SAS® Data Loader 3.1 for Hadoop: User's Guide

SAS® Documentation
September 19, 2017

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS® Data Loader 3.1 for Hadoop: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Data Loader 3.1 for Hadoop: User's Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

September 2017

3.1-P2:dmddug

Contents

What's New in SAS Data Loader 3.1 for Hadoop

Overview

The main enhancements for SAS Data Loader 3.1 include the following:

- Directives are now run by a selected SAS Workspace Server
- Multiuser web application is now based on the SAS Intelligence Architecture
- Centralized administrative support now manages servers, access, directives, and QKB updates
- New IT-friendly administrative deployment process uses SAS deployment tools
- Additional high-performance connectivity options available through SAS/ACCESS
- Updated support enables the latest Hadoop technologies
- Improved integration supports impact analysis and view and run directives from products such as SAS Data Integration Studio

Directives are now run by a selected SAS Workspace Server

In release 3.1M1, you can select the SAS Workspace Server to run the directives that you create. Your SAS administrator specifies a default SAS Workspace Server for all users and all directives. In SAS Data Loader, your personal setting is available in the **Hadoop Configuration** panel of the **Configuration** window.

Multiuser web application is now based on the SAS Intelligence Architecture

In release 3.1 the SAS Data Loader for Hadoop web application runs entirely on servers in the SAS Intelligence Architecture. Client hosts require only a supported web browser. The server-based web application enables multiuser capabilities, simplifies deployment, eliminates configuration requirements for users, improves integration, and enhances Hadoop distribution support. Administrators define access to data sources. Users control runtime preferences for directives.

Centralized administrative support now manages servers, access, directives, and QKB updates

In release 3.1, improvements to the administration of SAS Data Loader for Hadoop provide the following benefits:

- Increased ease-of-use for the web application. All administration and connectivity is now managed by administrators. Users retain the capability of specifying preferences in the web application.
- Centralized access control for data and directives customizes the user interface and promotes content sharing and reuse for users and groups.
- SAS server definitions are now managed in SAS Management Console.
- The new role SAS Data Loader Administrator now can now view and manage the directives of all users. This administrator also publishes updates of the SAS Quality Knowledge Base in Hadoop into the SAS Information Architecture.

New IT-friendly administrative deployment process uses SAS deployment tools

In release 3.1 the deployment process leverages the IT-friendly SAS deployment technologies to deploy the following components:

- SAS Data Loader for Hadoop web application.
- SAS servers in the SAS Intelligence Architecture.
- SAS content that is deployed to the Hadoop cluster.

The deployment process now automates the collection of required Hadoop configuration files. Also, the deployment process now supports Hadoop management technologies such as Ambari and Cloudera Manager.

Additional high-performance connectivity options available through SAS/ACCESS and LIBNAME engines

In release 3.1 SAS LIBNAME engines and SAS/ACCESS engines provide direct access to many types of external tables. For SAS data sets and files, access can be local or shared. Local SAS content is accessed on the client host. Shared SAS content is accessed on a SAS Workspace Server.

Updated support enables the latest Hadoop technologies

In release 3.1 SAS Data Loader for Hadoop supports the following Hadoop technologies:

- The latest versions of the supported Hadoop distributions.
- Enhanced support for authentication modes such as Kerberos.
- HDFS high availability and encryption.

Improved integration supports impact analysis and view and run directives from products such as SAS Data Integration Studio

In 3.1 products such as SAS Data Integration Studio have enhanced integration with SAS Data Loader for Hadoop, including saved content that can be shared in SAS folders.

Chapter 1

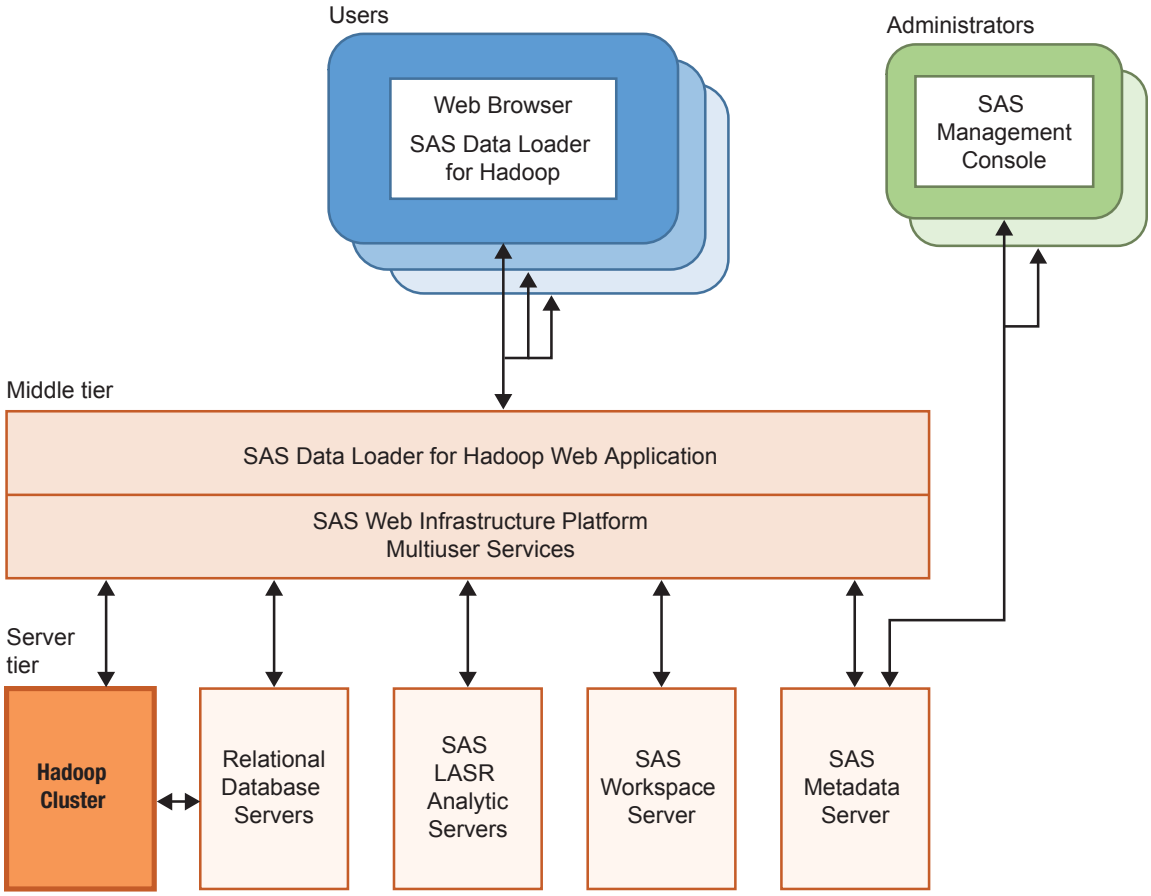
About SAS Data Loader for Hadoop

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 consists of a web application, elements of the SAS 9.4 Intelligence Platform, and SAS software on the Hadoop cluster.

The SAS Data Loader for Hadoop web application provides an interactive interface that guides you through the process of creating directives. You then run the directives on a SAS Workspace Server. The SAS Workspace Server executes generated code, sends code to Hadoop, and receives responses from Hadoop. During execution, directives provide access to generated code, log information, error messages, and results as they become available. You can save directives, update them, and execute them as needed.

Figure 1.1 *SAS Data Loader for Hadoop System Diagram*



Multiple users can access the SAS Data Loader for Hadoop web application. The web application uses the SAS Web Infrastructure Platform to connect to SAS servers, to a Hadoop cluster, and to network database management servers. The SAS Metadata Server manages access to data sources and software capabilities for individuals and groups. SAS Management Console provides an administrative interface that configures the web application and manages SAS servers.

SAS software is also deployed to each node in the Hadoop cluster. SAS software on the cluster includes the following components:

- SAS Quality Knowledge Base, which supports data cleansing capabilities in Hadoop
- SAS Embedded Process software, which runs SAS programs in Hadoop
- SAS Data Quality Accelerator, which runs data quality capabilities in Hadoop
- SAS Data Loader for Hadoop Spark Engine, which executes data integration and data quality tasks in Apache Spark

To view a presentation that demonstrates the capabilities of SAS Data Loader for Hadoop, see the SAS® Data Loader for Hadoop Demo.

What Does SAS Data Loader for Hadoop Do?

SAS Data Loader for Hadoop provides a point-and-click interface for profiling, managing, cleansing, and copying data to and from Hadoop. Power users can create directives that run SAS or HiveQL programs in Hadoop.

You can use a set of *directives* or wizards to perform tasks such as the following:

Tasks	Description
Browse data	Open data sources and selectively display sample data from source tables. For more information, see "Browse Tables" on page 24.
Copy data to and from Hadoop	Create directives that copy data from relational databases or SAS to Hadoop via parallel, bulk data transfer. For more information, see "Copy Data to Hadoop" on page 104, and "Copy Data from Hadoop" on page 115. Gain direct access to network data sources using SAS/ACCESS technology. Import data from delimited text files, such as comma-separated value (CSV) files. For more information, see "Import a File" on page 109. Upload and download from local workstations to SAS libraries in conjunction with copying or importing data to or from Hadoop.
Transform and transpose data	Transform data by filtering rows, managing columns, and summarizing rows. For more information, see "Transform Data" on page 77. Select columns and transpose or group them. For more information, see "Transpose Data" on page 86.
Delete rows	Delete selected rows from selected source tables. For more information, see "Delete Rows" on page 45.
Cleanse data	Standardize, match, parse, and perform other data quality functions in Hadoop. For more information, see "Cleanse Data" on page 26. Use rules and expressions to filter data. For more information, see "About Expressions and the Advanced Editor" on page 43.
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 72.

Tasks	Description
Query or join data	Query a table or join multiple tables. For more information, see "Query or Join Data" on page 64. Run aggregations on selected columns. For more information, see "About the Aggregations in the Summarize Rows Transformation" on page 85. 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 127.
Match-merge data	Merge two or more source tables, while replacing matching rows with a single row in the target table. The Match-Merge Data directive supports SAS DATA step merge in Hadoop. For more information, see "Match-Merge Data" on page 55.
Cluster-survive data	Define rules to cluster similar records into groups. You can also create a best record to represent the information collected from all group members. For more information, see "Cluster-Survive Data" on page 49.
Profile data	Analyze source columns from one or more tables to determine patterns, uniqueness, and completeness. For more information, see "Profile Data Directive" on page 90. Show history of changes to data over time. View data profile reports. Add notes to a data profile report to explain a result or ask a question.
Run user-written code	Use the Run a SAS Program directive to execute user-written SAS code. For more information, see "Run a SAS Program" on page 125. Use the Run a Hive Program directive to execute user-written Hive code. For more information, see "Run a Hadoop SQL Program" on page 127.
Chain directives	Specify a list of existing directives, including chain directives, that execute in series or in parallel. For more information, see "Chain Directives" on page 138.
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 132. 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 134.

Tasks	Description
Load data to SAS LASR Analytic Server	Copy Hadoop columns into memory on 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 121.
Specify preferences	Display and test Hadoop connections and set preferences for the use of Spark or Impala. Configure SAS character column width and performance criteria for profiling, among other settings. Administrators see additional configuration features.For more information, see "Set Global Options" on page 143.

About Administrative Roles

SAS Data Loader for Hadoop is administered by two roles, the SAS administrator and the SAS Data Loader administrator. The traditional SAS administrator manages installation, deployment, system configuration, and authorization to data sources and SAS Data Loader directives. The SAS administrator uses SAS Management Console to configure and manage the SAS servers inside and outside of Hadoop. The SAS administrator also defines libraries and connections to Hadoop servers and relational database servers outside of Hadoop.

The SAS Data Loader administrator is a SAS Data Loader user who has been assigned to the SAS Data Loader: Administrator group. Members of this group have access to the QKB Administration panel in the web application's Configuration window. In addition, the SAS Data Loader administrator can see and manage all current and previous run records in the Run Status directive.

Individuals can be assigned to both administrative roles.

Note: If you cannot access a particular data source, table, library, folder, saved directive, or other shared content, contact your SAS administrator and request access to that resource.

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

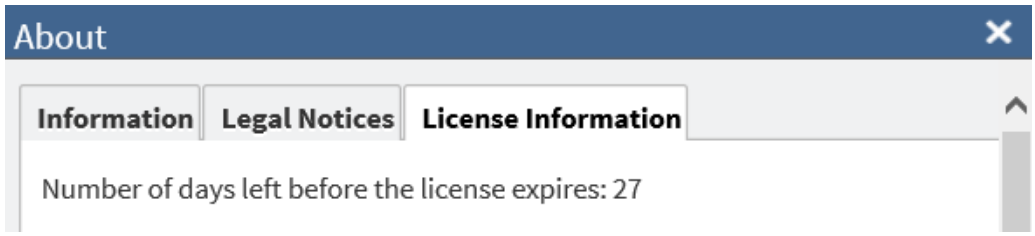
SAS provides customer support through self-help and assisted-help resources. See our Support page for more information about these resources.

Documentation and System Requirements

If you select **Help** from the Help icon at top right of most windows, documentation is displayed in the SAS Help Center. This documentation is also available from the SAS Data Loader for Hadoop documentation page on support.sas.com.

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 window, the **License Information** tab shows how many days are left before your license expires. For information about renewing your license, contact your SAS representative.



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.

Click this link to display SAS Data Loader 3.1 System Requirements. System requirements are also available from the SAS Data Loader for Hadoop documentation page on support.sas.com.

Chapter 2

Getting Started

Prerequisites

Verify That Your Browser Is Supported

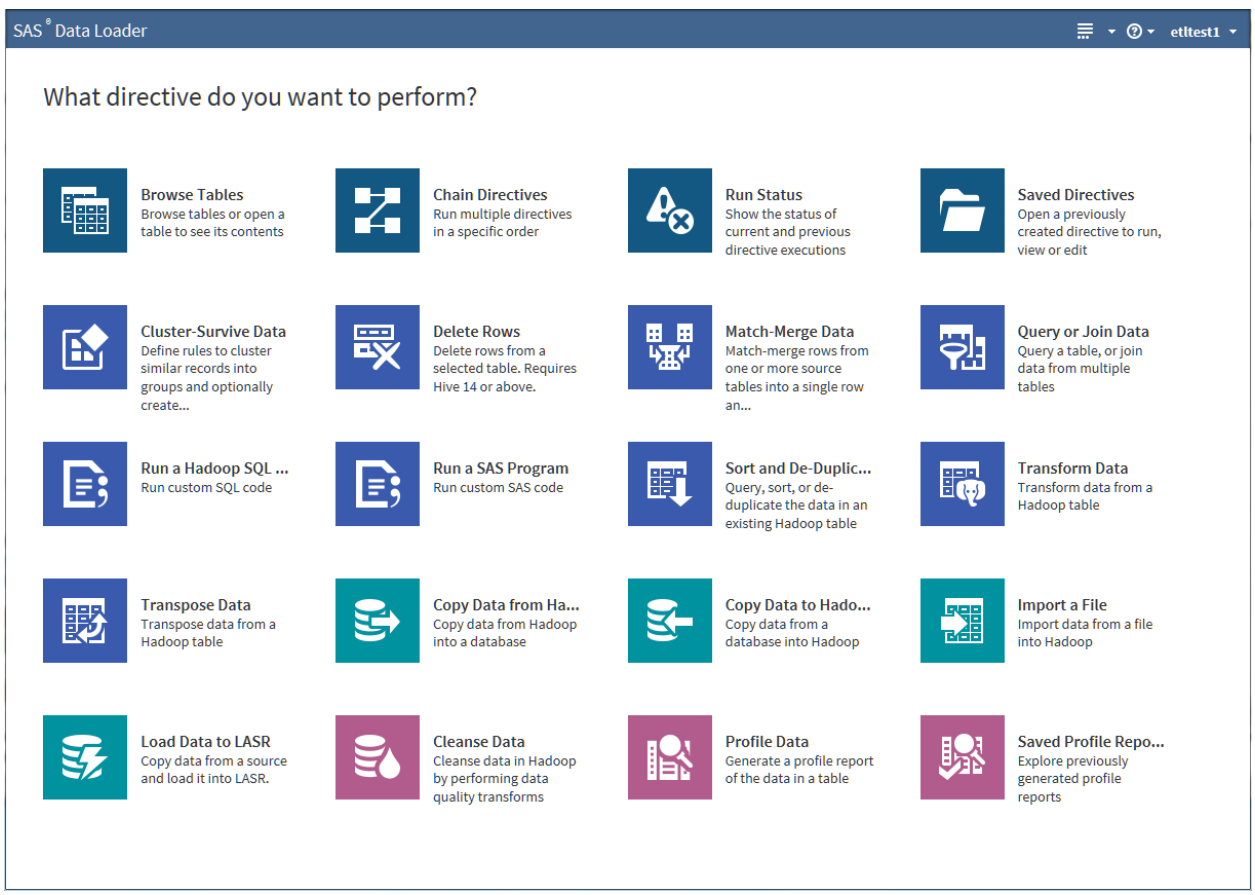
The list of web browsers that SAS supports is updated as new products are tested and approved. For the latest information, see the Support for Web Browsers and Plug-ins page.

Open the Web Application

To open SAS Data Loader for Hadoop, enter its URL into a supported web browser. If you were not given the URL, and if your site is using the default port number, then enter the following URL: http://*hostname*/SASDataLoader. If your site is not using the default port number, then use this URL: http://*hostname*:*port-number*/SASDataLoader, as shown in this example: http://etlhost17.ourco.com:7980/SASDataLoader.

The web browser responds to the URL in one of two ways. If your site uses Kerberos authentication, then no login is required, and the browser opens SAS Data Loader for Hadoop. Otherwise, you receive the SAS Sign In page, where you enter your user name and password, and then click **Sign In**.

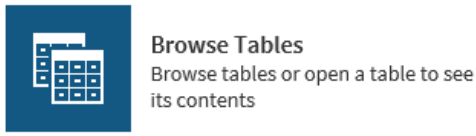
Successful authentication displays your authorized directives in SAS Data Loader for Hadoop. For assistance, contact your SAS administrator.



Take a Quick Tour

Follow these steps to get familiar with SAS Data Loader for Hadoop.

1. To learn about Hadoop tables, click the directive Browse Tables in the directives window.



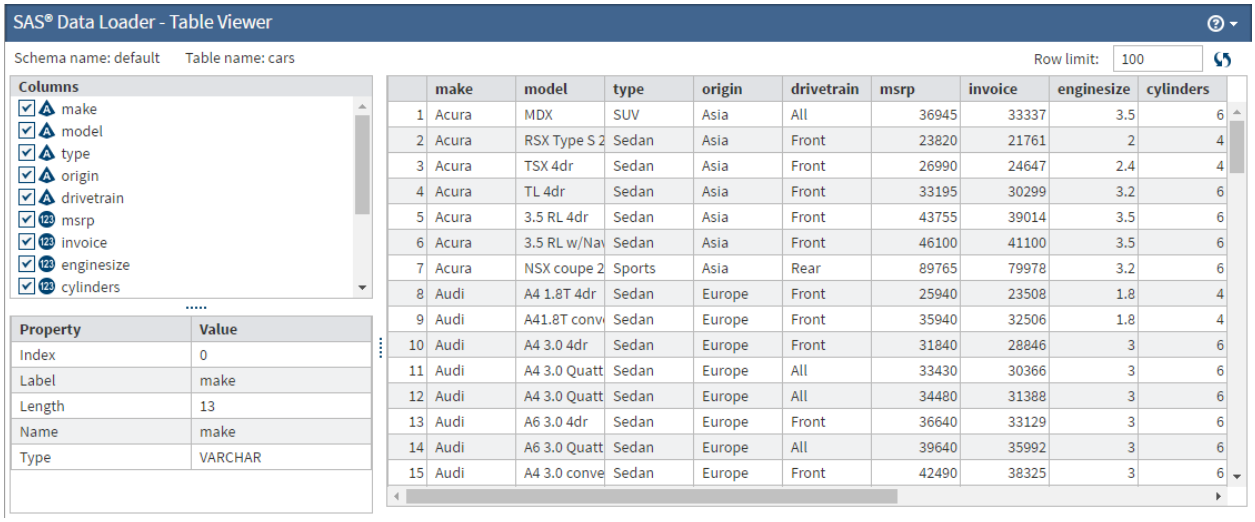
2. The system displays the **Source Table** task, which consists of a graphical list of authorized Hadoop data sources (also known as schemas).

Click the data source of your choice and then click **Next**. The **Source Task** displays a list of all the authorized tables in the data source. If the data source contains no tables, then click Return to Data Sources and open a different data source.

TIP Clicking the web browser's **Back** button generates a message and returns you to the directives page.

3. To look at the data in a Hadoop table, select a data source to display its tables, click a table, and then click **View Table**.

The **Source Task** responds by opening the SAS Table Viewer in a new browser tab. You are now viewing a sample of the data in a Hadoop table.



If you like, you can at this point create and run a directive using any table that is available to you.

4. To learn about user preferences, open the SAS Data Loader for Hadoop browser tab, click , and click **Configuration**.

5. In the Configuration window, the **Haddop Configuration** panel displays server connection information. You can change the system defaults for the SQL environment and the preferred runtime target. If your Hadoop cluster uses Spark or Impala, they will probably be selected by default. You can click **Test Connection** at any time.

6. Click **General Preferences**. The settings in this panel are applied to all of your new directives. In a new directive, you can override these values without changing the default. The field **Maximum length of SAS columns** has an impact on performance and target table size. For more information about this field, see "Change the Maximum Length for SAS Character Columns" on page 154.

7. Click **Profiles** to see the performance-related settings for the Profile Data directive. This directive generates reports on Hadoop tables that contain a wealth of information about the structure and quality of data. You can open profile reports in the Saved Profile Reports directive. You can also open profile reports when selecting source and target tables.

This concludes the quick tour. To learn more about the how to use directives, see Chapter 3, "About the Directive Interface," on page 15.

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 for Hadoop directives can benefit from enhanced performance. Cloudera Impala SQL and Apache Spark use distributed processes and an increased level of in-memory processing.

The following directives support Impala:

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

The following directives support Spark:

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

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. You can override Impala or Spark support as needed in specific directives. Any existing directives that use Hive will continue to run as they have in the past.

The Hadoop Configuration panel of the Configuration window specifies default settings for Impala and Spark. These values are set as system defaults during the deployment of SAS Data Loader for Hadoop. The default settings generally reflect the best performance option for your directives.

You can change the system defaults and override your own defaults in individual directives as needed.

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 Loader for Hadoop Spark Engine on your Hadoop cluster, as described in the *SAS Data Loader for Hadoop: Installation and Configuration Guide*. This book is available from the SAS Data Loader documentation page: http://support.sas.com/documentation/onlinedoc/dmdd

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 Hadoop administrator that Impala has been installed and configured on your Hadoop cluster.
2. Click  and select **Configuration** .
3. In the **Hadoop Configuration** panel, if Impala is configured and if the **Host** and **Port** fields under **Impala server** are blank, then ask your SAS administrator to specify values for those fields.

   If a host and port are specified, then click **Test Connection** to verify that they are operational.
4. In the **SQL environment** field, select **Impala**.
5. Click **OK** to close the Configuration window.

To restore Hive as the default SQL environment, select **Hive** in the **SQL environment** field.

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. The default can be overridden manually in new or existing directives.

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

1. Confirm with your administrator that Spark has been installed and configured on the nodes of your Hadoop cluster.
2. Click  and select **Configuration**.
3. Click **Preferred runtime target** and select **Spark**. Note that if Spark is detected on the Hadoop cluster, then the **Hadoop Spark** value is set by default.
4. Click **OK**.

To restore MapReduce as the default runtime target, click **MapReduce**. .

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.

Usage Notes for Spark

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 Loader for Hadoop Spark Engine. In those files, add or update the value of the option 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 selected as the runtime target, and if you run a Hive release earlier than 1.2, then any date values that are January 1, 1970 or 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 selected as the runtime target, Hive views cannot be used as source tables.

Parquet Cannot Be Specified as a Format for Target Tables

When Spark is selected as the runtime target, the Parquet table format cannot be selected for target tables. The Parquet format can be selected for source tables.

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.

Usage Notes for Impala

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.

Chapter 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 and several categories of user preferences.

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

Note: Access to directives is determined by permissions that were assigned to you by your SAS administrator. To change your permissions, contact your SAS administrator.

Accessing Data Sources, Source Tables, and Target Tables

Overview

SAS Data Loader for Hadoop enables you to browse and select data sources, source tables, and target tables. You can also view sample data in Hadoop tables.

Access to data sources and tables is provided in the following features:

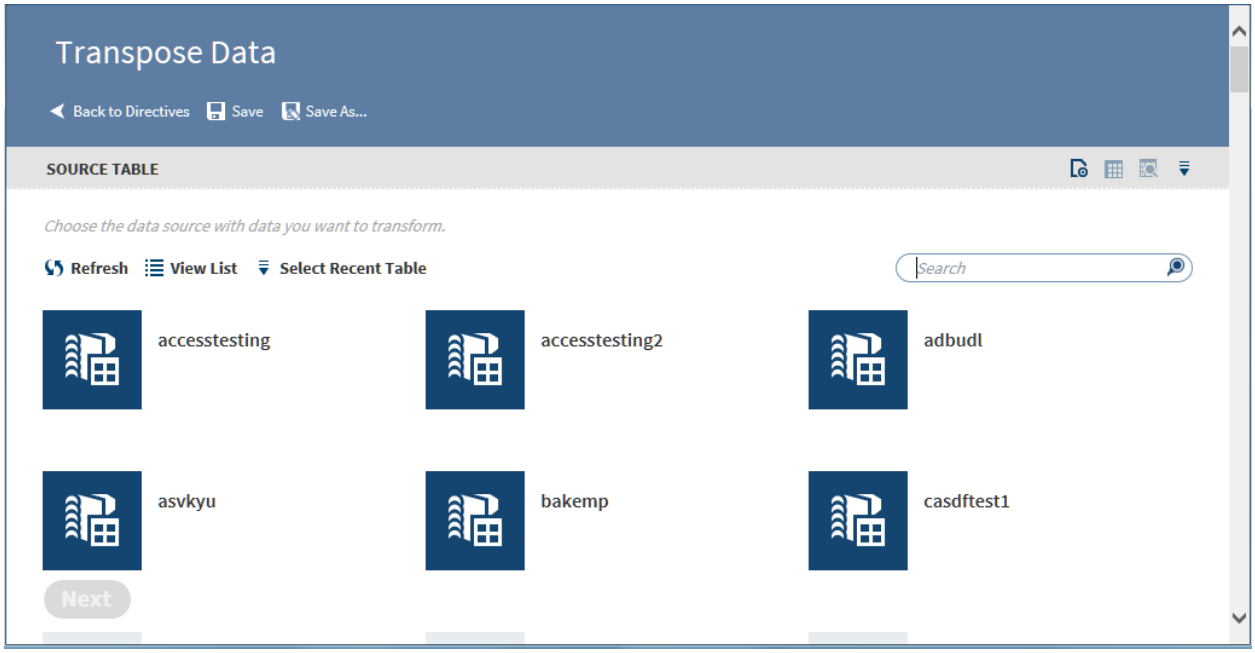
- Browse Tables directive, see "Browse Tables" on page 24.
- Source Table and Target Table tasks in many directives
- SAS Table Viewer
- Sample Data Viewer

Access to data sources and tables is determined by permissions that are managed by your SAS administrator.

About Data Sources and SAS Folders

Data sources are containers of source and target tables. Data sources can be Hadoop schemas, SAS libraries, or database schemas.

Directives that use source tables in Hadoop can display all authorized Hadoop data sources. You can open any of these data sources to select a source in a directive.



When you copy data into and out of Hadoop, the Source Table task and the Target Table task can display database servers, database schemas, SAS libraries, and SAS folders. Depending on your permissions, you can navigate to the highest level, which displays database servers and SAS Libraries.

Opening a database server displays all authorized database schemas. You can select a database server, select a database schema, and then select a database table as your source or target. SAS Data Loader for Hadoop moves data to and from database schemas using JDBC drivers and the Sqoop data transfer tool in Hadoop.

ORACLE_PROD

TIP Database server icons provide indicators of the type of the database, and enable you to test the JDBC connection to the relational database server. Click in the lower left corner to display the Test Connection window.

When you click a **SAS Libraries** icon, the directive displays the SAS libraries that are available to you on that particular SAS Workspace Server. You can select a SAS library, and then select a SAS data set as a source or a target. SAS Data Loader for Hadoop moves data into and out of SAS libraries using SAS/ACCESS.

SAS Libraries
SAS libraries available to the SAS Workspace Server in the SAS Metadata Server

FS_BASE_PROD

TIP To see the LIBNAME statement that is associated with a SAS library, click in the lower left corner of the SAS library icon. The directive responds by displaying details about how SAS connects using that library.

The Import a File directive imports files from SAS libraries into Hadoop. The directive uploads and downloads files from your workstation into SAS libraries.

The Copy Data to Hadoop directive loads SAS data sets from the SAS Workspace Server into Hadoop.

Get Started with the Source Table Task

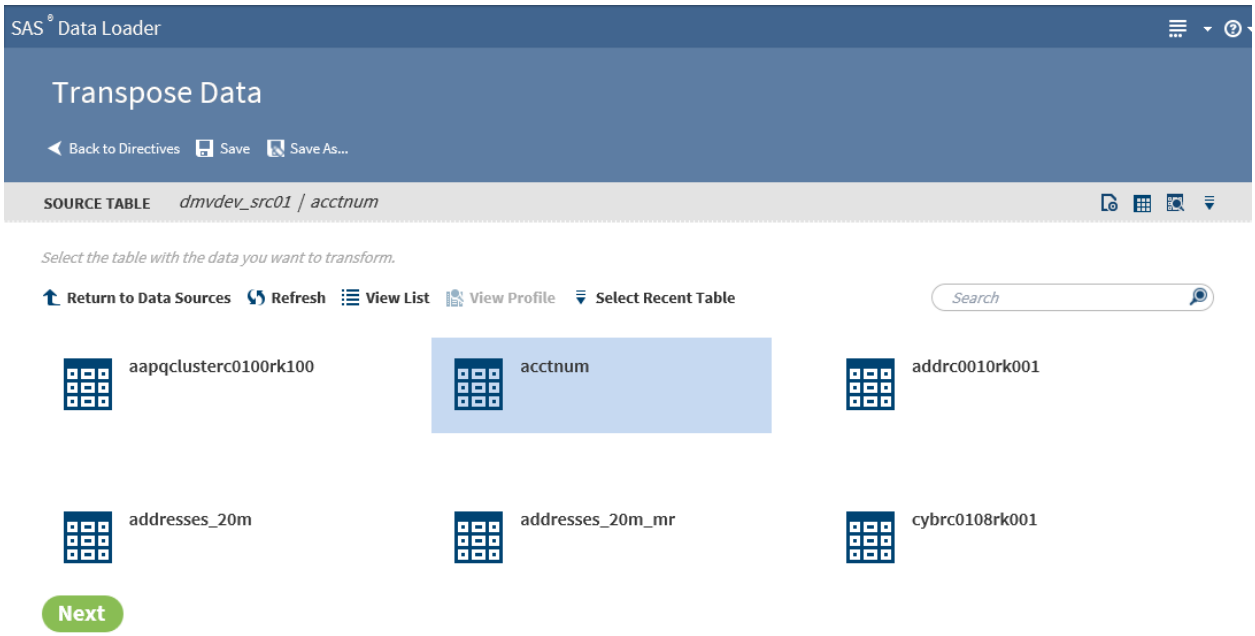
Most of the directives in SAS Data Loader for Hadoop begin with a Source Table task. Use this task to select a source table for your directive. The initial display for Hadoop source tables shows either the tables in the most recently accessed schema, or the available Hadoop data sources. This preference is determined by a setting in the General Preferences panel of the Configuration window  .

Source tables are displayed as icons or as an alphabetical list. Click View List or View Grid. In either view, you can click **Search** and enter part of a table name to select a source table without scrolling. The search feature filters according to name when applied to data sources or to name and description when applied to tables.

To select from a list of recently accessed tables in the current data source, click **Select Recent Table**.

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

Note: If you require access to a data source or a source table that does not appear in the web application, contact your administrator.



The Source Table task includes the following methods of viewing data:

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 a subset of source data, as that data has been transformed up to that point in the job.

View Profile
: click to view profile information for the selected table. If a profile exists for a table, PROFILED appears beneath the table icon in graphical mode or next to the table name in list mode.

If you scroll away from the top of the task, click to return to the top.

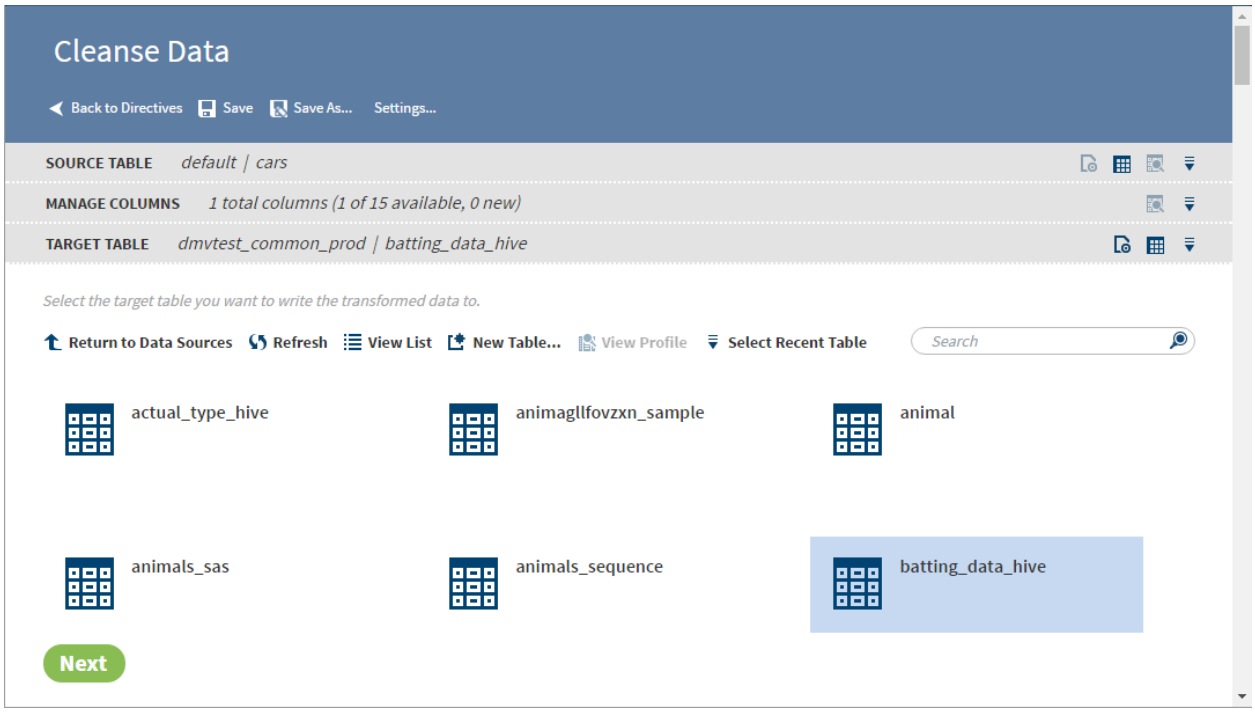
Get Started with the Target Table Task

Directives that include a Target Table task typically display that task toward the end of the directive process, just prior to code generation. The Target Table task displays the available data sources and tables in the directive's target environment. Target environments can be Hadoop, SAS libraries, SAS LASR Analytic Server, or data sources that connect to relational database management servers.

For directives that create or update target tables in Hadoop, the Target Table task displays either the most recent data source or all of the available data sources in Hadoop. Like the Source Table task, the display preference is determined by a setting in the General Preferences panel of the Configuration window .

In the Target Table task, you either select an existing table or click **New Table...** .

To locate a target table, click **Search** and enter part of the table name, or display a list of recently accessed tables in that data source by clicking **Select Recent Table** .



Other icon controls in the Target Table task are defined as follows:

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.

Action Menu icon
: Select from the following:

- **Open** opens the task.
- **Table Viewer** displays the Table Viewer for the selected table.
- **Delete Table** deletes the selected table from the target environment.

The **Delete Table** action is available only for target tables in Hadoop.

Back
: displays data sources for selection.

Refresh
: update display to add new target tables and remove target tables that were deleted by others.

View Profile

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

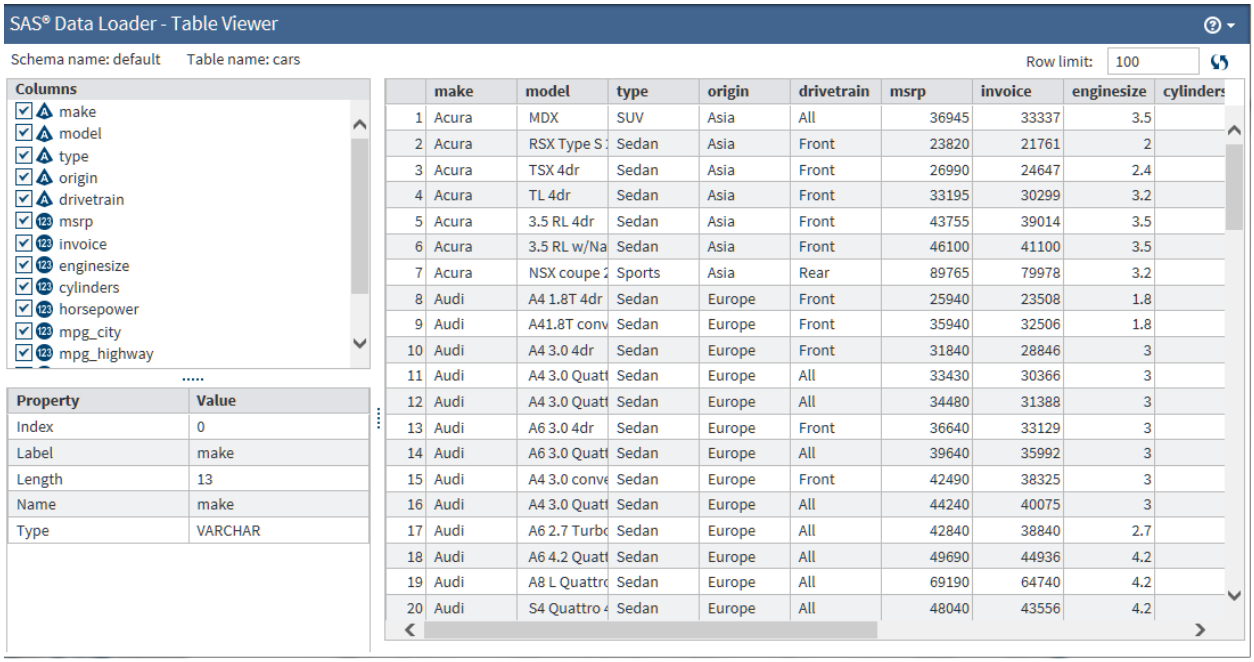
click to return to the top of the page when viewing a long list of target tables.

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 **Refresh** .

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.

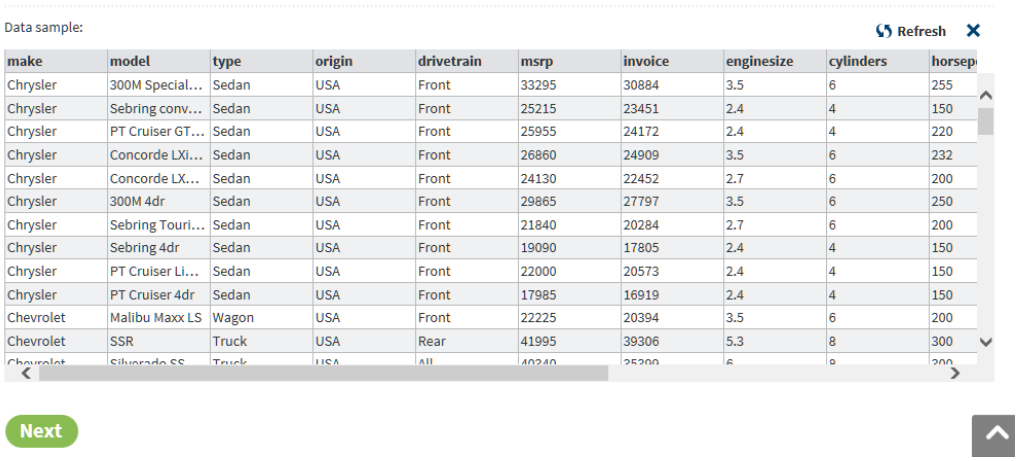
For information about data types and data conversions in SAS and Hadoop, see the chapter SAS/ACCESS Interface to Hadoop in the document *SAS/ACCESS Interface to Relational Databases: Reference*.

Usage Note

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, the values are the same.

About the Sample Table Viewer

In directives that transform data, you can display sample data by clicking **View a data sample** icon . The directive responds by displaying a subset of the source data, as that data has been transformed up to that point in the job. Viewing this sample data gives you a preview of your output before you run your job against the full source table in Hadoop.



make	model	type	origin	drivetrain	msrp	invoice	enginesize	cylinders	horsep
Chrysler	300M Special...	Sedan	USA	Front	33295	30884	3.5	6	255
Chrysler	Sebring conv...	Sedan	USA	Front	25215	23451	2.4	4	150
Chrysler	PT Cruiser GT...	Sedan	USA	Front	25955	24172	2.4	4	220
Chrysler	Concorde LXi...	Sedan	USA	Front	26860	24909	3.5	6	232
Chrysler	Concorde LX...	Sedan	USA	Front	24130	22452	2.7	6	200
Chrysler	300M 4dr	Sedan	USA	Front	29865	27797	3.5	6	250
Chrysler	Sebring Touri...	Sedan	USA	Front	21840	20284	2.7	6	200
Chrysler	Sebring 4dr	Sedan	USA	Front	19090	17805	2.4	4	150
Chrysler	PT Cruiser Li...	Sedan	USA	Front	22000	20573	2.4	4	150
Chrysler	PT Cruiser 4dr	Sedan	USA	Front	17985	16919	2.4	4	150
Chevrolet	Malibu Maxx LS	Wagon	USA	Front	22225	20394	3.5	6	200
Chevrolet	SSR	Truck	USA	Rear	41995	39306	5.3	8	300

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.

Hadoop supports a number of programming languages. SAS Data Loader for Hadoop generates code that is optimized for the task that is being performed. Not all code will be displayed in the code editor.

Code should be modified only by power users for advanced purposes only.

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 125.

Chapter 4

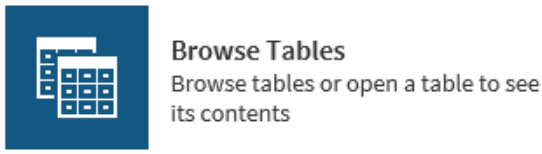
Manage Data

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. Additional directives delete tables from Hadoop, or browse available tables in Hadoop.

