SAS® Data Studio 2.5: User's Guide

What's New in SAS Data Studio 2.5

Ability to Select the Format When You Save Target Tables

You can select a format when you save target tables. The available formats depend on the library where the table is saved. For more information, see "Saving Plans and Tables" on page 46.

Double Data Type Available for Match and Cluster Transform Cluster ID

You can set the data type of the cluster ID generated by the Match and Cluster transform. The available options are **Double** (eight bytes) and **Char** (twelve bytes). The default data type is **Double**. For more information, see "Matching and Clustering" on page 24.

Interface Updates

The interface now includes buttons to support the **Suggestions** feature and a Status window.

Manage Columns Transform

The **Manage Columns** transform enables you to manage output table columns by modifying the source table. You can select or rename columns to get the output table that you need. For more information, see "Manage Output Columns" on page 12.

Remove Duplicates Transform

The **Remove duplicates** transform enables you to remove all but one row from a set of rows that have identical values in a list of user-specified fields. For more information, see “Remove Duplicates” on page 23.

Save as In-Memory Table Only Check Box

You can select the **Save as in-memory table only** check box to load a table to memory without saving a physical copy of the table to the target destination. This option can provide a significant performance boost for small tables. For more information, see the description of this check box in “Saving Plans and Tables” on page 46.

Simple Random Partitioning Method

The Simple Random partitioning method has been added to the Analytical Partitioning transform. Simple random specifies that the partitioning values are generated randomly from the values in the source data. For more information, see “Creating a Partition Column” on page 34.

Suggestions Feature

The Suggestions feature uses machine learning to analyze your data and suggest transforms and actions that you can add to your SAS Data Studio plans. For more information, see “Working with Suggestions” on page 38.

About SAS Data Studio

Overview of SAS Data Studio

SAS Data Studio offers an easy way for you to prepare data. The following list summarizes the tasks that you can perform using SAS Data Studio:

Perform data transforms
: You can perform data transforms such as joining tables, appending data to a table, transposing columns, creating calculated columns, and so on.

  If SAS Data Preparation is licensed at your site, you have access to data quality transforms.

Create plans
: You can create a plan, which is a collection of actions or data transforms performed on a table.

Create and view profiles

You can create and view profiles, which provide standard metric information about a table.

If SAS Data Preparation is licensed at your site, you have access to additional column metrics.

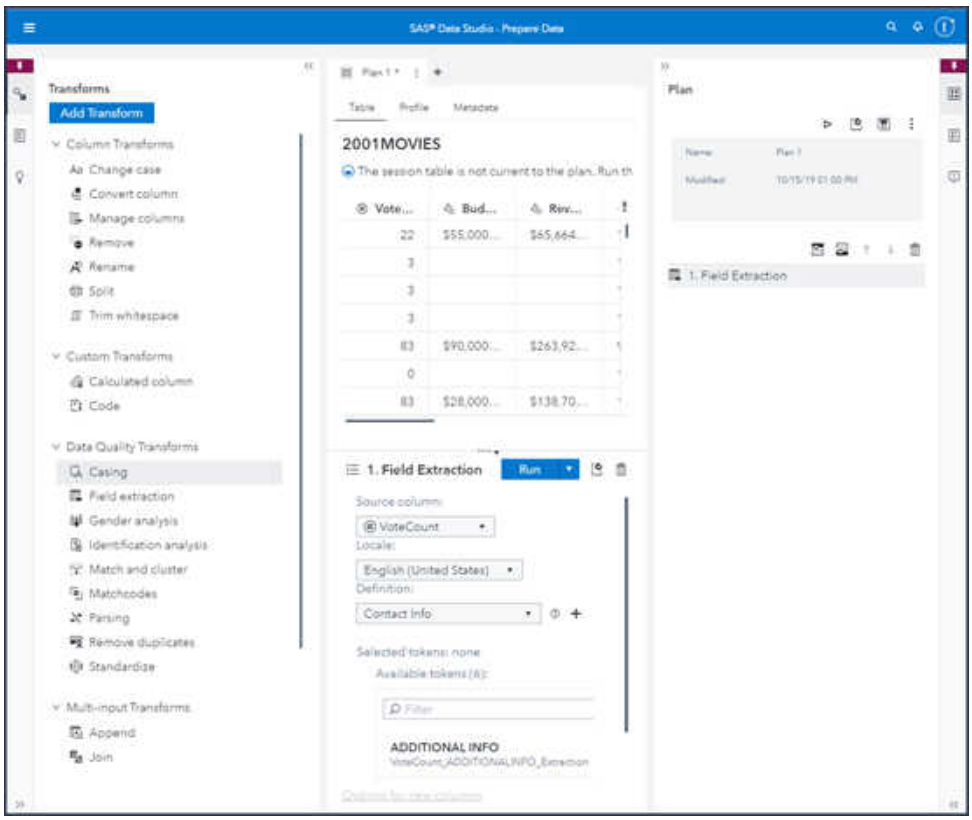
Support for Third-Party Software

Unless otherwise noted, SAS Data Explorer supports the databases, the browsers, and other third-party software that is supported by SAS Viya. For more information, see Third-Party Software Requirements for Use with SAS Viya.

Your First Look at the Interface

The main SAS Data Studio interface enables you to prepare and view data. Here are more details about the interface:

Figure 1 *SAS Data Studio Main Screen*



- The application bar at the top of the window enables you to access other SAS applications. You can search for items, access help, update your settings, and sign out of SAS Data Studio. For more information about application-specific settings, see “Modifying SAS Data Studio Settings” on page 48. For more information about search and global settings, see *SAS Viya Web Applications: General Usage Help*.
- The top pane in the center of the window enables you to view the details about a table, including table data, profiles, and metrics.

- The bottom pane in the center of the window enables you to configure a selected transform.
- The toolbar on the left edge of the window enables you to add transforms, view properties, and generate suggestions for the source table. These actions are displayed in a pane at the left side of the window.
- The toolbar on the right edge of the window enables you to view plan actions, view plan status, and view properties for the result table. These actions are displayed in a pane at the right side of the window.

About the Left Pane

Table 1 *Left Pane Controls*

Icon	Description
	enables you to add transforms.
	enables you to view properties for the source table.
	enables you to generate a list of suggested actions to perform on the table that is active in the plan.

About the Right Pane

Table 2 *Right Pane Controls*

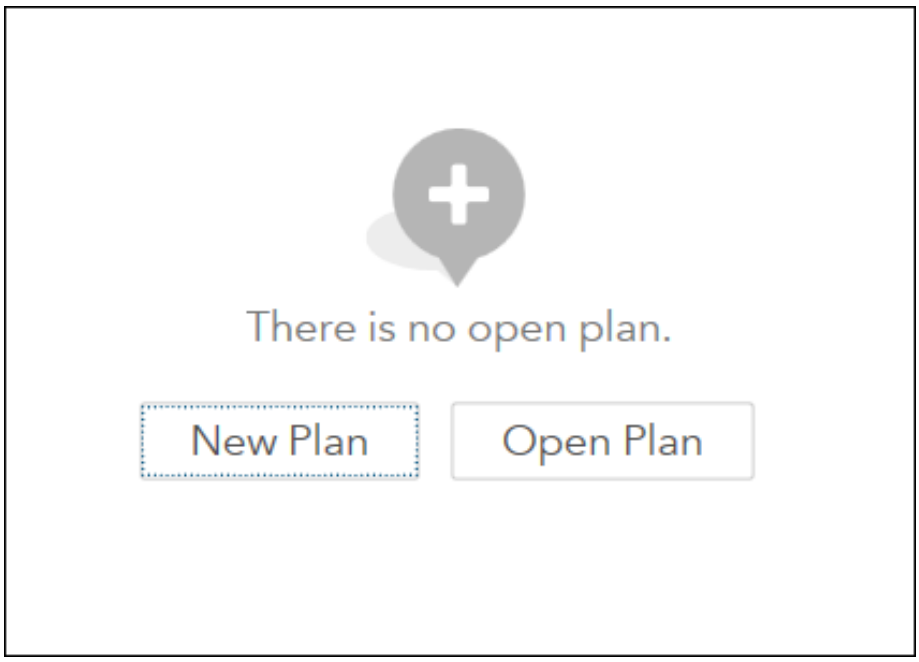
Icon	Description
	enables you to view and manage plan actions. You can use the toolbar above the list of transforms to insert, reorder, and delete transforms.
	enables you to view properties for the result table.
	enables you to view the status of your plans.

Getting Started with SAS Data Studio

Open a Data Source

Get started with SAS Data Studio by opening a data source, profiling it, and examining its metadata and properties.

1 In the SAS Data Studio window, click **New Plan**.



> **TIP** If you already have a plan open, you can change the source table by clicking ⋮ on the toolbar.

2 In the Choose Data window, select the name of the table that you want to open, and then click **OK**.

For more information about the Choose Data window, see "Understanding the Available, Data Sources, and Import Tabs" in *SAS Data Explorer: User's Guide*.

Note: If you choose a table that is located in an encrypted CAS library, and you do not have authorization to use the CAS library, then the table does not open. You will receive an error message.

If you have appropriate privileges, the Choose Data window enables you to add a session-based caslib or a global caslib. Source and target tables in SAS Data Studio plans must be in a global caslib if you want these tables to be visible in another application, such as SAS Visual Analytics. For more

information about adding a global caslibs, see "Data Sources Tab: Connect to External Data Sources" in *SAS Data Explorer: User's Guide*.

Run and View Profiles

1. Open a data source.
2. Click the **Profile** tab.
3. Click **Run Profile**.
4. Click the **Profile** tab, and then click the name of the column that you want to view. If SAS Data Preparation is licensed at your site, then you can view advanced column metrics. These metrics include pattern and frequency distributions.

Note: Click the **Run Profile** button to refresh the profile for the source table. To see the profile that includes changes that you made to the source table, you must first save the table.

For more information about profiles, see "Profiling Data" in *SAS Data Explorer: User's Guide*.

View Table Metadata

To view table metadata, open a data source, and then click the **Metadata** .

Note: After you run a transform, the **Metadata** tab will reflect the current state of the table, including any changes that you made.

View Table Properties

To view information about the source table, click in the left pane. To view information about the target table, click in the right pane.

Table properties include:

- the number of columns, number of rows, table size, label, and location
- any tags attached to the source table
- the timestamp for when the table was created and the timestamp for when the table was last modified

  Note: The timestamp for both the **Date created** and **Date modified** fields are always the same. This occurs because when you modify a table, a new target table is created.

- encoding type

Working with Columns

Access Columns

The table initially shows all the columns present in the data set. You can use the following procedure to determine which columns are hidden or displayed:

1. Open a data source, and then click in the left pane.
2. Click at the top right corner of the table to access the Manage columns window.
3. Use the scroll bar and the arrow controls in the **Displayed columns** pane to review the list of columns contained in the table.
4. Move any columns that you need to hide to the **Hidden columns** pane.
5. Click **OK** to close the window.

Change the Case in Columns

There are two transforms available to change the case in columns: **Change case** and **Casing**.

If SAS Data Preparation is licensed at your site, then you have access to more advanced casing options using the **Casing** transform. For more information about the **Casing** transform, see “Working with Data Quality” on page 16.

To change the case of the data in a column using the **Change case** transform:

1. Open a table. Then click in the left pane.
2. Select **Change case** in the transforms list. Then, click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.)
4. From the **Case** drop-down menu, select **Uppercase** or **Lowercase**.
5. Select **Replace source column** or **Create new column**. If you choose to create a new column, click **Options for new columns** to specify a new column name.
6. Click **Run**.

Change the Data Type for a Column

The data types for a column include **Character**, **Double**, **VarChar**, **DateTime**, **Date**, and **Time**. Not all of the data types are available for all tables. The availability of the types depends on how the table was imported.

Note: Column names that contain Windows line feed characters are not recognized by **Convert column**.

To change the data type for a column:

1. Open a table. Then click in the left pane.
2. Select **Convert column** in the transforms list. Then, click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.)
4. Specify the name of the new column in **New column**. You must create a new column. You cannot change the format for an existing column.
5. Select a data type from the **Conversion** drop-down menu.

   Here is information about the available data types:

   CHARACTER *(n)*
   specifies a fixed-length column of length *n* for character data. The maximum for *n* is 32,767.

