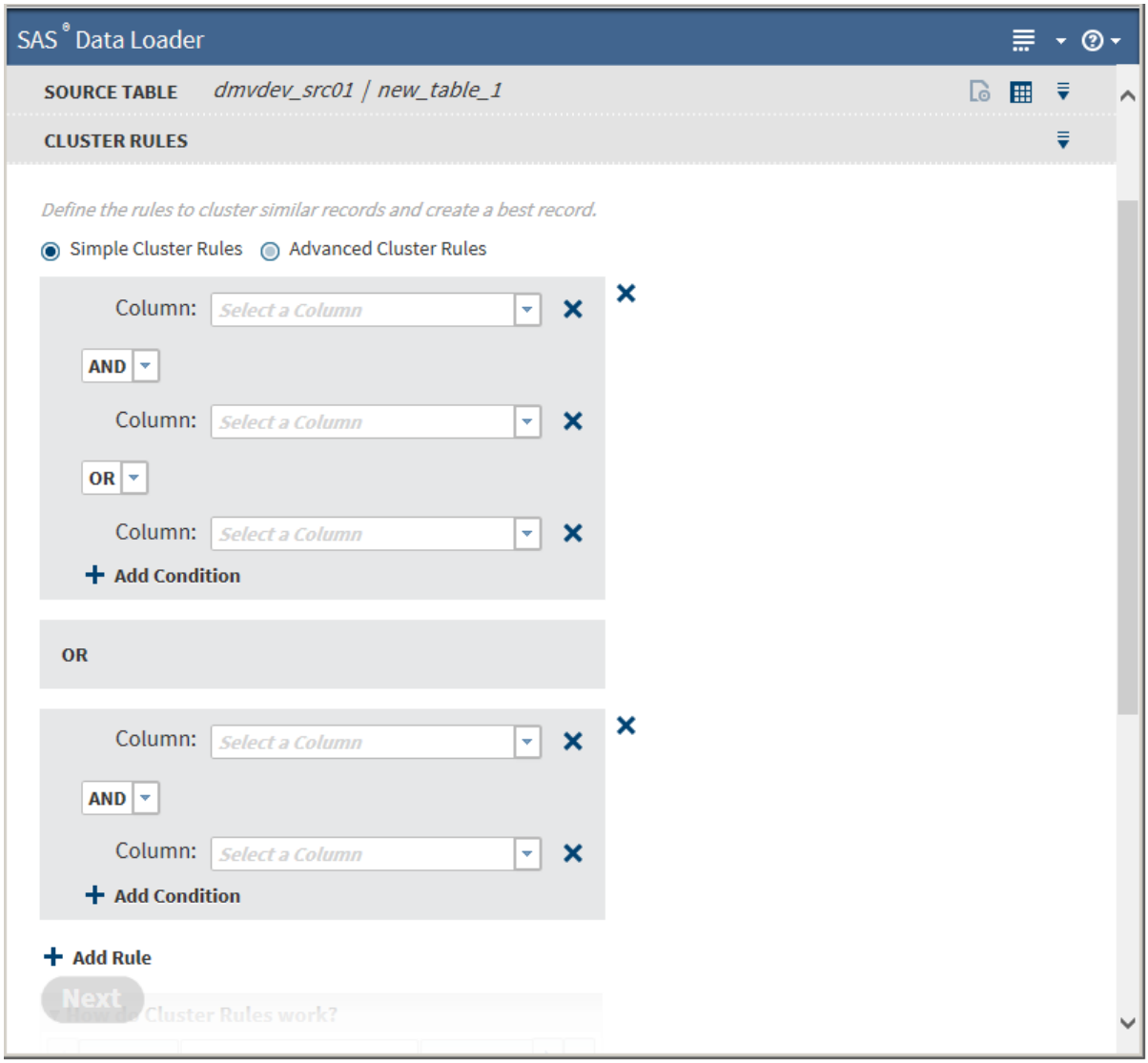
a Select a column from the **Column** drop-down list.

b Choose either the AND or the OR Boolean operator from the drop-down list.

c To add another condition, click + next to **Add Condition**, which adds another column field to the same rule:

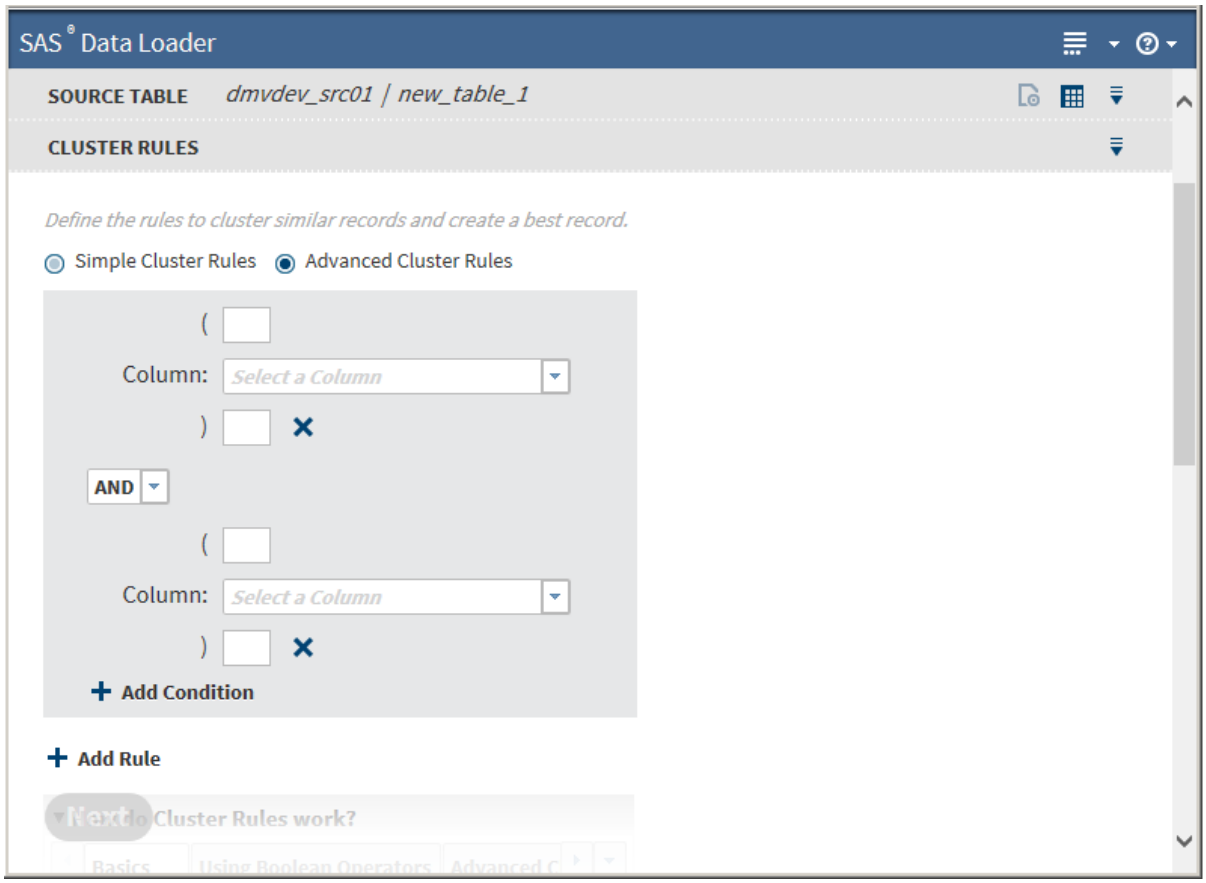


d To add another rule, click + next to **Add Rule** :



e To remove a rule, click ✕ next to the rule pane.

Clicking **Advanced Cluster Rules** adds fields for entering parentheses:



Clicking **Simple Cluster Rules** removes the fields for entering parentheses.

The Advanced Cluster Rules enable you to use parentheses within the rule. Generally, you need to choose advanced rules only if the rules contain a mix of AND and OR clauses. For more information, see "Clustering" on page 49.

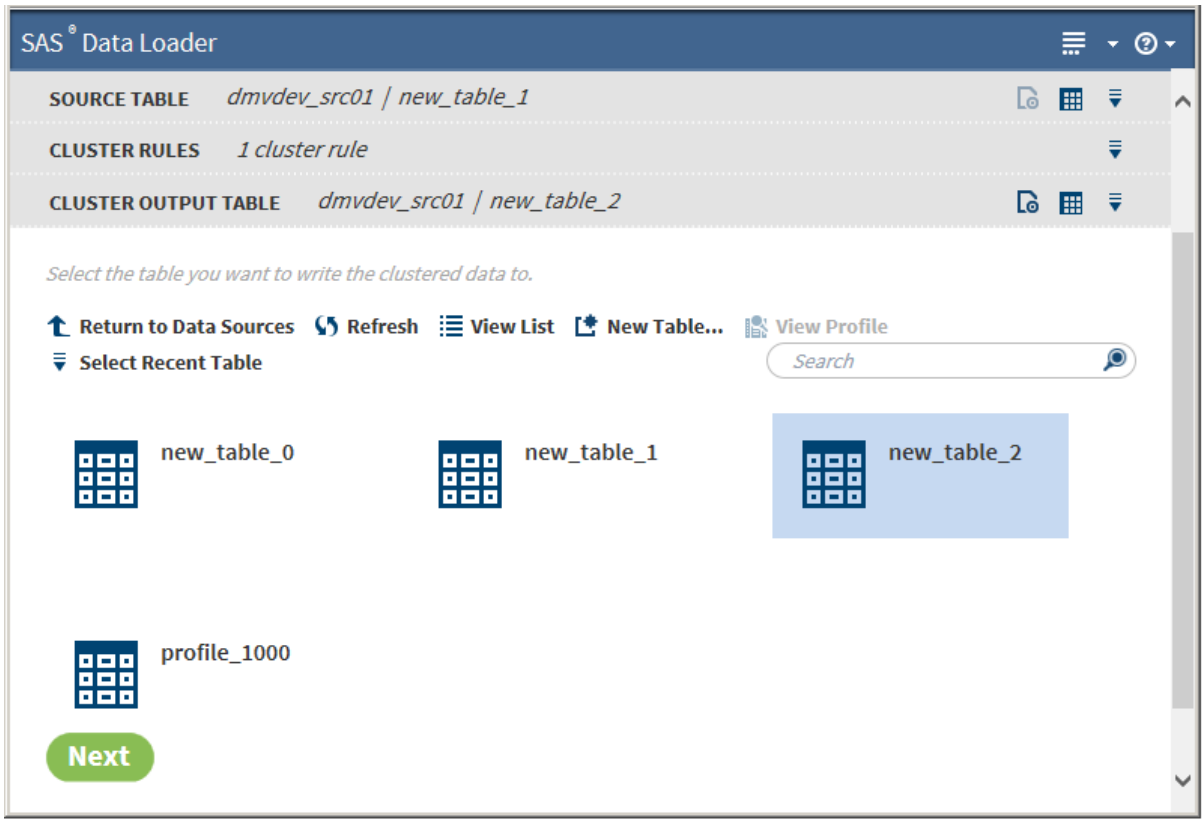
If you need to build complex rules, you can include multiple parentheses in the field. For example, you might create the following rule:

```
name and ((phone1 or phone2) or (email1 or email2))
```

Parentheses must be balanced within a rule, that is, every open parenthesis must have a corresponding close parenthesis.

All other operations on the Advanced Cluster Rules task are the same as on the Simple Cluster Rules task.

After creating rules, click **Next**. The Cluster Output Table task appears:



4 The Cluster Output Table task is the step at which you can choose to:

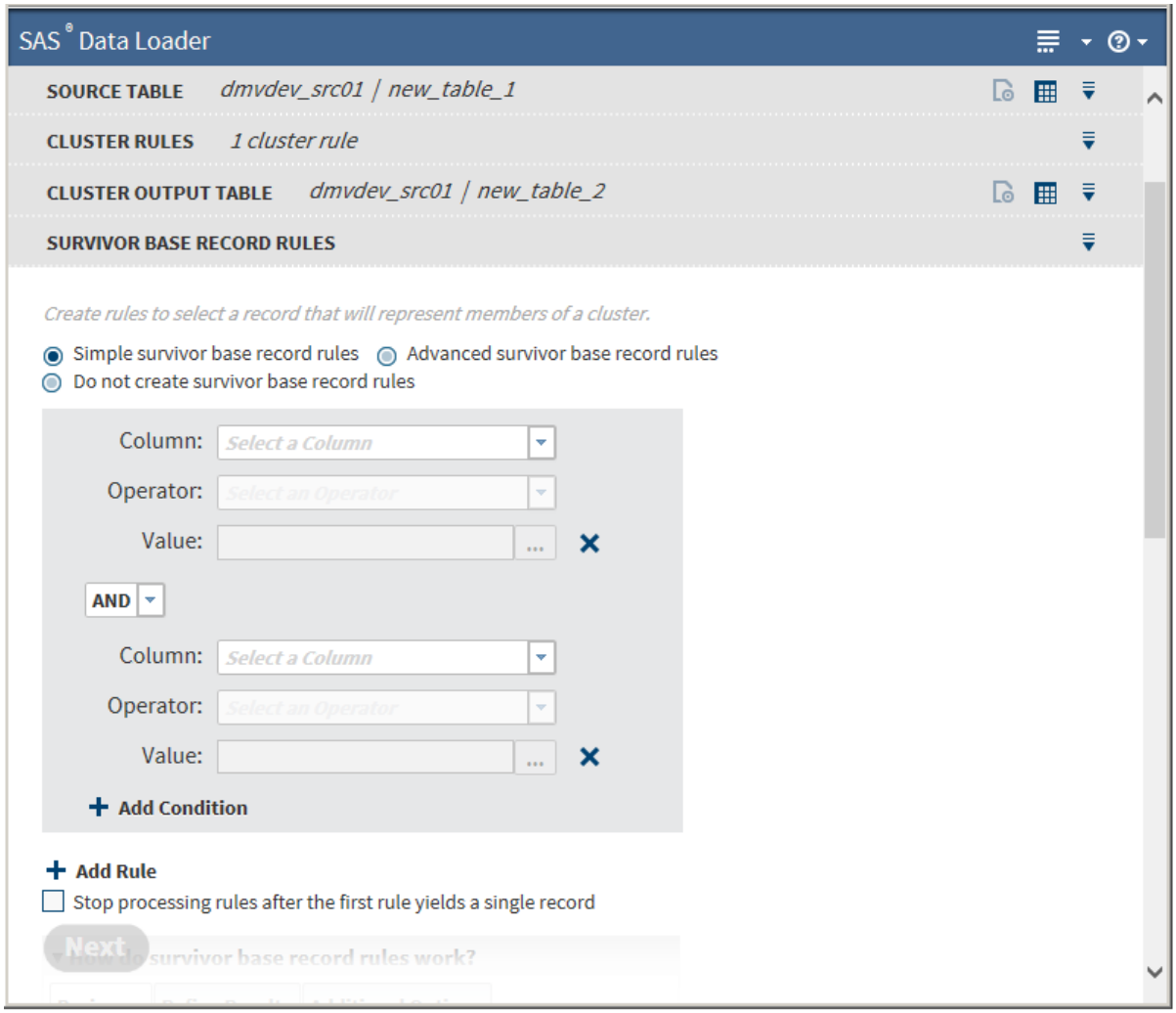
- designate an output table to which to save cluster information. If you choose to save to a table, you have the option of proceeding directly to processing the data without specifying any survivor rules in the following steps. If you want to save to a table:

  1 Select a table.

  Note: If you change your mind, you can click another table to select it or deselect the table. Most browsers use Ctrl-Click to deselect. If necessary, consult your browser help.

  2 Click **Next**. The Survivor Base Record Rules task appears.

- not designate an output table. If you do not designate an output table, you must create at least one survivor base record rule. If you do not want to save to a table, Click **Next** without selecting a table. The Survivor Base Record Rules task appears:



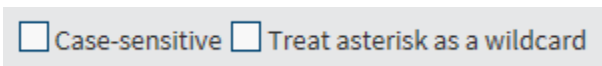
5 The initial Survivor Base Record Rules task defaults to **Simple survivor base record rules**, which does not allow the specification of parentheses. If you have previously selected a cluster output table, you can choose to skip Survivor Base Record Rules by selecting **Do not create survivor base record rules**, and then clicking **Next**.

Basic operations are the same as for Cluster Rules. You must specify a minimum of one rule. Clicking ✕ next to a value field removes that rule.

a Select a column from the **Column** drop-down list.

Note:

- If the selected column contains string data, the following additional fields appear:

  If appropriate, select one or both of the fields.

- If **Treat asterisk as a wildcard** option is selected, the asterisk must appear either at the beginning or the end of the string, and cannot appear more than once.

After selecting a column, the **Operator** field is enabled.

b Select an operator from the **Operator** drop-down list. The available values for this field depend on which column was selected.

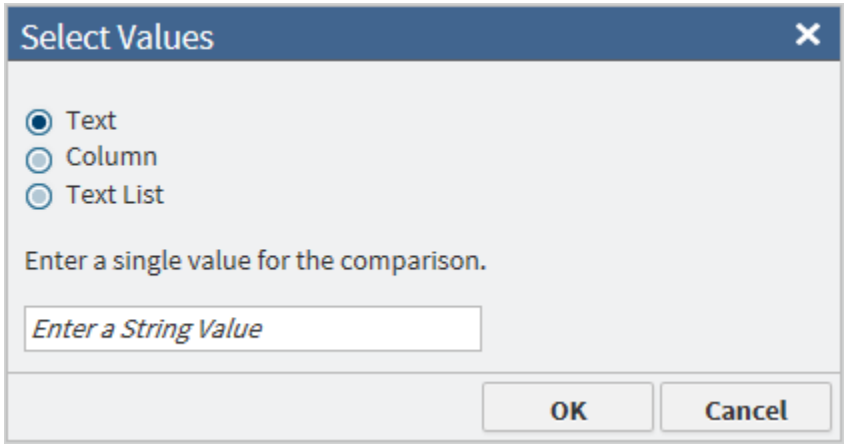
Operator	Description
Equal To **Note:** For time and date fields, this value becomes On.	Field value is ■ equal to another field ■ equal to a literal ■ contained in a list of literal values
Not Equal To **Note:** For time and date fields, this value becomes Not On.	Field value is not ■ equal to another field ■ equal to a literal ■ contained in a list of literal values
Greater Than **Note:** For time and date fields, this value becomes After.	Field value is greater than ■ another field ■ a literal
Less Than **Note:** For time and date fields, this value becomes Before.	Field value is less than ■ another field ■ a literal
Greater Than or Equal To **Note:** For time and date fields, this value becomes Before or On or After.	Field value is greater than or equal to ■ another field ■ a literal
Less Than or Equal To **Note:** For time and date fields, this value becomes On or Before.	Field value is less than or equal to ■ another field ■ a literal
Null	Field value is null or blank.
Not Null	Field value is not null or blank.
Min	Field value represents the minimal value of all the records in the cluster.
Max	Field value represents the maximal value of all the records in the cluster.
Longest	Field value is the longest value of all the records in the cluster. **Note:** ■ This value might not be unique as multiple records can have the same length, which would not lead to a single surviving record. Generally, a rule with this operator is followed by another rule to produce a single survivor. For more information, see Step 5h on page 62. ■ Applies to string data only.

Operator	Description
Shortest	Field value is the shortest value of all the records in the cluster. **Note:** ■ This value might not be unique as multiple records can have the same length, which would not lead to a single surviving record. Generally, a rule with this operator is followed by another rule to produce a single survivor. For more information, see Step 5h on page 62. ■ Applies to string data only.
Highest Occurrence	Field value occurs more frequently than other values in the records in the cluster. **Note:** This value might not be unique as multiple records can have the same frequency, which would not lead to a single surviving record. Generally, a rule with this operator is followed by another rule to produce a single survivor. For more information, see Step 5h on page 62.
Lowest Occurrence	Field value occurs less frequently than other values in the records in the cluster. **Note:** This value might not be unique as multiple records can have the same frequency, which would not lead to a single surviving record. Generally, a rule with this operator is followed by another rule to produce a single survivor. For more information, see Step 5h on page 62.

After selecting an operator, the **Value** field is enabled.

Note: Not all operators require values. If a unary operator, such as Max, is selected for the Operator value, the Value field is disabled.

c If enabled, click  to open the following dialog box:

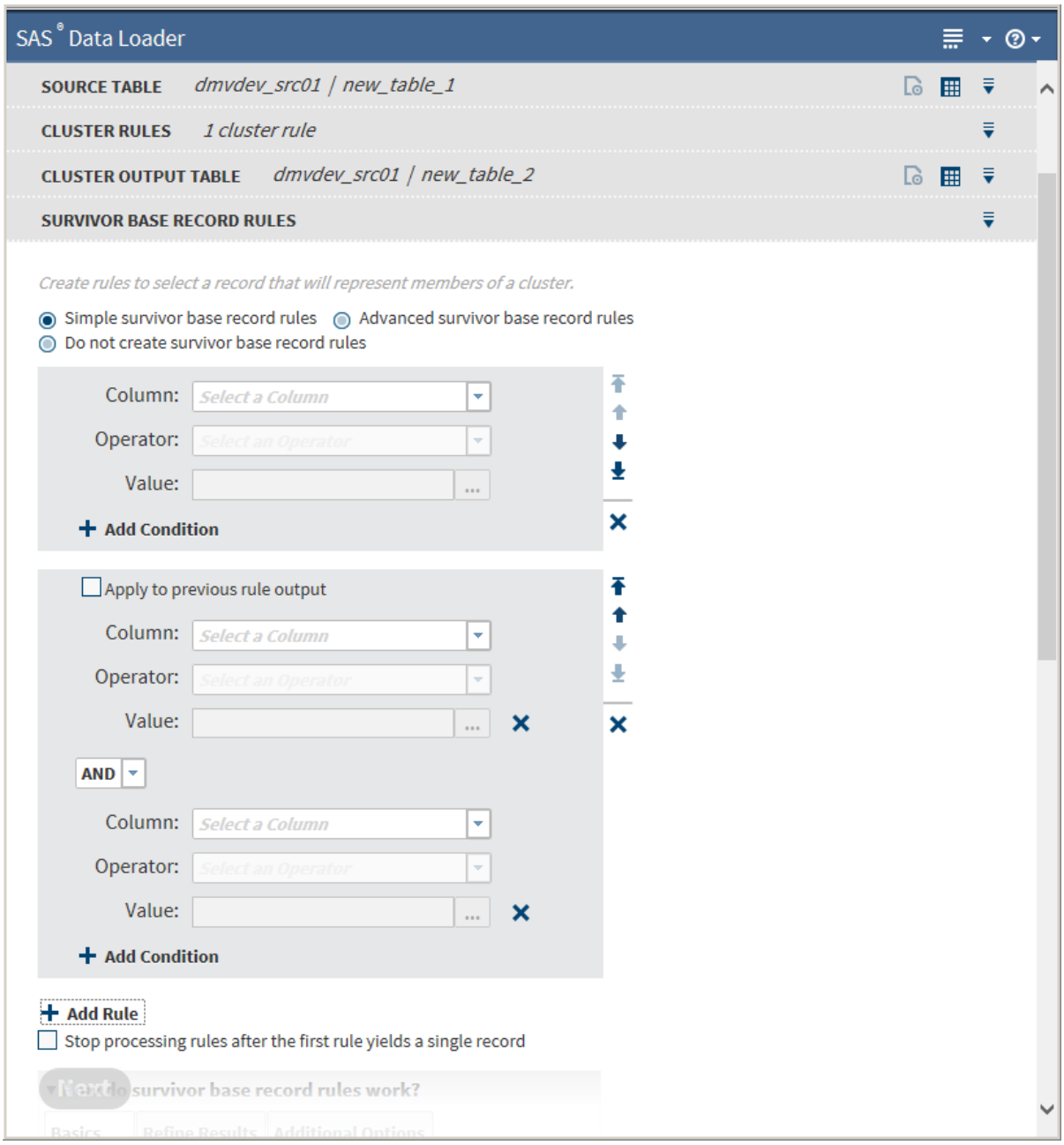


The dialog box presents several options for selecting the appropriate operator value. After selecting a value, click **OK**.

d If you have additional clauses in your rule, choose either the AND or the OR Boolean operator from the drop-down list between the clauses.

e To add another condition, click + next to **Add Condition**.

f To add another rule, click + next to **Add Rule** :



g Survivor base record rules are evaluated sequentially. To change the order in which a rule is processed, click the arrow icons to the right of the rule pane to move the rule up, down, or all the way to the top or bottom of the sequence.

h To process the output of a previous rule in the current rule, select **Apply to previous rule output**.

Because multiple records can have the same value depending on the operator, the previous rule might not lead to a single surviving record. Use **Apply to previous rule output** to refine the results further. If **Apply to previous rule output** is not selected, all the records from the source table are processed in the current rule.

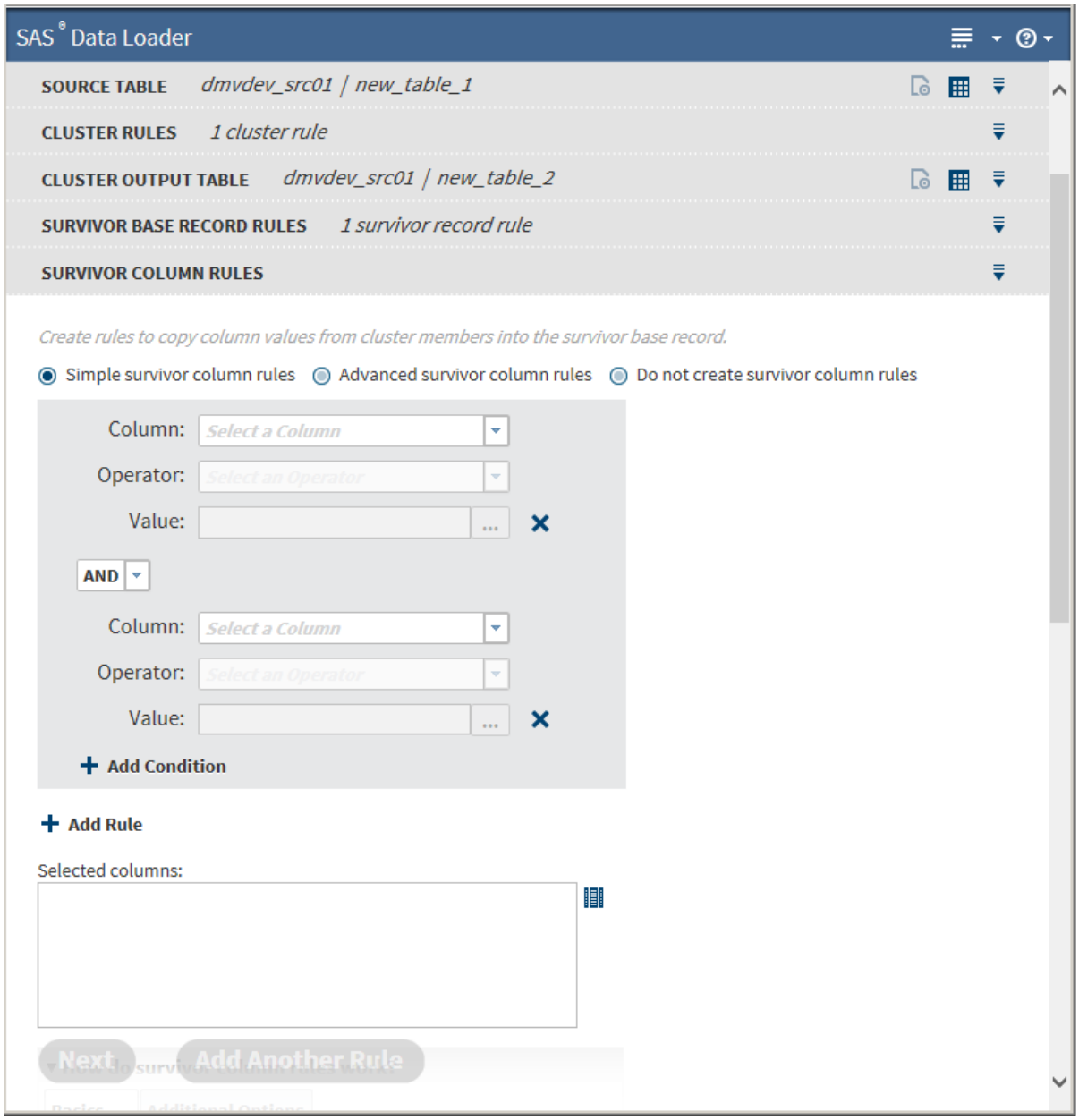
Note: This setting is turned off if you move the rule past another rule where the setting is turned off. If you choose to move the rule up or down sequentially, verify that **Apply to previous rule output** is correctly selected after you move the rule.

i If you want all rule processing to stop after a single record is produced and you want to use that single record as the survivor record, select **Stop processing rules after the first rule yields a single record**.

j To remove a rule, click ✕ next to the rule pane.

To use survivor base record rules that allow the specification of parentheses, click **Advanced survivor base record rules**. Parentheses use is the same as described in Advanced Cluster Rules on page 56.

Click **Next**. The Survivor Column Rules task is displayed:

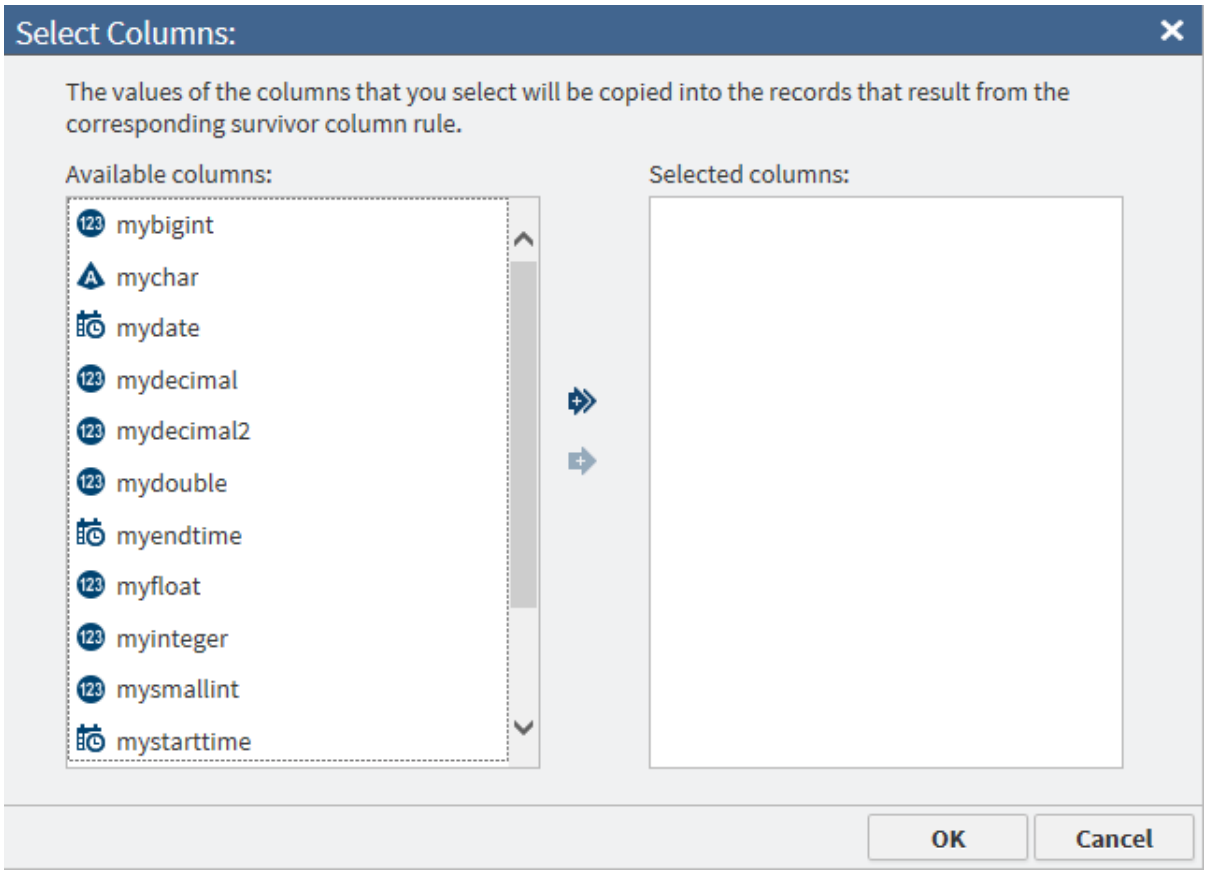


6 The initial Survivor Column Rules task defaults to **Simple survivor column rules**, which does not allow the specification of parentheses. You can choose to skip Survivor Column Rules by selecting **Do not create survivor column rules**, and then clicking **Next**.

To use survivor column rules that allow the specification of parentheses, click **Advanced survivor column rules**. Parentheses use is the same as described in Advanced Cluster Rules on page 56.

Basic operations are the same as for Survivor Base Record Rules. You must specify a minimum of one rule.

The only operation additional to those in Survivor Base Record Rules is the specification of a set of columns that are to be copied to the survivor record when the rule matches. Click to open the following dialog box:



Use the left and right arrow icons to select or remove columns, and then click **OK**.

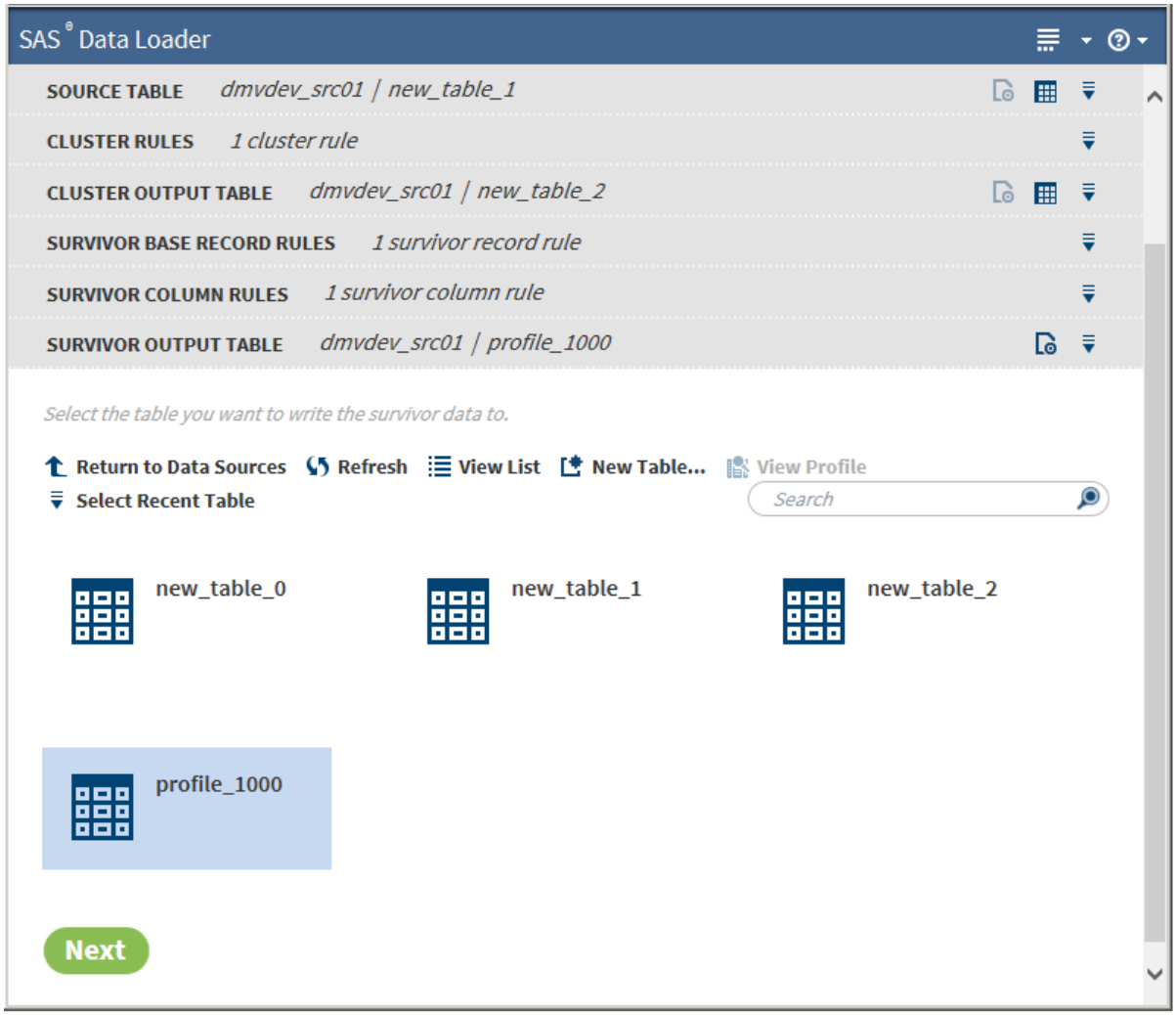
Because each set of rules can identify its own set of columns, the Survivor Column Rules task can be repeated in the directive. If you want to add another rule identifying its own set of columns, click **Add Another Rule**, which opens another Survivor Column Rules task.

Note:

- Survivor Column Rules tasks also run sequentially. If multiple tasks select different values for the same field, the value from the last task to run is incorporated into the survivor record.

- If you want to remove a Survivor Column Rules task after adding it, click on that task and select **Remove Survivor Column Rules Step**.

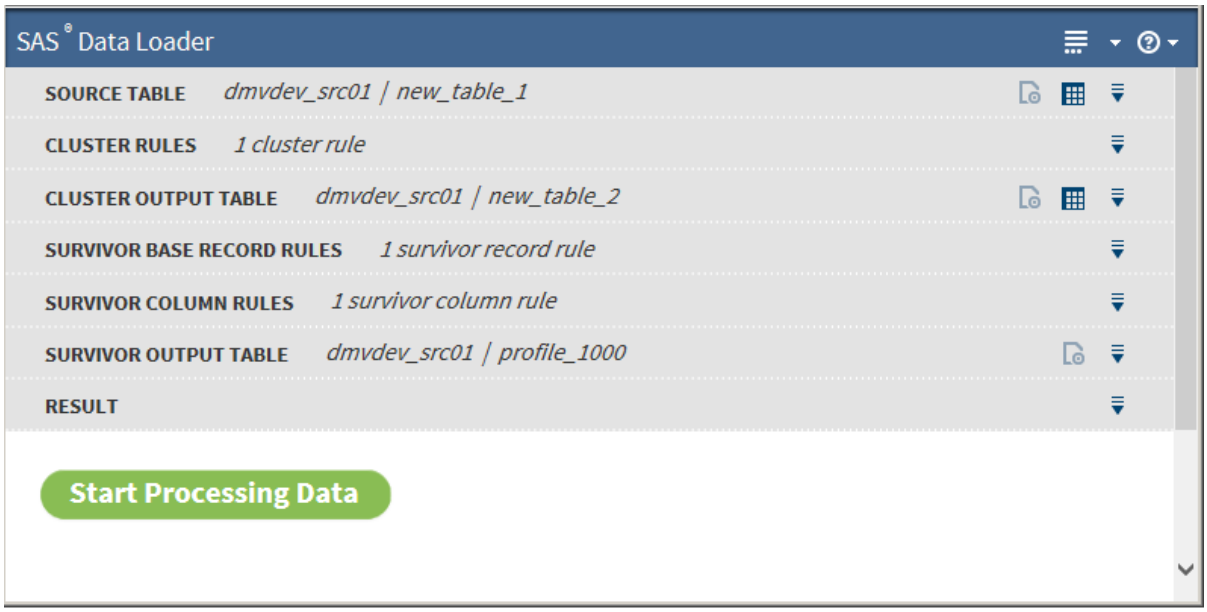
After you have completed adding rules and selecting columns, click **Next**. The Survivor Output Table task is displayed:



7 Select a survivor output table.

Note: If you select the same table for both clustering and survivorship output, the results from both steps are stored in this table.

Click **Next**. The Result task is displayed:



You can return to previous tasks if you need to make changes.

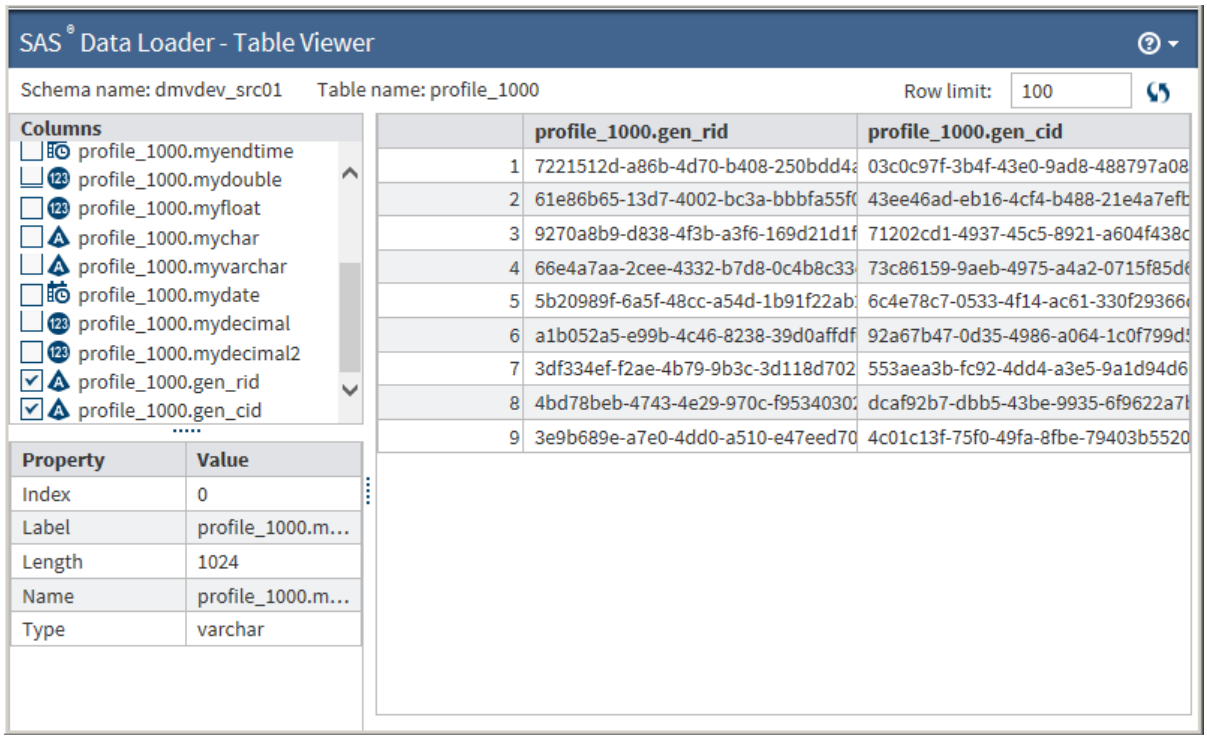
8 After you are satisfied that all the tasks are correct, click **Start Processing Data**, after which the results are returned:

Note: Leading and trailing spaces are trimmed from fields when evaluating survivorship rules.

After the job has completed, you can view the results, the log file, and the code that ran in Hadoop.

9 Click **View Results** on the Result task to see the table results. If you created two output tables, select one after clicking **View Results**, which opens the

selected table in the Table Viewer. If you created only one table, clicking **View Results** opens it in the Table Viewer:



The records in the table are not presented in any particular order. You can select which columns to view in the Columns pane and click the table headers to sort on a column.

Two reserved fields are appended to the table: gen_cid and gen_rid. Gen_cid represents the generated cluster ID and gen_rid represents the generated record ID.

- If you reprocess the output tables through this directive, filter these two fields as desired because they are generated again during the reprocess.
- Records in the cluster output table that belong to the same cluster have the same gen_cid. Sorting on the gen_cid column in the cluster output table enables you to view records that belong to the same cluster.
- You can join the cluster output table and survivor output table on the gen_cid field to identify the contributors that correspond to the survivor record.
- If you need to address a particular record, the gen_rid result is unique across all the records.

Match-Merge Data

Introduction

Match-Merge Data
Match-merge rows from one or more source tables into a single row and output a single target table

Use the Match-Merge Data directive to combine rows from two or more source tables into a single row in a target table. Rows are combined according to the values of one or more matched columns.

Matched columns are common to all source tables. Matched columns require compatible basic data types of numeric or character.

Rows are merged when values match in specified merge-by columns in two or more source tables. If you specify more than one merge-by column, then target rows are grouped, with one subgroup for each merge-by column.

The source table that provides the data for the merged row is determined by an ordered list of source tables. The source table that contributes merged data is the last table in the list that contains a matching merge-by value.

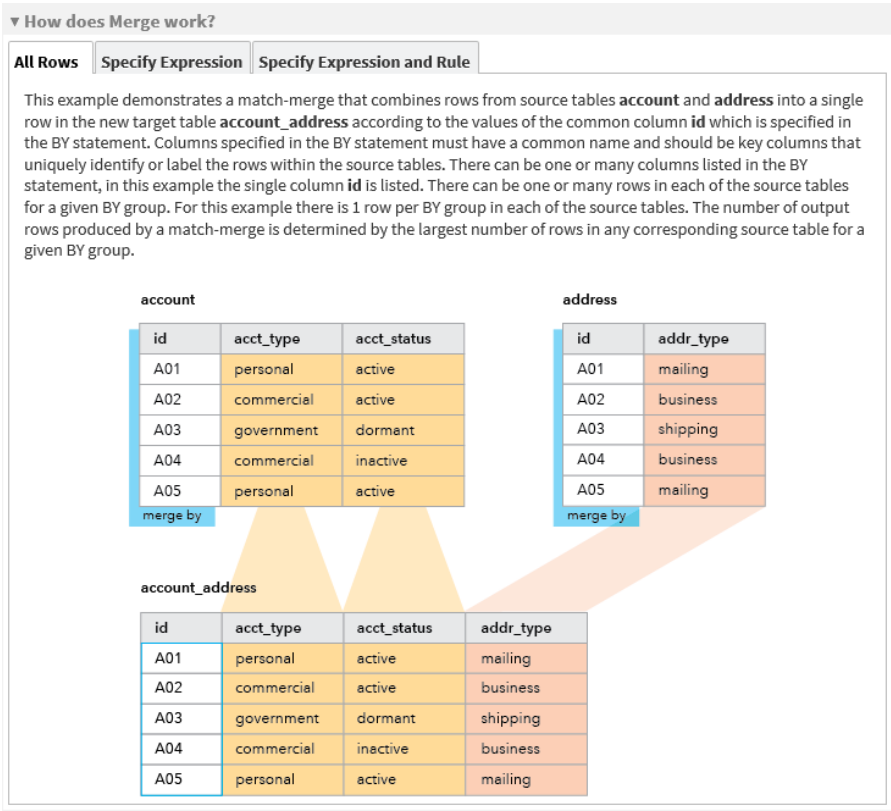
In the target, the merged row receives values from selected matching columns. Values from specified unmatched columns are also included in the merge.

Unmatched columns from any source table can be renamed and added as target columns. The target receives data from unmatched columns in merged rows and in rows that carry over without a merge.

You can define a filter to exclude unwanted rows from the target. The filter is a user-written SAS DS2 expression. If the expression evaluates to true, then the row is written to the target. The **Advanced Editor** expression builder in the **Filter Rows** task displays DS2 functions, provides syntax help, and enables you to add functions to your expression with a click.

You can add new columns to the target that contain calculated values for each target row. The calculations are specified as user-written DS2 expressions.

To learn more about the match-merge process, see **How does Merge work?** in a new Match-Merge directive. The Help information is provided in the **Order Source Tables** task, after you select source tables.



Example

Introduction

Follow through this section using your own source tables to create and run a job using the Match-Merge Data directive.

If you have a question about a particular task in the directive, refer directly to that section.

Select Source Tables

The Match-Merge Data directive opens in the **Source Table** task. The initial display lists the tables in the Hive default data source, or in the data source that you last accessed in your current session.

Select a source table, and then click **Next** to select the second table. To add another table, click **Add Another Source Table**.

To select a source table from a different data source, click **Return to Data Sources**.

For more information about data sources and tables, see “Viewing Data Sources and Tables” on page 20.

When your list of tables to be merged is complete, click **Next** to display the **Order Source Tables** task.

Change Maximum Character Length

At any point in the use of the Match-Merge directive, you can change the maximum length of character values as they are read, processed, and output by SAS. The source values are not changed in Hadoop.

When you change the maximum character length, the change applies to a single source table. The default maximum character length (1024) is not changed. You can change the maximum character length for all of the source tables in the match-merge job.

To change the default maximum character length in SAS, see "Change the Maximum Length for SAS Character Columns" on page 195.

Follow these steps to view or change the maximum character length:

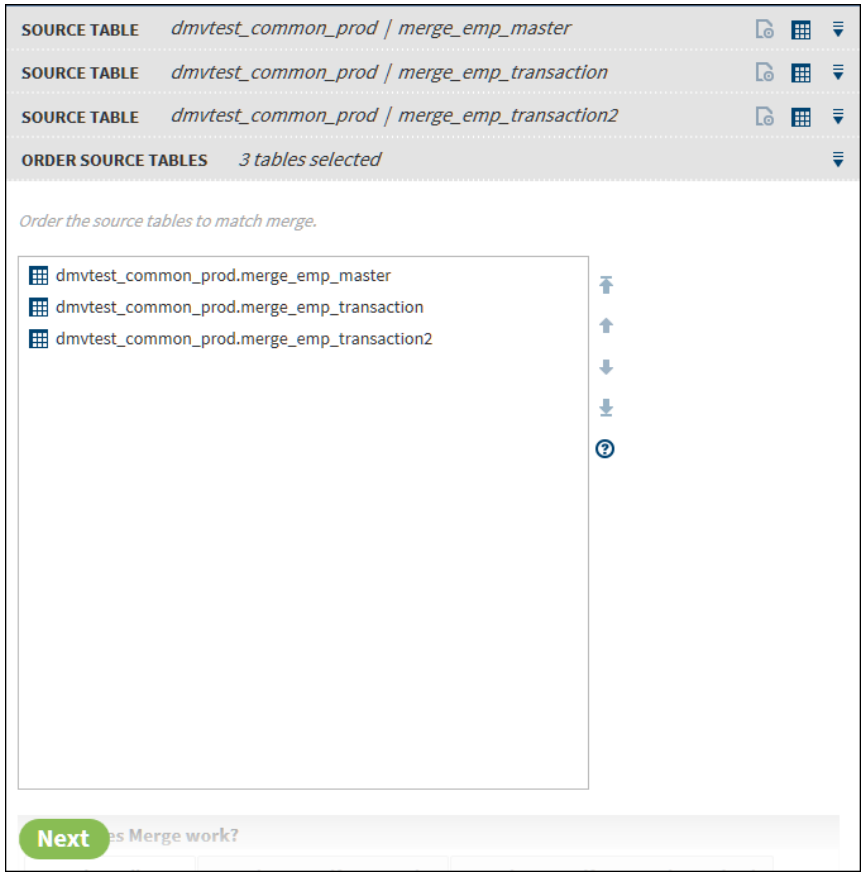
1. If necessary, click a **Source Table** taskbar at the top of the directive.
2. In the taskbar, click  .
3. In the Directive Settings window, view or change the value **Maximum length for SAS character columns**.

Order Source Tables

Use the **Order Source Tables** task to determine the source table that provides target data for a given merged row. The source table that provides data for a merged row is the last table in the list that contains a matching merge-by value for the given row.

In the **Order Source Tables** task, the top table is the first in the list. The bottom table is the last in the list.