Browse Tables

Introduction

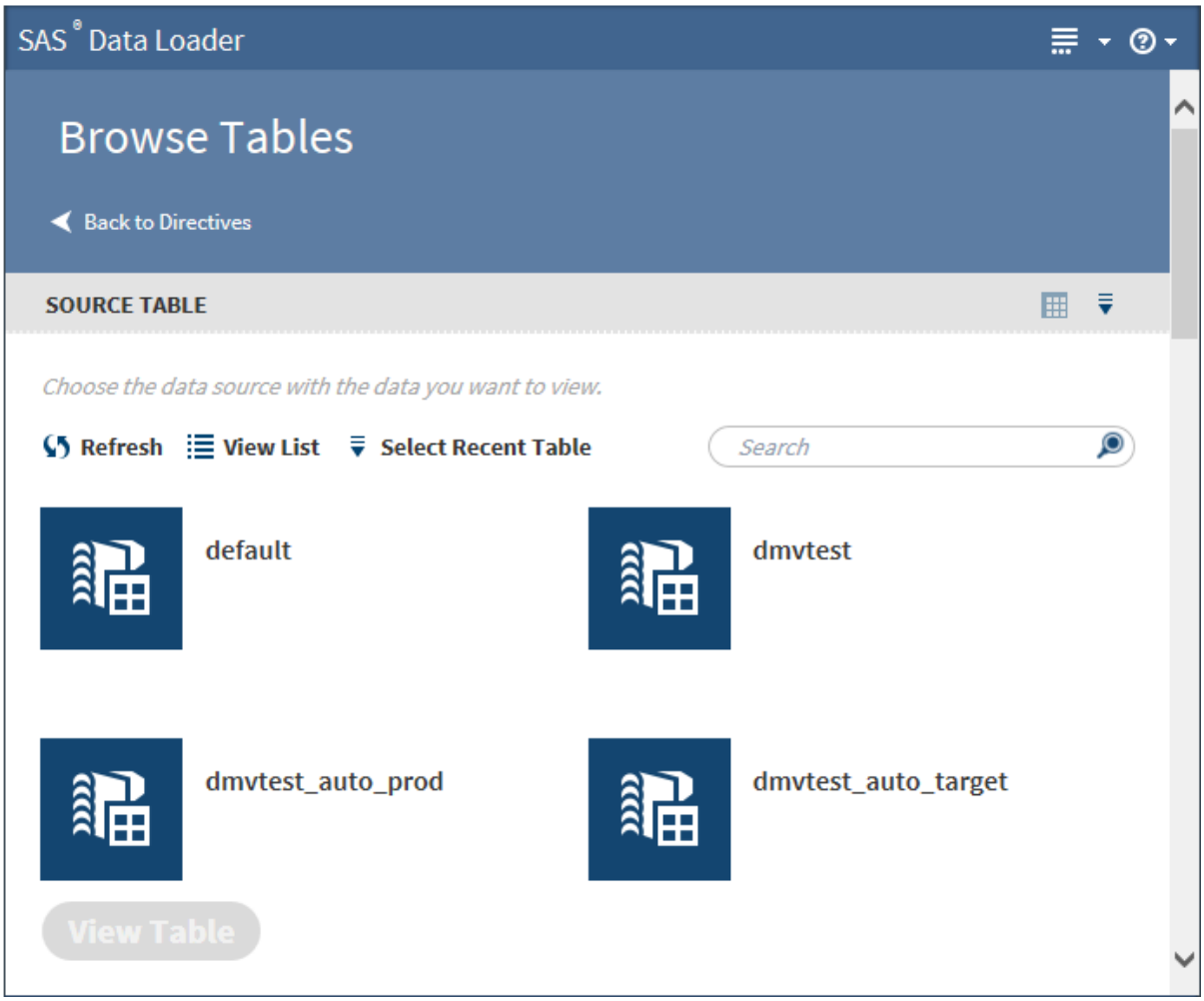


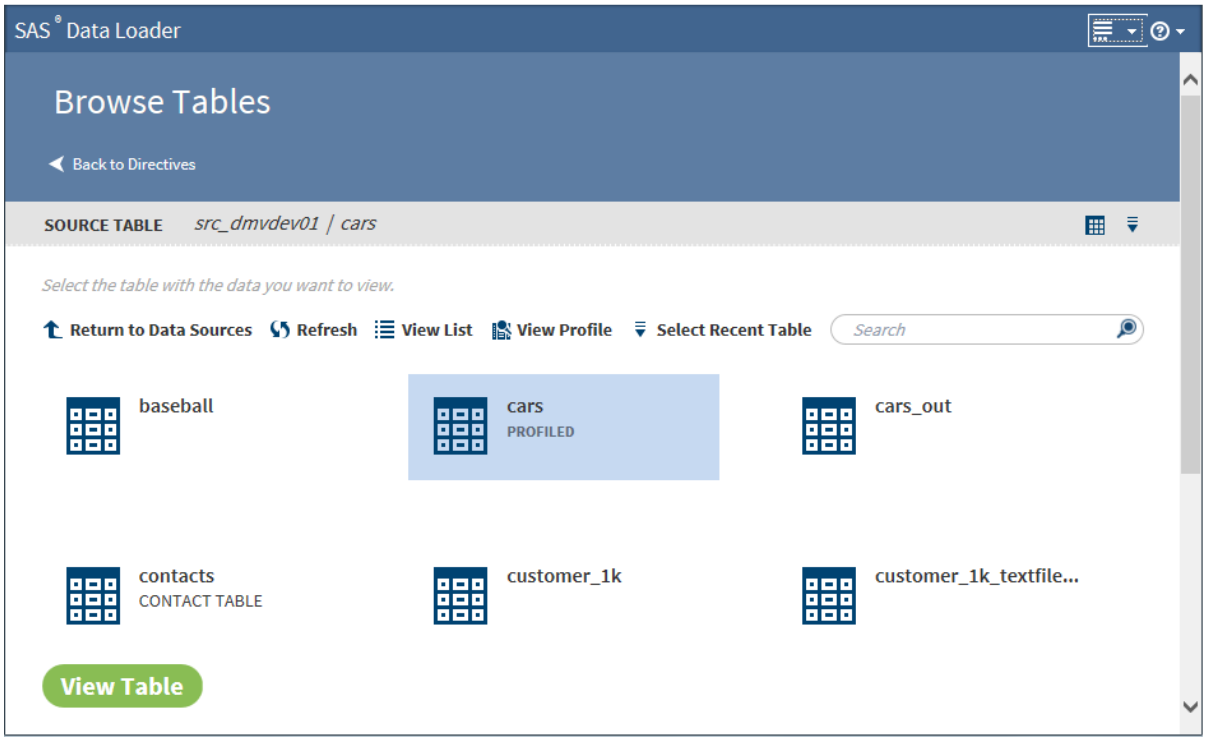
Use the Browse Tables directive to examine the accessible data sources, tables, and table data in Hadoop.

Example

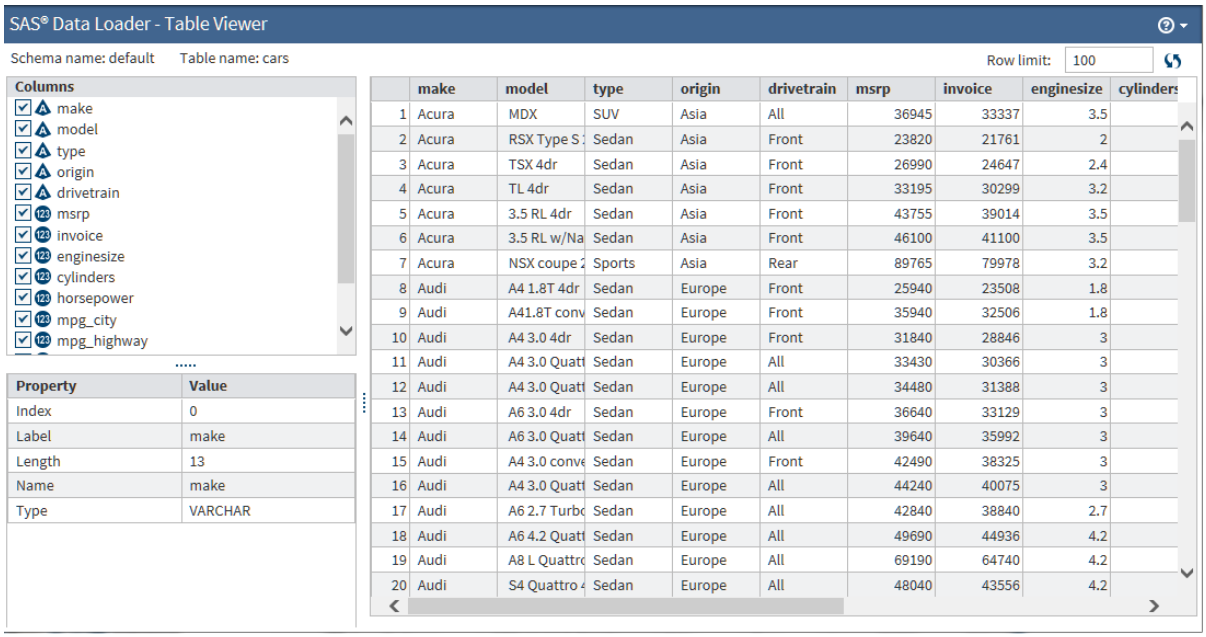
Follow these steps to use the Browse Data directive:

1. On the SAS Data Loader for Hadoop directives page, click **Browse Tables**. The **Source Table** task is displayed. Click a data source to display its tables.


SAS® Data Loader
Browse Tables
Back to Directives
SOURCE TABLE
Choose the data source with the data you want to view.
Refresh
View List
Select Recent Table
Search
default
dmvtest
dmvtest_auto_prod
dmvtest_auto_target
View Table


SAS® Data Loader
Browse Tables
Back to Directives
SOURCE TABLE
src_dmvdev01 / cars
Select the table with the data you want to view.
Return to Data Sources
Refresh
View List
View Profile
Select Recent Table
Search
baseball
cars
PROFILED
cars_out
contacts
CONTACT TABLE
customer_1k
customer_1k_textfile...
View Table

2. Click a source table and then click **View Table**. The SAS Table Viewer is displayed in a new browser tab.



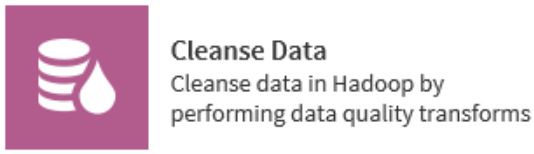
	make	model	type	origin	drivetrain	msrp	invoice	enginesize	cylinders
1	Acura	MDX	SUV	Asia	All	36945	33337	3.5	
2	Acura	RSX Type S	Sedan	Asia	Front	23820	21761	2	
3	Acura	TSX 4dr	Sedan	Asia	Front	26990	24647	2.4	
4	Acura	TL 4dr	Sedan	Asia	Front	33195	30299	3.2	
5	Acura	3.5 RL 4dr	Sedan	Asia	Front	43755	39014	3.5	
6	Acura	3.5 RL w/Na	Sedan	Asia	Front	46100	41100	3.5	
7	Acura	NSX coupe 2	Sports	Asia	Rear	89765	79978	3.2	
8	Audi	A4 1.8T 4dr	Sedan	Europe	Front	25940	23508	1.8	
9	Audi	A41.8T conv	Sedan	Europe	Front	35940	32506	1.8	
10	Audi	A4 3.0 4dr	Sedan	Europe	Front	31840	28846	3	
11	Audi	A4 3.0 Quatt	Sedan	Europe	All	33430	30366	3	
12	Audi	A4 3.0 Quatt	Sedan	Europe	All	34480	31388	3	
13	Audi	A6 3.0 4dr	Sedan	Europe	Front	36640	33129	3	
14	Audi	A6 3.0 Quatt	Sedan	Europe	All	39640	35992	3	
15	Audi	A4 3.0 conve	Sedan	Europe	Front	42490	38325	3	
16	Audi	A4 3.0 Quatt	Sedan	Europe	All	44240	40075	3	
17	Audi	A6 2.7 Turbo	Sedan	Europe	All	42840	38840	2.7	
18	Audi	A6 4.2 Quatt	Sedan	Europe	All	49690	44936	4.2	
19	Audi	A8 L Quattro	Sedan	Europe	All	69190	64740	4.2	
20	Audi	S4 Quattro 4	Sedan	Europe	All	48040	43556	4.2	

TIP Because the SAS Table Viewer appears in a separate browser tab, you can compare the data in multiple tables.

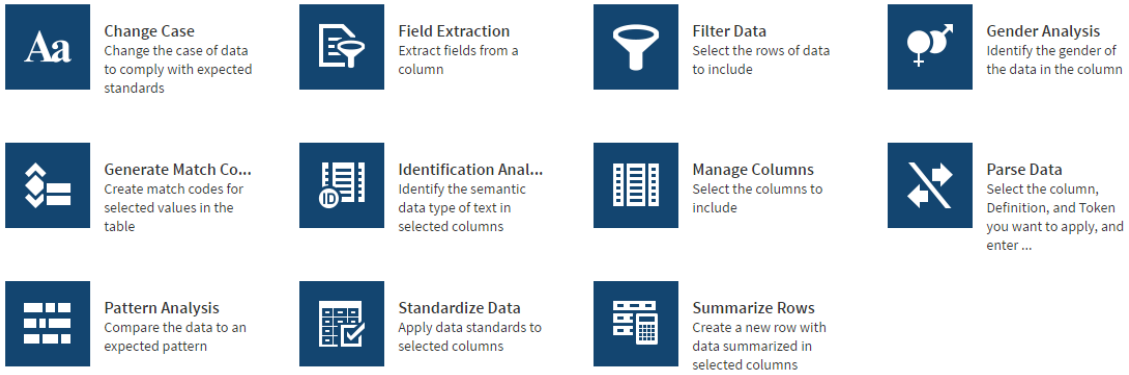
To learn more about viewing table data, see " About the SAS Table Viewer" on page 20.

Cleanse Data

Introduction



Use the Cleanse Data directive to improve the quality of your Hadoop data. Your directives can combine any of the data quality transformations in any order. When you run your directive, your combination of the following transformations are executed in the order in which you define them.



About Locales, Definitions, and the Quality Knowledge Base

Most of the data quality transformations apply a *QKB locale* and a *definition* to a selected source column. Definitions define data formats for specific types of content and types of data cleansing. For example, a parse definition for a street address describes how a street address can be parsed into identifiable segments.

QKB locales are collections of definitions that apply to a language or alphabet and a geographic region. The locale English (United States) contains definitions in the English language, as that language is used in the United States. Definitions in the locale English (England) accommodate England's different use of the same language. Examples of definitions that differ between the two locales include postal addresses and phone numbers.

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

To use the QKB in SAS Data Loader for Hadoop, administrators publish QKB metadata from Hadoop into the SAS middle tier. The QKB publish operation is required after initial deployment, and after the QKB has been updated or customized.

Administrators also specify a system default QKB locale, which is applied to all new directives that include data quality transformations. The system default QKB locale is displayed as your local default QKB locale when you first open the **General Preferences** panel of the Configuration window. You can change your local default QKB locale at any time. Within a given data cleansing transformation, you can specify a QKB locale that differs from your default, without changing that default.

To learn more about publishing the QKB and setting the system default QKB locale, see the SAS Data Loader for Hadoop: Installation and Configuration Guide.

To learn more about the SAS Quality Knowledge Base, including customization, refer to the related document titles in "Recommended Reading" on page 159.

Select a Source Table

When you use a data cleansing directive, you begin by selecting a source table.

Follow these steps to select a source table:

1. In the SAS Data Loader for Hadoop directives page, select **Cleanse Data**. The Cleanse Data directive opens at the **Source Table** task.

2. In the **Source Table** task, click the data source that contains your source table. Or you can click **Select a Recent Table** and choose a source table from that list.
3. In the selected 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".
4. In the **Transformation** task, click a data cleansing transformation to begin building your directive.

Select a Data Cleansing Transformation

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

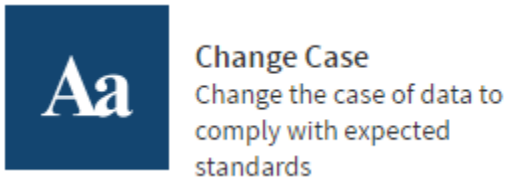
- "Change Case Transformation" on page 28.
- "Field Extraction Transformation" on page 29.
- "Filter Data Transformation" on page 30.
- "Gender Analysis Transformation" on page 33.
- "Generate Match Codes Transformation" on page 34.
- "Identification Analysis Transformation" on page 35.
- "Manage Columns Transformation" on page 36.
- "Parse Data Transformation" on page 38.
- "Pattern Analysis Transformation" on page 38.
- "Standardization Transformation" on page 40.
- "Summarize Rows Transformation" on page 41.

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).

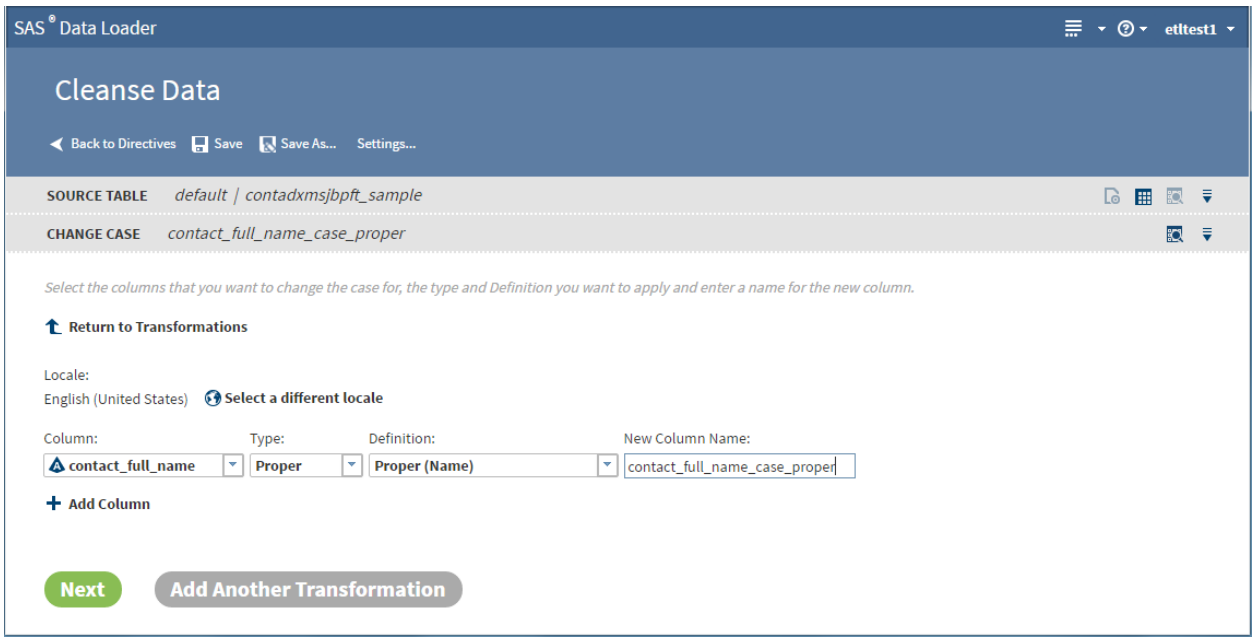
Follow these steps to use the Change Case transformation:

1. If this is the first transformation in a new directive, 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 QKB locale needs to reflect the language and region that applies to the content in the source table.
4. Click to select a source column, a type of casing, and the QKB definition that applies to the source data and to the desired casing. The example below shows the selection of proper casing for a column of full names, using the case definition **Proper (Name)**.
5. Accept or change the suggested target column name in the field **New Column Name**.
6. To change the casing of another column, click **Add Column**.
7. Click **Next** to select a target table and run your directive.

   To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 28.

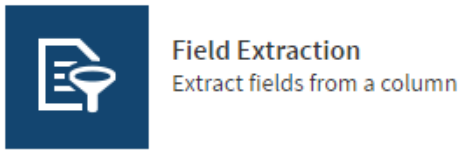


Field Extraction Transformation

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.

Follow these steps to use the Field Extraction transformation:

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

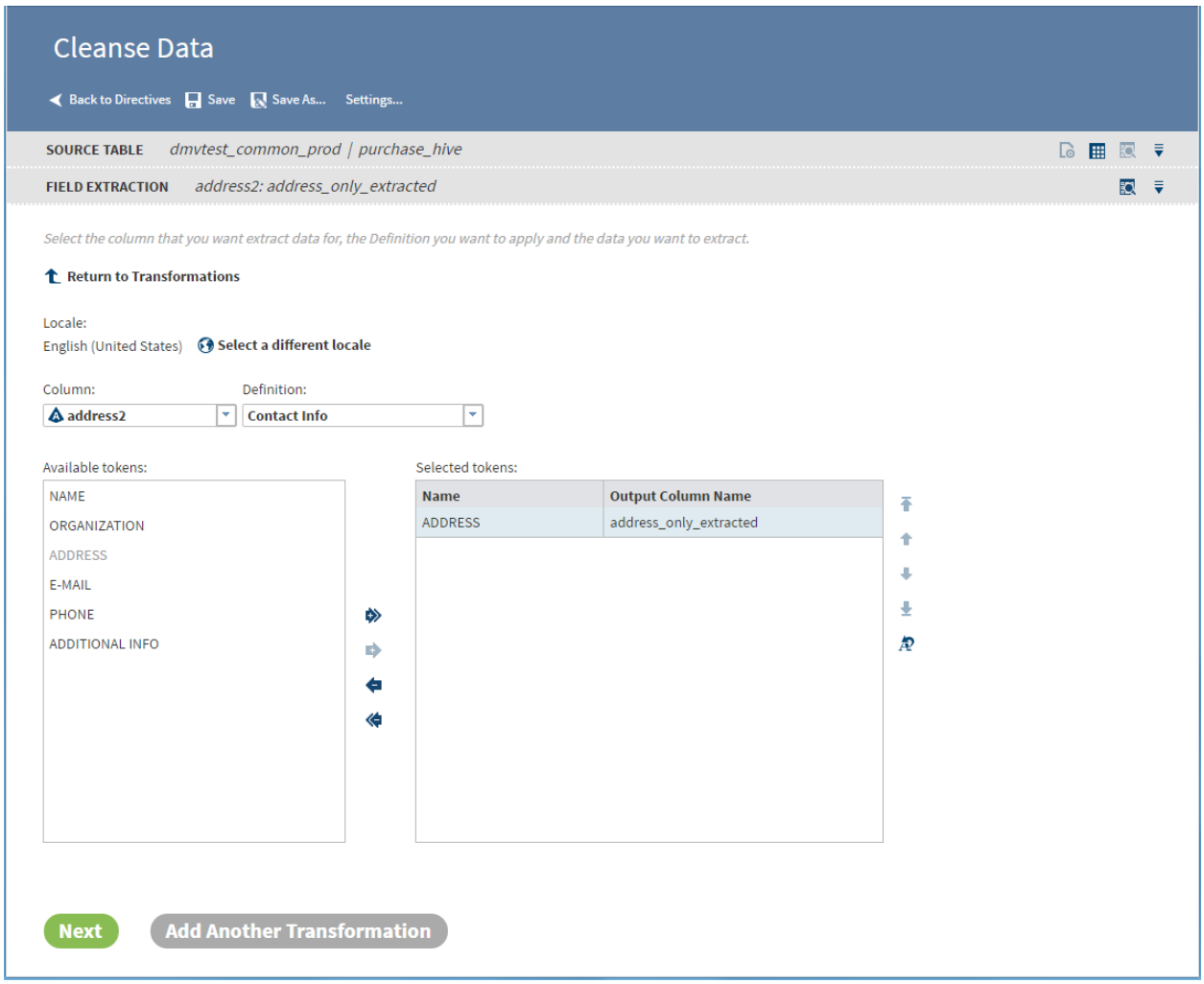


3. In the **Field Extraction** transformation, accept or change the default **Locale**.

4. Select a **Column** and a **Definition**.
5. Select one or more **Available tokens** and click or .

   The tokens that you select are used to parse each source row and extract values of the specified type.
6. To change the default target column name, double-click in **Output Column Name**.
7. Click **Next** to select a target table and run your directive.

   To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 28.

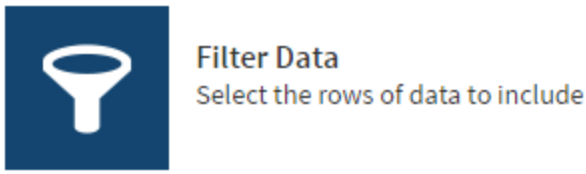


Filter Data Transformation

Use the Filter Data transformation at the beginning of a directive to decrease the number of rows that will be processed in subsequent transformations. Filters are specified either as one or more rules or user-written expressions. If the rules or expressions are true, then the row is written into the target table.

Follow these steps to use the Filter Data transformation:

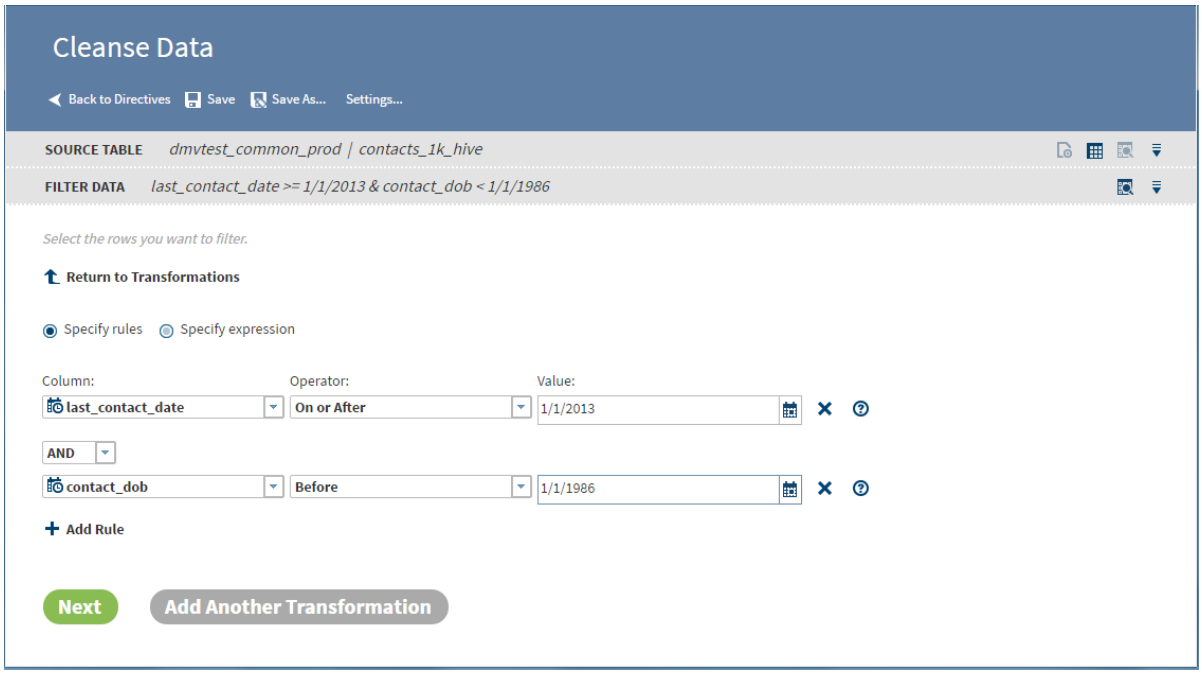
1. If this is the first transformation in a new directive, select a source table.
2. In the **Transformation** task, click **Filter Data**.



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 Directive" on page 90.

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 selected source column. For example, a column of datetime data uses operators such as Before and After. A numeric column uses operators such as Greater Than or Less Than.
   c. In the **Value** field, add the source column value that completes the expression.
   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**. The following expression reads as "Keep only those rows for which the last contact was in 2013 or later, and the contact was born before 1986."



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

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

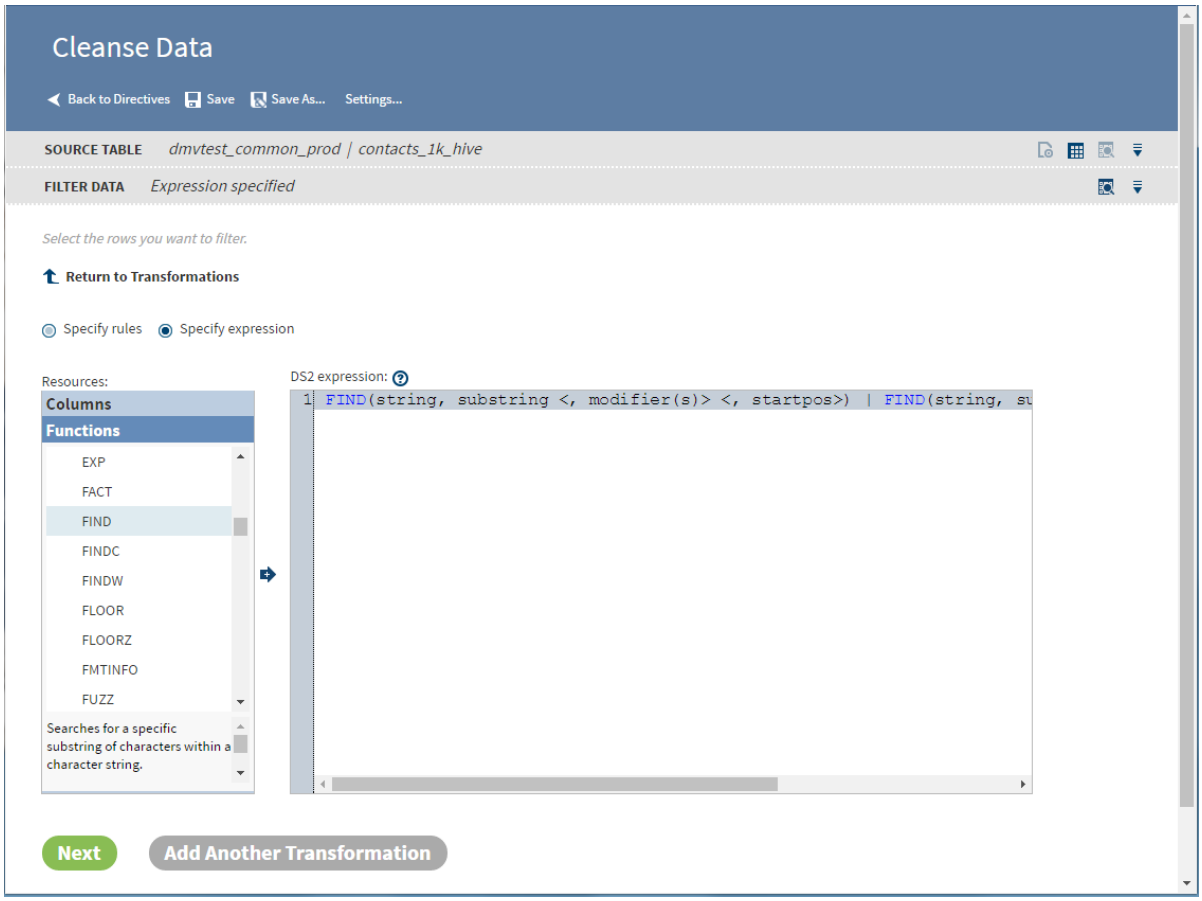
   a. In the expression text box, enter or paste your expression.

      Your expression can use either DS2 functions (with the MapReduce runtime target) or EEL functions (with the Spark runtime target). To view or change the runtime target for the current directive, click **Settings**.

      To learn more about runtime targets, see "Enable Support for Impala and Spark".

      To learn the requirements for expressions, see "Develop Expressions for Directives" . See also *SAS DS2 Language Reference* or *Expression Language: Reference Guide*.

   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 directive.

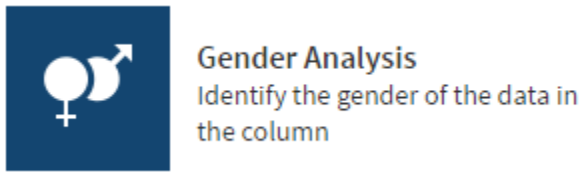
   To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 28.

Gender Analysis Transformation

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

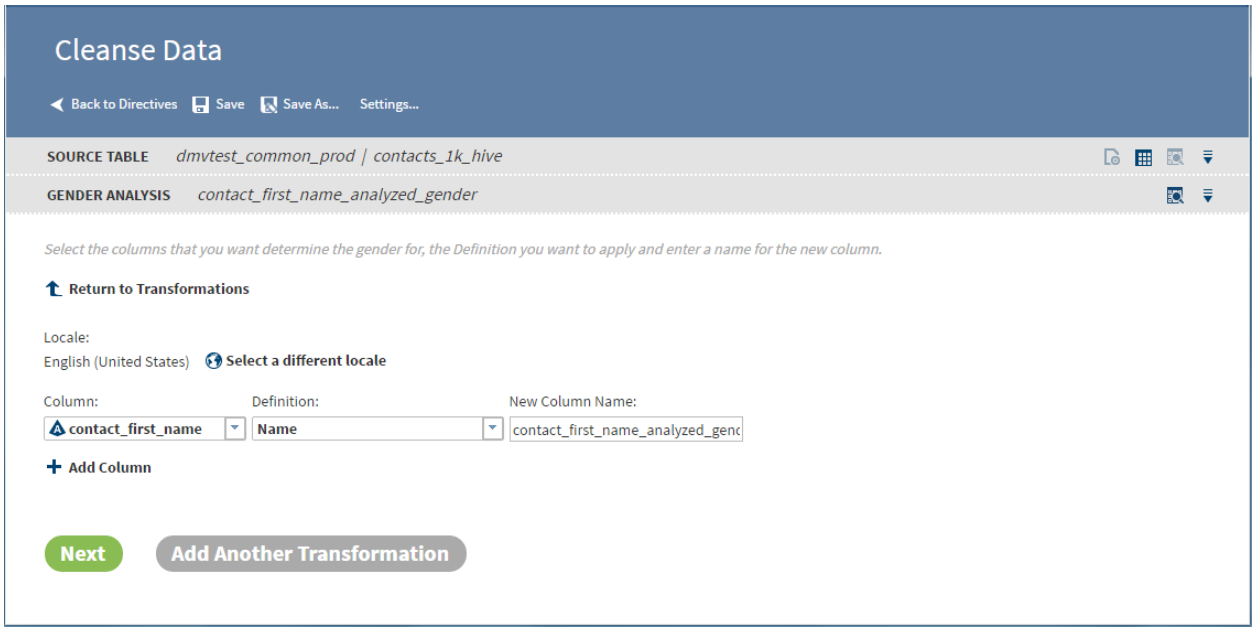
Follow these steps to use the Gender Analysis transformation:

1. If this is the first transformation in your directive, 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 suggested **New Column Name**.
8. Click **Next** to select a target table and run your directive.

   To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 28.



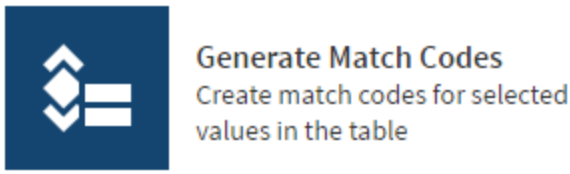
Generate Match Codes Transformation

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 QKB 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.

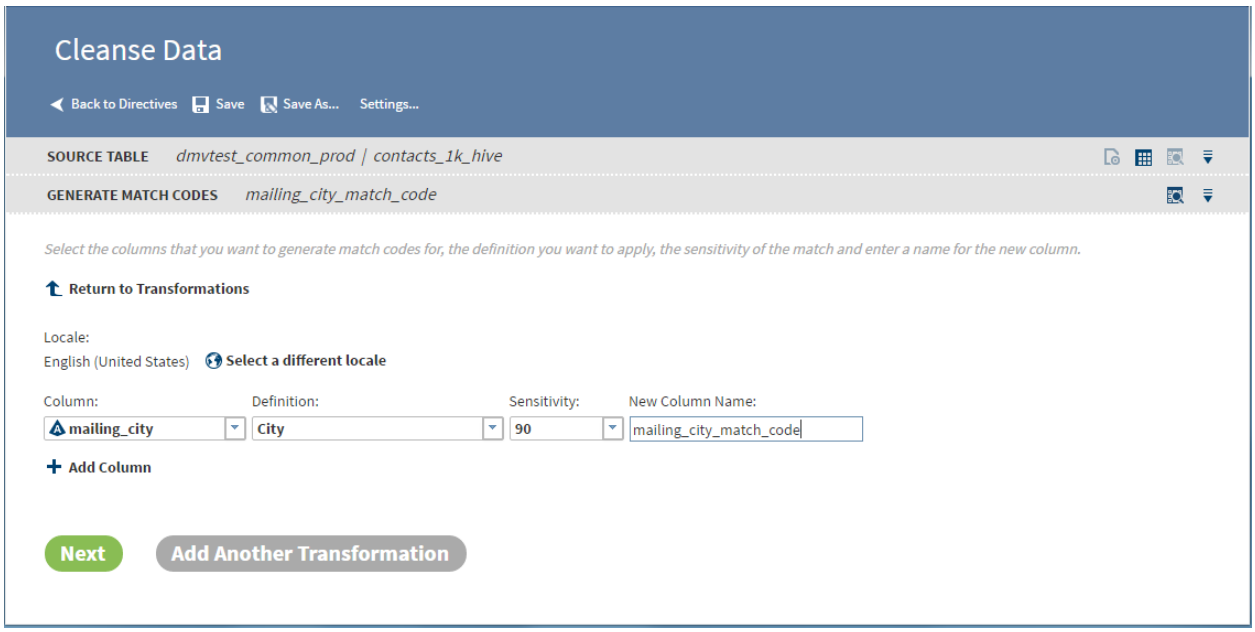
Follow these steps to use the Generate Match Codes transformation:

1. If this is the first transformation in your directive, 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 (and more identical match codes) and perhaps more matching errors. Higher sensitivity numbers produce identical match codes only when data values are very similar.
6. Click **Next** to select a target table and run your directive.

   To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 28.

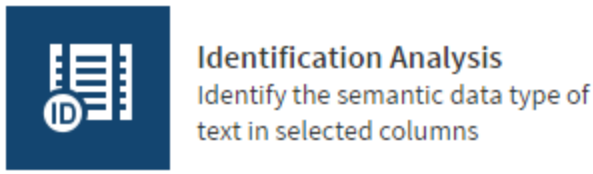


Identification Analysis Transformation

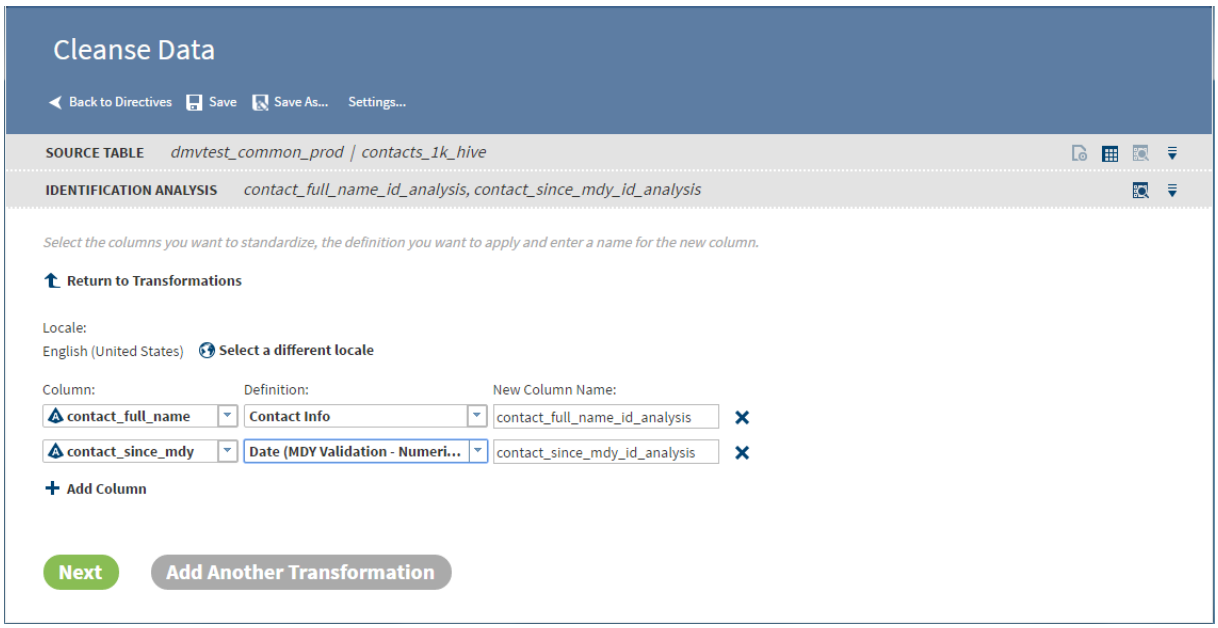
Use the Identification Analysis transformation to report on the type of the content in a given column. The content types that can be detected include contact information, dates, email, field names, offensive content, and phone numbers. The result of the analysis is added to a new column in the target 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 directive, 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, change the suggested name of the target column as needed.
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 directive.

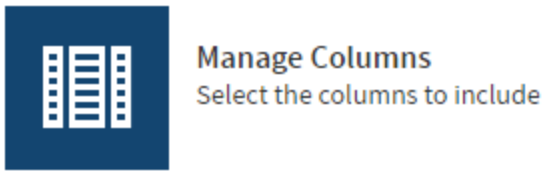
   To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 28.

Manage Columns Transformation

Use the Manage Columns transformation to support column mapping features such as remove, reorder, rename, change type, and change length. You can also apply user-written expressions to generate new data in new columns, or modify or replace data in existing columns.

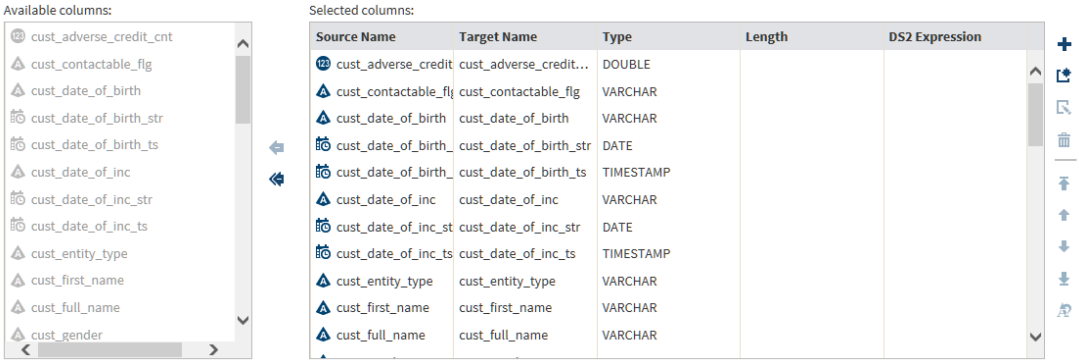
Follow these steps to learn how to use the Manage Columns transformation:

1. If this is the first transformation in your directive, select a source table.
2. In the **Transformation** task, click **Manage Columns**.



3. Select, arrange, and rename the columns in the target table.

   In **Available columns**, columns are initially listed in order of appearance in the source table. The top column in the list is the first or leftmost column in the source table. The same presentation is used for **Selected columns**.



   To remove one or more columns from the target table, select the columns and click .

   To remove all columns from the target, and then add a smaller number of selected columns, click , select available columns, and then click

   To change the order of columns, select columns and click , , , or .

   To rename a column, select it and click .

4. Add new columns and specify expressions.

   To add a new target column and name it, click .

   To quickly enter or paste an expression into a new or existing column, click the **Expression** cell for that column.

   To add a new column and compose an expression in the Advanced Editor, click .

   To use the Advanced Editor to add an expression to an existing column, select that column and click .

   To learn about the Advanced Editor, see "About Expressions and the Advanced Editor" on page 43.

   Your expression can use either DS2 functions (with the MapReduce runtime target) or EEL functions (with the Spark runtime target). To view or change the runtime target for the current directive, click **Settings**.

   To learn more about runtime targets, see "Enable Support for Impala and Spark".

   To learn the requirements for expressions, see "Develop Expressions for Directives" . See also *SAS DS2 Language Reference* or *Expression Language: Reference Guide*.

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

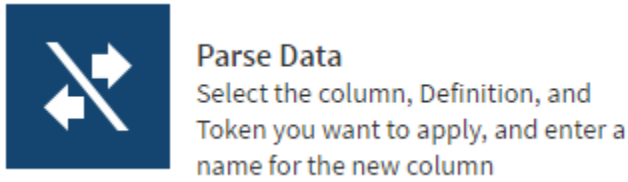
To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 28.

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 directive, 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 directive.

    To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 28.

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 directive, 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 Lllllllll*`**アアアアアアア.