   VARCHAR *(n)*
   specifies a varying-length column of length *n* for character data. The maximum for *n* is 536,870,911.

   DOUBLE
   specifies a column with numeric values.

   DATE *(n)*
   specifies date values in the format NLDATE20.

   DATETIME
   specifies date and time values in the format NLDATM30.

   TIME
   specifies time values in the format NLTIME20.
6. Click **Run**.

Change the Data Format for a Column

1. Open a table. Then click in the left pane.
2. Select **Convert column** in the transforms list. Then, click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.)

4 (Optional) Click in the **Informat or format** field to indicate an input informat or format for the column. The format or informat that you indicate is used to convert the values in the column. Depending on the column type, there might not be any informats or formats available.

   If you use an informat or format, be aware that format or informat works together with length and format fields. The length of the field should be equal to or greater than the string length that results from conversion. In general, you should be aware of formats and how they work in SAS.

5 Specify the name of the new column in **New column**. You must create a new column. You cannot change the format for an existing column.

6 Indicate the maximum character length. For numeric fields only, it is recommended that you leave the default value of `8`. If the length value that you enter is not supported by the server, it is ignored and the length of the new column is set to `8`.

7 (Optional) Indicate a format for the column. The format that you indicate in the **Format** field is used to change how the values in a column are displayed.

8 (Optional) Enter a label for the column in the **Label** field. (However, some data types do not support labels.)

9 Click **Run**.

Remove White Space in Columns

1 Open a table. Then click in the left pane.

2 Select **Trim whitespace** in the transforms list. Then, click **Add Transform**.

3 Select a source column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.)

4 Choose one of the following actions:

- Click **Compress all whitespace** to remove all white space from the column values, including trailing, leading, and in-between white space.
- Click **Trim leading and trailing whitespace** to remove trailing and leading white space from the column values.
- Click **Trim leading whitespace** to remove only leading white space from the column values.
- Click **Trim trailing whitespace** to remove only trailing white space from the column values. Choosing this option also right-justifies column values.

5 Select **Replace source column** or **Create a new column**. If you choose to create a new column, click **Options for new columns** to specify a new column name.

Note: Right-justifying column values might result in data loss if the length of the target column is less than the length of the source column.

6 Click **Run**.

Remove Columns

1. Open a table. Then click in the left pane.
2. To remove a column, select **Remove** in the transforms list. Then, click **Add Transform**.
3. Select a column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.), Then click **Run**.
4. (Optional) To remove multiple columns at the same time, click the plus sign and select an additional column from the **Source column** drop-down menu.

Rename Column Headings

1. Open a table. Then click in the left pane.
2. To rename a column heading, select **Rename** in the transforms list. Then, click **Add Transform**.
3. Select a source column, enter the new name in the **Name of new column** field, and then click **Run**.

   Note: The column heading cannot exceed 255 bytes.

   Note: Renaming a column heading does not change the label for the column.
4. (Optional) To change the column label, use the **Convert column** transform.

Split a Column

A delimiter is a character that represents a boundary between two or more areas of text (for example, a comma (,)). To split a column, there are several options:

On a delimiter	Use this option if you want to split the column using a delimiter that you specify. Choosing the **On the delimiter** option creates a new column that contains all of the characters to the left of the delimiter. It also creates another new column that contains all of the characters to the right of the delimiter.
On fixed length	Use this option if you want to split the column based on the position that you indicate in the **Fixed length** field.
Before a delimiter	Use this option if you want to split the column using a delimiter that you specify. Choosing the **Before a delimiter** option creates a new column that contains all of the characters to the left of the delimiter. It also creates another new column that contains all of the characters to the right of the delimiter and the delimiter itself.

After a delimiter	Use this option if you want to split the column using a delimiter that you specify. Choosing the **After a delimiter** option creates a new column that contains all of the characters to the right of the delimiter. It also creates another new column that contains all of the characters to the left of the delimiter and the delimiter itself.
Quick split	Use this option to split the column on a cell-by-cell basis, based on the first delimiter that appears in each cell. You do not have the ability to indicate the delimiter with the **Quick split** option, and the results vary depending on the delimiters that the data contains. For example, `Winston-Salem, NC` is split based on the hyphen instead of the comma. In this example, the result is `Winston` in one column and `Salem, NC` in the other column.

Note: The **Quick split** option supports the following delimiters only:

`< ( + & ! $ * ) ; ^ - / , % |.`

In ASCII environments without the **^** character, the **~** character is supported.

1. Open a data source, and then click in the left pane.
2. Select **Split** in the transforms list. Then, drag it into the plan.
3. Select a source column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.)
4. Select an option from the **Split data** drop-down menu:
   - Select **On a delimiter**, **Before a delimiter**, or **After a delimiter** to split the data using a delimiter that you specify (for example, a comma).

     Note: If there are multiple delimiters of the same type in the column, and you select **On a delimiter**, **Before a delimiter**, or **After a delimiter**, then the column is split based on the first occurrence of the delimiter in each cell.
   - Select **On fixed length** to split the data based on the position that you specify.
   - Select **Quick split** to split the column based on the first supported delimiter that appears in each cell. If you select this option, skip step 5.

   All of these split types maintain leading blanks. You might need to trim the leading white space before you split a column.
5. Depending on the type of split that you selected in the **Split data** drop-down menu, select a delimiter in the **Delimiter** drop-down menu, or indicate the position in the **Fixed length** field.

   If you want to split the column based on a delimiter other than a comma or a space, select **Other** from the **Delimiter** drop-down menu. You can indicate a custom delimiter in the text box that appears. Here is some key information about the **Other** text box:
   - You cannot use a combination of characters as a delimiter. For example, if you enter **EU** in the **Other** text box, the word **Europe** is split using the letter **E** only. A single character is treated as a separate delimiter.
   - There is no limit to the number of delimiters that you can enter in this text box.
   - If you enter multiple delimiters, then the split occurs on a cell-by-cell basis according to the delimiters that you indicated and in the order in which they appear in the **Other** text box. For example, if you enter **abc** in the **Other** text box, then the word **track** is split using the letter **a**, the word **box** is split using the letter **b**, and the word **code** is split using the letter **c**.
   - Control characters and unprintable characters are not supported.

- The **Other** text box is case sensitive.
- Column values that do not contain the delimiter that you indicate appear as blank cells in the new column on the right-hand side.

6 (Optional) Indicate names for the output columns in the **Name of new column 1** and **Name of new column 2** fields.

7 (Optional) Click **Options for new columns** to indicate additional options for the output columns (for example, column type, length, label, or format). (However, some data types do not support labels.)

8 Click **Run**.

Note: The sort order of the data in the output columns might be different from the sort order of the source column.

Manage Output Columns

1 Open a table. Then click in the left pane.

2 To manage output table columns by modifying the source table, select **Manage columns** in the transforms list. Then, click **Add Transform**. The Manage columns window is displayed.

3 Click the **Select Columns** link and review the list of columns. Use the horizontal arrow controls to move columns between the **Selected items** and **Available items** fields until you have the columns needed in your output. You can use the up and down arrows next to the **Selected items** field to move one or more selected columns higher or lower in the list. You can move these items one level at a time or all the way to the top or the bottom of the list.

4 Click the **Rename Columns** link. Rename any columns that need a different name in the **Output Column Name** field for the source column.

5 Click **OK** to save your changes and leave the window.

Note: You can enter the name of a column in the **Filter** fields in the **Select Columns** and **Rename Columns** sections of the transform. This feature is useful when the table contains many columns.

Creating Calculated Columns

1 Open a table. Then click in the left pane.

2 Select **Calculated column** in the transforms list. Then, click **Add Transform**.

3 In the Calculated Column window, enter a DATA step expression in the **Expression** field. Here are a few considerations:

- Do not include the name of the table, semicolons, or the **COLUMN=** statement in the expression. They are implicitly added for you.

- Enter a single value or single expression only. Do not enter conditional values.
- If your column name contains spaces, use the *columnName*'n syntax.
- UNIX and Windows use different carriage return characters. Do not use a column name with newline characters in the **Calculated column** transform.

For more information about DATA step expressions, see Dictionary of SAS DATA Step Statements.

4 Indicate how you want the calculated column to appear. Select **Replace existing column** to assign the value to the source column. Select **Create new column** to create a new column.

> **TIP** If you choose to create a new column, click **Options for new columns** to indicate column type, length, label, and format. (However, some data types do not support labels.) If you enter a length value for a column with a numeric type, that setting might not be supported by the server. It is ignored and the length of the new column is set to `8`.

5 Click **Run**.

In some cases, the results of a calculated column might appear blank. To see a value, position your pointer over the cell.

Creating Custom Code

About Custom Code

Note: The **Code** transform does not generate its output columns until it runs, so transforms downstream from it will not have any incoming columns if it has not yet run. Also note that UNIX and Windows use different carriage return characters. Do not use a column name with newline characters in the **Code** transform.

You can create custom code to perform actions or transformations on a table. There are two code languages available: CASL and DATA step.

Note: Each time you run a plan, table, and library names might change. To avoid errors, you must use variables in place of table and CAS library names. Indicating variables in place of table and CAS library names eliminates the possibility that the code might fail due to name changes. For more information about the variables that are available, see Step 4 on page 14.

Create Custom Code

1 Open a table. Then click in the left pane.

2 Select **Code** in the transforms list. Then, click **Add Transform**.

3 Select the code language from the **Language** drop-down menu that you access by clicking ⋮.

The following code languages are available: **CASL** and **DATA step**.

For more information about CASL, see SAS® Cloud Analytic Services 3.4: CASL Reference .

For more information about DATA step, see Dictionary of SAS DATA Step Statements .

4 Enter the code in the text box. The following variables are valid for both CASL and DATA step:

CAUTION
You must use the following variables in place of table and CAS library names. Errors occur if you use literal values. This is because session table and library names can change during processing.

_dp_inputCaslib
variable for the input CAS library name.

_dp_inputTable
variable for the input table name.

_dp_outputCaslib
variable for the output CAS library name.

_dp_outputTable
variable for the output table name.

Here is some key information about variables:

- For DATA step only, the variables must be enclosed in double braces. For example:

```
data {{_dp_outputTable}} (caslib={{_dp_outputCaslib}});
```

- Variable names are not case sensitive.

5 Click **Run**.

Example: Creating Custom Code

The following example creates a unique identifier in a table using custom code.

STUDENT	CLASS	GRADE	CREDIT
Ann	Math101	A	4.0
Ann	English101	B+	4.0
Ann	Biology101	B+	4.0
Ann	Biolab	A-	2.0
Bob	Math101	A-	4.0
Bob	Chemistry101	A-	4.0
Bob	Chemlab	A-	2.0
Carol	Spanish101	B	4.0
Carol	French101	B	4.0
Carol	History102	C	4.0
Carol	PoliSci111	B	4.0
David	Italian	C	4.0
David	Math210	C	4.0
David	Lit200	B	4.0
Fred	Chemistry101	B	4.0
Fred	ChemLab	B	2.0

1 Using the following source table, select **Code** in the transforms list. Then, drag it into the plan.

2 In the Code window, select **DATA step** from the drop-down menu.

3 Enter the following code in the text box:

```
data {{_dp_outputTable}} (caslib={{_dp_outputCaslib}}); set {{_dp_inputTable}}
(caslib={{_dp_inputCaslib}});
if _N_ = 1 then do;
    _mult = 10 ** (int(log10(_NTHREADS_)) + 1);
    retain _mult;
    drop _mult;
end;
"UniqueID"n = _THREADID_ + (_N_ * _mult);
run;
```

After you click **Run**, a new column named **UniqueID** will appear in the table:

STUDENT	CLASS	GRADE	CREDIT	UniqueID
Fred	Chemistry101	B	4.0	1001
Fred	ChemLab	B	2.0	2001
Fred	Anthro111	C	4.0	3001
Fred	Math110	A	4.0	4001
Ann	Math101	A	4.0	1049

Note: The entire table is not shown in the preceding image.

Working with Data Quality

About Data Quality

Note: The data quality transforms are available only with SAS Data Preparation. These transforms are displayed and available in SAS Data Studio only when SAS Data Preparation is licensed at your site.