The default order of tables in the **Order Source Tables** task replicates the order in which the source tables were selected. To change the default order of tables, click the up or down arrow icons.



When the order of tables is complete, click **Next** to open the **Matched Columns** task.

Matched Columns

Use the **Matched Columns** task to identify the columns in each source table that have similar content. Some or all of the matched columns are merged in the target. One (or more) will be identified as the merge-by column.

To be functional, matched columns need to have the same name, and their data types must resolve to either numeric or character. The Matched Column task helps you rename source columns and match data types.

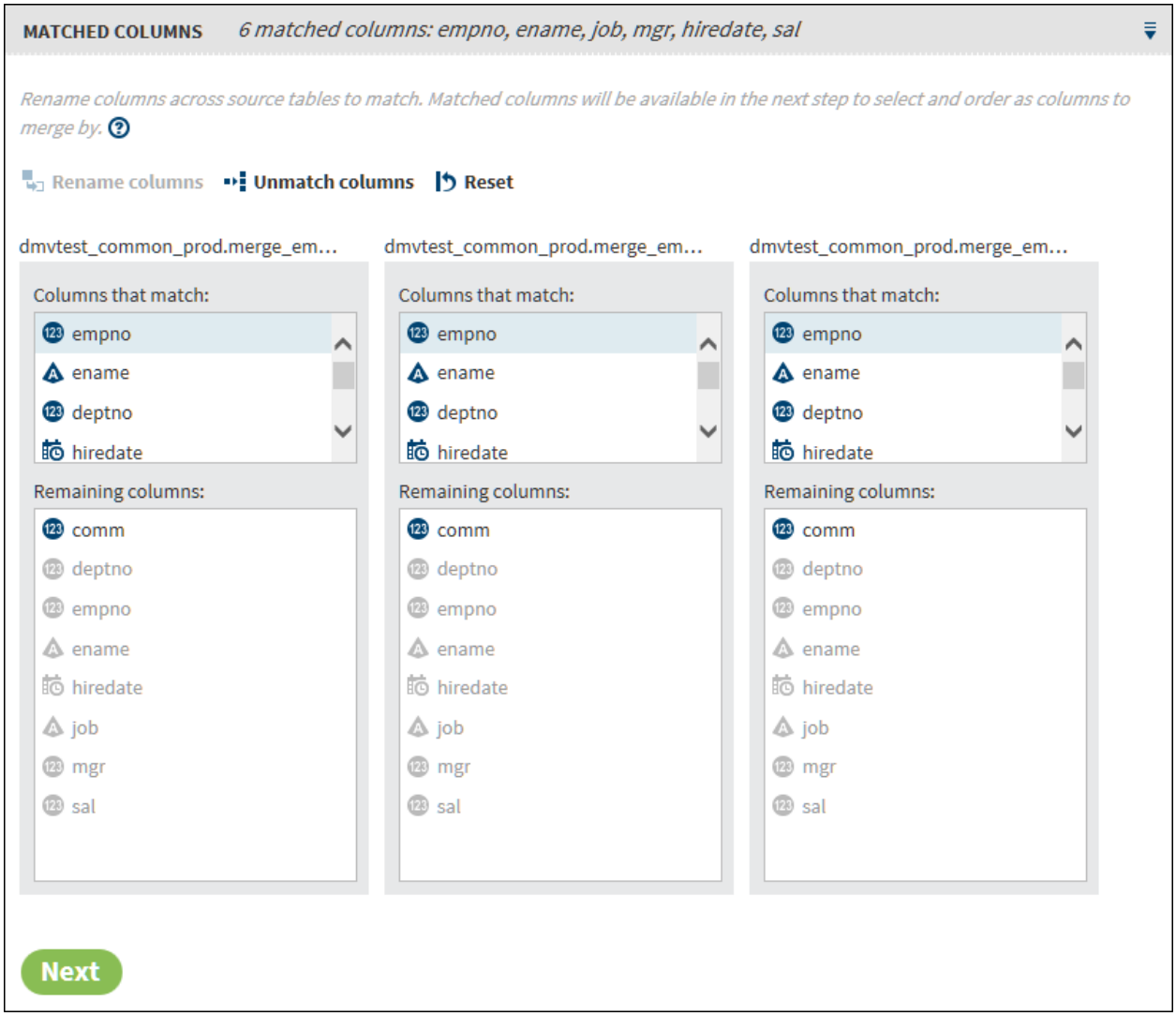
Note: The Matched Columns task displays by default all columns in all sources that have a matching name and type.

Column data types can differ. For example, an INTEGER column can be matched with a BIGINT column. Similarly, a CHAR type can be matched with a VARCHAR type. The type of the associated target column is the largest or longest of the initial input types.

When column types match, but names do not, select one column in each source table, and then click **Rename columns**. Enter the name of the column as it will appear in the target, or accept the default name from the initial source table.

To remove a matched column after you define it, click any instance of the column name in **Columns that match**, and then click **Unmatch columns**.

To remove all matched column definitions, click **Reset**.



When the matched columns are defined, click **Next** to open the **Merge By** task.

Merge By

Use the **Merge By** task to specify the matched columns that uniquely identify rows in the source tables. Rows that match in the merge-by column are merged into a single row in the target. All rows with unique values in the merge-by column will appear in the target table.

The merge-by column is similar in purpose to a primary key in a database table.

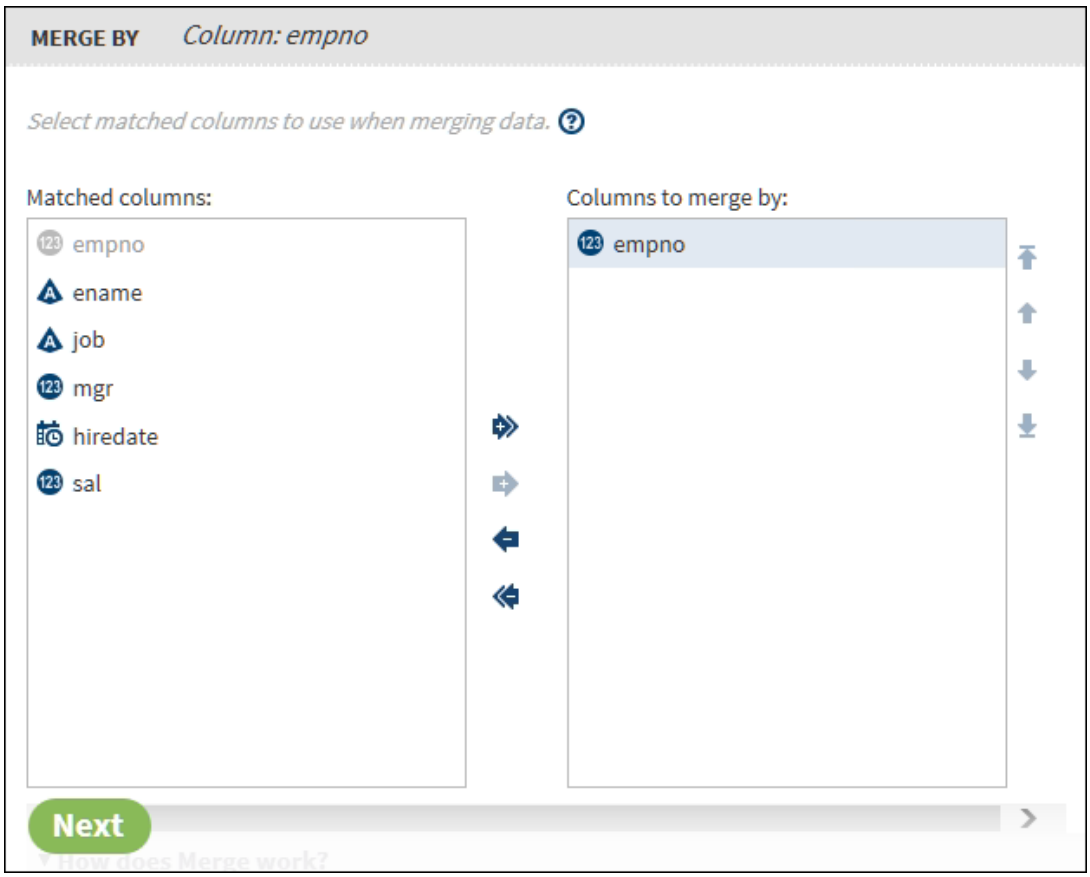
Merge results can be unpredictable when the selected columns present something other than a one-to-one merge. Seek to define your merge-by columns so that a match identifies a single row in the merge tables. If a given merge table contains several matching rows for a single row in the initial table, then the results of the merge can be unpredictable.

If you select more than one merge-by column, then the order of the columns specifies a group-by arrangement of the rows in the target table. The group-by arrangement arranges target rows by group and subgroup.

CAUTION! Specifying more than two merge-by columns produces incorrect results. To learn more about merge-by columns, see the *SAS DS2 Language Reference*.

Note: If a source table contains two or more instances of the same merge-by variables, then the result of the merge is nondeterministic. In other words, the merge can produce more than one correct result. To generate fully repeatable results, ensure that your source tables contain no more than one row for each set of merge-by variables.

Use **Columns to merge by** to change the order of the merge-by columns. Select columns and click the vertical arrow icons.



After you select and order your merge-by columns, click **Next** to open the **Input Columns** task.

Input Columns

Use the **Input Columns** task to specify the source columns that will be merged in the target.

Initially, **Available columns** lists all source columns, except for the merge-by columns. The source tables appear in merge-by order, from left to right, as defined in the **Order Source Tables** task. The values that are written to the target come from the participating source table that is position farthest to the right.

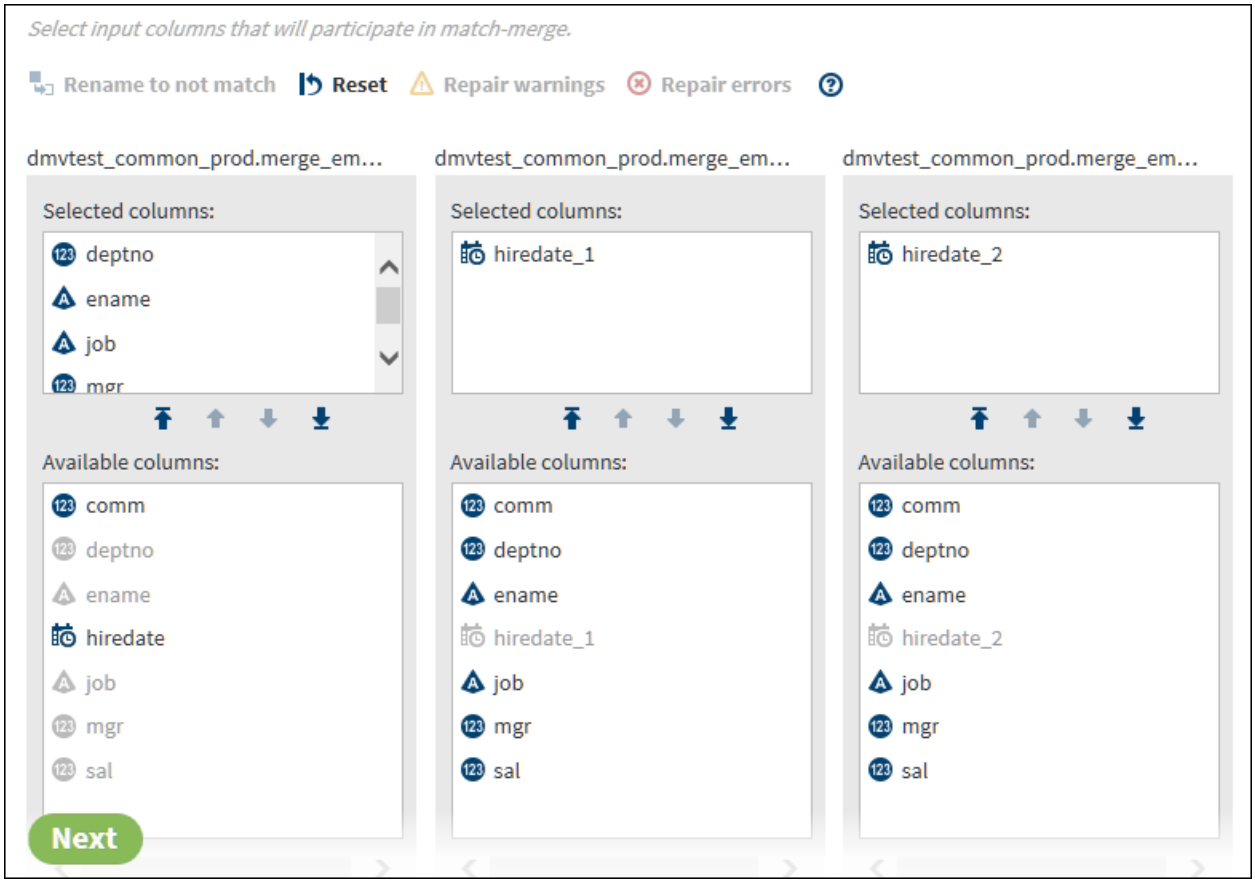
To specify the columns that will be merged in the target table, move columns from **Available columns** to **Selected columns**.

TIP To see the full name of a source table, position the cursor on the abbreviated name.

To include in the target matched columns or columns with the same name, move the columns to **Selected columns** and then click **Rename to not match**.

To automatically rename and not match two or more instances of a matched column, move those instances into **Selected columns**. This operation adds a repair warning icon to each instance of the selected columns. To rename the columns automatically and remove the warning icon, click **Repair warnings**.

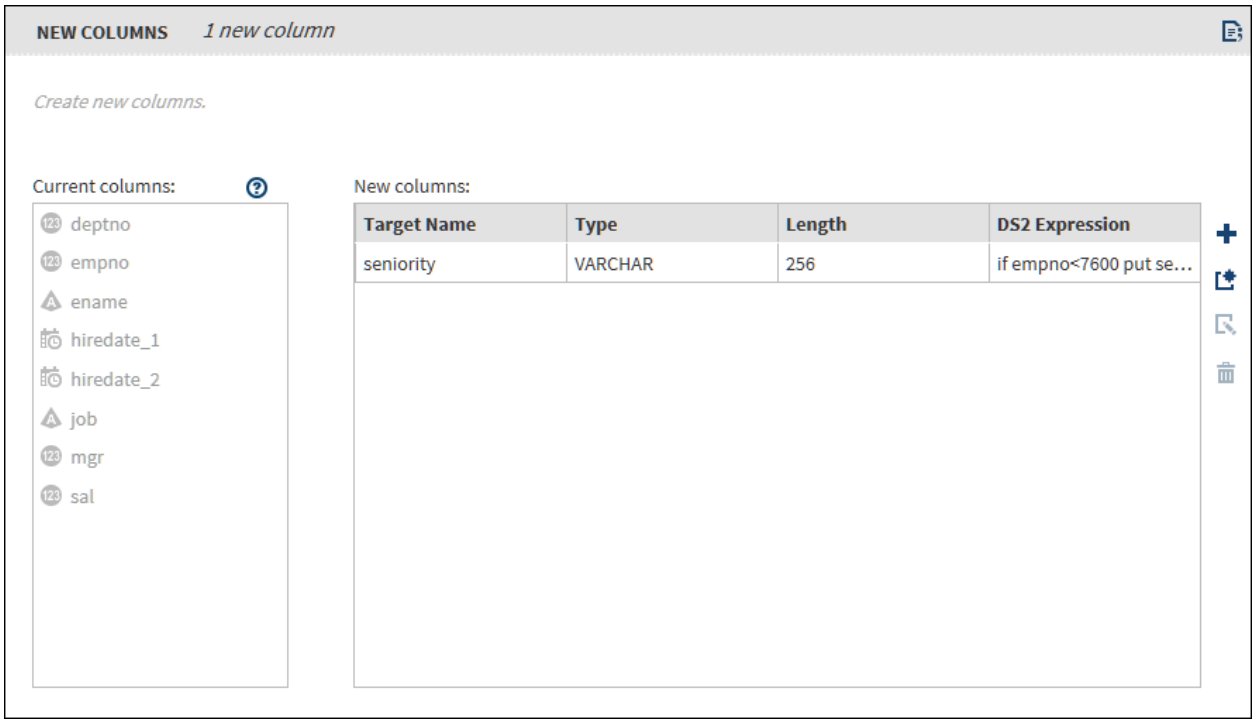
The following image depicts six columns that will be merged, two of which will be renamed.



When you have selected your input columns, click **Next** to open the **New Columns** task.

New Columns

Use the **New Columns** task to define target columns that receive the results of user-written DS2 expressions.



To add a new target column and paste or enter a DS2 expression, click +.

Note: If your expression contains more than one clause, see "Develop Expressions for Directives".

To add a new row and create an expression in the Advanced Editor window, click

To create an expression using the Advanced Editor window, click a new row and then click .

Filter Rows

Use the **Filter Rows** task to exclude rows from the target table by specifying one or more DS2 expressions.

Initially, **All Rows** is selected, which indicates that no rows will be filtered from the target. To continue to the next task without filtering rows, click **Next**.

To filter with a SAS DS2 expression, click **Specify expression**.

To create a rule that limits the rows to which the DS2 expression will be applied, click **Specify rule to indicate**. The rule tests each matching row to determine the source tables that include or do not include that row. If the rule is true for a row, then the DS2 expression is evaluated for that row. If the rule is not true, then the DS2 expression is not evaluated and the matching row is written to the target. You can create and apply multiple DS2 expressions. Each DS2 expression can have its own rule.

To specify a rule, select **In** or **Not In** for a selected source table. Click **Add condition** to apply a logical **AND** or **OR** and to specify a second source table. Continue to add conditions as needed.

To create a DS2 expression, you can paste an existing expression or enter the expression into the DS2 expression text box. When adding an expression, note

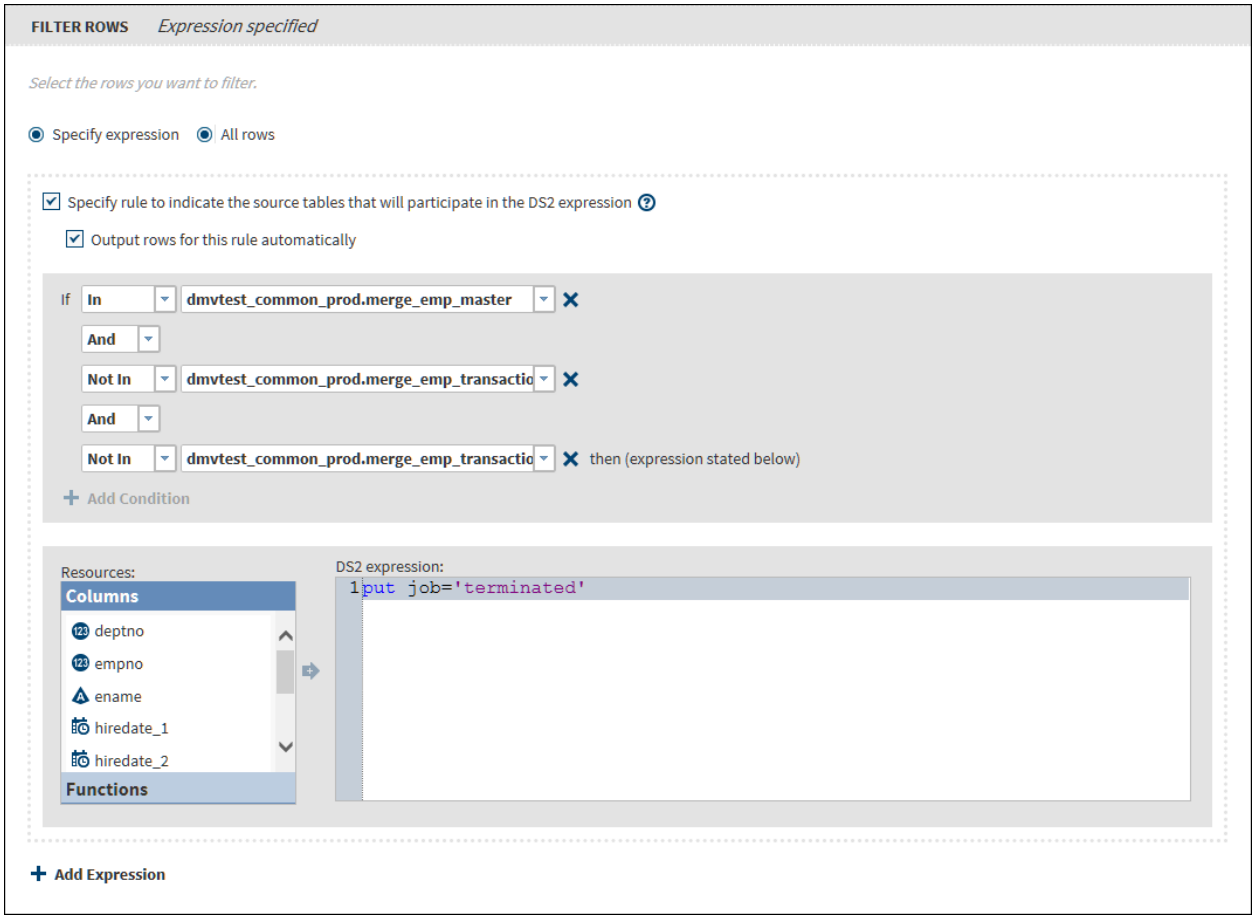
that the entire expression is inserted into SAS code. The expression should be syntactically correct with all the statements ending properly with a semi-colon. SAS Data Loader does make one exception to this rule: it adds a semi-colon at the end of an expression if there is no semi-colon anywhere in the expression.

To paste a previously copied DS2 expression, click **DS2 expression** and press Ctrl+V or your equivalent.

Note: If your expression contains more than one clause, see "Develop Expressions for Directives".

To enter a DS2 expression, use the column names and DS2 functions in the **Resources** list box.

To add a second DS2 expression, click **Add expression**. Multiple expressions are evaluated in the order in which they appear in the **Filter Rows** task. The top expression is evaluated first, the bottom expression last.



When your **Filter Rows** task is complete, click **Next** to display the **Management Output Columns** task.

Manage Output Columns

Use the **Manage Output Columns** task to reorder or remove the columns in the resulting target table.

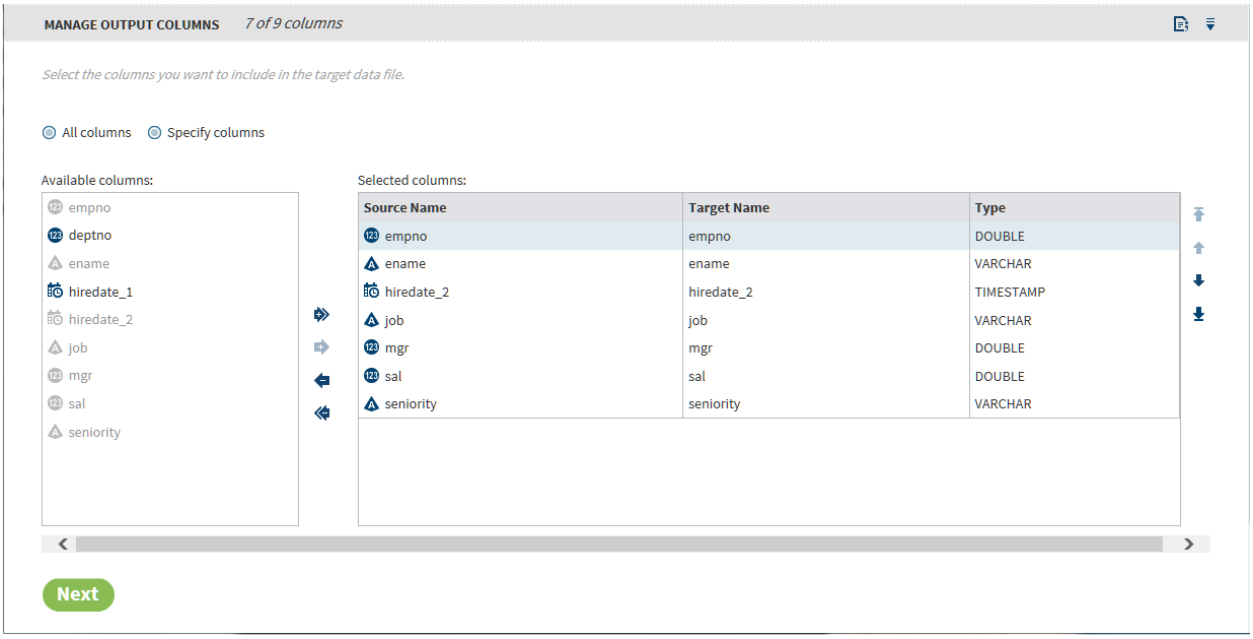
Initially, **All columns** is selected. To not reorder or remove target columns, click **Next**.

Note: To ensure data integrity, the first column in the target is required to be the first merge-by column, as defined in the **Merge By** task. The default order of columns is alphanumeric by column name.

Click **Specify columns** to display an alphabetic list of target columns. Included in the list are columns that were renamed in the **Input Columns** task. Also included are any new columns that were added to receive the results of DS2 expressions. New columns are added in the **New Columns** task.

To reorder a column, select it and then click a vertical arrow icon.

To remove a column, select it and click the left arrow icon.



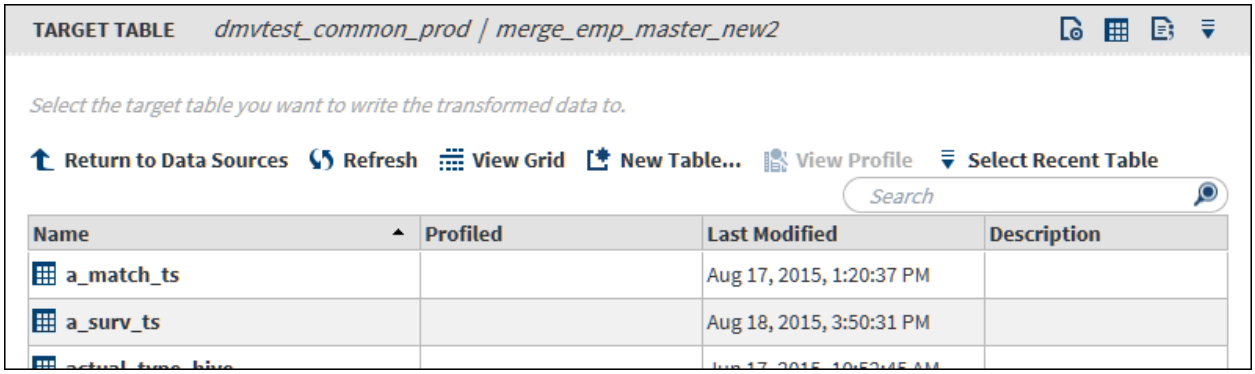
When your target columns are properly selected, named, and ordered, click **Next** to move to the **Target Table** task.

Target Table

Use the **Target Table** task to select a new or existing target table for your match-merge job. If you select an existing table, it will be completely overwritten by the match-merge job.

To select or create a target table in a different data source, click **Return to Data Sources**.

For further information, see “Browse Tables” on page 26.



After you select your target table, click **Next** to open the **Result** task.

Result

Use the **Result** task to run your match-merge job, and examine the resulting target table, generated code, error messages, and log file.

Delete Rows

Introduction

Delete Rows
Delete rows from a selected table.
Requires Hive 14 or above.

Use the Delete Rows directive to delete data from a selected source table. Data is deleted in the source table itself, rather than in a separate target table.

Prerequisites

The prerequisites for the Delete Rows directive are defined as follows:

- The Hadoop cluster needs to be configured with release 0.14.0 or later of the Apache Hive data warehouse software. This release supports transactional tables.
- Source tables must use a Hive file format, preferably ORC (Optimized Row Columnar.)
- Source tables must be bucketed and partitioned. Bucketing clusters data based on the values in a specified (key) column. Partitioning creates individually accessible subsets of data based on the values in one or more source columns. To determine whether a source table has been bucketed and partitioned, contact your Hadoop administrator.

Example

Follow these steps to use the Delete Rows directive:

1 On the SAS Data Loader directives page, click **Delete Rows**. The **Source Table** task is displayed. For more information about data sources and tables, see "Viewing Data Sources and Tables" on page 20.

2 In the **Source Table** task, select a data source and click **Next**, or click Select Recent Table . Refer to the prerequisites as needed.

3 In the **Delete Rows** task, choose one of the following:

   a To delete all of the rows in the source table, click **All rows** and then click **Next**.

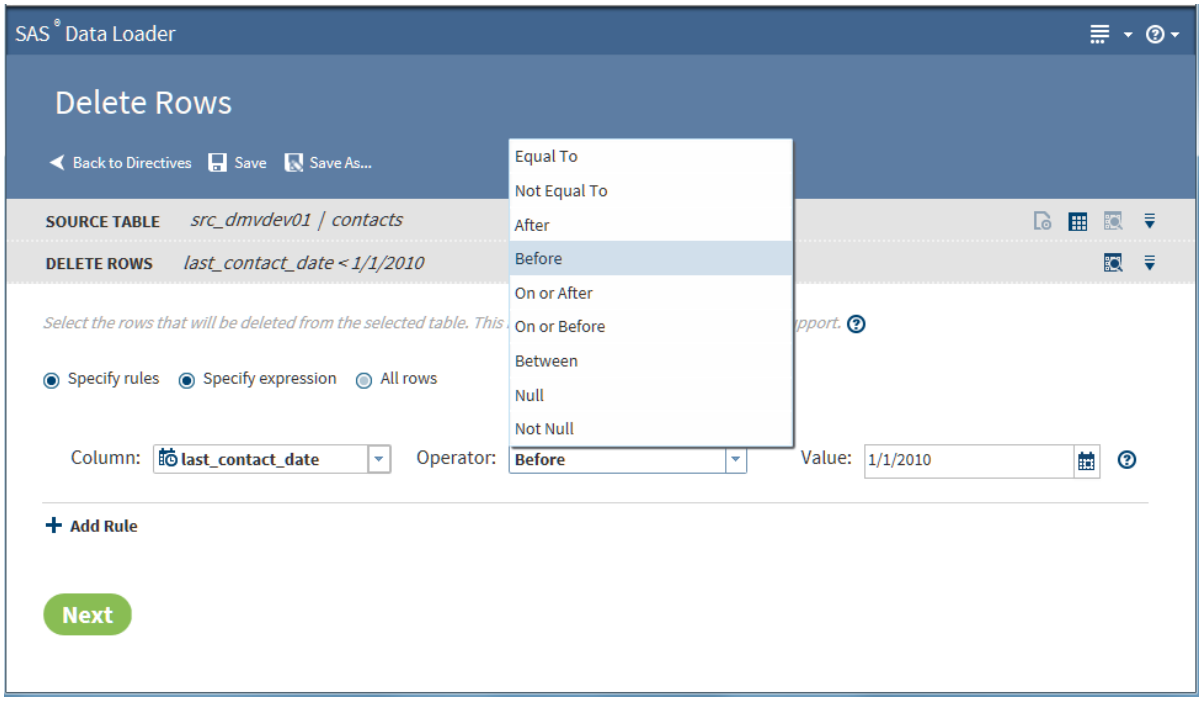
   b To delete rows using one or more rules, click **Specify rules** and proceed to the next step. The Delete Rows job deletes rows when the specified rules are true. Multiple rules can be applied with logical AND and OR operators.

   c To delete rows using a Hive expression, click **Specify expression** and go to Step 5 on page 80. Rows are deleted when the Hive expression returns true.

4 To delete rows by specifying one or more rules, follow these steps.

   a Click **Select a column** and choose the source column that forms the basis of your rule.

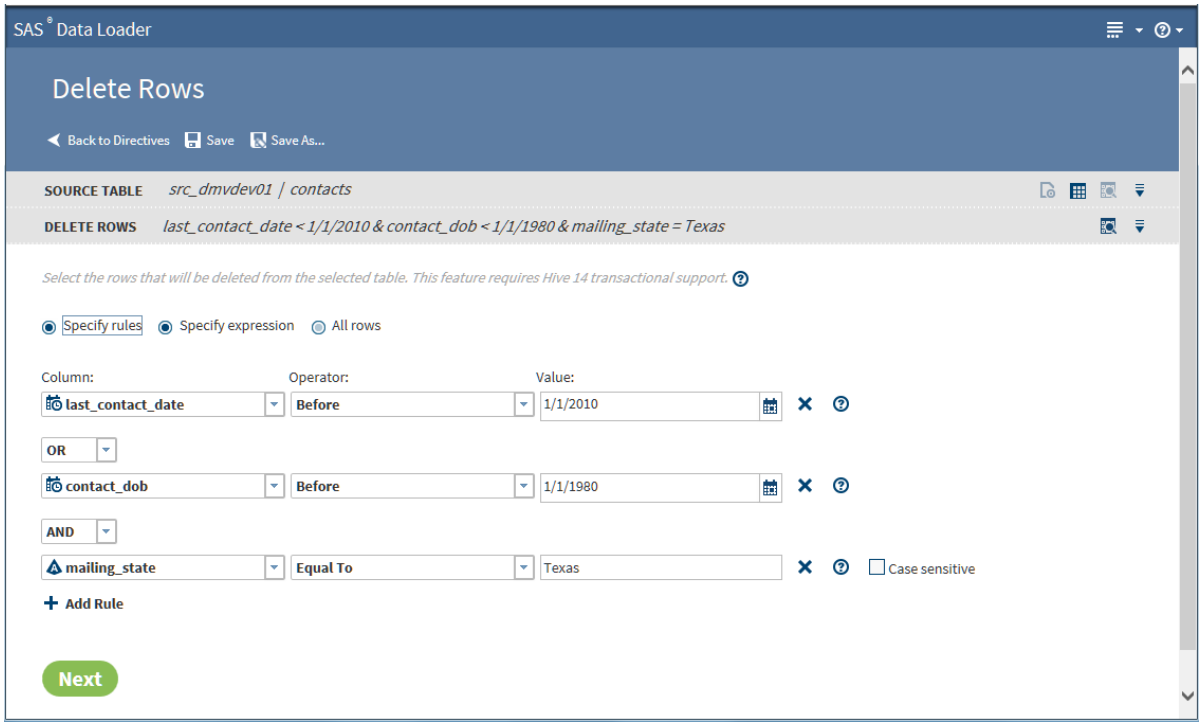
   b Click and select a logical **Operator**. The operators that are available depend on the type of the data in the source column. For example, the following image shows the operators that are available for the date/time data type:



   c In the **Value** field, add the source column value that completes the expression. In the preceding example, the rule can be read as "Delete from the source table all rows with a last contact date prior to January 1, 2010."

d Click **Add Rule** to add another rule. Select a different column, operator, and value.

e To delete rows when either the new rule or the preceding rule are true, change the **AND** condition to **OR**.



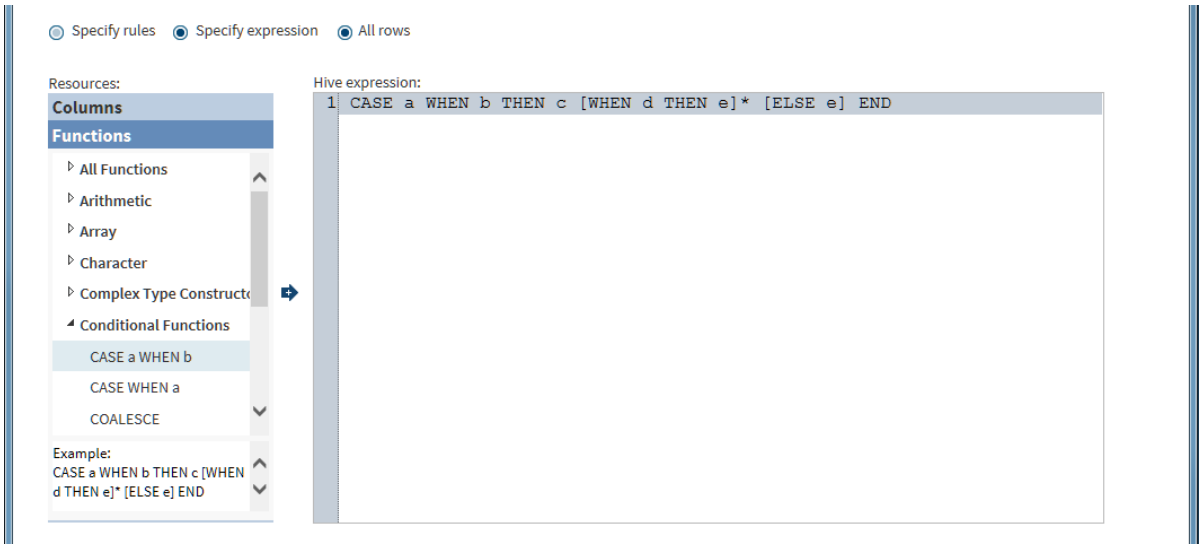
f When your rules are complete, click **Next** and go to Step 6 on page 81.

5 To delete rows using a Hive expression, follow these steps:

a In the **Hive expression** text box, either type or paste a Hive SQL expression.

Note: If your expression contains more than one clause, see "Develop Expressions for Directives".

b To add Hive SQL functions to your expression, click **Functions**, expand a category, select a function, and click .



To add column names to your expression, position the cursor in the **Hive expression** box, click **Columns** in the **Resources** box, click the source column, and then click .

6 When you have specified a rule or a Hive expression, click **Next**.

7 In the **Code** task, review the Hive code that will run in Hadoop. Click **Edit HiveQL Code** as needed.

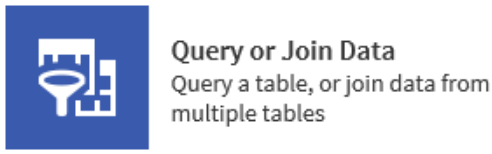
Note: When you edit the Hive expression in the **Code** task, you will lose those edits if you then change the content of the **Delete Rows** task.

8 Click **Next** to open the **Result** task, and then click **Start deleting data**.

9 When the job is complete, click **Log** to confirm the deletion of rows.

Query or Join Data

Introduction



Use queries to group rows based on the values in one or more columns and then summarize selected numeric columns. The summary data appears in new columns in the target table. You can also filter rows, sort columns, select and revise columns, and use expressions to modify data or add new columns.

Use joins to combine source tables. The join is based on a comparison of values in “join-on” columns that are selected for each of the source tables. The result of the join depends on matching values in the join-on columns, and on the selected type of the join. Four types of joins are available: inner, left, right, and full.

The Query or Join Data directive enables you to create jobs that execute a single query or join, or combine multiple joins. In the resulting table, you can remove unwanted rows and columns, remove duplicate rows, and rearrange columns. Before you execute the job, you can edit the generated SQL code and paste-in additional SQL code. The process of the directive is defined as follows:

- Select a source table.
- Join tables to the initial table as needed.
- Define summarizations that group columns and aggregate numeric values, again as needed.
- Use rules or expressions to filter unwanted rows from the target.
- Select, rename, rearrange, and change type and length of target columns.
- Apply SQL expressions to modify existing columns or add data to new columns.
- Sort target rows based on specified target columns.

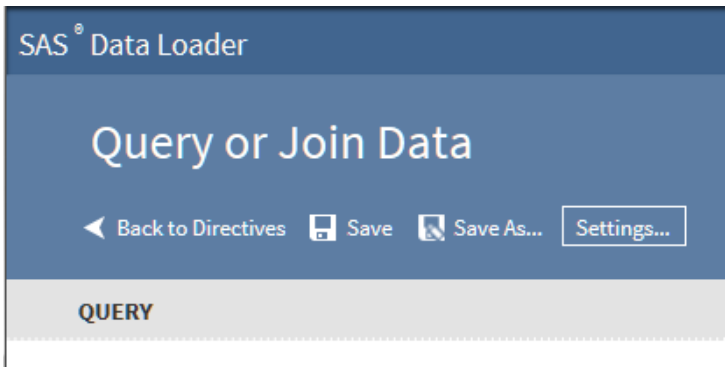
Enable the Cloudera Impala SQL Environment

Support for the Cloudera Impala SQL environment is enabled in the **Hadoop Configuration** panel of the **Configuration** window. When Impala is enabled, new instances of the following directives use the Cloudera Impala SQL environment by default:

- Query or Join
- Sort and De-Duplicate
- Run a Hadoop SQL Program

The default SQL environment can be overridden using the **Settings** menu. To learn more about SQL environments, see “Enable Support for Impala and Spark”.

Note: Changing the default SQL environment does not change the SQL environment for saved directives. Saved directives continue to run with their existing SQL environment unless they are opened, reconfigured, and saved.



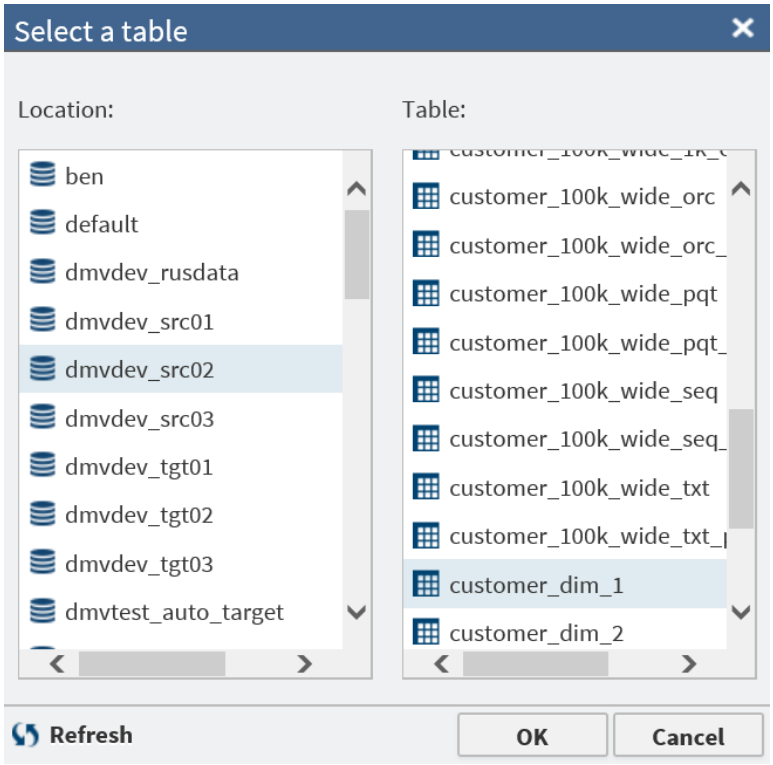
Example

Follow these steps to use the Query or Join Data directive.

1 On the SAS Data Loader directives page, click **Query or Join Data**.

2 In the **Query** task, click the browse icon ....

3 In the Select a Table window, scroll through the **Location** list and click a schema. Then click a source table in the **Table** list, and then click **OK**.



4 If your job includes no joins, click **Next** to open the **Summarize Rows** task.

5 To join your source table with other tables, click **Add Join**, and then click **Next**.

6 In the **Join** row, click the browse icon ... and select the table for the join.

7 As needed, click the **Join** field and select a join type other than the default join type **Inner**.

Inner
: The inner join finds matching values in the join-on columns and writes one row to the target. The target row contains all columns from both source tables. A row from either source table is not written to the target if it contains a null value in the join-on column. A row is also not written to the target if the value in the join-on column does not match a value in the join-on column in the other source table.

Left
: The left or left-full join writes to the target all rows from the left table of the join statement. If a match does not exist between the join-on columns, null values are written to the target for the columns of the right table in the join.

Right
: The right or right-full join reverses the definition of the left join. All rows from the right table appear in the target. If no values match between the

join-on columns, then null values are written to the target for the columns of the table on the left side of the join statement.

Full
The full join combines the left and right joins. If a match exists between the join-on columns, then a single row is written to the target to represent those two source rows. If the left or right table has a value in the join-on column that does not match, then the data for that row is written to the target. Null values are written into the columns from the other source table.

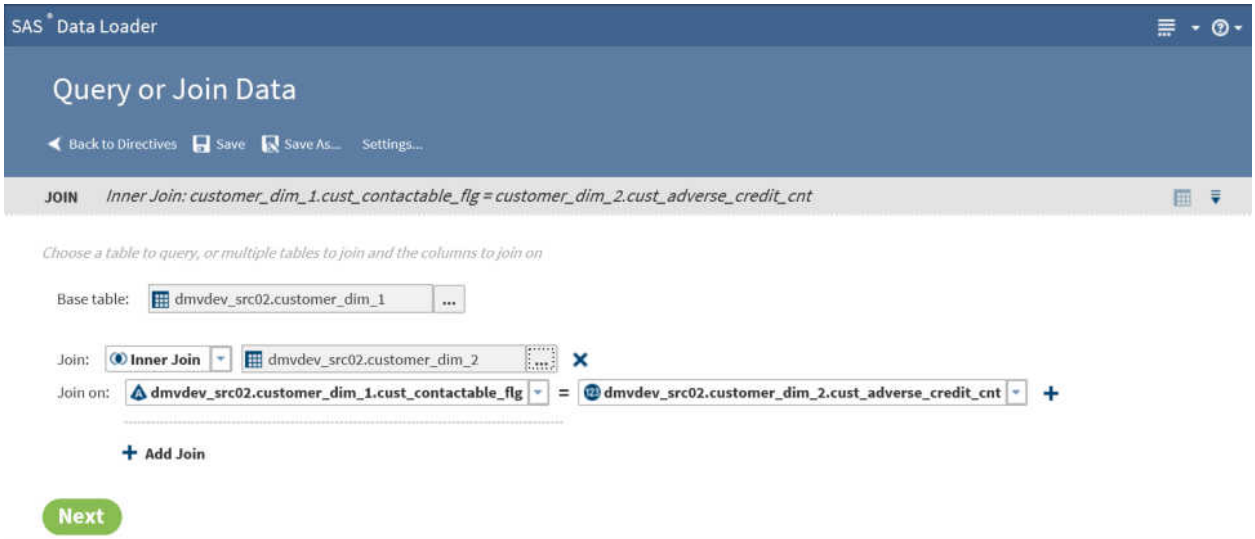
8 In the **Join-on** row, click the left join-on column and select a replacement for the default column, as needed.

Note: The left and right designations in the join-on statement define the output that is generated by the available left join and right join.

9 Click the right join-on column to select a replacement for the column, as needed.

10 To add more join columns, click the Add icon + at the end of the **Join-on** row. A match between the source tables consists of a match in the first pair of join-on values *and* a match between the second pair of join-on values.

11 To join a third table to the joined table that unites the two source tables, click **Add join**.



12 Click **Next** and wait a moment while the application assembles in memory the names of the joined columns.

13 In the **Summarize Rows** task, if you do not need to summarize, click **Next**.

Note: If your source data is in Hive 13 (0.13.0 or lower,) the **Summarize Rows** task will not handle special characters in column names. To resolve the issue, either rename the columns or ask your Hadoop administrator to upgrade to Hive 14 (0.14.0 or higher.)

14 To add summarizations, click the **Group rows by** field, and then click the column that you want to use as the primary grouping in your target table. For

example, if you are querying a table of product sales data, then you could group rows by the product type column.

Note:

- If your job includes joins, note that the **Group rows by** list includes all columns from your source tables.
- If you intend to paste an SQL query (HiveQL or Cloudera Impala SQL) into this directive, then you can click **Next** two times to display the **Code** task.

15 To subset the first group with a second group, and to generate a second set of aggregations, click **Add column**.

16 To generate multiple aggregations, you can add additional groups. The additional groups will appear in the target table as nested subgroups. Each group that you define will receive its own aggregations.

To add a group, click **Add Column**, and then repeat the previous step to select a different column than the first group. In a table of product sales data, you could choose a second group by selecting the column product_code.

17 In **Summarize columns**, select the first numeric column that you want to aggregate.

18 In **Aggregation**, select one of the following:

Count
: specifies the number of rows that contain values in each group.

Count Distinct
: specifies the number of rows that contain distinct (or unique) values in each group.

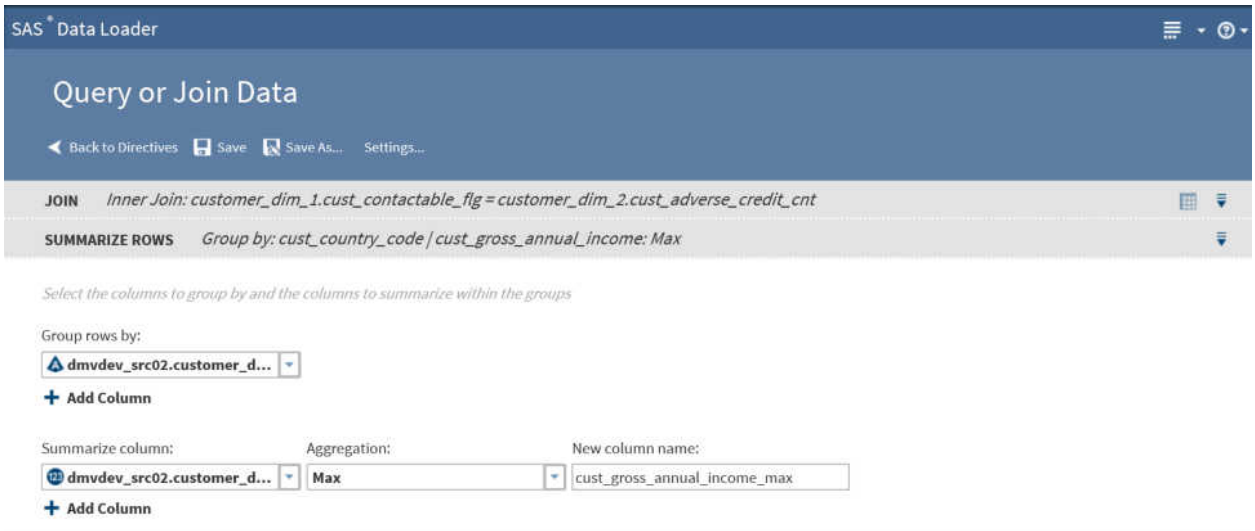
Max
: specifies the largest value in each group.

Min
: specifies the smallest value in each group.

Sum
: specifies the total of the values in each group.

19 In **New column name**, either accept the default name of the aggregation column, or click to specify a new name.

20 To add an aggregation, click **Add Column**.



21 When the aggregations are complete, click **Next**.

22 In the **Filter Data** task, all source rows are included in the target by default. To accept this default, click **Next**.

23 If your job includes no summarizations, then you can select **No duplicate rows** to remove duplicate rows from the target.

Note: If Hive is enabled for this directive, note that older versions of Hive do not support the selection of both **No duplicate rows** and **All Rows**.