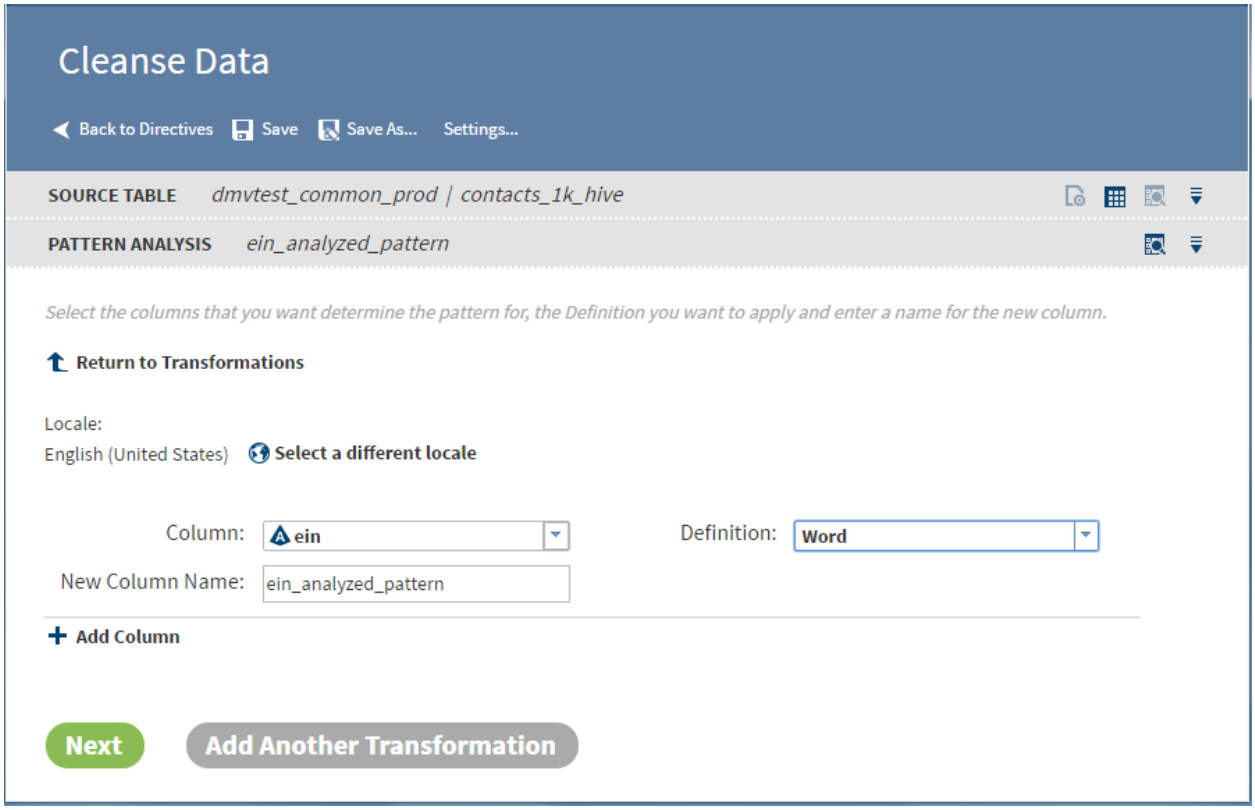
   Note: The full mapping of pattern characters to Unicode character sets is provided in the Pattern Analysis Definitions in the online Help for the Contact Information Quality Knowledge Base.

   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 directive.

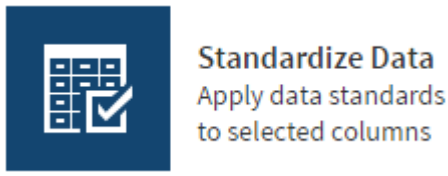
   To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 28.



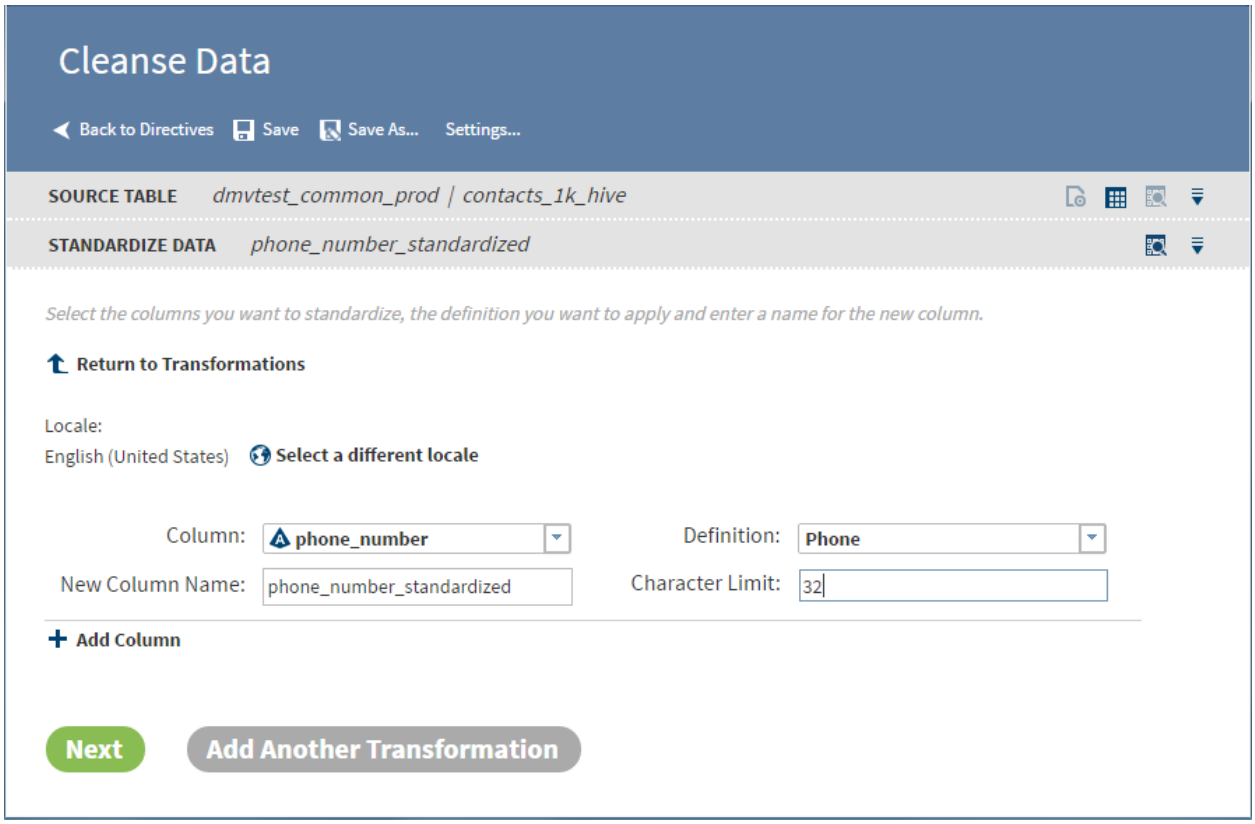
Standardization Transformation

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

1. If this is the first transformation in your directive, 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" transformation toward the end of your directive.

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

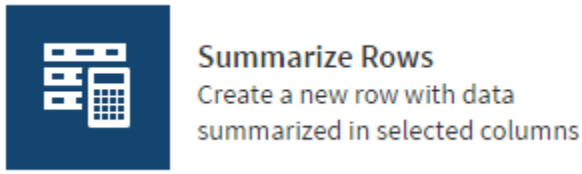
   To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 28.

Summarize Rows Transformation

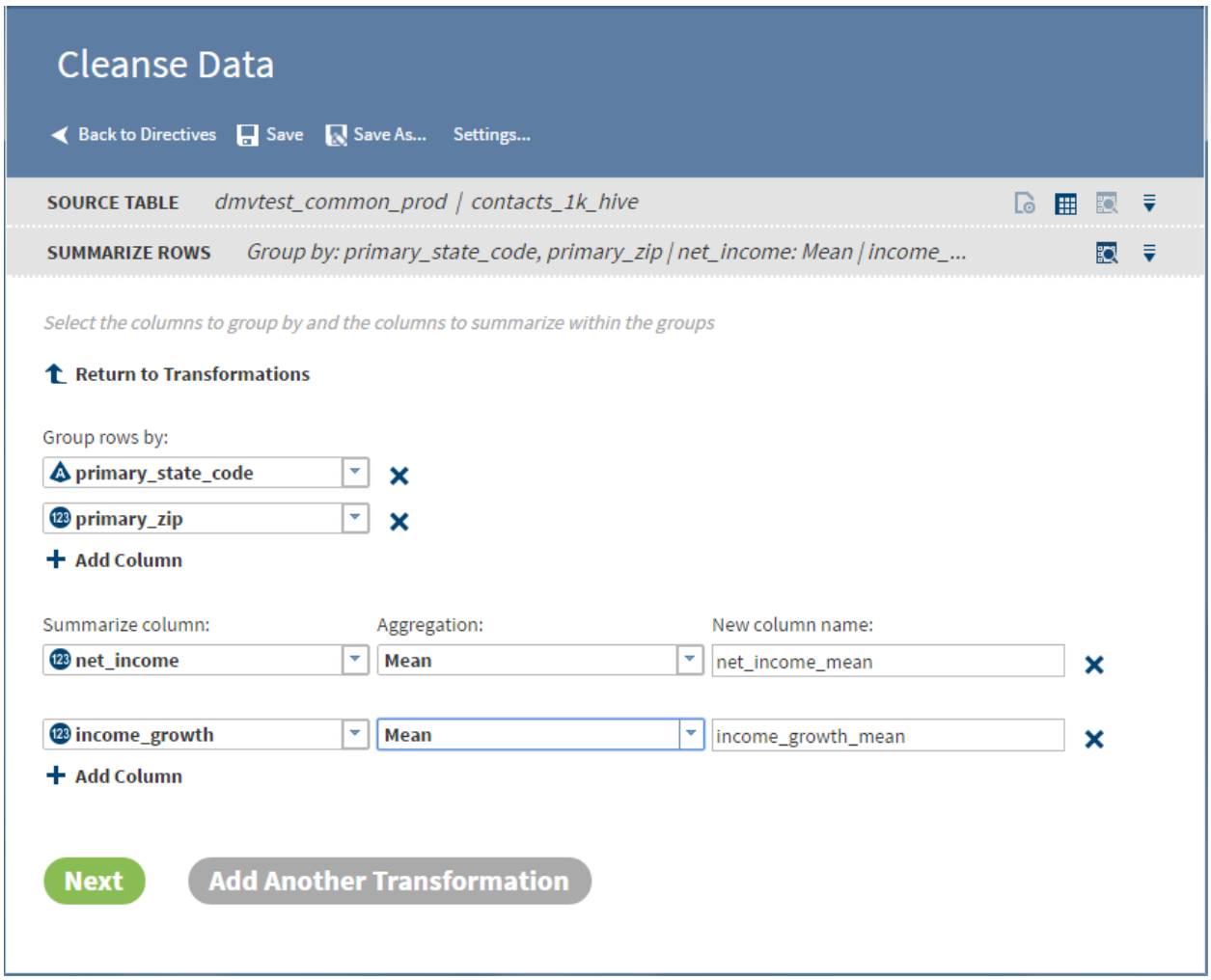
Use the Summarize Rows transformation to add summarized numeric values to your target table. To generate summaries, you first group rows by one or more columns. Then you select the columns that you want to summarize for each group and subgroup. The method of summarization is known as an aggregation. 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 directive, 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 directive.

   To add another data cleansing transformation, click **Add Another Transformation** and see "Select a Data Cleansing Transformation" on page 28.

Select a Target Table and Run Your Directive

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

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 learn more about browsing tables and viewing data, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16.

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

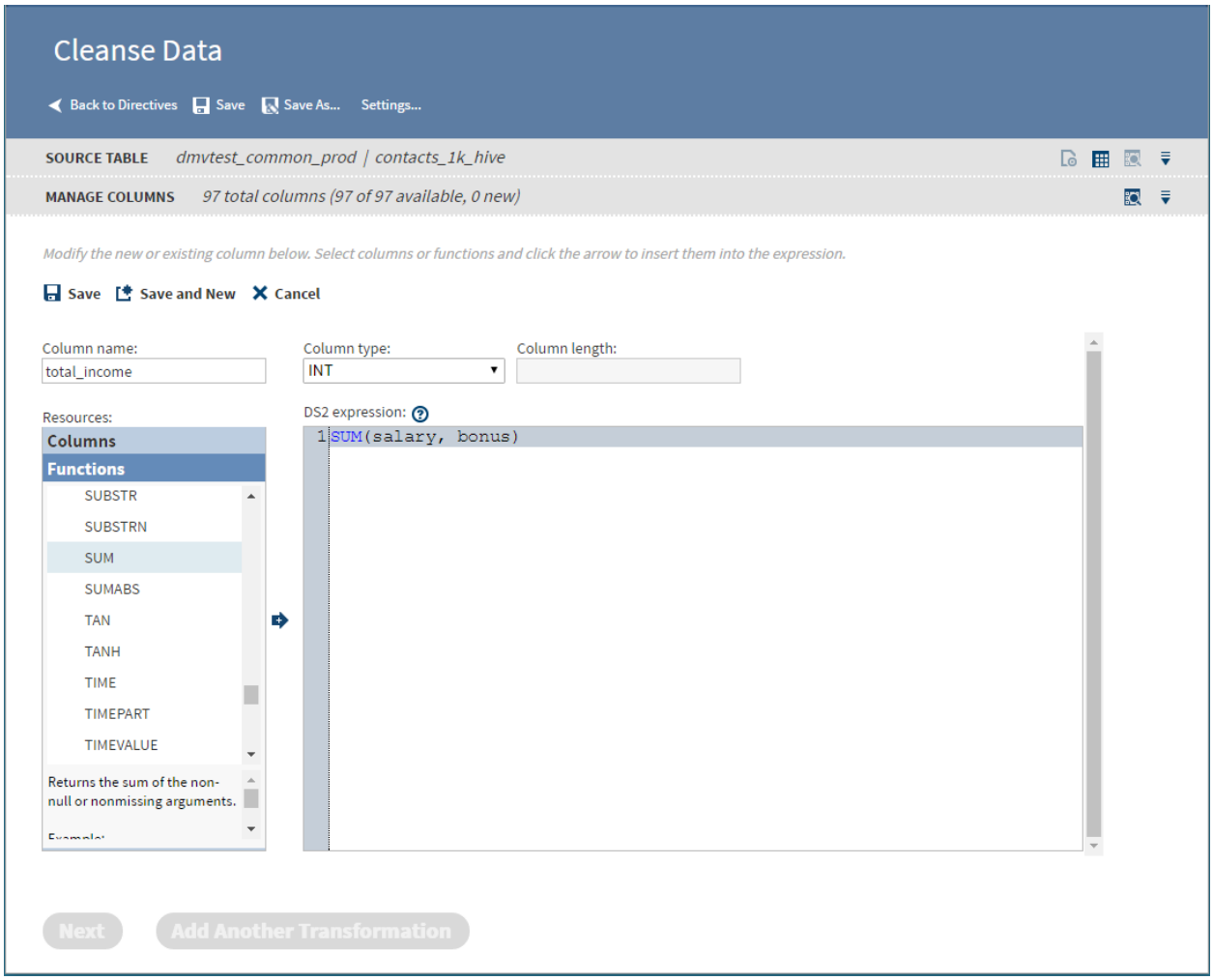
2. In the **Result** task, click **Save** or **Save As** to save your directive, and then click **Start Transforming Data**.

3. During the execution of the directive, you can view the code, log, and results as they are generated.

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 of the Cleanse Data directive.

Expressions can use DS2 functions when you select the MapReduce runtime target or EEL functions when you select the Spark runtime target. Click **Settings** to display the selected runtime target. For more information about runtime targets, see "Enable Support for Impala and Spark".



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

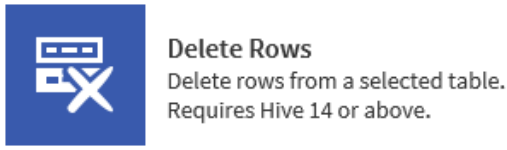
The **Resources** list box displays the categories of the available DS2 or EEL functions. Within the categories, selecting a function displays syntax help. To add a function to your expression, click the listing and click .

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" .

Delete Rows

Introduction



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:

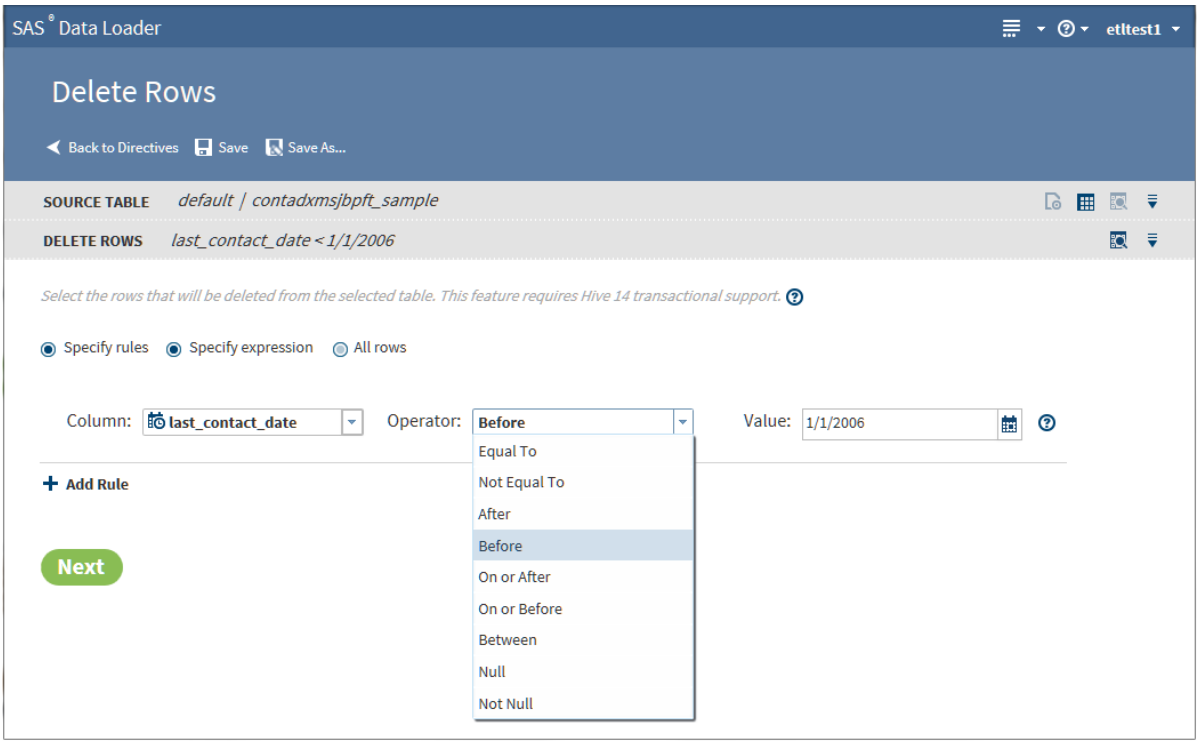
- The Hadoop cluster needs to be configured with release 0.14.0 or later of the Apache Hive data warehouse software. This release supports transactional tables.
- Source tables must use a Hive file format, preferably ORC (Optimized Row Columnar.)
- Source tables must be bucketed and partitioned. The bucketing process creates clusters of data based on the values in a specified (key) column. The partitioning process 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 for Hadoop directives page, click **Delete Rows**. The **Source Table** task is displayed.
2. In the **Source Table** task, select a source table and click **Next**, or click Select Recent Table to select a table from a different data source.
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 47. 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. All rows for which the rule is true will be deleted from the table.

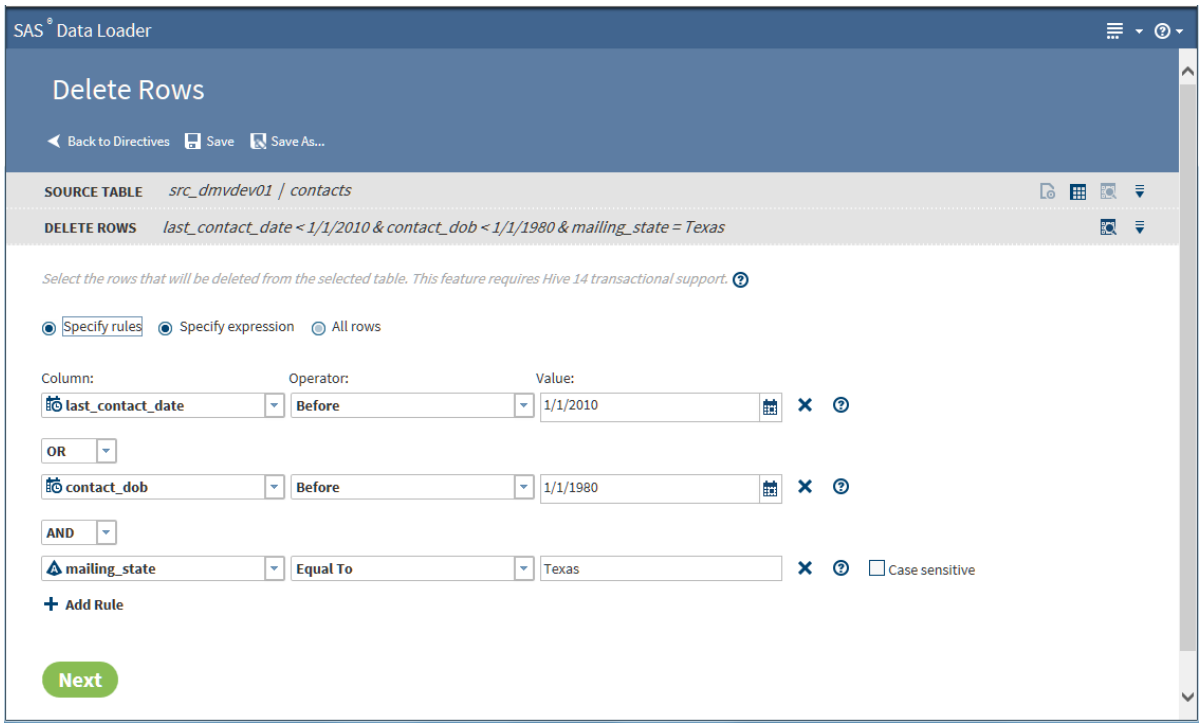
b. Click and select a logical **Operator**. The operators that are available depend on the type of the data in the source column. For example, the following image shows the operators that are available for the date/time data type:



c. In the **Value** field, add the source column data value that completes the rule. In the preceding example, the rule can be read as "Delete from the table all rows with a last contact date prior to January 1, 2006."

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**.



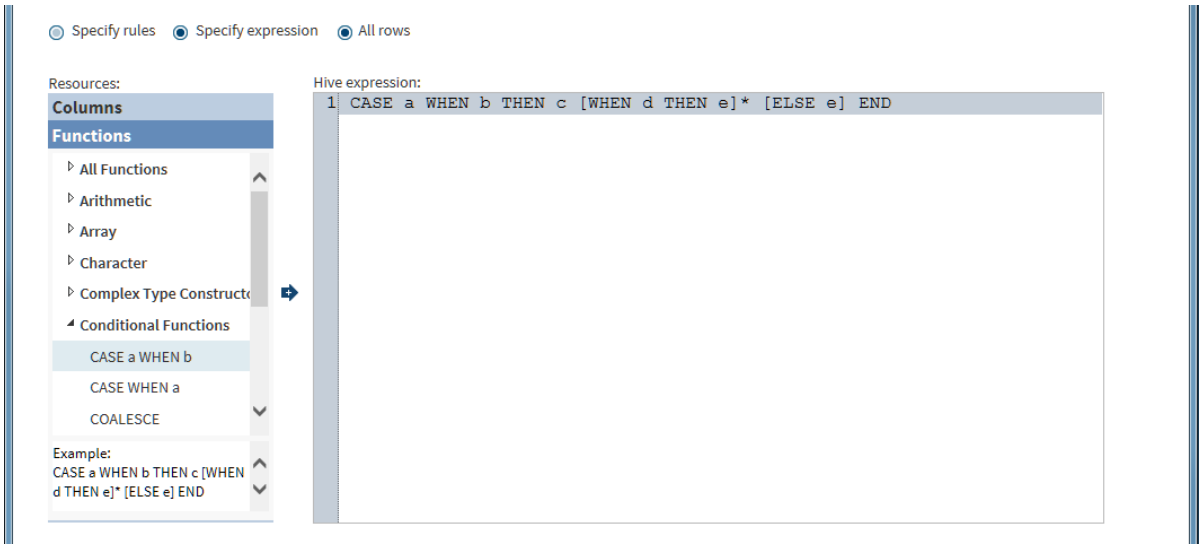
f. When your rules are complete, click **Next** and go to Step 5c on page 47.

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 ➡.

   c. When your expression is complete, click **Next** to display the **Code** task.

6. 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.

7. Click **Next** to open the **Result** task, and then click **Start deleting data**.
8. When the job is complete, click **Log** to confirm the deletion of rows.

Delete Tables

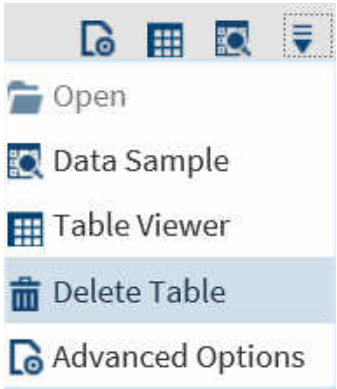
Introduction

To delete a table, use the **Action** menu in any directive that has a **Source Table** task.

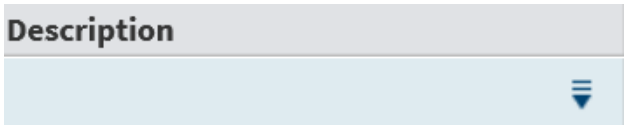
Example

Follow these steps to delete a table:

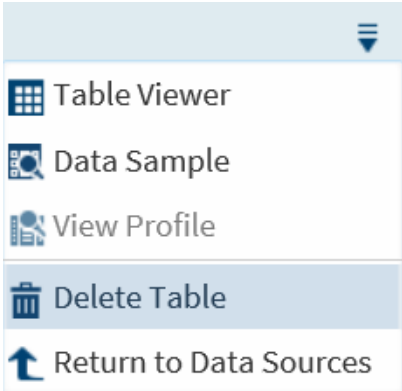
1. On a **Source Table** task, select the table that you want to delete.
2. Access the **Action** menu in one of two ways:
   - Click the **Action** menu and select the **Delete Table** action.



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

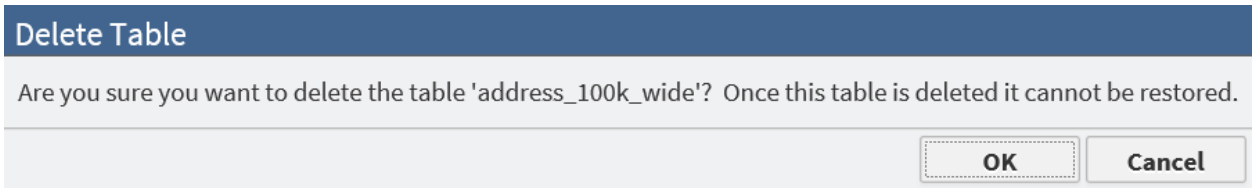


   Select the **Delete Table** action.



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 **OK** to delete the table.



CAUTION:
You cannot undo a table deletion.

Usage Notes

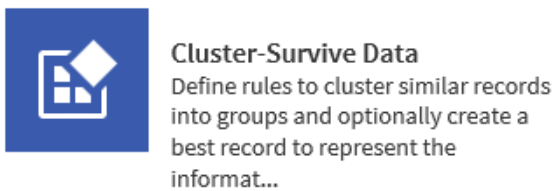
If you attempt to delete a table in HDFS, and if you are not authorized to complete that action, then Hive table metadata is deleted, but the table might remain.

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

You can use the **Delete Table** action to delete a view rather than a table. 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.

Cluster-Survive Data

Introduction



In the Cluster-Survive Data directive, the cluster process groups source rows that apply to a single entity. The survivorship process then uses other rules to select the best row or rows in the cluster, and write only those surviving row into the target for that cluster.

Additional rules can update selected columns in the surviving rows to include specified values from other rows in the cluster.

The Cluster-Survive Data directive can be defined to include clustering only, without survivorship.

Cluster IDs can be written into a new column in a separate target.

Note: Before running the Cluster-Survive directive, use the Cleanse Data directive on the source table to standardize or compute match codes for the columns that will be used for clustering. Use standardization to eliminate differences in casing and spelling. Generate match codes to enable fuzzy (inexact) matches.

Prerequisites

The Cluster-Survive Data directive, Apache Spark must be installed and configured on your Hadoop cluster.

The Data Loader for Hadoop Spark Engine must also be deployed on your Hadoop cluster.

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

About Clustering and Survivorship

In the Cluster-Survive Data directive, several tasks display detailed information that connects the directive interface to the output in the target table. You might need to scroll down to see the information in the following tasks: **Cluster Rules**, **Survivor Base Record Rules**, and **Survivor Column Rules**.

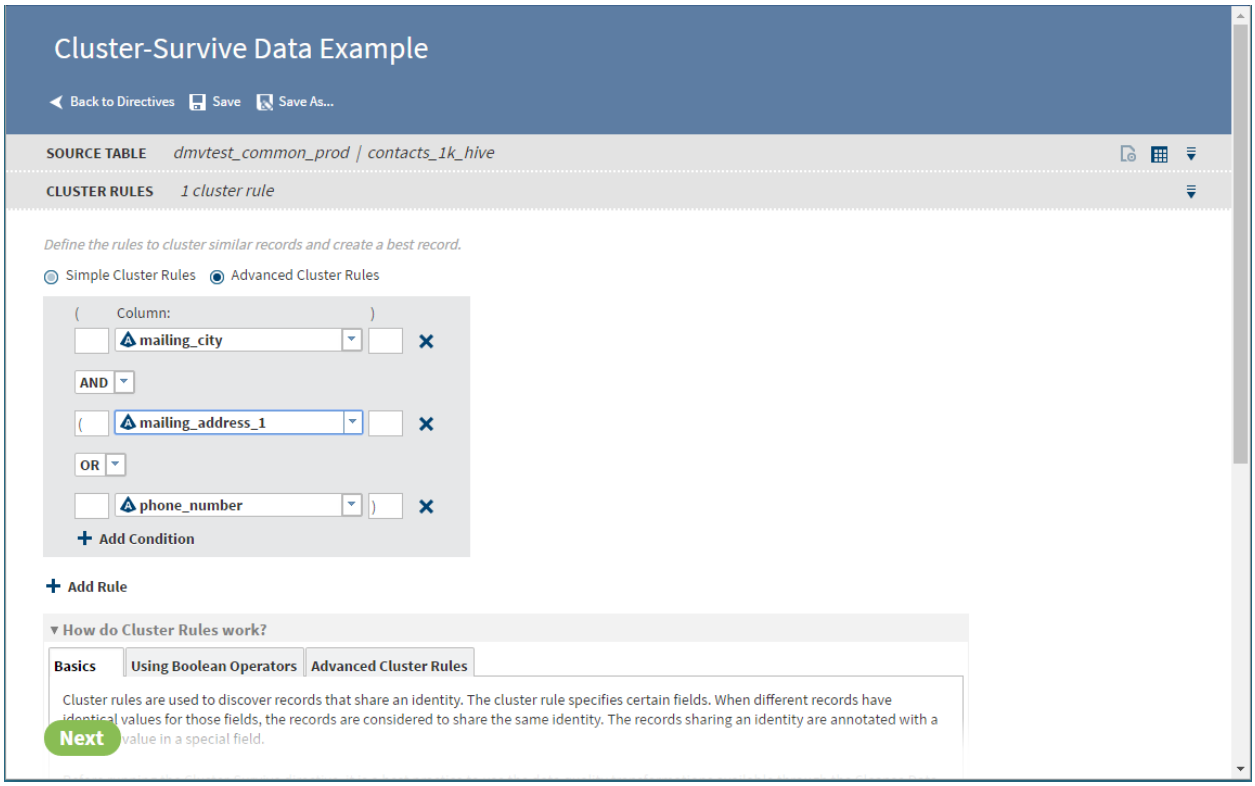
Example

Follow these steps to use the Cluster-Survive directive:

1. On the SAS Data Loader for Hadoop directives page, click **Cluster-Survive Data**.
2. In the **Source Table** task, select a source table and click **Next**.

   To learn about data sources, tables, and viewing data, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16.
3. Use the **Cluster Rules** task to define one or more rules that create clusters of source rows. The rules are applies to each row in the source table. Rows with matching results are assigned the same cluster ID.

   To define one or more rules without using parenthesis, click **Simple cluster rules**. To use parenthesis to separate AND clauses from OR clauses, click **Advanced cluster rules**. Enter open and closed parenthesis near the condition statements where they are needed.
4. Select at least one column to define a simple cluster rule, or at least three columns to define an advanced rule. Specify AND or OR clauses and add parenthesis to complete each rule. Add more conditions and more rules as needed. When the rules are complete, click **Next**.



5. In the **Cluster Output Table** task, either select a target table or add a new target table to receive the cluster ID column, and then click **Next**.

   Note: To not generate a cluster ID table, simply click **Next** without selecting a target.

   To learn how to browse data sources and target tables, and view data in tables, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16.

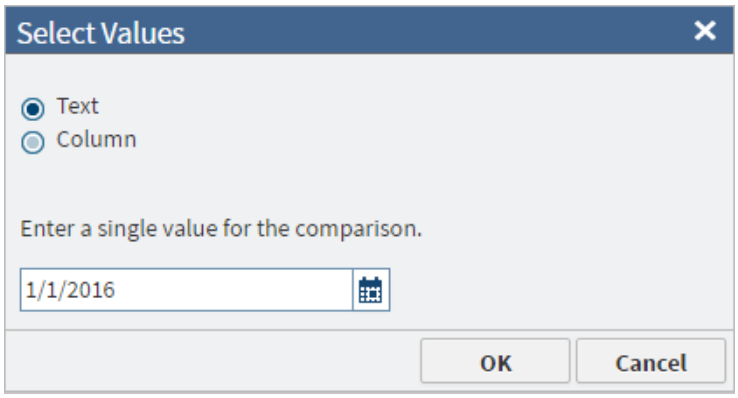
6. In the **Survivor Base Record Rules** task, define the simple or advanced rules that determine the selection of the base record or records in each cluster. All rows that match the criteria in the rules are written into the target for that cluster. Rows that do not match the rules are not written into the target.

   Note: To create a cluster ID target table without running survivorship, click **Do not create survivor base record rules**, and then click **Next**.

   To create rules, select a **Column** and **Operator**. If a **Value** is valid for your chosen operator, click the following icon:

   In the Selected Values window, you can specify a text value (including numbers or dates) or specify a value in another column in the same row. The value of the selected column will be applied to the condition to evaluate survivorship.



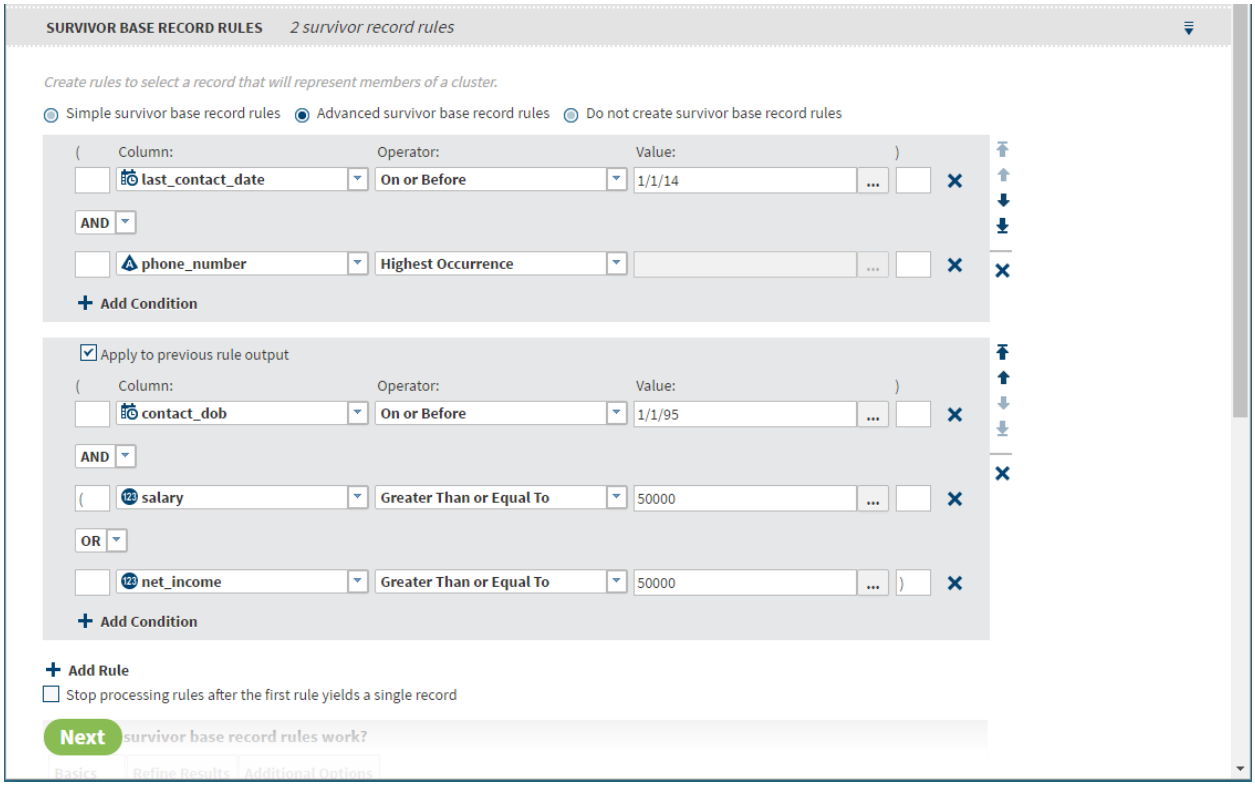
To learn how operators are applied to columns, see "About the Operators for Survivor Base Record Rules" on page 53.

Survivor base record rules are evaluated sequentially, in order of appearance. To change the order in which a rule is processed, click the arrow icons to the right of the rule pane.

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, then all of the records in the cluster are processed in the current rule.

In advanced rules, you can nest parenthesis to control precedence. Enter open and close parenthesis to the left or right of a condition.



7. In the **Survivor Column Rules** task, specify simple or advanced rules to modify the content of the survivor base record. You add or exclude content from specified columns in the surviving rows.

If you do not need to modify the survivor base record, click **Do not create survivor column rules**, and then click **Next**.

To use parenthesis in survivor column rules, click **Advanced survivor column rules**.

Add conditions, add rows, and enter parenthesis as needed.

To add source data to the target in the surviving rows, click . In the Select

Columns window, select the source columns that will be written into the target in the surviving rows, and then click **OK**.

When the survivor column rules are complete, click **Next**.

8. In the **Survivor Output Table** task, select or add a new survivor output table.

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

9. In the **Result** task, click **Save** or **Save As** to save the directive. To run the directive, click **Start Processing Data**. As the directive runs, view the code, log, errors, and results as they become available.

   When you view results, two reserved columns 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, do not write the columns gen_cid and gen_rid into the target. These columns 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.

About the Operators for Survivor Base Record Rules

In the task **Survivor Base Record Rules**, the following table defines how operators apply to selected columns.

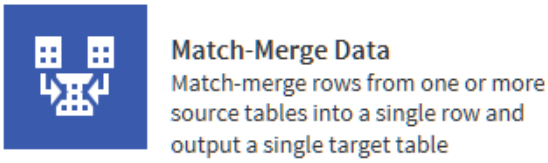
Operator	Description
Equal To *Note:* For time and date fields, this value becomes On.	Field value is • equal to another field • equal to a literal • contained in a list of literal values
Not Equal To *Note:* For time and date fields, this value becomes Not On.	Field value is not • equal to another field • equal to a literal • contained in a list of literal values

Operator	Description
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. • Applies to string data only.
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. • 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.

Operator	Description
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.

Match-Merge Data

Introduction



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 in 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 row 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 SAS DS2 expression. If the expression evaluates to true, then the row is written to the target. To develop DS2 expressions, the **Advanced Editor** in the **Filter Rows** task 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 also specified as user-written DS2 expressions.

To learn more about the match-merge process, see **How does Merge work?** in the Match-Merge directive. You might need to scroll down to display the information.

Example

Introduction

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

If you have a question about a particular task in the directive, refer to one of the following sections:

- "Select Source Tables"
- "Change Maximum Character Length"
- "Order Source Tables"
- "Matched Columns"
- "Merge By"
- "Input Columns"
- "New Columns"
- "Filter Rows"
- "Manage Output Columns"
- "Target Table"
- "Result"

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 a third 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 "Accessing Data Sources, Source Tables, and Target Tables" on page 16.

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 directive.

To change the *default* maximum character length in SAS, see "Change the Maximum Length for SAS Character Columns" on page 154.

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

1. Click a **Source Table** task at the top of the directive.

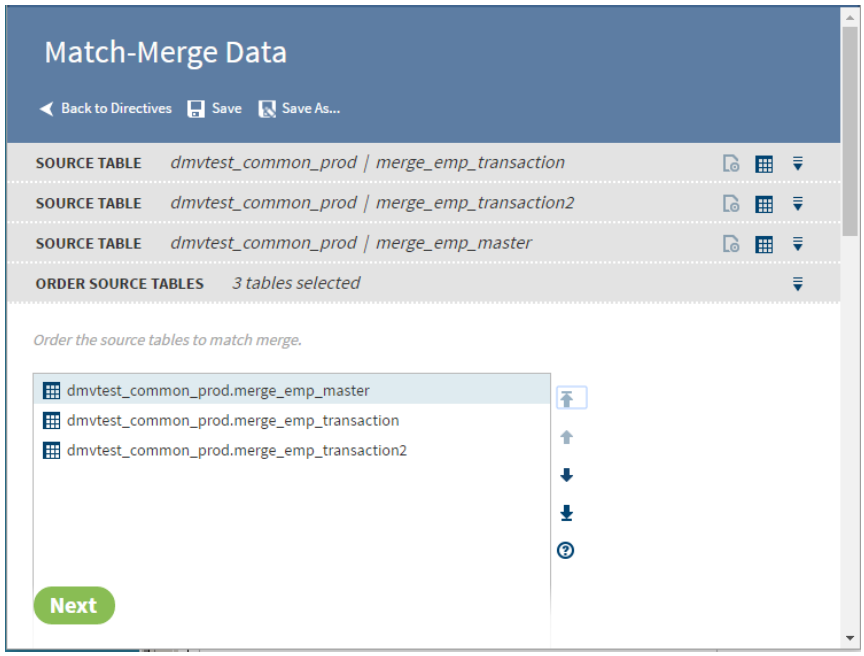
2. In the task, click .
3. In the Directive Settings window, view or change the value **Maximum length for SAS character columns**.

Order Source Tables

Use the **Order Source Tables** task to determine the source table that provides target data for a given merged row. The task defines an ordered list of source tables. The table that provides data for a merged row is the last table in the list that contains a matching merge-by value for that row. In general, source tables with the newest or latest data are often positioned at the bottom of the list.

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.

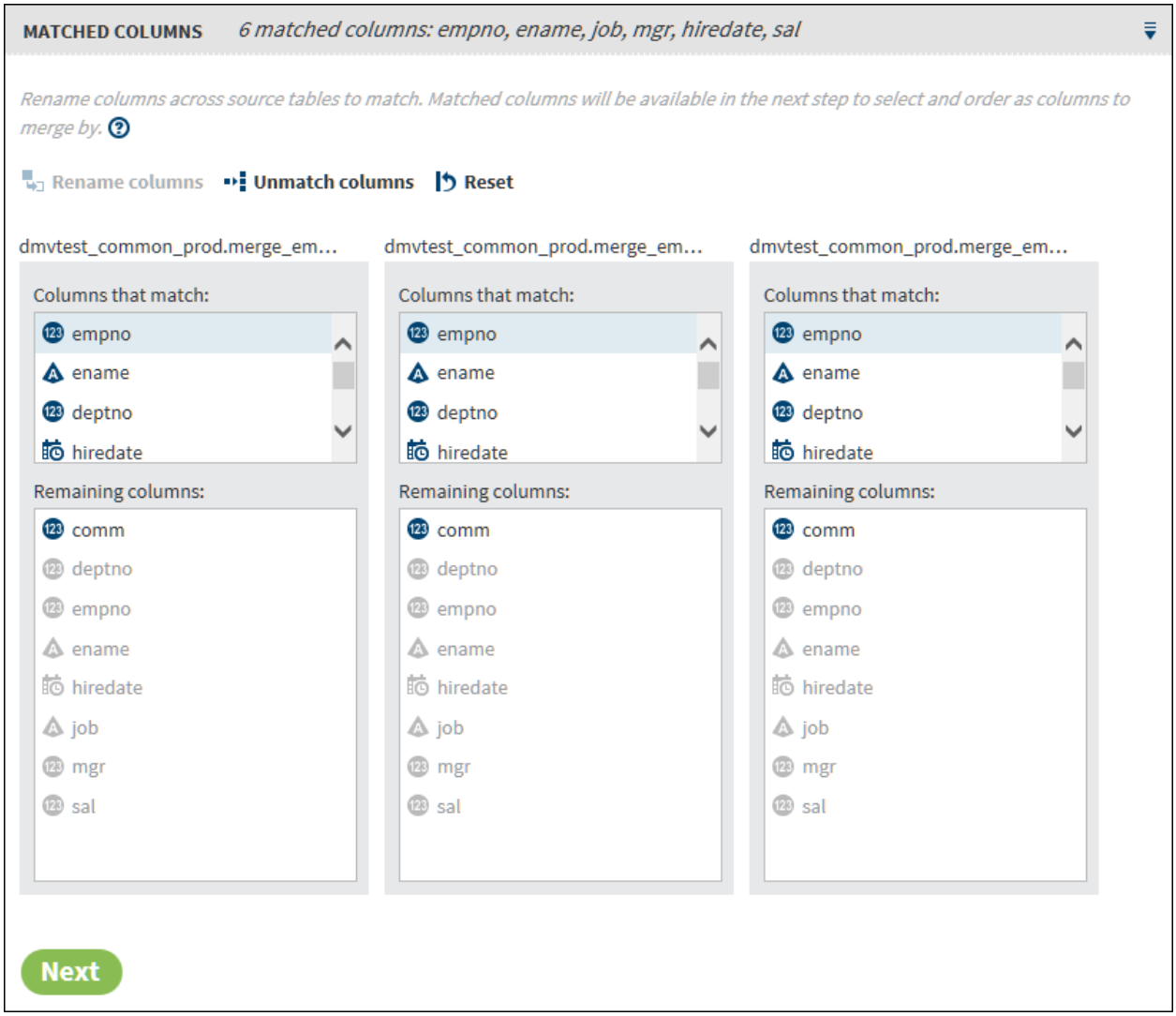
Note: The Matched Columns task displays by default all columns in all sources that have a matching name and type.