The data quality transforms use SAS Quality Knowledge Base (QKB). QKB is a collection of locales and other information that is referenced during data analysis and data cleansing. The data quality transforms apply a QKB locale and a definition to a selected source column. Definitions define data formats for specific types of content and data cleansing. For example, a parse definition for a street address describes how a street address can be parsed into identifiable segments. The **Match and cluster** transform does not use a QKB, but it does require the SAS Data Preparation license.

A locale reflects the language and linguistic conventions of a geographic region. These conventions can include word order or language selection for the country or region.

Note: For all data quality transforms, the size of a new column cannot exceed 1024 bytes. The length of a new column name cannot exceed 247 characters.

Prerequisites for Using Data Quality Transforms

Before you can use data quality transforms, the following prerequisites must be met:

- SAS Data Preparation software offering must be licensed at your site.
- Your administrator must import and configure the QKB in your CAS system. Typically, QKB is imported and configured immediately after the deployment of your SAS Viya software. For more information about importing and configuring a QKB, see "Import a QKB" in *SAS Viya Administration: QKB Management*.

If one or more of these prerequisites is not met, you will receive an error message.

Change Casing

The Casing operation in SAS Data Quality applies intelligent rules to uppercase, lowercase, or proper case your text data. Use Casing when you want your data to be rendered in a specific case for purposes of readability or compliance with a standard.

In many situations, the application of casing rules is straightforward. For example, the abbreviations of US states are always rendered in all-uppercase letters:

Input	Output
nc	NC
tx	TX

Similarly, you can decide to always render your website URLs in all-lowercase letters, such as www.sas.com.

However, there are situations in which casing rules are applied on an exception basis. The exceptions typically apply to proper casing. The following outputs demonstrate that SAS Data Quality applies rule-based and knowledge-based exceptions to correctly apply proper casing. The outputs go beyond the simple capitalization of the first letter in each word:

Input	Output
SAS INSTITUTE	SAS Institute
EBAY INC	eBay Inc
ronald mcdonald	Ronald McDonald

1. To change case, use the **Casing** transform.
2. Open a table. Then click in the left pane.
3. Select **Casing** in the transforms list. Then, click **Add Transform**.
4. Select a source column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.)
5. Select a locale from the **Locale** drop-down menu.
6. Select a definition from the **Casing** drop-down menu.
7. (Optional) Review the value in the **Character length** text box. Make any necessary changes. You can use this text box to increase or decrease the number of characters that appear in each cell in the output column.
8. (Optional) Click **Options for new columns** to change the name of the new column, the column type, or the length. You can indicate a label and format as well. (However, some data types do not support labels.)
9. Click **Run**.

Parse Data

Parsing breaks a text string into a set of constituent sub-strings. The sub-strings represent a set of semantically atomic portions of the original string. In other words, each of the outputs has meaning in its own right.

For example, consider the sub-strings that can make up a person's name. Most names contain a given name and a family name. A given name and a family name both have meaning of their own. You can use a Parsing operation to generate separate instances of the given name and family name:

Input	Output	
Bob Smith	Given Name	Bob
	Family Name	Smith

If a person's name consists of more than a given name and a family name, then the output includes additional sub-strings:

Input	Output	
Mr Bob C Smith Jr	Prefix	Mr.
	Given Name	Bob
	Middle Name	C
	Family Name	Smith
	Name Suffix	Jr

Different types of data yield different sub-strings:

Input	Output	
Cary, NC 27513	City	Cary
	State/Province	NC
	Postal Code	27513

1. Open a table. Then click in the left pane.
2. Select **Parsing** in the transforms list. Then, click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.)

4 Select a locale from the **Locale** drop-down menu.

5 Select a definition from the **Definition** drop-down menu.

Note: If the definition list is empty for a transform, then the transform is not supported by the locale that you selected.

6 Select tokens by highlighting them in the **Available tokens** list, and then clicking +›.

7 (Optional) Click **Options for new columns** to change the name of the new column, the column type, or the length. You can indicate a label and format as well. (However, some data types do not support labels.)

8 Click **Run**.

Perform Field Extraction

Sometimes, even in relational data, you can have text strings with little or no structure. It might not always be possible to parse such strings into constituent components. Instead, you might want to simply scan the string and extract a few meaningful attributes. An example of such an Extraction operation is as follows:

Input	Output	
William Smith – call after 6pm 919-123-4567	Phone	919-123–4567

1 Open a table. Then click in the left pane.

2 Select **Field extraction** in the transforms list. Then, click **Add Transform**.

3 Select a source column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.)

4 Select a locale from the **Locale** drop-down menu.

5 Select a definition from the **Definition** drop-down menu.

Note: If the definition list is empty for a transform, then the transform is not supported by the locale that you selected.

6 Select tokens by highlighting them in the **Available tokens** list, and then clicking +›.

7 (Optional) Click **Options for new columns** to change the name of the new column, the column type, or the length. You can indicate a label and format as well. (However, some data types do not support labels.)

8 Click **Run**.

Perform Gender Analysis

When your data represents individual persons, you can analyze that data to determine the gender of each person. Gender data can be useful in subsequent statistical analysis, particularly in the domains of medical reporting and product marketing.

The input for Gender Analysis is a text string, and the output is gender code, as shown in the following example:

Input	Output
William Smith	M
Susan B Anthony	F
P Jones	U

The output value U, meaning UNKNOWN, indicates that a gender cannot be determined from the input string.

1. Open a table. Then click in the left pane.
2. Select **Gender analysis** in the transforms list. Then, click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.)

   Note: In order to work, the **Gender analysis** transform needs the complete name of a person. Typically, a database might have columns like prefix, firstname, lastname, and suffix. For example, you cannot pass the firstname column only to **Gender analysis** and successfully perform gender analysis. You can run the **Calculated column** transformation to concatenate the fields into the full name. Then, you can apply the **Gender analysis** transformation to get the correct result.

4. Select a locale from the **Locale** drop-down menu.
5. Select a definition from the **Definition** drop-down menu. See the documentation for your Quality Knowledge Base locale to determine which inputs are accepted by the gender analysis definition that you selected. For example, the documentation for the “Name” gender analysis definition for the English, United States locale in the Quality Knowledge Base for Contact Info version 29 can be found here: Name definition.

   Gender analysis definitions vary. For example, some gender analysis definitions might require a full name including a given name and a family name. Other gender analysis definitions might require non-name data, such as a personal identification number.

   Note: If the definition list is empty for a transform, then the transform is not supported by the locale that you selected.

6. (Optional) Review the value in the **Character length** text box. Make any necessary changes. You can use this text box to increase or decrease the number of characters that appear in each cell in the output column.

7 (Optional) Click **Options for new columns** to change the name of the new column, the column type, or the length. You can indicate a label and format as well. (However, some data types do not support labels.)

8 Click **Run**.

Perform Identification Analysis

To take advantage of the value of your data, you need to know the types of data that you have. Identification Analysis helps you understand your data by naming the type of the content in each variable. Identifying your data enables data profiling, data preparation, data cleansing, and data analysis.

Identification Analysis generates text classifications; it does not determine the database data type of your data, such as CHAR, BOOLEAN, or INTEGER. Instead, Identification Analysis reads text values and determines the semantic type of those values.

The output of Identification Analysis is a named classification that is known as an *identity*. The following example shows how identities are derived from text values:

Input	Identity
William Smith	NAME
500 SAS Campus Drive	ADDRESS
+1 (919) 677–8000	PHONE

Identification Analysis evaluates text values only; it does not process numeric values.

Identification Analysis works best with short text strings in relational database tables. To classify the text in documents, consider using SAS Text Analytics.

In addition to assigning identities to text strings, Identification Analysis can also return confidence scores, as shown in the following example:

Input	Identity	Score
Washington	CITY	90
	STATE/PROVINCE	80
Sara Lee	INDIVIDUAL	60
	ORGANIZATION	50

Confidence scores show you when a text string is nearly certain to apply to one identity. Perhaps more importantly, confidence scores also show you when a text string needs further attention to determine its identity. The preceding examples do not have one identity that is clearly superior. Further investigation into the context of the data is required to assign identities to these text strings.

1 Open a table. Then click in the left pane.

2 Select **Identification analysis** in the transforms list. Then, click **Add Transform**.

3 Select a source column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.)

Note: The **Identification analysis** transform might require some fields to be concatenated.

4 Select a locale from the **Locale** drop-down menu.

5 Select a definition from the **Definition** drop-down menu.

Note: If the definition list is empty for a QKB-based transform, then the transform is not supported by the locale that you selected.

6 (Optional) Review the value in the **Character length** text box. Make any necessary changes. You can use this text box to increase or decrease the number of characters that appear in each cell in the output column.

7 (Optional) Click **Options for new columns** to change the name of the new column, the column type, or the length. You can indicate a label and format as well. (However, some data types do not support labels.)

8 Click **Run**.

Perform Matching Operations Using Matchcodes

Matching operations provide a way to apply fuzzy matching logic in various data cleansing and data integration operations. You can use fuzzy matching logic to find and remove duplicate records, implement fuzzy searches, perform fuzzy joins, and more.

In SAS Data Quality, matching operations are based on the generation of text strings called *matchcodes*. A matchcode is a fuzzy representation of an input text string. If two or more text strings yield the same matchcode, then those strings match. For example, the following records constitute a match:

Input	Input Address	Matchcode
Mr. Robert J. Beckeit	P.O. Box 2270 392 Main St.	M3~$$$$M@M$$$$!KH$$BP$$HHIO$$
Bob Beckett	392 S. Main St. PO Box 2270	M3~$$$$M@M$$$$$!KH$$BP$$HHIO$$
Rob Beckett	392 S. Main St. PO Box 2270	M3~$$$$M@M$$$$$!KH$$BP$$HHIO$$

The matchcodes here are based on the Input column.

Note: If you want to de-duplicate records, and you have not licensed SAS Data Preparation, use the **Code** transform to perform this task using the DATA step. For information, see "Creating Custom Code" on page 13.

1 Open a table. Then click in the left pane.

2 Select **Matchcodes** in the transforms list. Then, click **Add Transform**.

3 Select a source column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.)

4 Select a locale from the **Locale** drop-down menu.

5 Select a definition from the **Definition** drop-down menu.

Note: If the definition list is empty for a transform, then the transform is not supported by the locale that you selected.

6 (Optional) Review the value for **Sensitivity**. Make any necessary changes.

7 (Optional) Review the value in the **Character length** text box. Make any necessary changes. You can use this text box to increase or decrease the number of characters that appear in each cell in the output column.

8 (Optional) Click **Options for new columns** to change the name of the new column, the column type, or the length. You can indicate a label and format as well. (However, some data types do not support labels.)

9 Click **Run**.

Remove Duplicates

Sometimes a piece of data in a data source is found in more than one row. For example, a row for the same address or transaction might have been inserted from multiple data sources. Furthermore, these multiple rows can contain similarities and differences in their values. Two rows of address values might contain identical last names and street names but different state values (such as Virginia and VA).

The **Remove duplicates** transform enables you to remove all but one of these duplicated rows. You can specify the columns responsible for data problems and remove data duplicated only in them. You can also search for data in every column and remove data duplicates from them.

1 Open a table. Then click in the left pane.

2 Select **Remove duplicates** in the transforms list. Then, click **Add Transform**.

3 (Optional) If you need to remove duplicates found in all of the columns, select the **Remove duplicates across all columns** check box.

4 (Optional) If you need to remove duplicates found in a selected group of columns, specify one or more columns in the **Select columns to group by** section.

5 Click **Run**.

Standardize Data

Standardization transforms text strings by rendering them in a preferred format. A Standardization operation can rearrange words, change individual words or symbols, and apply casing rules.

The particular transformations that are applied to a text string are determined by the semantic type of the data. You provide the semantic type of the data as a context when you invoke a Standardization operation. For example, if you specify that the string "Virginia" is a US state, then the transformed value is "VA". In contrast, if you specify that the string "SMITH, VIRGINIA" is the name of an individual, then the transformed value is "Virginia Smith".

Here are some example inputs and outputs of a Standardization operation:

Input	Output
north carolina	NC
JONES, DOCTOR JAMES	Dr James Jones
9195425602	(919) 542-5602
apartment 3 100 main street	100 Main St, Apt 3

1. Open a table. Then click in the left pane.
2. Select **Standardize** in the transforms list. Then, click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu. (Only the first 1000 columns are available in this field.)
4. Select a locale from the **Locale** drop-down menu.
5. Select a definition from the **Definition** drop-down menu.