24 To filter rows from the target, choose one of the following:

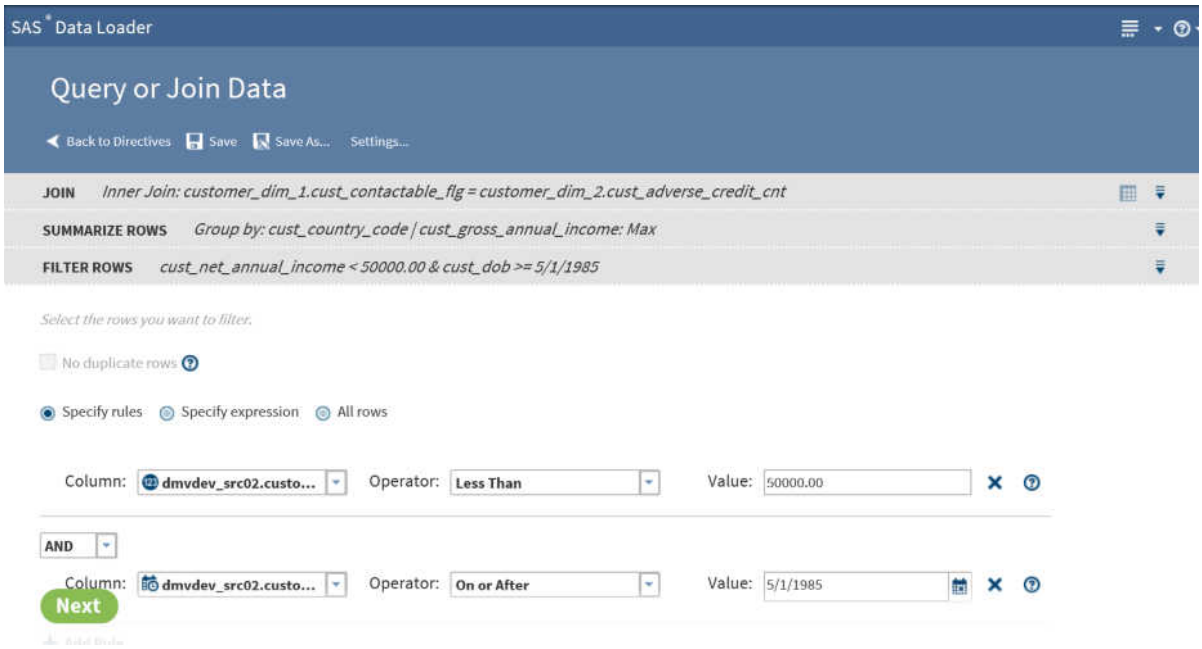
a To filter rows using one or more rules, click **Specify rules** and proceed to the next step. You can specify multiple rules and apply them using logical AND and OR operators.

b To filter rows using an expression, click **Specify expression** and go to Step 26 on page 87.

25 To filter rows by specifying one or more rules, follow these steps:

a Click **Select a column** and choose the source column that forms the basis of your rule.

b Click and select a logical **Operator**. The operators that are available depend on the type of the data in the source column.



c In the **Value** field, add the source column value that completes the expression. In the preceding example, the two rules combine to read "Filter from the target all source rows with an income less than 50,000.00 and born on or after May 1, 1985."

d Click **Add Rule** as needed to add another rule. Select a different column, operator, and value. To associate a new rule with the previous rules, either retain the default **AND** operator or click **AND** and select **OR**.

e When your rules are complete, go to Step 27 on page 88.

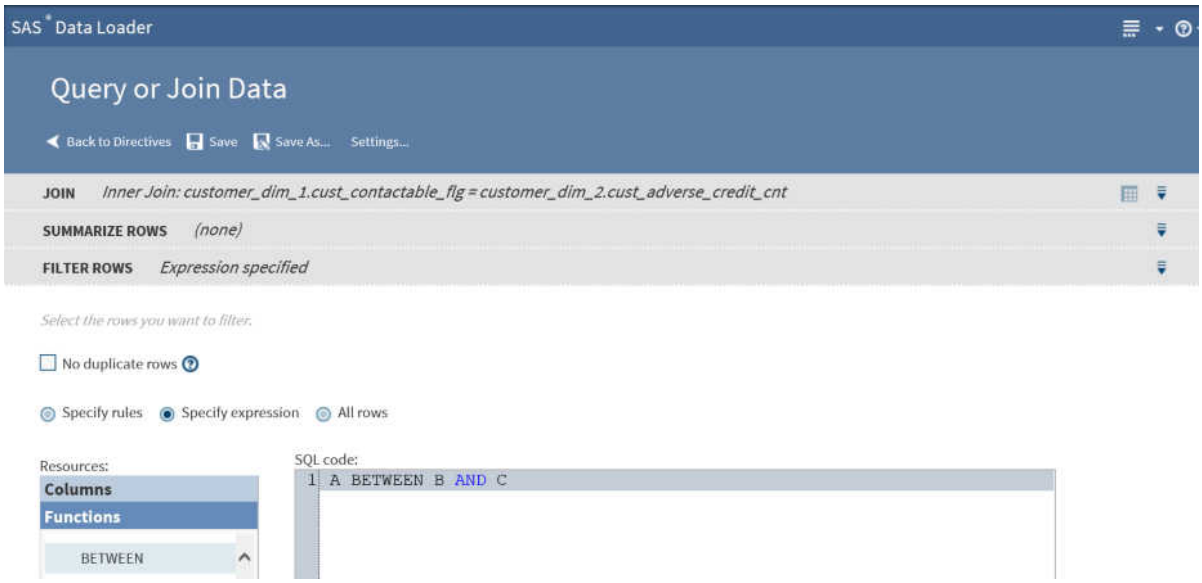
26 To filter rows using an expression, follow these steps:

a In the expression text box, either enter or paste an expression.

Your expression must use either HiveQL functions or Cloudera Impala SQL functions, depending on the selected SQL environment. Click **Settings** to display the selected SQL environment. To learn more about SQL environments, see "Enable Support for Impala and Spark".

To learn about the requirements for expressions, see "Develop Expressions for Directives".

b To add functions to your expression, click **Functions** in the **Resources** box, expand a category, select a function, and click .



c To add column names to your expression, position the cursor in the expression text box, click **Columns** in the **Resources** box, click a source column, and then click .

27 When your rules or expression are complete, click **Next** to open the **Columns** task.

28 Use the Columns task to do the following:

Note: The **Columns** task is available only if your job *does not* contain summaries. If your job *does* contain summaries, then click **Next** to display the **Sort** task.

- Add all source columns to the target , add one source column to the target , remove one source column from the target (or), and remove all source columns from the target .
- Rename columns by clicking in the **Target Name** column.
- Move column to first or full-left position , move column left one position , move column right one position , and move column to last or full-right position .
- Add a new column that will be populated with the results of a user-written expression. Click the **Add** icon .

  The expression must use either HiveQL or Cloudera Impala SQL. To see the selected SQL environment, click **Settings**. For more information about SQL environments, see “Enable Support for Impala and Spark”,

  To learn the requirements for expressions, see “Develop Expressions for Directives”.
- Add a new column and open the Advanced Editor to develop an expression. Click .
- Modify or replace the data in an existing column using data that is returned by a user-written expression. Click .

29 Follow these steps to specify a user-written expression:

a Either add a new column or verify the name of the column to be modified In the **Column name** field.

b Paste or enter SQL code into the expression column, or open the Advanced Editor to create an expression.

c In the Advanced Editor, select functions and column names from the **Resources** box to create your expression.

30 Click **Next** to close the **Column name** task and open the **Target Table** task.

31 In the **Target Table** task, to learn about the contents of a table, click the table and click the **Table Viewer** icon .

32 To write your target data to an existing table, click that table and click **Next**. Any and all existing data will be replaced.

33 To save data to a new target table, click New Table..., enter a table name in the New Table window, and click **OK**.

The names of tables must meet the naming conventions of SAS and Hive.

34 To display your target data as a view, select Save as a View. Saving as a view displays your target data in Sample Data Viewer without saving the results to a table on disk.

When your target selection is complete, click **Next** to open the **Code** task

35 In the **Code** task, click **Edit Code** to edit the generated code. Click **Reset Code** to restore the original generated code. Click **Next** to open the **Result** task.

Note: Edit your SQL code with care. The code in the editor is the exact code that will be executed by your job, regardless of previous selections. Also, code changes are not reflected in prior tasks, so code regeneration does not retain code edits.

36 In the **Result** task, you can review the previous tasks by clicking on the gray bars at the top of the window.

37 Click **Save** or **Save As** to save your job.

38 Click **Start** to execute your directive. To monitor the progress of your job, see the "Run Status" directive.

Sort and De-Duplicate Data

Introduction

Sort and De-Duplicate Data
Query, sort, or de-duplicate the data in an existing Hadoop table

Use the Sort and De-Duplicate Data directive to create jobs that include some or all of the following steps:

1. Group rows based on selected columns and then summarize numeric columns for each group and subgroup.
2. If not summarizing, specify the removal of duplicate rows and filter rows from the target.
3. Remove, reposition, and rename the columns in the target table. Add columns that receive the results of SQL expressions.
4. Sort target rows by selecting one or more columns for ascending or descending values.

Enable the Impala SQL Environment

Support for the Cloudera Impala SQL environment is enabled in the **Hadoop Configuration** panel of the **Configuration** window. When Impala is enabled, new instances of the following directives use the Cloudera Impala SQL environment by default:

- Sort and De-Duplicate
- Query or Join
- Run a Hadoop SQL Program

The default SQL environment can be overridden using the **Settings** menu. To learn more about SQL environments, see "Enable Support for Impala and Spark".

Note: Changing the default SQL environment does not change the SQL environment for saved directives. Saved directives continue to run with their existing SQL environment unless they are opened, reconfigured, and saved.

Example

Follow these steps to use the Sort and De-Duplicate directive:

1. On the SAS Data Loader directives page, click **Sort and De-Duplicate Data**. The **Source Table** task is displayed. For more information about data sources and tables, see "Viewing Data Sources and Tables" on page 20.

2 In the **Source Table** task, click a table in your default data source and click **Next**. To select a source table from another data source, click
Return to Data Sources .

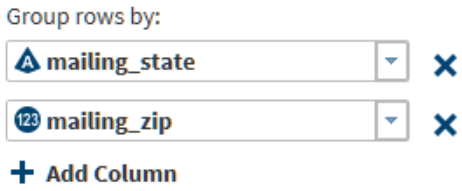
3 Use the **Summarize Rows** task to group rows in the target according to column values, and then summarize numeric values for each group or subgroup.

If you do not want to generate summary values for groups of rows, or if you want to remove duplicate rows, click **Next** to display the **Filter** task.

Note: If your source data is in Hive 13 (0.13.0) or lower, the **Summarize Rows** task will not handle special characters in column names. To resolve the issue, either rename the columns or ask your Hadoop administrator to upgrade to Hive 14 (0.14.0 or higher).

Follow these steps to use the **Summarize Rows** task:

a Click **Group rows by** and select a column. To generate nested groups with additional summary values, click **Add Column**.



b Click **Summarize column** and select a column that will be used to generate summary values for each group. The summarized values will appear in a new column in the target.

c Click **Aggregation** and select the summary type. The available summary types are defined as follows:

Count
specifies the number of rows that contain values in each group.

Count Distinct
specifies the number of rows that contain distinct (or unique) values in each group.

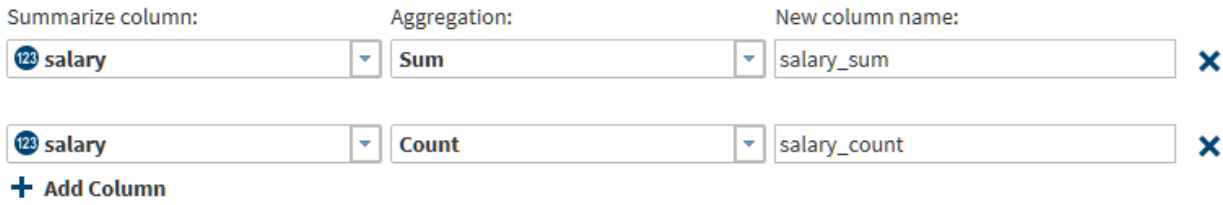
Max
specifies the largest value in each group.

Min
specifies the smallest value in each group.

Sum
specifies the total of the values in each group.

d Click **New column name** to change the default column name for new target column that will receive summarized data. The new target column will contain a summary value for each group and subgroup.

e Click **Add Column** to specify a second summary and target column.



f When your groups and summaries are complete, click **Next** to display the **Filter** task.

4 The **Filter** task enables you to remove duplicate rows and filter (remove) rows from the target.

If you specify summaries or if you do not need to filter rows from the target, then click **Next** to display the **Columns** task.

Follow these steps to use the **Filter** task:

a To remove from the target any rows that are identical to another row, click **No duplicate rows**.

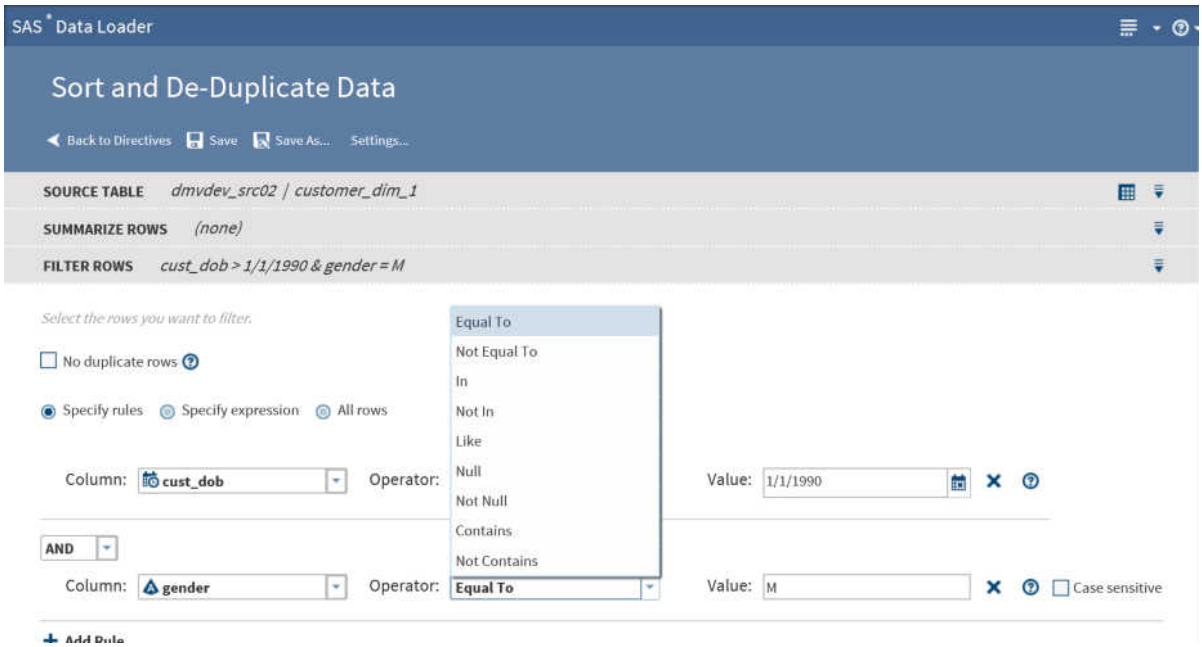
b To filter rows from the target using one or more rules, click **Specify rules**.

To filter rows using an SQL expression, click **Specify expression**.

c To filter rows by specifying one or more rules, follow these steps:

i Click **Column** and choose the source column that forms the basis of your rule.

ii Click and select a logical **Operator**. The operators that are available depend on the type of the data in the source column. For example, the following image shows the operators that are available for the character data type:



iii In the **Value** field, add the source column value that completes the expression. In the preceding example, the two rules combine to read

"Filter from the target all male contacts who were born on or after January 1, 1990."

iv Click **Add Rule** as needed to add another rule. Select a different column, operator, and value. To associate a new rule with the previous rules, either retain the default **AND** operator or click **AND** and select **OR**.

v When your rules are complete, click **Next** to display the **Columns** task and go to Step 5.

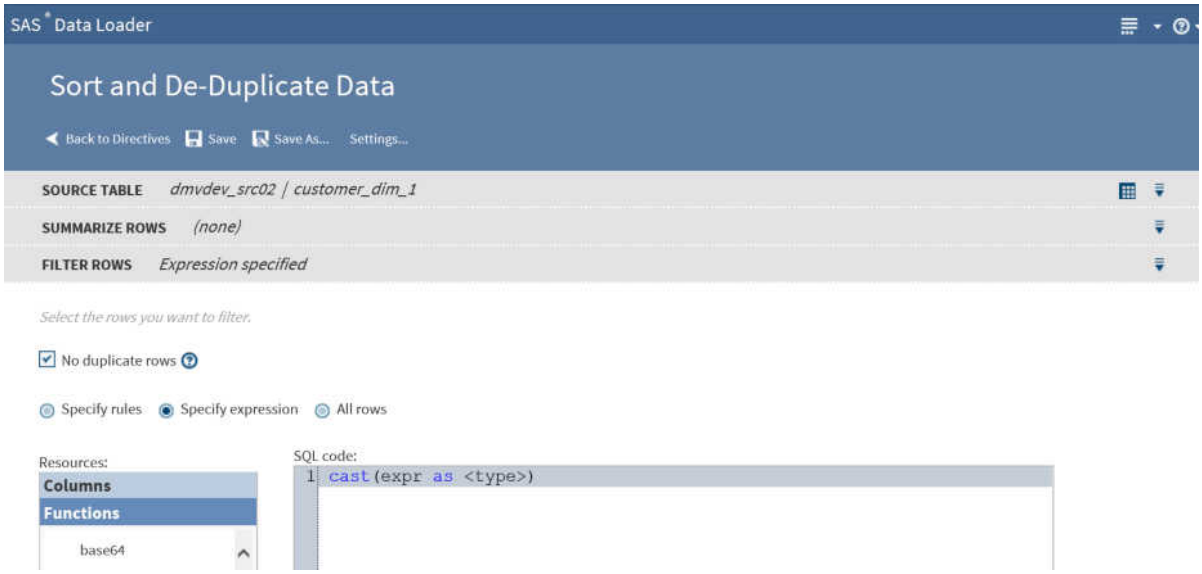
d To filter rows using a user-written expression, follow these steps:

i In the **SQL Code** text box, either type or paste an expression using HiveQL or Impala SQL.

Your expression can use either HiveQL functions or Cloudera Impala SQL. Click **Settings** to display the selected SQL environment.

To learn about the requirements for expressions, see "Back Up Directives".

ii To add functions to your expression, click **Functions** in the **Resources** box, expand a category, select a function, and click .



To add column names to your expression, position the cursor in the **SQL code** text box, click **Columns** in the **Resources** box, click a source column, and then click .

iii When your expression is complete, click **Next** to open the **Columns** task.

5 Use the **Columns** task when you have not defined summaries to remove, reorder, rename, modify target column data, or add a column of newly calculated data. New data is generated by user-written expressions.

If you defined summaries in the **Summarize Rows** task, or if you do not need to modify columns, then click **Next** to display the **Sort** task and go to Step 6.

Follow these steps to use the **Columns** task:

a Click **Specify Columns** to display the **Selected Columns** and **Available Columns**.

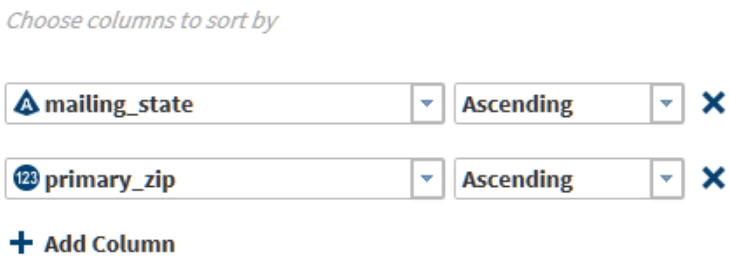
b In **Selected Columns**, click icons to perform the following tasks:

- To rename columns, click , or click and type in the **Target Name** column.
- To rearrange or reorder columns, click the up and down arrow icons. Note that the top row in **Selected Columns** is the first column position (full-left.)
- To remove columns from the target, click the trash can icon or the left arrow icons.
- To add a new target column that will be populated with data returned from a user-written expression, click .
- To add a new column and develop an expression to add data to that column, click . To learn how to use the Advanced Editor, see "Using the Advanced Editor for Expressions".
- To modify column data using a new expression, click .
- To modify or add column data, click the **Expression** column, and either paste an existing expression or enter an expression.

c When your columns are complete, click **Next** to display the **Sort** task.

6 If you have not defined any summaries in the **Summarize Rows** task, then the **Sort** task enables you to group rows based on ascending or descending values.

If you defined summaries, click **Next** to display the **Target** task and go to the next step.



7 In the **Target Table** task, select a location for the target table. When the table list appears, either select an existing target or click **New Table**. To generate a temporary table that is not saved to disk, select **Save as a View**. Click **Next**.

8 In the **Code** task, review and edit the generated code for the directive. Note that you will lose any edits you make if you change a task and regenerate code. Click **Next**.

9 In the **Result** task, click **Save** or **Save As** to save your job. You can then access that job in Saved Directives. Click **Start querying data** to run your job.

Using the Advanced Editor for Expressions

In the directive Sort and De-Duplicate Data, in the **Columns** task, you can use the Advanced Editor to add or edit user-written expressions. Each expression returns one value per row to either modify existing data or to add data to a new column.

The Advanced Editor enables you to insert column names and SQL function syntax and column names into your expressions. Syntax descriptions are provided for the supported SQL functions.

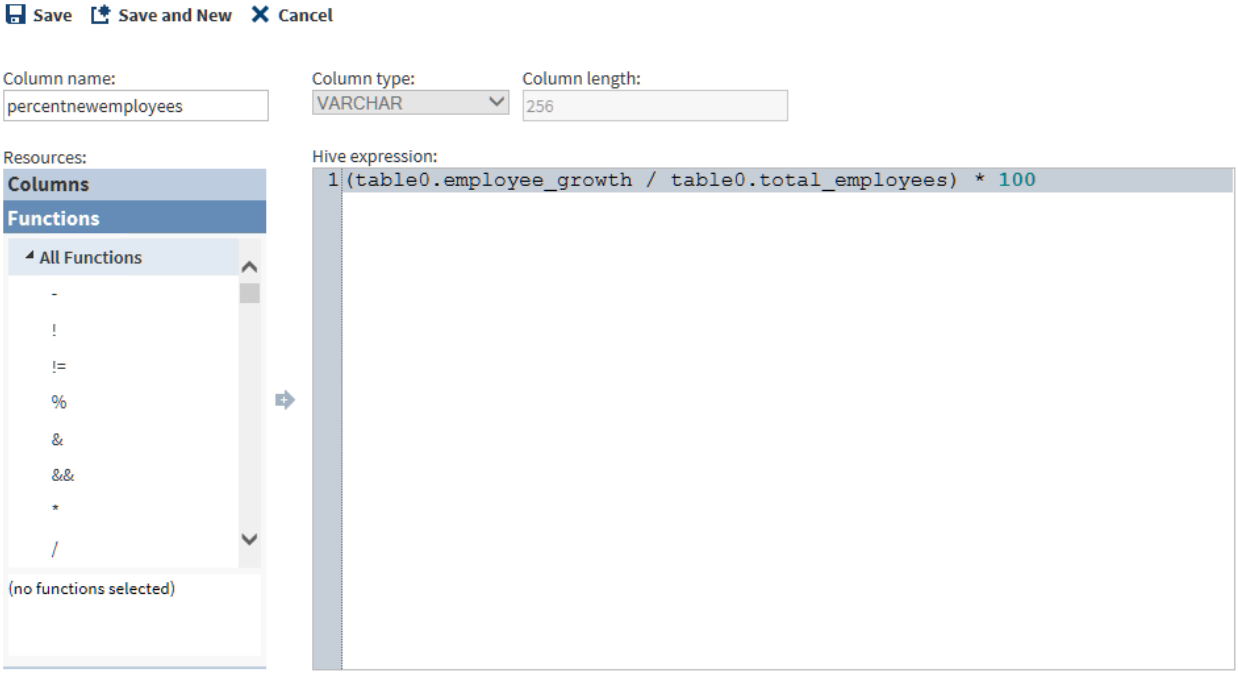
Follow these steps to use the Advanced Editor:

1 As needed in the **Columns** task, click or to open the Advanced Editor.

2 In the **Advanced Editor**, in the **Column Name** field, enter a name for a new column or rename an existing column. The fields **Column type** and **Column length** describe the selected column.

3 To build an expression, you can start by pasting expression code from your clipboard, or you can select and add function syntax from the **Resources** box.

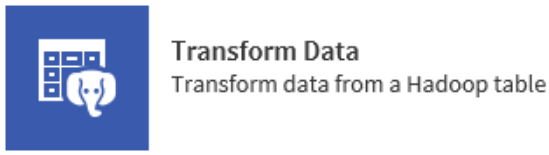
To learn about the requirements for expressions, see “Develop Expressions for Directives”.



4 To save your expression and return to the **Columns** task, click **Save**. To save and create another new column and expression, click **Save and New**. In the **Columns** task, new columns are displayed at the bottom of the **Selected Columns** box.

Transform Data

Introduction



Use the Transform Data directive to filter data, manage columns, and summarize data in one or more Hadoop source tables.

Enable the Spark Runtime Target

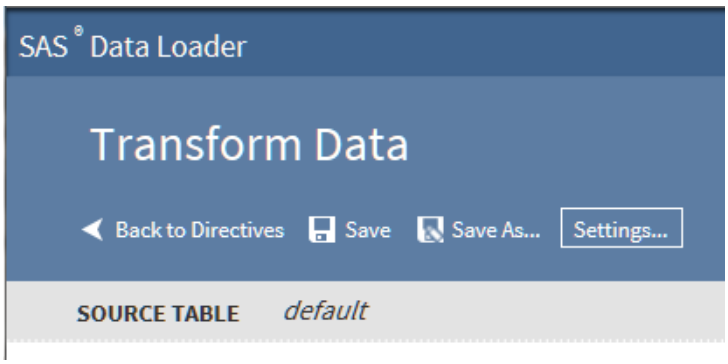
Support for the Apache Spark runtime target is enabled in the **Hadoop Configuration** panel of the **Configuration** window. When Spark is enabled, new instances of the following directives use Spark by default:

- Transform Data
- Cleanse Data
- Cluster-Survive Data

The default runtime target can be overridden using the **Settings** menu. To learn more about runtime targets, see “Enable Support for Impala and Spark”.

Note: Changing the default runtime target does not change the runtime target for saved directives. Saved directives continue to run with their existing runtime target unless they are opened, reconfigured, and saved.

Note: Enabling Spark changes the truncation of character columns, as described in the “Usage Notes for Spark”.



Example

The following example depicts the process of creating and running a directive that contains several transformations. The example opens a source table of customer information, selects columns for the target, and applies two filters.

1 On the SAS Data Loader directives page, click **Transform Data**. The **Source Table** task is displayed. For more information about data sources and tables, see “Viewing Data Sources and Tables” on page 20.

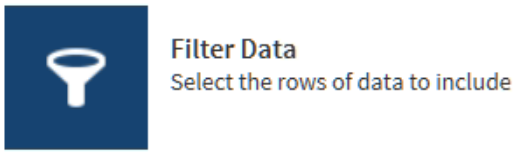
2 In the **Source Table** task, click the schema that contains the source table that you will transform. When the tables appear, select the source table and click **Next**.

3 In the **Transformation** task, click a transformation:

- Click **Filter Data** to exclude unwanted rows from the target table using rules or a user-written expression.
- Click **Manage Columns** to reorder, rename, or remove columns from the target table. You can also apply user-written expressions to modify column data or add new data to new columns. The Advanced Editor is provided to display available functions, provide syntax help, and load function syntax into your expression.
- Click **Summarize Rows** to group rows based on the values in one or more columns. For each group, you can generate summary aggregations from selected numeric columns.

Your job can consist of one or more transformations. Multiple transformations are executed in the order in which you define them. A logical order for all three transformations is filter data, manage columns, and summarize rows.

4 Click **Filter Data**.



5 In the **Filter Data** transformation, click **Specify rules** or **Specify expression**.

6 To specify rules, follow these steps.

a Select the first column, operator, and value. Note that the available operators change based on the type of the column. To learn about the available operators, see “About the Operators in the Filter Data Transformation” on page 100.

b Choose a logical operator (AND or OR)

c Select the second column, logical operator, and value.

d To add another rule, click **Add Rule**.

7 To filter rows with a user-written expression, follow these steps:

a To paste in an existing expression, click the expression text box and press Ctrl+V or your equivalent.

Your expression can use either SAS DS2 functions (with the MapReduce runtime target,) or DataFlux Expression Engine Language functions (with the Spark runtime target.) Click **Settings** to display the selected runtime target. To learn more about runtime targets, see “Enable Support for Impala and Spark”.

To learn about the requirements for expressions, see "Develop Expressions for Directives".

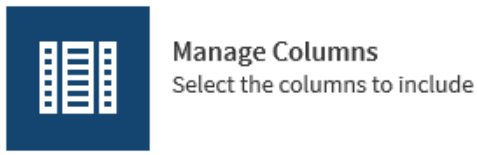
b To write an expression, expand the **Functions** and categories in the **Resources** box.

c Select a function for your expression and click . Refer to the syntax help at the bottom of the **Resources** box as needed.

d In the expression text box, replace the placeholders in the default function syntax with column names and values. To insert column names, expand **Columns** in the **Resources** box.

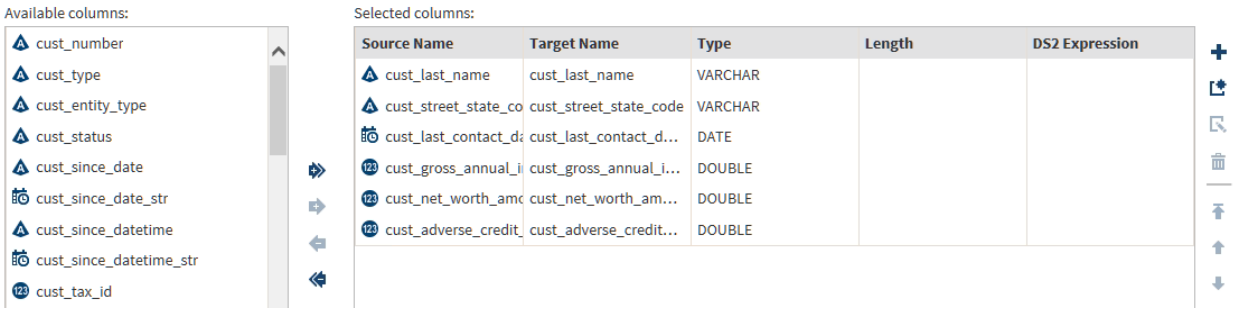
8 After you define rules or an expression, click **Next** to end the directive, select a target table, and run the directive. To add a **Manage Columns** or **Summarize Rows** transformation, click **Add Another Transformation**.

9 In the **Transformation** task, click **Manage Columns**.



10 In the **Manage Columns** transformation, you can reorder, rename, change type, change length, and remove source columns from the target. You can also apply user-written expressions to modify column data or to generate new data for new columns.

11 To reorder columns, use the vertical arrow icons. The first row in **Selected columns** is the target column in the first position (fully left.)



12 To change the target column name, type, or length, click **Target Name**, **Type**, or **Length**.

13 To remove a source column from the target, click the row in **Selected columns** and then click .

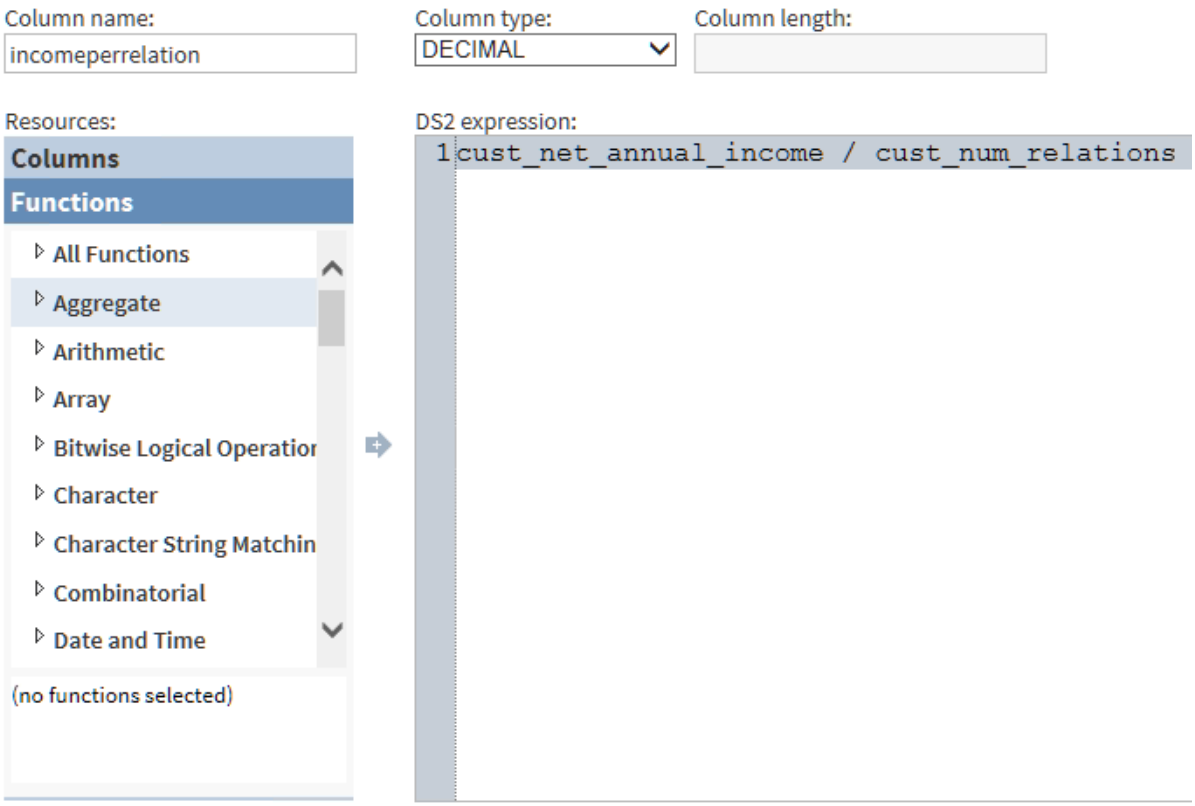
14 To replace existing column data with data that is generated by a user-written expression, click a **Selected column** and click **Expression**. At this point, you can enter or paste an existing expression in the corresponding text field.

Your expression can use either SAS DS2 functions (with the MapReduce runtime target,) or DataFlux EEL functions (with the Spark runtime target.) Click **Settings** to display the selected runtime target. To learn more about runtime targets, see "Enable Support for Impala and Spark".

To learn about the requirements for expressions, see “Develop Expressions for Directives”.

15 To add a new target column, and to use the Advanced Editor to write an expression for that column, click .

16 To use the Advanced Editor, select functions and column names from the **Resources** box. When your expression is complete, select **Save** or **Save New** to return to the Manage Columns transformation. The new column appears at the bottom of **Selected Columns**.



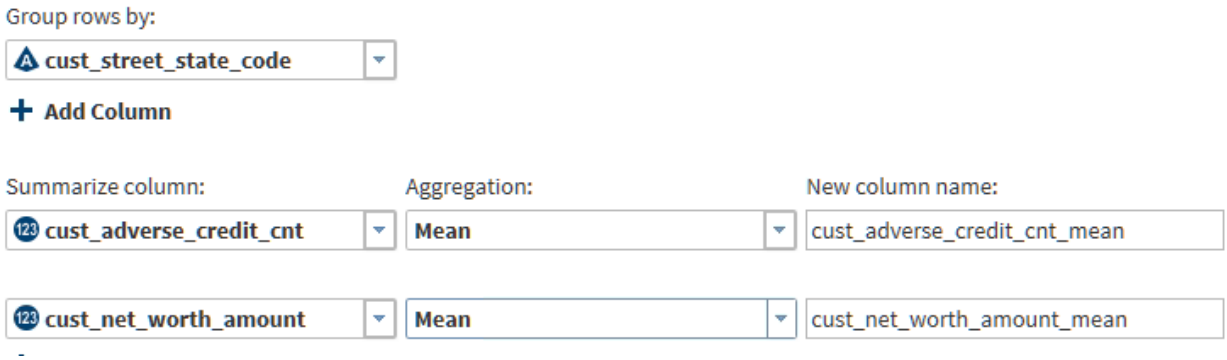
17 Click **Add a new transformation**, and then, in the **Transformation** task, click **Summarize**.

18 In the **Summarize Rows** task, click **Group rows by** to specify a column whose values will be used to group rows. You can specify additional columns that will form subgroups. Each group and subgroup will receive a value in each aggregation column.

Note: If your source data is in Hive 13 (0.13.0 or lower), then the **Summarize Rows** task will not handle special characters in column names. To resolve the issue, either rename the columns or ask your Hadoop administrator to upgrade to Hive 14 (0.14.0 or higher.)

19 Click **Select a column** to specify a summarization, and then click and select an aggregation. To learn about the available aggregations, see “About the Aggregations in the Summarize Rows Transformation” on page 104.

20 Click **New column name** and enter or paste replacement names for the aggregation columns.



21 When your summaries are complete, click **Next** to conclude your job.

22 In the **Target Table** task, select the schema that contains or will contain your target table.

23 Click **New Table...** to create a new table, or click an existing table that will be overwritten by your job.

> **TIP** If you select a table and the **View Profile** icon is enabled, you can click that icon to display a profile report for that table.

24 Click **Next** to display the **Result** task. In the **Result** task, click **Save** or **Save As** to save your directive. If you want to run your job now, click **Start transforming data**. Otherwise, you can run your job later from "Saved Directives".

About the Operators in the Filter Data Transformation

The following table describes filter operators by the data type of the selected column.

Table 4.9 Logical Operators in the Filter Transformation

Operator	Source ColumnData Types	Description and Example

Equal To	The Equal To operator is available for use with all source data types, which include the following:
	Character
	Numeric
	Datetime
		The source value is accepted and its row is written to the target table only when the source value exactly matches the comparator.
		Character values can be case-sensitive. Blank spaces are included in the comparison.
		Datetime values in the comparator use the SAS format DATETIME(w.p).
		`Gender Equal To Male`
		`PrefCustomer Equal To 1`
		`SaleDate Equal To 5/1/2014`

Operator	Source Column Data Types			Description and Example

Not Equal To	A	123		Accepts the source row when the column value is anything other than the comparator.
				`Region Not Equal To Europe`
				`NumChildren Not Equal To 0`
				`SaleDate Not Equal To 11/25/2013`

Null	A	123		When the runtime target is MapReduce, the Null operator accepts the source row when the column value is NULL or if no source value is present. When the runtime target is Spark, the Null operator accepts only column values of NULL.
				`CreditScore Null`
				`AnnualIncome Null`

Not Null	A	123		Accepts the source row when the column value is present and when the value is not NULL.
				`PostalCode Not Null`
				`PhoneNumber Not Null`

In	A	123		Accepts the source row when the column value is included in its entirety within the comparator. The comparator consists of a list of constant values. The list consists of a vertical list of individual entries, without commas. Blank spaces are interpreted literally. Case sensitivity can be enabled.
				`CarManuf In`
				`BMW`
				`VW`
				`Benz`
				`WaistSize In`
				`32`
				`34`
				`36`
				`38`

Operator	Source Column Data Types	Description and Example

Not In	A 123	Accepts the source row when the column value is not included anywhere within the comparator's list of constant values.
		`City Not In New York`
		`Chicago`
		`Los Angeles`
		`WaistSize Not In 32`
		`34`
		`36`
		`38`

Like	A	Accepts the source row when the column value matches the result of an expression in the comparator. The source value and the comparator are compared on a character-by-character basis. Case-sensitivity can be enabled.
		Use the pattern-matching character `%` to indicate any string of characters. Use the underscore character _ to indicate any single character in that position.
		Note that trailing blank characters are written to the target table when using `%` at the end of the comparator.
		Use the word `escape` to include literal instances of `%` and _ in the comparator.
		`SalesRegion Like NorthAmer%`
		`AnnualSales Like 199_`
		`CustSatisfaction Like 100 escape %`

Not Like	A	Accepts the source row when the column value does not match the result of an expression in the comparator. The source value and the comparator are compared on a character-by-character basis. Case-sensitivity can be enabled. Pattern-matching characters `%` and _ and `escape` are valid as described for the Like operator.
		`Sports Not Like %ball`
		`FootballFieldLength Not Like 100%`

Operator	Source ColumnData Types	Description and Example

Contains	A	Accepts the source row when the column value is found within the character string of the comparator. Case-sensitivity can be enabled.
		`Address Contains IL`
		`LicenseNumber Contains 7227`

Not Contains	A	Accepts the source row when the column value is not found within the character string of the comparator, or is null. Case-sensitivity can be enabled.
		`Month Not Contains OctNovDec`
		`SalesMonthly Not Contains 0`

Between	123 	Accepts the source row when the column value or date is between the two values or dates in the comparator, but is not equal to either.
		`GradeAverage Between 87.5 93`
		`DailySales Between December 20, 2014 December 27, 2014`

Greater Than	123	Accepts the source row when the column value is greater than the value of the comparator. `AnnualSales GreaterThan 100000`
Greater Than Or Equal To	123	Accepts the source row when the column value is equal to the comparator or greater than the comparator. `CarsInFamily Greater Than or Equal To 3`
Less Than	123	Accepts the source row when the column value is less than the value of the comparator. `GamerAge Less Than 30`
Less Than Or Equal To	123	Accepts the source row when the column value is equal to the value of the comparator, or less than the value of the comparator. `SalesYear Less Than Or Equal To 2010`

Operator	Source ColumnData Types	Description and Example
After		Accepts the source row when the column date is later than the date in the comparator. `HomePurchaseDate After January 1, 2013`
Before		Accepts the source row when the column date is earlier than the date in the comparator. `BirthDate Before March 17, 1980`
On Or After		Accepts the source row when the column date is later than, or the same date as, the date in the comparator. `DailySales On Or After January 1, 2014`
On Or Before		Accepts the source row when the column date is earlier than, or the same date as, the date in the comparator. `DailySales On Or Before December 31, 2013`

About the Aggregations in the Summarize Rows Transformation

The aggregations that are available in the Summarize Rows transformation are defined as follows:

Count
: the number of rows in the group that contain valid values.

Count Distinct
: the number of unique values in the column for each group.

Corrected Sum of Squares
: measures variability or dispersion around the mean. To learn more about this (and other) statistical summaries, see the *Introduction to Statistical Modeling with SAS/STAT Software*.

Covariance
: measures the strength of the correlation of the values in the group. A positive value indicates that values move in the same direction within the group. A negative value indicates that values move in opposite or random directions.

Max
: the maximum value in the column for each group.

Mean
: the calculated center value between the maximum and minimum values in the group.

Min
the minimum value in the group.

Number of Missing Values
the number of rows in the group that contain a blank or NULL value.

Range
the difference between the lowest and highest values in the group.

Standard Deviation
measures the degree of variance, or the degree in which the values in the group deviate from the mean. A small value indicates little deviation. The standard deviation is the square root of the Variance.

Standard Error
measures the applicability or accuracy of the mean as it applies to the values in the group. A small value indicates that the mean is a more accurate reflection of the values in the group.

Sum
adds the values in the group

Variance
the average of the squared differences from the mean, which measure diversity in the group

Transpose Data

Introduction

Transpose Data
Transpose data from a Hadoop table

Use the Transpose Data directive to transpose one or more columns in a source table into rows in a target table. The columns in the target are the values of a specified column in the source. For example, you could specify that the columns of the target be taken from the values of a source table column that contains customer ID numbers. Each unique customer ID value in the source becomes a separate column in the target.

You do not have to transpose all of the columns in the source. You can select source columns that will be copied directly to the target.

This directive contains embedded help that includes examples of transposed data.

CAUTION! Selecting columns with a high degree of cardinality (number of unique values) can decrease performance in Transpose jobs. To maximize performance, profile your source columns and filter your source rows. You can filter source rows in the directives Cleanse Data or Query or Join Tables.

Example

Follow these steps to use the Transpose Data directive.

1 On the SAS Data Loader directives page, click **Transpose Data**. The **Source Table** task is displayed. For more information about data sources and tables, see "Viewing Data Sources and Tables" on page 20.

2 In the **Source Data** task, click the data source that contains your source table, click the source table, and then click the Table Viewer .

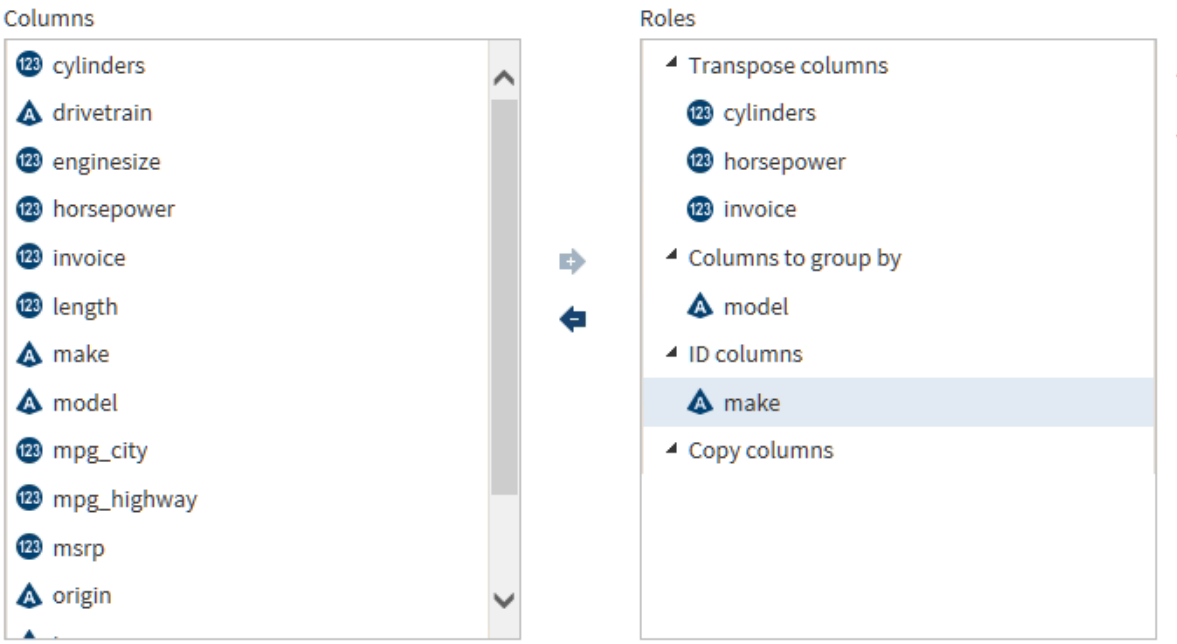
Examine the source table to determine the roles for the columns.

Note: Valid source table selections must have names that contain no more than 32 characters. Longer table names cause transpose jobs to fail. For information about other requirements, see "Usage Notes" on page 107.

3 In the **Transpose Data** task, click the required **Transpose data**, click the columns that you want to see as rows, and click the right arrow. If you transpose multiple columns, then you can arrange them in **Roles** using the up and down arrows.

4 Click the required **Columns to group by**, click an available column, and then click the right arrow. The group-by column becomes the leftmost column. Each row in that column receives a set of values from the transposed columns.

5 As needed, click **ID column**, click an available column, and then click the right arrow. The values of the ID column become column names in the target.



6 To copy a column from the source to the target, select **Copy column**, select an available column, and click the right arrow. The copied column will be positioned as the last, or rightmost, column.

Usage Notes

Changing the Maximum Length of Character Columns

If necessary, you can change the maximum length of character columns for source tables to this directive. For more information, see “Change the Maximum Length for SAS Character Columns” on page 195.

Avoid Using DS2 Reserved Keywords as Column Names

Do not use a DS2 reserved keyword for the name of a column that is the target of the Transpose directive. For example, assume that a source table contains a column named OTHER. If the column that is named OTHER is specified as a column to transpose, a runtime error is generated because OTHER is a DS2 reserved keyword.

For more information about DS2 keywords, see *SAS 9.4 DS2 Language Reference*.

5

Profile Data

Overview of Profile Directives

Data profiling jobs help you assess the composition, organization, and quality of Hadoop tables. They help you recognize patterns, identify scarcity in the data, and calculate frequency and basic statistics. Data profiling can also aid in identifying redundant data across tables and cross-column dependencies. All of these tasks are critical to optimal planning and monitoring.

The profile directives enable you to generate and view reports for one or more Hadoop tables. The reports display sample data, column information, and measurements of data quality. You create profile reports with the Profile Data directive and use the Saved Profile Reports directive to access and manage profile reports.

Here's an example of a profile report:

SAS 8 Data Loader - Profile Reports

contacts_report

Go to Profile Report List | Show Outline | Show Trends | Show Notes | Add Note... | Report Version: May 13, 2015, 2:09:00 PM

sgfdemo.contacts_cleanse

Count: 48

Data Quality Metrics

Column	#	Unique (n)	Unique (%)	Pattern (n)	Pattern (%)	Null (n)	Null (%)	Blank (n)	Blank (%)
contact_full_...	1	48	100	40	83	0	0	0	0
contact_first_...	2	38	79	9	19	0	0	0	0
contact_midd...	3	18	38	3	6	0	0	10	21
contact_last_...	4	48	100	10	21	0	0	0	0
contact_dob	5	45	100	*	*	3	6	*	*
company_name	6	32	67	8	17	0	0	16	33
primary_addr...	7	48	100	47	98	0	0	0	0
primary_city	8	45	94	42	88	0	0	0	0
primary_state...	9	37	77	15	31	0	0	0	0
primary_zip	10	5	100	*	*	43	90	*	*
primary_coun...	11	45	94	5	10	0	0	1	2
phone_number	12	42	95	5	10	4	8	2	4
fax_number	13	0	0	0	0	48	100	0	0

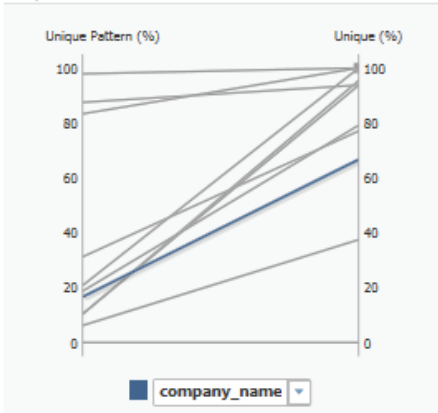
* indicates data not available or not applicable for this column.

Descriptive Measures

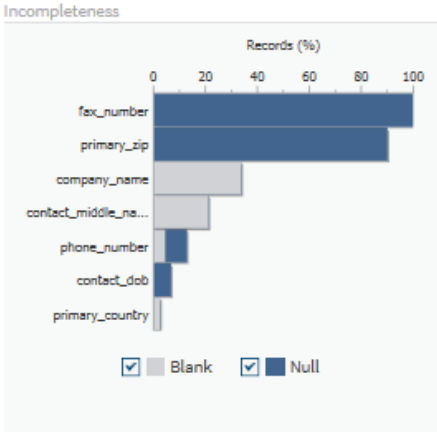
Metadata Measures

Charts

Uniqueness



Incompleteness



Profile Data

Introduction

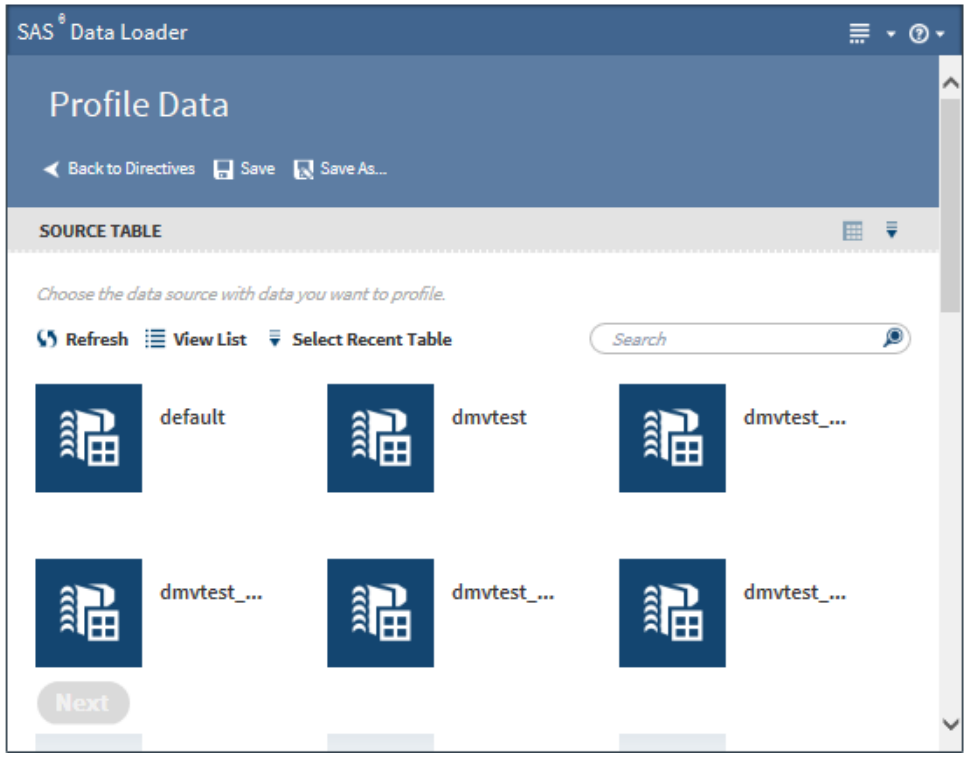
Profile Data
Generate a profile report of the data in a table

Use the Profile Data directive to generate profile reports for one or more tables. You can select a subset of the columns that you want to include in the profile report. The **Profiles** panel of the Configuration window enables you to change the default behavior of new profiles in order to improve performance. For example, you can limit the number of parallel processes that are used in new profile jobs. For more information, see “Profiles Panel” on page 190.