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 columns 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 columns are similar in purpose to the primary keys 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.

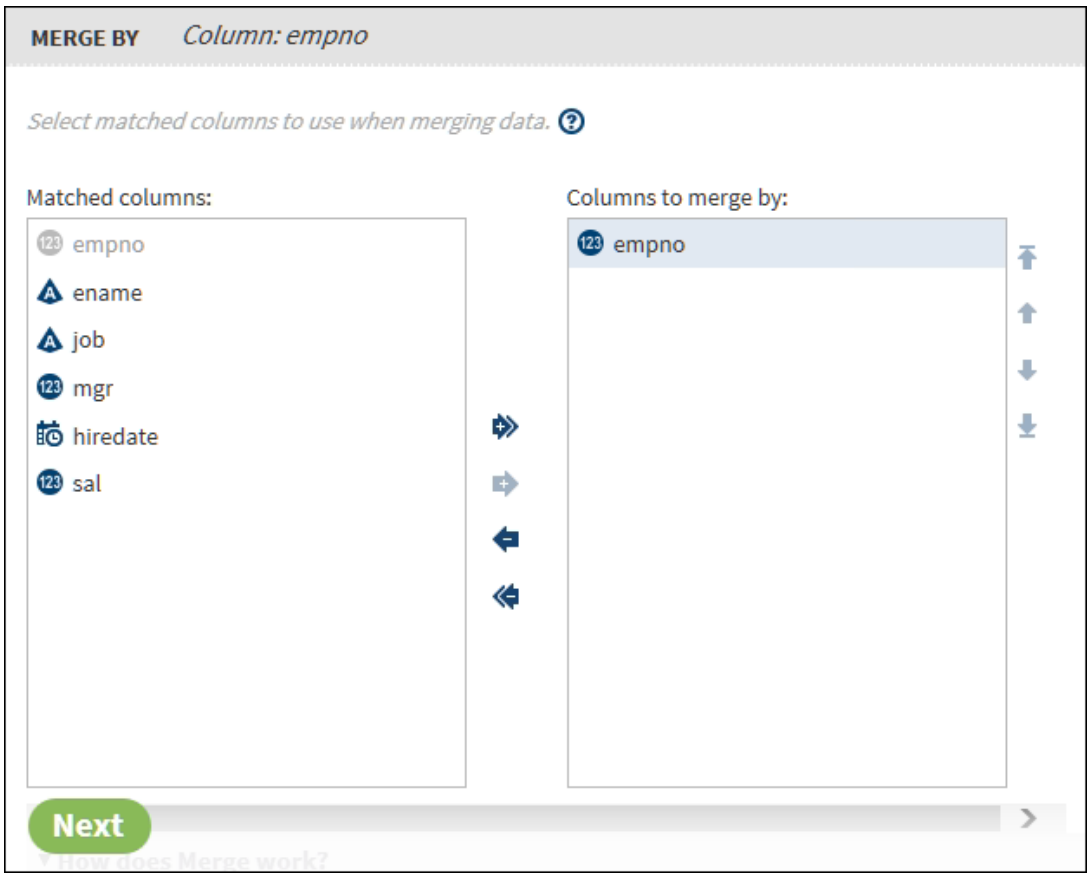
If you select more than one merge-by column, then the order of the columns specifies a group-by arrangement of the rows in the target table. The group-by arrangement arranges target rows by group and subgroup.

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.

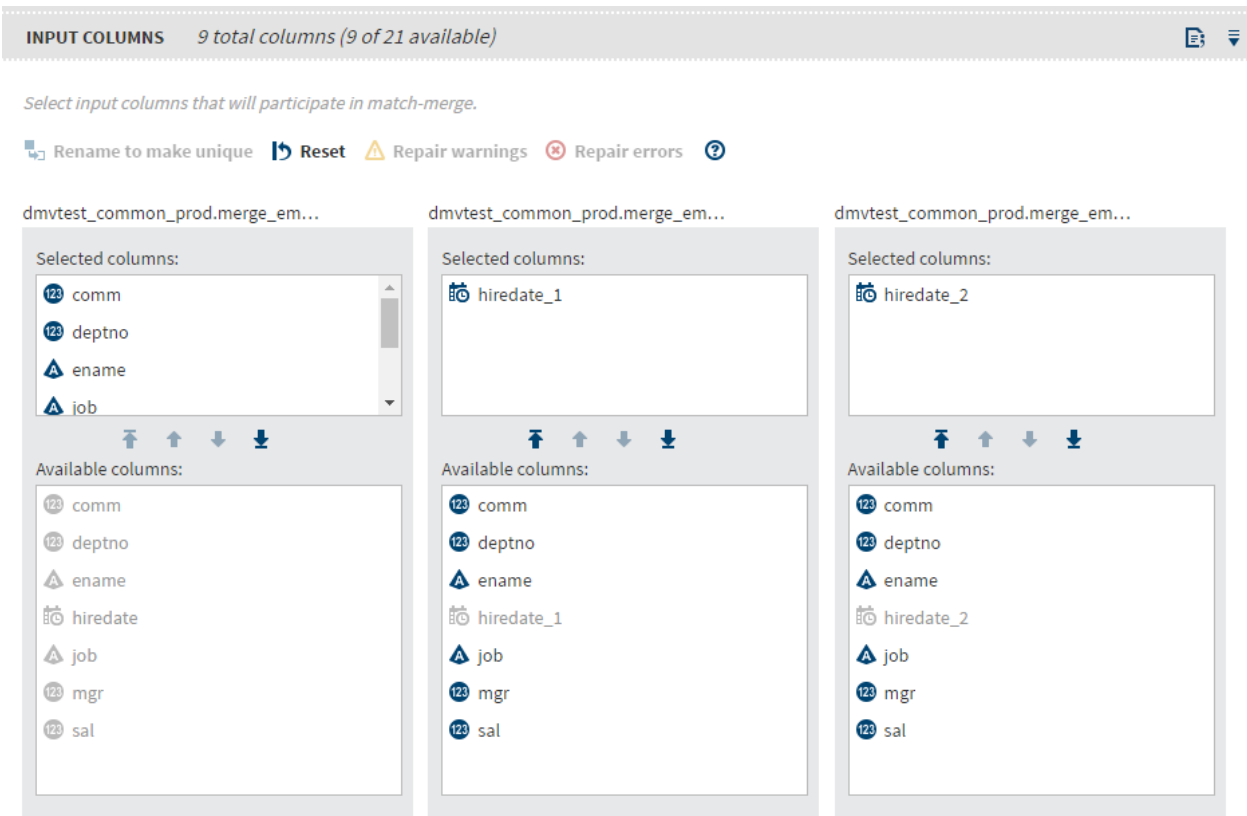
To specify the columns that will be merged in the target table, move columns from **Available columns** to **Selected columns**.

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 make unique**.

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**.

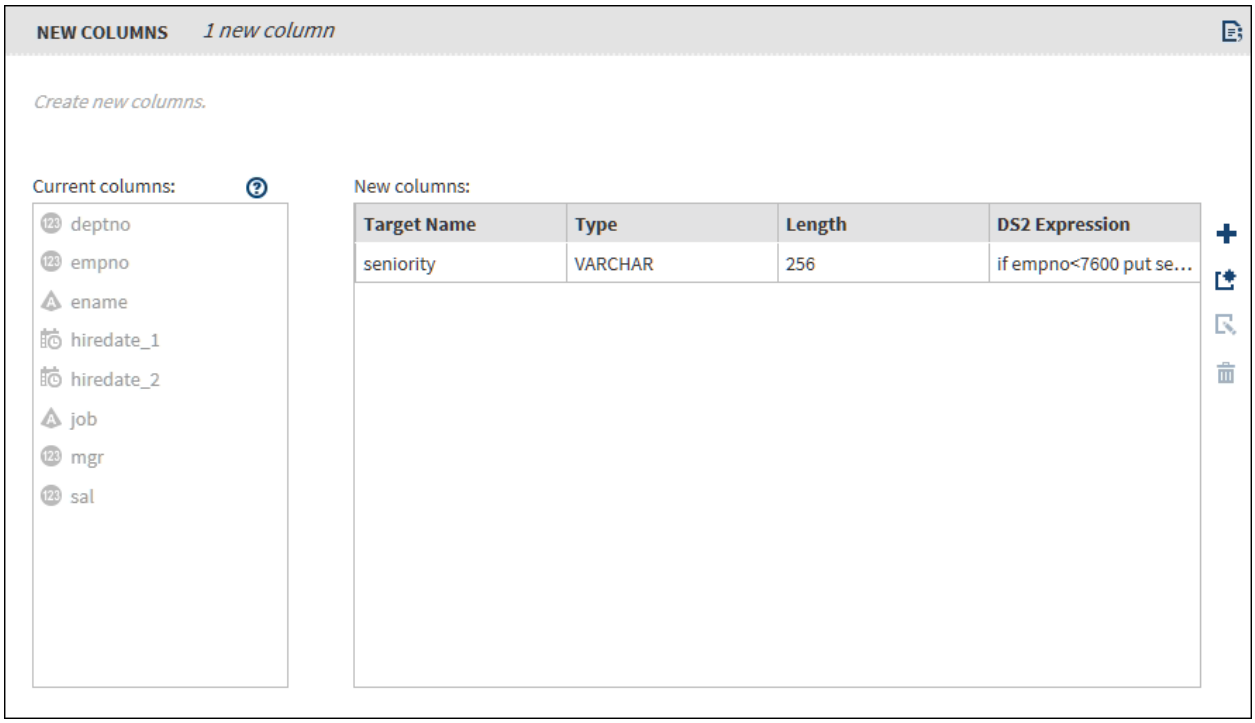
The following image depicts six columns that will be merged. The hiredate column will expand to three columns (hiredate, hiredate_1, and hiredate_2) in the target to show how that data changed over time.



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 column** 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 create a rule that filters rows based on the presence or absence of rows in source tables, click **Specify rule to indicate**. Selecting this option displays the option **Output rows for this rule automatically**, which is selected by default. When **Output rows** is selected, all source rows that evaluate to true for the rule are written into the target, regardless of the outcome of the DS2 expression.

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.

Note: You can create and apply multiple DS2 expressions, and each expression can have its own rule.

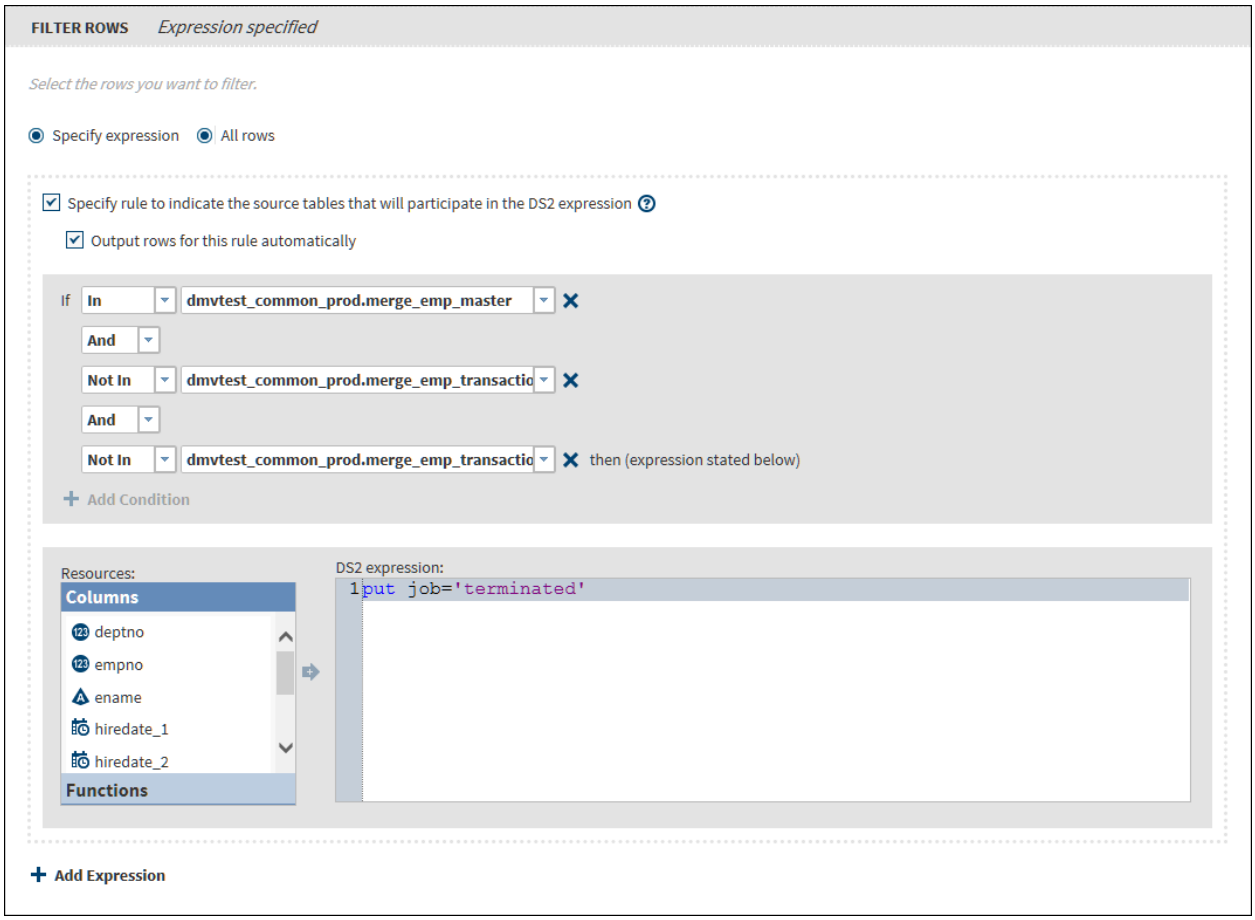
To specify a rule, select **In** or **Not In** for a selected source table. Click **Add condition** to apply a logical **AND** or **OR** and to specify a second source table. Add conditions as needed.

To create a DS2 expression, either paste an existing expression into the **DS2 expression** text box, or enter the expression into the DS2 expression text box. When entering an expression, note that the entire expression is inserted into SAS code. The expression should be syntactically correct with all the statements ending with a semi-colon. SAS Data Loader for Hadoop does make one exception to this rule: it adds a semicolon at the end of an expression if there is no semi-colon anywhere in the expression.

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.

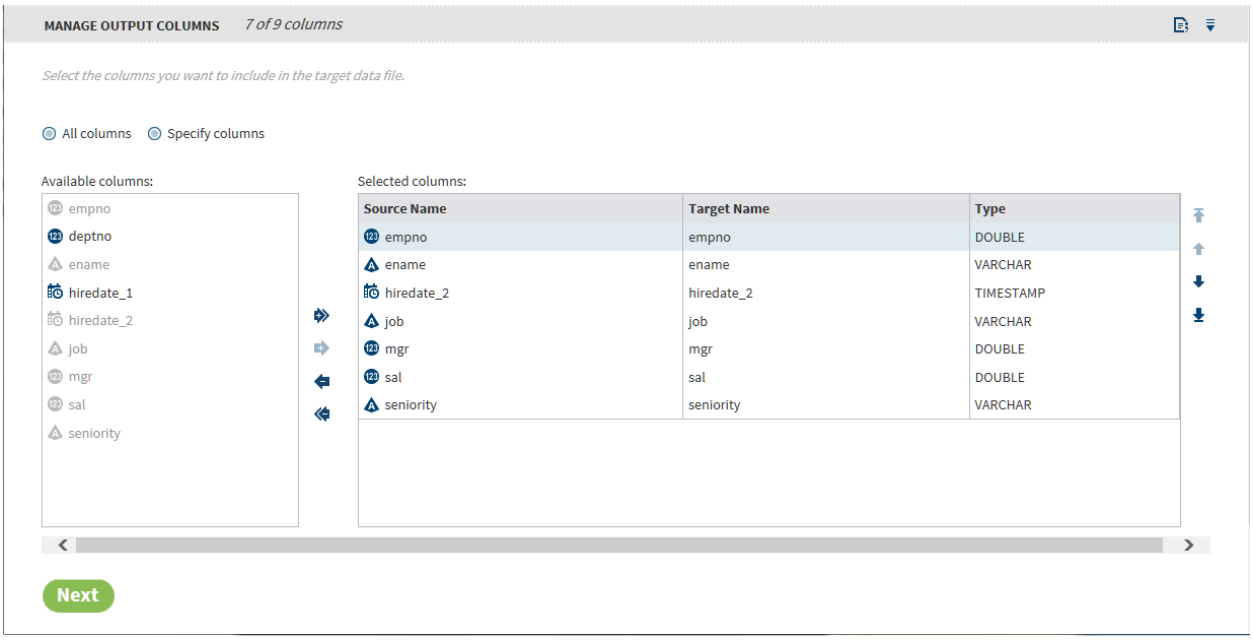
Initially, **All columns** is selected. To not reorder or remove target columns, click **Next**.

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.



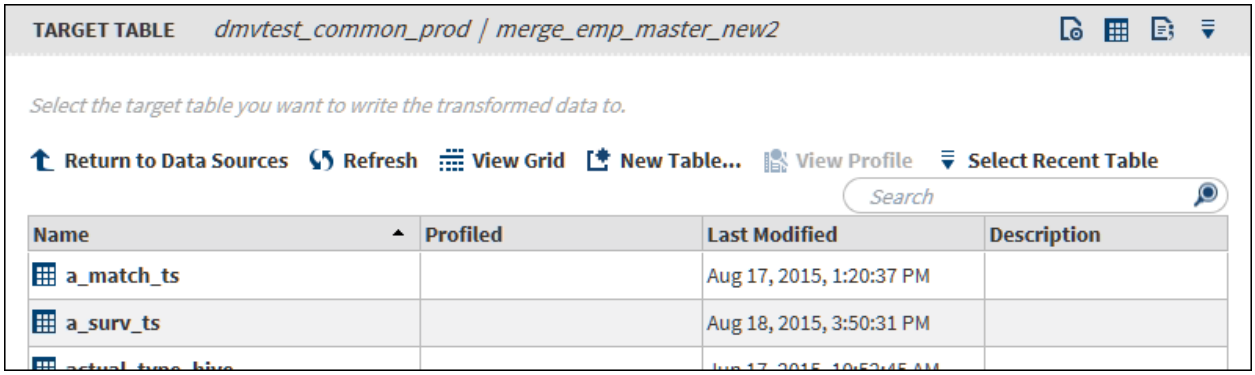
When your target columns are properly selected, named, and ordered, click **Next** to display the **Target Table** task.

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 directive.

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 24.



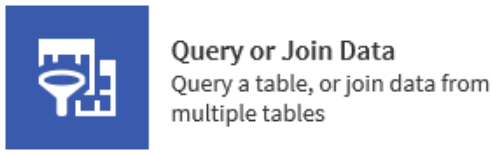
After you select your target table, click **Next** to open the **Result** task.

Result

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

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, as defined in the following example

The Query or Join Data directive can 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 directive, 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

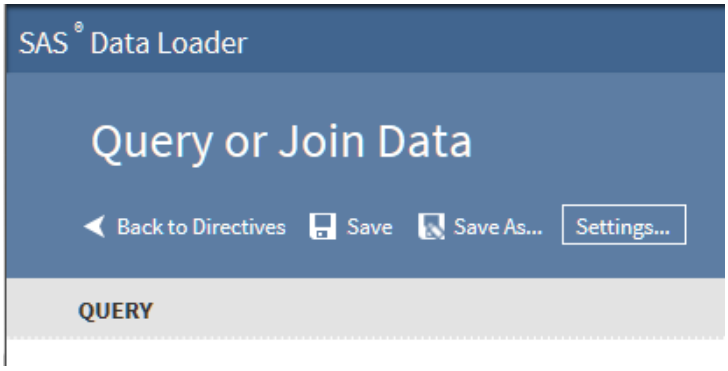
Support for the Cloudera Impala SQL environment is enabled in the **Hadoop Configuration** panel of the **Configuration** window. When Impala is enabled, new instances of the following directives use the Cloudera Impala SQL environment by default:

- 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".

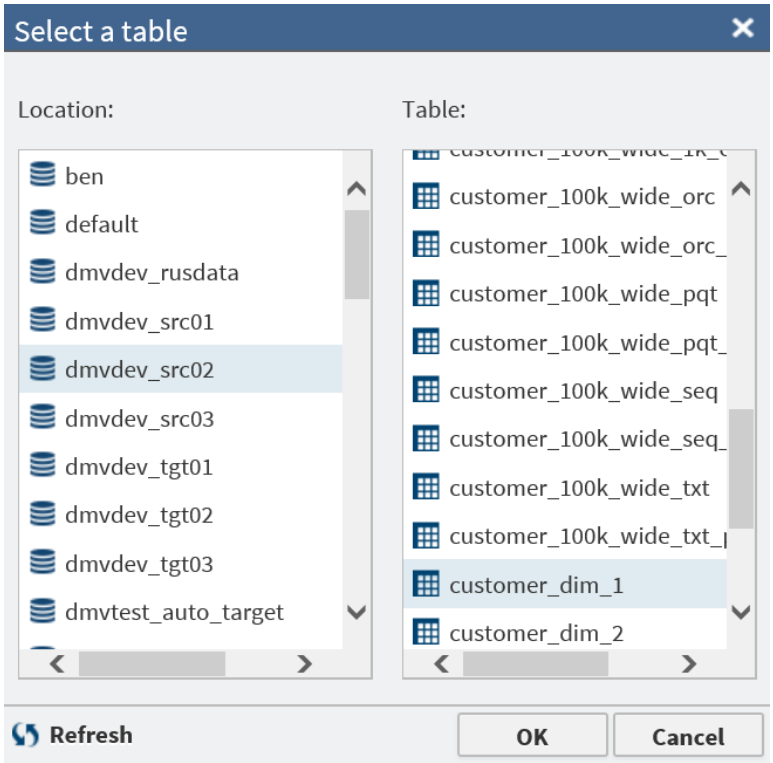
Note: Changing the default SQL environment does not change the SQL environment for saved directives. Saved directives continue to run with their existing SQL environment unless they are opened, reconfigured, and saved.



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 data source. Then click a source table in the **Table** list, and then click **OK**.



4. If your directive 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.

   TIP To display the full data source and table name, position the cursor over the base table or join table field.

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.

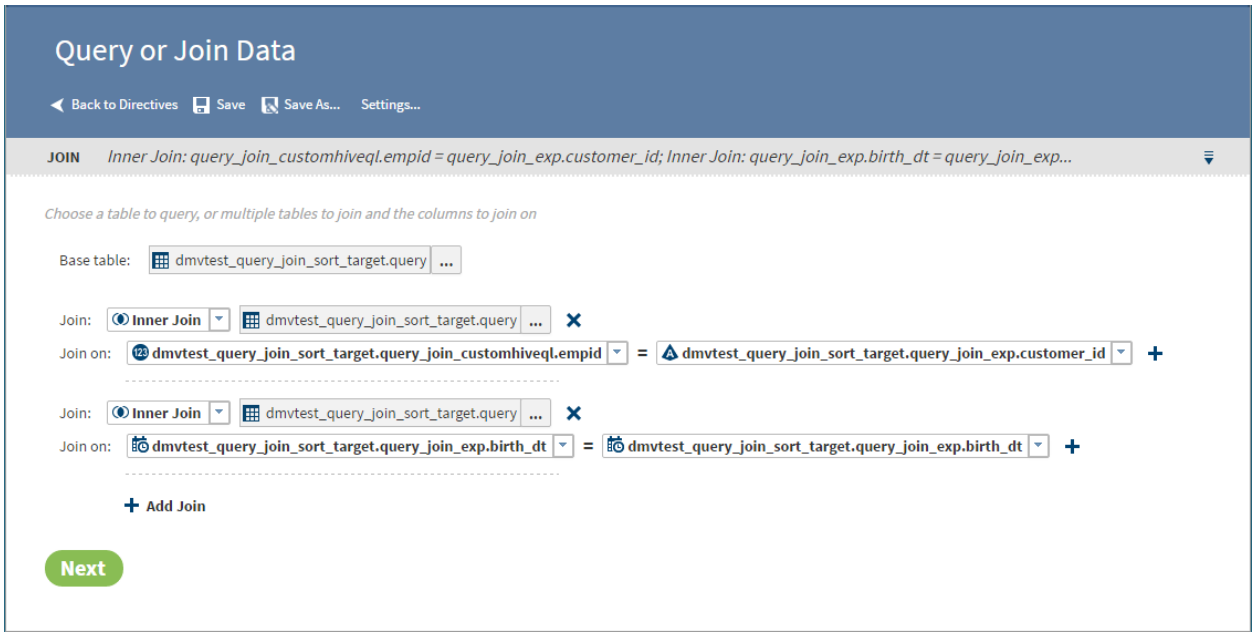
   Right
   : The right or right-full join reverses the definition of the left join. All rows from the right table appear in the target. If no values match between the join-on columns, then null values are written to the target for the columns of the table on the left side of the join statement.

   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 +. 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. Click **Next** and wait a moment while the application assembles in memory the names of the joined columns.
12. 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.)
13. 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 directive 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.
14. To subset the first group with a second group, and to generate a second set of aggregations, click + Add Column.
15. In **Summarize columns**, select the first numeric column that you want to aggregate.
16. In **Aggregation**, select one of the following:

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

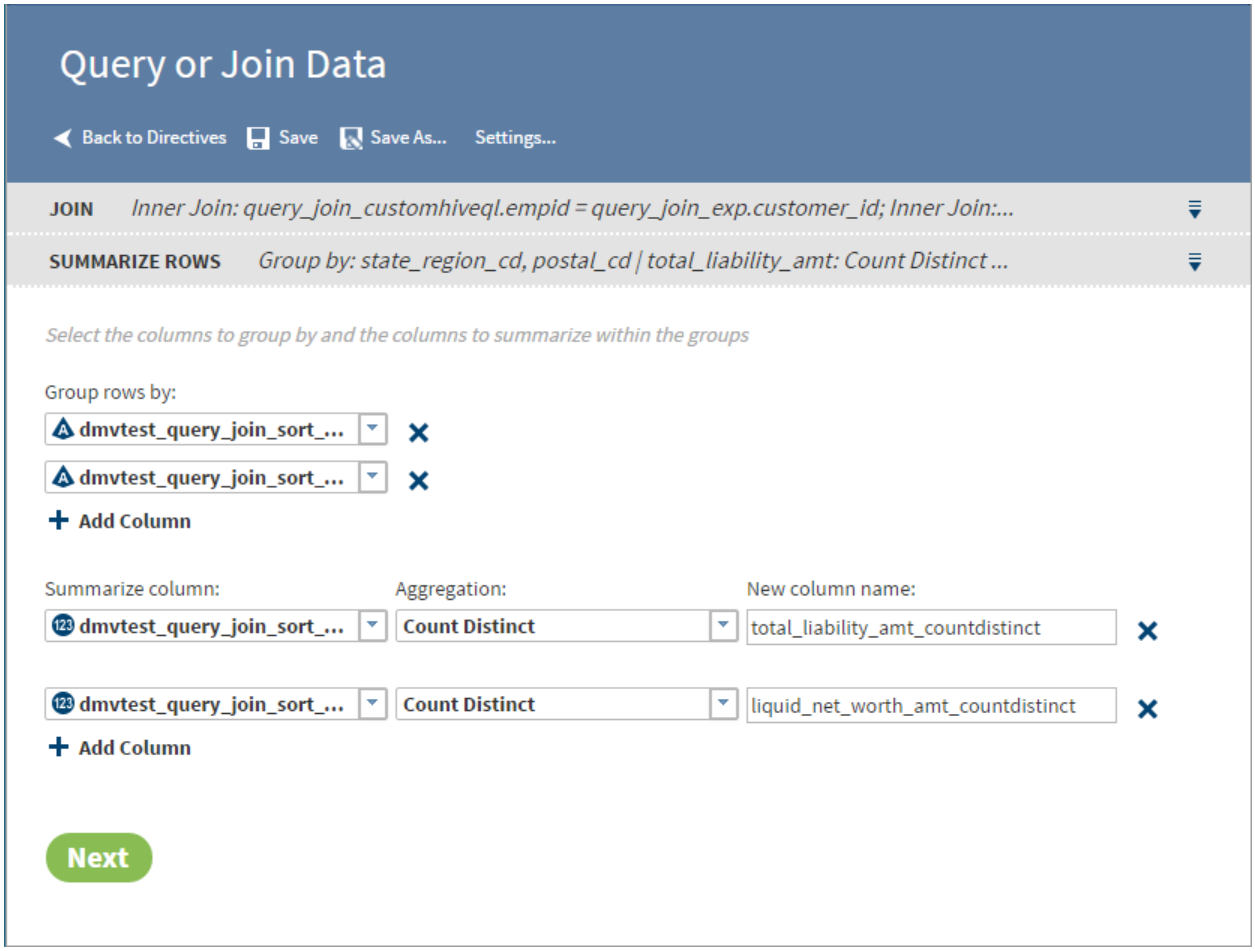
    Count Distinct
    : specifies the number of rows that contain distinct (or unique) values in each group.

    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.

17. In **New column name**, either accept the default name of the aggregation column, or click to specify a new name.
18. To add an aggregation, click + Add Column.



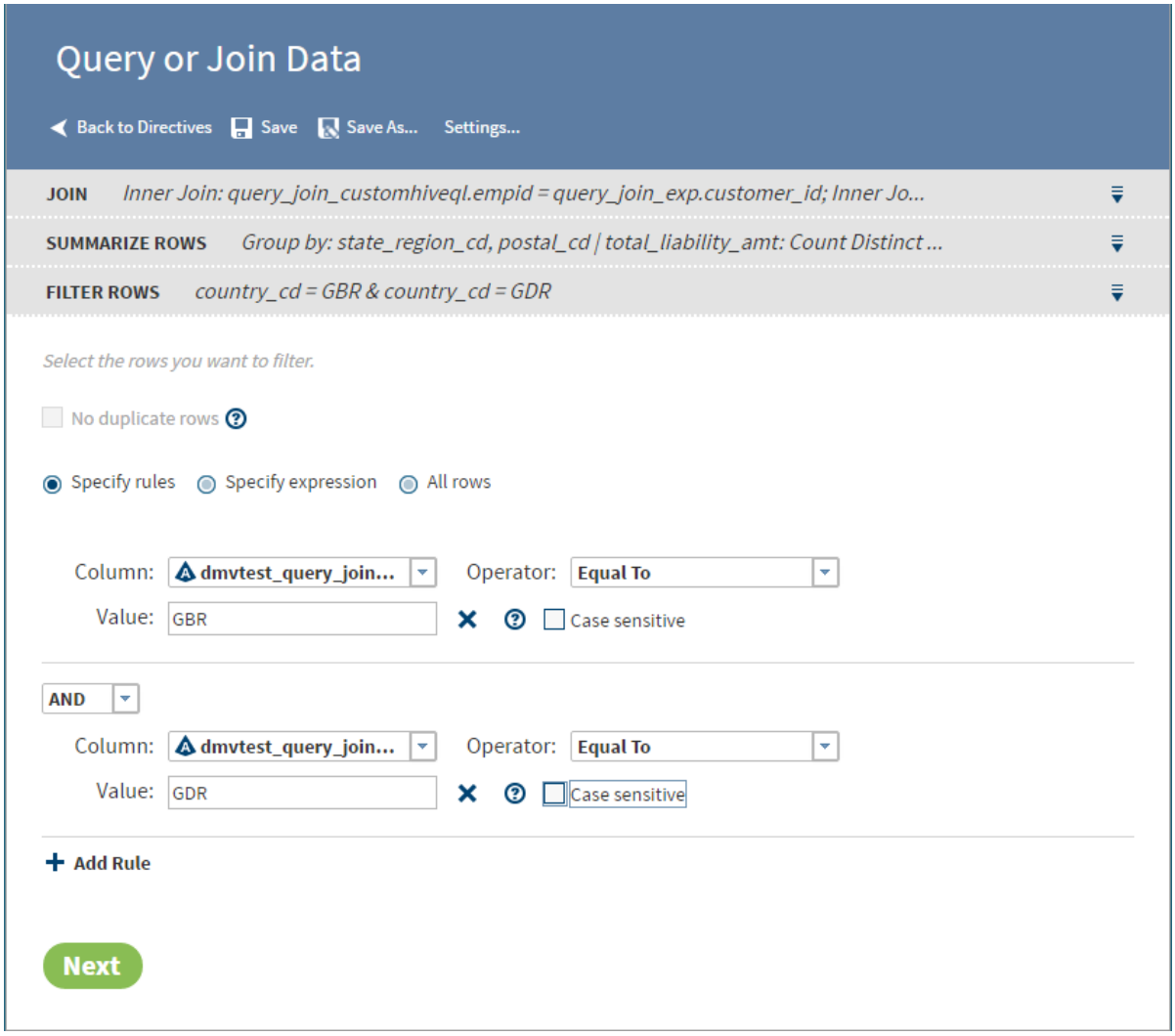
19. When the aggregations are complete, click **Next**.
20. In the **Filter Data** task, specify the source rows to exclude from the target table. All source rows are included in the target by default. To accept this default, click **Next**.
21. If your directive includes no summarizations, then you 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**.

22. 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 SQL expression, click **Specify expression** and go to Step 24 on page 69.
23. 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 rule.

d. Click **Add Rule** as needed to add another rule. 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 25 on page 70.

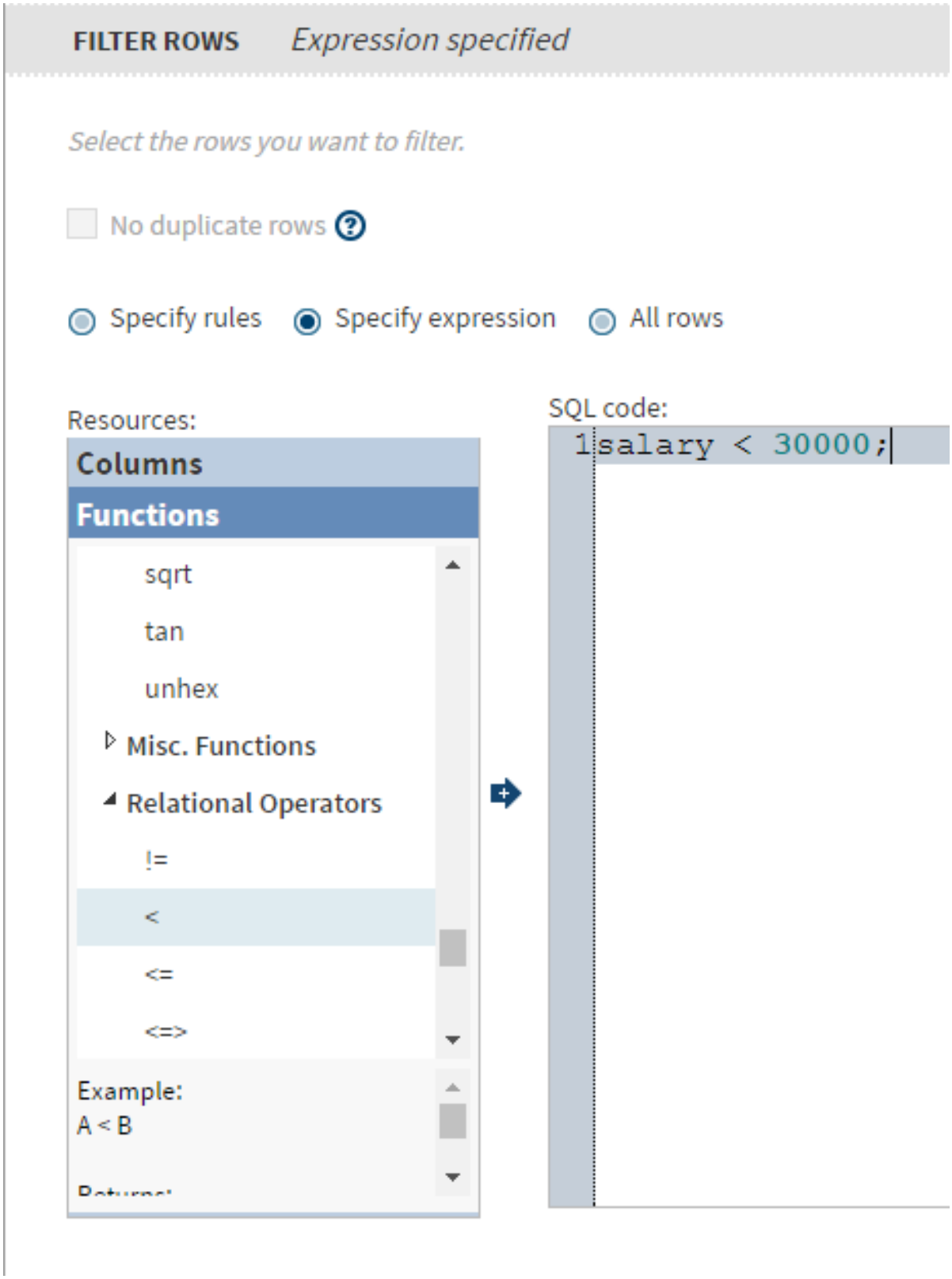
24. To filter rows using an SQL expression, follow these steps:

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

Your expression must use either HiveQL or Cloudera Impala SQL syntax, 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 or columns to your expression, click **Functions** or **Columns** in the **Resources** box, select a function or column, and click .



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

26. Use the **Columns** task to remove, reorder, or rename target columns. You can also replace the values in existing columns with the results of user-defined expressions. or add new columns for expressions. To learn more about this task, see "Manage Columns Transformation" on page 36.

    Note: If you defined summaries in the **Summarize Rows** task, or if you want to accept the default column definitions, click **Next**.

    When your column definitions are complete, click **Next** to display the **Sort** task.

27. Use the **Sort** task to arrange target rows based on specified columns. Click **Select a column**, click a column, and then select an ascending or descending sort. To subset the sort, click + Add Column and select another column.

    Note: If your directive includes summaries in the **Summarize Rows** task, then your choices of sort columns are limited accordingly.

    When your sort columns are complete, click **Next** to display the **Target Table** task.

28. In the **Target Table** task, choose an existing table to overwrite existing data or create and name a new target table. You can select a recently selected table or select Return to Data Sources to choose a target from a different data source.

    To learn about browsing tables and viewing table data, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16. ..

    After you specify your target table, click **Next** to display the **Code** task.

29. 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.

30. 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

31. Use the **Code** task to view and edit the code that is submitted to Hadoop for your directive. Click **Edit Code** to edit the generated code. Click **Reset Code** to restore the original generated code. Click **Next** to open the **Result** task.

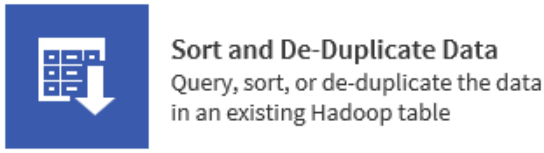
    Note: Edit your SQL code with care. The code in the editor is the exact code that will be executed by your job, regardless of previous selections. Code changes are not reflected in prior tasks, so code regeneration does not retain code edits.

32. In the **Result** task, click **Save** or **Save As** to save your directive.

33. Click **Start** to execute your directive. During execution, you can view the code, log, error, and results as they become available.

Sort and De-Duplicate Data

Introduction



Use the Sort and De-Duplicate Data directive to execute some or all of the following steps on source data in Hadoop:

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 for Hadoop directives page, click **Sort and De-Duplicate Data**. The **Source Table** task is displayed.
2. In the **Source Table** task, select a table from the default data source or click Select Recent Table. To select a table from a different data source, click Return to Data Sources. After you select a source table, click **Next** to display the **Summarize Rows** task.

   For more information about browsing data sources, selecting tables, and viewing table data, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16.
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. To specify a second summary, click + Add Column ..

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

4. Use the **Filter** task to remove from the target table duplicate rows and rows that are identified by a rule or a filter.

If you specified 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**.

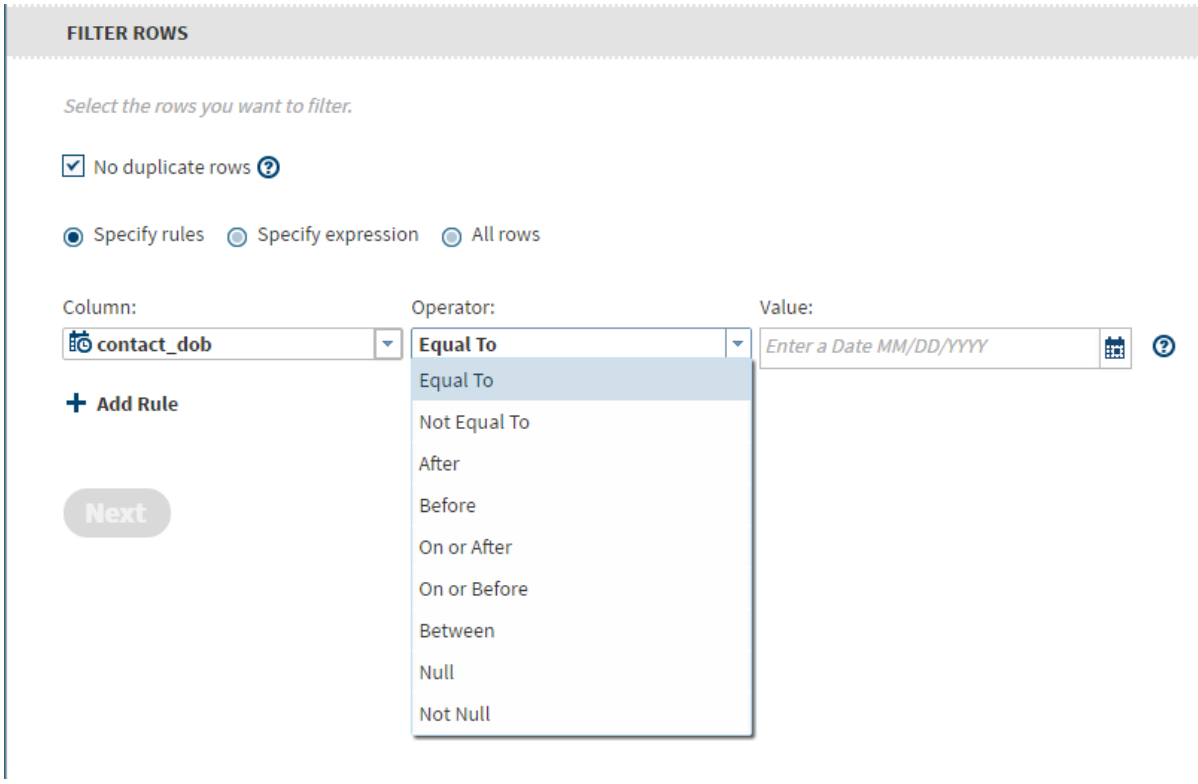
b. To filter rows from the target using one or more rules, click **Specify rules**.