   Note: If the definition list is empty for a transform, then the transform is not supported by the locale that you selected.

6. (Optional) Review the value in the **Character length** text box. Make any necessary changes. You can use this text box to increase or decrease the number of characters that appear in each cell in the output column.
7. (Optional) Click **Options for new columns** to change the name of the new column, the column type, or the length. You can indicate a label and format as well. (However, some data types do not support labels.)
8. Click **Run**.

Working with Match and Cluster Rows

Matching and Clustering

Use the **Match and cluster** transform to match data based on user-defined match rules. These rules indicate which rows form a single entity. The matching rows are clustered together and given the same cluster ID.

1 Open a table. Then click in the left pane.

2 Select **Match and cluster** in the transforms list. Then, click **Add Transform**.

3 Keep the default value in the **New column name** field, or enter an appropriate column name. The **Match and cluster** transform creates this new column to store the cluster IDs. Rows that share the same cluster IDs belong together and represent a single entity.

4 (Optional) Click **Options for new columns** at the bottom of the **Match and cluster** transform to set options for the new column. Changes that you make to the **Name** and the **Label** fields are saved when you close the Options for new columns window. However, some data types do not support labels. Also, changes that you make to the **Format** field are not saved when you close the Options for new columns window.

Note: You can set the data type of the cluster ID in the **Type** field in the Options for new columns window. Click ▾. The available options are **Double** (eight bytes) and **Char** (twelve bytes). The default data type is **Double**.

5 Create one or more match rules. For more information, see "Creating Match Rules" on page 25.

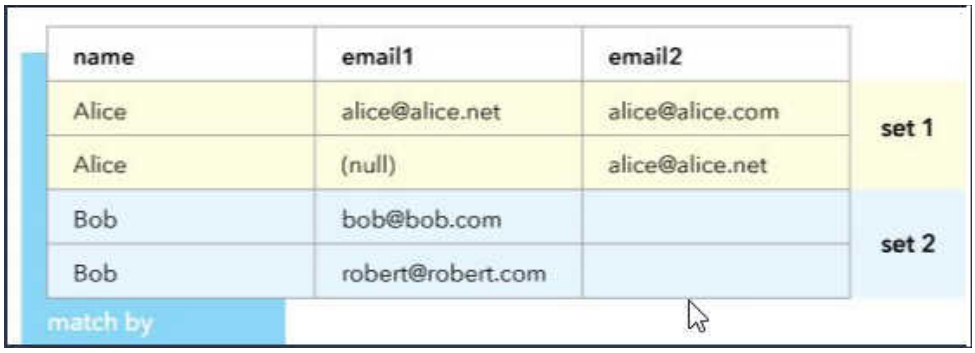
6 (Optional) Set advanced options for the **Match and cluster** transform. For more information, see "Setting Advanced Options" on page 26.

Creating Match Rules

Match rules are used to discover records that share an identity. The match rule specifies certain columns. When different records have identical values for those columns, the records are considered to share the same identity. The records sharing an identity are annotated with the same cluster ID.

The simplest example of a match rule uses a single column, which is *name* in this example. The rows in the table that have the same value for this *name* column are given the same cluster ID and belong to a cluster set. Therefore, the result yields two sets, as shown in the following figure:

Figure 2 *Single Column Rule*



This rule yields four sets. The two Alice records are separated into two sets because the values for *email1* differ. The Bob records are separated in the same way.

The match rule for the Multiple Columns Rule explained above requires the following steps:

1 Select **name** in the first **Column** field.

2 Click +.

3 Select **email1** in the second **Column** field.

Setting Advanced Options

The following advanced options are available for the **Match and cluster** transform:

Interpret empty strings as null values
when selected, treats empty strings in your data as null values.

Allow null values to match
when selected, considers null values in your data as a match and clusters them together according to the match rules. By default, null values are interpreted as missing values and not considered as a match.

Column
Click ▾ in the **Column** field when you need to specify a column that contains the do not cluster flag. A row is excluded from matching if the value of the selected column for that row is interpreted as TRUE.

The only data types allowed for the do not cluster column are VARCHAR, DOUBLE, INT32, and INT64. If a column with a different type is selected, an error message is returned and the transform does not run. For numeric column types, the following values are interpreted as FALSE: `0`, `NULL` , or a missing value. All other values are interpreted as TRUE. For character column types, the following values are interpreted at TRUE: `true` (case-insensitive) and `1`. All other values are interpreted as FALSE.

If the specified (do not cluster) column value for the row is interpreted as TRUE, the following information applies:

- The transform does not attempt to match the row to any other rows in the input data.
- The row is placed in a cluster by itself and given a unique cluster ID.

Appending Data to a Table

You can add incremental data to a single table. For example, if sales data is loaded on a daily basis in separate tables, you can create a table that shows cumulative sales data by appending all of the daily tables together.

1. Open a table. Then click in the left pane. This table is used as the base table.
2. Select **Append** in the transforms list. Then, click **Add Transform**.
3. Click **Browse** to select a table to append to the base table.
4. In the Choose Data window, select a table, and then click **OK**.

   Note: The table that you want to append to the base table must be loaded on the same server as the base table. Only tables loaded on the same server as the base table are displayed in the Choose Data window.

5. (Optional) To append additional tables, click +, and then choose a table.
6. Click **Run**.

Working with Joins

Overview

The Join transform enables you to join two or more tables together. You can select the appropriate join type and select and rename columns.

Join Tables

1 Open a table. Then click in the left pane.

2 Select **Join** in the transforms list. Then, click **Add Transform**. This table is used as the base table.

CAUTION
To avoid poor performance when you run joins, you should not join tables that contain more than 250 columns. Clear the **Select all columns** check box. Then, click **Select** to select a subset of available columns. See "Select or Rename Join Columns" on page 28 for more information. (A **Rename** tab is also available to help you manage your columns). You can also select the **No duplicate rows** check box to ensure that your join output does not include duplicate rows. Duplicate rows can also slow join performance. Finally, joined tables must come from the same server as the original input table. Switching to an input table from a different server when there are one or more join or append transformations causes an error. The joined or appended tables must be reselected in the user interface for each of those transformations.

3 Choose a second table to use in the join by clicking .

In the Choose Data window, select a table, and then click **OK**. Only tables loaded on the same server as the base table are displayed in the Choose Data window.

4 (Optional) To change the join type, click , and select the join type from the menu. Options include inner, left, right, and full.

5 When you specify the tables that you want to join, the join condition is determined automatically by matching column names and data types. You can change the join condition by selecting a different column in the drop-down menus that appear under each table name.

You can add join conditions by clicking .

6 (Optional) Add tables to the join by clicking .

You can remove a table from the join by clicking next to the table name.

Note: You can join up to 256 tables.

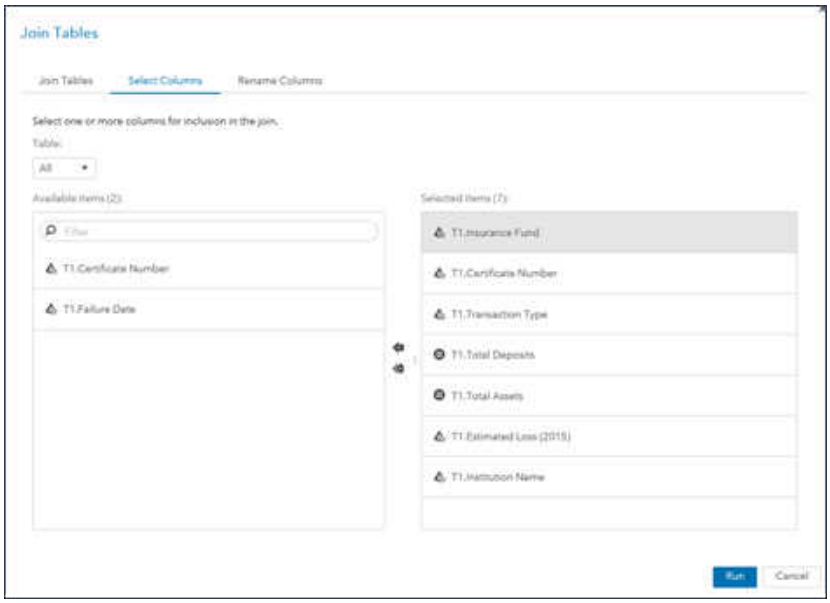
7 Click **Run**.

8 (Optional) You can modify the steps that you took to join the tables by clicking **Edit Join**.

Select or Rename Join Columns

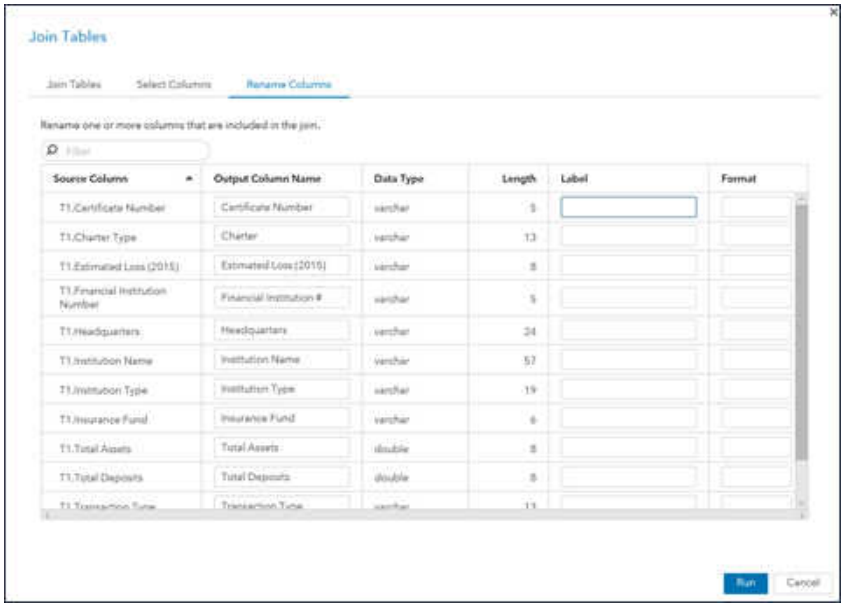
You can use **Select Columns** and **Rename Columns** tabs when you need to select or rename the columns included in the output of your join. Selecting columns enables you to base your join on a subset of the columns available in the input tables for the join. Renaming columns enables you to change the names, labels, and formats of the columns to meet your needs. (However, some data types do not support labels.)

Figure 3 *Select Columns*



By default, all columns are displayed in the **Selected items** list. Select the columns that you do not want to include in the join and move them from the **Selected items** list to the **Available items** list. You can use the **Table** drop-down field and the **Filter** field to filter the **Available items**.

Figure 4 *Rename Columns*



You can use the **Rename Columns** tab to edit the fields in the Output Column Name, Label, and Format columns for each row. (However, some data types do not support labels.) You cannot change the values in the Data Type and Length columns.

In this example, the Charter Type and Financial Institution Number source columns now have changed output column names: *Charter* and *Financial Institution #*. The **Filter** field enables you to filter the columns listed in the column for the Source Column.

When you are finished working with the **Select Columns** and **Rename Columns** tabs, click **Run** to make the changes in your join.

Filtering Data

Filter Data

There is no limit to the number of filters that you can apply to a table. To filter data:

1. Open a table. Then click in the left pane.
2. Select **Filter** in the transforms list. Then, click **Add Transform**.
3. Select a column from the **Column** drop-down menu.
4. Select an operator from the **Operator** field. For more information about operators, see “Filter Operators” on page 30.
5. Enter a filter value in the **Value** text box, or click **Browse**.

   Here is some key information about **Value**:

- You can filter on multiple values only using the **In** and **Not in** operators. Enter multiple values by pressing the **Enter** key after each value, or click **Browse**.
- If you choose the **Match** operator or the **Not match** operator, then the value that you enter must be surrounded by leading and trailing forward slashes (**/**). (For example, **/regularExpression/**).
- The **Filter using formatted values** check box appears only if a format is associated with the column that you selected. To filter using formatted column values, make sure that the **Filter using formatted values** check box is selected, and then enter the formatted value in the **Value** text box, or click **Browse**. To filter using raw column values, deselect the **Filter using formatted values** check box, and then enter the raw value in the **Value** field, or click **Browse**. If you enter a formatted value in the **Value** text box, make sure it matches the formatted value from the table exactly, including the case, length, and decimal places.