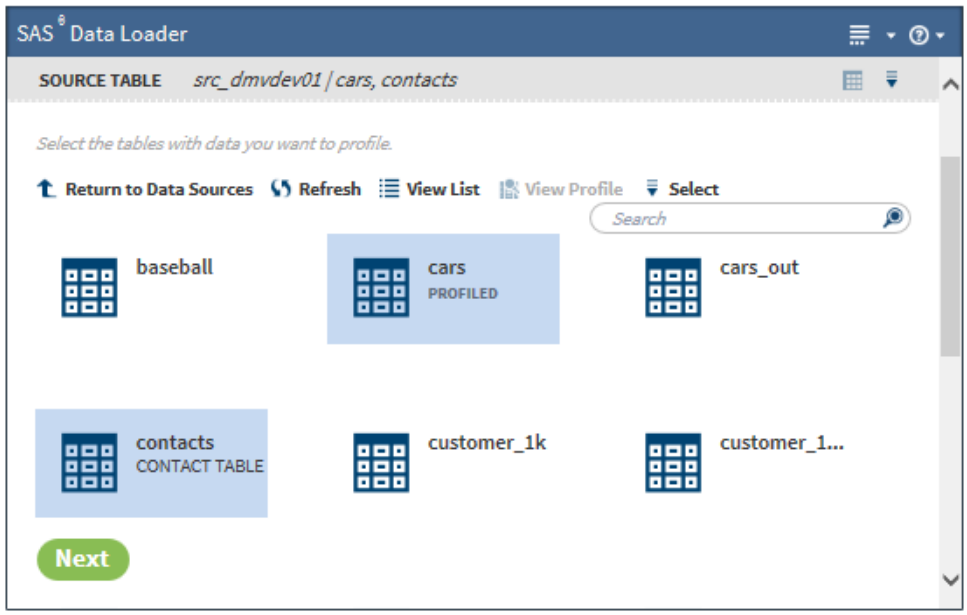
Create a Profile

To create a profile:

1 On the SAS Data Loader directives page, click the Profile Data directive. The **Source Table** task is displayed:



2 Click a data source to display its tables:



3 Select the table or tables for the profile report.

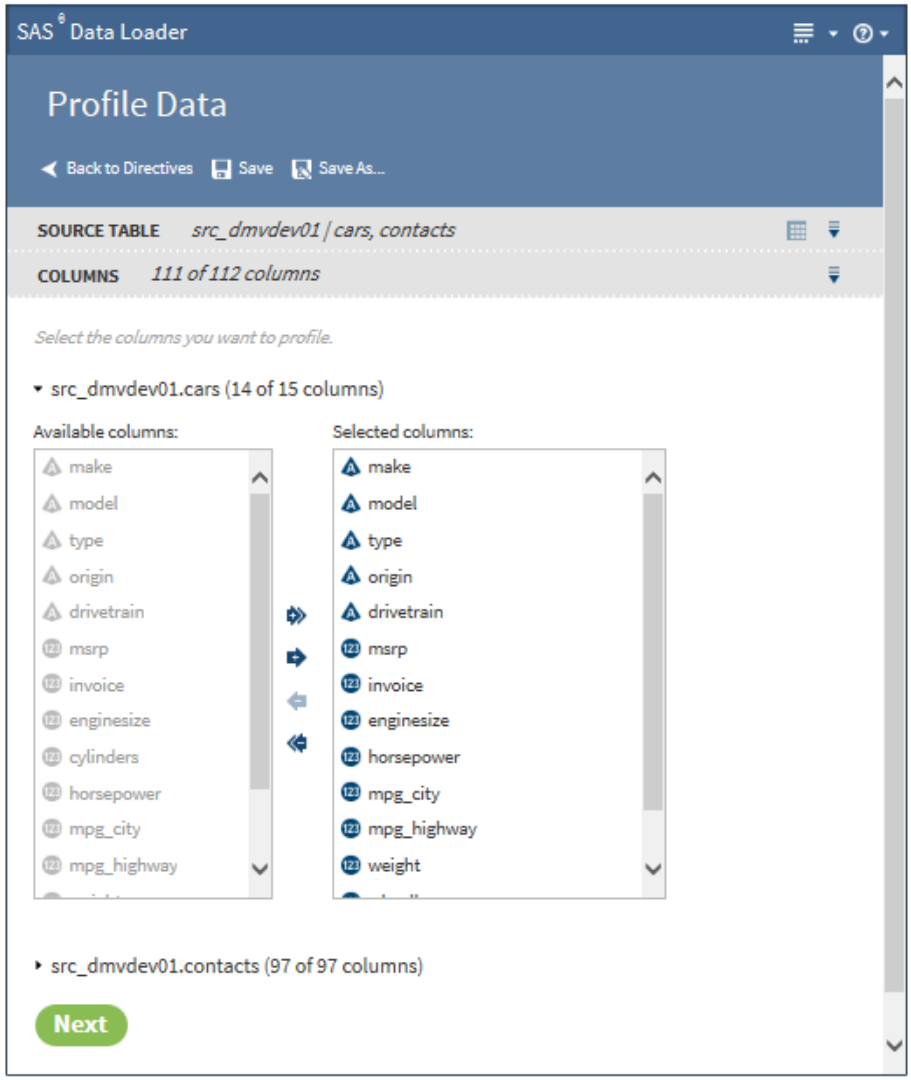
If a profile already exists for a table, PROFILED appears beneath the table name. You can view the existing profile by selecting the table and clicking **View Profile**.

The **Select** menu (Select) provides several options to make selecting tables easier:

- **Select All New Tables**: Automatically selects all new tables in the current data source.
- **Select Recent Table**: Enables you to choose from a list of recently used tables. If you select a table from a different data source, the source table information is adjusted accordingly.
- **Deselect All Tables**: Deselects all tables that you have selected in the current data source.

TIP To view sample data from a table, select the table, and then click in the Source Table header to display the SAS Table Viewer.

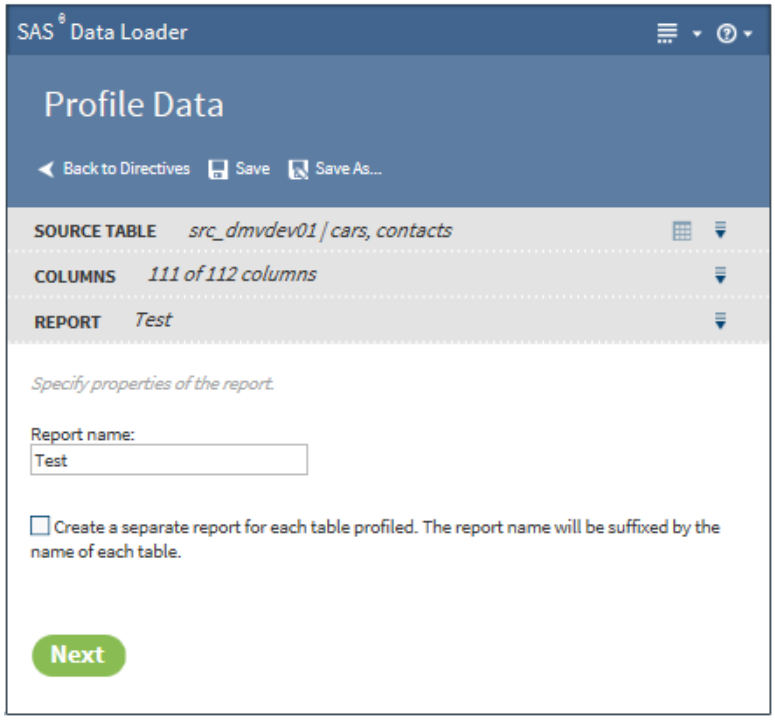
Click **Next**. The **Columns** task is displayed:



4 The **Columns** task displays the total number of columns that are to be processed in the profile report. If you selected more than one table for your report, the tables are listed by name. Click ▸ next to the tables to display the columns that are included in the profile report.

5 The column names in the **Selected columns** pane appear in the report. Select an individual column name and click ◂ or ▸ to move the column name between the **Available columns** pane and the **Selected columns** pane until the correct list of names appears in the **Selected columns** pane. Click ◂◂ or ▸▸ to move all column names at once.

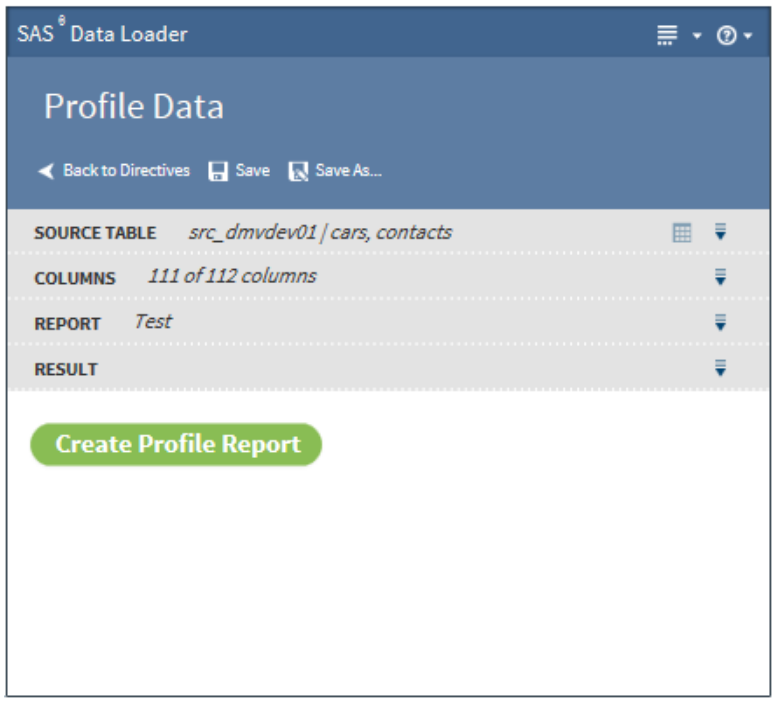
When the column selection is complete, click **Next**. The **Report** task is displayed:



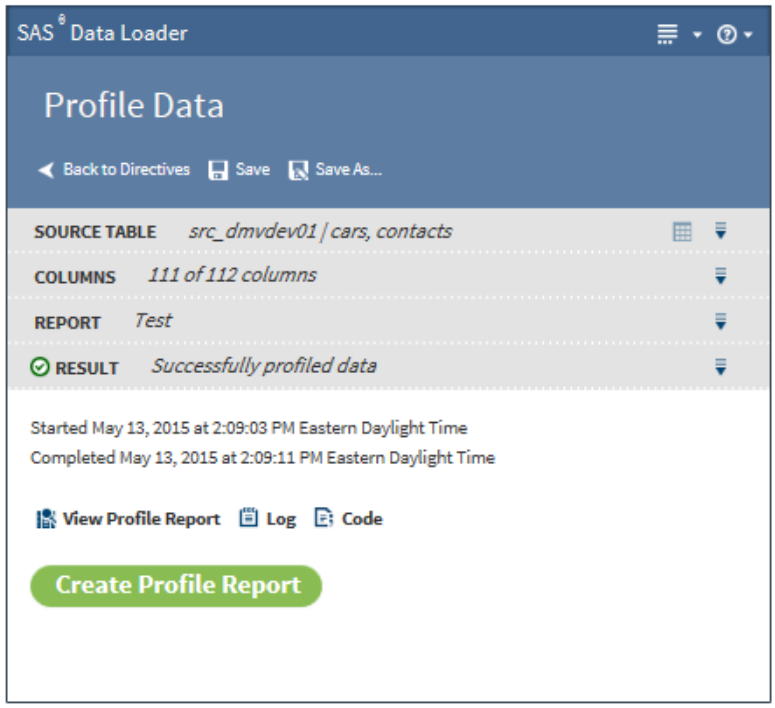
6 In the **Report** task, enter a name for the profile report in the **Report name** field.

If you selected multiple tables and want a separate report for each table, click **Create a separate report for each table profiled**.

Click **Next** to display the **Result** task:



7 Click **Create Profile Report**. After successfully creating any profile reports, a screen similar to the following is displayed:



The following actions are available:

View Profile Report
enables you to view the Profile Report. See "Saved Profile Reports" on page 115 for more information about the profile report.

Log
displays the SAS log that is generated during the creation of the profile.

Code
displays the SAS code that generates the profile.

Usage Notes

Hive tables have a maximum table name length of 132 characters. Many of the SAS Data Loader directives can create tables with names that exceed the SAS table name length limit of 32 characters. The tables that you submit for profiling in the Profile Data directive must conform to the 32-character name length limit. Table names that exceed 32 characters generate error messages.

Saved Profile Reports

Introduction

Saved Profile Reports
Explore previously generated profile reports

Use the Saved Profile Reports directive to view the results of previously executed data profiles and to create notes about the results. The profiles are created with the Profile Data directive. The profile reports and notes are stored as XML documents on the file system. Saved Profile Reports displays these XML files in a readable format.

About Profile Reports

Profile reports can provide valuable information about a Hadoop table and help identify issues that might exist before you use the table for data management or analysis. A profile report includes a summary view with information about the table that was profiled and detail views with information about individual columns in the table.

Summary View

The summary view of a profile report includes the following information:

Count
the total number of rows in the table that was profiled.

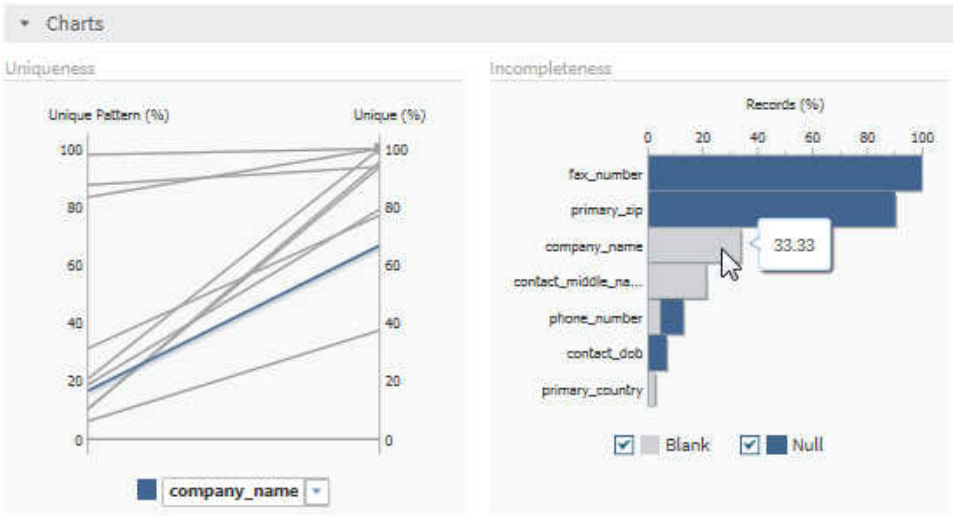
Data Quality Metrics
measurements of data quality for the columns in the table. Measurements include information about the uniqueness of column values, pattern analysis results, and completeness information, including null or blank values.

Note: The measurement of percent null (**Null (%)**) is rounded to the nearest tenth of a percent. Percentages of null values that are smaller than 0.01 are rounded to zero. Refer to the number of null values (**Null (n)**) as needed.

Descriptive Measures
descriptive statistics for columns in the table, including information about the central tendency of the data and how it is dispersed. Depending on the data type of the column, these measures might not be available.

Metadata Measures
metadata for the columns in the table, including the data type, the column length, and whether the column is a primary key candidate.

Charts
summary graphics that provide information about the uniqueness and incompleteness of column values.



Column Detail Views

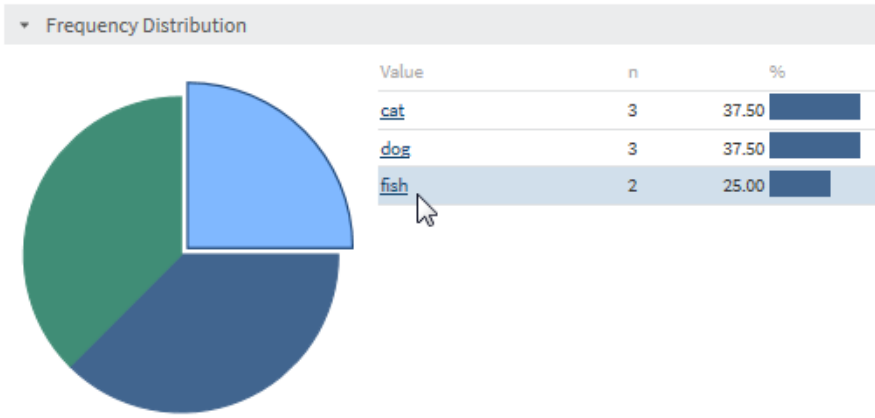
When you click on a column from the summary view in a profile report, another view is displayed that provides more detailed information about the selected column.

The detail view of a profile report includes the following information:

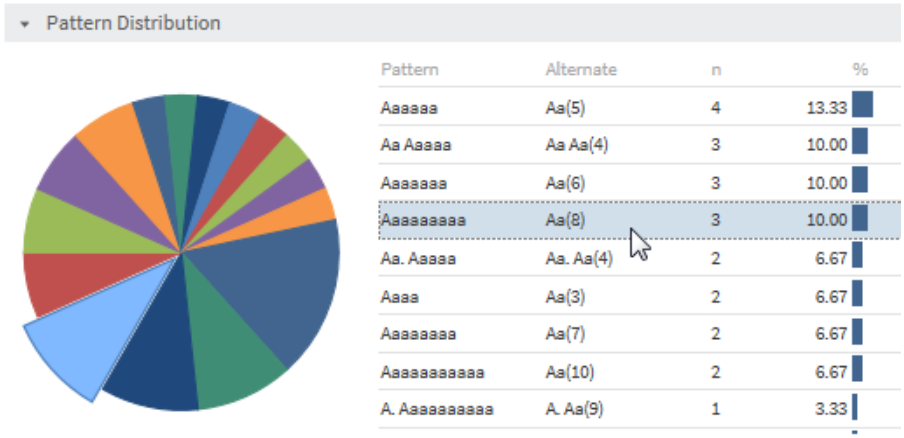
Count
the total number of rows in the table that was profiled.

Standard Metrics
a combined listing of the data quality metrics, the descriptive measures, and the metadata measures for the column that were displayed on the summary view.

Frequency Distribution
a listing of the unique values for the column, including information about how frequently a value occurs in the table. When you select a value from the list, the associated section of the pie chart is highlighted.



Pattern Distribution
a listing of the distinct pattern values that were derived from performing pattern analysis on the values for the column. The content of the pattern value describes the content of the data and indicates whether each character is uppercase, lowercase, or numeric. When you select a value from the list, the associated section of the pie chart is highlighted.

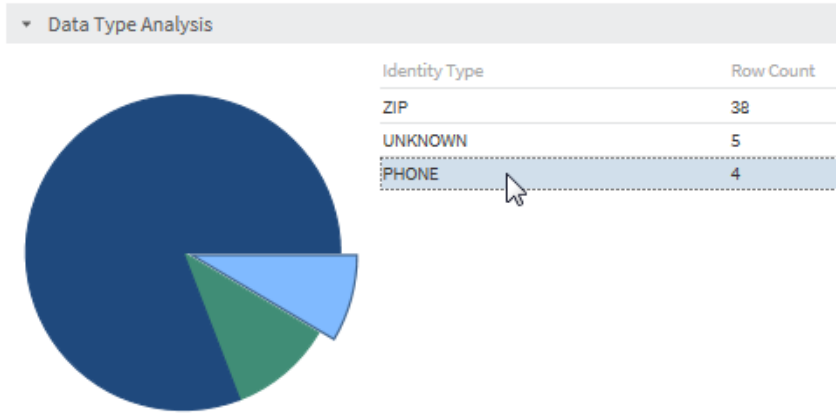


Outliers

a listing of extreme values for the column. By default, the 10 lowest values and the 10 highest values are saved, but you can change the number of outliers that are saved in the profile configuration settings. For more information, see "Profiles Panel" on page 190.

Data Type Analysis

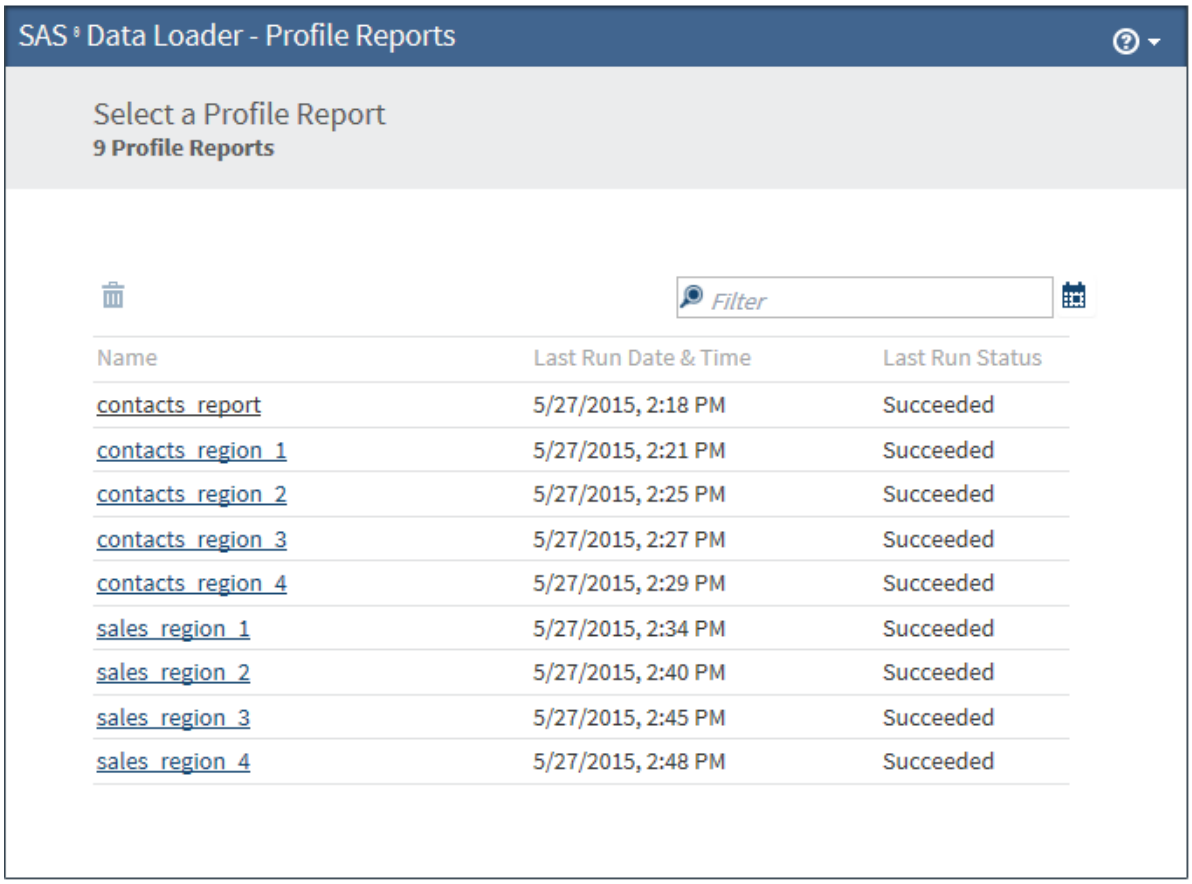
a listing of possible types of data for the information in the column, as determined by data type analysis that is automatically performed by SAS Data Loader. Results for data type analysis are available only for columns that contain string characters (for example, contact information such as name, address, state, ZIP code, and so on).



Open Saved Profile Reports

To open a saved profile report:

1 In the SAS Data Loader directives page, click the Saved Profile Reports directive to open a new browser tab. The Select a Profile Report page is displayed on the new tab:



Note: Any profile job that runs longer than five days is deleted from the Select a Profile Report page.

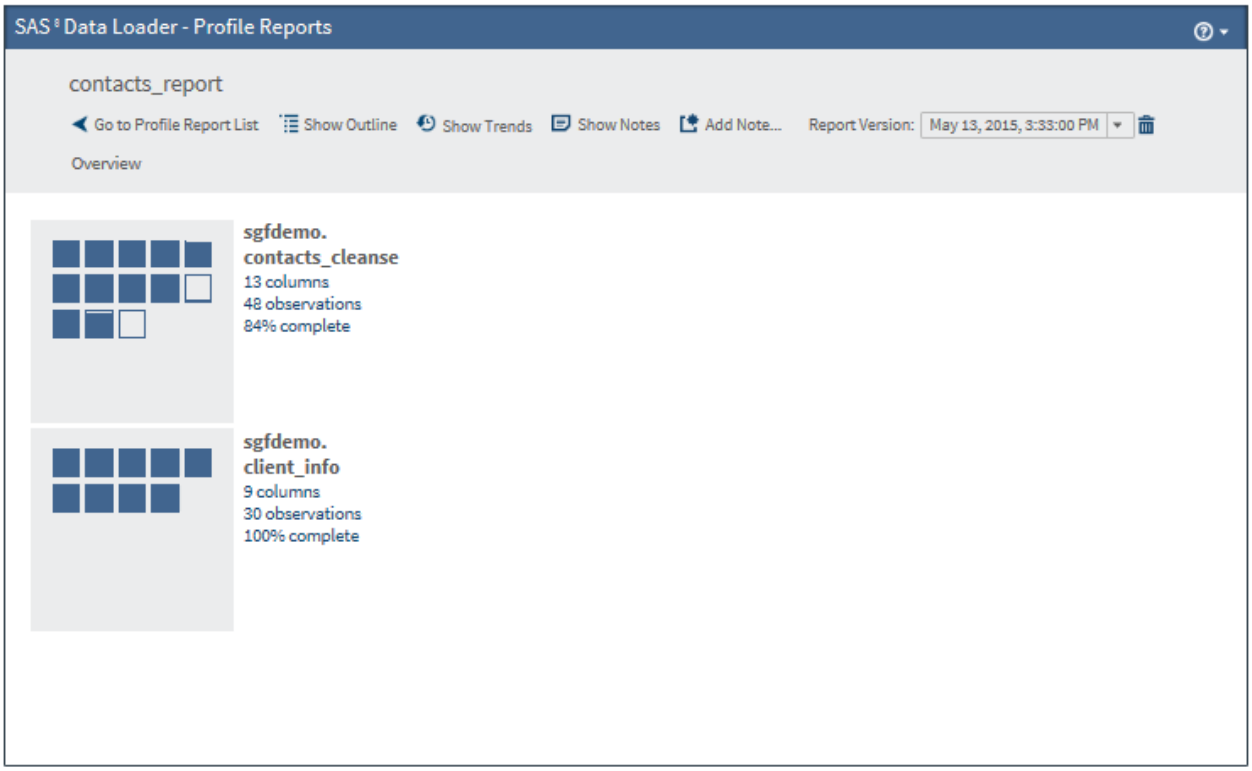
2 You can filter the list of reports using the following methods:

- Click and select a date. This filter displays profile reports that were generated on or after the selected date.
- Enter a text string into the search field.
- Click to remove the filter and restore the full list.

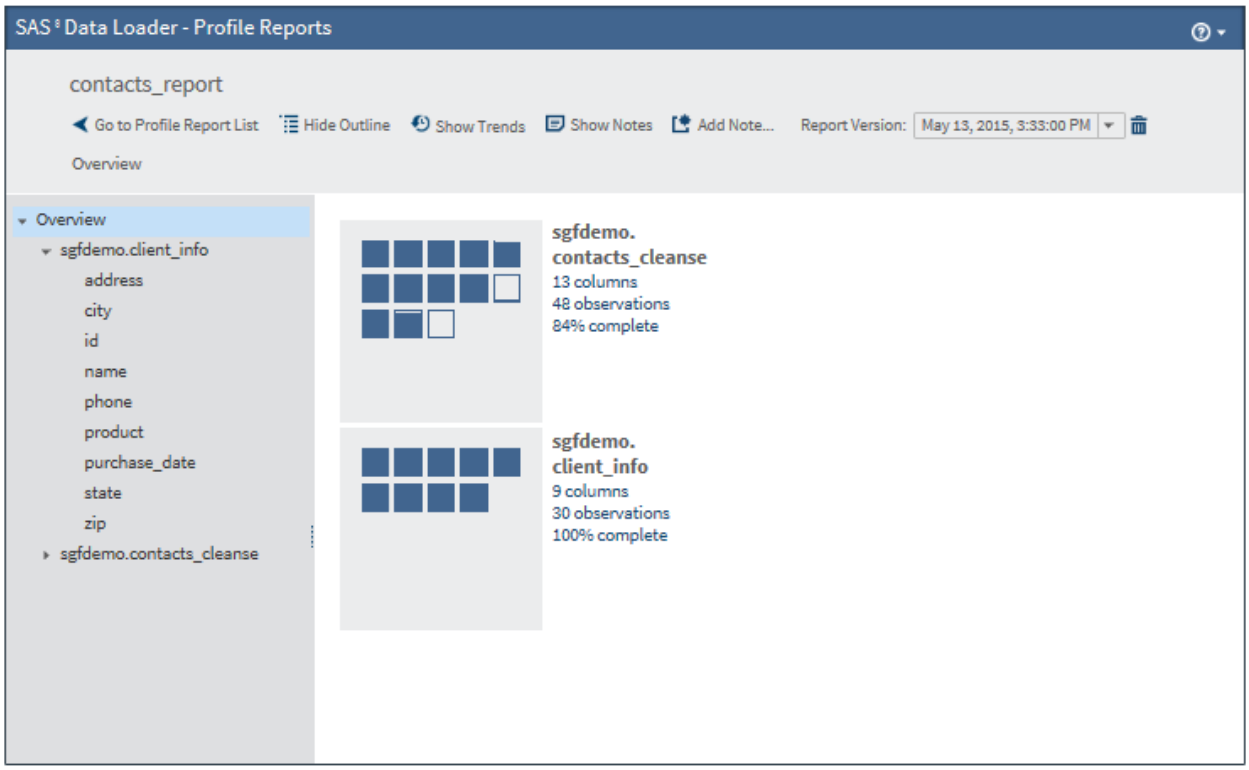
3 To delete profile reports, select one or more reports and click .

4 To open a profile report, click its name.

- If the report contains a single table, the table opens directly in the detail view shown in Step 6.
- If the report contains multiple tables, the table opens in an overview:



5 You can click a table to go directly to a more detailed view or you can click to open the outline view:

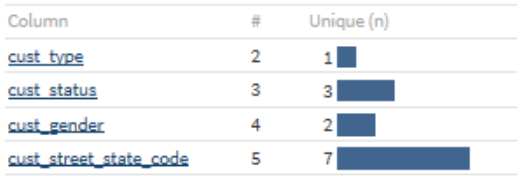


The following actions are available:

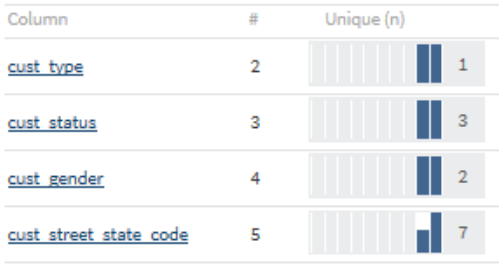
Go to Profile Report List
returns you to the Profile Report List.

Show or Hide Outline
displays or hides the outline in the left pane.

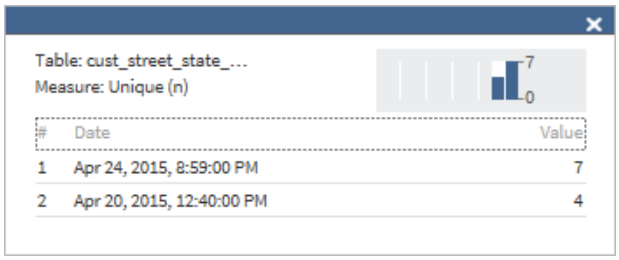
Show or Hide Trends
displays or hides the trend graphs for data that is presented in the summary view. You can use trend graphs to quickly visualize changes in the data across multiple versions of the same report. When trend graphs are not displayed, the current value of the metric is shown. For example:



When trend graphs are on, each graph displays the 10 most recent values of a metric, as determined by the selected version of the report. For example:



To view the complete list of values for the metric, you can click the trend graph. A window is displayed:

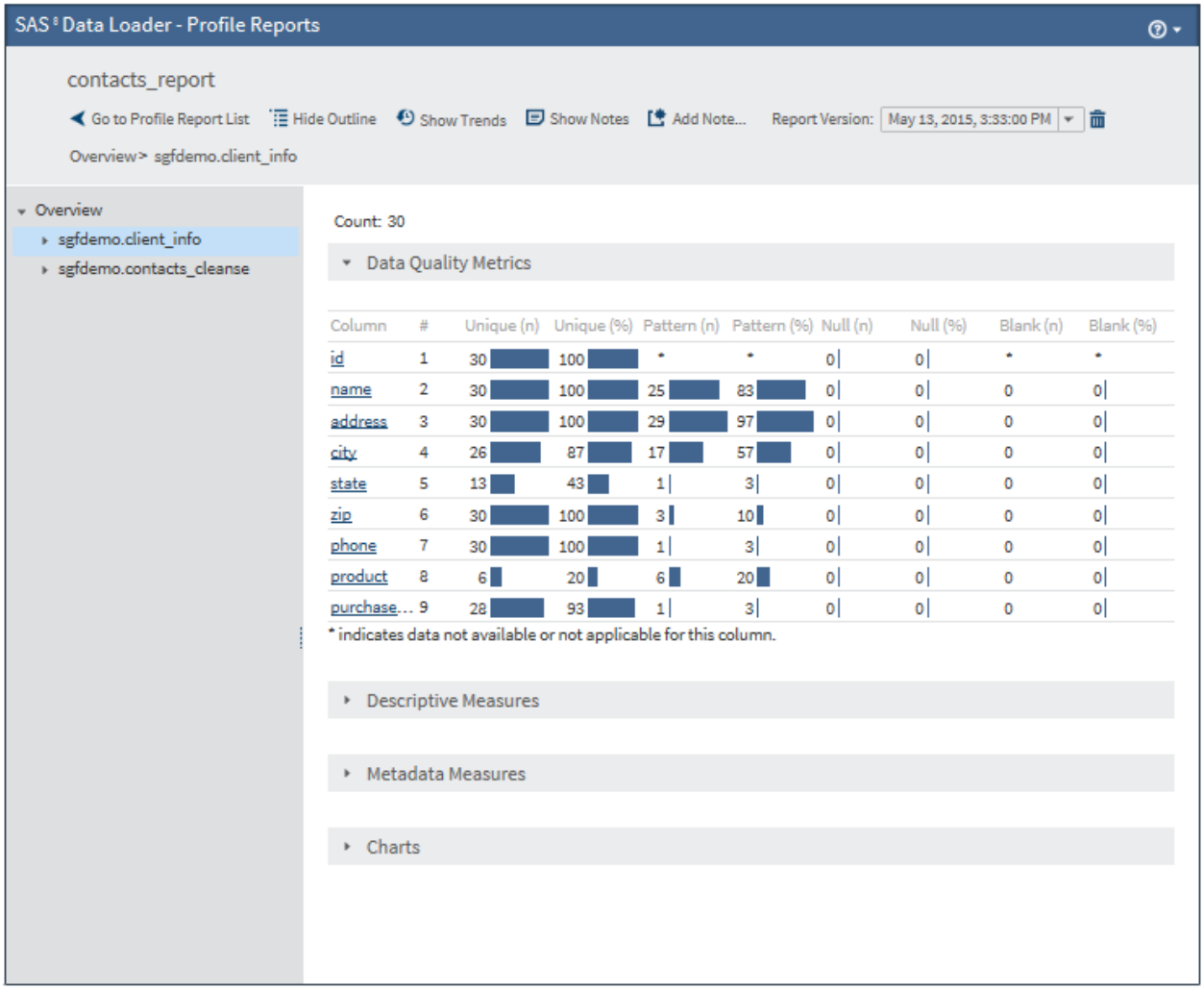


Show or Hide Notes
displays or hides notes in the right pane. You can filter the notes by entering a text string into the filter field.

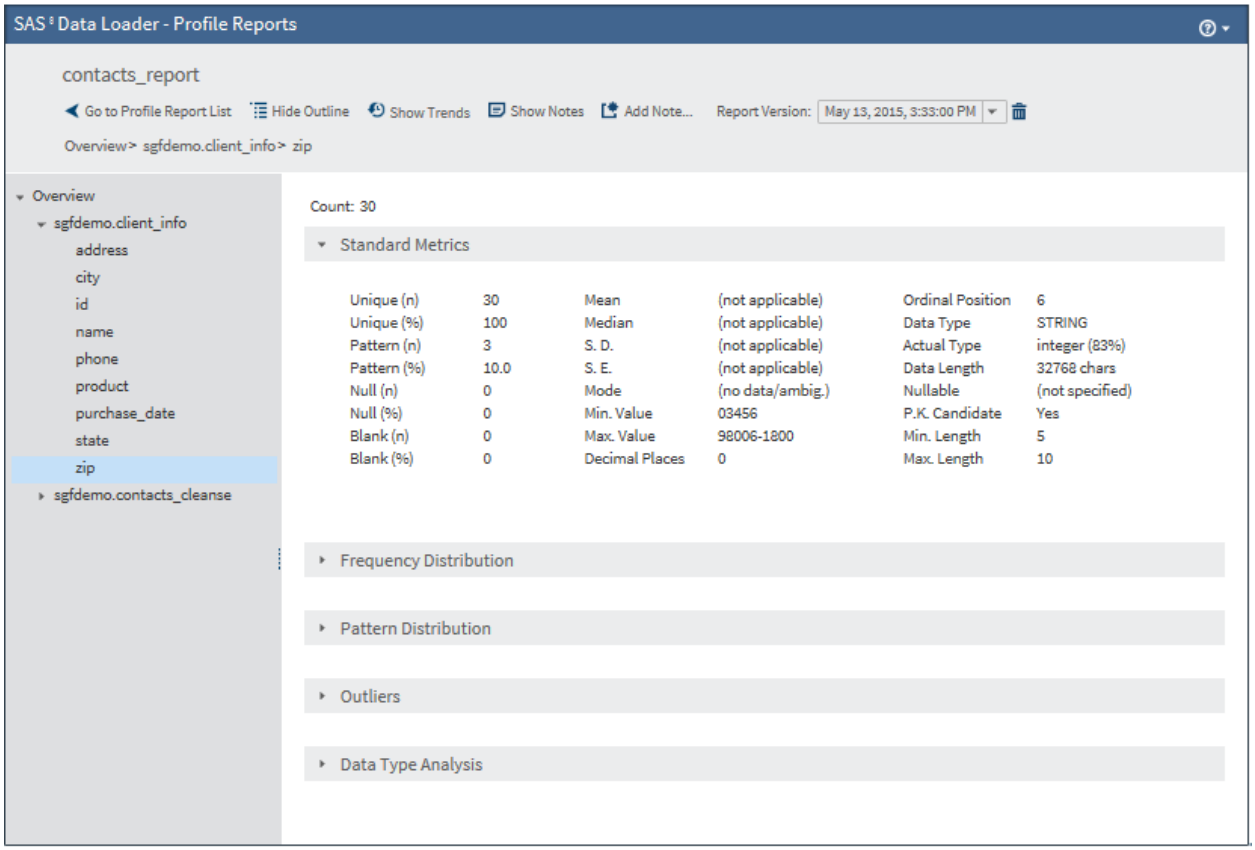
Add Note
opens a dialog box in which you can add a note.

Report Version
enables you to select the version of the report by date.

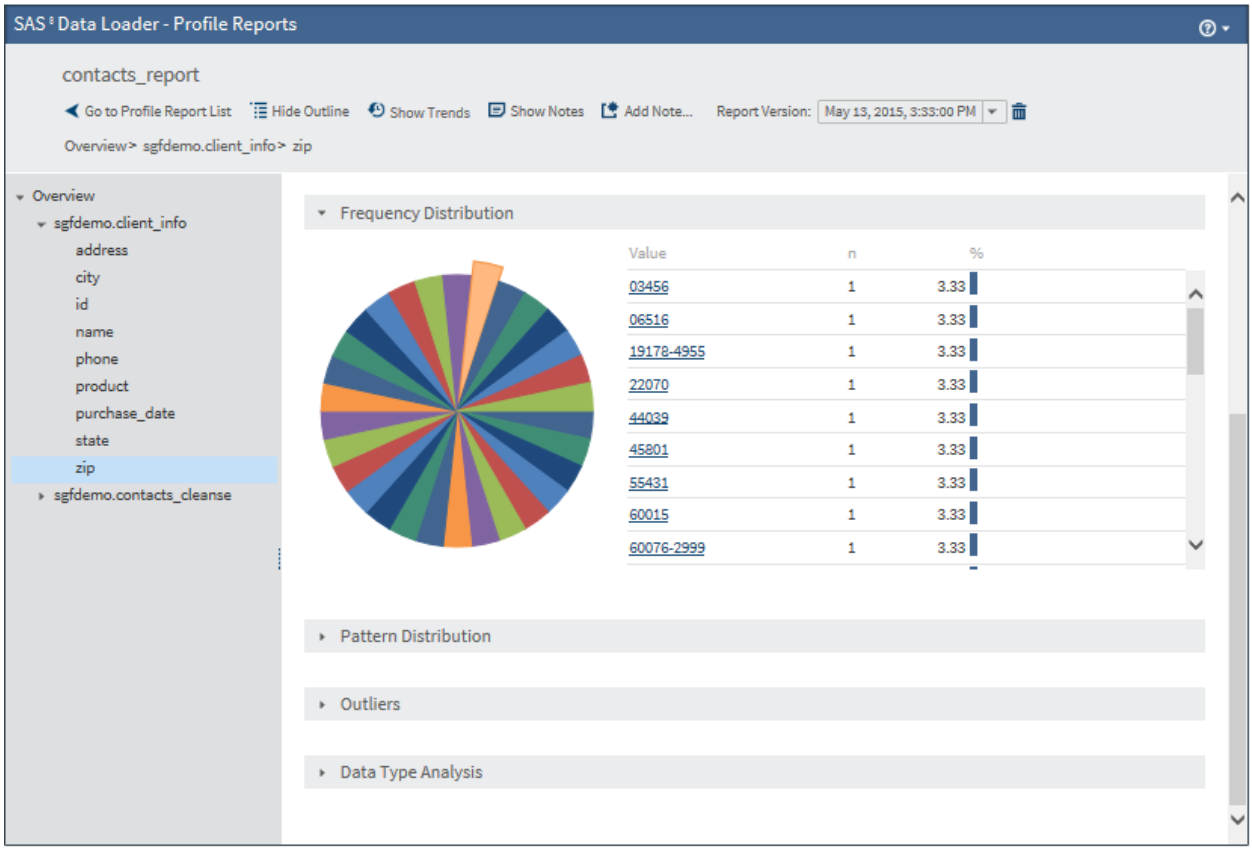
6 Select a table in the **Overview** pane or click directly on the table icon to display detailed information in the right pane. The Data Quality Metrics are displayed by default.



7 Click ▸ next to a table name to display columns. Select a column to display detailed column information in the right pane:



8 Click ▸ in the gray header bars to display the metrics in those sections. For example, clicking on Frequency Distribution icon displays the following metrics.



Clicking links in the detail view opens SAS Table Viewer.

6

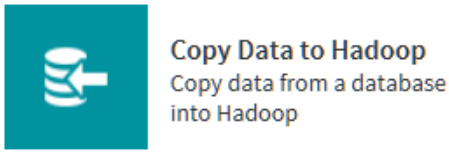
Copy Data To and From Hadoop

Overview of the Copy Data Directives

The directives Copy Data to Hadoop, Import a File, and Copy Data from Hadoop enable you to move data from your files system or database management systems into and out of Hadoop. The copy directives require database connections to be defined on the **Databases** panel of the Configuration window. For more information, see “Set Global Options” on page 176. The Import a File directive helps you import miscellaneous files from your file system into Hadoop as columnar tables.

Copy Data to Hadoop

Introduction



The Copy Data to Hadoop directive enables you to copy data from your database management systems into Hadoop. You can also copy SAS data into Hadoop.

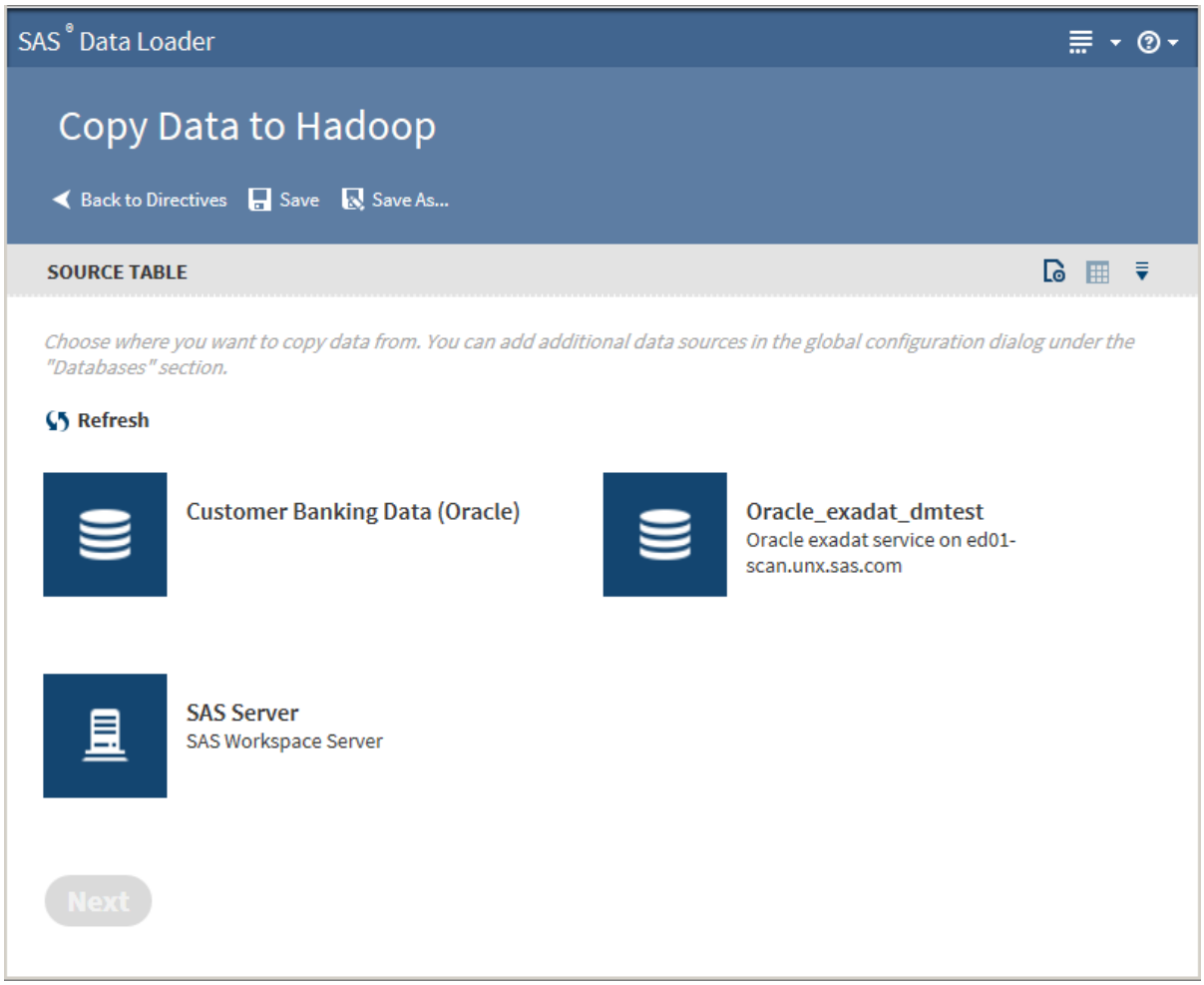
Prerequisites

When you open the Copy Data to Hadoop directive, the **Source Tables** task shows the data sources that are currently defined in SAS Data Loader. If you do not see the database from which you want to copy, you must add a connection to that database. See “Databases Panel” on page 187 for more information.