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:



To learn more about operators, see "About the Operators in the Filter Data Transformation" on page 81.

iii. In the **Value** field, add the source column value that completes the rule

iv. To add another rule, click **+ Add Rule**. To associate a new rule with the previous rules, either retain the default **AND** clause or select the **OR** clause.

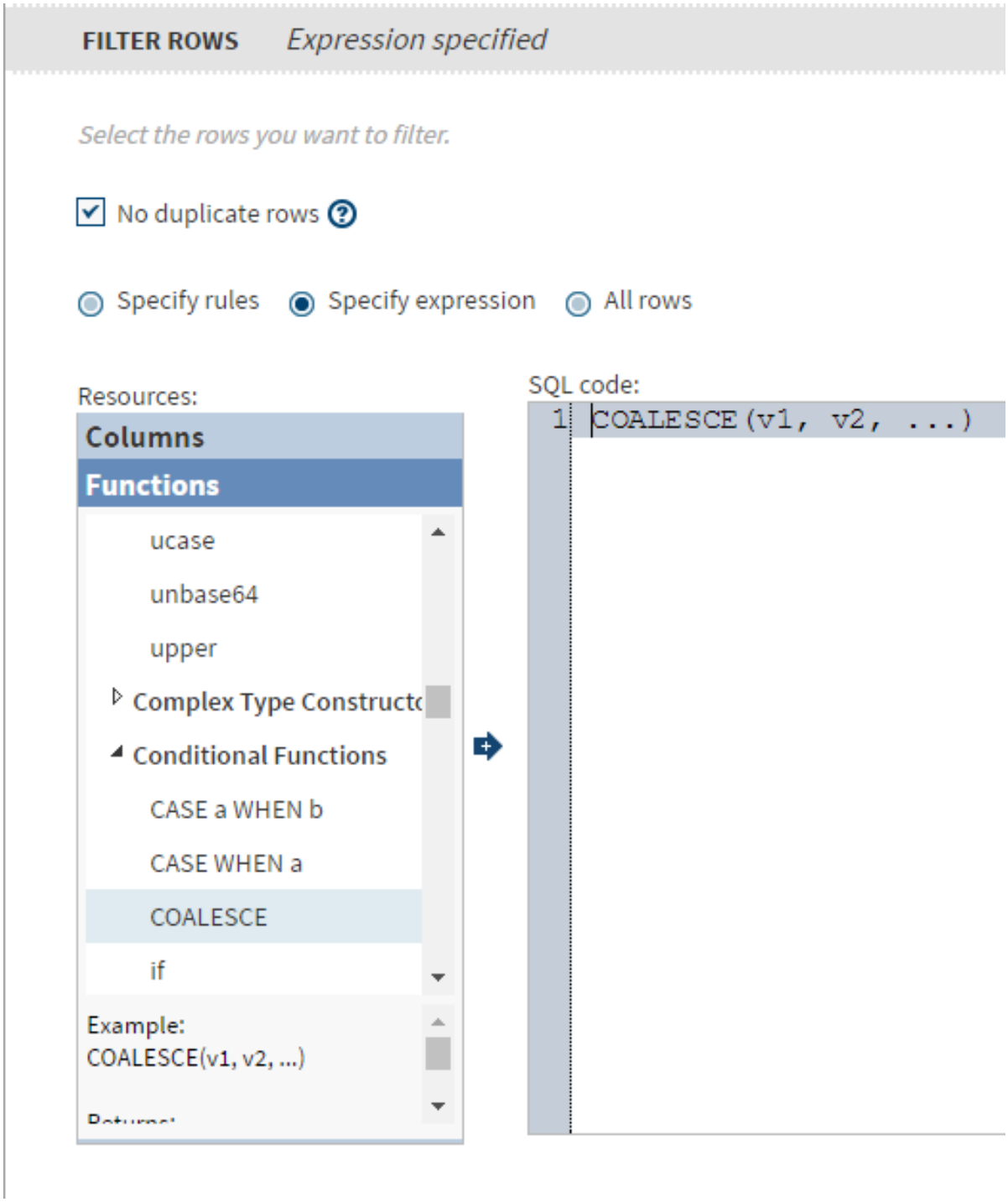
v. When your rules are complete, click **Next** to display the **Columns** task and go to Step 5.

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 syntax or Cloudera Impala SQL syntax. Click **Settings** to display the selected SQL environment.

ii. To develop your expression, select functions and columns from the **Resources** box, and then click to move them into the **SQL expression** box.



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

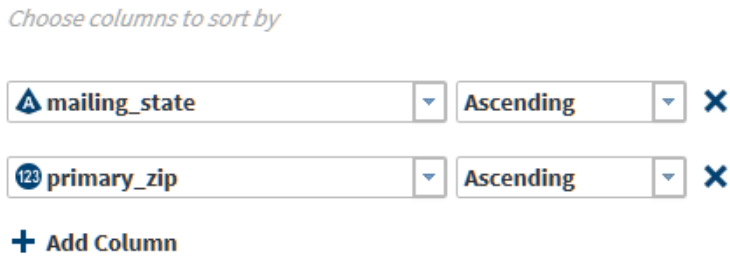
5. Use the **Columns** task to remove, reorder, or rename target columns. You can also replace the values in existing columns with the results of user-defined expressions. or add new columns for expressions. To learn more about this task, see "Manage Columns Transformation" on page 36.

   Note: If you defined summaries in the **Summarize Rows** task, or if you want to accept the default column definitions, click **Next**.

   When your column definitions are complete, click **Next** to display the **Sort** task.

6. Use the **Sort** task to change the order of the rows in the target table based on ascending or descending values in specified columns.

   The **Sort** task is available only if you did not define summaries in the **Summarize Rows** task. If you defined summaries, or if you do not need to sort your rows, then click **Next** to display the **Target**.



7. Use the **Target Table** task to specify an output table. You can create a new target, select a target in your most recent data source, select a recently accessed table, or select a table from a different data source. .

   To generate a temporary table that is not saved to disk, select **Save as a View**.Click

   When you have selected your target table, click **Next** to display the **Code** task.

8. Use the **Code** task to review and edit the generated code for the directive.

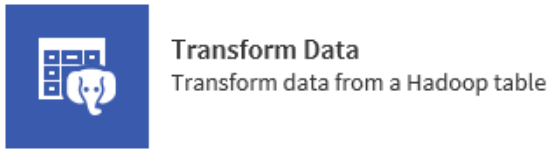
   Note: Code edits are not reflected in the tasks of the directive. Edits are lost if you update a task and regenerate code.

   Click **Next** to display the **Result** task.

9. Use the **Result** task to run your directive in Hadoop. Click **Save** or **Save As** to save your directive. . Click **Start Querying Data** to begin execution. As your directive runs in Hadoop, you can examine the code, log, errors, and results as they become available..

Transform Data

Introduction



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

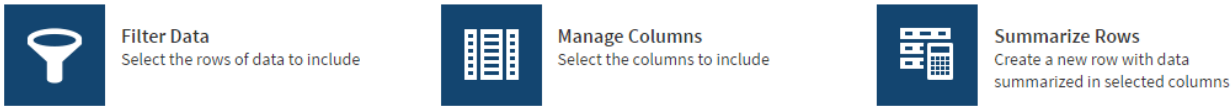
Example

Follow these steps to use the Transform Data directive:

1. On the SAS Data Loader for Hadoop directives page, click **Transform Data**. The **Source Table** task is displayed.
2. In the **Source Table** task, select the table that you will transform. You can choose from your last data source, or from a list of recently accessed tables. To select a table from a different data source click Return to Data Sources.

   For more information about data sources, tables, and viewing table data, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16.

   After you select a source table, 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. You can paste existing expressions, or use the Advanced Editor to create expressions.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 directive 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. Use the Filter Data transformation to reduce the number of rows that are written to the target. To identify the rows that you want to keep in the target, you apply an expression or one or more rules. Click **Specify rules** or **Specify expression**.
6. To specify rules, follow these steps.
   a. To define a rule, select a column and an operator, and then specify a value. 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 81.
   b. To add another rule, click + Add Rule, specify a column, operator, and value, and then choose a logical operator (AND or OR). Add more rules as needed.
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 DS2 functions (with the MapReduce runtime target) or 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".

      See the **Resources** box for syntax help for the DS2 or EEL functions. See also *SAS DS2 Language Reference* or *Expression Language: Reference Guide*.

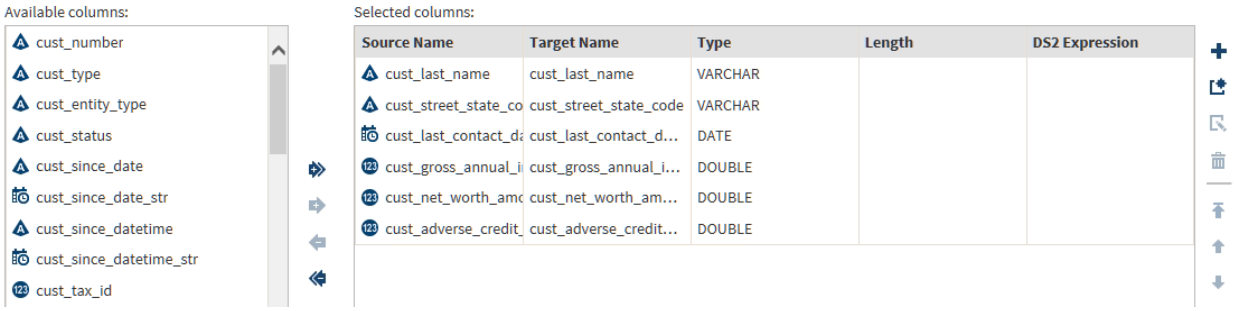
b. To write an expression, select functions and source columns from the **Resources** box. To move columns and functions into the **expression** box, click . To enter values or delete text, click the expression box.

8. After you define rules or an expression, click **Next** to 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 that will be 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

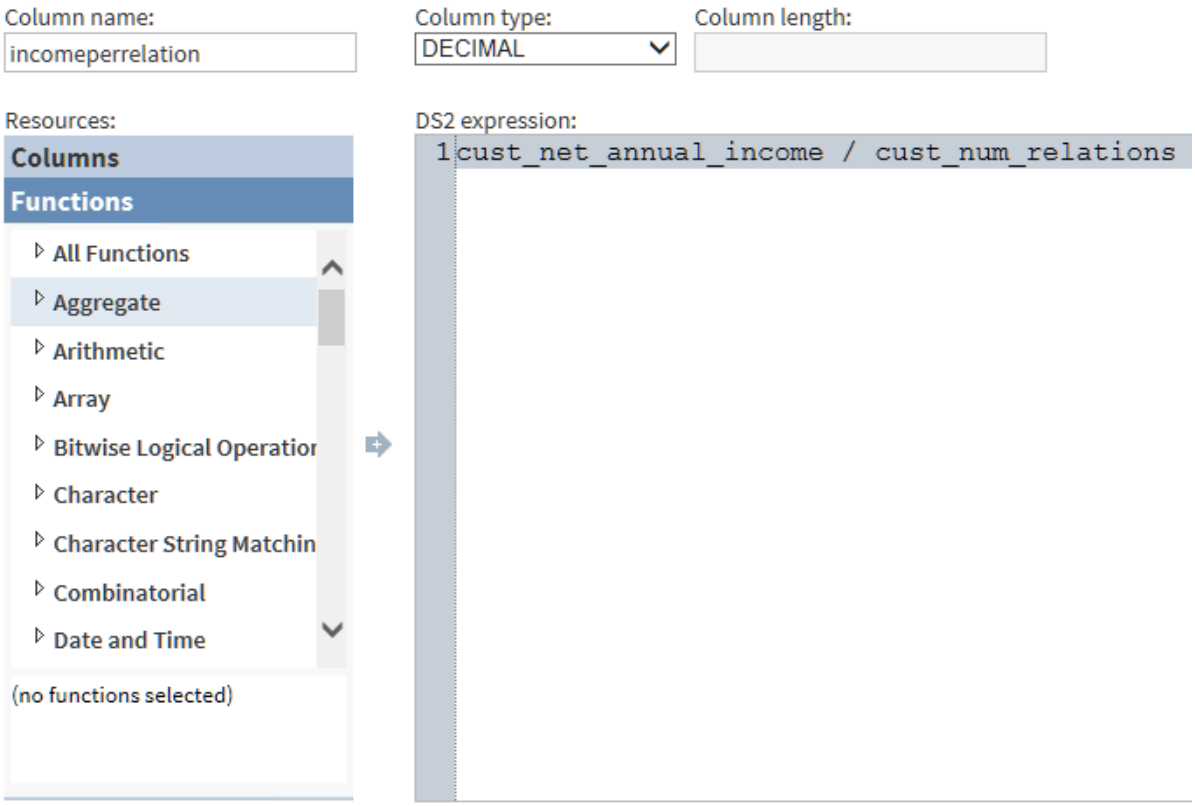
14. To replace existing column data with data that is generated by a user-written expression, click a **Selected column** and click **Expression**. At this point, you can enter or paste an existing expression in the corresponding text field.

    Your expression can use either DS2 functions (with the MapReduce runtime target) or 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" . See also *SAS DS2 Language Reference* or *Expression Language: Reference Guide*.

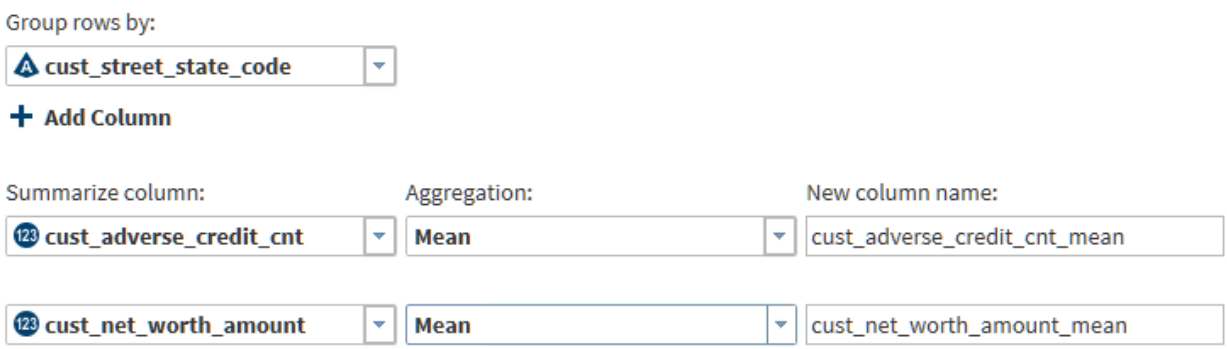
15. To add a new target column, and to use the Advanced Editor to write an expression for that column, click .

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 can receive a value in an aggregation column in the target.

    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 85.
20. Click **New column name** as needed to revise the default name.



21. When your summaries are complete, click **Next** to display the **Target Table** task..
22. In the **Target Table** task, click New Table..., select a target table from the current data source, or click Select Recent Table . To select a target table from a different data source, click Return to Data Sources.

    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.

    Click **Next** to display the **Result** task.
23. In the **Result** click **Save** or **Save As** to save your directive. Click **Start transforming data** to run your directive in Hadoop. Otherwise, you can run your directive later from "Saved Directives".

    During the execution of the directive, you can access the code, log, errors, and results as they become available.

About the Operators in the Filter Data Transformation

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

Table 4.1 *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`

Not Equal To		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`

Operator	Source Column Data Types			Description and Example

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`

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`

Operator	Source ColumnData Types	Description and Example
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%**
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**

Operator	Source ColumnData Types	Description and Example
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`**
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`**

Operator	Source ColumnData Types	Description and Example
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.

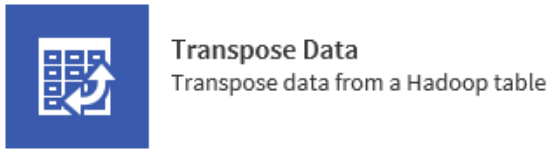
Standard Error
: measures the applicability or accuracy of the mean as it applies to the values in the group. A small value indicates that the mean is a more accurate reflection of the values in the group.

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



Use the Transpose Data directive to transpose source columns into rows in the target. The directive defines target rows based on group-by columns. Values in transposed columns can appear in new target columns whose names are the data values from a selected source column. Other columns from the source can be copied to the target without transposition.

An example of transposition could involve a source table that contains a column of country names. If you transpose the country column, the target will contain one row for every row in the country column. If the source column Government Type was selected as the ID column, then the target would contain one column for each type of government. In those columns, each transposed country name would appear in its respective Government Type column.

To learn more about how the Transpose Data directive works, see the Help that is provided in the directive. The Help contains examples of transposed data.

CAUTION:
Selecting columns with a high degree of cardinality (number of unique values) can decrease performance. 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 for Hadoop directives page, click **Transpose Data**.
2. In the **Source Data** task, select a source table from the current data source or click Select Recent Table . To select a table from a different data source, click Return to Data Sources .

   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 88.

   To learn more about browsing data sources and tables, or viewing data in tables, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16.

   After you select your source table, click **Next**.

3. Use the **Transpose Data** task to specify transposed columns, group-by columns, ID columns, and the source columns that are copied to the target without transposition. Click a row in the **Roles** box, click a column in the **Columns** box, and then click .

   At least one column is required for **Transpose columns** and **Columns to group by**.

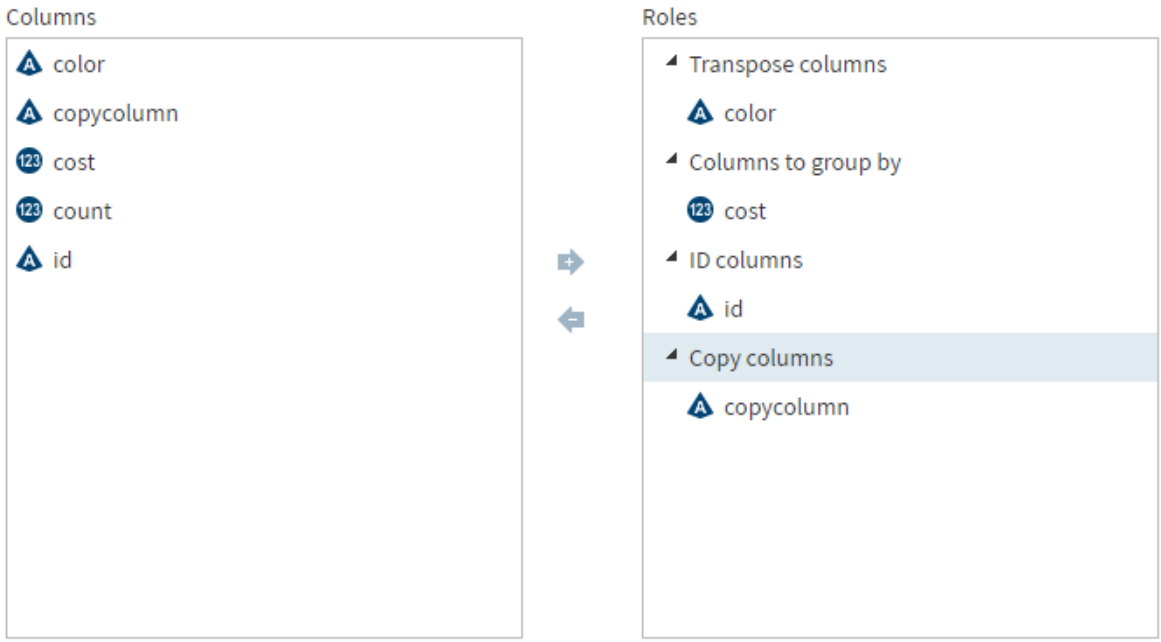
   In the target table, the group-by columns appear on the left, followed by generated columns that identify each row for the transposed column that it came from. Any copied columns follow. The ID columns appear on the right, in alphanumeric order. If an ID column is not specified, then numbered columns are created for each transposed column.

   In the following images, a source table is transposed to yield a target table.

   Source Table

	color	id	count	cost	copycolumn
1	Green	A100	20	2.7	one
2	Green	B40	15	4.4	two
3	Green	B50	25	1.3	three
4	Green	A200	30	5.5	four
5	Red	A300	15	7.1	five
6	Red	B400	30	1.7	six
7	Violet	B20	40	2.66	seven

   Column Selections in the **Transpose Data** Task



Transposed Data in the Target Table

	cost	dl_name_	copycolumn	a100	a200	a300	b20	b40	b400	b50
1	1.7	color	six						Red	
2	5.5	color	four		Green					
3	7.1	color	five			Red				
4	2.66	color	seven				Violet			
5	1.3	color	three							Green
6	2.7	color	one	Green						
7	4.4	color	two					Green		

When your column selections are complete in the **Transpose Data** task, click **Next**.

4. In the Target Table task, click to open a data source or select a table from your most recently accessed data source. Inside a data source, you can click icons to create a new target table, select from a list of recently accessed tables, or return to data sources. You can also view data in tables and view profile reports for profiled tables.

   To learn more about data sources, tables, and viewing data, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16,

   After you select your target table, click **Next**.

5. In the **Results** task, click **Start Transposing Data** to run your directive. During execution, click icons to view code, log, errors, and results as they become available. Click **Save** or **Save As** to store the directive and access it in Saved Directives.

   You can also track the status of the directive in Run Status.

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 154.

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*.

Chapter 5

Profile Data

Overview of Profile Directives

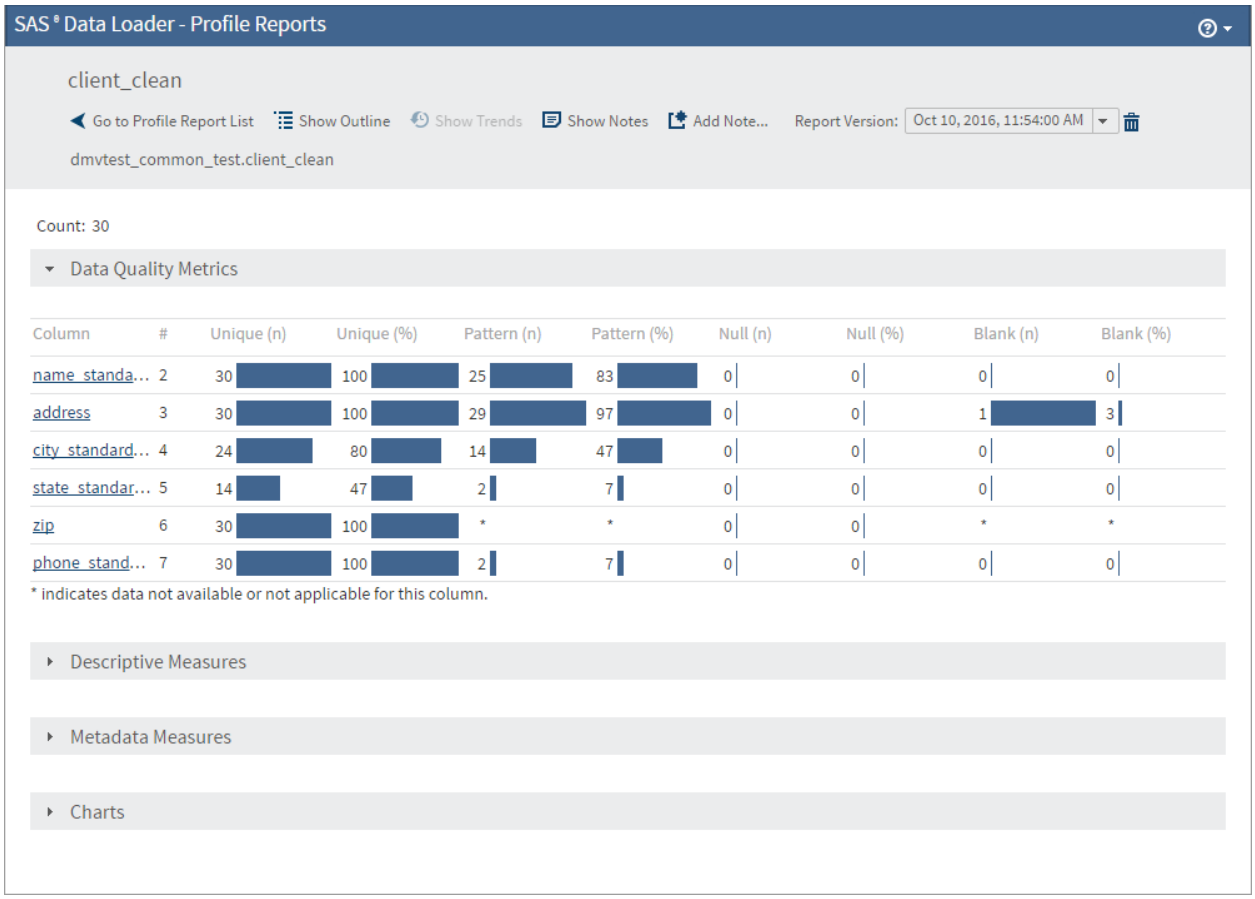
Data profiling directives assess the composition, organization, and quality of Hadoop tables. The profile reports help you recognize patterns, identify scarcity in the data, and calculate frequency and basic statistics. Data profiling can also aid in identifying redundant data across tables and cross-column dependencies. All of this information is essential to achieving the goal of understanding your data. Profile reports indicate the types of transformations and data cleansing that are needed prior to analysis.

The Profile Data directive generated profile reports for one or more Hadoop tables. The reports display data, column information, and measurements of data quality.

The Saved Profile Reports directive displays profile reports and manages profile reports.

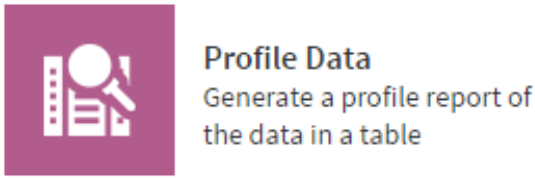
Note: Profile reports can be opened in conjunction with their respective Hadoop tables within many directives. If a table has a profile report, then the Source Table and Target Table tasks enable the display of that report.

Here is an example of a profile report:



Profile Data Directive

Introduction



Use the Profile Data directive to generate profile reports for one or more tables in Hadoop. You can select a subset of the columns that you want to include in the profile report.

Profile reports indicate how source tables can be improved to maximize performance and analytical quality. For example, if you learn that a primary key column has mostly unique values, you could decide to subset columns in the target.

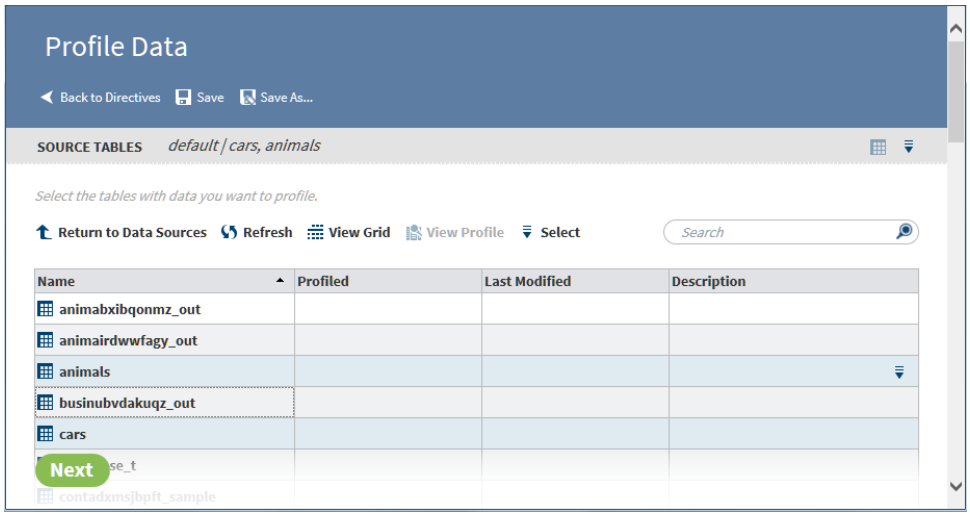
The **Profiles** panel of the Configuration window enables you to change the default behavior of new profile directives in order to improve performance. For more information, see "Profiles Panel" on page 149.

To learn more about the content of profile reports, see "About Profile Reports" on page 94.

Example

Follow these steps to use the Profile Data directive:

1. On the SAS Data Loader for Hadoop directives page, click the Profile Data directive. The directive opens in the **Source Table** task. By default, the **Source Table** task displays the source tables in your last-accessed data source. Click one or more source tables, and then click **Next**.



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

   If a profile already exists for a table, PROFILED appears next to the table name. You can view the existing profile by selecting the table and clicking **View Profile**.

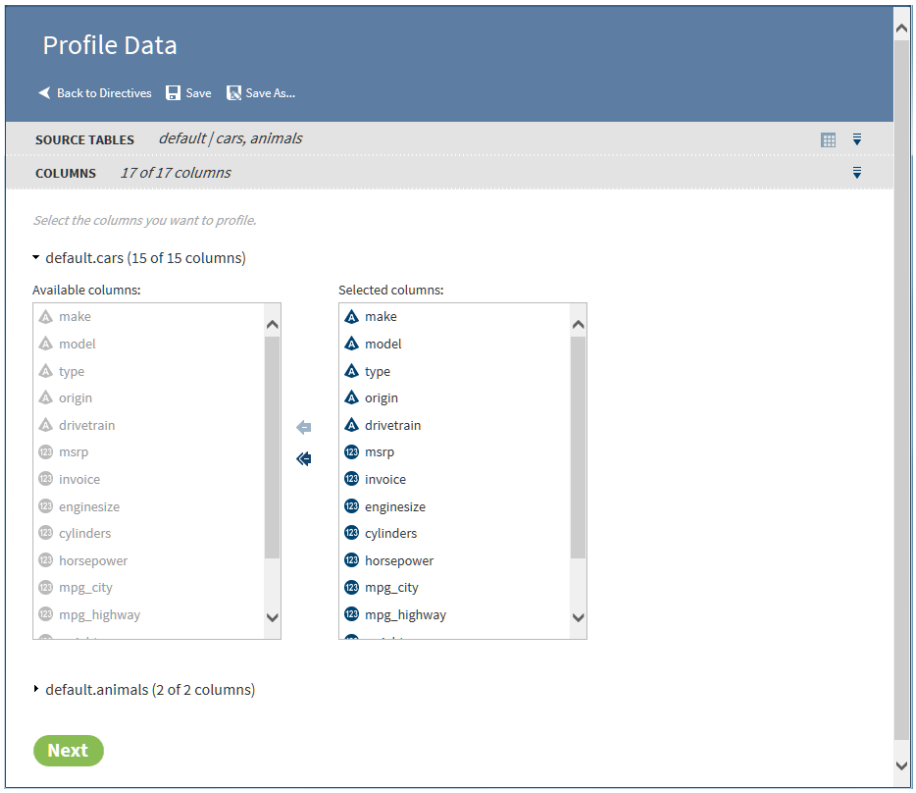
   The **Select** menu (**Select**) provides several options to make selecting tables easier:

   - **Select All New Tables**: Automatically selects all new tables in the current data source. The identification of new tables, and the duration of their new status, is determined in the General Preferences panel of the Configuration window.
   - **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.

   To learn more about data sources and the **Source Tables** task, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16.

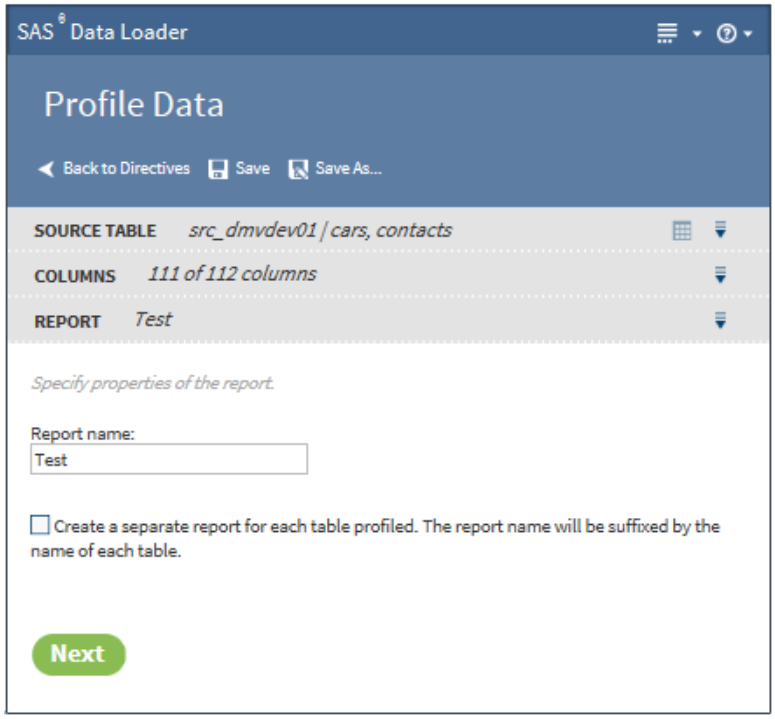
2. In the **Columns** task, select the columns that will appear in the profile report.



TIP To improve performance, avoid selecting columns such as primary keys that have a large percentage of unique values.

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.

3. In the **Report** task, enter a name for the profile report in the **Report name** field, and then click **Next**.



If you selected multiple tables and want a separate report for each table, click **Create a separate report for each table profiled**.

4. In the **Result** task, click **Create Profile Report**. As the directive runs, you can access the code, log, errors, and the profile report as they become available. To learn about the profile report, the Profile Report. see "Saved Profile Reports" on page 94.

   The following actions are available:

Usage Notes

Table Length Name Limit

Hive tables have a maximum table name length of 132 characters. Many of the directives in SAS Data Loader for Hadoop 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 32-character name length limit for tables and columns.

Performance Tips for Profiles

Follow these tips to improve the performance of your profile directives:

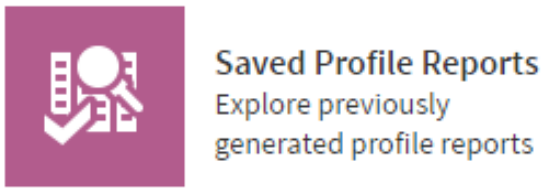
- Profile a subset of columns based on need or benefit.
- The processing of frequency distributions decreases performance. You can limit the number of frequency distributions that are performed on each column in the **Profiles** panel of the Configuration window.
- Avoid profiling primary key columns and any columns that contain a significant percentage of unique values.
- Primary/foreign key analysis and redundancy analysis require at least a partial frequency distribution of primary key columns.
- To ensure that no column is profiled if it contains too many unique values, set the cutoff value in the **Profiles** panel.
- Consider profiling a sample of the source data. A sample table could consist of every 10th row in the original table. To create a sample table, use the Delete Rows directive.

About Profile Results

The Profile Data directive treats varying-length character types as fixed-length character types. The length, minimum length, and maximum length of varying-length character columns is reported as the size of the longest value found in the column. NULL values are ignored.

Saved Profile Reports

Introduction



Use the Saved Profile Reports directive to view and manage profile reports that were generated by the Profile Data directive.

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.

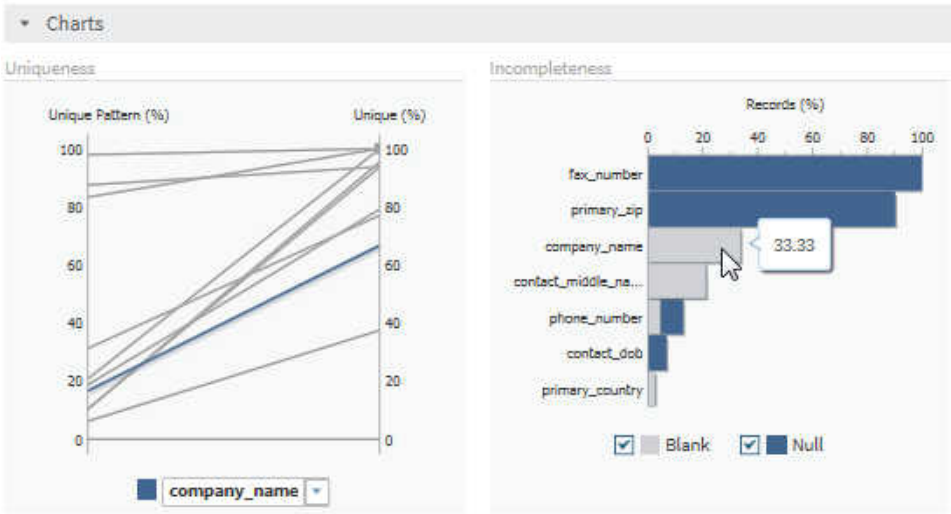
Data Quality Metrics
measurements of data quality for the columns in the table. Measurements include information about the uniqueness of column values, pattern analysis results, and completeness information, including null or blank values.

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. Columns that have a high number of unique or distinct values can be primary key candidates.

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:

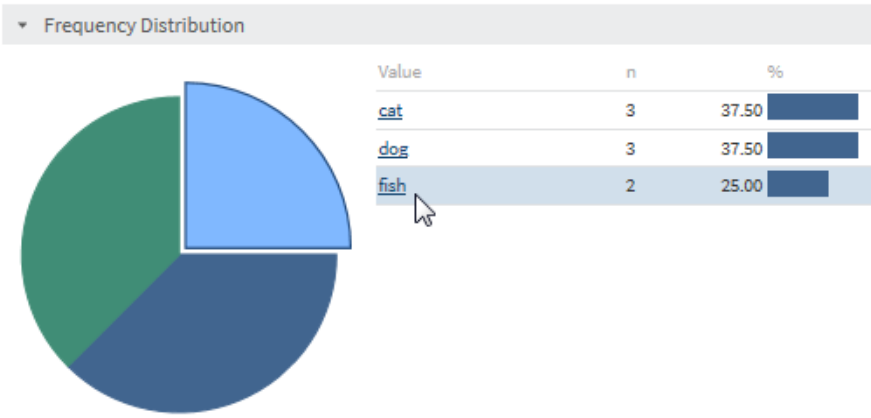
Count
the total number of rows in the table that was profiled.

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.

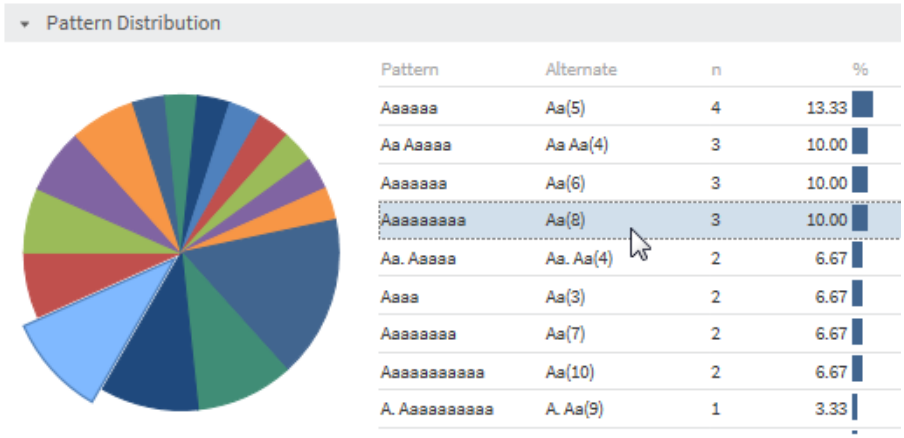
When you select a value from the list, the associated section of the pie chart is highlighted.

By default, the profile report for a given table collects a maximum of 1000 frequency distributions. Your SAS administrator can change this system default. You can change the value for your profile reports using the Profiles panel of the Configuration window .



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 structure of the data and indicates whether each character is uppercase, lowercase, or numeric.

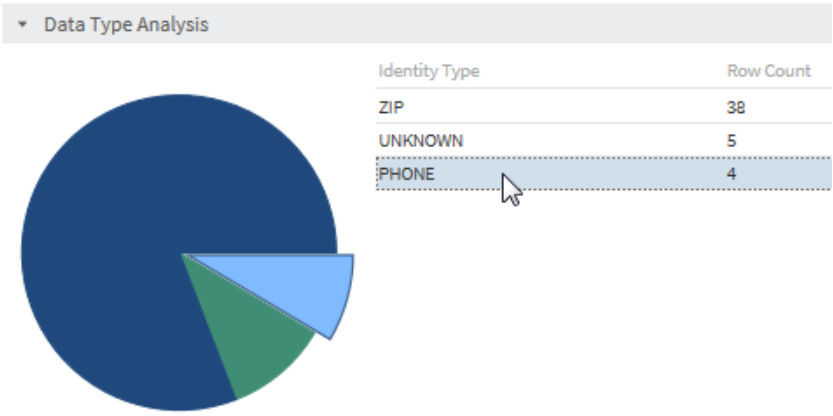


Outliers

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 149.

Data Type Analysis

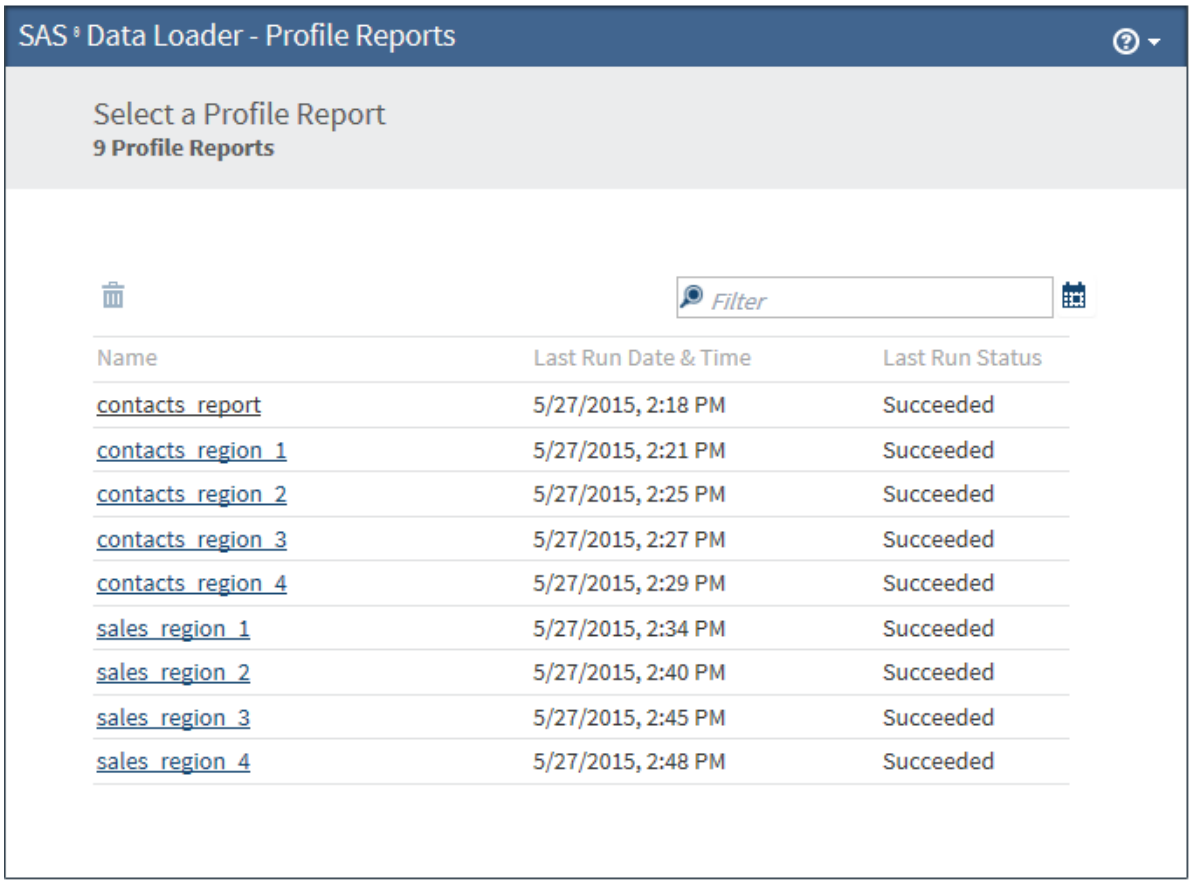
a listing of possible types of data for the information in the column, as determined by data type analysis. Results for data type analysis are available only for columns that contain contact information (such as name and address).