  When filtering numeric columns that have an associated format, and the **Filter using formatted values** check box is selected, SAS Data Studio converts numeric values to strings to perform comparisons. This might lead to unexpected results, especially when using operators other than **Equal to** and **Not equal to**. If unexpected results occur, you can deselect the **Filter using formatted values** check box to filter the table based on raw numeric values in the column.

6 (Optional) Add additional filter conditions by clicking **+**.

Note: The filter transform uses an **AND** operator when you filter on multiple conditions. This means that the transform returns only rows where all conditions are met. Filtering using the **OR** operator, where rows that meet either condition are returned, is not supported.

7 Click **Run**.

Here are a few key points about the Filter transform:

- If there are white spaces in front of a value, then the sort order of the values in the Filter window might be different from what you expect.
- When filtering columns with numeric formats, SAS Data Studio converts numeric values to character values, and unexpected results might occur. If unexpected results occur, deselect the **Filter using formatted values** check box to filter the table based on raw column values.
- The **Filter** transform displays up to 1000 distinct values only. For columns that have more than 1000 distinct values, you might not receive results when searching for values in the **Choose a Filter Value** window. If this occurs, increase the number of distinct values by changing the **maximumFrequencyValues** configuration property in SAS Environment Manager.

Filter Operators

Equal to
: returns all rows that contain a value that is equal to the value that you enter.

Not equal to
: returns all rows that contain a value that is not equal to the value that you enter.

Greater than
: returns all rows that contain a value that is greater than the value that you enter.

Less than
: returns all rows that contain a value that is less than the value that you enter.

Greater than or equal to
: returns all rows that contain a value that is greater than or equal to the value that you enter.

Less than or equal to
: returns all rows that contain a value that is less than or equal to the value that you enter.

Between
: returns all rows where the first value is within the range defined by the second and third values, including the bounding values.

In
: returns all rows where the column is in the value that you enter. Enter a filter value in the **Value** text box, or click **Browse**.

Not in
: returns all rows where the column is not in the value that you enter. Enter a filter value in the **Value** text box, or click **Browse**.

Contains
: specifies that a matching value must contain the specified string.

Not contains
: specifies that a matching value must not contain the specified string.

Match
: returns rows that match the pattern that you specify in the regular expression. The value that you enter must be surrounded by leading and trailing forward slashes (**/**). Here is an example: **/regularExpression/**.

Not match
: returns rows that do not match the pattern that you specify in the regular expression. The value that you enter must be surrounded by leading and trailing forward slashes (**/**). Here is an example: **/regularExpression/**.

Null
: returns rows that contain empty cells only.

Not null
: returns all rows except for rows that contain empty cells.

Transposing Columns

Transpose Columns

Note: The **Transpose** transform does not generate its output columns until it runs, so transforms downstream from it will not have any incoming columns if it has not run yet.

Transposing columns moves data from columns to rows. To transpose columns:

1. Open a table. Then click  in the left pane.
2. Select **Transpose** in the transforms list. Then, click **Add Transform**.

3 On the **ID Columns** tab, specify the columns that contain the row values that you want to transform into columns. Click the column name in the **Available items** list, and then click **+**›. You must specify at least one column on the **ID Columns** tab.

   The row values in each column become the new column headings. The column headings of the columns that are transposed are deleted.

4 (Optional) On the **Transpose Columns** tab, specify the columns that contain the data with which you want to populate the output table. Click the column name in the **Available items** list, and then click **+**›.

   Unspecified columns are not included in the output table. If you do not specify any columns on the **Transpose Columns** tab, then all numeric columns will be included in the output table.

   Note: Be careful if you enter ID columns or transpose columns that contain leading whitespace characters. The **Transpose** transform could create column names that contain spaces at the beginning or other unexpected characters. These column names might be invalid in certain transformations. You can use the **Trim whitespace** transform to correct this issue.

5 (Optional) On the **Group By Columns** tab, specify the columns by which the rows of the newly transposed columns are grouped. Click the column name in the **Available items** list, and then click **+**›.

6 (Optional) In the **Options for Output Column Headings** section on the **ID Columns** tab, specify the following options:

   - In the **Include column prefix** field, enter a prefix to be appended to all new column headings.
   - In the **Rename the _NAME_ column** field, enter a name to use as the column heading in place of the **_NAME_** default heading.

7 (Optional) Select the **Eliminate redundant values** field when more than one input row maps to a single output column within a **Group By Columns** group. Selecting this option could lead to the loss of data.

8 Click **Run**.

Example: Transposing Columns

Here is an example of how to transpose columns. Using the following source table:

STUDENT	CLASS	GRADE	CREDIT
Ann	Math101	A	4.0
Ann	English101	B+	4.0
Ann	Biology101	B+	4.0
Ann	Biolab	A-	2.0
Bob	Math101	A-	4.0
Bob	Chemistry101	A-	4.0
Bob	Chemlab	A-	2.0
Carol	Spanish101	B	4.0
Carol	French101	B	4.0
Carol	History102	C	4.0
Carol	PoliSci111	B	4.0
David	Italian	C	4.0
David	Math210	C	4.0
David	Lit200	B	4.0
Fred	Chemistry101	B	4.0
Fred	ChemLab	B	2.0

Select **Transpose** in the transforms list. Then, drag it into the plan. Add the following columns to each tab:

- For the **ID Columns** tab, select **Student**.
- For the **Transpose Columns** tab, select **Grade**.
- For the **Group By Columns** tab, select **Class**.

After you click **Run**, the resulting table will look like the following image:

CLASS	_NAME_	Ann	Bob	Carol	David	Fre
Anthro111	GRADE					C
Biolab	GRADE	A-				
Biology101	GRADE	B+				
ChemLab	GRADE					B
Chemistry101	GRADE		A-			B
Chemlab	GRADE		A-			
English101	GRADE	B+				
French101	GRADE			B		
History102	GRADE			C		
Italian	GRADE				C	
Lit200	GRADE				B	
Math101	GRADE	A	A-			
Math110	GRADE					A
Math210	GRADE				C	
PoliSci111	GRADE			B		
Spanish101	GRADE			B		

Note: The entire table is not shown in the preceding image.

Creating a Partition Column

You can use the **Analytical Partitioning** transform to create a column in the target table that specifies training, validation, and test values randomly in a new field. These values are used to create partitions for validation purposes in SAS Visual Analytics and SAS Visual Data Mining and Machine Learning. The **Analytical Partitioning** transform is displayed and available in SAS Data Studio only when the SAS Visual Statistics software offering is licensed at your site.

Note: When you partition on a column with high cardinality, the partition can fail if the server runs out of memory. To avoid this issue, base partitions on low cardinality columns.

1 Open a table. Then click in the left pane.

2 Select **Analytic partitioning** in the transforms list. Then, click **Add Transform**.

3 Review the default value in the **New column name** field. Change as needed.

4 Select an appropriate value in the **Method** drop-down field:

- **Simple random** specifies that the partitioning values are generated randomly from the values in the source data. When you select **Simple random**, the Available columns section is disabled. However, you can change the values in the **Training**, **Validation**, and **Test** fields.

- **Stratify** specifies that up to four columns are selected to serve as the basis for the partitioning values. The table is divided into the strata that you specify. Then, values are randomly assigned within those strata. **Stratify** is selected by default.

5 If you selected **Stratify** in the **Method** drop-down field, set your stratification values. First, select one or columns from the **Available columns** pane. Then, move the selected columns to the **Selected columns** pane.

6 Review the default values in the **Training**, **Validation**, and **Test** fields. Change as needed.

7 Click **Run**.

8 (Optional) Click **Options for new columns** to access the available options for the new column.

Generating a Unique Identifier

You can use the **Unique identifier** transform to create a column in the target table that contains a unique value for each row in the table. These unique row identifiers are used in text topics in SAS Visual Analytics.

To generate a unique identifier column in a target table:

1 Open a table. Then click  in the left pane.

2 Select **Unique identifier** in the transforms list. Then, click **Add Transform**.

3 Select the **Replace existing column** or **Create new column** check box. If you replace an existing column, use the drop-down menu to select the column that is replaced. If you create a new column, keep the default column name or enter an appropriate column name.

4 Click **Run**.

5 (Optional) Click **Options for new columns** to access the available options for the new column.

Note: The user interface supports lengths between 3 and 8. However, some server types do not support these lengths. In this case, the user-entered length is ignored. Then, the length is set to 8.

Working with Plans

About Plans

A plan is a collection of data transforms or actions performed on a table. It provides a convenient way for you to prepare data in tables. It also helps you to keep track of the changes that you make to tables or to modify or view the history of actions that you made to tables.

Here is some additional key information about working with plans:

- When you open a table, the table is automatically added to the current plan. If you do not have a plan open, a new plan is created.
- When you run one or more plans, the changes that are configured in the plan are applied to the source table. If the plans run successfully, the changes are visible in the SAS Data Studio window.

  You can also see the changes on the **Monitoring** tab in SAS Environment Manager. From the application bar, click  in the top left corner. Select **Manage Environment**. In SAS Environment Manager, click (Jobs) in the navigation bar on the left. Click the **Monitoring** tab in the Jobs window.
- After you make changes to a table, you must run the plan before you can save the table. For more information about saving tables, see "Saving Plans and Tables" on page 46.
- When a plan run ends in an error, you can download the log and look for error messages. If you do not have the proper permissions to load a table in your plan, you might encounter a 403 status code.
- You can add multiple transforms to a plan.
- Concurrency for editing plans is not supported. If two or more people are working simultaneously on a plan, you might overwrite each other's changes to the plan. In this case, it is recommended that you work on a copy of the plan, and then update the master copy of the plan when your changes are ready.
- You cannot save a plan to the **My Favorites** folder. Only shortcuts can be saved to the **My Favorites** folder.
- Data plans that were created in the SAS Visual Data Builder component of SAS Visual Analytics 8.1 cannot be migrated to SAS Visual Analytics 8.4 or later. You must create new data plans in the SAS Data Studio component of SAS Visual Analytics 8.4 or later.

Note: When a transform is moved to a different position in the series of transforms within a plan, you might encounter an error message similar to the following: "Unable to locate input column Action.<*Local_data.*> for action Action.<*Local_data.*>."

This message is displayed when some of the columns that were available to a transform are no longer available due to the change in position of the transform in the plan sequence.

The solution to this issue is to edit the transform by removing the offending column. In certain rare cases, you might need to drop and redo the entire transformation.

Work with Plans

You can perform the following tasks with plans:

Table 3 *Plan Tasks*

Task	Steps	Additional Information
Create a new plan	Click the **New Plan** button in the workspace, and open a table.	

Task	Steps	Additional Information
Open an existing plan	Click the **Open Plan** button in the workspace, select a plan, and then click **Open**.	If a plan includes an unloaded table, then the table is loaded automatically when you open the plan. If the table is not loaded because it has been deleted, then the plan still opens. You will receive an error message stating that the source table was not found.
Change the source table in a plan	To change the source table used in a plan, click ⋮ on the navigation bar, and then click **Change source table**.	
View plan actions	To view plan actions, open the Plan window by clicking ⊞ in the right pane.	The Plan window contains a toolbar with the following buttons:  Insert transforms above or below  Move transforms up or down  Remove transforms Be sure to save your plan whenever you insert, move, or delete its transforms.
Modify plan actions	In the workspace, click the name of the transform that you want to modify. After you make your changes, click **Run**.	
Undo plan actions	To undo a plan action, click ↺ in the Plan window. The table updates automatically.	You can undo only the last action that was performed on a table in a plan. If you want to undo actions that were made previously, you must first undo each action that was subsequently made.
Save a plan	Click **Save** on the navigation bar to save a previously saved plan. If the plan was not saved previously, or if you want to save the plan with a different name, click ⋮ on the navigation bar, and then click **Save as**.	
Export a plan	Save a plan in SAS Data Studio to a SAS folder. Open SAS Environment Manager and export the plan.	See "Export Content" in *SAS Viya Administration: Folders*.
Close a plan	Click ✕ on the tab for the plan that you want to close.	

Task	Steps	Additional Information
Delete a plan	Click the **Open Plan** button in the workspace, navigate to the plan that you want to delete, and select it. Click in the upper right-hand side of the window. Click **Delete** in the warning window that is displayed.	
Download log	Click , and then click **Download log**.	
Download code	Click , and then click **Download code**.	