Example

Follow these steps to copy data into Hadoop from a database:

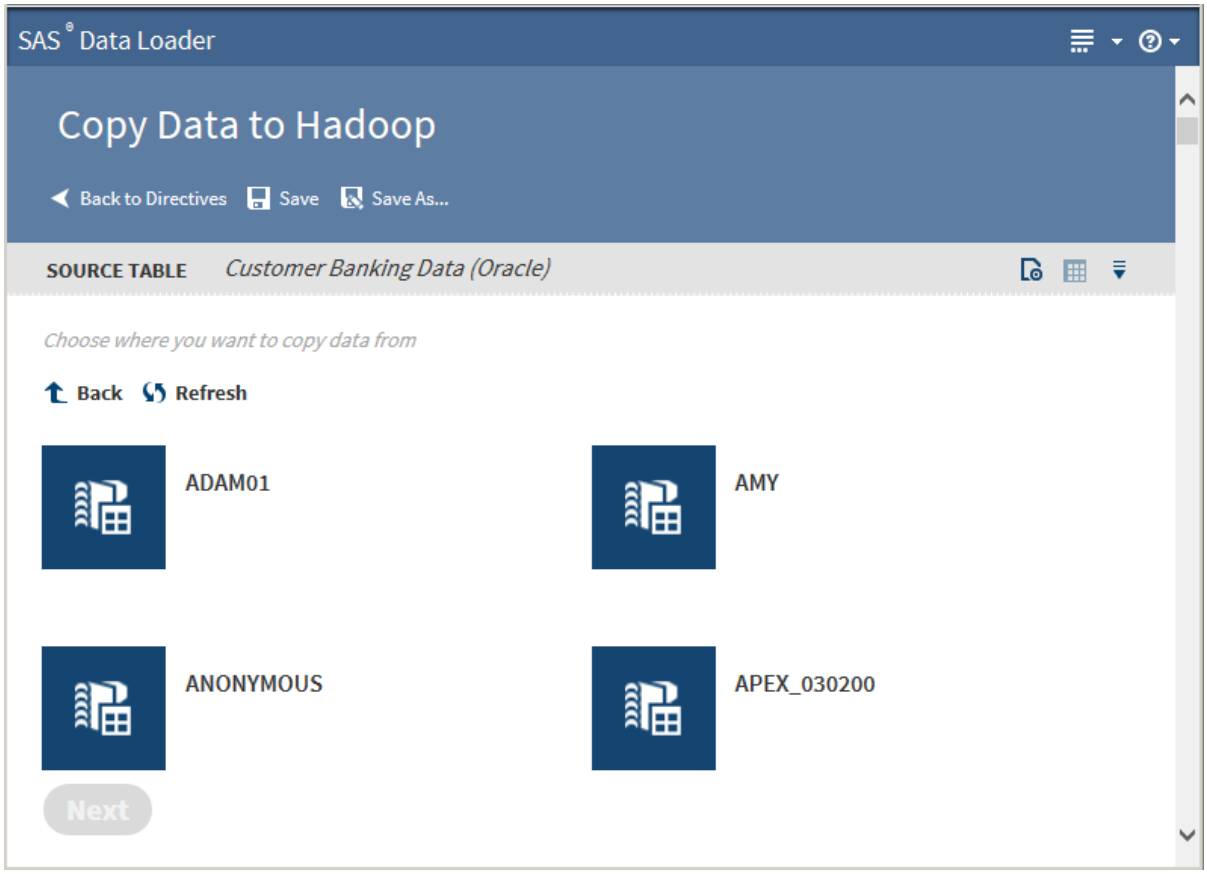
1 On the SAS Data Loader directives page, click the Copy Data to Hadoop directive. The **Source Table** task that lists available databases is displayed:



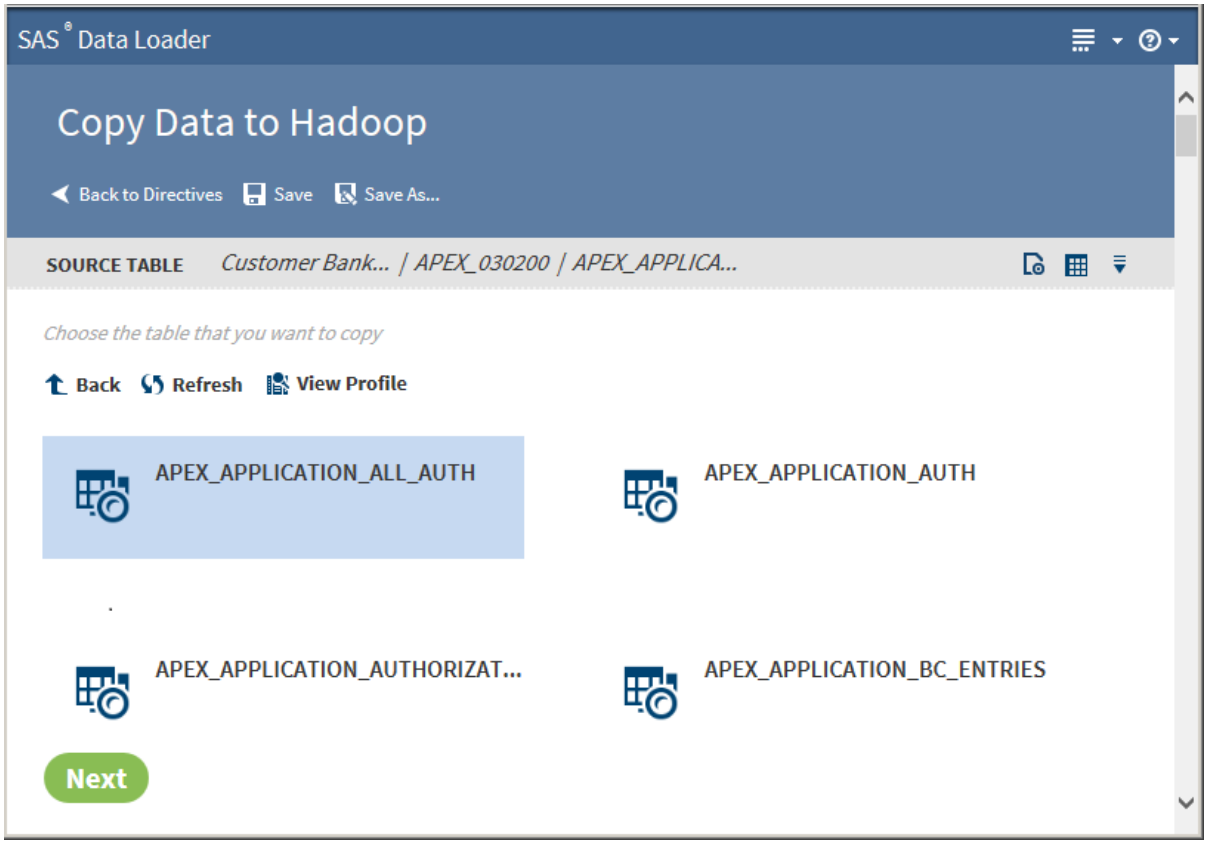
Note that the **SAS Server** data source points to the following location on the vApp host: ***vApp-shared-folder*/SASData/SAS Data Location**. To copy SAS data to Hadoop, all source tables must first be copied to this location.

Note: When you select the **SAS Server** folder, filenames are not translated. For locales other than English, this means that files that exist in the **SAS Server** folder are not displayed for selection. To work around this issue, you can import entire SAS files. In the **Source Table** task, select a file outside of the **SAS Server** folder and click through the directive. Select or create a target table. In the **Code** task, open the Code Editor to change the source file information, and then run the job.

2 Click a database to display its data sources:



3 Click a data source to display its tables:



4 Select the table from which to copy data.

> **TIP** If a profile already exists for a table, PROFILED appears beneath the table name. You can view the existing profile by selecting the table and clicking **View Profile**.

Clicking the **Action** menu  enables the following actions:

Open
opens the current directive.

Table Viewer
enables you to view sample data from a table. Select the table, and then click  to display SAS Table Viewer.

Advanced Options
opens a dialog box that enables you to modify the advanced options. The advanced options enable additional character variable length to accommodate converted non-UTF8 encoding.

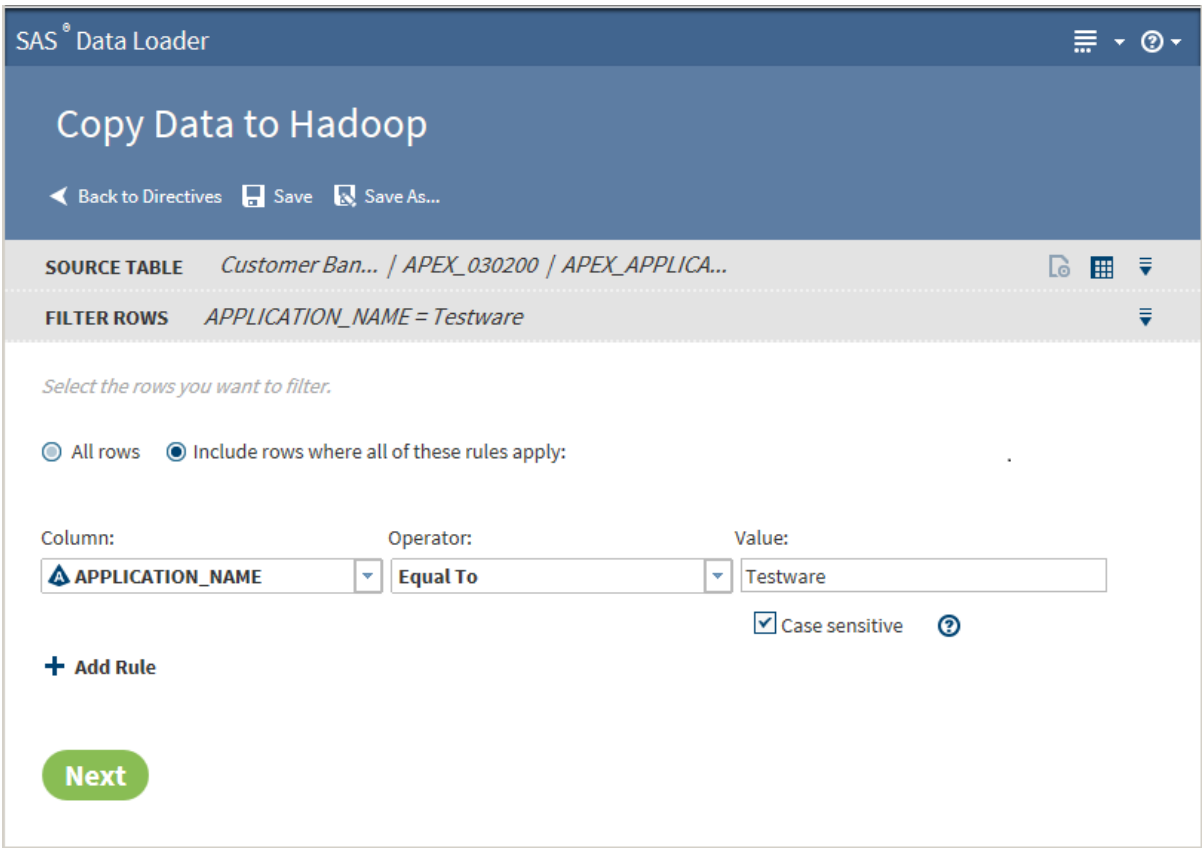
> **TIP** It is recommended that you use UTF8 encoding in SAS data when copying data from SAS to Hadoop. The vApp always uses UTF8 encoding. If you copy a non-UTF8 encoded data set from elsewhere, then the Hadoop target table is not able to accommodate all the characters. This limitation is due to the increased number of bytes when the data is converted to UTF8 encoding.

Note: Modify only one of the following two advanced options. If you fill in both fields, then the value in the multiplier field is ignored.

Number of bytes to add to length of character variables (0 to 32766)
Enter an integer value from 0 to 32766.

Multiplier to expand the length of character variables (1 to 5)
Enter an integer value from 1 to 5.

Click **Next**. The **Filter Rows** task is displayed:



5 The **Filter Rows** task enables you to filter the rows to be copied. You can select **All rows** or create filter rules. To create filter rules:

   a Select **Include rows where all of these rules apply**.

   b Select a column and an operator from the drop-down lists.

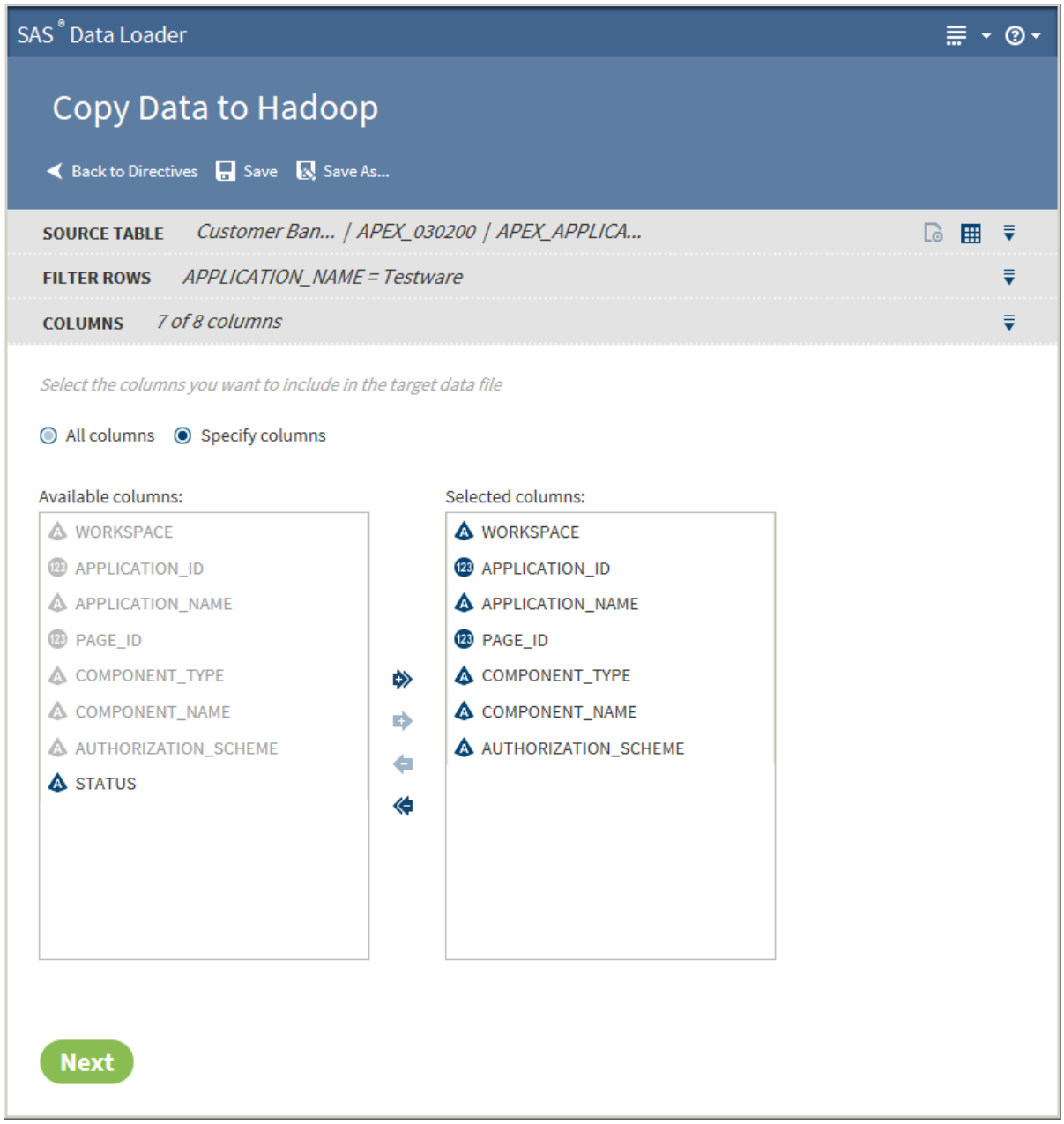
   Note: If the table for which you are defining a filter is in the OTHER database format, the database might not support all operators. You should use only those operators that are supported by your database in the filter.

   c Enter a value in the **Value** field.

   d If appropriate, select **Case sensitive** for a string value.

   e If you want to filter with additional rules, click **Add Rule**.

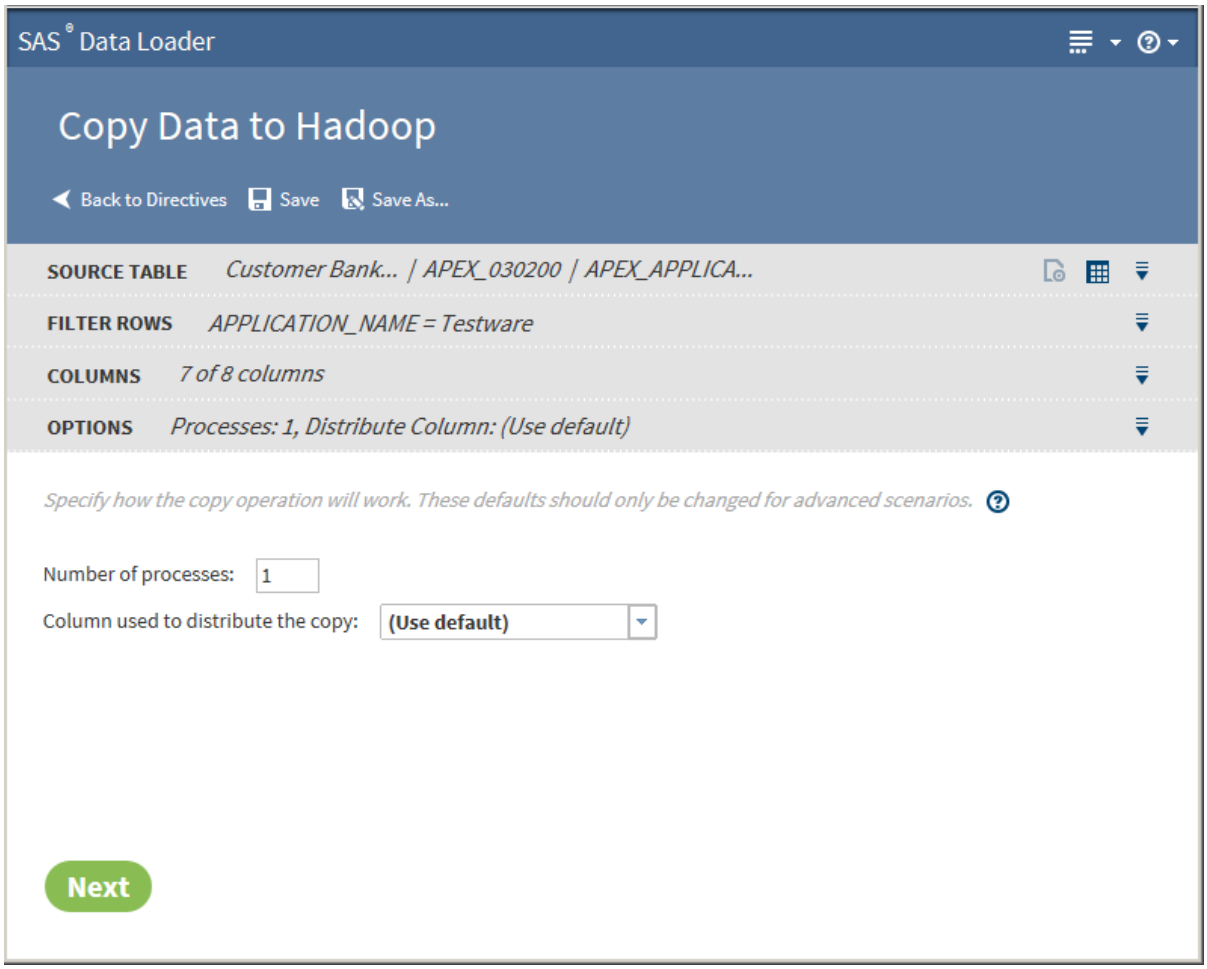
   Click **Next**. The **Columns** task is displayed:



6 The **Columns** task enables you to choose the columns to be copied. You can select **All columns** or **Specify columns**.

The columns in the **Selected columns** pane are copied to Hadoop. Select an individual column name and click or to move the column name between the **Available columns** pane and the Selected columns pane until the correct list of names appears in the **Selected columns** pane. Click or to move all column names at once.

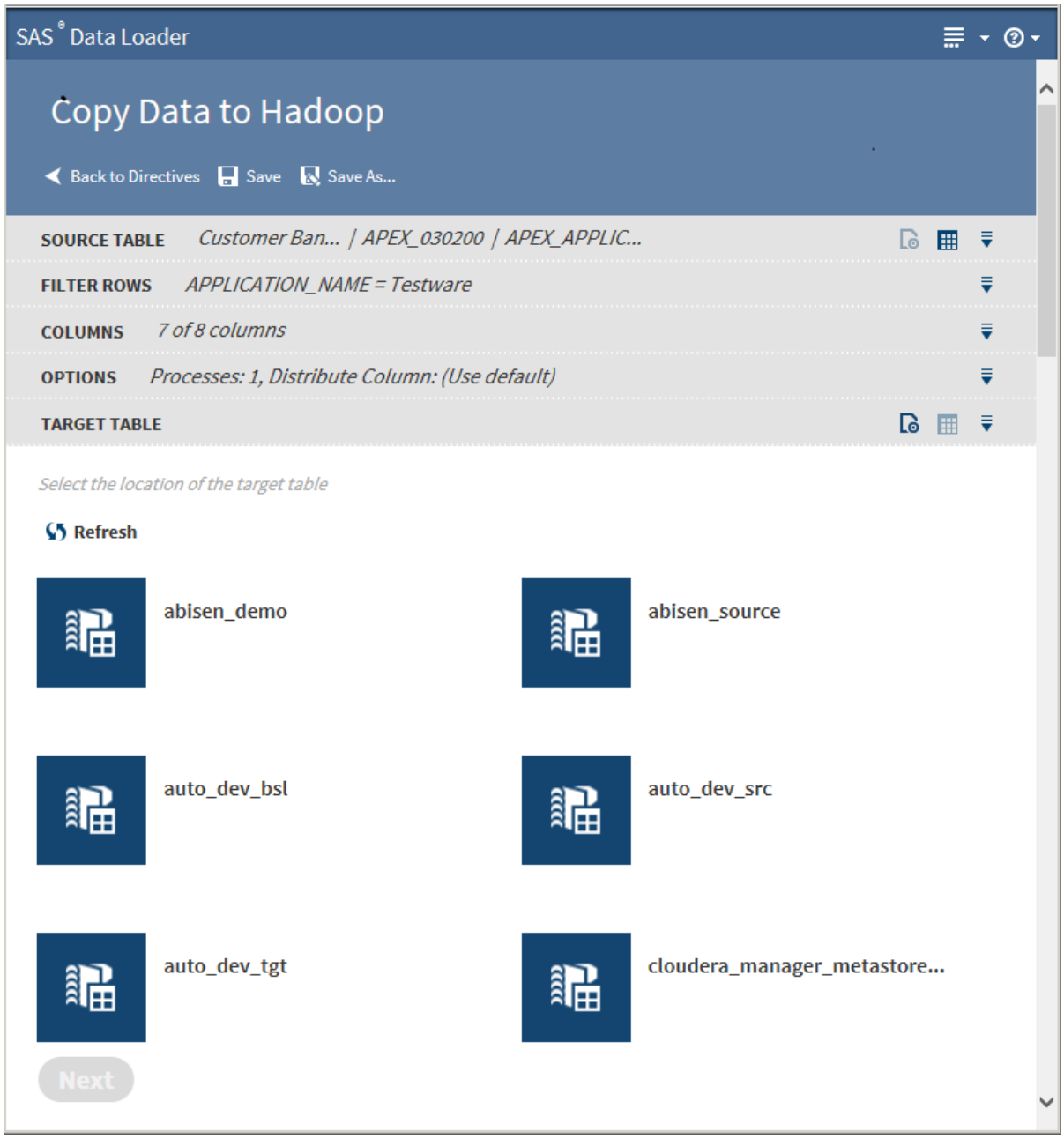
When the column selection is complete, click **Next**. The **Options** task is displayed:



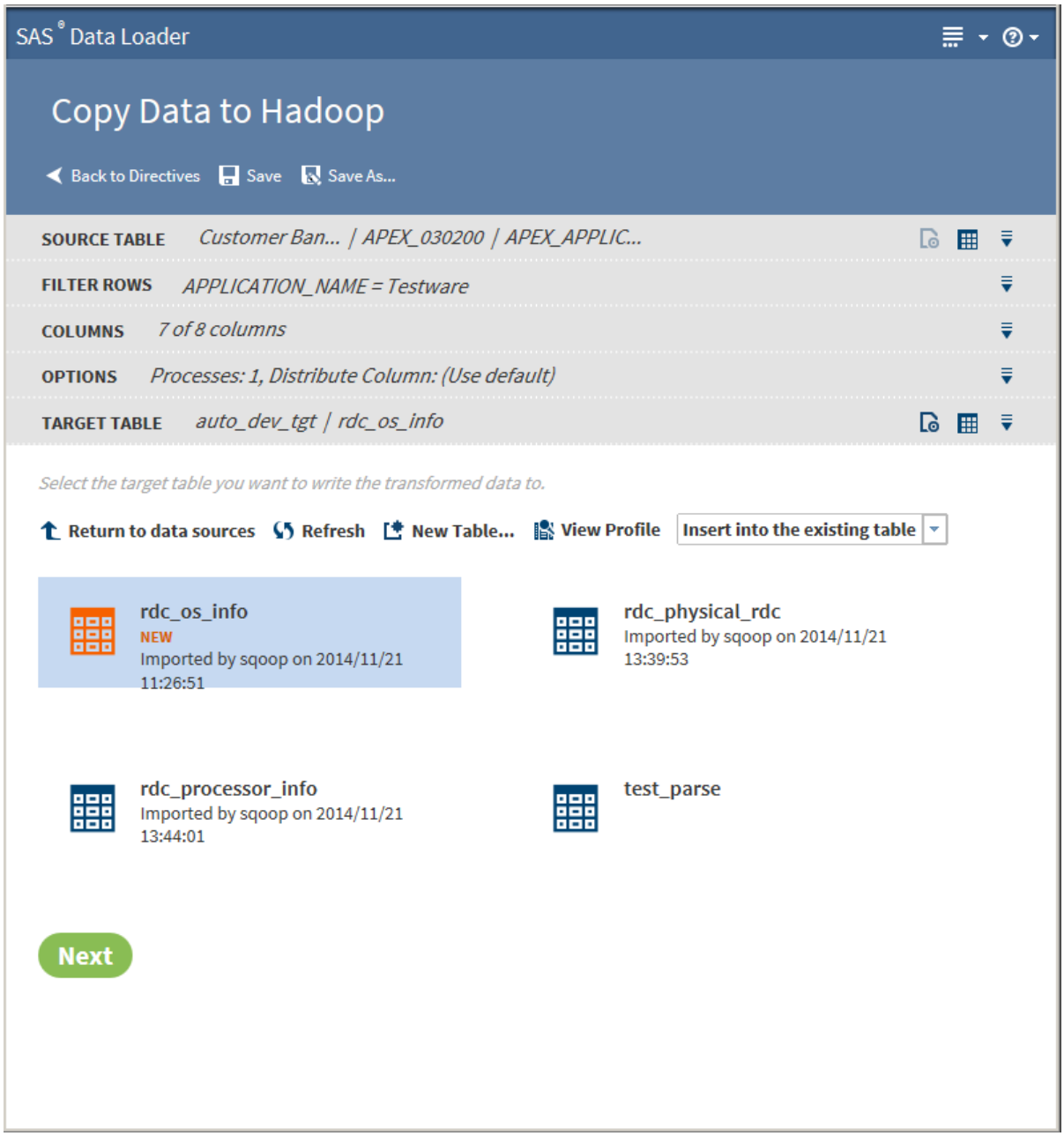
7 The values in the **Options** task should not be changed unless you have advanced knowledge of database operations.

CAUTION! If you change the number of processes, you are required to select a distribution column. Changing the number of processes to greater than one expands the number of processes and source data connections that are used to import data. When running in this mode, a column must be identified in order to distribute the data across the parallel processes. This column is typically the primary key or index of the table in the data source. Only single columns are allowed. Numeric integer values that are evenly distributed in the data are recommended

Click **Next**. The **Target Table** task is displayed with data sources:



8 Click a target data source to display its tables:



9 Select the target table to which to copy data.

TIP

- You can create a new table by clicking **New Table**.
- If a profile already exists for a table, PROFILED appears next the table icon. You can view the existing profile by selecting the table and clicking **View Profile**.

Clicking the **Action** menu enables the following actions:

Open
opens the current directive.

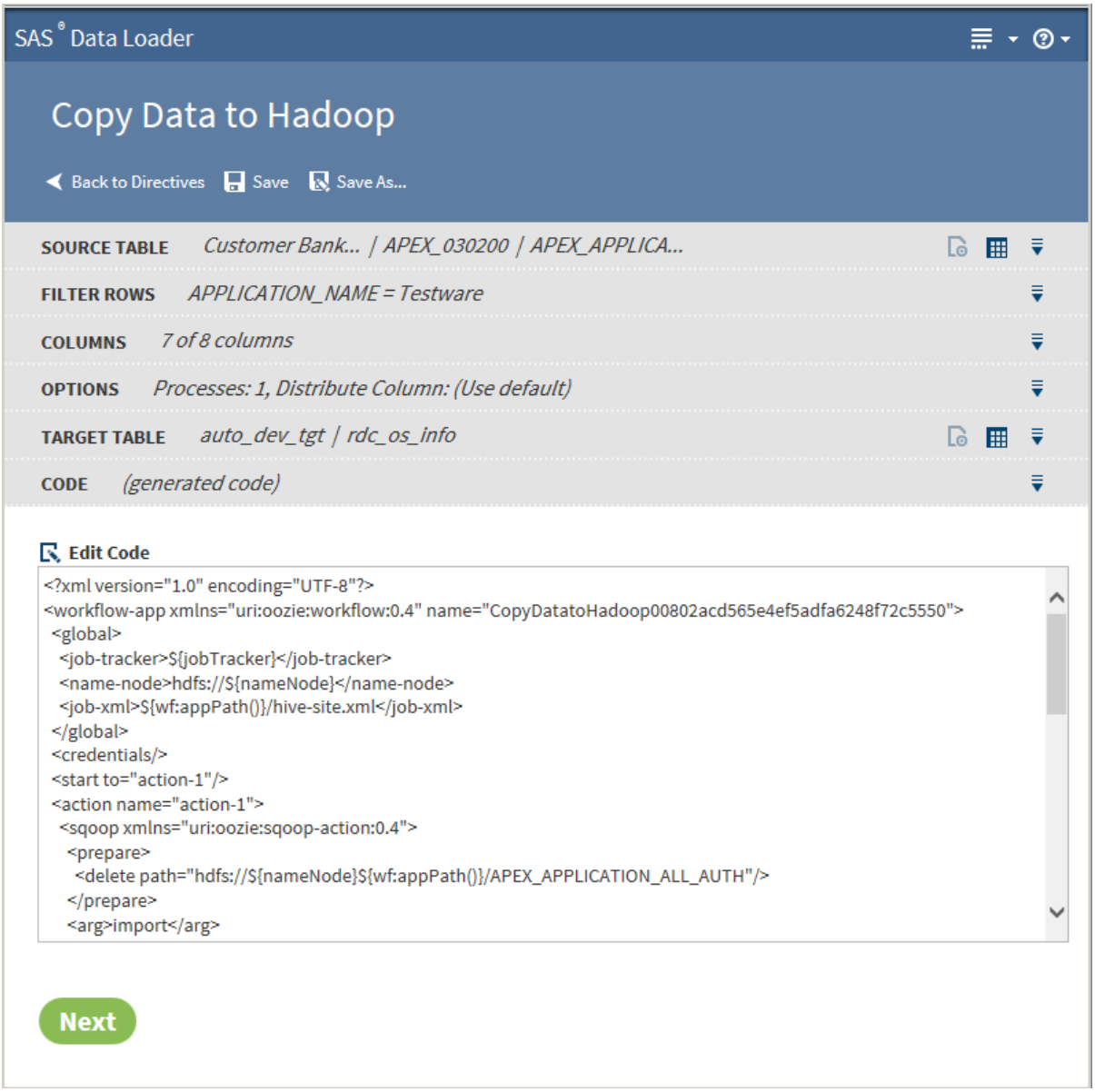
Table Viewer
enables you to view sample data from a table. Select the table, and then click to display SAS Table Viewer.

Advanced Options
opens a dialog box that enables you to modify the following advanced options:

Output table format
Use the drop-down list to select one of five output table formats: Hive default, Text, Parquet, ORC, or Sequence. The Parquet format is not supported for MapR distributions of Hadoop.

Delimiter
Use the drop-down list to select one of five output table formats: Hive default, Comma, Tab, Space, or Other.

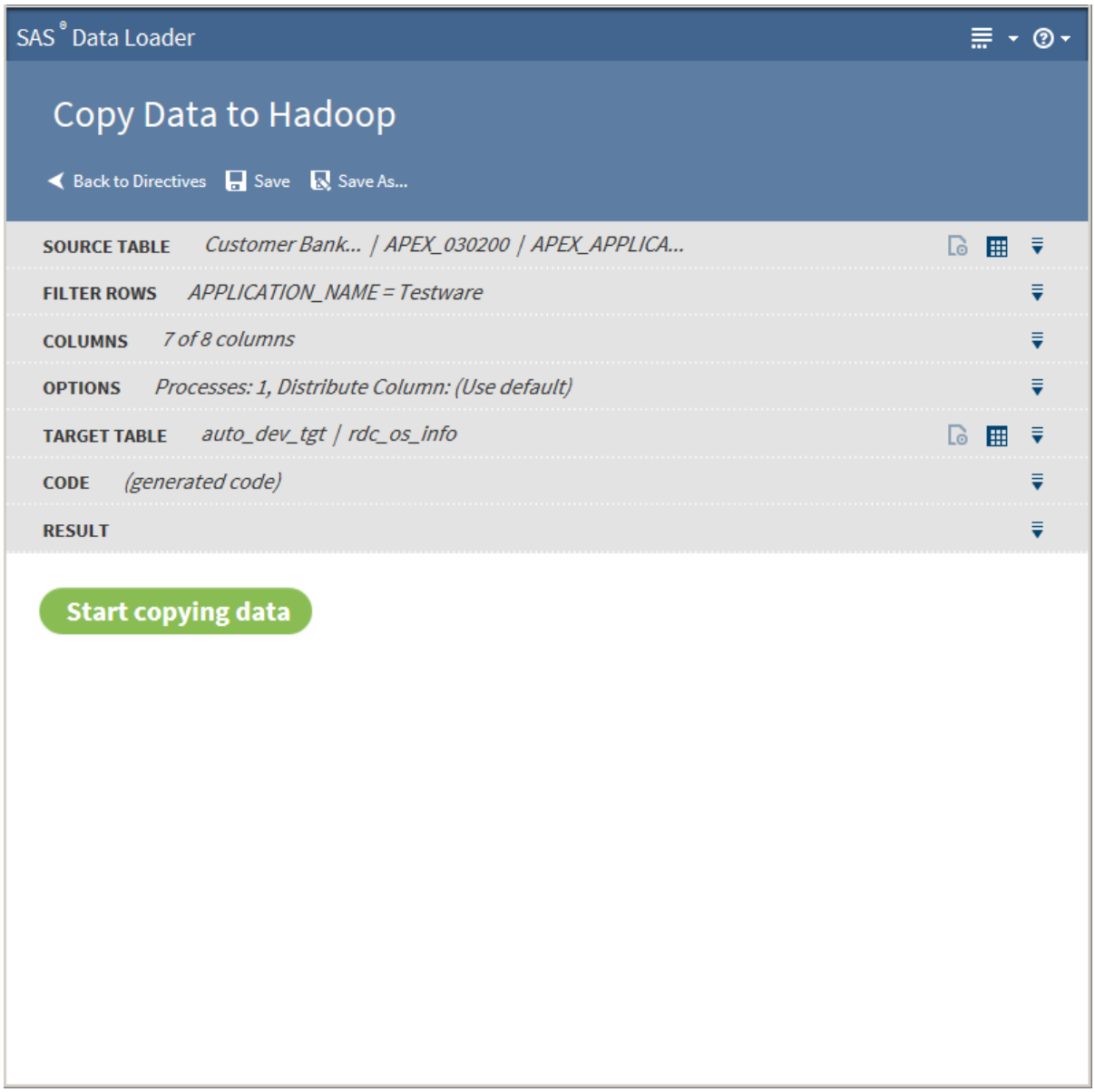
Click **Next**. The **Code** task is displayed:



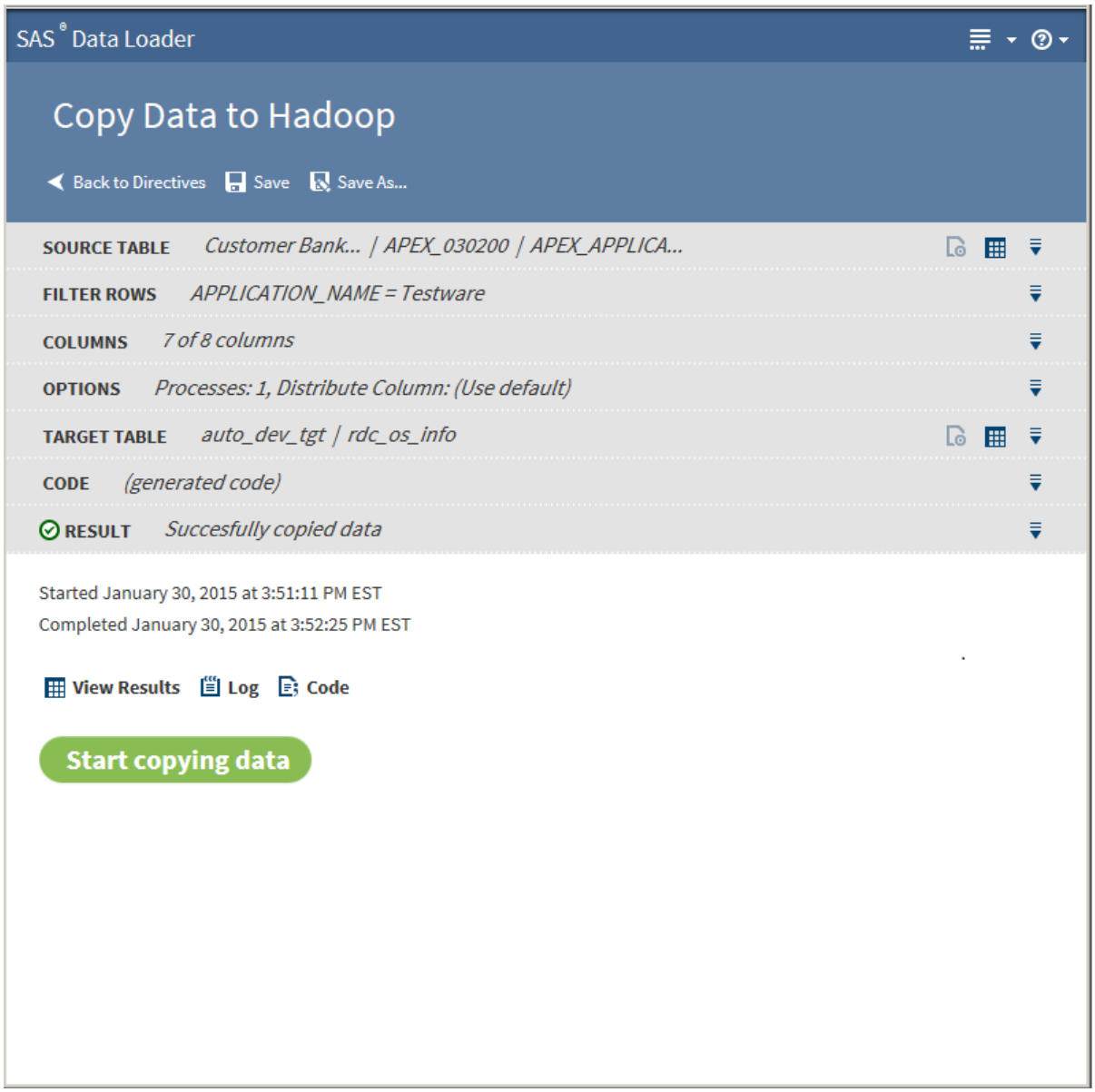
10 Click **Edit Code** to modify the generated code. To cancel your modifications, click **Reset Code**.

CAUTION! Code edits are intended to be used only to support advanced features. Code edits are not needed or required under normal circumstances.

11 Click **Next**. The **Result** task is displayed:



12 Click **Start copying data**. The **Result** task displays the results of the copy process:



The following actions are available:

View Results
enables you to view the results of the copy process in SAS Table Viewer.

Log
displays the SAS log that is generated during the copy process.

Code
displays the SAS code that copies the data.

Usage Notes

- If LDAP is used to protect your Hadoop cluster, you cannot use the Copy Data To Hadoop directive to copy data from a DBMS. For more information, see “Active Directory (LDAP) Authentication” on page 193.

- If necessary, you can change the maximum length of character columns for source tables for this directive. For more information, see “Change the Maximum Length for SAS Character Columns” on page 195.
- Error messages and log files that are produced by the Copy Data to Hadoop directive include the URL of the Oozie log file. Oozie is a job scheduling application that is used to execute Copy Data to Hadoop jobs. Refer to the Oozie log for additional troubleshooting information.
- When copying data from Teradata:
  - In Cloudera 5.2 or later, the Teradata source table must have a primary key defined, or you must specify a distribution column on the Options task.
  - In Hortonworks 2.1 or later, you are required to insert Teradata data into existing tables. The creation or replacement of tables is not supported. This is due to a limitation in the HortonWorks Sqoop connector. One workaround is to ask your Hadoop administrator to drop an existing table, and then create an empty table with the desired schema. At that point, you can use the Append option in the Copy Data to Hadoop directive to copy a Teradata table into the empty table. For more information, see Step 9 on page 134 in the Example section.
- When copying data from SQL Server, note that SQL Server does not support the SQL standard syntax for specifying a Date literal, which is: `DATE 'date_literal'`. Edit the generated code and remove the word DATE that appears prior to the quoted date literal. For example, you would change `( table0.BEGDATE >= DATE '1990-01-01' )` to `( table0.BEGDATE >= '1990-01-01' )`. For more information about the Code task, Step 10 on page 136 see in the Example section.
- When copying data from Oracle, note that Oracle table names must be uppercase.

Import a File

Introduction

Import a File
Import data from a file into Hadoop

Use the Import a File directive to copy a delimited source file into a target table in HDFS and register the target in Hive.

The directive samples the source data and generates default column definitions for the target. You can then edit the column names, types, and lengths.

To simplify future imports, the Import a File directive enables you to save column definitions to a file and import column definitions from a file. After you import column definitions, you can then edit those definitions and update the column definitions file.

The directive can be configured to create delimited Text-format targets in Hadoop using an efficient bulk-copy operation.

In the source file, the delimiter must be a single character or symbol. The delimiter must have an ASCII character code in the range of 0 to 127 (\000 to \177 octal).

To learn more about delimiters and column definitions files, see the following example.

To copy database tables into Hadoop using a database-specific JDBC driver, use the "Copy Data to Hadoop" directive.

Example

Follow these steps to use the Import a File directive:

1 Copy the file to be imported, or copy a directory of files to be imported, into the directory ***vApp-install-path**shared-folder*\Files\MyData**. A common name for the vApp shared folder is **SASWorkspace**.

2 On the SAS Data Loader directives page, click **Import a File**.

3 In the **Source File** task, click to open folders as needed, click the file that you want to import, and click **Next**.

> **TIP** To open or save a copy of a selected file, click  and select **Download**.

4 In the **File Specification** task, click the **View File** icon. Identify the delimiter that separates the variable values. Check to see whether the delimiter is used as part of a variable value.

Notes:

- In the source file, all variable values that contain the delimiter character must be enclosed in quotation marks (**"**).
- In Hadoop distributions that run Hive 13 (0.13.0) or earlier, a backslash character (\ ) is inserted into the target when the delimiter appears in source values. For example, the source data **One, "Two, Three", Four** would be represented in the target as Column A: **One**, Column B: **Two\, Three**, and Column C: **Four**. In Hive 14 (0.14.0) and later, the backslash character is not inserted into the target.

5 Click **Input format delimiter** to display a list of available delimiters. Click the delimiter that you see in your source file, or click **Other**. If you clicked **Other**, then enter into the text field the single-character delimiter or the octal delimiter that you see in the source file. Octal delimiters use the format *nnn*, where *n* is a digit from 0 to 7. The default delimiter in Hive is \001.

Note: Input delimiters must have ASCII character codes that range from 0 to 127, or octal values that range from \000 to \177.

6 To efficiently register and store the source data in Hadoop using a bulk-copy, select (add a check mark to) **Use the input delimiter as the delimiter for the target table**. The bulk-copy operation is efficient, but the source data is not analyzed or validated in any way. For example, the directive does not ensure that each row has the expected number of columns.

Note:

- The bulk-copy operation is used only if the next two options are not selected. If this condition is met, then the source file is bulk-copied to the target. The format of the target is Text. The Text format is used even if another format is specified in the **Target Table** task.
- If your source file uses `\N` to represent null values, you can preserve those null values in the target. A bulk-copy operation is required. In Hive, the default null value is `\N`.

CAUTION! Newline characters should be found only at the end of each record in the source data. If your source data contains newline characters within the field data, bulk-copies will include newline characters in the target without generating error messages or log entries. Newline characters in the target can cause data selection errors. Remove newline characters from the source as needed to create a usable target in Hadoop.

7 If the source file is formatted accordingly, select **Check the input file for records wrapped in quotation marks (")**. Quotation marks are required when the delimiter appears within a variable value.

Quotation marks that appear inside a quoted value must be represented as two quotation marks (**""**).

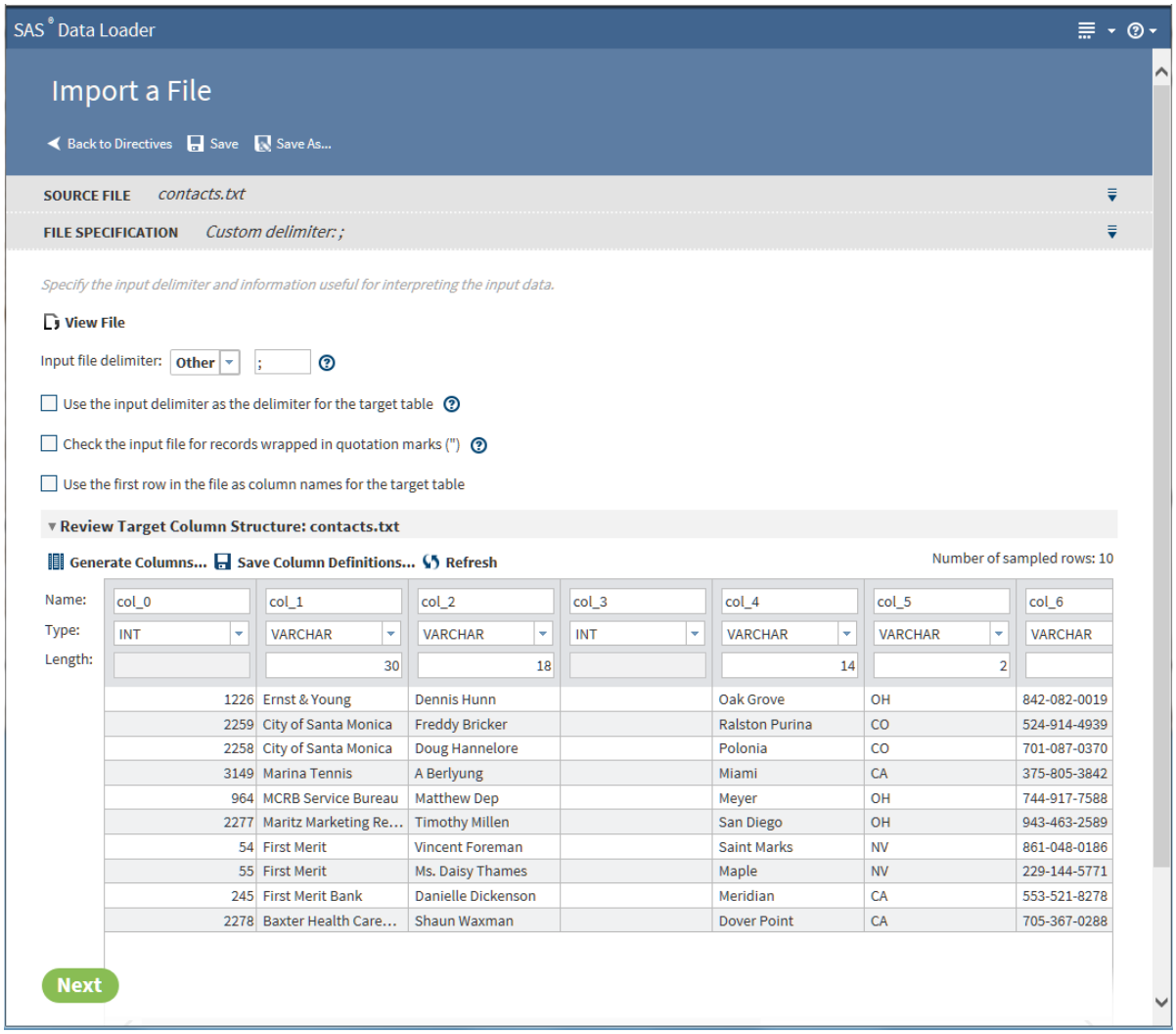
CAUTION! Except for bulk-copy operations, jobs will fail if the source contains newline characters. For all jobs other than bulk-copies, ensure that the source file does not contain newline characters.

8 If your source file includes column names in the first row, then select **Use the first row in the file as column names for the target table**.

9 Click **Review Target Column Structure** to display a sample of the target table. The target columns are displayed with default column names (unless they were specified in the source), types, and lengths (based on type.) Review and update the default column definitions as needed, or apply a column definitions file as described in subsequent steps.

Note: The default column definitions are generated by a programmatic analysis of sample data. You can change the default column definitions using the fields **Type** and **Length**. For example, a default column type could be DOUBLE. If you felt that the BIGINT type would be more useful, then you could select that type from the **Length** field.

TIP To display a larger data sample, click the **Generate Columns** icon. In the Generate Columns window, enter a new value for **Number of rows to sample**.



CAUTION! Time and datetime values in the source must be formatted in one of two ways in order for those columns to be assigned the correct type in the target. To accurately assign a column type, the directive requires that the source file use a DATE column format of `YYYY-MM-DD` and a DATETIME column format of `YYYY-MM-DD HH:MM:SS.fffffffff`. The DATETIME format requires either zero or nine decimal places after the seconds value `SS`. Source columns that do not meet these requirements are assigned the VARCHAR type. In the directive, you can manually change a column type to DATE or TIMESTAMP. If the data in that column is improperly formatted, subsequent queries can return unexpected values.

10 When your columns are correctly formatted, you can save your column definitions to a file. You can then reuse that file to import column definitions the next time you import this or another similar source file. To generate and save a column definitions file, click the **Save Column Definitions** icon. In the Save Column Definitions window, enter a filename to generate a new file, or select an existing file to overwrite the previous contents of that file. Click **OK** to save your column definitions to the designated file.

11 If you previously saved a column definitions file, and if you want to import those column definitions to quickly update the defaults, then follow these steps:

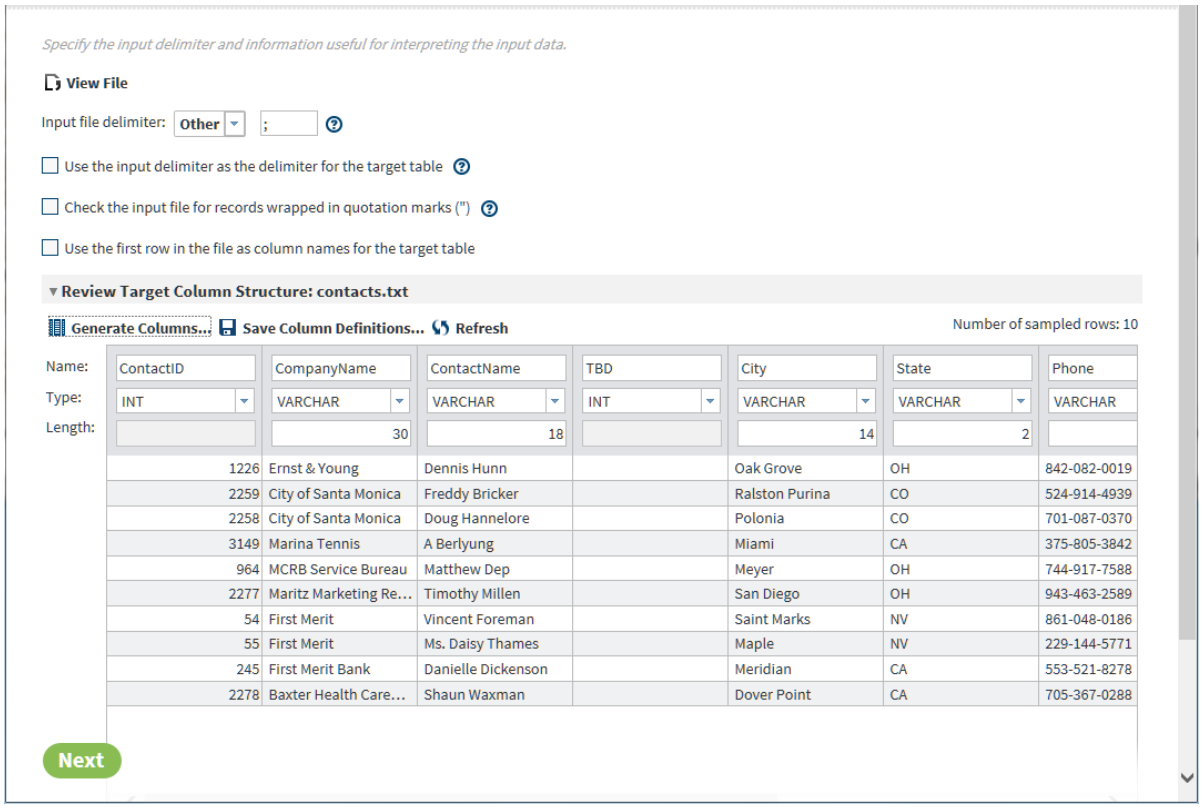
a Click the **Generate Columns** icon.

b In the Generate Columns window, click **Use column definitions from a format file**, and enter the filename or select the file using ... to display the Select a Format File window.

c As needed in the Select a Format File window, click to open folders, select a column definitions file, and click **OK**.