Open Saved Profile Reports

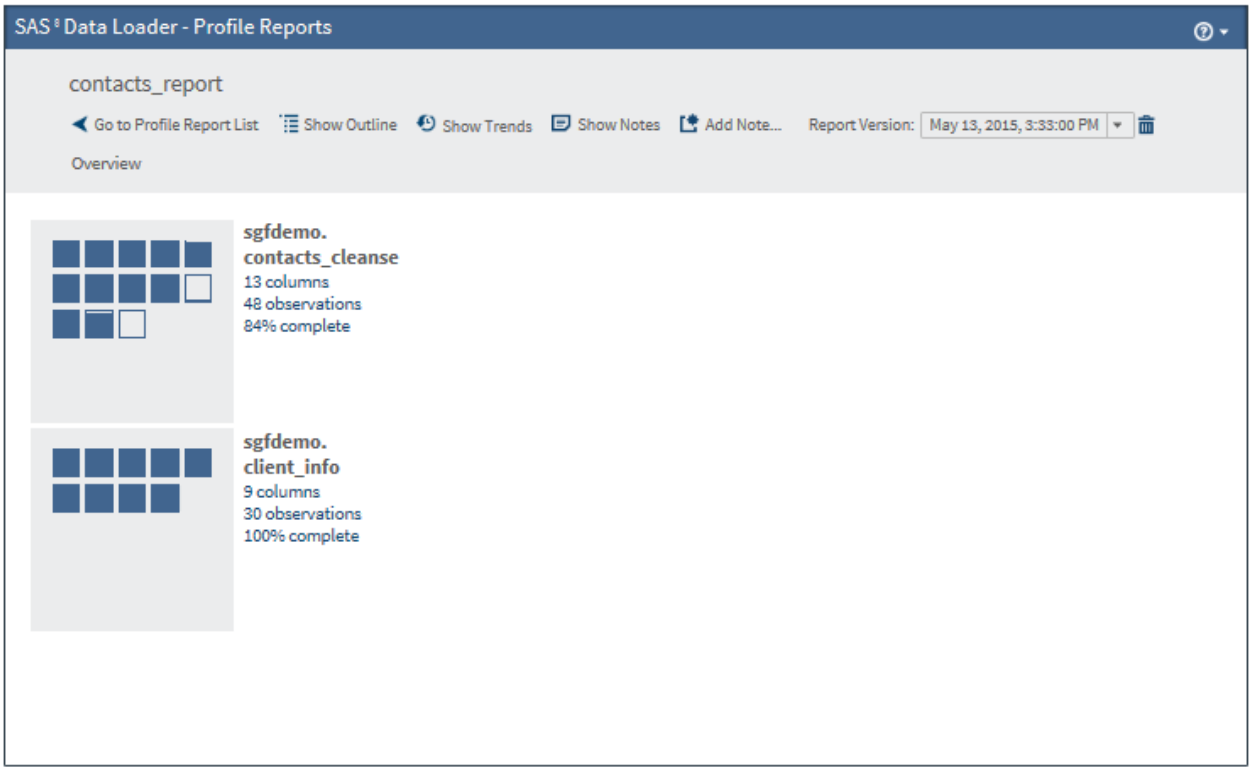
Follow these steps to open a saved profile report:

1. In the SAS Data Loader for Hadoop 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:

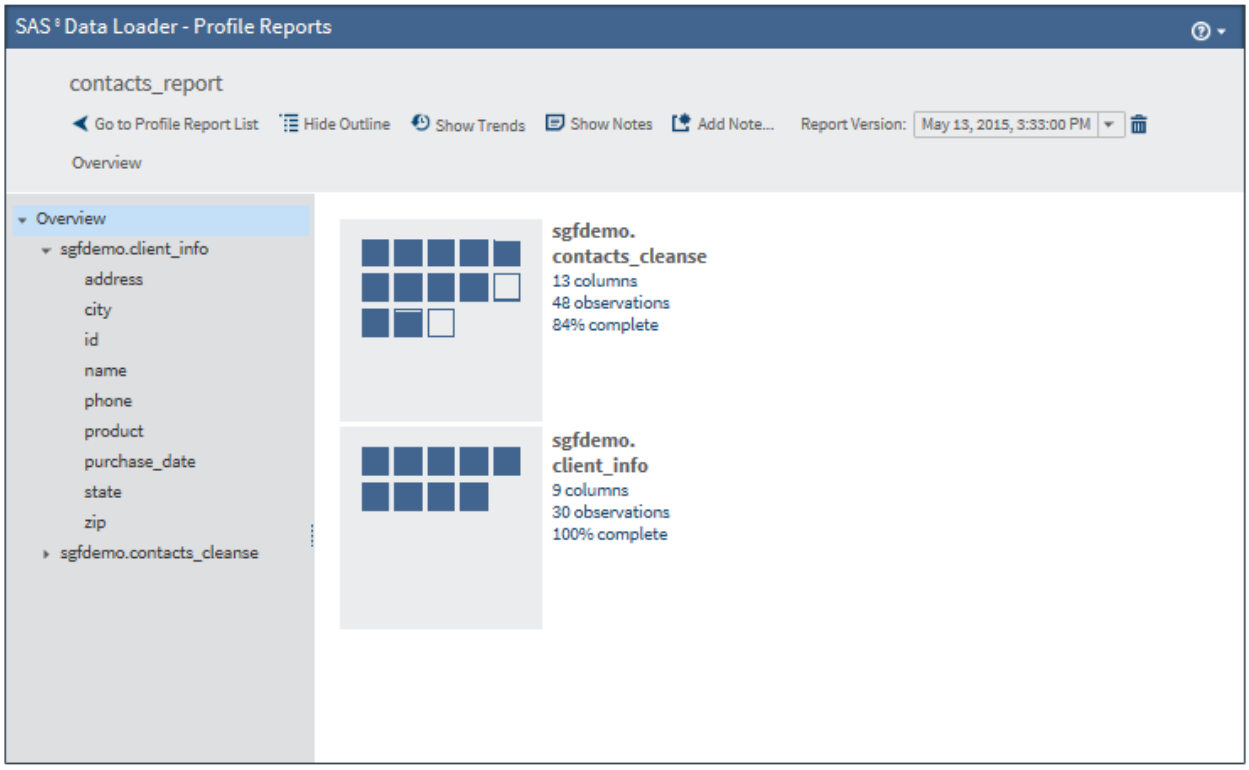


Note: Any profile directive that runs longer than five days is deleted from the Select a Profile Report page.

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:

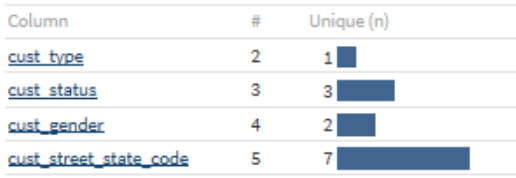


The following actions are available:

Go to Profile Report List
returns you to the Profile Report List.

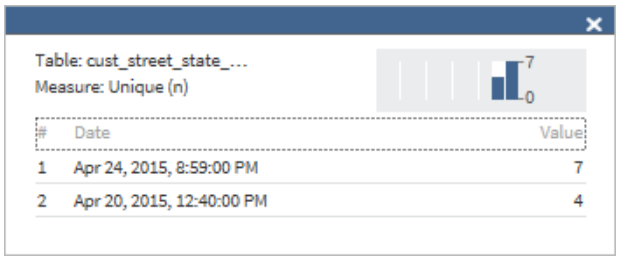
Show or Hide Outline
displays or hides the outline in the left pane.

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:

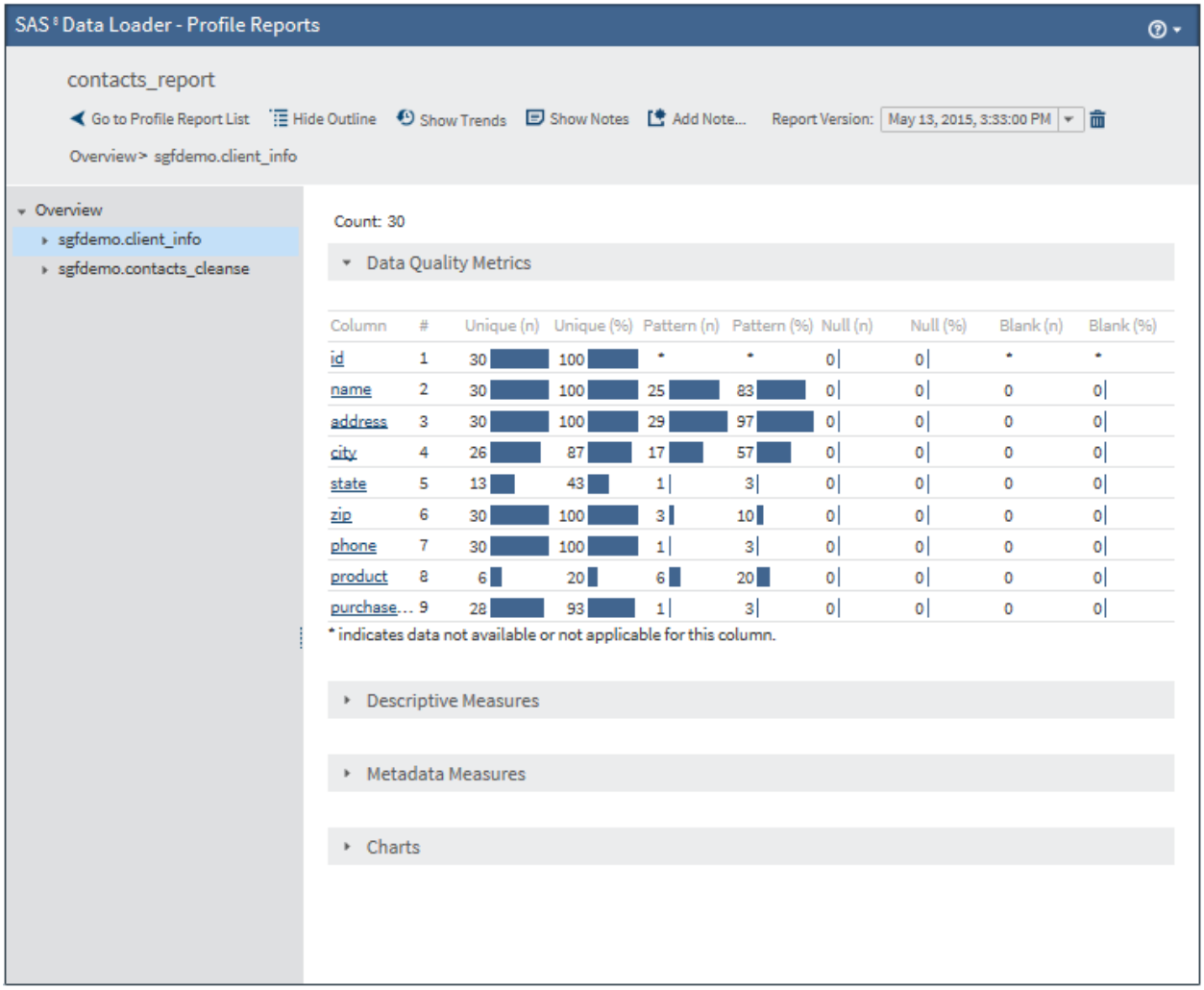


Show or Hide Notes
displays or hides notes in the right pane. You can filter the notes by entering a text string into the filter field.

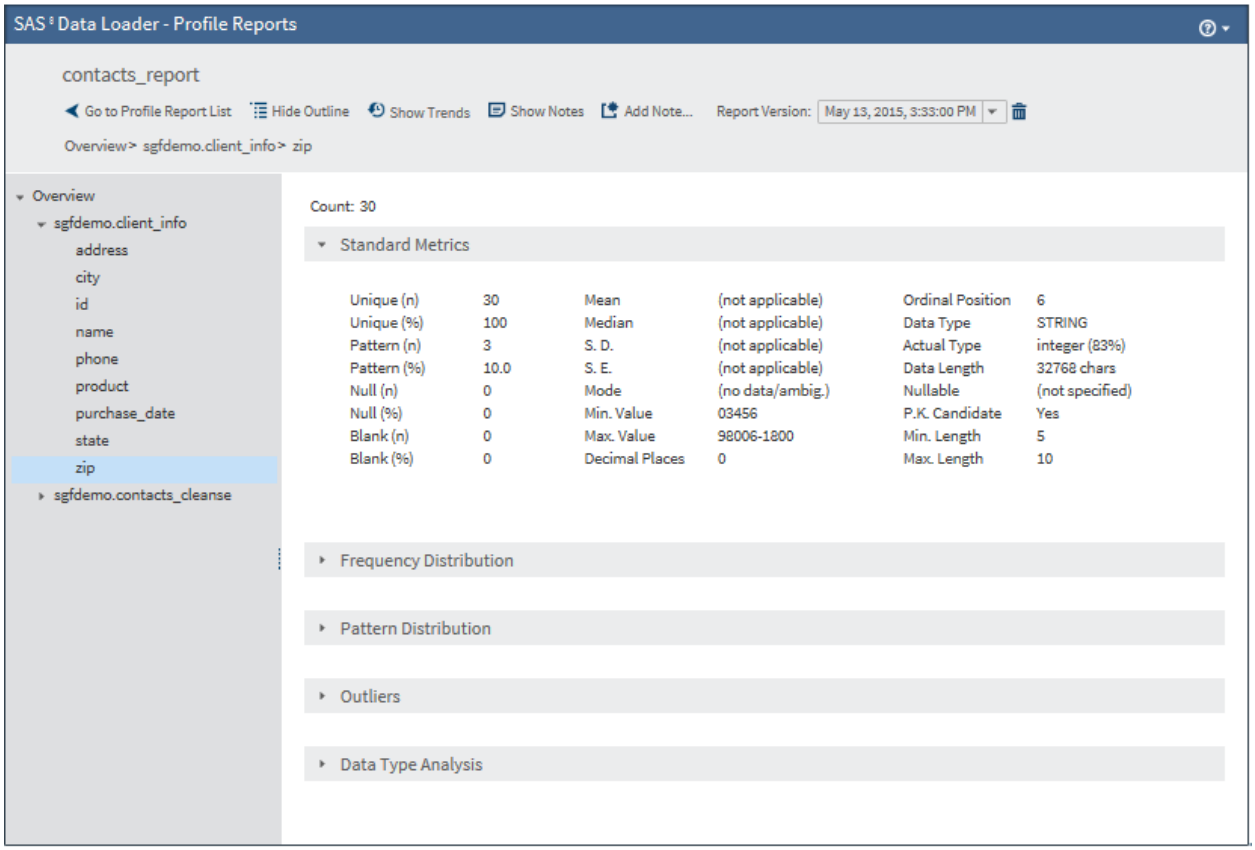
Add Note
opens a dialog box in which you can add a note.

Report Version
enables you to select the version of the report by date.

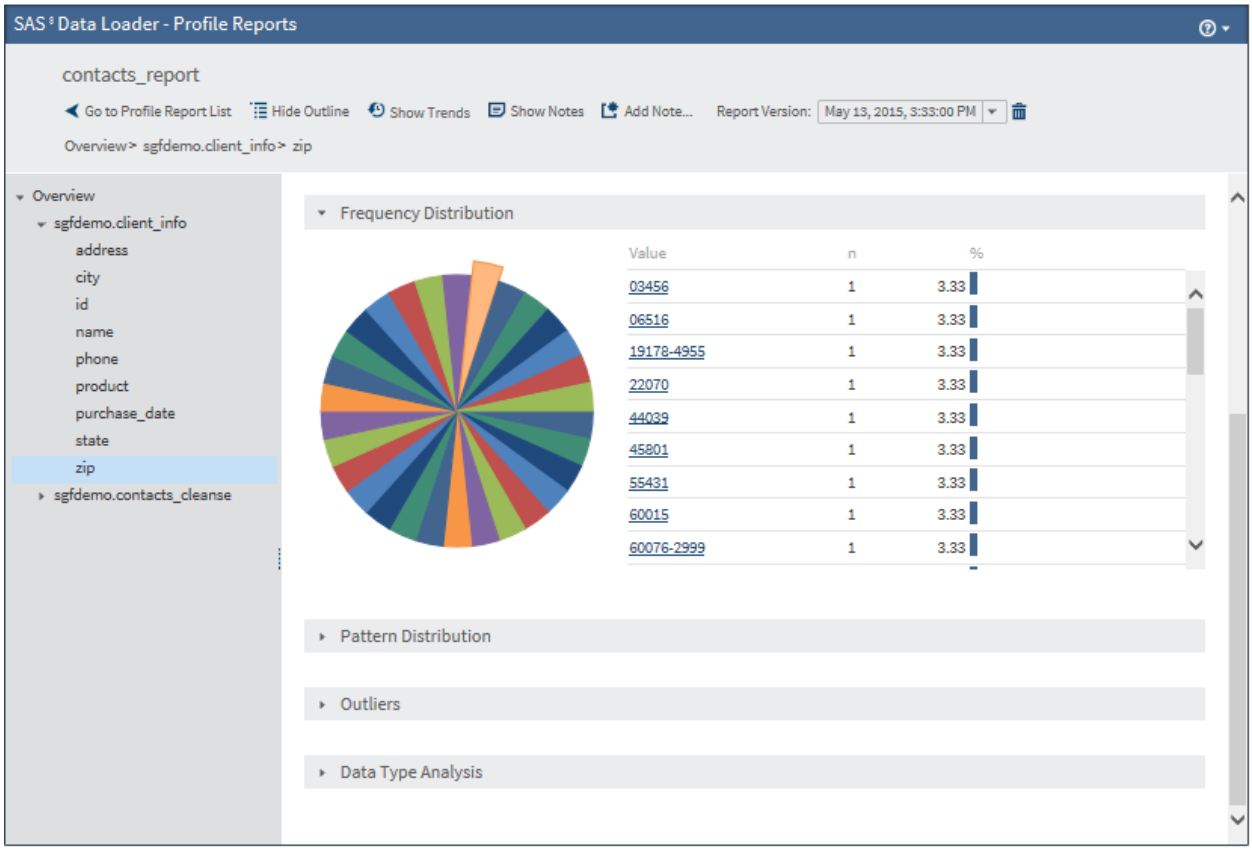
6. Select a table in the **Overview** pane or click directly on the table icon to display detailed information in the right pane. The Data Quality Metrics are displayed by default.



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 the **Frequency Distribution** icon displays the following metrics.



Clicking links in the detail view opens SAS Table Viewer.

Chapter 6

Copy Data To and From Hadoop

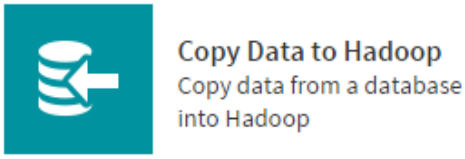
Overview of the Copy Data Directives

The directives Copy Data to Hadoop, Import a File, Copy Data from Hadoop, and Load Data to LASR enable you to move data between Hadoop and your file system, relational databases, or SAS libraries. The SAS libraries can use any SAS LIBNAME engine that is installed into SAS Workspace Server. You can also copy data to SAS LASR Analytic Server.

The copy directives require the definition of database or server connections in SAS Metadata Server. These connections were originally specified by administrators during the deployment of SAS Data Loader for Hadoop. To add, modify, or delete database or server connections, contact your SAS administrator.

Copy Data to Hadoop

Introduction



The Copy Data to Hadoop directive copies data into Hadoop from relational databases and from SAS data sources. SAS data sources include SAS LASR Analytic Server, SAS libraries, and any data source (including relational databases) that can be connected to using SAS/ACCESS or a SAS LIBNAME engine. When you copy data to Hadoop from relational databases outside of a SAS library, you use JDBC connections and the Sqoop bulk data transfer application in Hadoop.

SAS data sets are copied to Hadoop from a SAS library. SAS libraries are defined in the SAS middle tier on instances of SAS Workspace Server. Access to SAS libraries is determined by administrators. Usually, all users receive access to at least one SAS library.

To copy a SAS data set to Hadoop, you often begin by uploading the data set from your local file system to a SAS library. To upload SAS data sets, use the "Import a File" directive. When the SAS data set is available in a SAS library, use Copy Data to Hadoop to load the data set into Hadoop.

To copy delimited text files into Hadoop, use the Import a File directive.

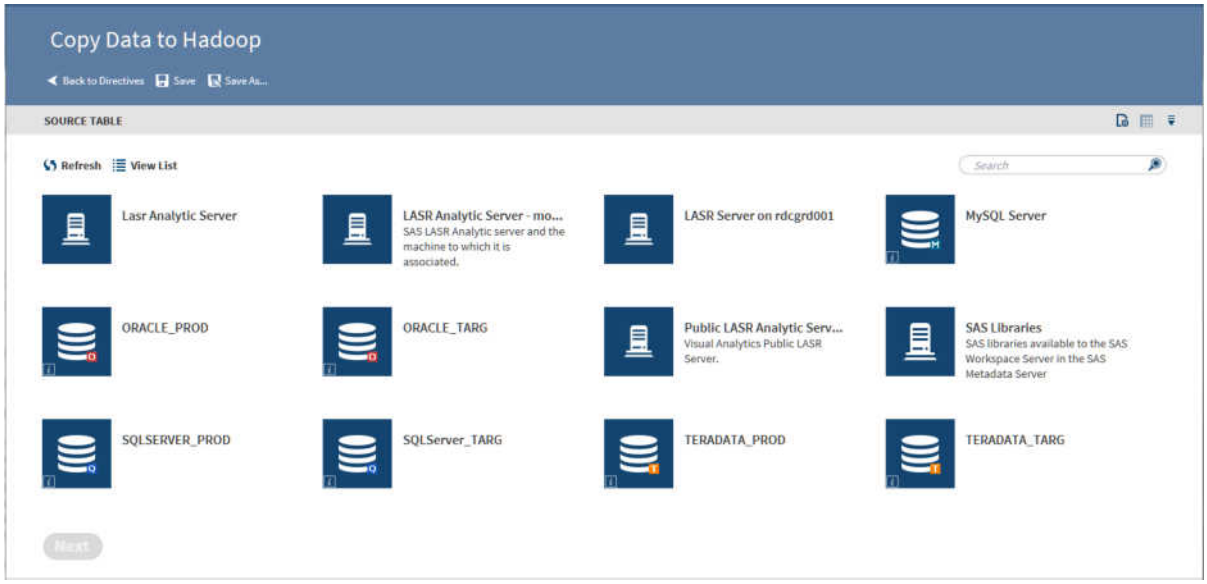
Example

Follow these steps to copy data into Hadoop from SAS or a database:

1. On the SAS Data Loader for Hadoop directives page, click the Copy Data to Hadoop directive.

   Note: If you do not see the Copy Data to Hadoop icon, then you do not have permission to access that directive. To change your permissions, contact your SAS administrator.

   The directive opens on the **Source Table** task.



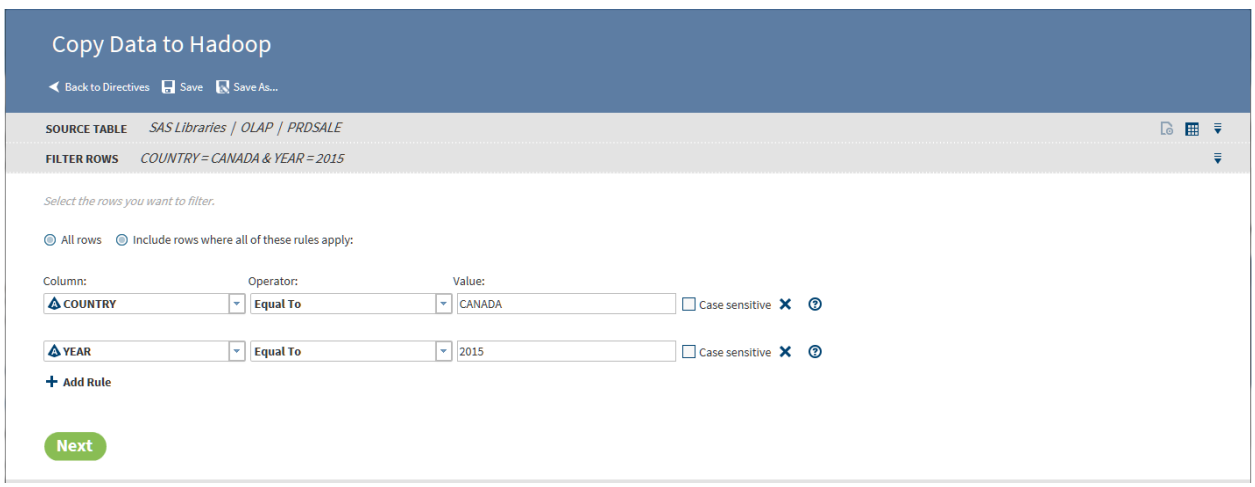
Click an icon to display its data sources, and then click a data source to display its source tables.

Click a source table to copy into Hadoop, and then click **Next** to display the **Filter Rows** task.

TIP In database icons, click the lower left corner to test the connection to the database.

To learn more about data sources and source tables, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16.

2. Use the **Filter Rows** task to copy selected rows to the target. 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.

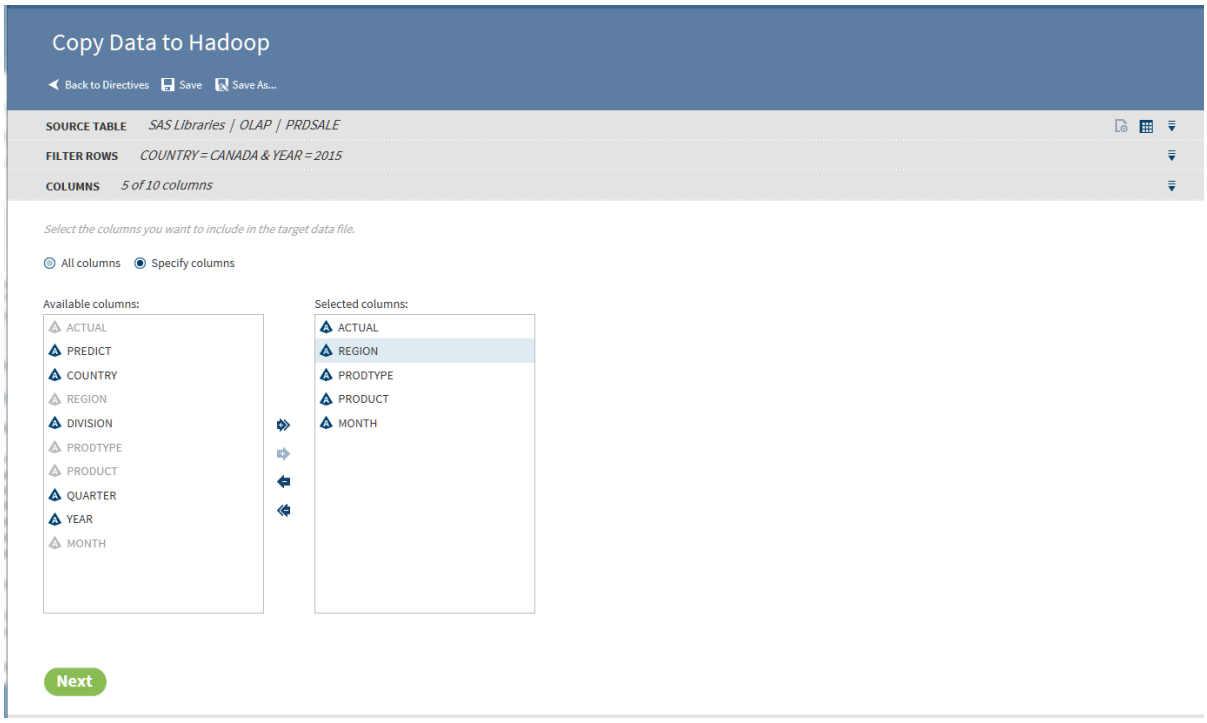
   Note: If the table for which you are defining a filter is in the OTHER database format, the database might not support all operators. You should use only those operators that are supported by your database in the filter.

c. Enter a **Value** to complete the filter rule.

d. To define another filter rule, click **Add Rule**.

e. When your filter rules are complete, click **Next** to display the **Columns** task.

3. Use the **Columns** task to copy selected columns to the target.



In the **Columns** task, all columns are selected for copying to Hadoop by default.

Use the following controls to select columns:

de-select the highlighted columns

de-select all columns

select the highlighted columns

select all columns

When all columns are selected, click **Next** to display either the **Options** task or the **Target Table** task. The **Options** task is displayed only when you load data into Hadoop from a database. If the **Target Table** task is displayed, then skip the next step.

4. Use the **Options** task to specify the number of processes that your database connection will use to copy data into Hadoop.

The values in the **Options** task should not be changed unless you have advanced knowledge of database operations.

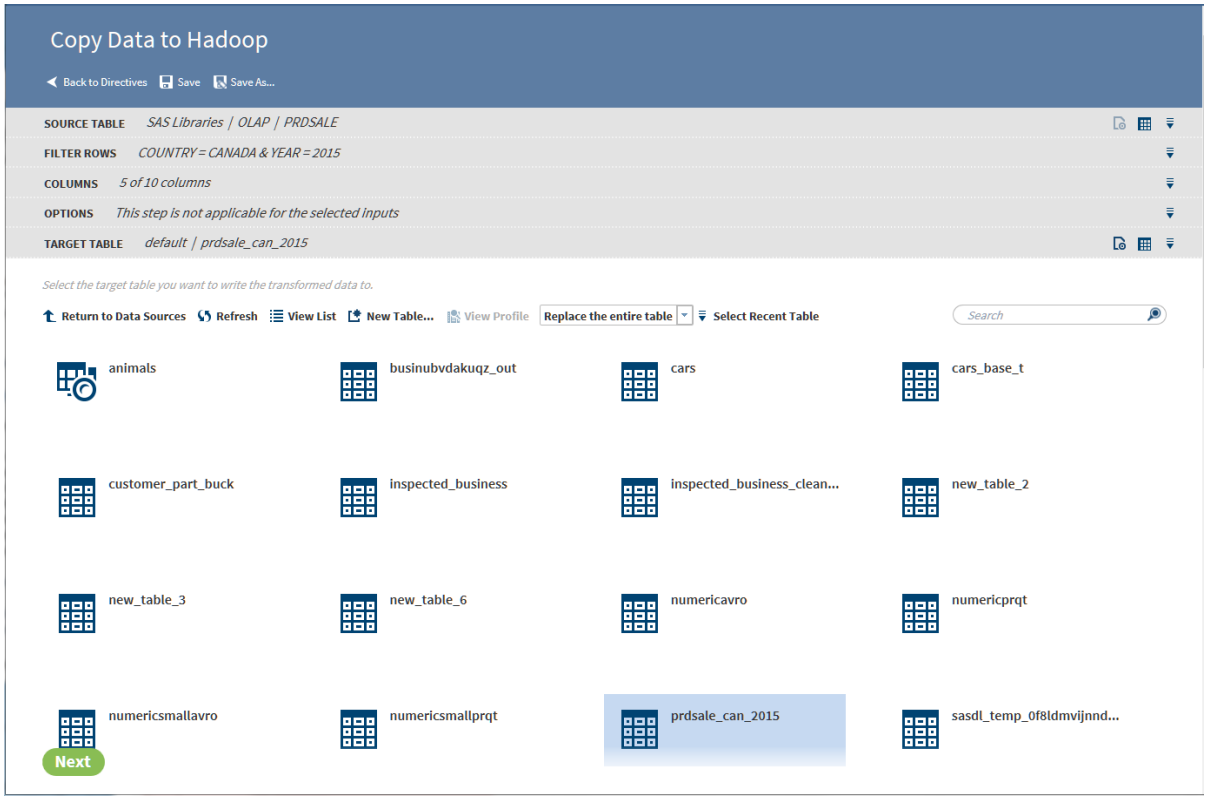
CAUTION:

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

After you have selected a number of processes, click **Next** to display the **Target Table** task

5. Use the **Target Table** task to select a destination table in Hadoop. By default, the **Target Table** task displays the target tables in your most recently accessed data source. If the task displays a list of data sources, click on a data source to display the available target tables.



To create a new target table in Hadoop, click **New Table** and enter a table name in the New Table window. The new table appears in to **Target Table** task.

To write data into an existing target table, select a target table, and select either **Replace the entire table** or **Insert into the existing table**.

To examine and perhaps change the default output table format or storage location, click **Edit advanced options** . To learn more about the advanced options, see "General Preferences Panel" on page 145.

To learn more about target tables, see "Get Started with the Target Table Task" on page 18.

6. When your target table is ready, click **Next** to display the **Code** task. Click **Edit Code** as needed or click **Next** to display the **Result** task.

CAUTION:

Code edits are intended to be used only to support advanced features. Code edits are not needed or required under normal circumstances.

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

   Click **View Results**, **Log**, **Code**, and **Error Details** as they are displayed during the execution of the directive.

8. Click **Save** to retain your directive. Use Saved Directives or Run Status to access your directive in the future.

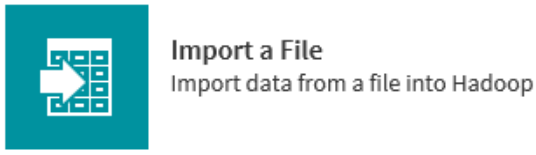
Usage Notes

- If LDAP is used to protect your Hadoop cluster, then you cannot copy data from a database management system into Hadoop. This limitation is imposed by the Oozie workflow scheduler in Hadoop.
- 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 154.
- 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 workflow scheduling application that is used to execute Copy Data to Hadoop directives. Refer to the Oozie log for additional troubleshooting information.
- The following restrictions for Teradata apply only when you copy data from Teradata to Hadoop using a JDBC database connection and the Scoop bulk transfer tool in Hadoop. These restrictions do not apply when you copy data from Teradata to Hadoop using a SAS LIBNAME engine:
  - In Cloudera 5.2 or later, the Teradata source table must have a defined primary key, or you must specify a distribution column in the 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 create an empty table in the desired data source. At that point, you can use the **Insert into the existing table** option in the Target Table task to copy a Teradata table into the new table.
- 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' )`**. in the Example section.
- When copying data from Oracle, note that Oracle table names must be uppercase.

Import a File

Introduction

About the Directive and Delimited Source Files



Use the Import a File directive to copy delimited source files into target tables in HDFS and register the targets in Hive. Delimited source files contain data that can be loaded into Hadoop as tables or as delimited text files.

The delimited source files contain unformatted tabular data. Data values are separated by delimiter characters. Newline characters indicate the ends of rows. The first row in the source file can specify the names of the columns. Delimited source files can be generated by software such as Microsoft Excel and relational databases.

Delimited source files are imported into Hadoop from SAS libraries. The SAS libraries are managed by instances of SAS Workspace Server in the SAS middle tier. Your SAS administrator assigns access permissions to SAS libraries.

About Upload and Download

The Import a File directive provides an upload capability that copies delimited source files from your local file system to a SAS library. You upload first, and then you import into Hadoop. The download capability enables you to copy from a SAS library to your local file system.

In addition to uploading delimited source files, you can also use the Import a File directive to upload SAS data sets. After you upload, you can then copy SAS data sets into Hadoop using the "Copy Data to Hadoop" directive.

Features of the Directive

After you select a delimited source file, the Import a File directive samples the source data and generates default column definitions for the target. You can then edit the default column names, types, and lengths.

To simplify future imports, you can save column definitions to 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 target in Text format in Hadoop. Importing to Text format uses an efficient bulk-copy algorithm.

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).

Prerequisites

Delimited source files are imported into Hadoop SAS libraries only.

Ensure that your delimited source files are structured as follows:

- Instances of the delimiter character that are embedded in data values must be enclosed in quotation marks (`"`).
- Quotation marks that are embedded in data values must be preceded by a second quotation mark (`""`).
- Newline characters are required to indicate the end all rows.
- To minimize errors in Hadoop, minimize or eliminate newline characters embedded in data values.

Example

Follow these steps to use the Import a File directive:

1. On the SAS Data Loader directives page, click **Import a File**.
2. To upload a SAS data set or a delimited source file into a SAS library, click **Upload** and select a file from your local file system. Click **OK** to upload the file, and then confirm that the file is available in the SAS library.

   To load the SAS data set into Hadoop, click **Back to directives** and click **Copy Data to Hadoop**.

   To import a delimited source file, continue to the next step.
3. In the **Source File** task, select a source file and click **Next** to display the **File Specification** task.

   Click folders to open them, and click **Back** to move up one level.

   To copy a selected file out of a SAS library and into your local file system, click  and select **Download**.
4. In the **File Specification** task, specify a delimiter, select a value for **Input file delimiter**. To see the current delimiter and the existing columns, click **View File**.

   Note:

   - 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.
   - If you select **Other** as the delimiter, then enter into the corresponding 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. Input delimiters must have ASCII character codes that range from 0 to 127, or octal values that range from \000 to \177.
5. 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**.

   Note:

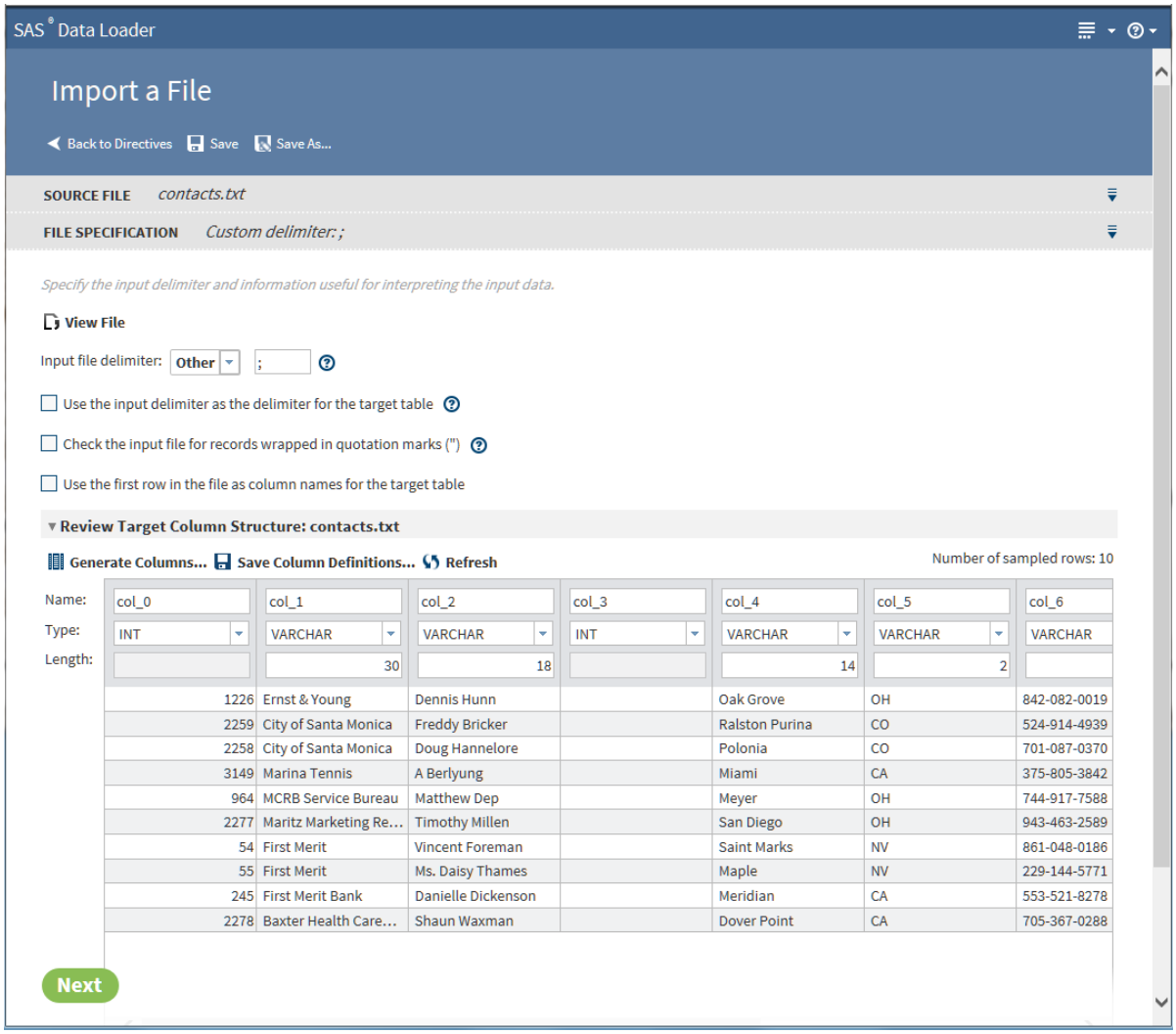
   - The bulk-copy operation does not analyze or validate source data in any way. For example, the directive does not ensure that each row has the expected number of columns.

- The bulk-copy operation is used only if the next two options are not selected.
- The format of the target is always TEXTFILE. The TEXTFILE format is used even if another format is specified in the **Target Table** task or by default in the **General Preferences** panel of the **Configuration** window.
- 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`**.

6. If the delimiter appears in source table cells, select **Check the input file for records wrapped in quotation marks (")**.
7. 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**.
8. Click **Review Target Column Structure** to display a sample of the target table. The target columns are displayed with default column names, types, and lengths (based on type.) You can edit the column definitions as needed, and save a column definitions file. If you have an applicable column definitions file, you can replace the default column definitions, as described in Step 9.

   To manually edit column definitions and save a column definitions file, edit the fields **Name**, **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 for that column in 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.

9. When your manual column edits are complete, 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. All column definitions files are stored on your local file system. Click **OK** to save your column definitions to the designated file.

10. 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:

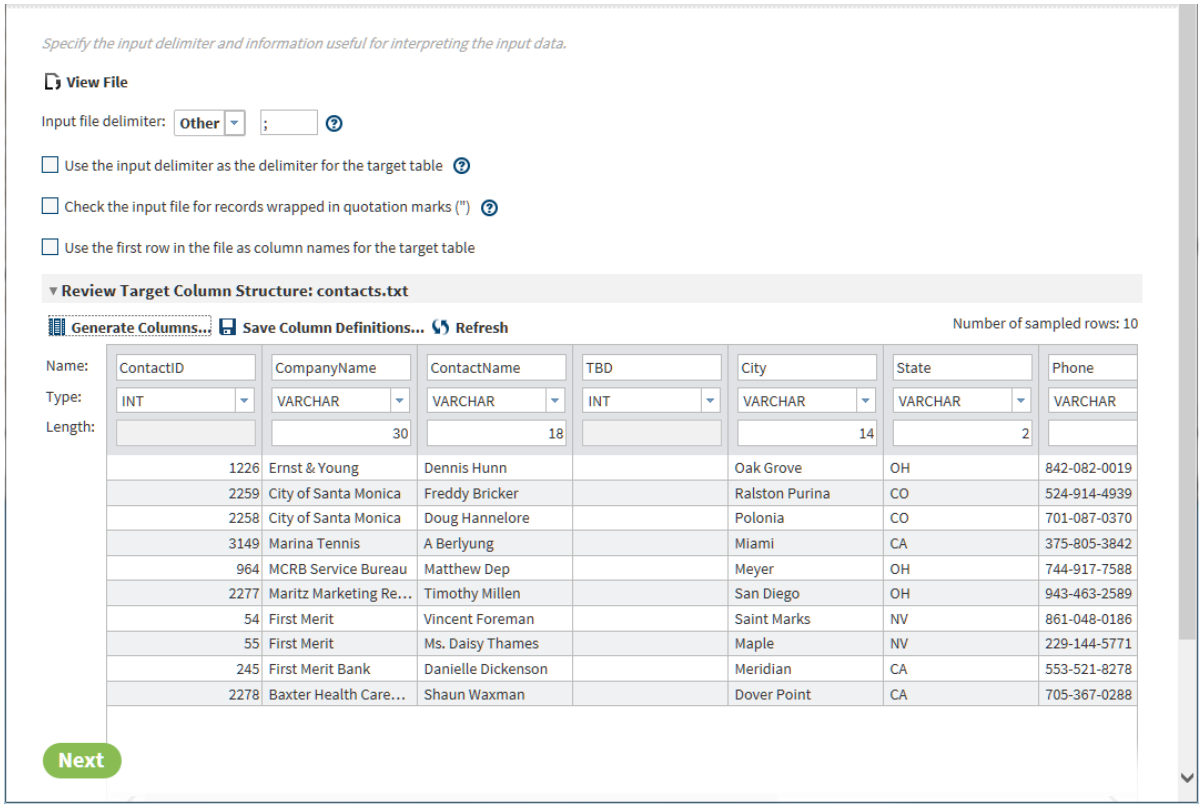
a. Click the **Generate Columns** icon.