Working with Suggestions

About Suggestions

The Suggestions feature uses machine learning to analyze your data and suggest transforms that you can add to your SAS Data Studio plans. These suggestions can address problems in the data and enhance the performance of your data jobs. The Suggestions feature is available in SAS Data Studio only when SAS Data Preparation is licensed at your site.

The Suggestions feature analyzes your data with a set of models that have been registered for your installation. These models are registered from the Data Table Provider Service. For more information, see Step 3 on page 39 in “Generating and Applying Suggestions” on page 38.

The models are designed to generate appropriate suggestions for the type of data they encounter. Some of the suggestions should lead you to transforms that you frequently apply to your data. Others, though, might expose you to the possibilities in unfamiliar transforms and actions.

Generating and Applying Suggestions

1 Open a data source, and then click in the left pane to access the Suggestions pane.

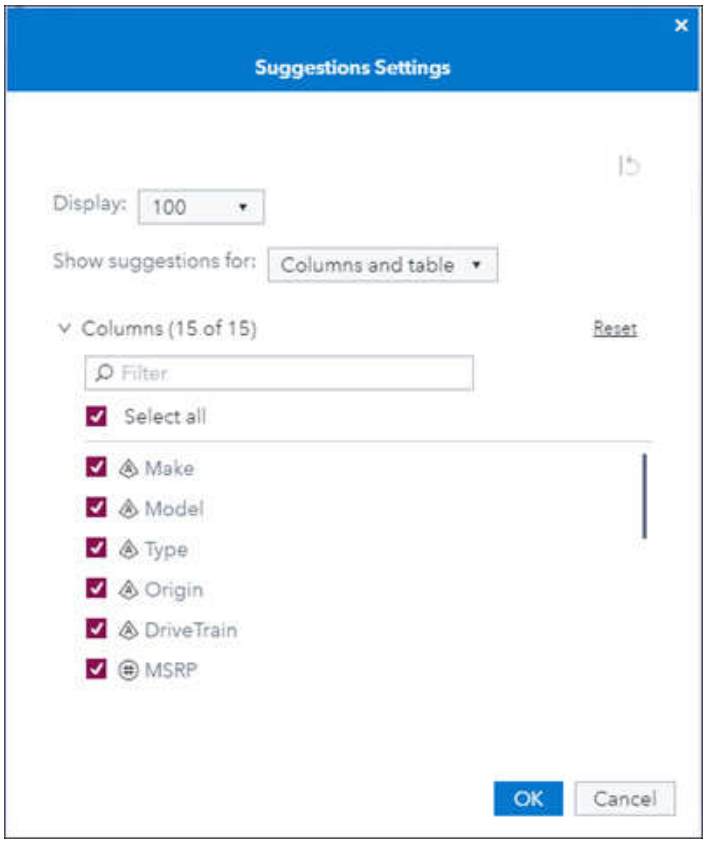
2 (Optional) Click **Settings** to determine the scope of your suggestions. To do this, click to access the **Suggestions Settings**. You can use this window to specify the number of suggestions that are displayed and determine the scope of the suggestions process. The **Show suggestions for** field contains items to run suggestions for the **Columns and the table** as a whole, **Columns only**, and **Table only**. The Columns section enables you to filter the list of columns and select all

or some of the columns available in the table. You can click **Reset** to reset just the column settings or . The default settings are as follows:

- *100* in the **Display** field
- *Columns and table* in the **Show suggestions for field**
- **Select all** checked

A reset to defaults button is also available at the main level of the Suggestions feature.

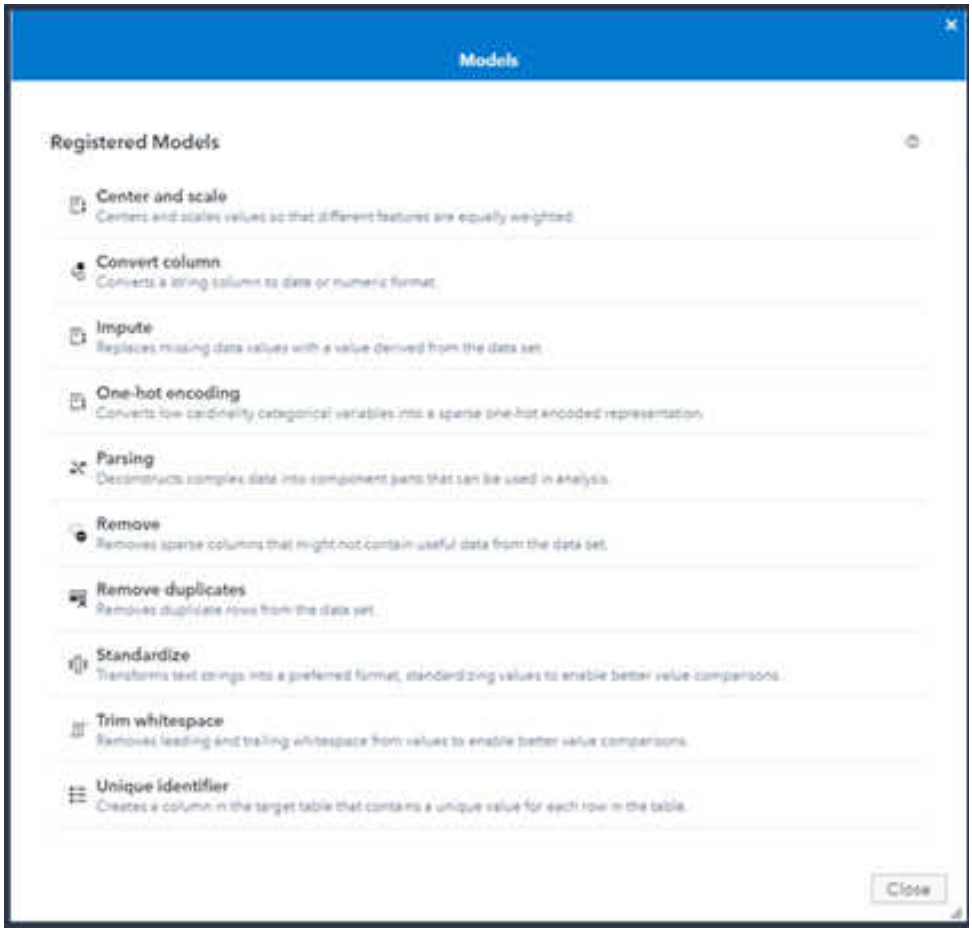
Figure 5 *Suggestions Settings Window*



3 Click **OK** to apply the suggestions settings. In this case, the table and all of the columns are enabled.

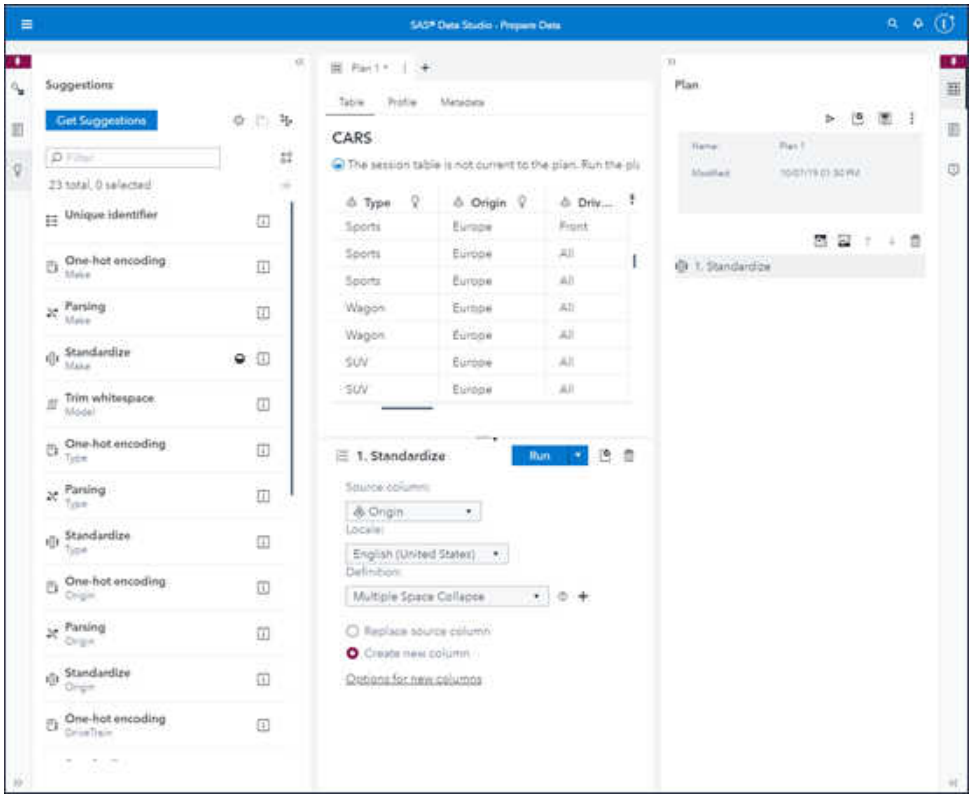
4 Click **Get Suggestions**. If the models have been registered, the Suggestions feature generates a list of suggestions, if any are appropriate. If models have not been registered, a link prompting you to register them is displayed. A similar link is available in the Models window, which you can access by clicking . Once the models are registered, you can review their descriptions in the **Models** window.

Figure 6 Models Window



5 Click **Get Suggestions** to analyze your data with the models registered for Suggestions.

Figure 7 *Sample Suggestions*



One suggestion has been added to the plan and configured. Click **Run** to see in the center pane a preview of the transform's effect on the data.

Status indicators are displayed next to the suggestions in the suggestions list:

- Added but not run
- Run

You can click **Get Suggestions** to refresh the suggestions list. The suggestions that have already been run are filtered out of the list.

Usage Notes for Suggestions

Only CASL code is enabled for suggestions

Some suggestions such as Impute, Center and Scale, and One-hot encoding are displayed as code transformations. Note that only CASL code is enabled for these suggestions. You can select **DATA step** in the drop-down field, but the code does not update and the transform does not run.

Default Models caslib provided

A default Models caslib is provided for the <cas-shared-default> server. If you need to register models on another server, that server also must have a Models caslib. Otherwise, models cannot be registered successfully through the user interface.

Quality Knowledge Base (QKB) required

You must have a Quality Knowledge Base (QKB) registered to generate suggestions. Specifically, SAS Quality Knowledge Base (QKB) for Contact Information 29 must be installed. If your site has licensed SAS Data Preparation, a QKB is typically installed with the rest of the software. A QKB

can also be installed later, as described in “Set the Default QKB and the Default Locale” in SAS Viya Administration: QKB Management.

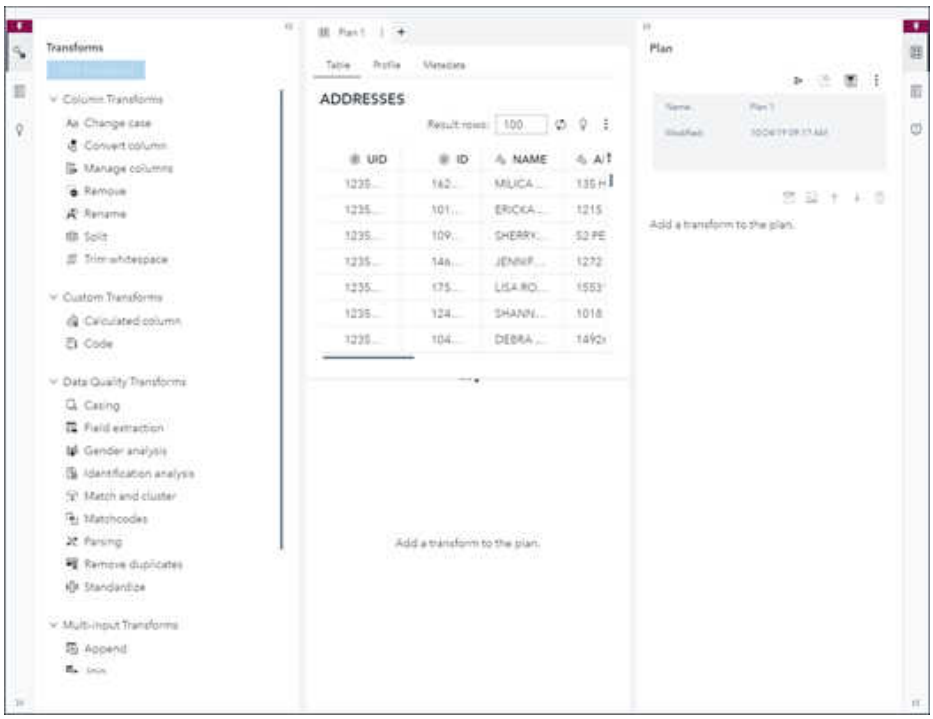
The Suggestions feature uses identity analysis from the QKB. If you try to generate suggestions on a table whose server lacks a registered QKB, suggestions fail.