> **TIP** Use the Select a Format File to manage your column definitions files (refresh, rename, delete.) You can also download the files as needed.

d In the Generate Columns window, click **Generate** to close the window and format the target columns as specified in the column definitions file.



> **TIP** As is the case with the default column definitions, you can enter changes to imported column names, types, and lengths. You can then save your changes to the original column definitions file or to a new file.

> **TIP** During the definition of columns, you can replace your changes with the default column definitions at any time. Select Generate Columns..., click **Guess the columns based on a sample of data** , and click **Generate**.

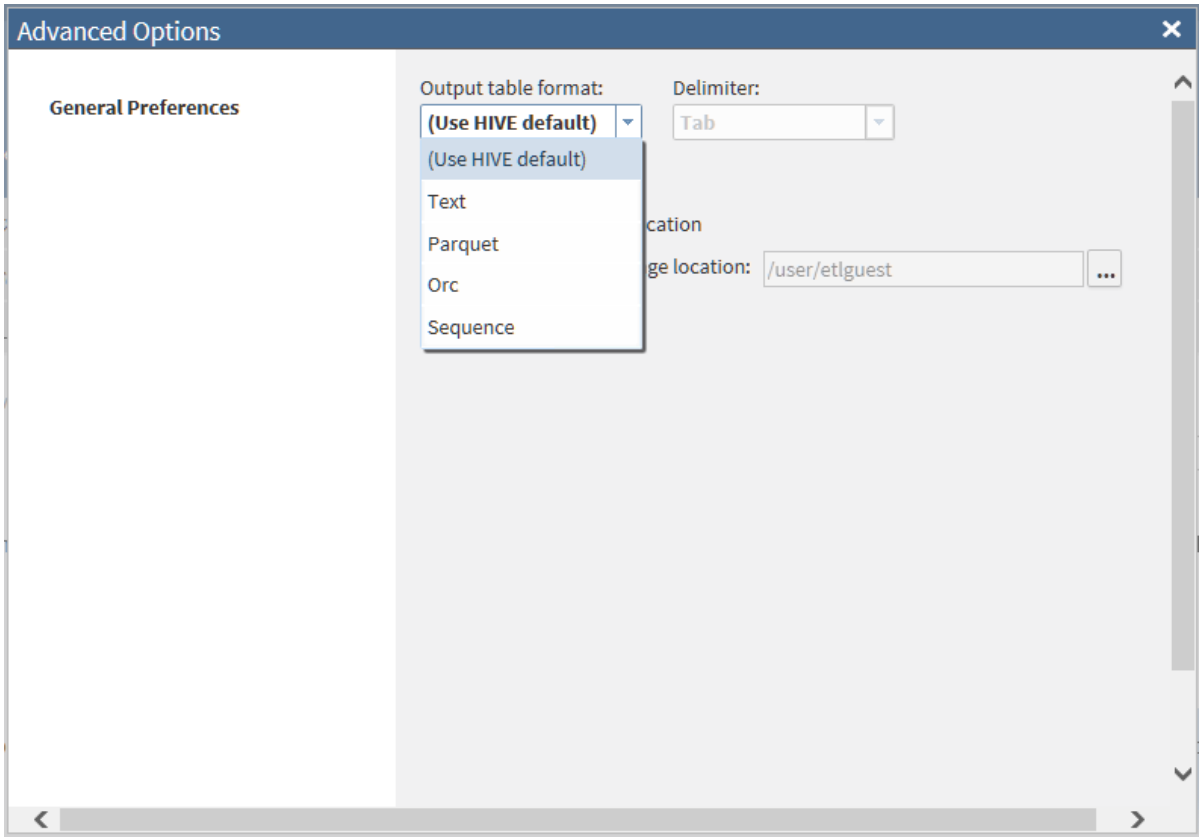
12 In the **Target Table** task, click to open a data source and select a target, or click Select Recent Table and choose a target. Existing targets are overwritten entirely when you run your job.

To name a new target table, select a data source and click the **New Table** icon, enter the new table name, and click **OK**.

13 The format of the target table is specified by default for all new directives in the Configuration window. To see the default target format, click the **More** icon, and then select **Configuration**. In the Configuration window, click **General Preferences**.

To override the default target file format for this one target, click the target and click **Advanced Options**.

Note: If you are using a bulk-copy operation, as described in Step 6, then the target will always receive the Text format, regardless of the selections in the Advanced Options and Configuration windows.



To save the table data to a non-default Hive storage location, click **Specify alternate storage location**, and then click ... . You need appropriate permission to store your imported table or file to a non-default Hive storage location.

When your target selection is complete, click **Next**.

14 In the **Result** task, click **Start Importing Data** to generate code and execute your job. You can monitor long-running jobs in the Run Status directive. At the completion of execution, you can click the **Code**, **Log**, and possibly the **Error Details** icon to learn more about your job.

15 Click **Save** or **Save As** to retain your job in **Saved Directives**.

Copy Data from Hadoop

Introduction

The Copy Data from Hadoop directive enables you to copy data from Hadoop into database management systems such as Oracle and Teradata.

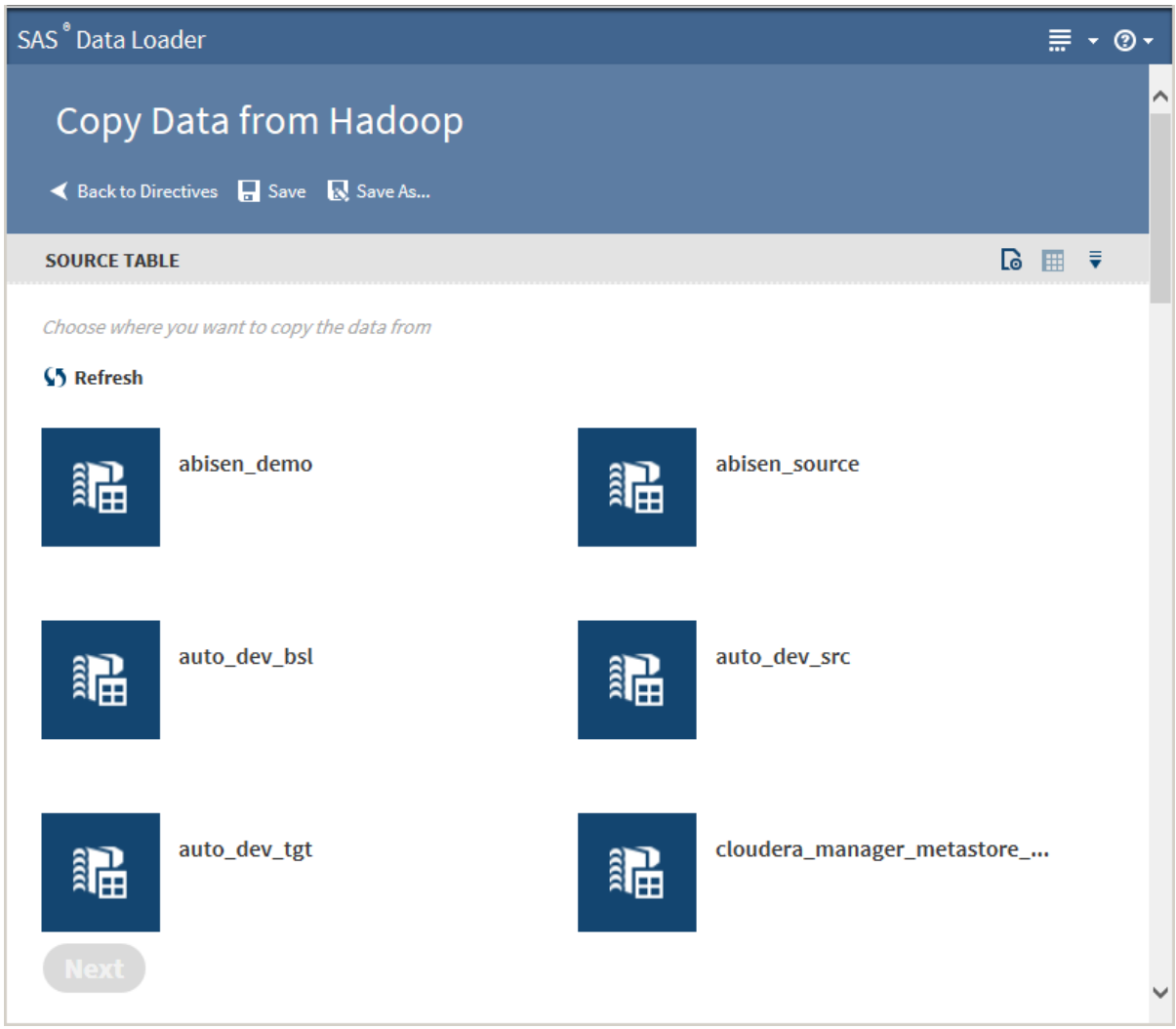
Prerequisites

When you open the Copy Data from Hadoop directive, the **Source Tables** task shows the data sources that are on the Hadoop cluster. When you come to the **Target Tables** task, you see the databases that are defined in SAS Data Loader. If you do not see the database to which you want to copy, you must add a connection to that database. To create or update database tables, the credentials that are specified in the database connection require appropriate permission. For more information, see “Databases Panel” on page 187.

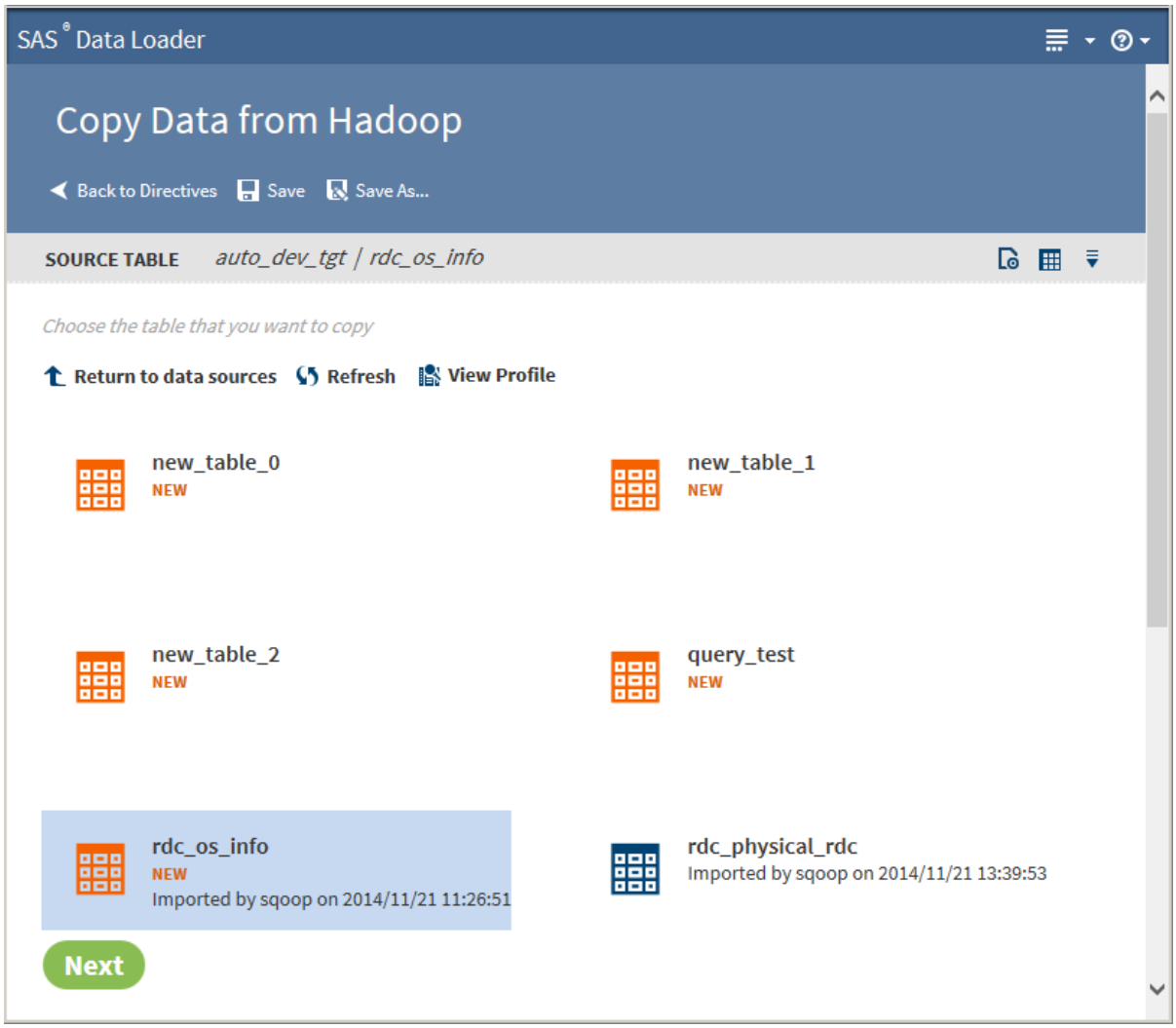
Example

Follow these steps to copy data from Hadoop into a database:

1 On the SAS Data Loader directives page, click the Copy Data from Hadoop directive. The **Source Table** task that lists available data sources is displayed:



2 Click a data source to display its tables:



3 Select the table from which to copy data.

Clicking the **Action** menu enables the following actions:

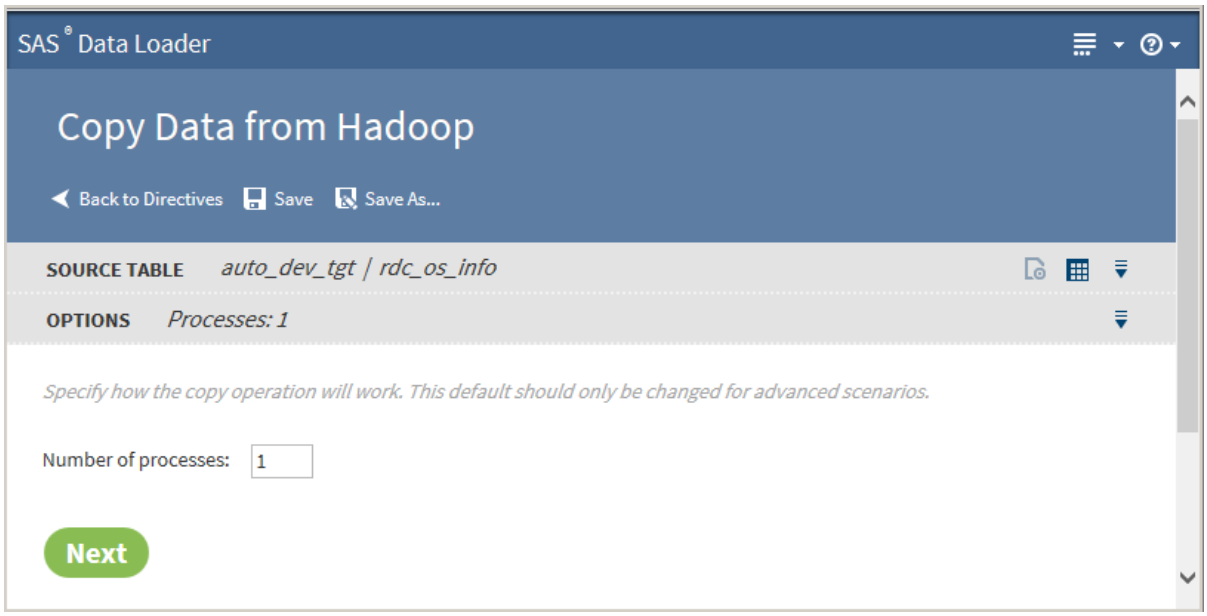
Open
opens the current task.

Table Viewer
enables you to view sample data from a table. Select the table, and then click to display SAS Table Viewer.

Advanced Options
opens a dialog box that enables you to specify the maximum length for SAS columns. Entering a value here overrides the value specified in the **Configuration** options.

Note: If the source table has String data types, the resulting SAS data set could be very large. The length of the target field in the SAS data set is determined by the value of this option.

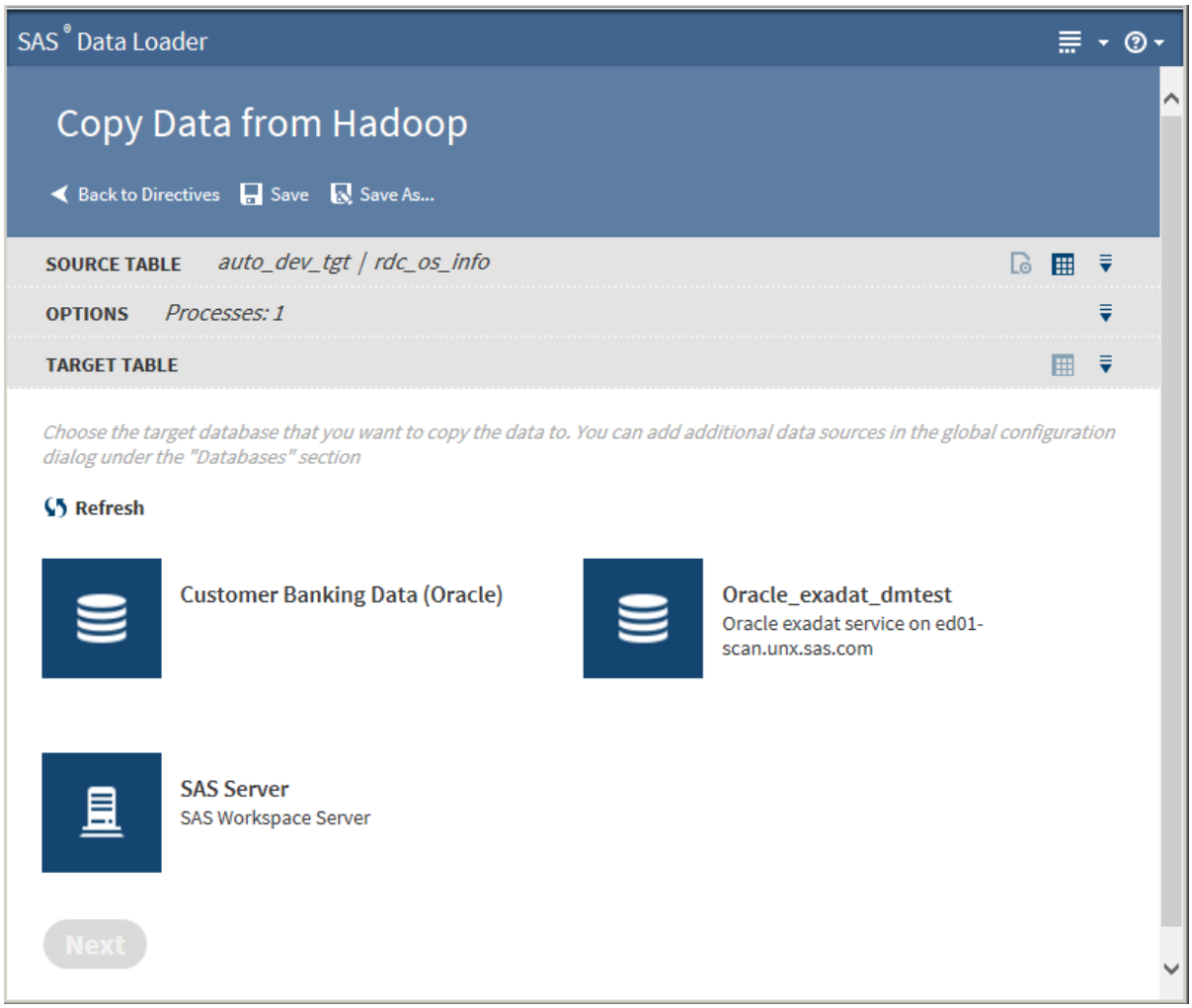
When table selection is complete, click **Next**. The **Options** task is displayed:



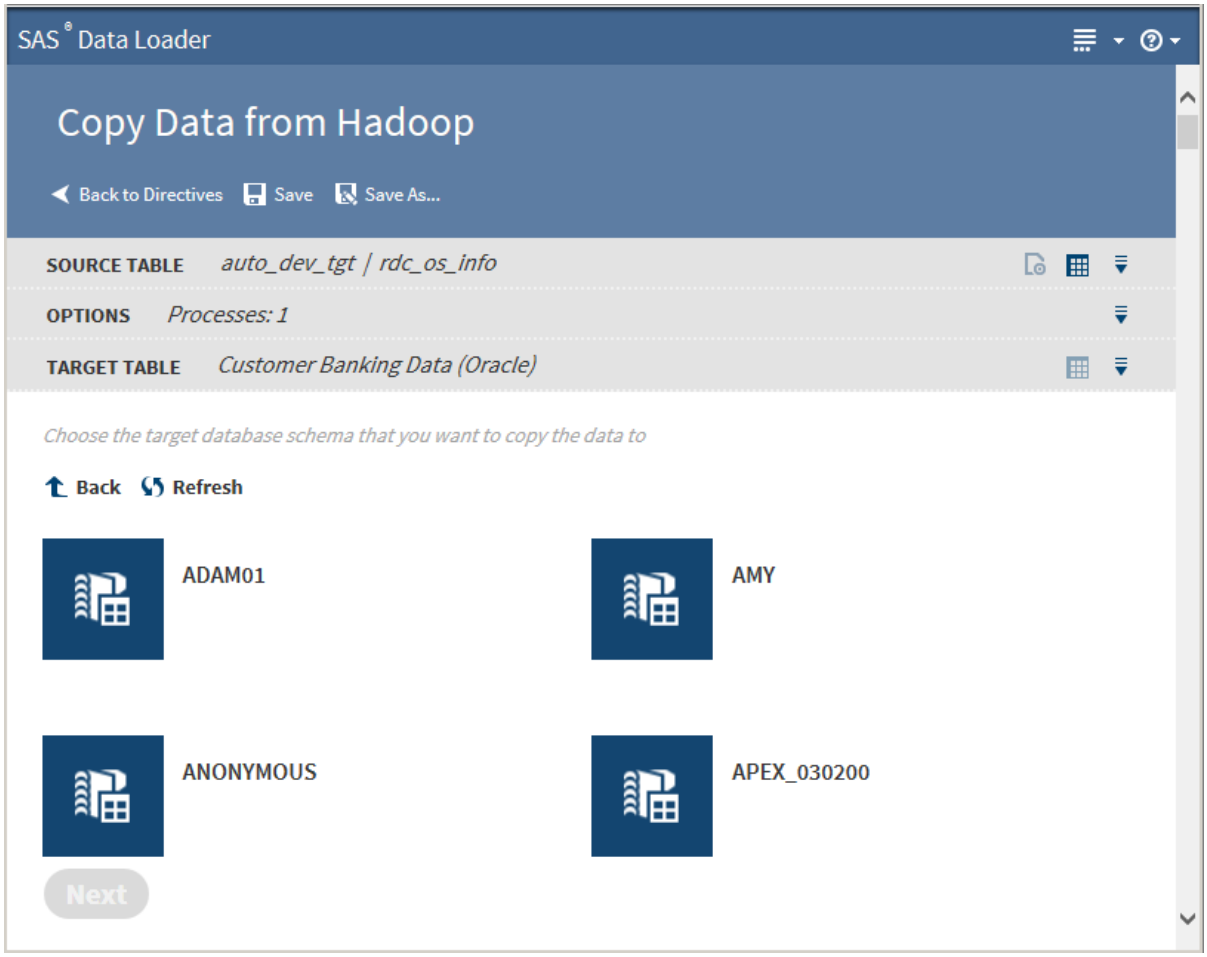
4 The value on the **Options** task should not be changed unless you have advanced knowledge of database operations.

Note: Changing the number of processes to greater than one expands the number of processes and source data connections that are used to import data.

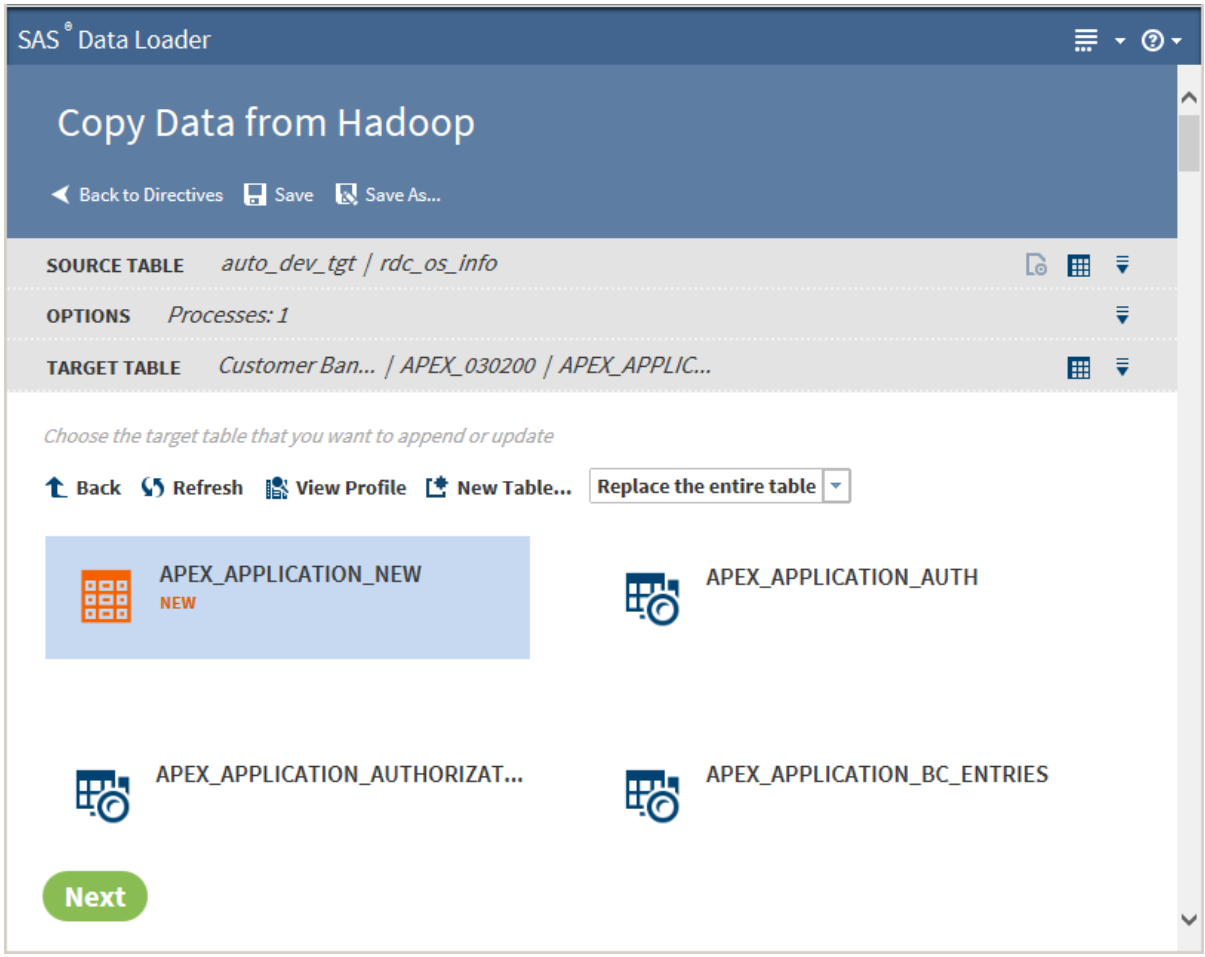
Click **Next**. The **Target Table** task is displayed with target databases:



5 Click a database to display its data sources:



6 Click the target database schema to display its tables:



7 Choose how and where to copy the data. You have three options:

- Select an existing table as the target of the directive. Use the **Insert into the existing table** option to append the data to the selected existing table.
- Select an existing table as the target of the directive. Use the **Replace the entire table** option to update the data in the selected existing table.
- Create a new table to use as the target of the directive. Click the New Table... icon. In the New Table dialog box, enter a name for the new table and click **OK**.

  Note: The Copy Data from Hadoop directive automatically creates the new table for you before it copies the data.

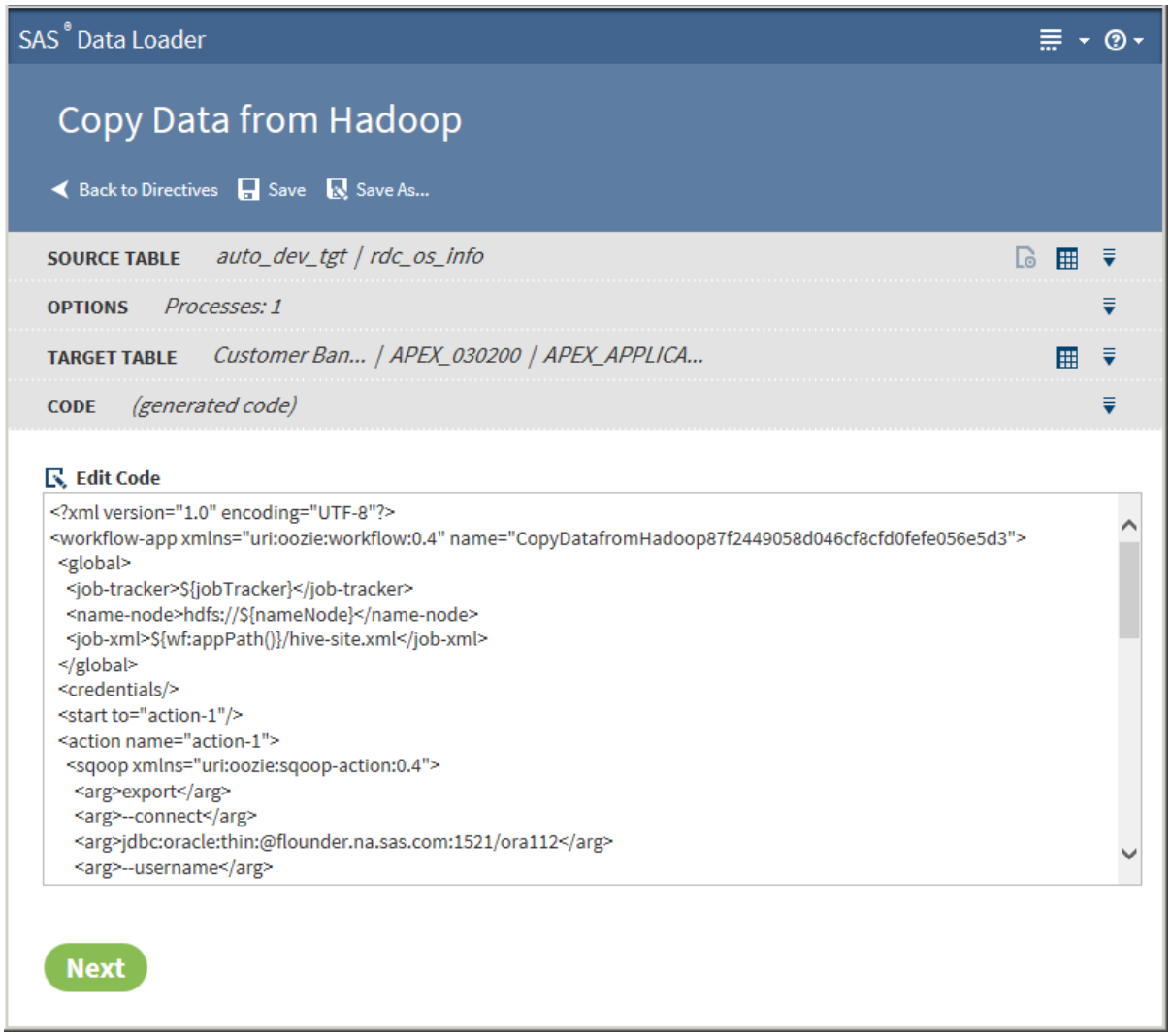
TIP If a profile already exists for a table, PROFILED appears next to the table icon. You can view the existing profile by selecting the table and clicking **View Profile**.

Click ☰ to enable the following actions:

Open
opens the current task.

Table Viewer
enables you to view sample data from a table. Select the table, and then click to display SAS Table Viewer.

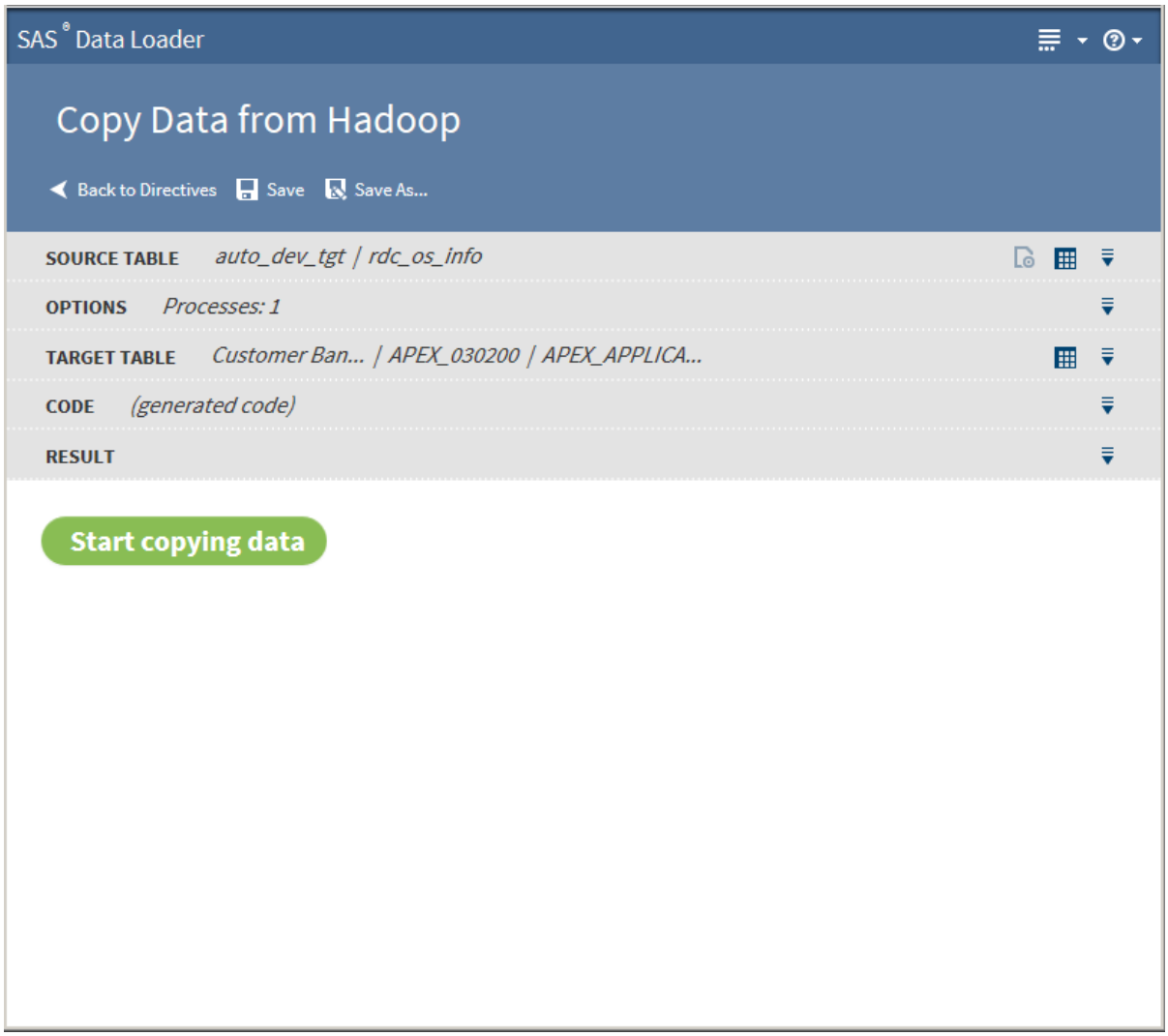
Click **Next**. The **Code** task is displayed:



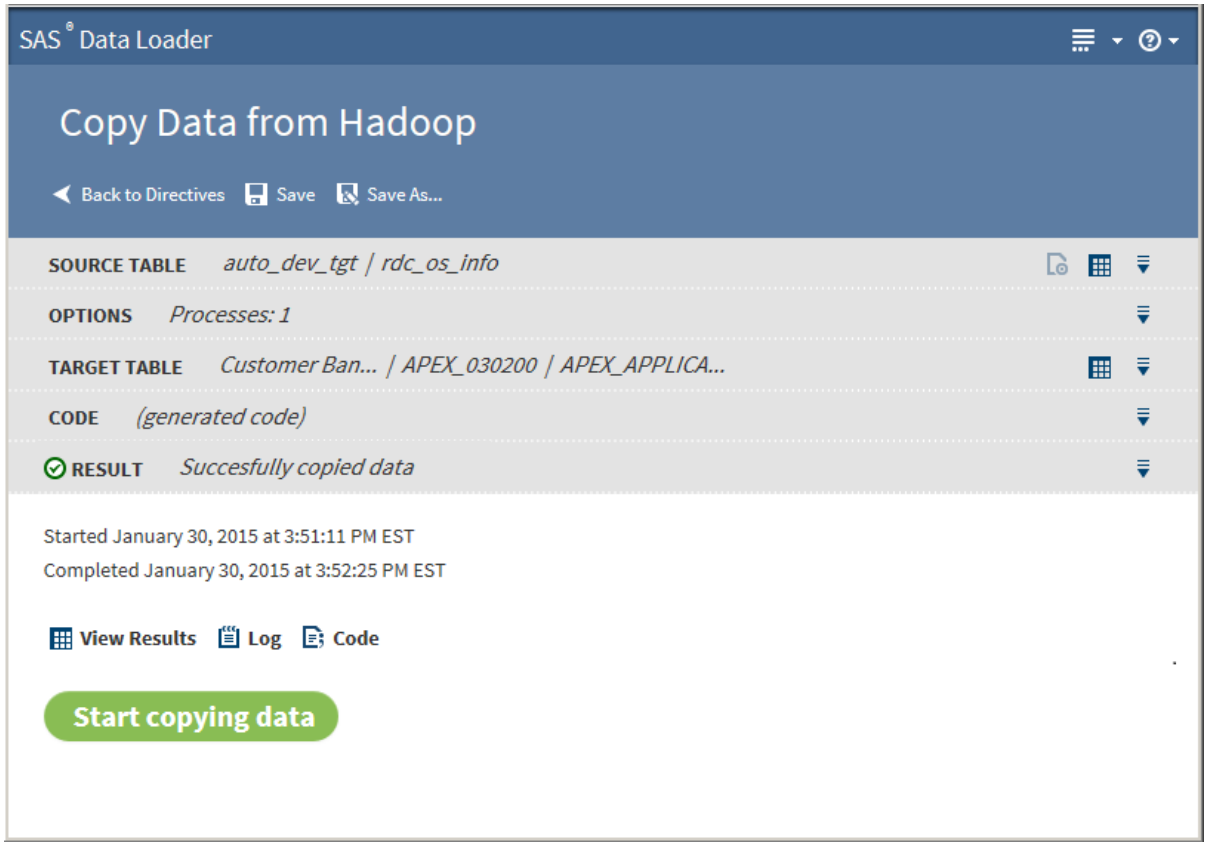
8 Click **Edit Code** to modify the generated code. To cancel your modifications, click **Reset Code**.

CAUTION! Edit code only to implement advanced features. Under normal circumstances, code edits are not needed or required.

9 Click **Next**. The **Result** task is displayed:



10 Click **Start copying data**. The **Result** task displays the results of the copy process:



The following actions are available:

View Results
enables you to view the results of the copy process in the SAS Table Viewer.

Log
displays the SAS log that is generated during the copy process.

Note: If your directive fails, and if the Error Log does not help you resolve the issue, ask your Hadoop administrator to consult the log files for Apache Sqoop.

Code
displays the SAS code that copies the data.

Usage Notes

- The Copy From Hadoop directive does not preserve the case of an HDFS table name when the target database is DB2. In this case, use DB2 commands to create an empty target table with the appropriate table names. In SAS Data Loader, select the empty table as the target of the Copy From Hadoop directive. Choose the Insert method to update the target table.
- By default, the Copy Data from Hadoop directive uses a VARCHAR length of 4000 when copying string data from an HDFS table. For some databases (such as DB2), this could be too long to fit into the default tablespace. If you encounter this error, use DBMS commands to create an empty target table with the appropriate attributes. In SAS Data Loader, select the empty table as

the target of the Copy From Hadoop directive. Choose the Insert method to update the target table.

- If necessary, you can change the maximum length of character columns for source tables for this directive. For more information, see “Change the Maximum Length for SAS Character Columns” on page 195.
- Source tables with a large number of columns can cause Copy From Hadoop jobs to fail. The job runs until the target table reaches the maximum number of columns that are supported in the target database. To resolve the problem, reduce the number of columns that are selected for the target and run the job again.
- If one or more VARCHAR or STRING columns from a source Hadoop table contains more string data than the target database column, the Copy Data from Hadoop request times out. For example, a source Hadoop table might contain a string column named myString and a target Oracle table might contain a varchar(4000) column also named myString. If data in the Hadoop myString column has a length greater than 4000, then the copy request fails.
- When copying a Hadoop table to a database, a column name specified in the array of STRUCT in the Hadoop table is not copied to the database table. This happens because of how STRUCT is mapped to VARCHAR in Sqoop.
- A copy from Hadoop is likely to fail if the name of a source column is also a reserved word in the target database.
- When copying a Hadoop table to Oracle, a mixed-case schema name generates an error.
- When copying a Hadoop table to Oracle, timestamp columns in Hadoop generate errors in Oracle. The Hive timestamp format differs from the Oracle timestamp format. To resolve this issue, change the column type in the Oracle target table from timestamp to varchar2.
- To copy Hadoop tables to Teradata, when the source contains a double-byte character set (DBCS) such as Chinese, follow these steps:

  1 Edit the default connection string to include the option `charset=utf8`, as shown in this example:

  ```
  jdbc:teradata://TeradataHost/Database=TeradataDB,charset=utf8
  ```

  To edit the configuration string, open the Configuration window  , click **Databases**, and click and edit the Teradata connection.

  2 Ensure that the default character type for the Teradata user is UNICODE.

  3 In new Teradata tables, set , VARCHAR CHAR columns to CHARACTER SET UNICODE to accommodate wide characters.

Load Data to LASR

Introduction

Load Data to LASR
Copy data from a source and load it into LASR. Existing data in the target table will be replaced

Use the Load Data to LASR directive to copy Hadoop tables to a single SAS LASR Analytic Server, or to a grid of SAS LASR Analytic Servers. On the SAS LASR Analytic Servers, you can analyze tables using software such as SAS Visual Analytics.

When you load data onto a single SAS LASR Analytic Server, you configure a connection that is optimized for symmetric multi-processing (SMP). When you load data onto a grid of SAS LASR Analytic Servers, you configure a connection that is optimized for massively parallel processing (MPP).

Note: The Load Data to LASR directive is distinct and separate from the Load to LASR capability that is provided by SAS LASR Analytic Server.

Prerequisites

An administrator must set some special options for SAS Data Loader on the SAS LASR Analytic Server. You must specify a connection in the **LASR Analytic Server** panel of the Configuration window. For more information about these tasks, see “LASR Analytic Servers Panel” on page 181.

Example

Follow these steps to create and run the Load Data to LASR directive:

1. In the Directives directives page, click **Load Data to LASR**.
2. In the Source Table task, click the schema that contains the source table that you want to load. Clicking the schema displays the tables in that schema. Click the table that you want to load into the SAS LASR Analytic Server software, and then click **Next**.
3. In the Target Table task, click the SAS LASR Analytic Server that you want to receive the target table. Clicking displays target table configuration fields and controls.
4. As needed, change the name in the **Target table name** field. The field defines the name of the table in the SAS LASR Analytic Server software.
5. Select options as needed to replace any existing table of the same name or to compress the target table in the SAS LASR Analytic Server software.

6 Click the **Locations** link to view or change the default storage options for the target table in the SAS LASR Analytic Server software.

7 In the Locations window, you can change the SAS folder, the library name, and the required tag that accompanies the table name.

8 In the Target Table task, click **Next**.

9 In the Result task, click **Start loading data**. SAS proceeds to generate code for the directive and displays the **Code** icon  . Click the icon to open or save the text of the SAS code that comprises the directive.

10 During the execution of the directive, the **Result** task displays the **Log** icon  . Click the icon to open or save the SAS log file that is generated during the execution of the directive.

11 At the conclusion of the directive, the Result banner receives a status icon that indicates the success or failure of the directive. To view the target table on the SAS LASR Analytic Server, click the **View Results** icon  .

Usage Notes

In MapR distributions of Hadoop, massively parallel processing (MPP) is not supported in the LASR procedure. To load data from MapR Hadoop to a SAS LASR Analytic Server, the server definition must assert the SASIOLA option. The SASIOLA option implements symmetric multiprocessing (SMP.) Server definitions are available in the SAS Data Loader Configuration window, in the **LASR Analytic Servers** panel. For more information about server definitions, see “Add or Update Connections to SAS LASR Analytic Servers” on page 185.

The Load Data to LASR directive moves entire tables. To improve performance, you can filter the rows and manage the columns before you load the table to the SAS LASR Analytic Server. To reduce table size, use the directives “Transform Data” or “Query or Join Data”

The Load Data to LASR directive loads Hive tables to a SAS LASR Analytic Server. It does not load HDFS or NFS data directly. This is because the Load Data to LASR directive performs the load using the SAS embedded process. To support the embedded process, the **LASR server tag** text box in the LASR Server Configuration dialog box is limited to eight characters and must be valid as a SAS libref. These restrictions will not work for HDFS or NFS data because this data requires the server tag to represent the source path in dot-delimited form.

7

Run User-Written Programs

Overview

The directives Run a SAS Program and Run a Hadoop SQL Program enable you to execute existing code in SAS Data Loader. You can also create and execute new programs. SAS Data Loader enables you to receive and retain log files and save these jobs for reuse.

Run a SAS Program

Introduction

The directive Run a SAS Program provides the primary means of submitting user-written SAS code in SAS Data Loader for Hadoop. The code runs as you submit it, without the code generation step that is used in other directives. The code that you submit generates the same log and error information as in other directives. Also, the running code is tracked in the Run Status directive, and you can save and reuse jobs in Saved Directives.

The code execution process begins and ends in the vApp. The Workspace Server inside the vApp runs the code and executes all Base SAS language elements. If your code contains procedures that are enabled for DS2, or if your

code contains native DS2 methods, then that code might be passed into the Hadoop cluster for execution. In your Hadoop cluster, DS2 code is executed by the SAS In-Database Code Accelerator for Hadoop.

Upon completion of DS2 execution on the cluster, the vApp receives notification and continues or concludes execution in the local Workspace Server.

For examples of DS2–enabled SAS code, refer to the code that is generated by directives such as Transform Data.

CAUTION! Data sets in Hadoop are of indeterminate size. Any data that is indiscriminately returned from Hadoop to the vApp can overload the client. To avoid overloading the vApp, your SAS programs need to minimize or eliminate the transfer of data from Hadoop to the vApp. It is generally preferable to define a result set or target table that remains in Hadoop. You can then analyze the data in Hadoop, or load data for further analysis onto a grid of SAS LASR Analytic Servers.

Note that you can generate code in any of the following software, and copy and paste that code into the Code task of the directive Run a SAS Program:

- SAS Data Management Studio
- SAS Enterprise Guide
- SAS Data Integration Studio

Conversely, you can copy the code that is generated in any SAS Data Loader directive and paste into any SAS text editor. One suggested location for pasting SAS Data Loader code is the SAS Code Node in DataFlux Data Management Studio.

To include DS2 syntax in your SAS programs, you can use a number of SAS procedures that support DS2 language elements, as described in the *SAS In-Database Products: User's Guide*. For DS2 syntax information, see the *SAS DS2 Language Reference*.

To run DS2 code directly in Hadoop using the SAS In-Database Code Accelerator, see the "SAS In-Database Code Accelerator for Hadoop" section of the *SAS In-Database Products: User's Guide*.

User-written SAS DS2 code can be submitted in an expression builder in the following directives:

- Delete Rows
- Cleanse Data (Filter Transformation)
- Transform Data (Filter Data task)

Example

Follow these steps to use the directive Run a SAS Program:

1. In the SAS Data Loader directives page, click **Run a SAS Program**.
2. In the **Code** task, type SAS code, or right-click to cut-and-paste existing SAS code.

   TIP The pop-up menu enables you to display line numbers and to navigate to the beginning or the end of the program.

3. When your program is ready to run, click **Next**.

4 In the **Result** task, click **Start SAS program**. As your program runs, you receive start and end date/time information, along with **Log**, **Code**, and possibly **Error Details** icons. Click the icons as needed to resolve errors.

   The final status of the job is displayed in the **Result** taskbar.

5 Click **Save** to save your program for reuse. To edit or run your job in the future, go to the SAS Data Loader directives page and click **Saved Directives**.

Run a Hadoop SQL Program

Introduction

Use the directive Run a Hadoop SQL Program to create jobs that execute SQL programs in Hadoop. The directive enables you to browse available SQL functions, obtain syntax and usage information, and click to add function syntax into the directive's text editor. You can also copy and paste existing SQL programs directly into the text editor.

The user credentials that are specified in the Hadoop Configuration panel of the SAS Data Loader Configuration window are used to submit SQL code to the Hadoop cluster.

The Run a Hadoop SQL Program directive enables you to use either Cloudera Impala SQL functions or HiveQL functions.

Note: Similar support for user-written SQL is also provided in the directives Delete Rows, Query or Join Data, or Sort and De-Duplicate Data.

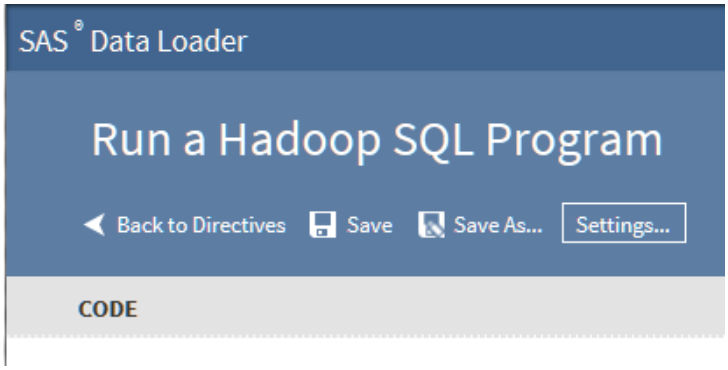
Enable the Impala SQL Environment

Support for the Cloudera Impala SQL environment is enabled in the **Hadoop Configuration** panel of the **Configuration** window. When Impala is enabled, new instances of the following directives use the Cloudera Impala SQL environment by default:

- Run a Hadoop SQL Program
- Sort and De-Duplicate
- Query or Join

The default SQL environment can be overridden using the **Settings** menu. To learn more about SQL environments, see "Enable Support for Impala and Spark".