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 libraries, 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**.

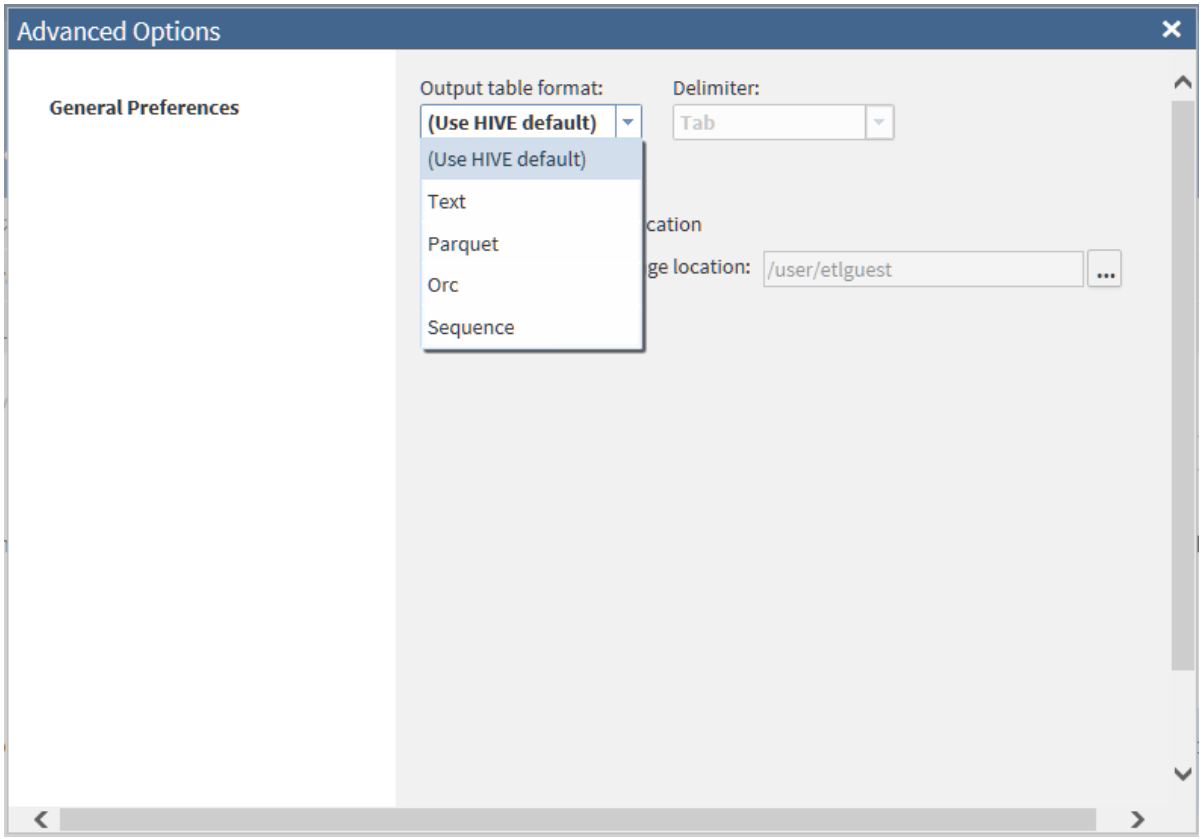
11. In the **Target Table** task, click to open a data source and select a target, or click Select Recent Table and choose a target. Existing targets are overwritten entirely when you run your job.

    To name a new target table, select a data source and click the **New Table** icon, enter the new table name, and click **OK**.

12. 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 , 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 5, then the target will always receive the Text format, regardless of the selections in the Advanced Options and Configuration windows.



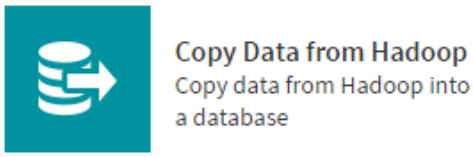
To save the table data to a non-default Hive storage location, click **Specify alternate storage location**, and then click ... . You need appropriate permission to store your imported table or file to a non-default Hive storage location.

When your target selection is complete, click **Next**.

13. 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 **View Results**, **Code**, **Log**, and possibly **Error Details** to learn more about your job.
14. Click **Save** or **Save As** to retain your job.

Copy Data from Hadoop

Introduction



The Copy Data from Hadoop directive copies data from Hadoop into relational databases or SAS. SAS targets include SAS LASR Analytic Server, SAS libraries, and any data source (including relational databases) that can be connected to using SAS/ACCESS or a SAS LIBNAME engine. you can also copy data directly into relational databases using JDBC connections and the Sqoop bulk data transfer application in Hadoop.

Example

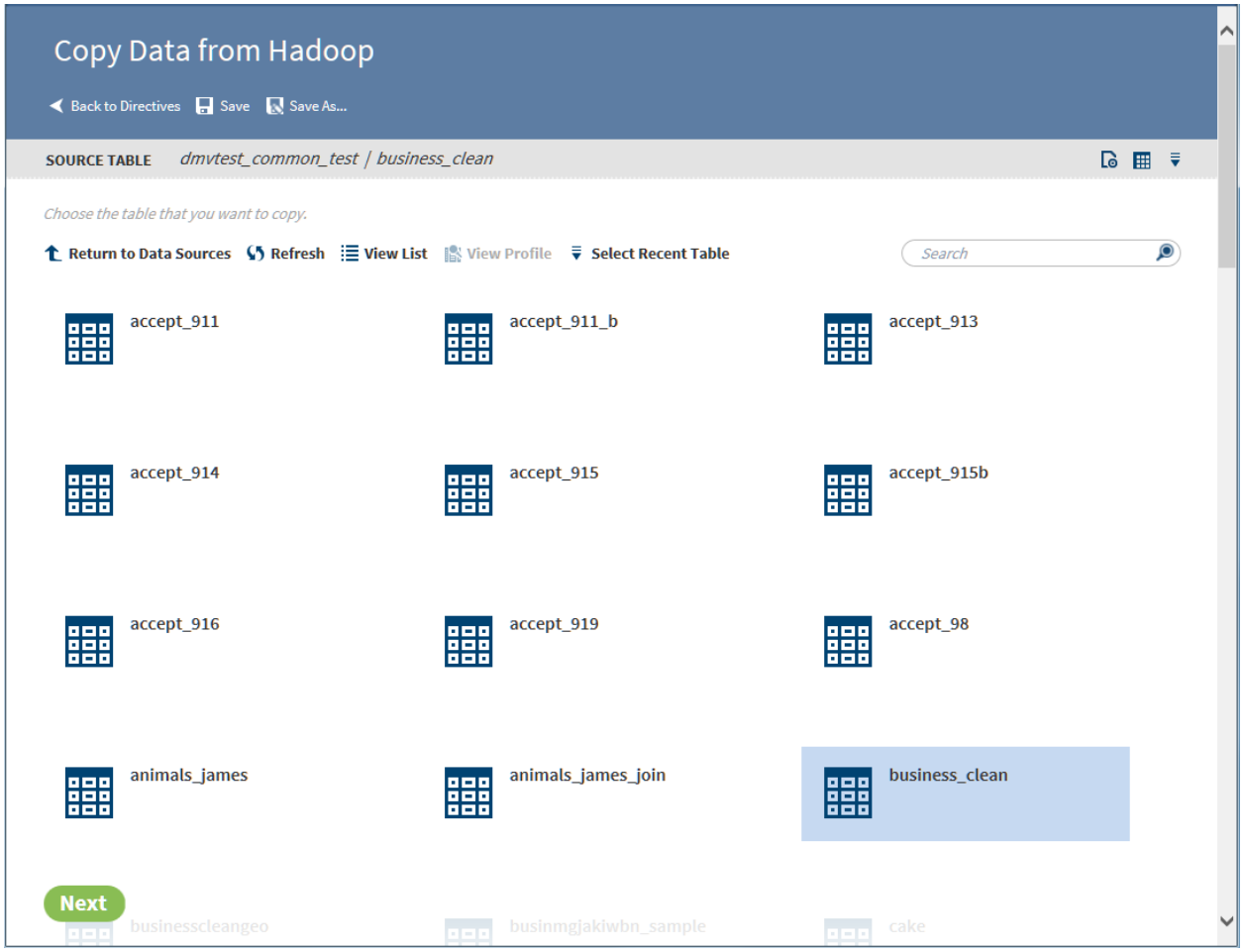
Follow these steps to copy data from Hadoop into SAS or into a database management system:

1. On the SAS Data Loader for Hadoop directives page, click the Copy Data from Hadoop directive.

   Note: If you do not see the Copy Data from Hadoop icon, then you do not have permission to access that directive. To change your permissions, contact your SAS administrator.

2. Use the **Source Table** task to select a Hadoop source table to copy to a database or to SAS.

   Either click a source table, click a data source, or click **Return to Data Sources** to choose a source table from a different data source.



To display and change the maximum length of SAS character columns, select a source table and click **Edit advanced options** . In the Directive Settings window, the value that is initially displayed is the default length. You can specify a different length for the current directive without changing the default. For more information, see "Change the Maximum Length for SAS Character Columns" on page 154.

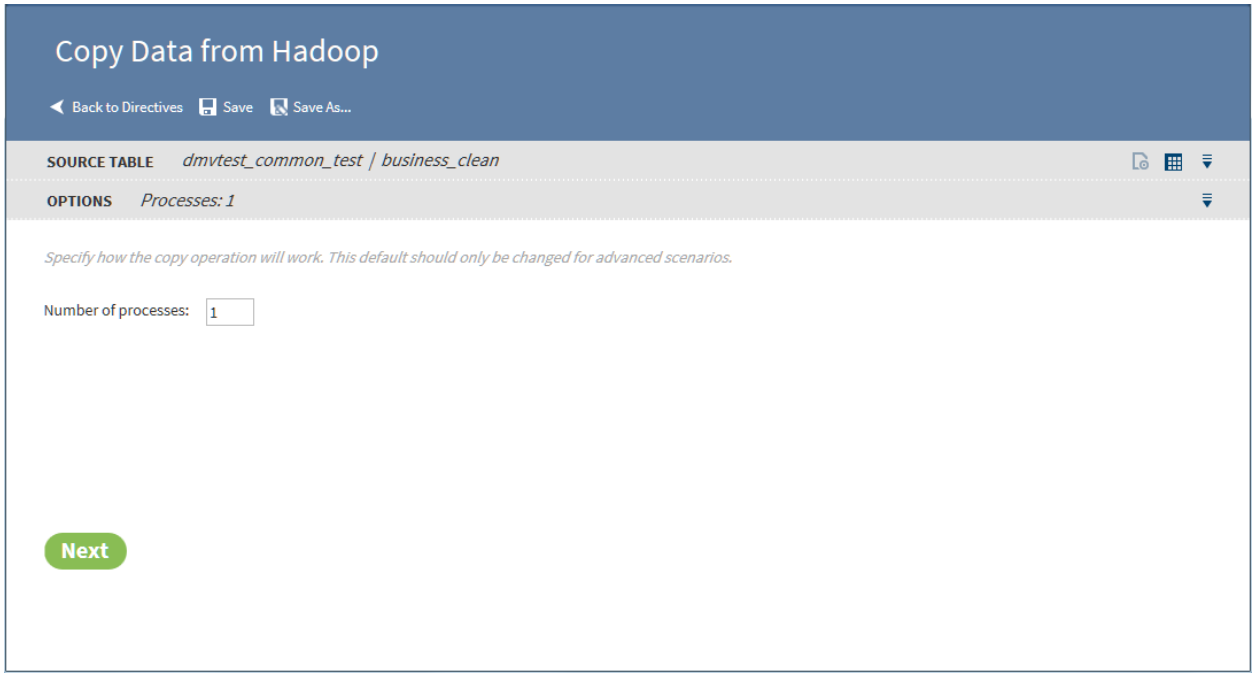
To learn more about data sources and source tables, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16.

Select a source table and click **Next** to display the **Options** task.

3. Make changes in the **Options** task only when you intend to copy data to a database. Advanced knowledge of database operations is required.

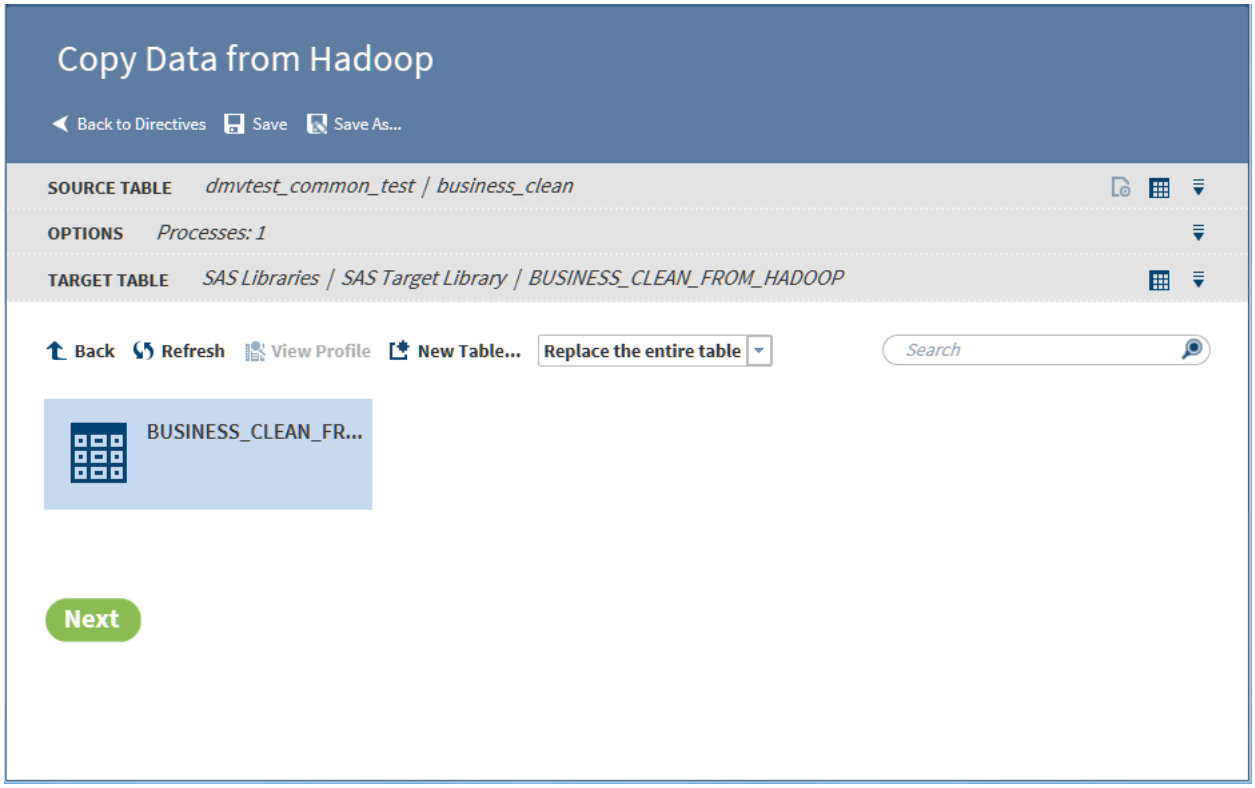
   Do not make changes in this task if you intend to copy data to a target table in a SAS library.

   In most cases, you should click **Next** without making changes. The **Target Table** task is displayed.



Click **Next** to display the **Target Table** task.

4. In the **Target Table** task, click a target data source and click a target table, or click **New Table**.

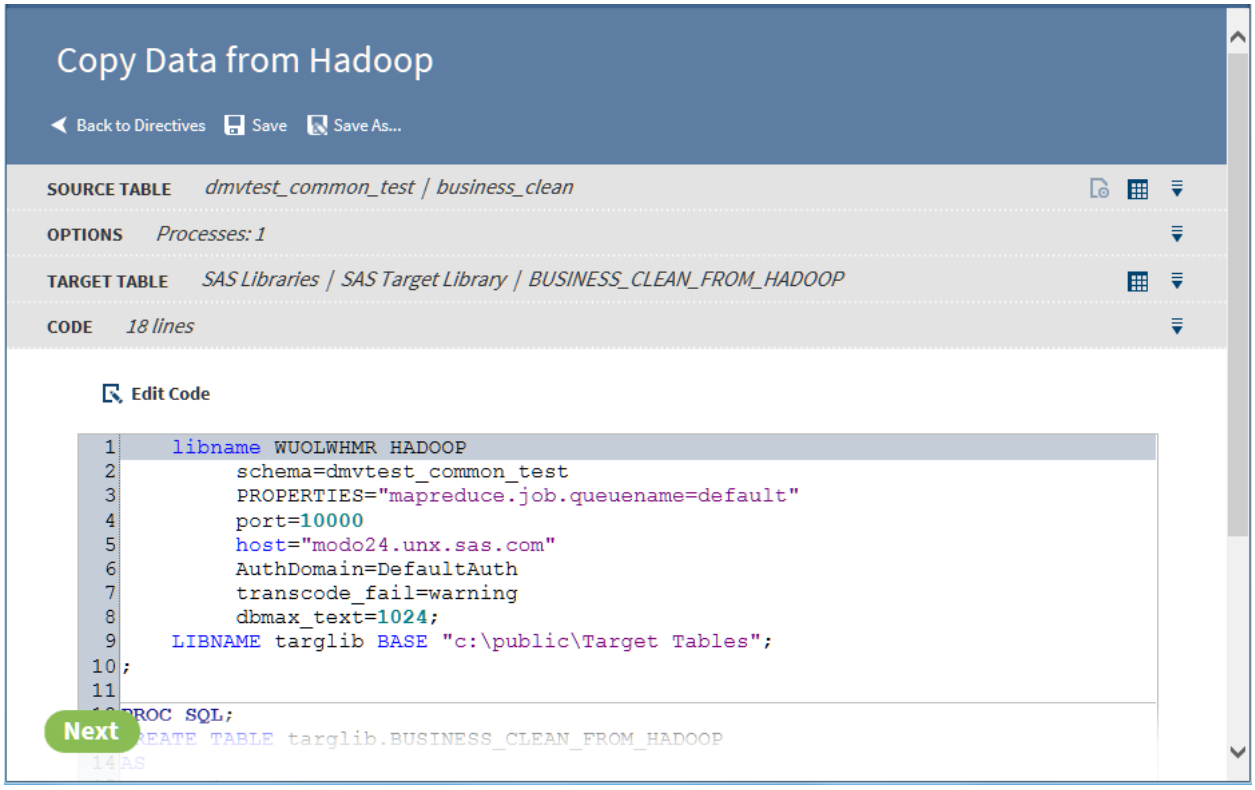


To copy data to an existing target table, select **Replace entire table** or **Insert into the existing table**.

To learn more about browsing target tables, see "Get Started with the Target Table Task" on page 18.

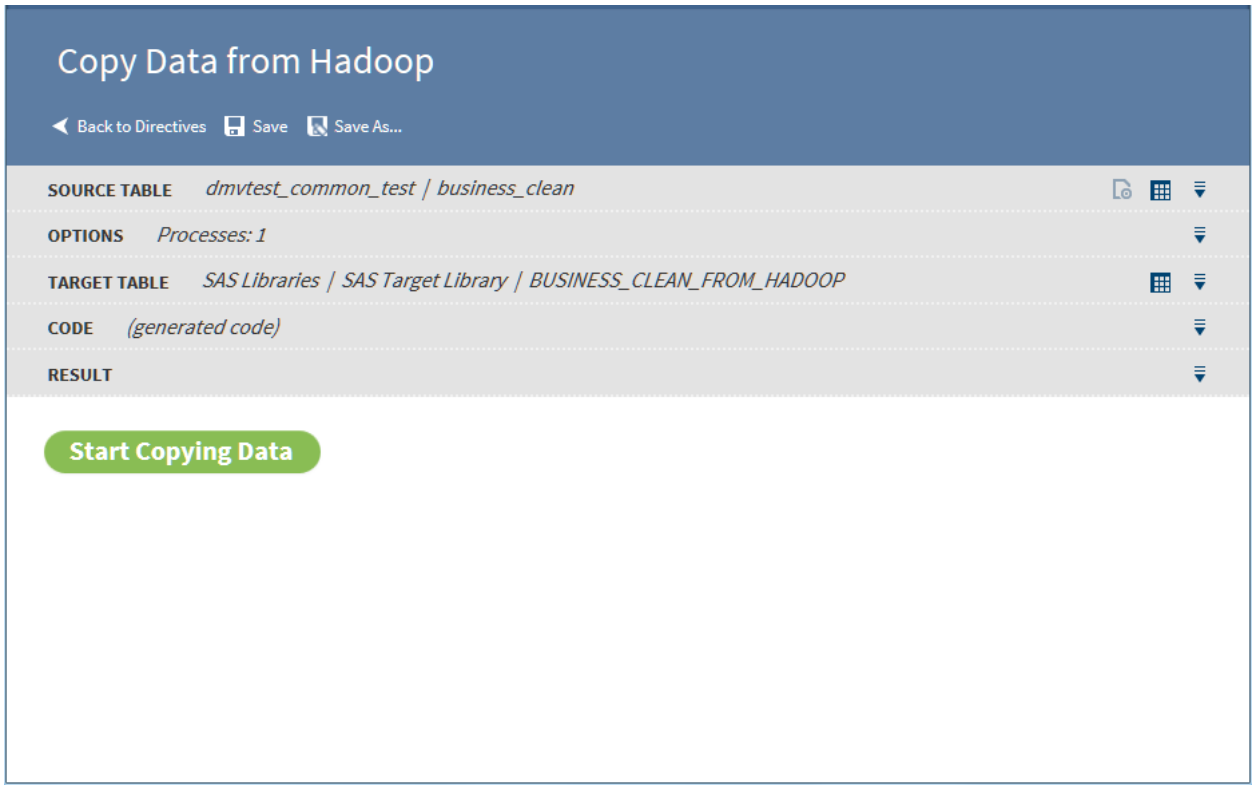
Click **Next** to display the **Code** task.

5. Use the **Code** task to review the code that is generated for your directive. Code changes require extensive knowledge of SAS programming and SAS Data Loader for Hadoop.



Click **Next** to display the **Result** task.

6. Use the **Result** task to run your directive. Click **Start Copying Data**.
7. When execution is complete, the **Result** task indicates success or failure.



As the directive runs in Hadoop, the **Result** tab indicates success or failure. Select from the following icons as they appear:

View Results
displays the target table in the SAS Table Viewer.

Log
displays the SAS log that is generated during the copy process.

Note: If the log does not resolve the directive, ask your administrator to consult the log files for Apache Sqoop.

Code
displays the SAS code that copies data from Hadoop.

8. Click **Save** or **Save As** to store your directive fro reuse.

Usage Notes

If you copy data to a target data source that uses a SAS/ACCESS LIBNAME engine, then your directive can be subject to the usage limits of that engine. Refer to the respective SAS/ACCESS document as needed, such as *SAS/ACCESS for Relational Databases: Reference*.

If you copy data to a target data source that uses a JDBC driver, then the following usage notes might apply:

- The Copy Data 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 for Hadoop, select the empty table as the target of the Copy From Hadoop directive. Select **Insert into the existing table** in the **Target Table** task.

- 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 for Hadoop, select the empty table as the target of the Copy From Hadoop directive. Select **Insert into the existing table** in the **Target Table** task.
- Source tables with a large number of columns can cause Copy From Hadoop directives to fail. The directive runs until the target table reaches the maximum number of columns that are supported for that target. To resolve the problem, reduce the number of columns that are selected for the target and run the directive again. To reduce columns, use a directive such as Cleanse Data, which contains a **Manage Columns** task.
- 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 the manner in which STRUCT is mapped to VARCHAR in Sqoop.
- A Copy Data from Hadoop directive is likely to fail if the name of a target 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, request assistance from a SAS administrator. The SAS administrator needs to follow these steps to update the Teradata database connection:

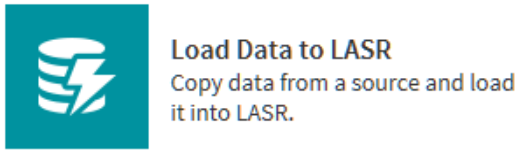
  1. Edit the default connection string to include the option **`charset=utf8`**, as shown in this example:

  ```
  jdbc:teradata://TeradataHost/Database=TeradataDB,charset=utf8
  ```

  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



Use the Load Data to LASR directive to copy Hadoop tables to SAS LASR Analytic Server. On SAS LASR Analytic Server, you can analyze data using software such as SAS Visual Analytics.

The Load Data to LASR directive is compatible with instances of SAS LASR Analytic Server that are configured for SMP (symmetric multiprocessing) or MPP (massively parallel processing.)

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

To maximize performance, subset your source tables in Hadoop before you load them to SAS LASR Analytic Server. To subset your source tables, use a directive such as Cleanse Data, which provides a Filter task and a Manage Columns task.

Example

Follow these steps to copy data from Hadoop to SAS LASR Analytic Server:

1. On the SAS Data Loader for Hadoop directives page, click the Load Data to LASR icon.

   Note: If you do not see the Load Data to LASR icon, then you do not have permission to access that directive. To change your permissions, contact your SAS administrator.

   The directive opens to the **Source Table** task.

2. The **Source Table** task displays the Hadoop source tables in your most recently selected data source.

   Either click a source table and click **Next**, or click **Return to Data Sources** to select a source table from a different data source.

   Note: To initially display Hadoop data sources, see the General Preferences panel in the Configuration window  .

   Note: To learn more about working with data sources and tables, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16.

3. Use the **Options** task to specify compression in the SAS target table, and to select a SAS folder to contain the target table or tables. Click **Next** to display the **Target Table** task.
4. The **Target Table** task displays the tables in the selected SAS folder. Click a target table to display its configuration fields and controls. Note that the **Delete Table** option in the action menu is not available. Only tables in Hadoop can be deleted. Existing tables in SAS LASR Analytic Server can be replaced.
5. As needed, change the name in the **Target table name** field. The field defines the name of the table as it will appear in SAS LASR Analytic Server. Click **Next** to display the **Result** task.
6. In the **Result** task, click **Start loading data**. SAS generates code for the directive and executes the code. As execution proceeds, the Result task displays the selectable **Code** and **Log** icons.

   A successful result displays the selectable **View Results** Unsuccessful execution displays the selectable **Error Details** icon .

Usage Notes

SASIOLA

To load data from MapR Hadoop to SAS LASR Analytic Server, the SAS LASR Analytic Server definition must assert the SASIOLA option. The SASIOLA option implements symmetric multiprocessing (SMP). Administrators add, modify, and delete connections to SAS LASR Analytic Server.

Name Length Limit for Tables and Columns

The names of tables and columns are limited to less than 32 characters, for both sources and targets.

Direct Loads from HDFS or NFS

The Load Data to LASR directive loads Hive tables to SAS LASR Analytic Server. It does not load HDFS or NFS data directly. This is because the Load Data to LASR directive can use the SAS embedded process software in Hadoop when parallel loading is configured by an administrator.

Truncation

In order for the Load to LASR directive to be successful, SAS LASR Analytic Server cannot be configured to truncate incoming character data. To prevent truncation, do not set the DBMAX_TEXT data set option on the source table of the LASR directive.

If truncation is necessary, use a different directive to create a truncated source table in Hadoop, and then copy the truncated table to SAS LASR Analytic Server. To create a truncated table, follow these general steps:

1. Open a directive such as Cleanse Data and select the source table that you want to truncate.
2. In the **Source Table** task, click **Edit advanced options** and enter an appropriate value for **Maximum length for SAS character columns**.
3. In the **Target Table** task, create a new target table, and then run the directive.

Libraries Required on New Instances of SAS LASR Analytic Server

When a new instance of SAS LASR Analytic Server is registered in metadata, at least one library must also be defined in metadata, with appropriate access permissions. Without a library, directives that attempt to create new tables will fail.

Chapter 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 for Hadoop directives. You can also create and execute new programs. Running your SAS and SQL programs in directives enables the use of the directive management capabilities in Run Status and Saved Directives.

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 the code can be saved and reused in Saved Directives.

The code execution process begins and ends in the SAS Information Architecture, outside of Hadoop. A SAS Workspace Server 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 SAS Workspace Server receives notification and continues or concludes the execution of the directive.

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 SAS Workspace Server can overload that server. To avoid overloading, your SAS programs need to minimize or eliminate the transfer of data from Hadoop to the SAS Workspace Server. 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 into SAS LASR Analytic Server.

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 9.4 In-Database Products: User's Guide. In that same document, see also the section “SAS In-Database Code Accelerator for Hadoop” to run DS2 code directly in Hadoop.

For DS2 syntax information, see the SAS 9.4 DS2 Language Reference.

User-written SAS DS2 code can also 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 for Hadoop directives page, click **Run a SAS Program**.
2. In the **Code** task, enter 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 directive is displayed in the **Result** taskbar.

5. Click **Save** to save your program for reuse. To edit or run your directive in the future, go to the SAS Data Loader directives page and click **Saved Directives**.

Run a Hadoop SQL Program

Introduction

Use Run a Hadoop SQL Program to 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 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, and Sort and De-Duplicate Data.

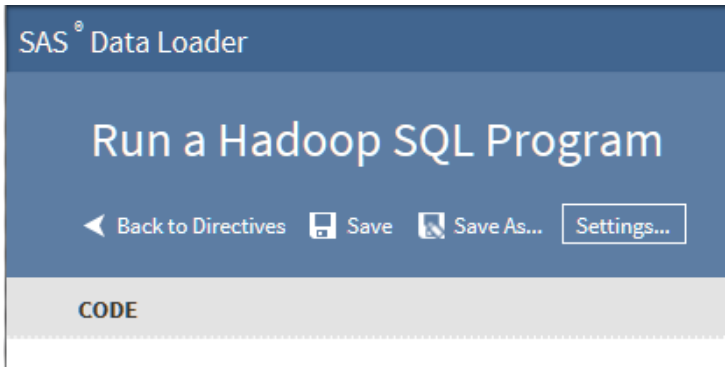
Enable the Impala SQL Environment

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.



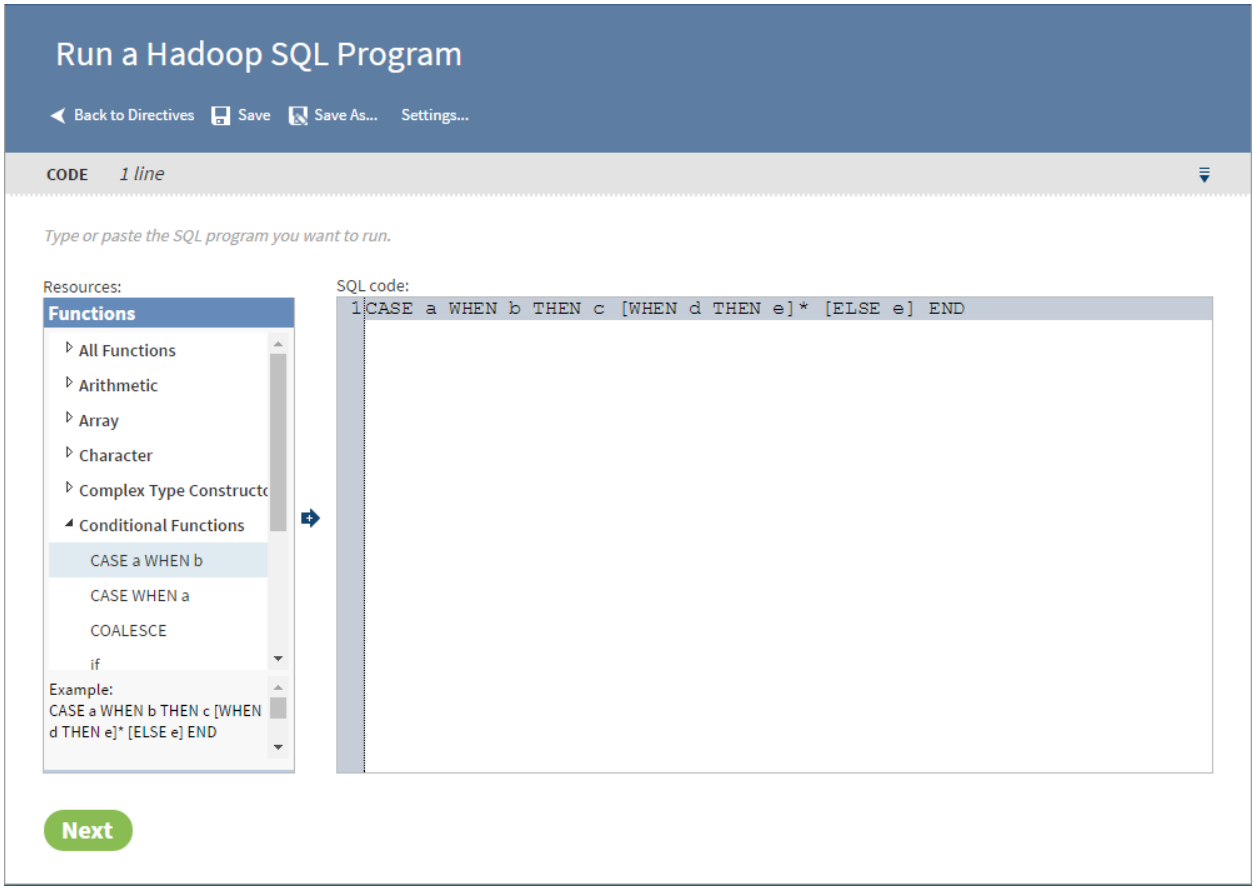
Example

Follow these steps to use the directive Run a Hadoop SQL Program:

1. In the SAS Data Loader for Hadoop 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 directive is displayed in the **Result** taskbar.
8. Click **Save** to save your program for reuse. To edit or run your directive in the future, go to the SAS Data Loader directives page and click **Saved Directives**.

Chapter 8

Manage Directives

Overview of the Management of Directives

The management of directives is handled by the directives Run Status, Saved Directives, and Chain Directives.

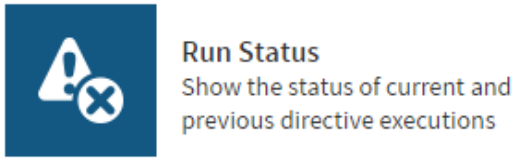
Run Status displays information for all directives that you have submitted for execution. You can also open and execute your directives.

Saved Directives lists all of the directives that you saved to a SAS folder on an instance of SAS Workspace Server.

Chain Directives creates directives that include serial or parallel execution of multiple existing jobs, including other chain directives.

Run Status

Introduction



Use the Run Status directive to display status and access the directives that you have run over a specified period of time. Each run record is listed with its execution status, start time, end time, and run time.

For each run record, a menu enables you to work with the directive or delete the run record.

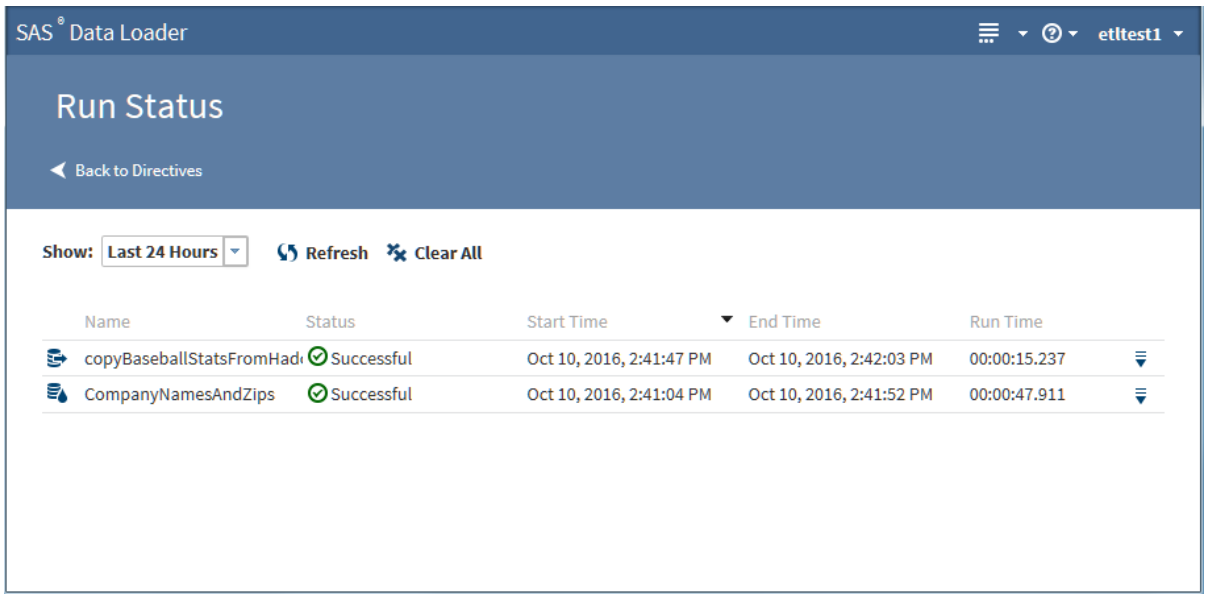
TIP To update the status of active directives, click Refresh

Note: If you logged in with a user ID that has SAS Data Loader for Hadoop administrative permissions, you see all run records for all users in the specified time period. A **Users** column is added to the directive to identify the individual that ran the directive.

Example

Follow these steps to use the Run Status directive:

1. In the SAS Data Loader for Hadoop directives page, click the Run Status directive. The Run Status page is displayed:



2. To display older run records, select a different value for **Show**.

3. To update the display to include recently submitted directives, or to obtain updates to directives that are currently running, click Refresh
4. To delete all of your run records, click Clear All.
5. The **Status** column displays the most recently reported status of each directive. The status can be In Progress, Stopped, Failed, or Successful.
6. To access a directive, select its run record and then click . Select from the following actions:
   - **Open** the directive as it was last run. Note that the version of the directive that you open in Run Status might not be the latest version. As needed, compare the directive's dates in Run Status and Saved Directives.
   - View the SAS **Log** that was generated by the directive.
   - View the **Code** that was submitted to run the directive.
   - **Start** a directive that stopped or failed to complete.

     The directive runs with the configuration that was specified when you last saved the directive. To run an updated directive, open the directive in Run Status, make changes in the directive, and run the updated directive. The new directive is displayed when you return to Run Status.

     For directives that were created in SAS Data Loader for Hadoop 2.3 or earlier, enabling support for Apache Spark or the Cloudera Impala require you to create a new directive. For more information about Spark and Impala, see "Enable Support for Impala and Spark".
   - **Stop** the execution of a directive.

     It is often not possible to stop the current task in SAS or to stop the directive that is running in Hadoop. In such cases, the stop is enforced after the completion of the current task.

     TIP If you select **Stop**, your directive can continue to display **In Progress** in the **Status** column. 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.
   - **Delete** the directive from the Run Status page and from the SAS Data Loader for Hadoop repository. Saved directives remain available in Saved Directives.

About Unsaved Directives

If you run a directive without saving it, the run record is displayed as usual in the Run Status directive. When processing stops on the unsaved directive, you can select **Open** from the menu in the run record. You can then edit and save the unsaved directive.

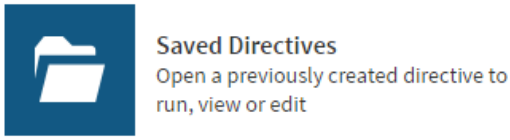
About Incomplete Directives

An incomplete directive is one that has a status of Failed , or that has been stopped by you or by an administrator. Depending on the type of the directive and the point where execution ceased, the Log and Code menu options can assist with troubleshooting.

You can start or edit an incomplete directive at any time.

Saved Directives

Introduction



Use Saved Directives to manage your directives. All of the directives that you save remain available until you delete them. The saved directives are stored in the SAS middle tier and in the SAS Metadata Repository at the path **`User Folders/`*`your-user-id`*`/My Folder`**.

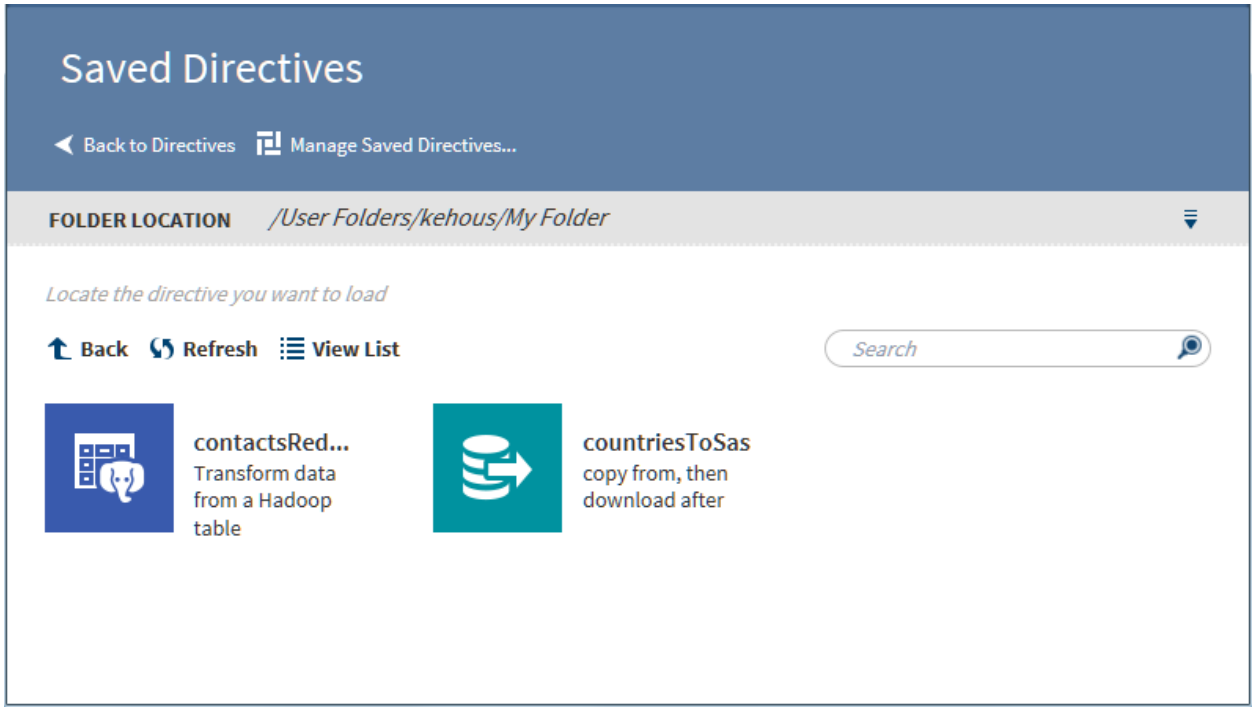
In the Saved Directives page, the Manage Saved Directives window opens, duplicates, deletes, refreshes, and renames directives. With appropriate permission, you can create new folders, move directives into those folders, and navigate among those folders.

Administrators have full access to all saved directives from all users.

Example

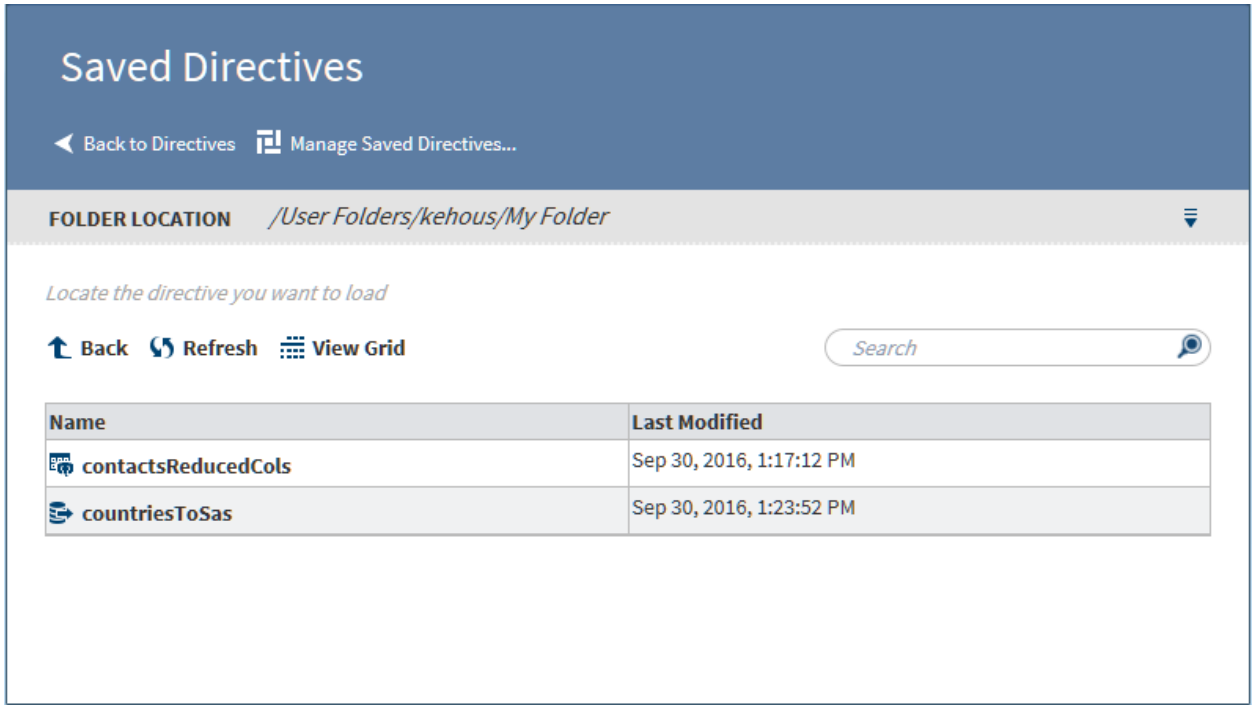
Follow these steps to use Saved Directives.