Suggestions does not support varbinary columns

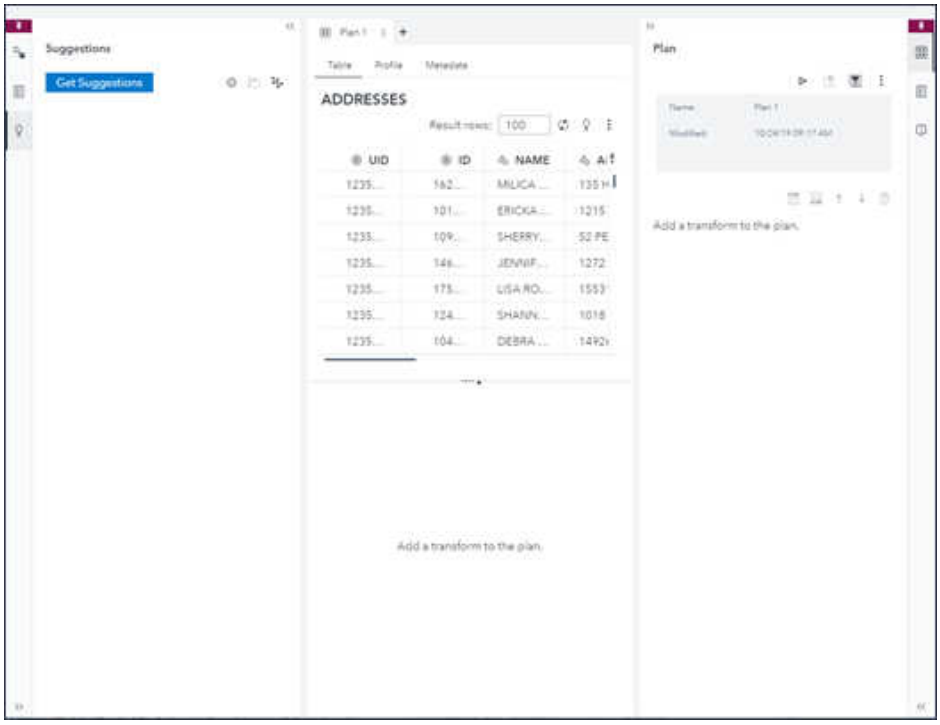
Do not run Suggestions on tables that contain varbinary columns, such as image directory tables.

Example: Create and Run a Plan That Uses Suggestions

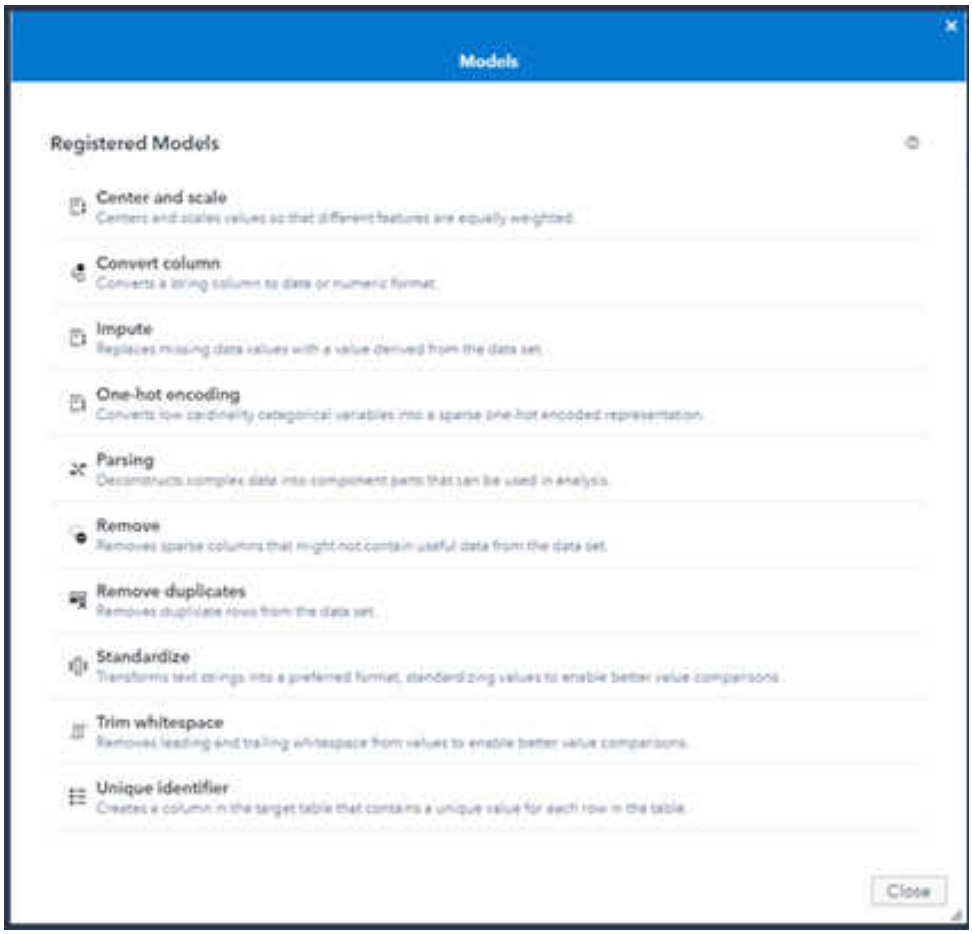
The Suggestions feature enables you to use machine learning to analyze your data and select transforms to add to your SAS Data Studio plans. This example shows how to apply this power to a simple table. For more information about suggestions, see “Working with Suggestions” on page 38.



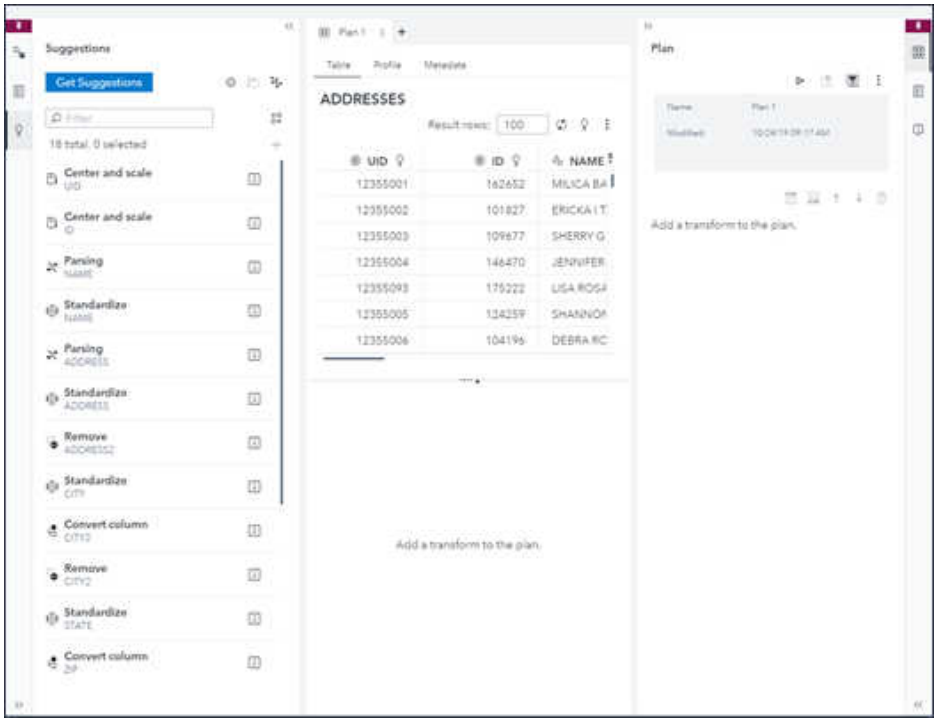
This SAS Data Studio plan prepares the Addresses table for further analysis. Click  at the left edge of the window to access the Suggestions pane.



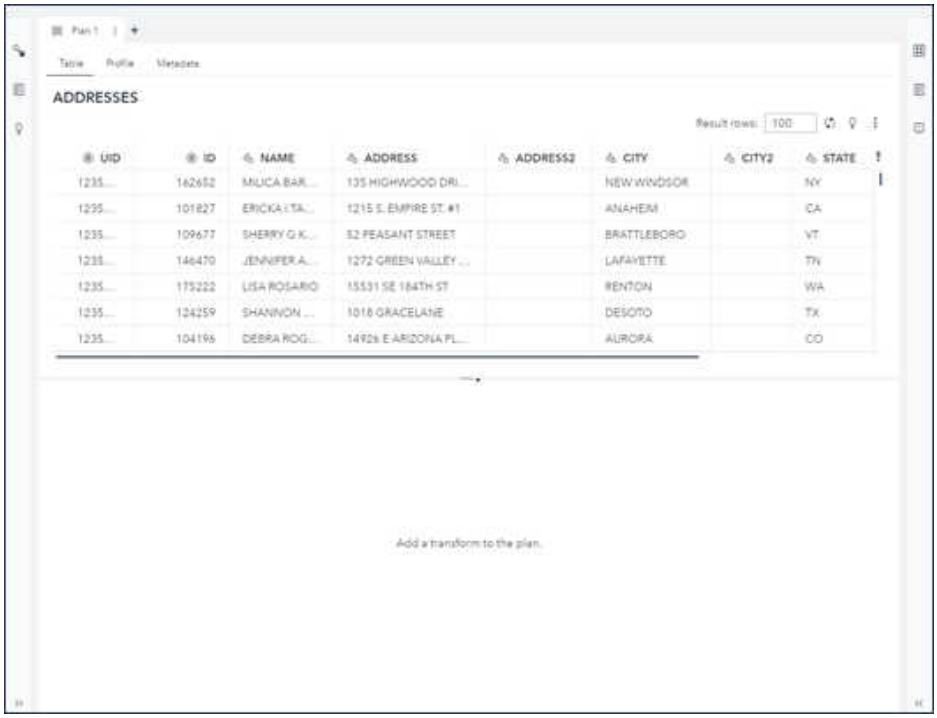
The Suggestions pane enables you to compare all or part of your table to a set of models that have been registered for your installation. Your data is analyzed by these models and a list of suggested transforms for your data is displayed.



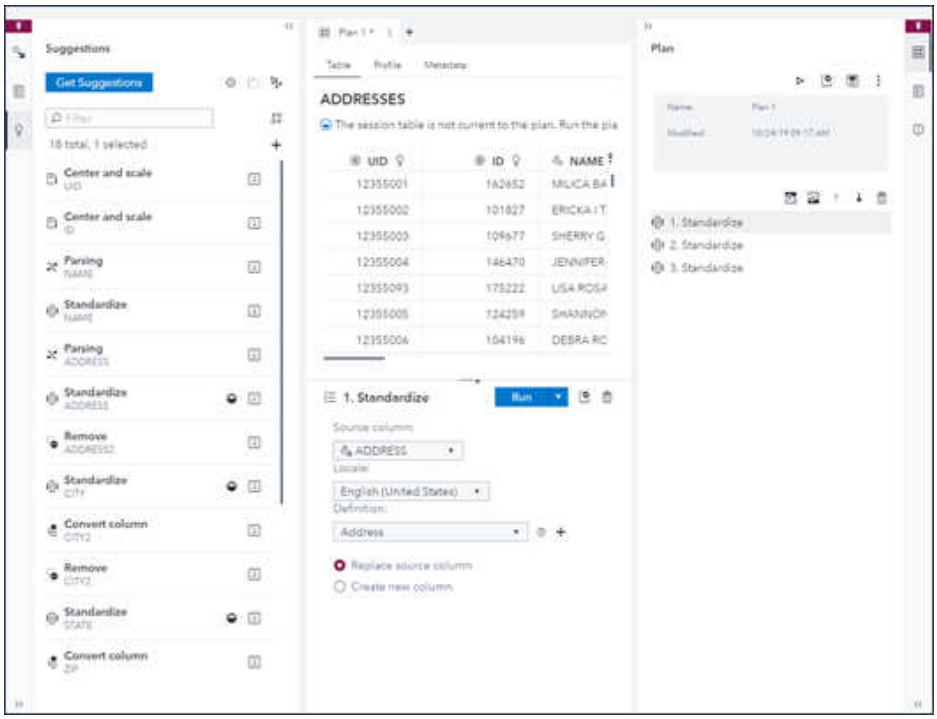
You can use the **Models** window to review descriptions of the models and register the models when they have not been registered already.