Note: Changing the default SQL environment does not change the SQL environment for saved directives. Saved directives continue to run with their existing SQL environment unless they are opened, reconfigured, and saved.



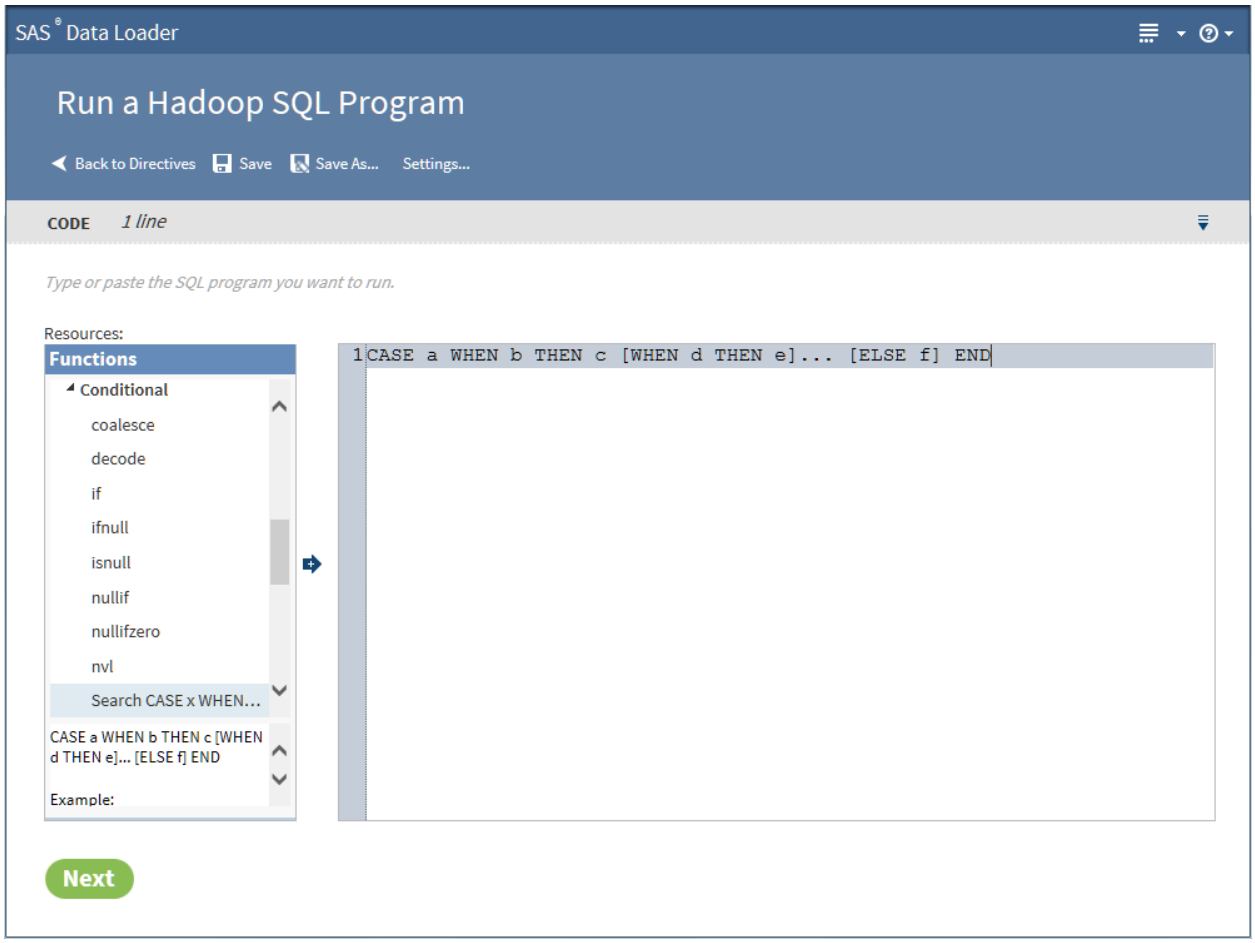
Example

Follow these steps to use the directive Run a Hadoop SQL Program:

1. In the SAS Data Loader directives page, click **Run a Hadoop SQL Program**.
2. In the **Code** task, click the text editor and enter SQL code.
3. To paste SQL code, use the pop-up menu in the text editor.

   Note:

   - Pasted SQL must be supported in the selected SQL environment (Impala SQL or HiveQL.)
   - The SQL program needs to explicitly define data sources and targets.
   - The pop-up menu also enables you to display line numbers and to navigate to the beginning or the end of the program.
4. To add SQL functions to your program, click in **Resources**, expand categories, display syntax help, and add syntax to your program.
5. To move function syntax into your program, click the function and click  .



6 When your program is ready to run, click **Next**.

7 In the **Result** task, click **Start SQL program**. As your program runs, you receive start and end date/time information, along with **Log**, **Code**, and possibly **Error Details** icons. Click the icons as needed to resolve errors.

The final status of the job is displayed in the **Result** taskbar.

8 Click **Save** to save your program for reuse. To edit or run your job in the future, go to the SAS Data Loader directives page and click **Saved Directives**.

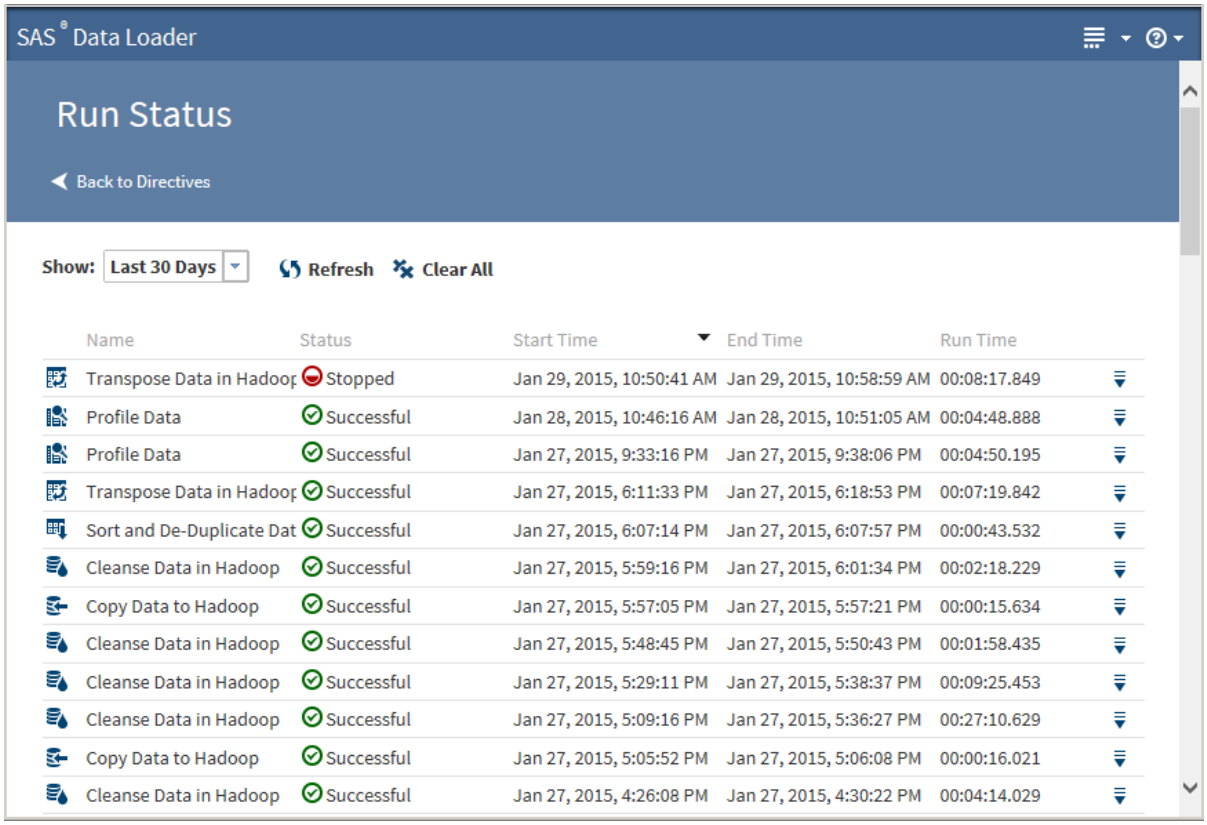
8

Manage Jobs

Overview of Job Management Directives

The job management directives enable you to view the status of current and previous jobs and to modify and execute saved directories. The Run Status directive displays information about the current execution state of jobs. The Saved Directives directive enables you to open, edit, and manage your existing directives.

Here's an example of the Run Status directive:



Run Status

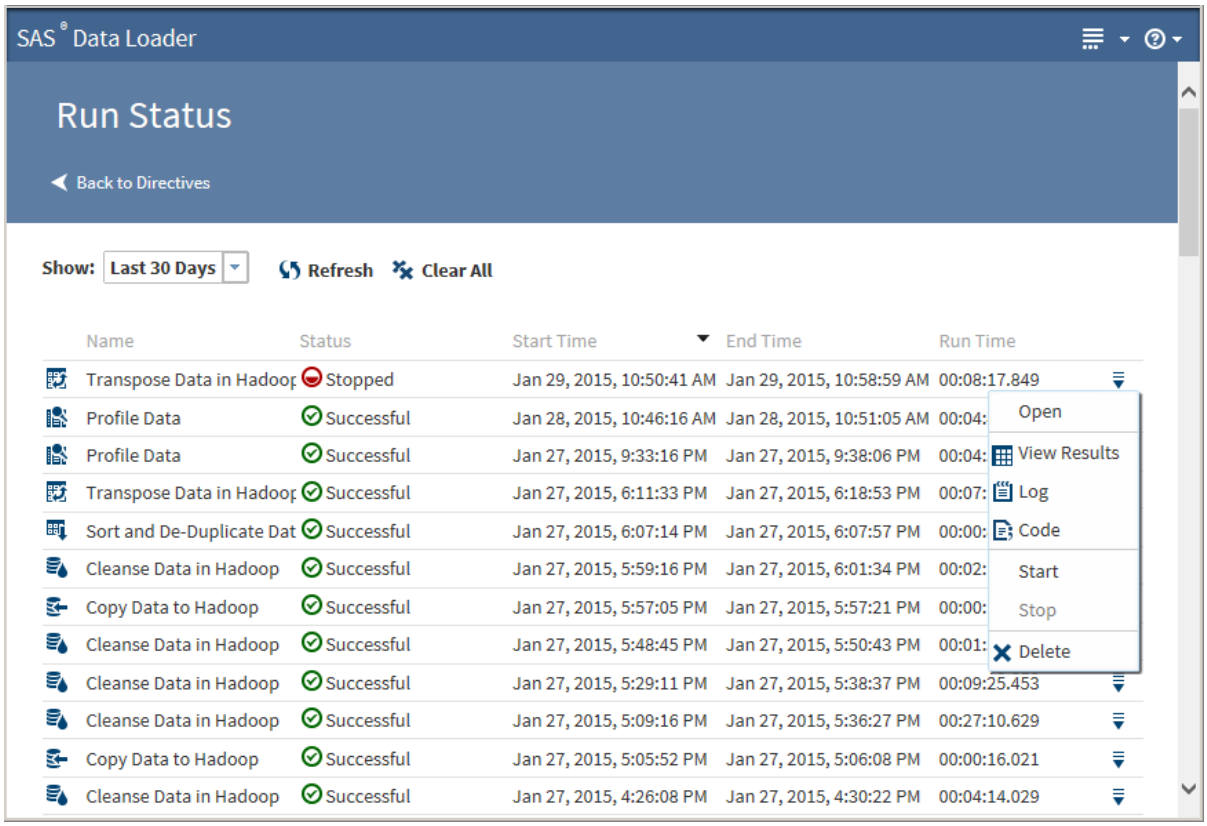
Introduction

Run Status
Show the status of current and previous directive executions

Use the Run Status directive to view job runs. Each run is listed with its current execution status, start time, end time, and run time. The Status column value can be In Progress, Stopped, Failed, or Successful.

Using Run Status

In the SAS Data Loader directives page, click the Run Status directive. The Run Status page is displayed:



By default, the Run Status page displays all of the directive jobs that have run in the past 30 days. The most recent runs appear at the top of the list. You can change the default of 30 days by selecting a new value from the **Show** drop-down list. Job listings are identified by the given name or by the generic name of the directive (for example, Transform Data.) Given names are created when you save a directive.

When you click **Refresh**, you receive updates for all running jobs, including any that were started or completed after you opened the Run Status page.

Clicking **Clear All** clears all of the reports from the Run Status page. Clearing reports permanently removes the reports from the vApp database.

Clicking a job and clicking the **Action** menu  enables the following actions:

Open
opens the directive associated with the job.

View Profile Report
for successful Profile Data jobs, enables you to view the Profile Report unless the report has been deleted from the Saved Profiles directive. See "Saved Profile Reports" on page 115 for more information about the profile report.

View Results
for completed transformations or queries, enables you to view a sample of the target table in the SAS Table Viewer.

Log
displays the SAS log that is generated during the execution of the profile job.

Code
displays the SAS code that is generated during the execution of the profile job.

Start
starts a failed or successful job.

The job runs with the configuration that was specified when the directive was last saved. Any changes in the Configuration window  that were made after the job was saved are not applied. For example, if you save a directive, and then enable the Spark runtime target, the saved directive continues to use its saved runtime target, such as MapReduce.

To apply new configuration settings to a saved directive, open the directive, change the settings, advanced options, and tasks as needed, and save the directive. The next time the directive runs, new code is generated using the latest configuration.

For directives that were created in release 2.3 or earlier, enabling support for Apache Spark or the Cloudera Impala require you to create a new directive, rather than upgrading the existing directive. For more information about Spark and Impala, see “Enable Support for Impala and Spark”.

Stop
stops an in-progress job.

> **TIP** If you select **Stop**, your directive continues to display its In Progress status. In this situation, the directive is stopping, but it has not yet reached a suitable stopping point. Click **Refresh** periodically until the status changes to Stopped or reopen Run Status later to confirm the Stopped status.

A directive typically consists of multiple tasks or steps. When stopping a job, it is often not possible to terminate the task currently in progress. In such cases, the stop is enforced when the current task completes. Any further tasks that are part of the directive are skipped.

Delete
clears a single report from the Run Status page. Clearing a report permanently removes the report from the vApp database.

About Unsaved Jobs

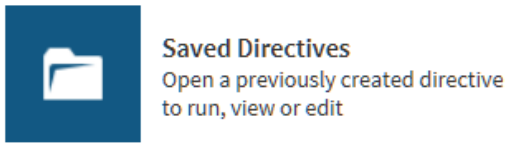
If you run a directive without saving it, the directive is displayed in Run Status like any other directive. When processing stops on the unsaved directive, you can select **Open** from its Action menu. You can then edit and save the unsaved directive.

About Incomplete Jobs

An incomplete job is one that you have stopped using the **Action** menu or one whose status is Failed. Depending on the type of the job and the point where execution ceased, log and code results might or might not be available.

Saved Directives

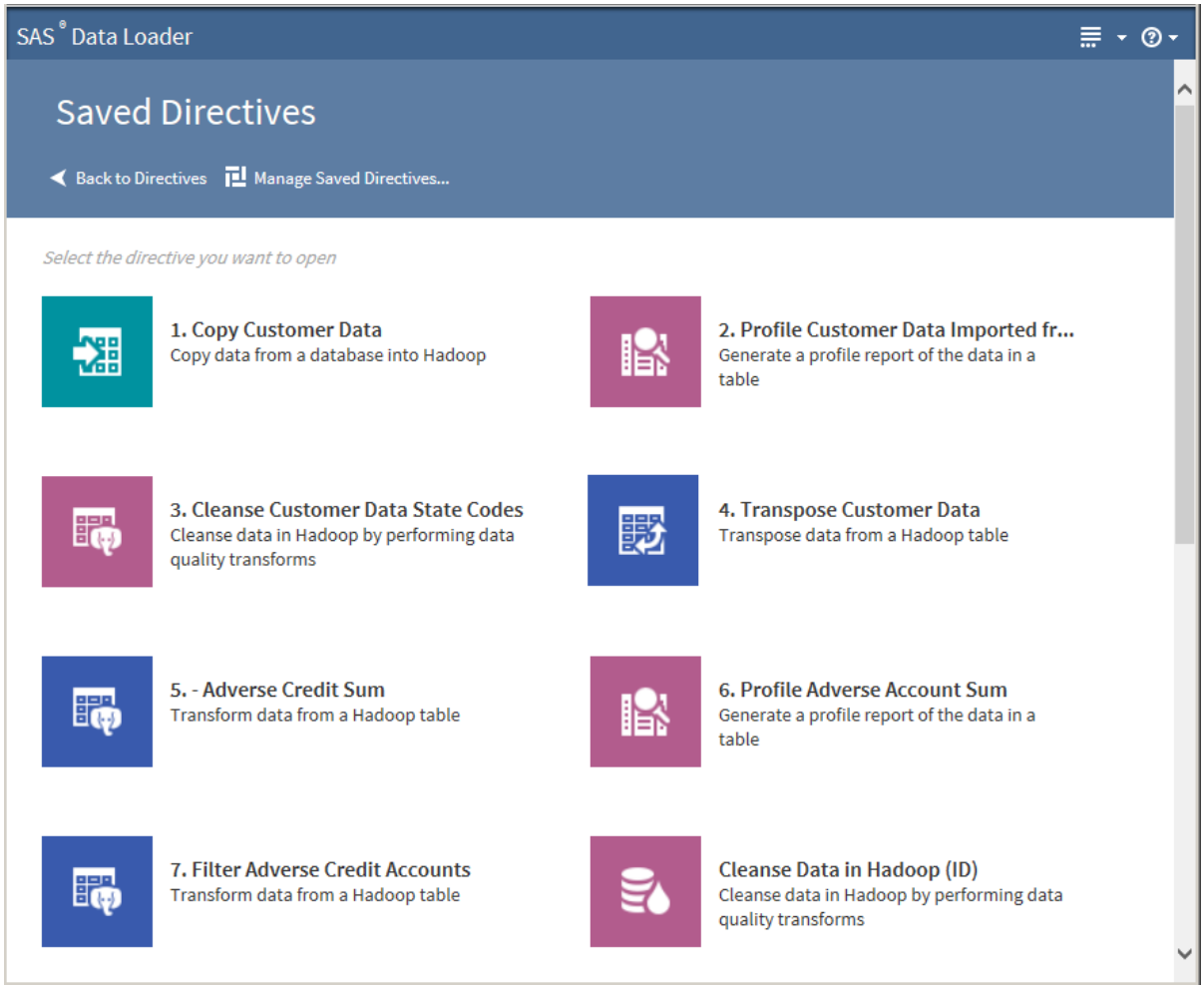
Introduction



Use Saved Directives to open, edit, and execute your saved directives. From the Saved Directives page, opening Manage Saved Directories enables you to open, duplicate, delete, refresh, or rename the selected directive.

Opening Saved Directives

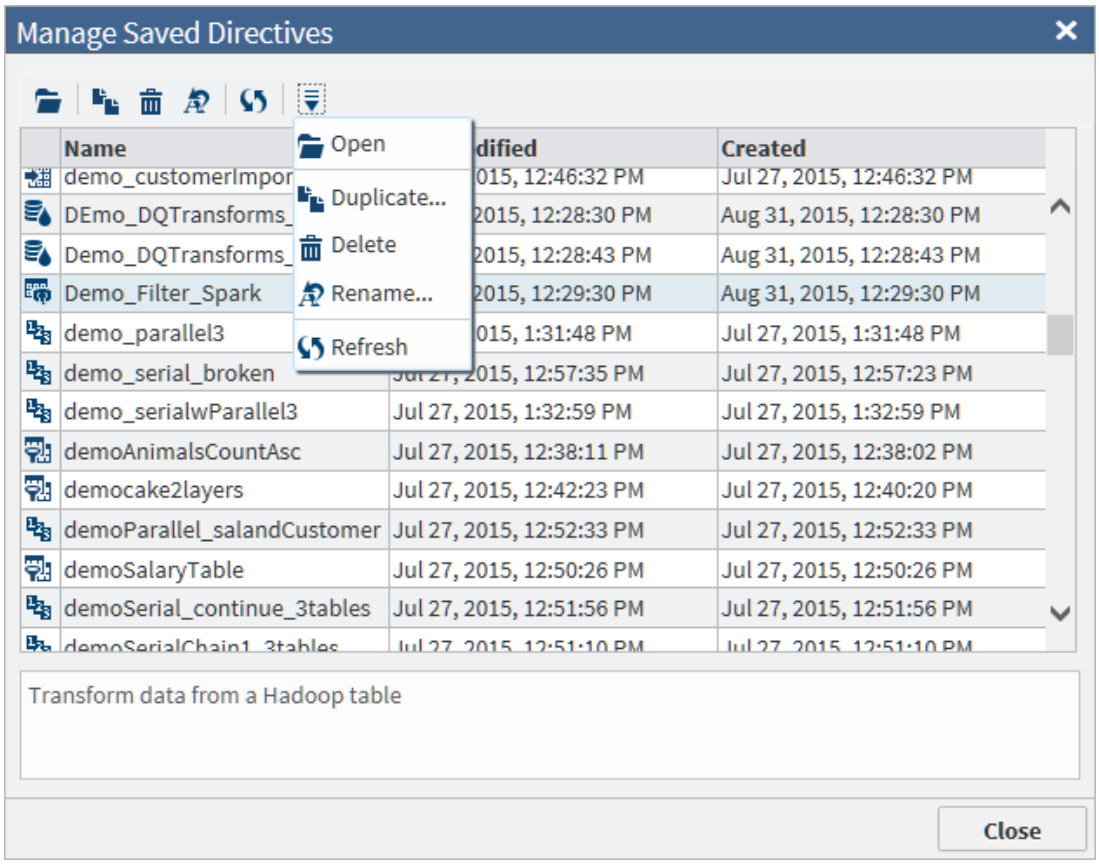
In the SAS Data Loader directives page, click the Saved Directives directive. A Saved Directives page similar to the following is displayed:



In the Saved Directives page, click a saved directive. The directive opens and can be edited or executed.

Managing Saved Directives

In the Saved Directives page, click **Managed Saved Directives**  . The Managed Saved Directives dialog box appears:



Clicking the **Action** menu  enables the following actions:

Open
opens the selected directive.

Duplicate
duplicates the selected directive by opening a dialog box that enables you to assign a new name to the duplicated directive.

Delete
deletes the selected directive.

Rename
renames the selected directive.

Refresh
refreshes the selected directive, or, if no directive is selected, refreshes all of the saved directives in the list. Any duplicate, rename, or delete actions that you have taken are then reflected in the saved directives list.

Chain Directives

Introduction

Use Chain Directives to create jobs that execute a list of jobs, either in series or in parallel.

You can nest Chain Directives jobs. One Chain Directives job can include and execute other Chain Directives jobs. A serial Chain Directives job can contain nested parallel jobs. Parallel jobs can contain nested serial jobs.

A Chain Directives job can contain multiple stances of a component job.

When you run your Chain Directives job, you can view the results of the component jobs in the chain, as the results become available.

You can run Chain Directive jobs by opening them in the Chain Directives interface, or you can run them from the "Run Status"page or the "Saved Directives"page.

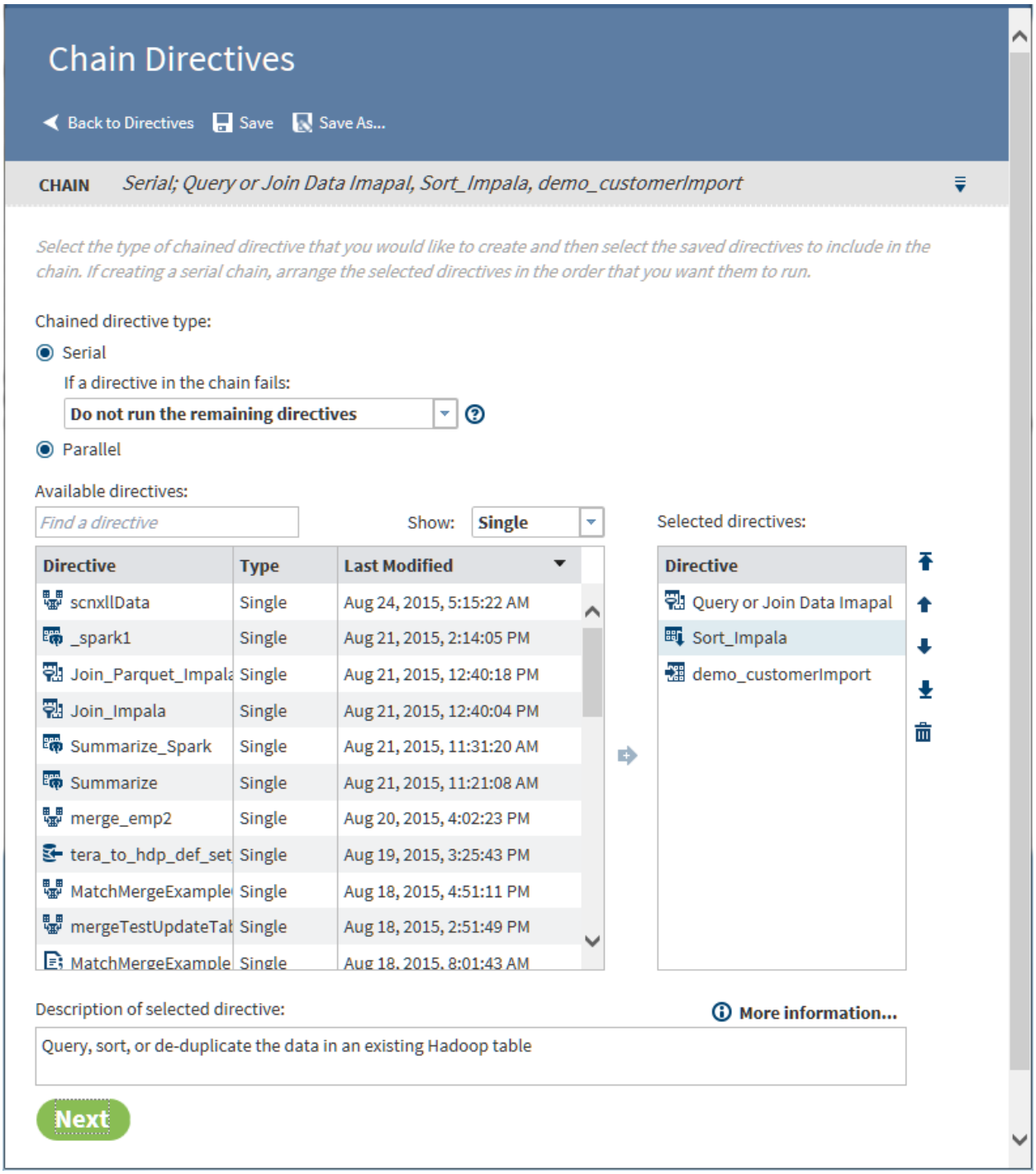
Example

To create and run your own Chain Directives job, open the directive in SAS Data Loader for Hadoop, and then follow through this example.

Specify Serial or Parallel

In the **Chain** task, under **Chain directive type**, click **Serial** or **Parallel**. Choose **Serial** when one job in the chain depends on the results of a preceding job. Choose **Parallel** to run a group of independent jobs simultaneously.

You can change your selection of **Serial** or **Parallel** at any point in the process of developing your chain directive.



Select Directives

To select directives for your **Chain Directives** job, first locate a directive in **Available directives**. You can search, filter, and rearrange **Available directives** as follows:

- To locate a directive by name, click and enter text in **Find a directive**.
- To filter the list in order to display only single, serial, or parallel directives, click the **Show** field and select a value.
- To rearrange the list from last-modified to first-modified, click **Last modified**.

To add a directive to your job, click the row in **Available directives** and then click the right arrow . Use Ctrl+click to select multiple directives.

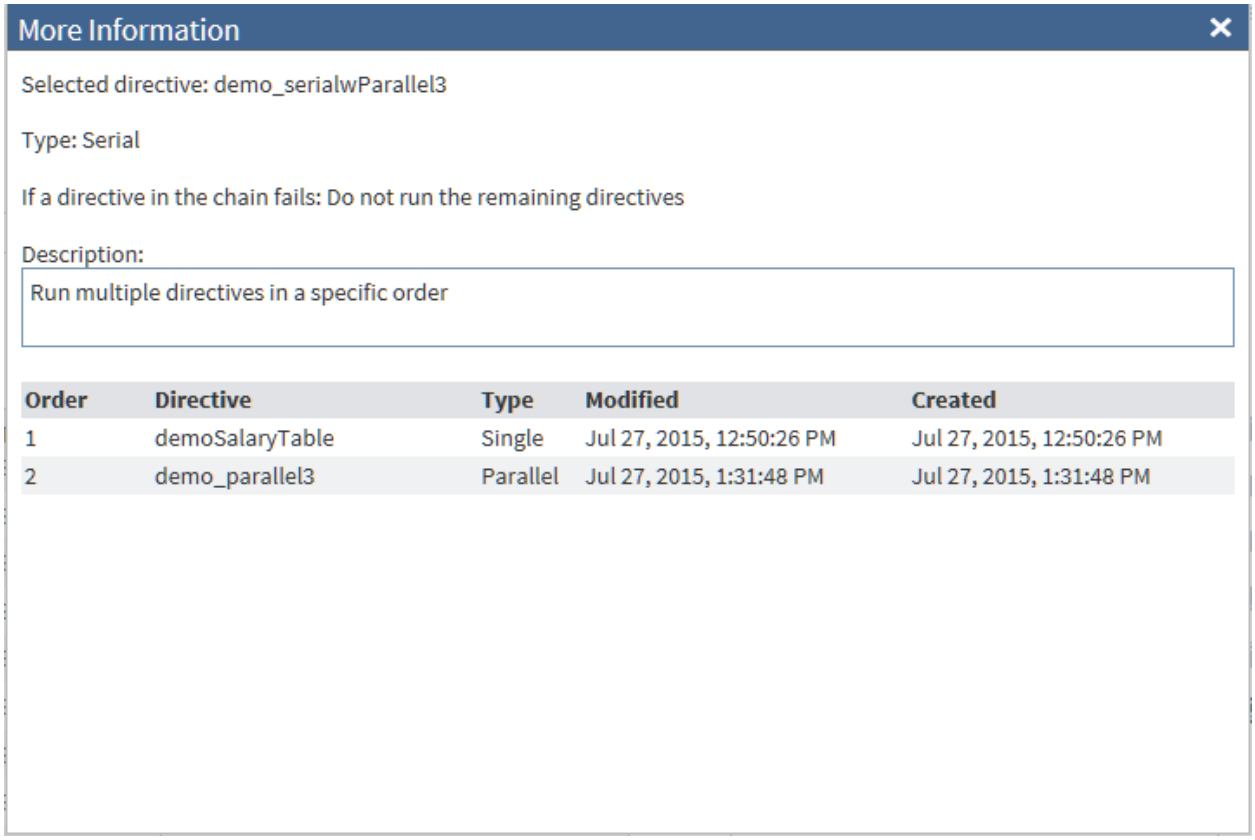
To remove a directive from your job, select the directive in **Selected directives** and click . Clicking the trash can icon removes the directive from **Selected directives** list. The directive is not deleted from the file system. Nor is it deleted from Saved Directives or Run Status.

In serial chain directives, to rearrange the order of the selected jobs, click the jobs in the list and click the vertical arrows. The top job runs first and the bottom job runs last.

Note: For parallel Chain Directives jobs, ensure that the component directives access source and target tables without conflict. Simultaneous Read access is supported. Simultaneous Write access of individual target tables is not supported. Also, one directive cannot write a target table while another job is reading that same table as a data source. To ensure accurate reads and writes in a parallel chain directive, open the directives as needed to ensure that target tables are unique. Also ensure that a target table in one directive is not used as a source in another directive.

If you choose **Serial**, the default behavior stops the execution of the chain directive if one of the directives in the chain fails to reach completion. To continue the execution of directives regardless of failures, click **If a directive in the chain fails**, and then select **Continue to run the remaining directives**.

To learn about available or selected directives, click the directive and read the description in **Description of selected directive**. To learn more, click **More information**. At the top of the More Information window, the **Type** row indicates that the selected directive is either **Single** (not a chain directive), **Serial**, or **Parallel**.



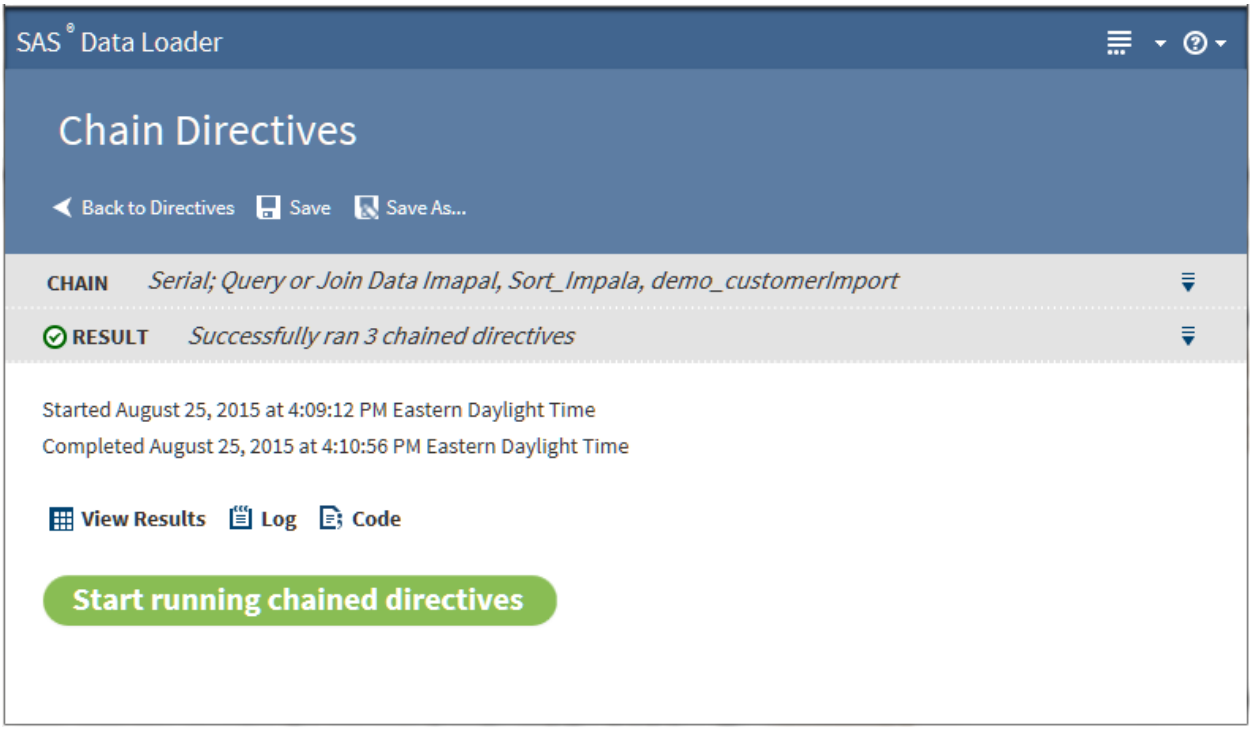
The table at the bottom of the More Information window describes the contents of the selected directive. If the selected directive is a chain directive, the window lists all of the component directives. The **Order** column displays **None** for a single directive. The value is **Concurrent** for the components of a parallel chain directive. For a serial chain directive, the value specifies the execution order of the component directives.

When your directives have been selected and ordered, click **Next** to display the **Result** task.

Result

In the **Result** task, click **Start running chained directives**. As the job runs, you can examine the generated code and the SAS log file.

When the job is complete, click **View Results** and click a directive to view the results that were generated by that directive. The resulting target table is displayed in the Table Viewer, in a new browser tab.



Click **Save** to make your directive available in Saved Directives.

9

Maintaining SAS Data Loader

Back Up Directives

You can back up your saved directives for later use, in case you need to restore the vApp for SAS Data Loader.

To perform a backup, click the More icon  on the SAS Data Loader page, and then select **Back Up Directives**.

Note:

- To perform a backup successfully, the backup location must be defined in the vApp settings before SAS Data Loader is started. For more information about setting up the backup location and restoring data, see *SAS Data Loader for Hadoop: vApp Deployment Guide*.
- SAS Data Loader supports only one backup. If you perform a backup, any previous backup data in the backup location is overwritten with the new data.

Set Global Options

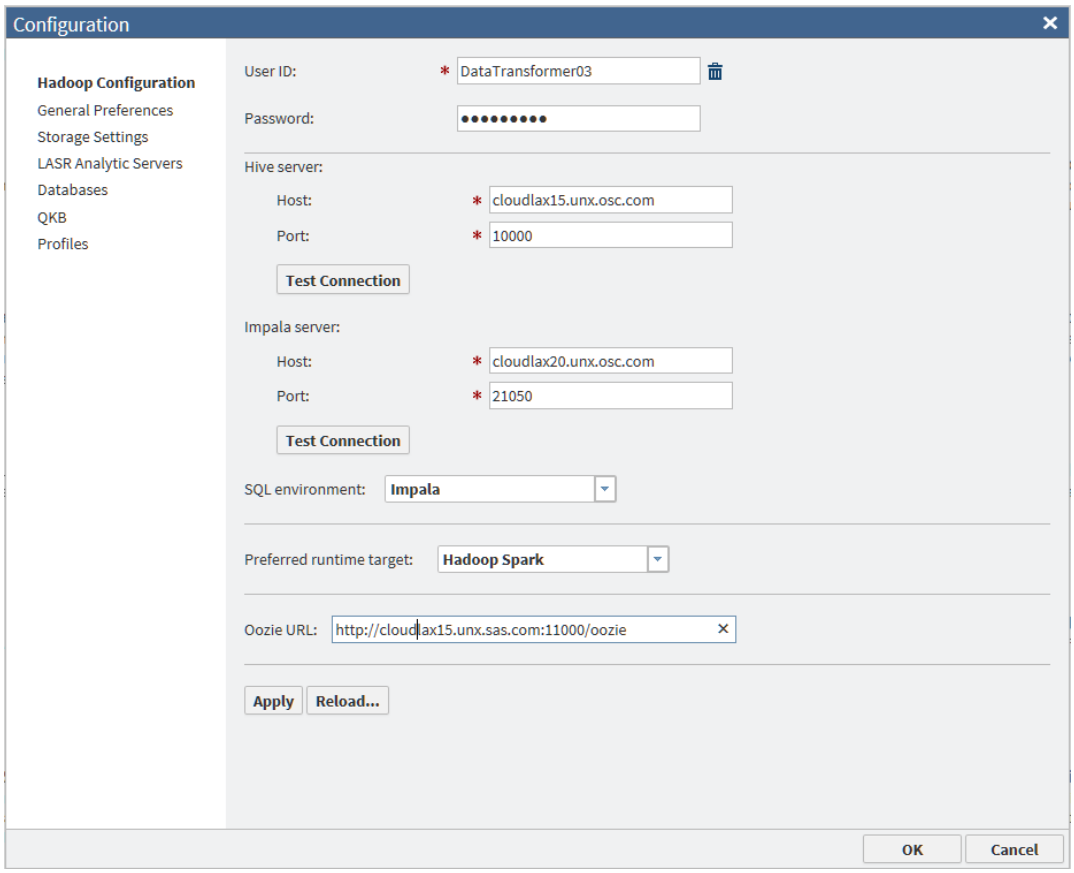
Overview of the Configuration Window

You can use the Configuration window to specify server connections, data sources, global options, and other settings for SAS Data Loader. To display this window, click the More icon  in the top right corner of SAS Data Loader. Then select **Configuration**. See the following topics for details about the options in each panel of the window.

Hadoop Configuration Panel

Use the **Hadoop Configuration** panel of the Configuration window to specify credentials, Hive and Impala server connections, and preferences for the SQL environment and the run-time target.

A **Reload** button enables you to load a predetermined Hadoop configuration from a configuration file.



The values for **Hive server**, **Impala server**, and **Oozie URL** are often populated when SAS Data Loader is first initialized. Review these settings and contact your Hadoop administrator as needed.

Specify the appropriate **User ID**. If you are using LDAP authentication, enter a **Password**.

To reconfigure SAS Data Loader for a different Hadoop cluster, you must copy a new set of configuration files and JAR files into the shared folder of the vApp. For more information about configuring a new version of Hadoop, see *SAS Data Loader for Hadoop: vApp Deployment Guide*.

The fields and controls in the **Hadoop Configuration** panel are defined as follows:

User ID

The name of the user account that is used to connect to the Hadoop cluster. If this field can be edited, specify the name that is provided by your Hadoop administrator.

CAUTION! Enter a user ID only when using LDAP authentication. Entering a user ID in any other environment disables the use of the Cloudera Impala SQL environment. If you are not using LDAP authentication, and a **User ID** value is displayed, click the trash can icon to remove that value.

When your cluster uses a MapR distribution of Hadoop without Kerberos authentication, the **User ID** field is populated from a configuration file when you start the vApp. To change the **User ID** field, first enter the new value in the file ***vApp-home**shared-folder*\hadoop\conf\mapr-user.json**. Next, restart the vApp to read the new value. Finally, open the **Hadoop Configuration** panel and enter the new user ID.

Note: If your site uses a MapR cluster, when the user ID is changed, review the settings in the Configuration window. It is especially important to ensure that the **Maximum length for SAS columns** setting on the **General Preferences** panel is not blank. For more information, see "Replace User Preferences after User ID Changes in MapR or Kerberos Environments" on page 198.

Password
The password for the user account that is used to connect to the Hadoop cluster. If your system administrator indicates that the cluster uses LDAP authentication, a password is required. Enter the password that is provided by the administrator.

CAUTION! Enter a password only when using LDAP authentication. Entering a password in any other environment disables the use of the Cloudera Impala SQL environment.

The **Password** is not editable if Kerberos security has been specified in the vApp.

Host (Hive server)
The fully qualified host name for the Hive server on your Hadoop cluster. A continuously operational Hive server connection is required by SAS Data Loader for Hadoop. This value is always required.

Port (Hive server)
The port number on the Hive server that receives client connection requests. This value is always required.

Test Connection (Hive server)
Click this button to validate your **Host** and **Port** values, and to verify that the Hive server is operational.

Host (Impala server)
The fully qualified host name for the Cloudera Impala server on your Hadoop cluster. This value is required when the value of **SQL environment** is **Impala**. This value is optional when the value of **SQL environment** is **Hive**.

To increase performance, the Cloudera Impala server is used by certain SAS Data Loader directives instead of Hive. To learn more, see "Enable Support for Impala and Spark" on page 12.

Port (Impala server)
The number of the port on the Cloudera Impala server that receives client connection requests. This value is required when the value of **SQL environment** is **Impala**. This value is optional when the value of **SQL environment** is **Hive**.

Test Connection (Impala server)
Click this button to validate your **Host** and **Port** values, and to verify that the Impala server is operational.

SQL environment
Choose the **Impala** value to specify Cloudera Impala as the default environment for new directives, and to enable job execution in that environment. This value applies only to the set of directives that support Impala, as listed in "Enable Support for Impala and Spark" on page 12.

Directives that do not support Impala continue to run in the HiveQL environment as usual.

Individual instances of the supporting directives can be configured to override the default value.

Specify the **Hive** value in the **SQL environment** field to establish Hive as the default SQL environment for new directives.

Note: Changing this value does not change the SQL environment of saved directives.

Preferred runtime target

Select the value **Hadoop Spark** to enable new instances of the supporting directives to run with Apache Spark by default. Apache Spark must be installed and fully configured on the Hadoop cluster. If Apache Spark was detected on the Hadoop cluster during the installation of the SAS In-Database Technologies for Hadoop, then the **Hadoop Spark** value will be set by default.

To learn more about Apache Spark, including the directives than support it, see "Enable Support for Impala and Spark" on page 12.

Select the value **MapReduce** to enable new directives to run with the MapReduce run-time target by default.

Individual instances of the supporting directives can be configured to override this default value.

Oozie URL

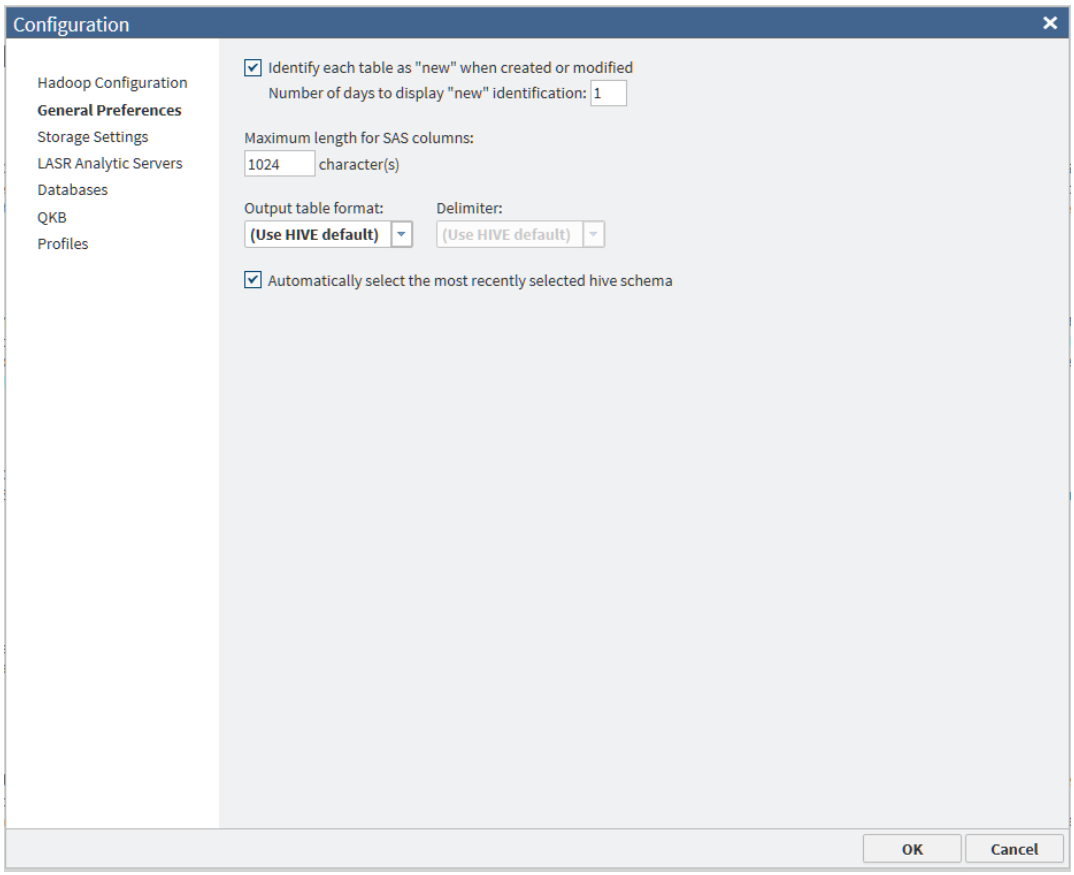
Specify the HTTP address of the Oozie web console, which is an interface to the Oozie server.

Oozie is a workflow scheduler in Hadoop that manages the execution of jobs. SAS Data Loader uses Oozie to copy data to and from databases such as Oracle and Teradata, and to execute directives in the Spark run-time environment.

- URL format: http://*host_name*:*port_number*/oozie/
- URL example (using default port number): http://my.example.com:11000/oozie/

General Preferences Panel

Use the **General Preferences** panel of the Configuration window to specify various global options for SAS Data Loader.



You can change the following default options:

Identify each table as "new"
specifies the number of days in which tables are identified as “new” in SAS Data Loader. The default value is 1 day.

Maximum length for SAS columns
specifies the default maximum length of string columns of types such as VAR and VARCHAR in certain directives. The default value of 1024 characters should perform well in most cases. Strings that exceed the maximum length are truncated when the source data is read into SAS. For more information, see “Change the Maximum Length for SAS Character Columns” on page 195.

Output table format and **Delimiter**
specifies the default file format and delimiter for target tables. Use the **Output table format** drop-down list to select one of five output table formats: **Use HIVE default**, **Text**, **Parquet**, **ORC**, or **Sequence**.

The **Delimiter** field is enabled when you select **Text** as the output table format. Select from the drop-down list the character that is applied by default to delimit the rows in target tables. Available selections are **Use HIVE default**, **Comma**, **Tab**, **Space**, or **Other**. If you select **Other** you are required to enter a delimiter value. The value can consist of a single character or 3-digit octal value. Valid values in octal values range from 0 to 177, which is 0 to 127 in decimal. The octal value indicates the ASCII character number of the delimiter.

For more information, see “Change the File Format of Hadoop Target Tables” on page 193.

Note: If your cluster runs a MapR distribution of Hadoop, or if Apache Spark is selected as the run-time environment, then the Parquet output table format is not supported.

Automatically select the most recently selected hive schema
If you frequently work with the same data source across multiple directives, you can have SAS Data Loader display the most recently used schema in the Source Table and Target Table tasks.

Note: For more information, see "Viewing Data Sources and Tables" on page 20.

Storage Settings Panel

Use the **Storage Settings** panel in the Configuration window to specify non-default storage locations for schema temporary files, Hive, and HDFS.

The fields and controls in the **Storage Settings** panel are defined as follows:

Schema for temporary file storage
To specify a non-default schema for temporary file storage, click **Specify a different schema** and enter the name of an existing schema in Hive.

Hive storage location
To specify a non-default Hive storage location, click **Specify alternate storage location**. Then click the browse button ..., or enter an HDFS path that can be read and written by the user specified in the **Hadoop Configuration** panel.

The browse button displays the Select Directory window, which lists accessible directories only. For more information, see "Override the Default Hive Storage Location for Target Tables" on page 196.

SAS HDFS temporary storage location
To specify a non-default HDFS temporary storage location for SAS files, click **Specify alternate storage location**. Next, click the ..., or enter an HDFS path that can be read and written by the user specified in the **Hadoop Configuration** panel.

The browse button displays the Select Directory window, which lists accessible directories only.

A non-default temporary storage location might be required if directives cannot write to the default directory. This can occur if the sticky bit is set on the default directory, which typically is `/tmp`. When a directory's sticky bit is set, only the directory's owner, a file's owner, or the root user can rename or delete the directory or files in the directory. This is a security measure to avoid deletion of critical folders and their contents, even when other users have full permissions. Contact your Hadoop Administrator to receive an alternate directory that meets your needs.