1. On the SAS Data Loader for Hadoop directives page, click the Saved Directives icon. The directive displays the directives that you have created and saved.



TIP If you forget to save a directive that you created and ran, you can open the directive in Run Status and then save the directive.

2. To display the saved directives in a list, with last-modified dates, click **View List**

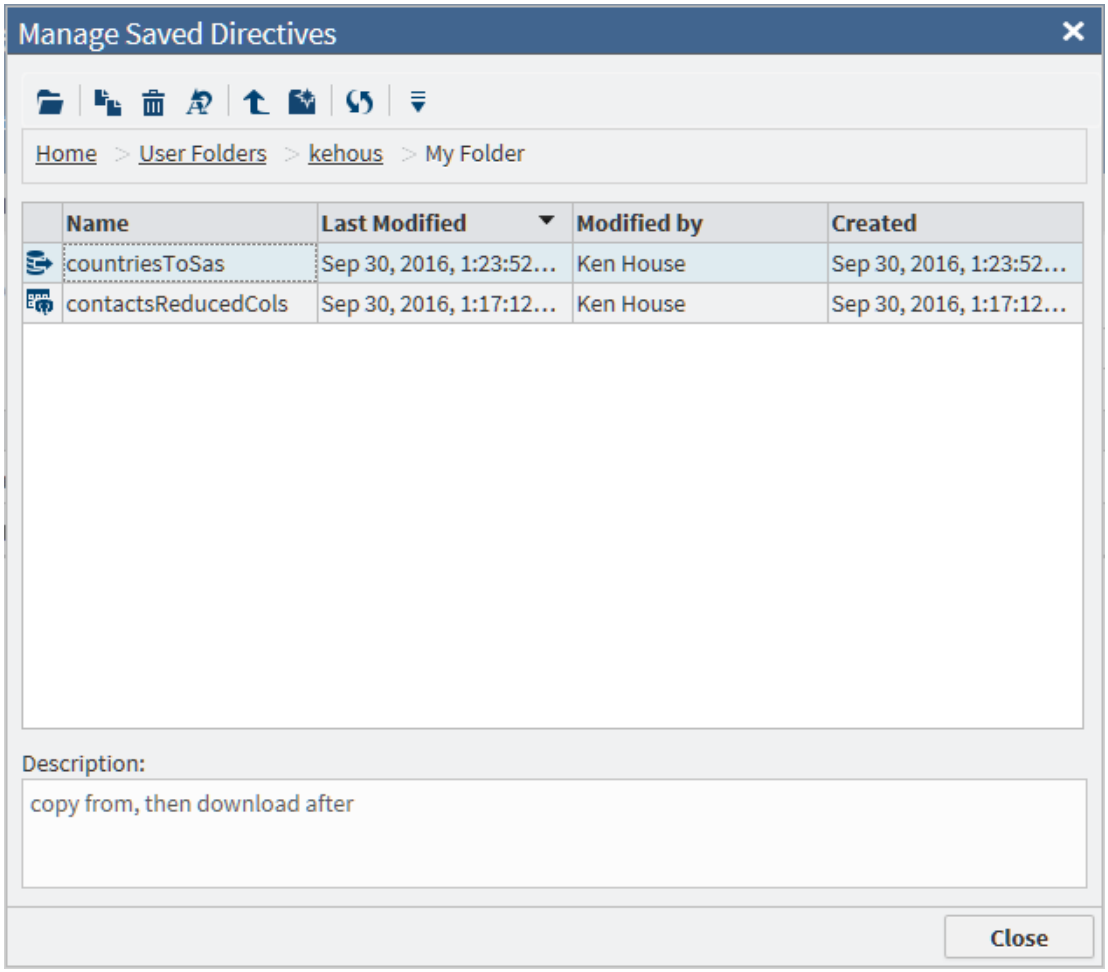


3. Click **Back** to navigate to other folders.

4. To open a directive, click the row or icon, click

in the **Folder Location** banner, and click **Open**. The saved directive opens and displays the previous task contents. You can also delete the saved directive.

5. Click **Managed Saved Directives** to display the Manage Saved Directives window.

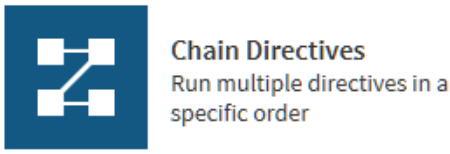


Click a directive and then click an icon or select a menu option to perform the following actions:

- Open the directive.
- Duplicate the directive.
- Delete the directive.
- Rename the directive.
- Open a different folder.
- Add a new folder.
- Create a new folder.
- Refresh the selected directive, or refresh all of the directives in the current folder.

Chain Directives

Introduction



Use Chain Directives to create a single directive that runs two or more of your existing directives. The existing directives run in series or in parallel.

You can nest one chain directive inside another. A serial chain directive can contain parallel chain directives. Parallel chain directives can contain serial chain directives.

A chain directive can run multiple instances of a single directive.

During execution, a chain directive displays the results of its component directives, as those results become available.

To run chain directives, open them in the Chain Directives interface, or run them from "Run Status" or from "Saved Directives".

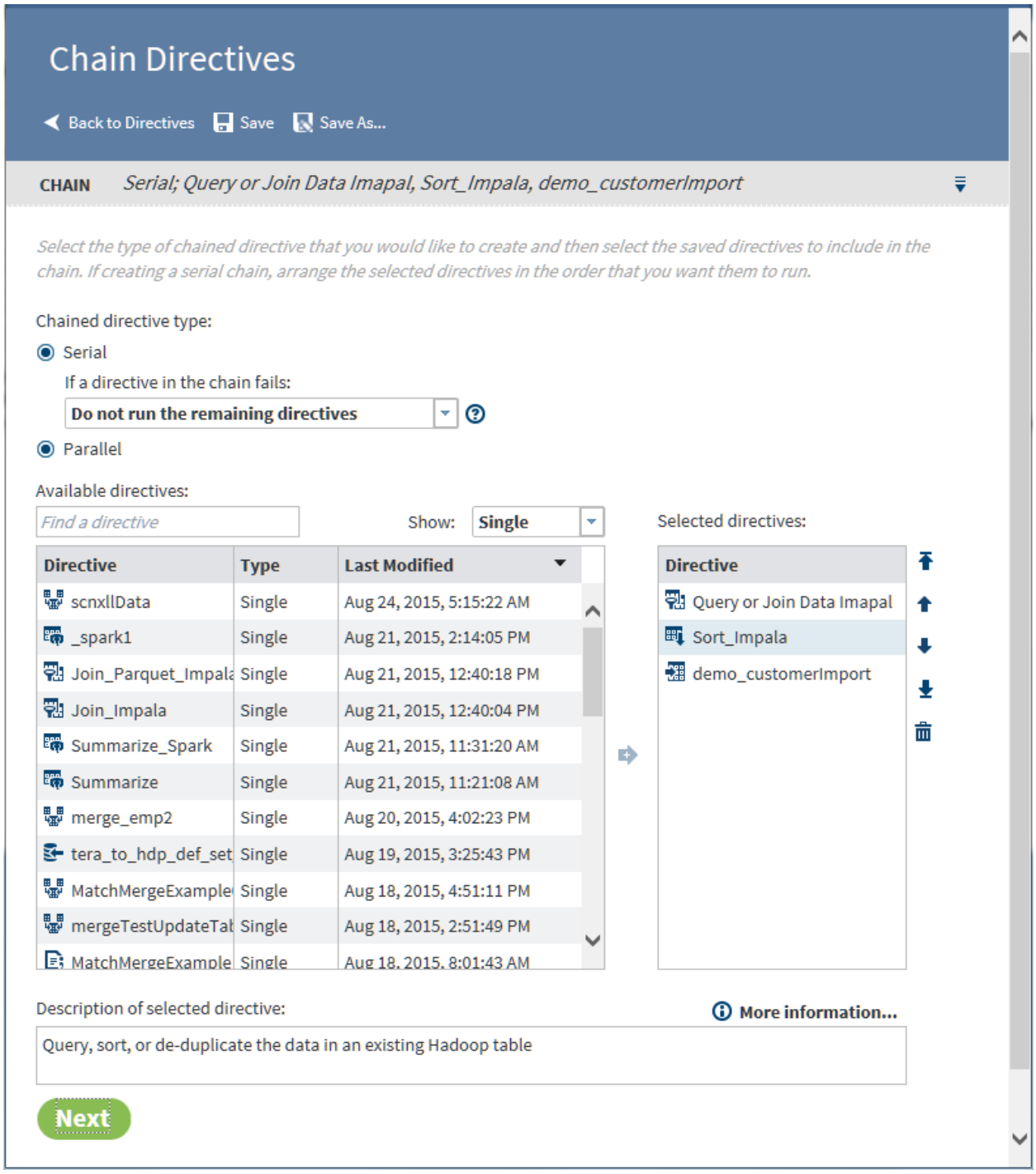
Example

Follow these steps to use Chain Directives:

1. On the SAS Data Loader for Hadoop directives page, click the Chain Directives icon.

   Note: If you do not see the Chain Directives icon, then you do not have permission to access that directive. To change your permissions, contact your SAS administrator.

   The directive opens in the **Chain** task.

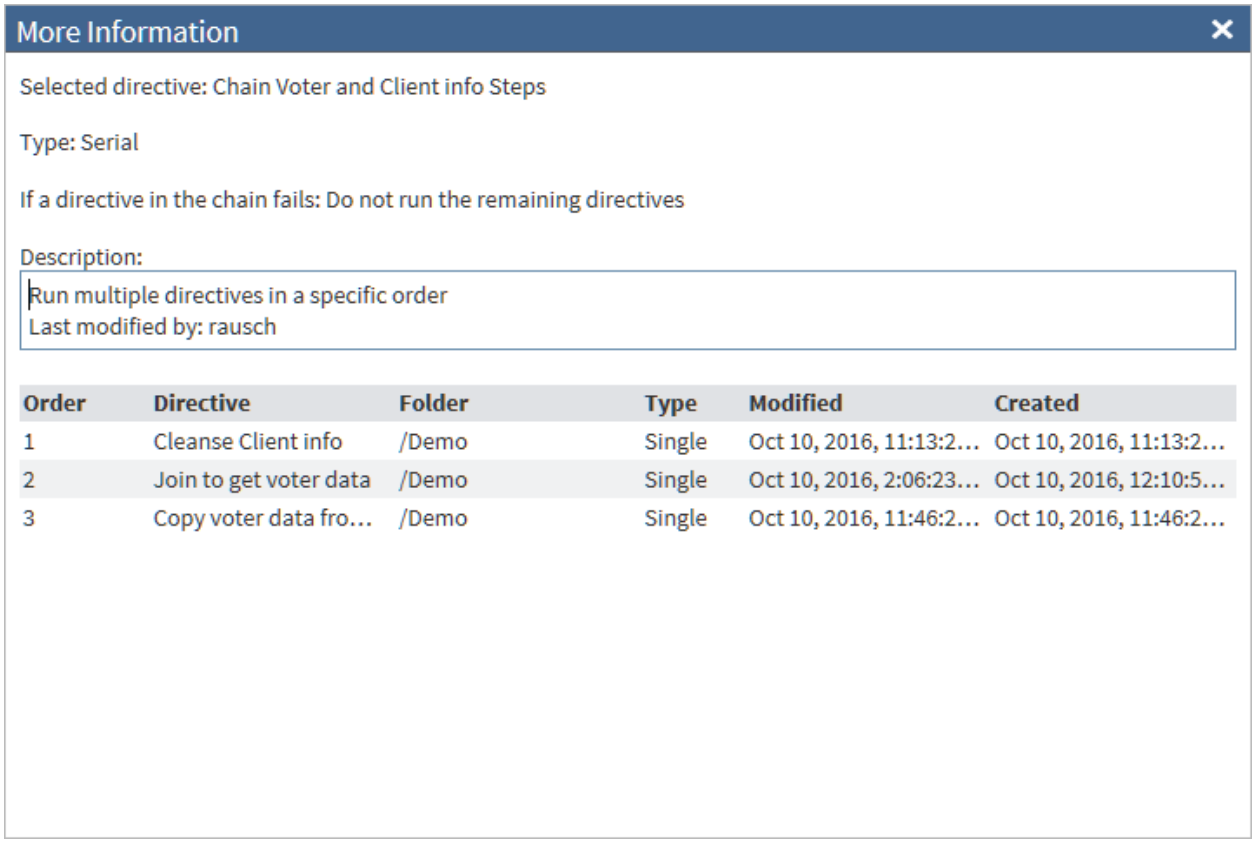


2. Click **Serial** or **Parallel** to define the execution type. Choose **Serial** when one directive in the chain depends on the results of a preceding directive. Choose **Parallel** if all of the directives in the chain can run independently and simultaneously.
3. If you select **Serial**, specify how execution should proceed in the event of a failure. Select either **Do not run the remaining directives** or **Continue to run the remaining directives**.
4. Select the directives for the chain. The **Available directives** list shows all of the saved directives that were created with your current user ID. Double-click a directive

to move it to **Selected directives**, or select multiple directives and then click the arrow icon.

To locate a particular directive in **Available directives**, search by name, or display directives by type (single, serial, or parallel.)

To learn about a selected directive, read the description or click **More information**.



If the selected directive is a serial chain, the **Order** column shows the order of execution. For a single directive, the **Order** value is blank. For a parallel directive, the **Order** values are **Concurrent**.

5. If you are creating a parallel chain directive, ensure that the component directives can 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 directive is reading that same table as a data source. To ensure accurate reads and writes, open the directives as needed to ensure that their target tables are unique. Also ensure that a target table in one directive is not used as a source in another directive.

   TIP If serial directives contribute values to a common target table, ensure that the directives append data to the target rather than replacing the data in the target.

6. If you are creating a serial chain directive, arrange the directives in order of execution. The directive at the top of the **Selected directives** list runs first. Execution proceeds down the list from the top. To arrange the selected directives, click a directive and click an icon to the right of the list.

   To move a directive from **Selected directives** to **Available directives**, click 🗑.

7. When **Selected directives** is complete, click **Next** to display the **Result** task.

8. In the **Result** task, click **Start running chained directives**. As the chain directive runs, you can examine the log, code, and results of each component directive as it becomes available.
9. Click **Save** to make your chain directive available in Saved Directives.

Chapter 9

Maintaining SAS Data Loader for Hadoop

Set Global Options

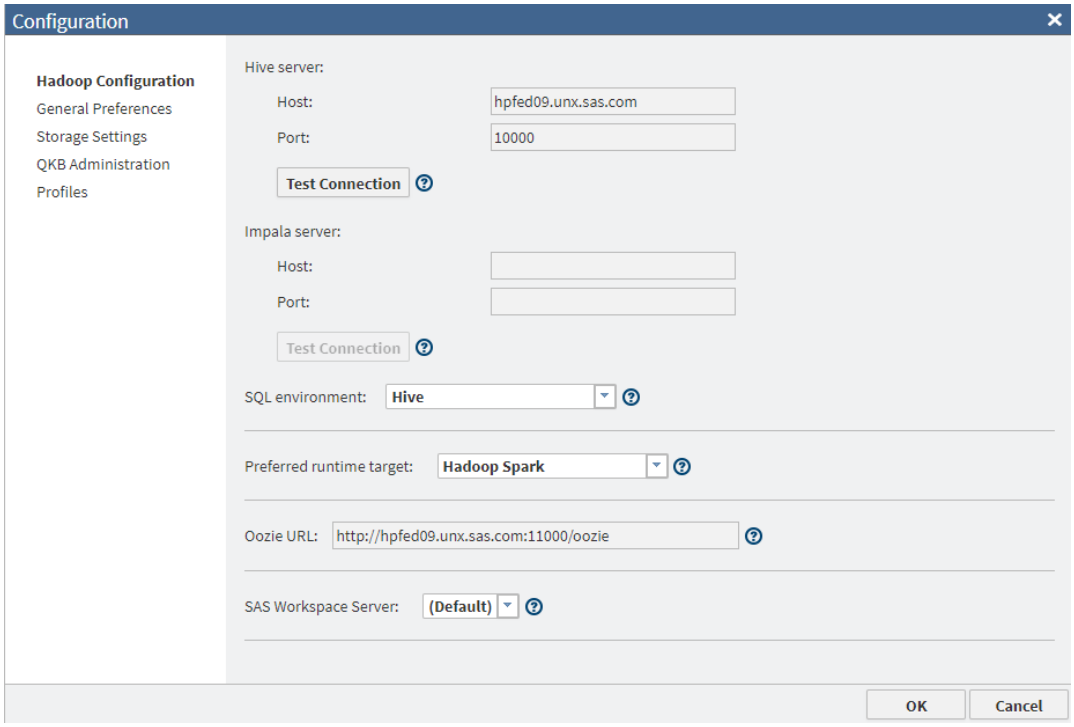
Overview of the Configuration Window

Use the Configuration window to display global settings and defaults, set preferences, and, with administrative authority, manage the SAS Quality Knowledge Base. To display the Configuration window, click  in the top right corner of SAS Data Loader for Hadoop, and then select **Configuration**. See the following topics for information about each panel in the window.

Hadoop Configuration Panel

The **Hadoop Configuration** panel displays the Hadoop connections that were established by your SAS administrator.

You can change the Spark and Impala defaults for new directives.



The fields and controls in the **Hadoop Configuration** panel are defined as follows:

Hive server host and port
: these fields display the Hive host computer name and port number that were configured by your SAS administrator. Apache Hive is a data warehouse infrastructure for the Hadoop Distributed File System (HDFS).

Impala host and port
: these fields display the Impala host computer name and port number that are configured by your SAS administrator. Cloudera Impala is an SQL environment that supports massive parallel processing.

Test Connection
: attempts to connect to the specified host and port number and displays a message window that indicates success or failure.

SQL environment
: select **Impala** to specify Impala as the default SQL environment for new directives that support Impala. Select **Hive** to specify the SQL environment that is supported by all directives.

  Individual directives that support Impala can be configured to use the Hive SQL environment.

  To learn more about Impala, including the subset of directives than support it, see "Enable Support for Impala and Spark" on page 11.

Note: Changing the **SQL environment** value does not change the SQL environment of saved directives.

Preferred runtime target

select **Hadoop Spark** to specify Apache Spark as the default runtime target for new directives that support Spark. Select **MapReduce** to specify the runtime target that is supported by all directives, with one exception. The Cluster-Survive Data directive requires the Spark runtime target.

Spark is a high-performance cluster computing framework that runs in coordination with existing Hadoop file systems.

If Spark is detected on the Hadoop cluster, then the **Hadoop Spark** value is set by default.

Individual directives that support Spark can be configured to override the default and run with the MapReduce runtime target.

The Spark runtime target is valid only when Spark is fully configured on the Hadoop cluster.

To learn more about Apache Spark, including the subset of directives than support it, see "Enable Support for Impala and Spark" on page 11.

Oozie URL

displays 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 directives. SAS Data Loader for Hadoop uses Oozie to copy data to and from databases such as Oracle and Teradata, and to execute directives in Spark. The Oozie URL is set and managed by your SAS administrator.

SAS Workspace Server

lists all of the SAS Workspace Servers that are defined on your SAS Metadata Server. Select the SAS Workspace Server that will run the new directives that you create and save.

CAUTION:

Be sure to select a SAS Workspace Server that is fully configured to support SAS Data Loader for Hadoop. Consult with your SAS administrator before you select a SAS Workspace Server other than the default.

General Preferences Panel

Introduction

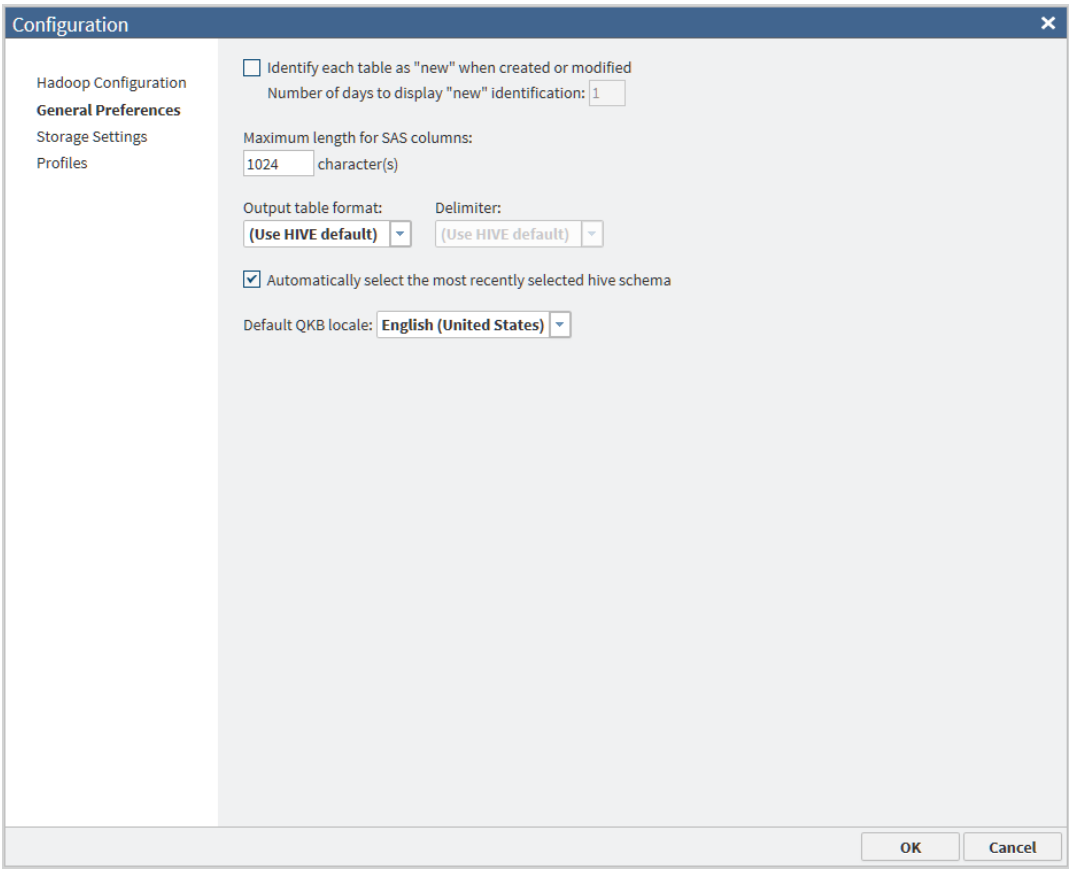
The **General Preferences** panel specifies various global options for SAS Data Loader for Hadoop.

Initial Notification Window

When you open the **General Preferences** panel, you might receive a notification window. The notification states that the default QKB locale cannot be loaded. This message indicates that the Quality Knowledge Base metadata on the Hadoop cluster needs to be published into the SAS middle tier. To resolve the issue, click **OK** to close the notification window. Ask a SAS administrator to use the QKB Administration panel to run the QKB publish operation.. The **QKB Administration** panel is available only to administrators.

After a successful QKB publish operation, access to the **General Preferences** panel is restored and new locales can be displayed for the **Default QKB Locale**.

Fields and Controls



The fields and controls in the **General Preferences** panel are defined as follows:

Identify each table as "new"
specifies the length of time (in number of days) that tables are identified as “new” in the Hadoop table browsers. 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 154.

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 the following output table formats: **Use HIVE default**, **Text**, **Parquet**, **ORC**, **Avro**, 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 153.

Automatically select the most recently selected hive schema
For directives in which the **Source Table** task opens in Hadoop, select this option to initially display the source tables in the data source that you opened in your preceding directive. Deselect this option to display all of the available data sources in Hadoop.

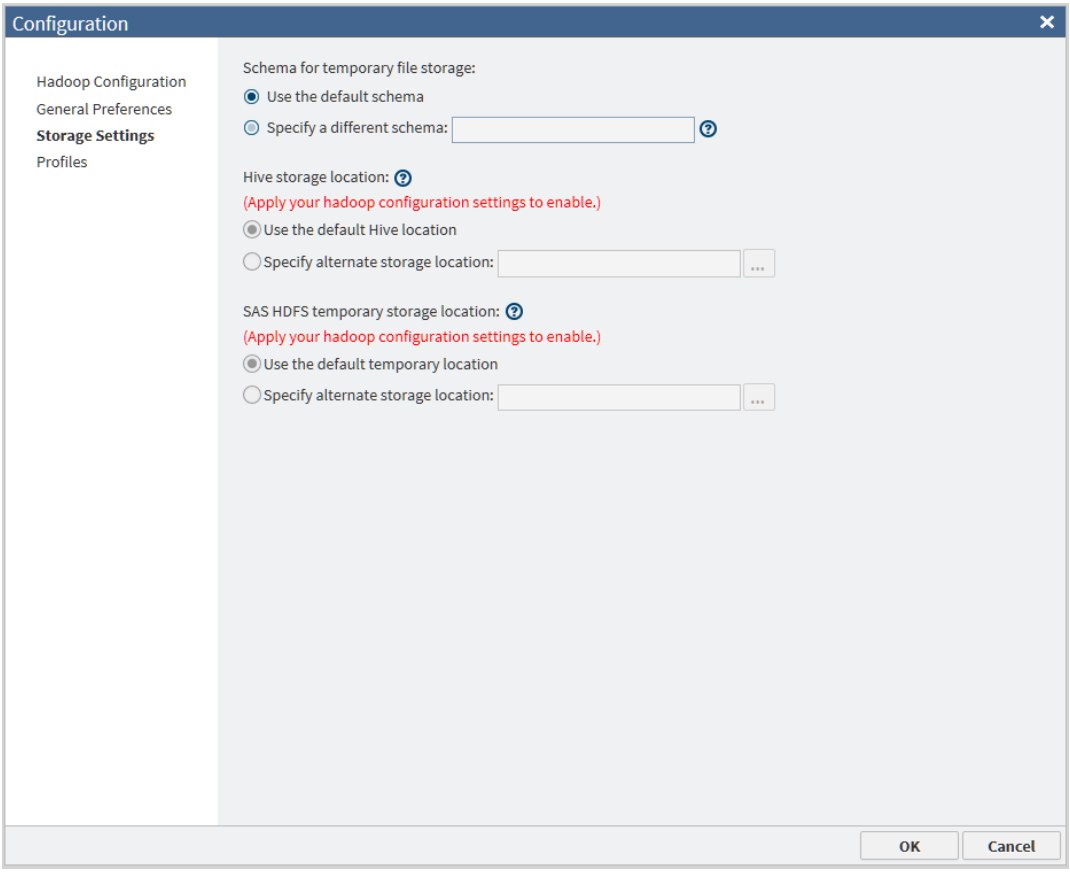
Note: For more information, see "Accessing Data Sources, Source Tables, and Target Tables" on page 16.

Default QKB locale
specifies the locale that applies to the majority of your source data. You can override the default locale and specify a different locale as needed in your data cleansing directives.

The SAS Quality Knowledge Base provides data definitions that are referenced in Hadoop during the execution of data cleansing directives. To learn more about QKBs, see "About Locales, Definitions, and the Quality Knowledge Base" on page 27.

Storage Settings Panel

The **Storage Settings** panel specifies non-default storage locations for schema temporary files, Hive, and HDFS.



The fields and controls in the **Storage Settings** panel are defined as follows:

Specify a different schema
click and enter the name of a Hadoop schema to use that schema to store Hadoop temporary files that are created during the execution of directives.

Specify alternate storage location (for Hive)
click the field and then click the browse icon to select an alternate storage location for your Hive content.

The Hive storage location can be changed only with appropriate permission, as determined by your SAS administrator.

Specify alternate storage location (SAS HDFS)
click the field and then click the browse icon to enter the name of a Hadoop schema to use that schema to store SAS temporary files that are created during the execution of directives.

The SAS HDFS temporary storage location can be changed only with appropriate permission, as determined by your SAS administrator.

Non-default temporary storage locations 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`. If the sticky bit is set, then files can be deleted from that directory only by the person who created the directory or by the root user.

QKB Administration Panel

Introduction

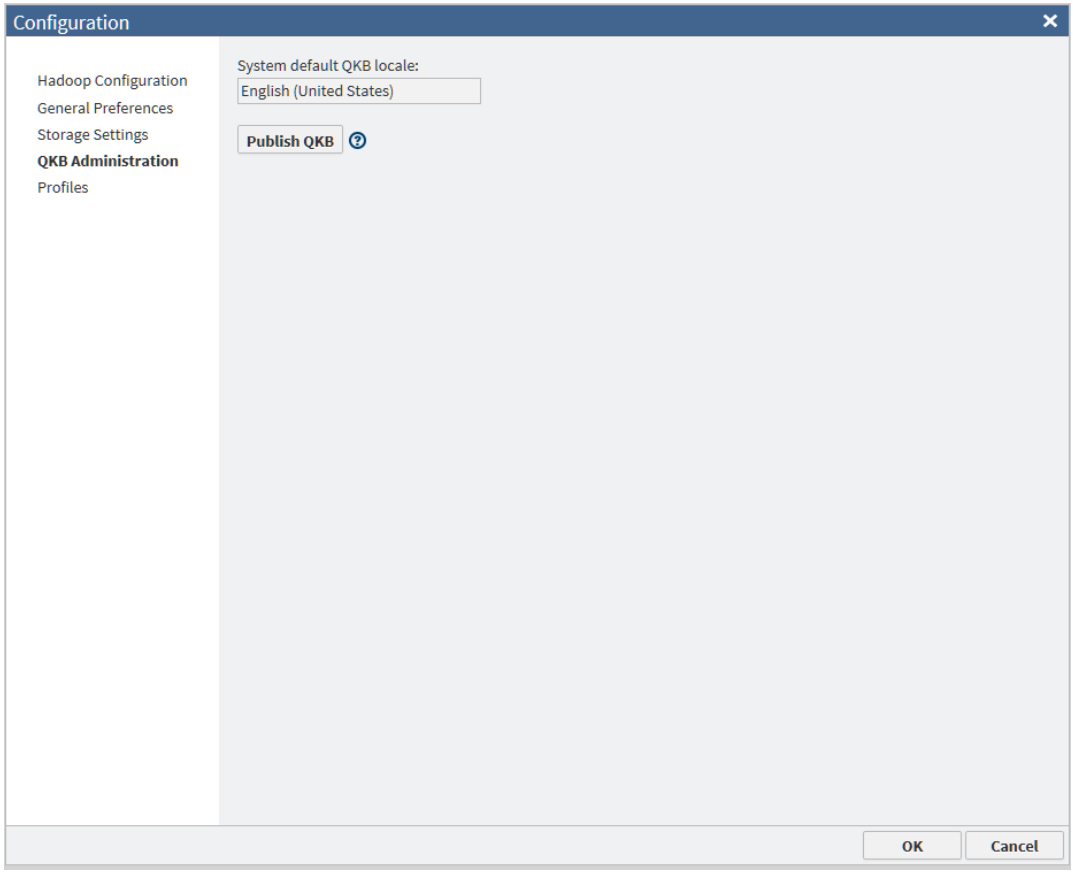
The **QKB Administration** panel is displayed only to SAS Data Loader administrators. The panel enables administrators to set a system default locale and synchronize QKB metadata between Hadoop and the SAS middle tier.

The SAS Quality Knowledge Base provides data definitions that are referenced in Hadoop during the execution of data cleansing directives. To learn more about QKBs, see "About Locales, Definitions, and the Quality Knowledge Base" on page 27.

Initial Notification Window

When you select the **QKB Administration** panel, you might receive a notification window. The notification states that QKB locales cannot be found. This message indicates that the QKB metadata on the Hadoop cluster needs to be published into the SAS middle tier. To resolve the issue, click **OK** to close the notification window, and then click **Publish QKB**. After the QKB publish operation, access to the QKB Administration panel is restored.

Fields and Controls



The fields and controls in the **QKB Administration** panel are defined as follows:

System default QKB locale
displays the system-wide default locale for the SAS Quality Knowledge Base. The default applies to all users who have not selected a different locale in the **General Preferences** panel. The system-wide default value is managed by SAS Data Loader administrators.

Publish QKB
copies updated QKB metadata from Hadoop into the SAS Data Loader for Hadoop middle tier. The publish operation is required under the following circumstances:

- after the QKB in Hadoop has been installed for the first time
- after the QKB in Hadoop has been updated to a new release
- after the QKB in Hadoop has been customized.

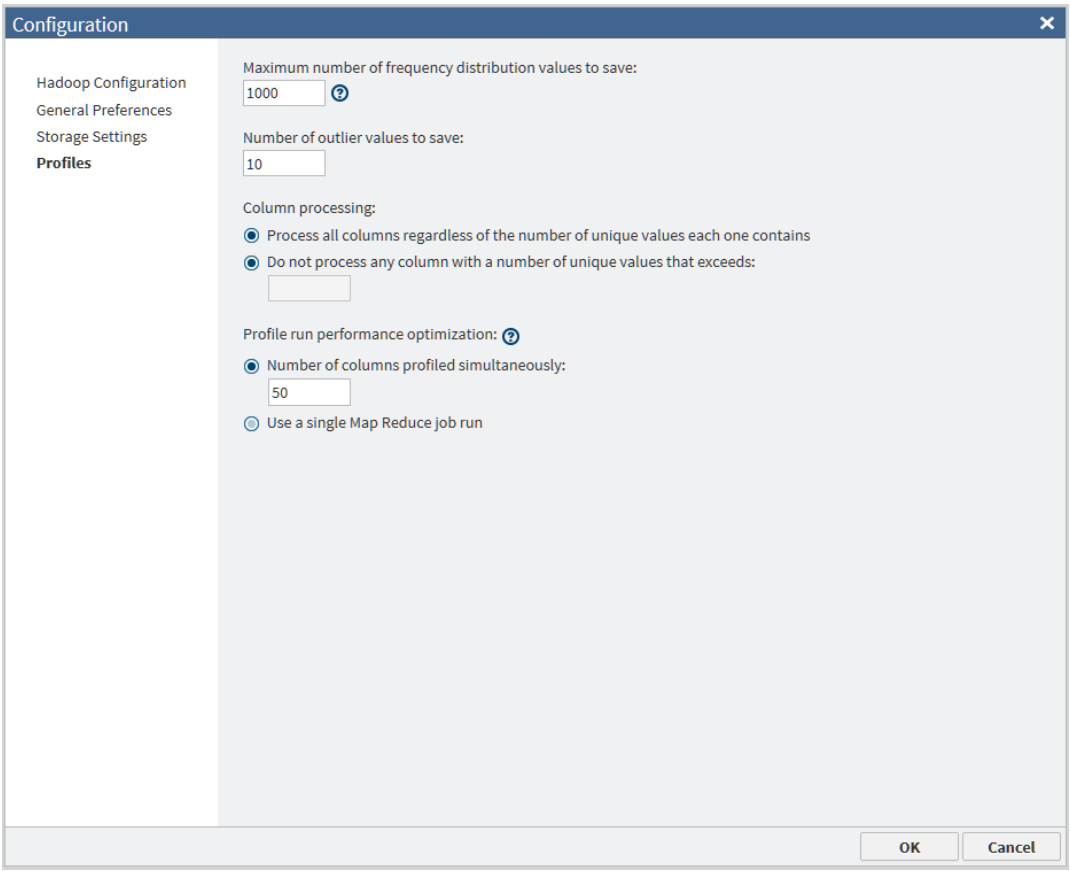
The QKB publish operation updates the QKB index file on the SAS middle tier host, at **`/sas/qkb/default.idx`**.

Profiles Panel

The **Profiles** panel configures the reports that are collected on specified Hadoop tables using the Profile Data directive. For more information about the Profile Data directive, see Chapter 5, “Profile Data,” on page 89.

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 enables you to change defaults, which can improve the performance of new profile directives.



The fields and controls in the **Profiles** panel are defined as follows:

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.

Process all columns regardless of the number of unique values each one contains
select this default option when the performance of profile directives is not an issue.

Do not process any column with a number of unique values that exceeds
select this option to improve the performance of profile directives. The text field specifies when a source column is excluded from the profile based on the number of unique values in that column. Change the default value of 1000 as needed.

Number of columns profiled simultaneously
select this option if your source tables frequently contain 50 or more columns. Increase the default value of 50 as needed, based on the performance of your profile directives.

Rather than specifying a value less than 50, it is recommended that you select **Use a single MapReduce job run**.

Use a single MapReduce job run
select this option if your source tables frequently contain 50 columns or less.

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);
```

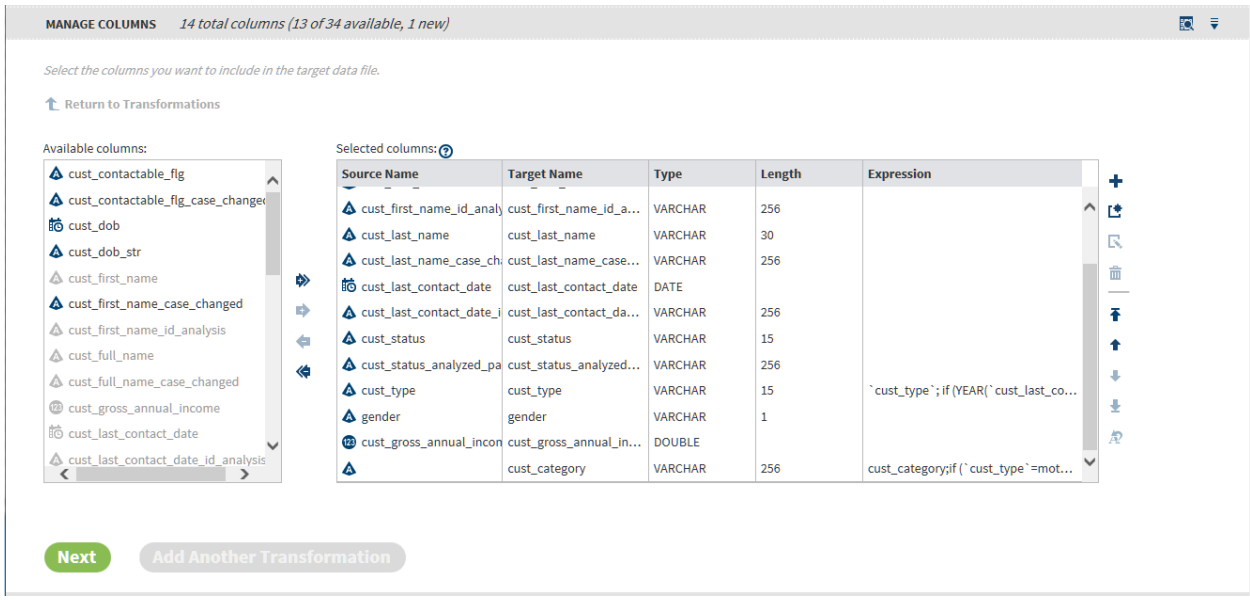
If the expression contains more than one clause, then the first clause needs to be a placeholder value, as shown in the following example:

```
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 runtime target, user-written EEL expressions can be applied in the Filter Data and Manage Columns transformations. These transformations are available in directives such as Cleanse Data and Transform Data. EEL is a SAS expression language that is optimized for Spark, as described in *Expression Language: Reference Guide*.

In the Filter Data and Manage Columns transformation, 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 EEL expressions 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.

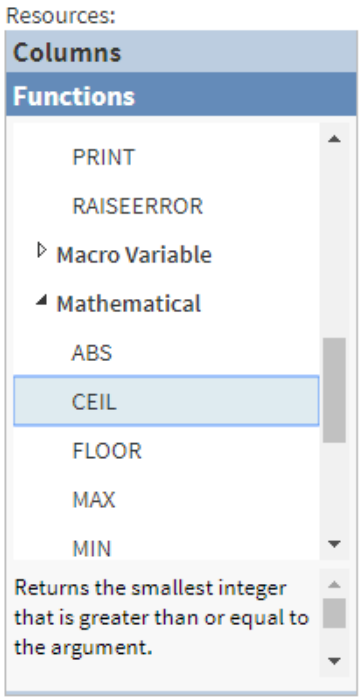
In the preceding example, a new column is positioned at the bottom of the **Selected columns** list, as the last column in the target table. In this position, the expression can reference any of the other columns.

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.

To display EEL syntax help, follow these steps:

1. In Managed Columns, open the advanced editor by clicking or .
2. In the Advanced Editor, click **Functions** in the **Resources** box.
3. Click a function to display the syntax help.



Additional information is available in *Expression Language: Reference Guide*

Troubleshooting

Change the File Format of Hadoop Target Tables

In the 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 for Hadoop, when you create a new target table, the Output Table Format option is specified by the directive. The directive can use the default output table format or apply a different format.

The default output table format is specified in the **Output table format** field in the **General Preferences** panel of the Configuration window.

To open the Configuration window, click in the banner at the top of the browser window.

You can also access the **Output table format** field within a directive using the **Action** menu in the **Target Table** task.

The available values of the **Output table format** field are defined as follows:

Use HIVE default
: specifies that new target tables receive 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 for Hadoop.

Text
: specifies that output target tables are formatted as a series of text fields that are separated by delimiters. For this format, 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 output table format, which is optimized for nested data. The Parquet algorithm is considered to be more efficient than using flattened nested namespaces. The Parquet format requires HCatalog to be enabled on the Hadoop cluster.

Orc
: specifies the Optimized Row Columnar output table format, which is a columnar format that efficiently manages large amounts of data in Hive and HDFS.

Sequence
: specifies the SequenceFile output table format, which enables Hive to efficiently run MapReduce.

Avro
: specifies the Avro output table format. Avro files consist of two parts: a JSON (JavaScript Object Notation) data definition and a compact binary representation of data. The data definition defines data types and protocols, which can be read and written with a variety of compiled languages and scripted languages. The binary representation contains markers to efficiently locate data in the file.

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 following directives can be affected by the length of SAS character columns:

- Cleanse Data
- Transform Data
- Transpose Data
- Load Data to LASR
- Copy Data to Hadoop
- Copy Data from Hadoop
- Match-Merge
- Run a SAS Program

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 with the Spark runtime target. 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`). (The sticky bit is a permission that limits file deletions to the owner and administrator.) 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 Directive 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

Avoid using DS2 or DBMS reserved keywords in column names. For some directives, if you use a reserved keyword in a column name, the directive can fail. For example, in a Copy Data from Hadoop directive, if you copy data to Teradata, and if a target column name contains a Teradata reserved word, then the directive fails. For more information, see "Naming Requirements for Schemas, Tables, and Columns" on page 10.

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 "Storage Settings Panel" on page 147 in 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 and run the directive.

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 for Hadoop first converts the Hadoop column name and data type into its SAS equivalent (numeric or character.) When the transformation is complete, SAS Data Loader for Hadoop 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 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 for Hadoop 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 for Hadoop 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 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.

Recommended Reading

- System Requirements--SAS Data Loader for Hadoop
- SAS 9.4 Support for Web Browsers and Plug-Ins
- SAS Data Loader 3.1 for Hadoop product page
- *SAS Data Loader for Hadoop: Installation and Configuration Guide*
- *SAS DS2 Language Reference*
- *Expression Language: Reference Guide*
- *SAS/ACCESS for Relational Databases: Reference*
- SAS Quality Knowledge Base for Contact Information 27: Help
- Ten Tips to Unlock the Power of Hadoop with SAS

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