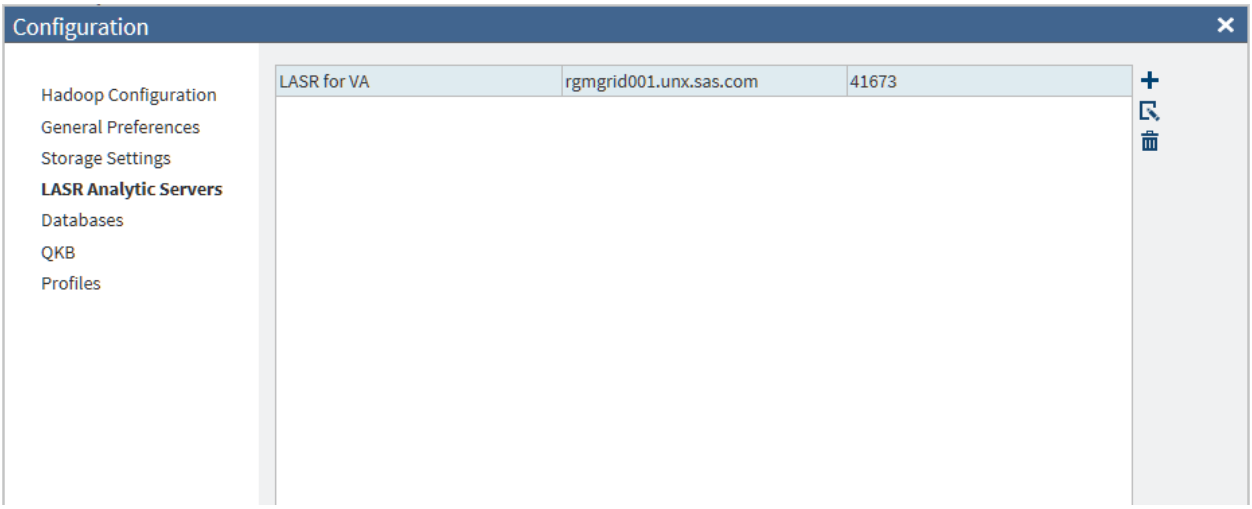
LASR Analytic Servers Panel

Overview

Use the **LASR Analytic Servers** panel to configure connections to SAS LASR Analytic Servers. The server connections can identify independent SAS LASR

Analytic Servers and grids of SAS LASR Analytic Servers. The connections are required to use the directive Load Data to LASR.

To load data onto a grid of SAS LASR Analytic Servers, the connection must be optimized for massively parallel processing (MPP). To load data onto an independent SAS LASR Analytic Server, the connection must be optimized for symmetric multi-processing (SMP). The MPP or SMP configuration options are defined as part of the configuration process.



To add or update a connection to a SAS LASR Analytic Server or grid of servers, ensure that your site meets the prerequisites in the following sections. When the prerequisites are met, see “Add or Update Connections to SAS LASR Analytic Servers” on page 185.

General Prerequisites for SAS LASR Analytic Server

The prerequisites in this section apply to all instances of SAS LASR Analytic Server. Ask your SAS LASR Analytic Server administrator to verify that the following prerequisites have been met:

- SAS LASR Analytic Server must be release 2.5 or later. The server must be fully operational and configured to start automatically.
- SAS Visual Analytics 6.4 or later must be installed and configured on the SAS LASR Analytic Server.
- SAS LASR Analytic Server must be registered on a SAS Metadata Server.
- SAS LASR Analytic Server must have memory and disk allocations that are large enough to accept Hadoop tables. Jobs created with the Load Data to LASR directive cannot ensure that sufficient storage is available in SAS LASR Analytic Server.
- The GatewayPorts option on the SAS LASR Analytic Server must be enabled for the **User ID** that is specified in the LASR Server Configuration window.

If the GatewayPorts option is not enabled, the Load Data to LASR directive will fail. You will get an error about failing to load analytical extension for the distributed computing environment. To set the GatewayPorts option for a specific user, ask an administrator to perform the following steps on the SAS LASR Analytic Server or to the head node on the SAS LASR Analytic Server grid:

1 Log on as root or as a user with sudo.

2 Edit the file `/etc/ssh/sshd_config`. For LASR_USER_ID (the **User ID** that is specified in the LASR Server Configuration window), make the following edits.

3 To set the GatewayPorts option for a specific user, add this line to the end of the file: `Match User LASR_USER_ID`

4 Add the line to the end of the file: `"GatewayPorts clientspecified"`

5 Restart sshd. For example, in many Linux environments, you would issue a command similar to this: `service sshd restart`

Additional Prerequisites for Kerberos Authentication

Display the Hadoop Configuration panel on page 177 of the Configuration window. If the **User ID** field is not editable, the Hadoop login for SAS Data Loader has been configured for Kerberos authentication. The following additional prerequisites apply.

- The user ID used to log on to the Hadoop cluster and the user ID used to log on to SAS LASR Analytic Server must be identical. Take note of the **User ID** that is specified in the **Hadoop Configuration** panel. Ask the SAS LASR Analytic Server administrator to create an account for that user ID on the SAS LASR Analytic Server.
- SAS Data Loader, the Hadoop cluster, and the SAS LASR Analytic Server must share a single Kerberos realm. The Kerberos realm for SAS Data Loader and the Hadoop cluster is specified in the SAS Data Loader: Information Center Settings window in the vApp. Ask the SAS LASR Analytic Server administrator to verify that the user ID on the SAS LASR Analytic Server is in the same Kerberos realm.
- When SAS Data Loader is configured, a Kerberos user ID and realm are entered into the SAS Data Loader: Information Center Settings window in the vApp. When this information is saved, a public key for that user is placed in the shared folder for SAS Data Loader. Ask the SAS LASR Analytic Server administrator to copy this public key to the SAS LASR Analytic Server or to the head node on the SAS LASR Analytic Server grid. The public key must be appended to the authorized keys file in the .ssh directory of that user.
- Review the fields in the LASR Server panel on page 186 of the Configuration window. Ask the SAS LASR Analytic Server administrator to provide the information that is required to specify a connection in this window.

After these prerequisites have been met, you can add a connection to a SAS LASR Analytic Server. See “Add or Update Connections to SAS LASR Analytic Servers” on page 185.

Additional Prerequisites When Kerberos Authentication Is Not Used

Display the Hadoop Configuration panel on page 177 of the Configuration window. If the **User ID** field is editable, the Hadoop login for SAS Data Loader has been configured for no authentication or for an authentication method other than Kerberos. The following additional prerequisites apply.

- The user ID used to log on to the Hadoop cluster and the user ID used to log on to SAS LASR Analytic Server must be identical. Take note of the **User ID** that is specified in the **Hadoop Configuration** panel. Ask the SAS LASR

Analytic Server administrator to create an account for that user ID on the SAS LASR Analytic Server.

- The user account above must be configured with Secure Shell (SSH) keys on the SAS LASR Analytic Server.
- All banners must be disabled for the SSH login for the **User ID** that is specified in the LASR Server Configuration window.

Configure Secure Shell (SSH) keys. To configure Secure Shell (SSH) keys on the SAS LASR Analytic Server, ask the SAS LASR Analytic Server administrator to perform these steps:

1. The administrator generates a public key and a private key for the SAS Data Loader user account and installs those keys in SAS LASR Analytic Server, as described in the *SAS LASR Analytic Server: Reference Guide*.
2. The administrator copies the public key file (for example, sasdemo.pub) from the SAS Data Loader `Configuration` directory.

   Note: Note: If MapR is the Hadoop environment, then the SSH key file is a PUB file named after the user name found in the mapr-user.json file (for example, etlguest.pub).
3. The administrator appends the SAS Data Loader public key to the file `~designated-user-account/.ssh/authorized_keys`.

   If SAS LASR Analytic Server is configured across a grid of hosts, then the public key is appended in the head node of the grid.

   CAUTION! To maintain access to SAS LASR Analytic Server, you must repeat step 3 each time you replace your installation of SAS Data Loader for Hadoop.

   Note: It is not necessary to repeat this step if you update your vApp by clicking the **Update** button in the SAS Data Loader: Information Center.

Disable banners for the SSH login. The SAS LASR Analytic Server might have banners enabled for SSH logins. Login banners interfere with communication between SAS Data Loader and SAS LASR Analytic Server. Accordingly, all banners must be disabled for the SSH login for the **User ID** that is specified in the LASR Server Configuration window.

To disable banners for a specific user ID, ask an administrator to perform the following steps on the SAS LASR Analytic Server or to the head node on the SAS LASR Analytic Server grid:

1. Login as root or as a user with sudo.
2. Edit the file `/etc/ssh/sshd_config`. For LASR_USER_ID (the **User ID** that is specified in the LASR Server Configuration window, make the following edits.
3. If you want to disable banners for a specific user, add this line to the end of the file: `Match User LASR_USER_ID`
4. Locate any lines in the file that have the Banner option in them. Example: `Banner /etc/issue.net`
5. Comment out these lines by adding a number sign (#) in front of them. Example: `# Banner /etc/issue.net`

6 Restart sshd. For example, in many Linux environments, you would issue a command similar to this: `service sshd restart`

Get connection information from the SAS LASR Analytic Server administrator. Review the fields in the LASR Server panel on page 186 of the Configuration window. Ask the SAS LASR Analytic Server administrator to provide the information that is required to specify a connection in this window.

After these prerequisites have been met, you can add a connection to a SAS LASR Analytic Server. See “Add or Update Connections to SAS LASR Analytic Servers” on page 185.

Additional Prerequisites for SSL Connections

If you want SAS Data Loader to connect to a SAS LASR Analytic Server in a deployment where the SAS Web Server is secured with Secure Socket Layer (SSL), you must do the following tasks.

1 Contact the administrator who is responsible for SSL certificates for your site.

2 Obtain the SSL certificate file that is required to access the SAS LASR Analytic Server. The SSL file contains the public certificates for the trusted certification authorities (CA) for your site. The CA file must be PEM-encoded (base64). The name of the file will be **cacert.pem**.

3 Locate the shared folder (`SASWorkspace\Configuration`) on the SAS Data Loader host.

4 Create a subfolder named **certs** under the shared folder: `SASWorkspace\Configuration\certs`.

5 Copy the SSL certificate file to the **certs** subfolder: `SASWorkspace\Configuration\certs\cacert.pem`.

After these prerequisites have been met, you can add a connection to a SAS LASR Analytic Server, as described in the next section.

Add or Update Connections to SAS LASR Analytic Servers

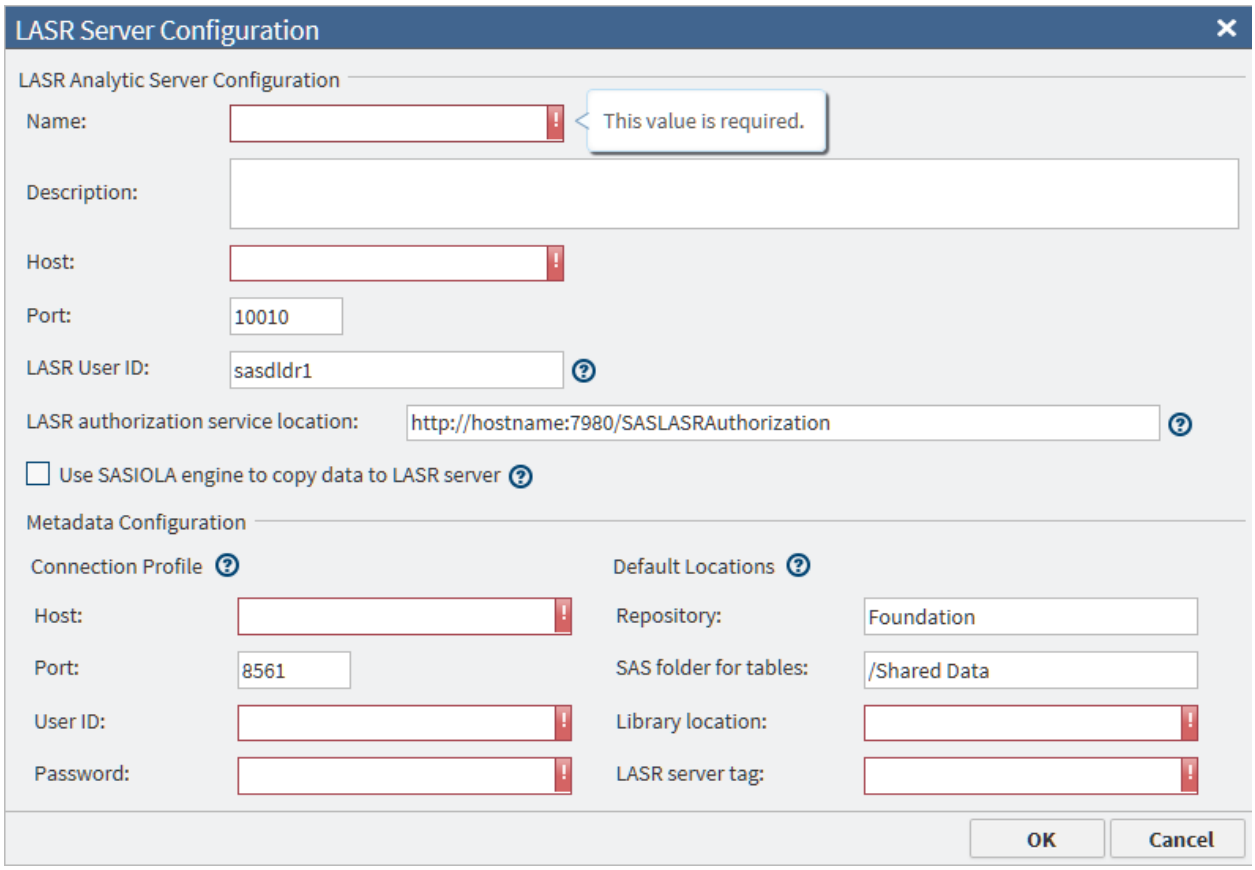
After the prerequisites above have been met, you can add a connection to a SAS LASR Analytic Server. Perform these steps:

1 In the SAS Data Loader directives page click the **More** icon  and select **Configuration**.

2 Click **SAS LASR Analytic Servers**.

3 To configure a new connection to SAS LASR Analytic Server, click the **Add** icon  . To change an existing connection to SAS LASR Analytic Server, click that connection in the list, and then click the **Edit** icon  . To delete a connection to SAS LASR Analytic Server, select it and click the **Delete** icon  .

4 In the LASR Server panel of the Configuration window, enter or change your choice of server name and description in the **Name** and **Description** fields.



5 In the **Host** field, add or change the full network name of the host of the SAS LASR Analytic Server. A typical name is similar to lasr03.us.ourco.com.

6 In the **Port** field, add or change the number of the port that the SAS LASR Analytic Server uses to listen for connection requests from SAS Data Loader. The default port number is 10010.

7 If your Hadoop cluster uses Kerberos for authentication, then the value of the **LASR User ID** field is not used. It is assumed to be the same as the **User ID** that is specified in the **Hadoop Configuration** panel.

If your Hadoop cluster does not use Kerberos for authentication, enter the name of the user account on the SAS LASR Analytic Server that received SSH keys, as described in "Additional Prerequisites When Kerberos Authentication Is Not Used" on page 183. Consult your administrator to confirm whether you should specify a user ID in this field and, if so, which user ID you should use. If no user ID is specified, the user sasdldr1 is used.

8 In the field **LASR authorization service location**, add or change the HTTP address of the authorization service. You can specify an HTTPS URL if you have done some additional set up. See "Additional Prerequisites for SSL Connections" on page 185.

9 If your SAS LASR Analytic Server is configured to run on a grid of multiple hosts, deselect **Use SASIOLA engine to copy data to LASR server**. Not selecting this field indicates that massively parallel processing (MPP) will be used in the SAS Data Loader jobs that use this connection.

If your SAS LASR Analytic Server supports symmetric multiprocessing (SMP) on a single host, click **Use SASIOLA engine to copy data to LASR server**.

10 Under **Connection Profile**, in the lower of the two **Host** fields, add or change the network name of the SAS Metadata Server that is accessed by the SAS LASR Analytic Server.

11 In the lower of the two **Port** fields, add or change the number of the port that the SAS Metadata Server uses to listen for client connections. The default value 8561 is frequently left unchanged.

12 In the **User ID** and **Password** fields, add or change the credentials that SAS Data Loader will use to connect to the SAS Metadata Server. These values are stored in encrypted form.

13 Under **Default Locations**, in the **Repository** field, specify the name of the repository on the SAS LASR Analytic Server that will receive data from Hadoop. The default value `Foundation` might suffice.

14 In the field **SAS folder for tables**, specify the path inside the repository on the SAS LASR Analytic Server that will contain the data that is loaded from Hadoop. The default value `/SharedData` might suffice.

15 In the **Library location** field, add or change the name of the SAS library on the SAS LASR Analytic Server that will be referenced by the Load Data to LASR directive.

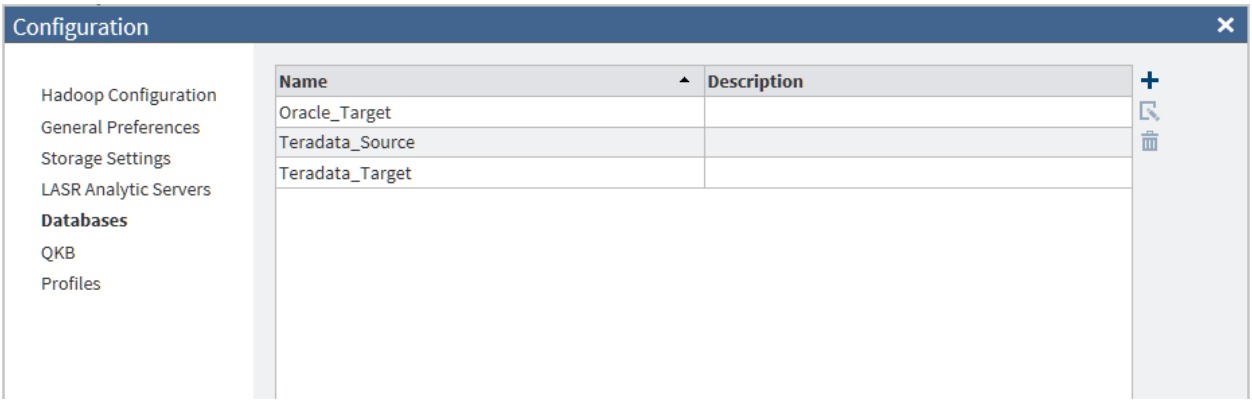
16 In the **LASR server tag** field, add or change the name of the tag that the SAS LASR Analytic server will associate with each table that is loaded from Hadoop. The tag is required to uniquely identify tables.

17 Review your entries and click **OK** to return to the Configuration window.

Databases Panel

Overview

Use the **Databases Panel** to define connections to the databases that supply data to Hadoop and receive data from Hadoop. SAS Data Loader directives such as Copy Data to Hadoop and Copy Data from Hadoop require JDBC connections in order to access tables in databases. The **Databases** panel of the Configuration window enables you to maintain these connections.



To prepare to add or update database connections, ensure that the JDBC database driver in Hadoop matches the driver in the SASWorkspace folder on your SAS Data Loader host. As needed, see "Copy JDBC Drivers to the SAS Data Loader Host" on page 188.

When the JDBC drivers are in place on your local host, see "Add or Update Database Connections" on page 188.

Copy JDBC Drivers to the SAS Data Loader Host

SAS Data Loader uses the SQOOP and Oozie components installed with the Hadoop cluster to move data to and from external databases. SAS Data Loader accesses those same databases directly, to display schemas and tables. For this reason, your instance of SAS Data Loader needs to receive the same set of JDBC drivers that are installed in the Hadoop cluster.

During the installation of SAS Data Loader, as described in the *SAS Data Loader for Hadoop: vApp Deployment Guide*, your Hadoop administrator is asked to provide you with the required JDBC drivers. Those are the drivers that you will install on in your SASWorkspace folder.

You can also follow these steps if your site adds support for new databases, after the installation of SAS Data Loader:

1. As needed, ask your Hadoop administrator for a copy of the JDBC drivers that are installed on your Hadoop cluster.
2. On the SAS Data Loader host, navigate to the SASWorkspace folder and open the JDBCDrivers folder. Here is a typical path to the JDBCDrivers folder:

   ```
   C:\Program Files\SAS Data Loader\2.x\SASWorkspace\JDBCDrivers
   ```

3. Copy the files for the JDBC drivers into JDBCDrivers folder.
4. Restart the vApp so that it can pick up the JDBC drivers.

   Note: Before you stop the vApp, check the "Run Status" directive to ensure that all jobs are stopped and saved.

   Note: Suspending the vApp is not sufficient to detect the new drivers.

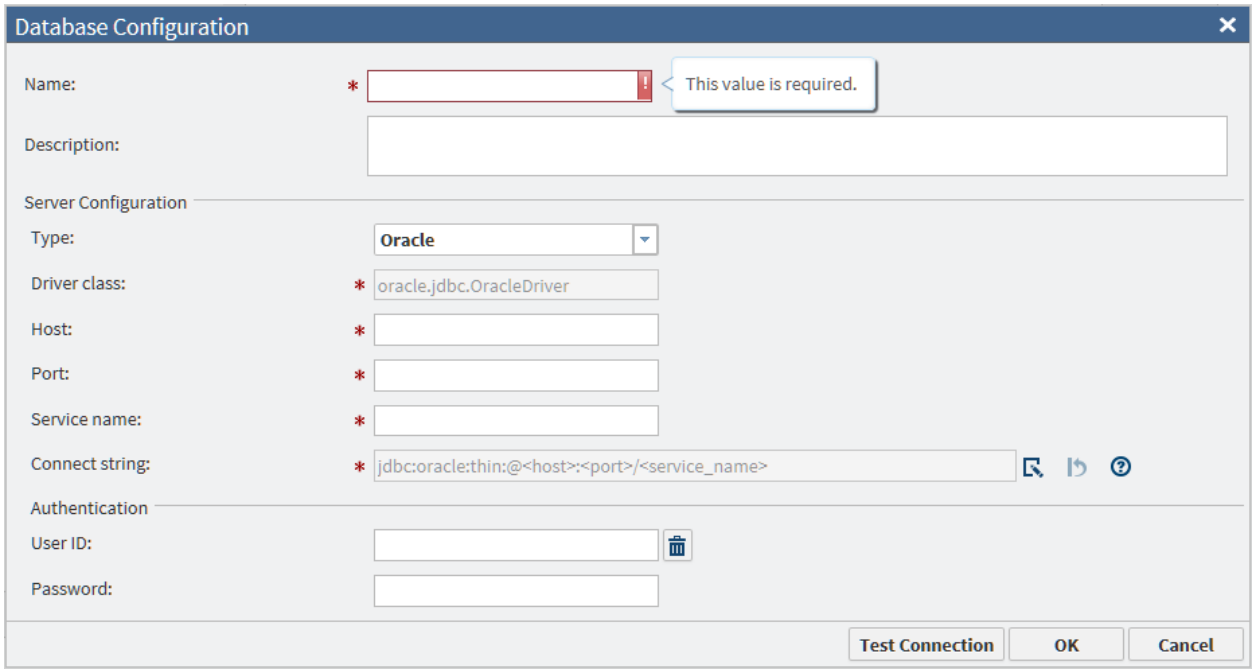
SAS Data Loader now has access to the JDBC drivers. The next task is to add connections to the databases for which you have new JDBC drivers.

Add or Update Database Connections

After you have copied the appropriate JDBC drivers into the shared folder on the SAS Data Loader host, you can add connections to the corresponding databases.

1. Contact the administrators of the databases to which you want to connect. Ask for the usual information that you would need to connect to a database: host name, port, log on credentials, and so on.
2. In SAS Data Loader, click the **More** icon  and select **Configuration**.
3. In the Configuration window, click **Databases**. To add a new database connection, click **Add**  . To edit an existing database connection, click the name of the connection, and then click **Edit**  .

4 The values of **Driver class** and **Connect string** are generated automatically when you select either Teradata or Oracle in the **Type** field. For an Oracle connection that requires a Service ID (SID), enter the SID in the **Database name** field. If you select **Other**, you must obtain these values from the JDBC driver provider.



5 When the configuration data is ready, click **Test Connection** to verify that the connection is operational.

6 If the test fails for a new Oracle connection, then examine the **Connect string** field. If the string has either of the following formats, then change the string to the other format and test the connection again.

```
jdbc:oracle:thin:@raintree.us.ourco.com:1521:oadev

jdbc:oracle:thin:@raintree.us.ourco.com:1521/oadev
```

One version uses a final colon character. The other version uses a final slash character.

To edit the **Connect string** field, click **Edit** .

7 Click **OK** to close the window. SAS Data Loader directives can now use this database connection.

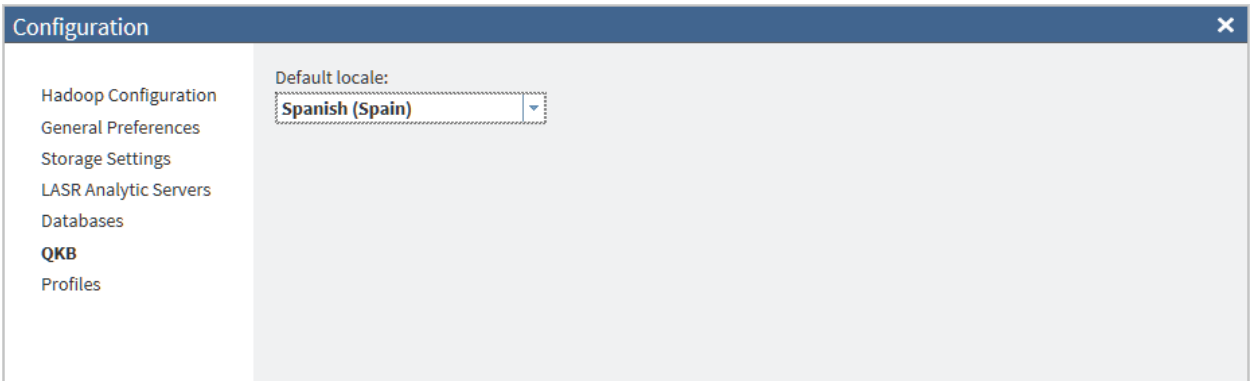
QKB Panel

Use the **QKB** panel in the Configuration window to specify the national language that is used by default in the Quality Knowledge Base.

A SAS Quality Knowledge Base (QKB) is a collection of files that store data and logic that define data management operations such as parsing, standardization, and matching. SAS Data Loader for Hadoop refers to the QKB when cleansing data in Hadoop.

A QKB supports locales that define how spoken language is written and used in geographic regions.

To choose a different **Default locale**, select a locale from the menu. The default locale should match the typical locale of your source data.



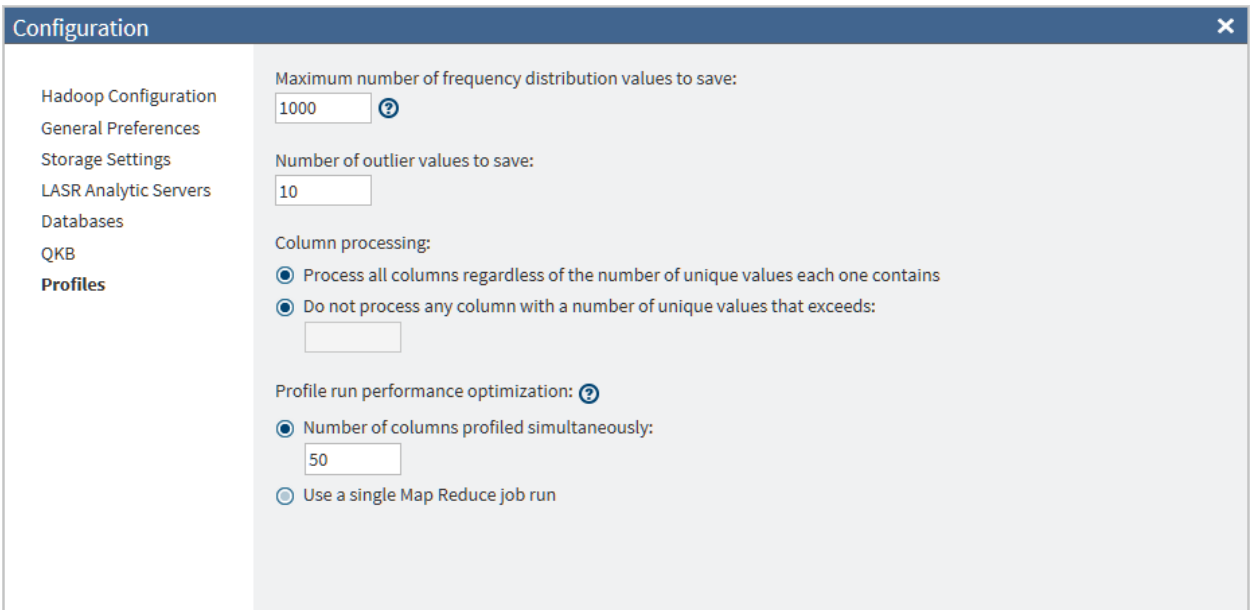
You can override the default locale in any of the data quality transformations in the Cleanse Data directive. For more information about this directive, see “Cleanse Data” on page 30.

Profiles Panel

Use the **Profiles** panel to configure the reports that are collected on specified Hadoop tables using the Profile Data directive. For more information about this directive, see Chapter 5, “Profile Data,” on page 109.

Profile reports enable you to assess the composition, organization, and quality of tables in Hadoop.

Data profiling tasks can be resource-intensive. Accordingly, the **Profiles** panel of the Configuration window enables you to change defaults, which can improve the performance of new profile jobs.



You can change the following default options for profiles:

Maximum number of frequency distribution values to save
specifies the maximum number of frequency distribution values (1–99999999) to save during the profile run. The default value is 1000. If there are more frequency distribution values than this number, the less-frequent values are combined into an Other frequency distribution.

Number of outlier values to save
specifies the maximum number of outlier values (1–99999999) to save during the profile run. The default value is 10, which indicates that the 10 highest and 10 lowest values are saved.

Column processing
specifies how columns are processed with regard to unique values. When you select **Do not process any column** , the default number of 1000 appears in the text box. Any column that has more than this number of unique values is, in effect, an outlier column. Continued processing of outlier columns adds little value to the profile report. You can increase the performance of your profile jobs by excluding columns with unique values.

Number of columns profiled simultaneously
specify the number of columns that are processed simultaneously by default. The default value of this field is 50.

Use a single MapReduce job run
select this option when you primarily profile small tables with less than 50 columns.

Develop Expressions for Directives

Introduction

Many directives provide tasks that incorporate the results of user-written expressions. For example, in the Transform Data directive, the Filter task enables you to specify a user-written expression that excludes source rows from the target. Use this section to help you write your own expressions.

About Implicit Assignment

In all expressions in SAS Data Loader for Hadoop, the return value of the expression is always implicitly assigned to the currently processed row of the specified column. The expression does not use the usual format *variable*=[*expression*]. Instead, the value of the first clause in the expression is written into the specified target column.

If the expression contains only one clause, then the returned value is obvious, as shown in the following example:

```
UPCASE(customer_first_name);
```

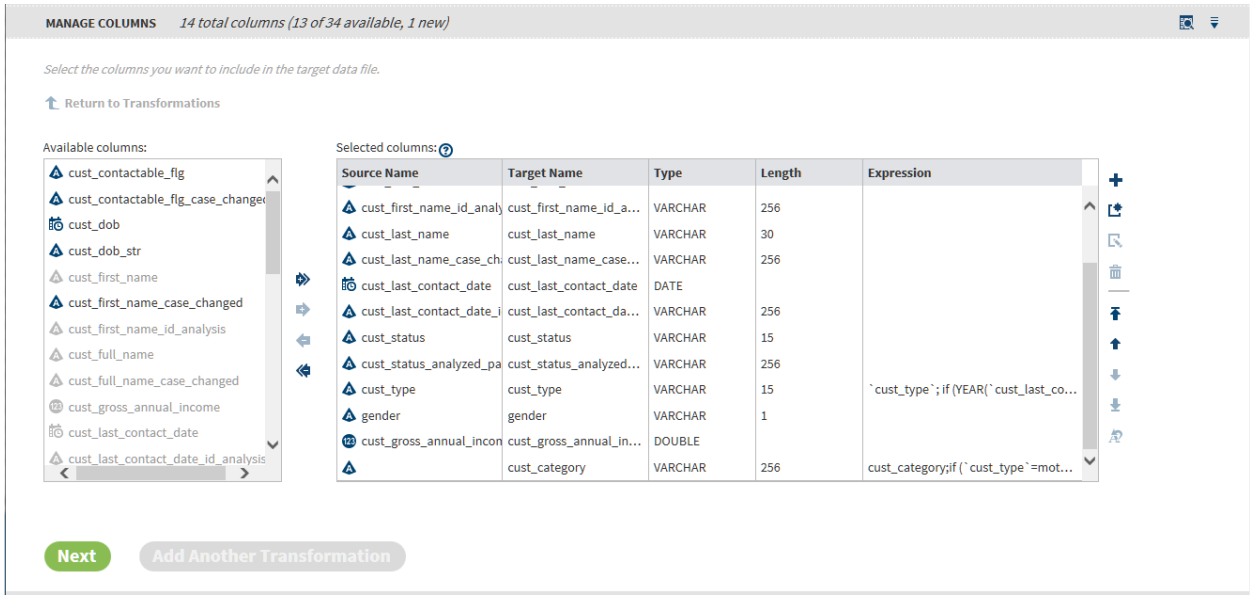
If the expression contains more than one clause, then the first clause needs to be a placeholder value, as shown in the following example:

```
customer_first_name;
if(LENGTH(customer_last_name) > 10) then
    customer_first_name=UPCASE(customer_first_name);
```

About Column Names in EEL Expressions

When Spark is selected as the run-time target, user-written DataFlux Expression Engine Language expressions (EEL expressions) can be applied in the Manage Columns transformations. The Manage Columns transformations are available in the Cleanse Data directive and the Transform Data directive.

In the Manage Columns transformations, EEL expressions can modify data in existing target columns, or they can generate new data for new target columns. In both cases, all of the columns that are named in the EEL expression need to appear in the **Selected columns** list in the Manage Columns transformation. Column names from the **Available columns** list cannot be named in EEL expressions.



A second important consideration is that any column that is named in an expression has to be listed in **Selected columns** above the column that contains the expression. This ordering ensures that all of the variables that appear in the expression are defined before they are referenced.

In the preceding example, a new column is positioned at the bottom of the **Selected columns** list, as the last column in the target table. In this position, the expression can reference any of the other columns.

The preceding example also shows an expression that modifies values in the cust_type column. In that position, the expression could not reference the gender column.

If you need to reorder your columns to accommodate your expressions, then you can create a second Managed Columns transformation. In that second transformation, you can move any column into any position, as needed to meet the requirements of the target table.

Troubleshooting

Active Directory (LDAP) Authentication

If Active Directory and LDAP (Lightweight Directory Access Protocol) are used to protect your Hadoop cluster, an LDAP user and password must be specified in the Hadoop connection for SAS Data Loader. For more information about the Hadoop connection, see “Hadoop Configuration Panel” on page 176.

Oozie does not support LDAP authentication. SAS Data Loader uses the SQOOP and Oozie components installed with the Hadoop cluster to move data to and from a DBMS. Accordingly, when LDAP authentication is used with your Hadoop cluster, directives that rely on Oozie, such as Copy Data To Hadoop, do not receive the authentication benefits provided by LDAP. However, the operation of these directives is otherwise unaffected.

Change the File Format of Hadoop Target Tables

In Hadoop file system (HDFS), tables are stored as one or more files. Each file is formatted according to the Output Table Format option, which is specified in each file. In SAS Data Loader, when you create a new target table in Hadoop, the Output Table Format option is set by the value of the **Output table format** field.

You can change the default value of the **Output table format** field in the SAS Data Loader Configuration window. In any given directive, you can override the default value using the **Action** menu icon in the **Target Table** task.

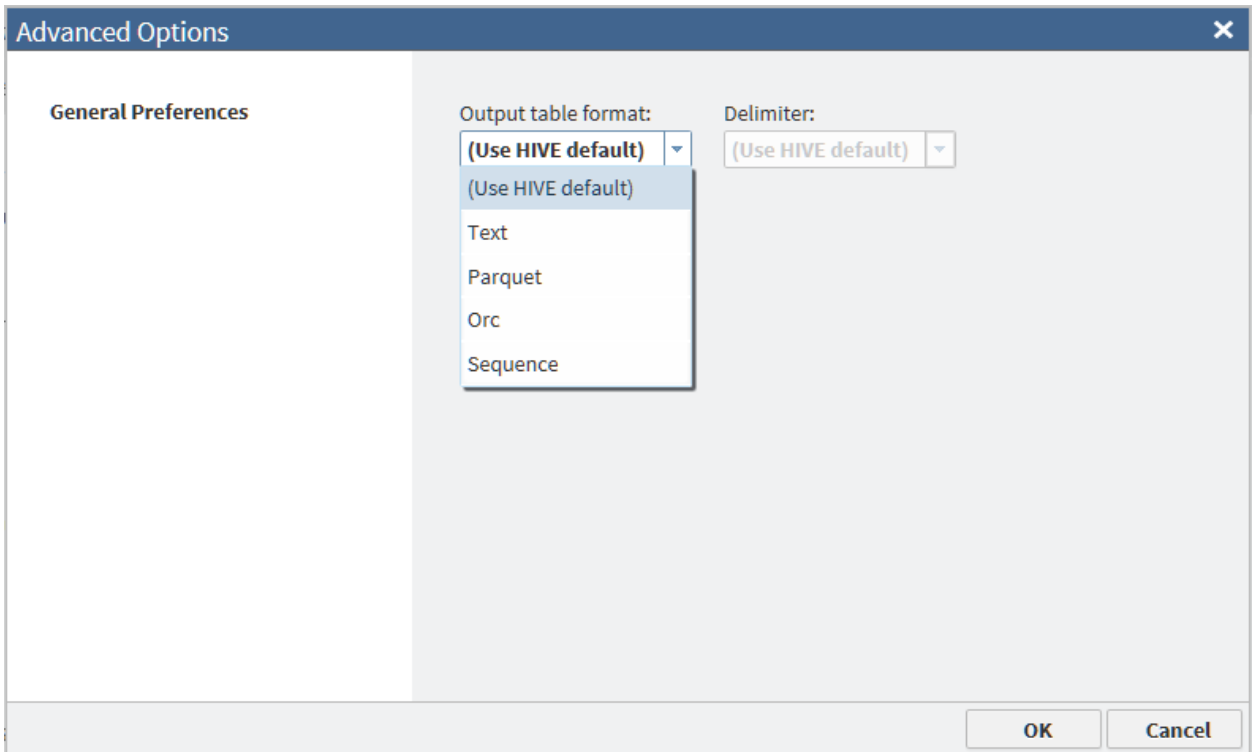
The default format is applied to all new target tables that are created with SAS Data Loader. To override the default format in a new table or an existing table, you select a different format in the directive and run the job.

To change the default value of the **Output table format** field, go to the SAS

Data Loader directives page, click the **More** icon  , and select **Configuration**. In the Configuration window, click **General Preferences**. In the General Preferences panel, change the value of **Output table format**.

To override the default value of the **Output table format** field for a given target in a given directive, open the directive and proceed to the **Target Table** task.

Select a target table, and then click the **Action** menu  for that target table. Select **Advanced Options**, and then, in the Advanced Options window, set the value of **Output table format**. The format that you select will apply only to the selected target table. The default table format is not changed. The default format will continue to be applied to all new target tables.



The available values of the **Output table format** field are defined as follows:

Use HIVE default
: specifies that the new target table receives the Output Table Format option value that is specified in HDFS. This is the default value for the **Output table format** field in SAS Data Loader.

Text
: specifies that the new target table is formatted as a series of text fields that are separated by delimiters. For this option, you select a value for the **Delimiter** field. The default value of the **Delimiter** field is **(Use HIVE default)**. You can also select the value **Comma**, **Space**, **Tab**, or **Other**. If you select **Other**, then you enter a delimiter value. Valid values consist of single ASCII characters that are numbered between 0 and 127 (decimal). An ACII character number can be specified as the delimiter using a three-digit octal value between 0 and 177.

Parquet
: specifies the Parquet format, which is optimized for nested data. The Parquet algorithm is considered to be more efficient that using flattened nested namespaces. The Parquet format requires HCatalog to be enabled on the Hadoop cluster. Hadoop administrators can refer to the topic "Additional Configuration Needed to Use HCatalog File Format" in the SAS 9.4 In-Database Products: Administrator's Guide.

  Note: If your cluster uses a MapR distribution of Hadoop, or if the selected run-time target is Spark, then the Parquet output table format is not supported.

ORC
: specifies the Optimized Row Columnar format, which is a columnar format that efficiently manages large amounts of data in Hive and HDFS.

Sequence
specifies the SequenceFile output format, which enables Hive to efficiently run MapReduce.

Consult your Hadoop administrator for advice about output file formats. Testing might be required to establish the format that has the highest efficiency on your Hadoop cluster.

Change the Maximum Length for SAS Character Columns

By default, the character columns of the source tables of these directives are expanded or truncated to 1024 characters in length. The valid range for the maximum length option is 1–32,767 characters. The default length should perform well in most cases, though there might be situations where a larger value is required.

Note: As you increase the maximum length for SAS character columns, you also increase the likelihood that performance will be affected.

The affected directives are as follows:

- Cleanse Data
- Transform Data
- Transpose Data
- Load Data to LASR
- Copy Data to Hadoop
- Copy Data from Hadoop

To change the default maximum length for SAS character columns for all new directives, go to the SAS Data Loader directives page, click the **Action** menu  , and select **Configuration**. In the Configuration window, select **General Preferences**, and specify the desired length for SAS character columns.

Note: If you change the default maximum length of SAS columns, the new value applies only to new directives.

You can override the default maximum length for SAS character columns in individual directives, without changing the default. In one of the directives listed above, open the **Source Table** task, click the **Action** menu, and select **Advanced Options**. In the Advanced Options window, specify the desired length for SAS character columns.

Note: The directives Cleanse Data and Transform Data can be enabled to run in the Spark run-time environment. When Spark support is enabled for one of these directives, the maximum length of SAS columns can be determined by the value of a configuration option. To learn about the configuration option and the change in behavior, see “String Truncation in Spark-Enabled Directives”.

Change the Temporary Storage Location

If the default temporary storage directory for SAS Data Loader is not appropriate for some reason, you can change that directory. For example, some SAS Data Loader directives might fail to run if they cannot write to the temporary directory. If that happens, ask your Hadoop administrator if the sticky bit has been set on

the default temporary directory (typically `/tmp`). When a directory's sticky bit is set, only the directory's owner, a file's owner, or the root user can rename or delete the directory or files in the directory. This is a security measure to avoid deletion of critical folders and their contents, even when other users have full permissions. If that is the case, specify a new location for temporary storage. Click  to open the Configuration window. In the **Storage Settings** panel, use the **SAS HDFS temporary storage location** field, as described in "Storage Settings Panel".

Discover New Columns Added to a Source after Job Execution

When you add columns to a source table, any directives that need to use the new columns need to discover them. To make the new columns visible in a directive, open the Source Table task, click the source table again, and click **Next**. The new columns will then be available for use in the body of the directive, in a transformation or query, for example.

Avoid Using Reserved Keywords in Column Names

For column names, avoid using words that are DS2 or DBMS reserved keywords. For some directives, if you use a reserved keyword for the name of a column that is the target of the directive, it can result in a run-time error. For more information, see "Naming Requirements for Schemas, Tables, and Columns" on page 11.

Hive Limit of 127 Expressions per Table

Due to a limitation in the Hive database, tables can contain a maximum of 127 expressions. When the 128th expression is read, the directive fails and the SAS log receives a message similar to the following:

```
ERROR: java.sql.SQLException: Error while processing statement: FAILED:
Execution Error, return
       code 2 from org.apache.hadoop.hive.ql.exec.mr.MapRedTask
ERROR: Unable to execute Hadoop query.
ERROR: Execute error.
SQL_IP_TRACE: None of the SQL was directly passed to the DBMS.
```

The Hive limitation applies anytime a table is read as part of a directive. For SAS Data Loader, the error can occur in aggregations, profiles, when viewing results, and when viewing sample data.

Override the Default Hive Storage Location for Target Tables

When you work with directives that create target tables, those tables are stored in a directory location in the Hadoop file system. The default location is defined in the "Storage Settings Panel" on page 181 of the **Configuration** window.

Follow these steps to override the default Hive storage location for an individual directive:

1. Proceed through the initial tasks for the directive as usual. For example, if you are using the Transform Data directive, you would select a source table and specify any transformations for the data.
2. When you reach the Target Table task in the directive, click  to open the Advanced Options window.
3. On the General Preferences page, select **Specify alternate storage location** for the **Hive storage location** setting, and then click to open the Select Directory window.
4. Navigate to a folder where you want to store the target table and click **OK**. You can also create a new folder, if needed.

   Note: To use the alternate location, you must have appropriate permissions to the selected directory.
5. Continue through the remaining tasks for the directive to submit the job.

TIP Due to a defect in org.apache.sqoop.teradata.TeradataConnManager, an insert into an existing Teradata table at an alternative location is not supported for HortonWorks or any distribution that uses org.apache.sqoop.teradata.TeradataConnManager.

Unsupported Hive Data Types and Values

The Hive database in Hadoop identifies table columns by name and data type. To access a column of data, SAS Data Loader first converts the Hadoop column name and data type into its SAS equivalent (numeric or character.) When the transformation is complete, SAS Data Loader writes the data into the target table using the original Hadoop column name and data type.

If your target data is incorrectly formatted, then you might have encountered a data type conversion error.

The Hive database supports a Boolean data type. SAS does not support the Boolean data type in Hive at this time. Boolean columns in source tables will not be available for selection in SAS Data Loader.

The BIGINT data type in Hive supports integer values larger than those that are currently supported in SAS. BIGINT values that exceed +/– 9,223,372,036,854,775,807 generate a stack overflow error in SAS.

Complex data types are not supported by SAS Data Loader.

Although SAS Data Loader does not generate HiveQL UNION statements, you can submit them in the directive Run a Hadoop SQL Program. It is also possible to add UNION statements to the code that is generated by the directives Query or Join Data, or Sort and De-Duplicate Data. (Your version of Hive must be new enough to support UNION statements.)

Restarting a Session after Time-out

SAS Data Loader records periods of inactivity in the user interface. After a period of continuous inactivity, the current web page receives a session time-out warning message in a window. If you do not provide input within three minutes after you receive the warning, the current web page is replaced by the Session

Time-out page. You can restart your session by clicking the text **Return to the SAS Data Loader application**.

When a session terminates, any directives that you did not save or run are lost.

To open an unsaved directive that you ran before your session terminated, follow these steps:

1. Open the Run Status directive.
2. Locate the entry for your unsaved directive.
3. If the unsaved directive is still running, click the Refresh button.
4. If the directive continues to run, either click **Stop** in the action menu, or wait for the completion of the run.
5. In the action menu, select **Open** to open the directive.
6. In the open directive, select **Save** from the title bar.

Replace User Preferences after User ID Changes in MapR or Kerberos Environments

Overview

If your cluster uses a MapR distribution of Hadoop, or if your cluster uses Kerberos authentication, then user preferences are dropped when the user ID changes. A new user ID invalidates the existing user preferences, which are associated with the previous user ID. The following preferences can be invalidated, primarily in the **Configuration** window:

- values from the **General Preferences** panel
- storage locations from the **Storage Settings** panel
- default locale from the **QKB** panel
- configuration information in the **Profiles** panel
- list of recently used tables

User preferences are not affected by a user ID change when the Hadoop cluster is not secured with Kerberos and when the Hadoop distribution is not MapR.

Causes

User ID changes take place in MapR when a new ID is entered into the file mapr-user.json. That new user ID must also be entered into the **Hadoop Configuration** panel of the Configuration window. The new user ID invalidates the existing SAS Data Loader user preferences. The path to the JSON file on the client host is ***vApp-install-path*\SASWorkspace\hadoop\conf\mapr-user.json**.

In Kerberos environments, the primary cause of a user ID change occurs when a value appears in the **User ID** field of the **Hadoop Configuration** panel. This value needs to be deleted by clicking the trash can icon next to the **User ID** field. This field is blank in the Kerberos environment because Kerberos uses the operating system login user ID for authentication purposes. At times, a different

user ID is applied, particularly when you move from a non-Kerberos environment to a Kerberos environment.

A user ID change can also take place when a user other than yourself starts or restarts the SAS Data Loader vApp.

A third possible user ID change can occur if a SAS Data Loader backup is not secure, and if the restore or migration is secured with Kerberos. To learn more about backups and restores, see the *SAS Data Loader for Hadoop: vApp Deployment Guide*.

Resolution

To resolve user preferences after a user ID change, open the Configuration window and update as needed the values in the affected panels. Click **OK** to save your changes.

The settings in the panels **LASR Analytic Servers** and **Databases** panels are not invalidated as a result of a user ID change.

Note: In the **General Preferences** panel, if no value is specified for the field **Maximum length for SAS columns**, then SAS Data Loader supplies a system default of 1024. When Hadoop Spark is specified as the run-time environment, the default value or an explicit value can be overridden by a configuration option, as described in"String Truncation in Spark-Enabled Directives".

Recommended Reading

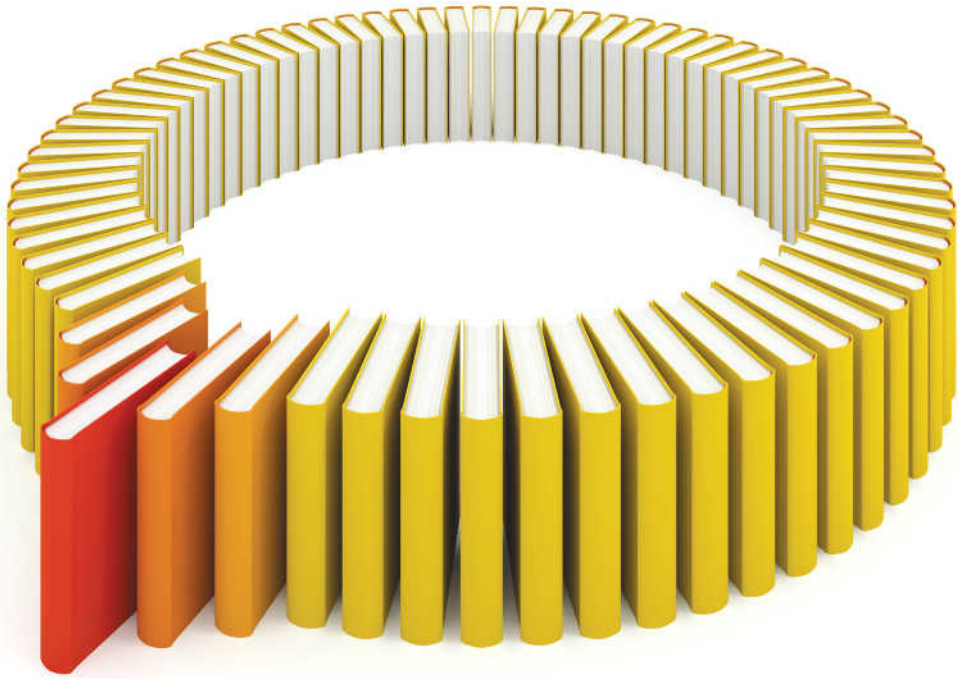
- *SAS Data Loader for Hadoop: vApp Deployment Guide*
- *SAS In-Database Products: Administrator's Guide*
- *SAS DS2 Language Reference*
- *SAS DataFlux Expression Engine Language: Reference Guide*
- *SAS/ACCESS for Relational Databases: Reference*
- SAS Quality Knowledge Base for Contact Information: Online Help

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

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-0025
Fax: 1-919-677-4444
Email: sasbook@sas.com
Web address: sas.com/store/books

Index

V



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

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

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