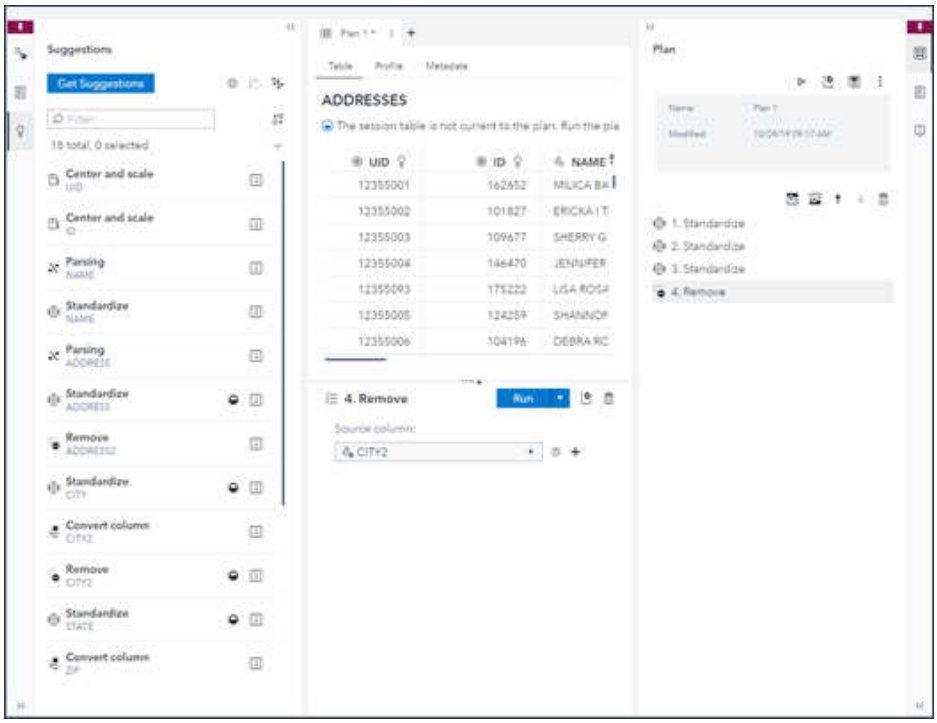
A list of suggestions for your table, the columns, or both is generated when you click **Get Suggestions** in the Suggestions pane. Some columns have more than one suggestion, and some suggestions are made for more than one column.



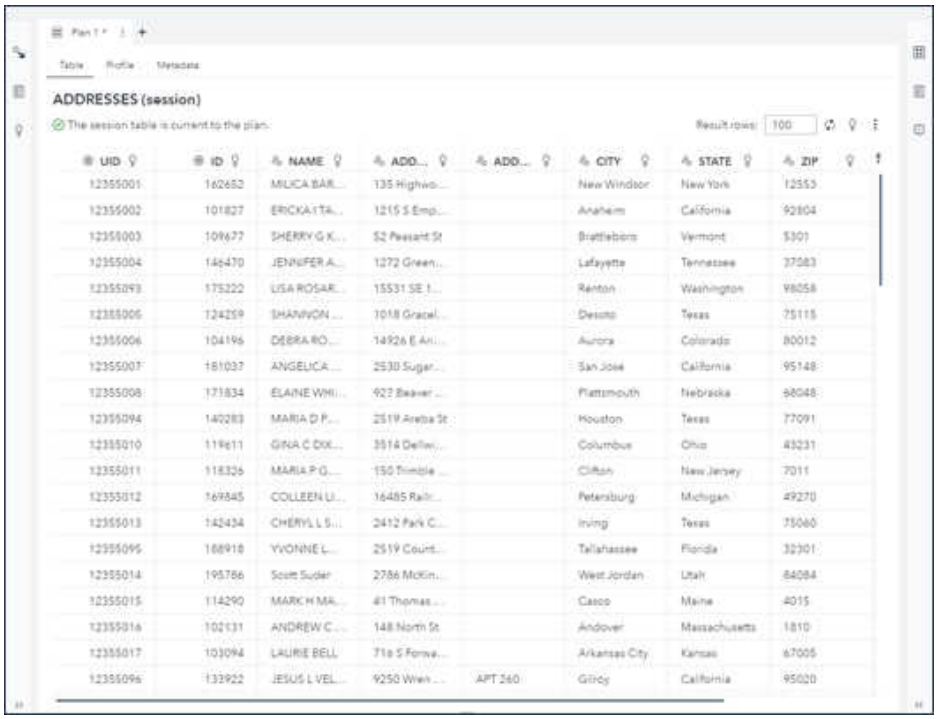
After you get and review the suggestions that are generated, you should examine your data and determine which suggestions suit your purposes. In this case, some of the formatting looks awkward and two columns look empty.



Based on an examination of the data, the standardize transform is added to the plan for the ADDRESS, CITY, and STATE columns. (Click + to add the selected transforms to the plan.) Each transform is configured to use the English (United States) locale and the appropriate definition from the Quality Knowledge Base (QKB).



The CITY2 column contains no data, so the suggested Remove transform is added to the plan. Removed is configured with a **Source column** of CITY2. To run the plan, click ▷ in the Plan pane.



The updated table contains the reformatted ADDRESS, CITY, and STATE columns. The empty CITY2 column has been removed.

Saving Plans and Tables

Overview

You can save plans and tables, plans, or tables. You should use the **Save plan and table**, **Save plan**, and **Save table** check boxes to select the scope that meets your business needs.

Saving Tables

When you are finished making changes to a table, you must save the table to use it in other applications. If you close the plan before saving, any changes that you made are lost.

Here are a few key points about saving tables:

- Overwriting an existing source table might invalidate plans that reference that table. In this case, a warning message is displayed when you attempt to save the table if the target table name is the same as the source table name. Due to structural changes that occur when you overwrite a source table, transforms in your open plan might appear incomplete.
- You can use the selection arrow in the **Format** field to select any format supported by the target table library. This Format selector for the target table is available when certain types of libraries are selected in the **Library** field. The selector enables you to copy the source table to a target

table in a different format. Select a format that is appropriate for your goals. For more information, see "Copying Tables in One Format to Another Format" in *SAS Data Explorer: User's Guide*.

- Table names cannot exceed 247 characters.

Note: The **Save as in-memory table only** check box enables you to load the table to memory without saving a physical copy of the table to the target destination. Only users who have the Promote permission on the target caslib can successfully perform an import using this option.

Use this option whenever you want to work with a table without incurring the costs in time and storage space required to write a physical copy. This option can provide a significant performance boost for small tables.

Save Plans and Tables

Before you save a table, make sure that you click **Run** to run the plan.

1. Click **Save** to access the Save As window. You can also click ⋮ and select **Save as**.
2. Click the **Save plan and table** check box.
3. Specify the plan **Name** and **Type** in the appropriate fields.
4. Review or change the name of the table in the **Table name** field. You can choose to give the table a new name, or you can specify an existing table to overwrite. If you want to overwrite the source table, remove the **_NEW** extension that is appended to the name of the source table by default.
5. (Optional) Add a label for the table in the **Label** field. However, some data types do not support labels.
6. (Optional) Review the library in the **Library** field, and make changes if necessary. For example, you can select an Oracle library to save the table with an Oracle schema.
7. Click **Save**.
8. If the table that you specified already exists, a window appears, asking if you want to overwrite the table. Click **Yes** to overwrite the table. Otherwise, click **No**, and specify a different table name.

You can save a plan without saving a table.

1. Select the **Save plan** check box.
2. Process the **Name** and **Type** fields.
3. Click **Save**.

You can save a table without saving a plan.

1. Select the **Save table** check box.
2. Process the **Table name** and **Label**, and **Library** fields. (However, some data types do not support labels.)
3. Click **Save**.

Creating Jobs for Scheduling

A plan can run slowly when the source table is large or the job is complicated. Fortunately, you can create a job that you can run or schedule for execution at a later time in SAS Environment Manager. The separation between creating a job and running it ensures that you can run large jobs at an appropriate time and reduce the load on your system.

1. Open a table.
2. Save the current plan and its target table.
3. Click ⋮ in the toolbar.
4. Click **Create job** to access the Create job window.
5. Specify a name and description for the job and click **OK**. The name cannot be longer than 100 characters. A status message is displayed, stating whether the job was created successfully.
6. From the application bar, click ☰ in the top left corner. Select **Manage Environment**.
7. In SAS Environment Manager, click ⚙ (**Jobs**) in the navigation bar on the left. The name of the job that you created is displayed on the **Scheduling** tab of the Jobs window. For more information about using this window, see "Jobs and Flows: How To (Jobs)" in *SAS Viya Administration: Jobs and Flows*.
8. (Optional) After the job runs, a copy of the table or file is loaded to memory on the CAS server that is specified in the caslib. Select the copy from the **Available** tab or the **Data Sources** tab.
9. (Optional) Change the expiration time for a job using the **interactiveJobExpiresAfter** and **saveTableJobExpiresAfter** configuration properties in SAS Environment Manager.

Modifying SAS Data Studio Settings

There are settings that are specific to SAS Data Studio, and there are global settings that are applied to all SAS web applications.

Settings for SAS Data Studio are saved on a per-user basis. All of your settings persist between sessions.

1. In the application bar, click your logon name, and then click **Settings**.
2. Click **Data Studio** in the side menu.
3. Click **General** to access the general settings for SAS Data Studio. The following settings are available:

   Default target location
   Click 🗃 to select a caslib where target tables are stored by default. This caslib serves as the default for various operations, including operations on the **Import** tab. A change to this setting is applied to the next table that uses the default target location. A change is not applied

retroactively. If a default target location is not set in this field, the default CAS server and caslib for the CAS Management Service are used. For more information, administrators can see For more information, administrators can refer to "CAS Management Service" in *SAS Viya Administration: Configuration Properties*.

Default locale for Quality Knowledge Base
Use this selector to specify the locale that is used by the Analyze column contents while running profile option. If the **Use the default server** check box is selected, the software uses the default locale specified for the CAS server for the profiled table. If no default locale has been defined for this server, the profile fails. You can use this control to select a locale. The locale should be appropriate for the data that you are profiling. For example, a table of names and addresses from the United States should be profiled with the English-United States locale. Locales from all QKBs on all CAS servers that appear on the **Data Sources** tab are listed here. Duplicate locales are filtered out.

4 Click **Geographic Mapping** to accept the terms and conditions for Esri ArcGIS Online Services.

5 Click **Profile** to access profiling settings. The following settings are available:

Apply formats to variables when profiling data
Select this check box to apply formats to the output data for the **Run profile** option. Some data is more meaningful when it is formatted. For example, currency values might be more meaningful if they are formatted as currency rather than as integers. The impact on data profiling performance is usually acceptable. For more information about profiling, see "Profiling Data" in *SAS Data Explorer: User's Guide*.

Analyze column contents while running profile
Select this check box to trigger column content analysis during profiling. If the analysis can determine what type of content is in the column, then the column is tagged with the appropriate content tag. For example, a column that contains street addresses might get a **Street Address** tag. Content tags can be used by other software, such as SAS Visual Text Analytics. The analysis is based on the locale that is specified in the **Default locale for Quality Knowledge Base** selector. Content analysis impacts profiling performance. Prerequisites for this option are described in "Enable Automatic Content Tagging for Columns in a Table" in *SAS Data Explorer: User's Guide*

6 Click **Close** to apply your changes.

> **TIP** When you click **Reset**, the settings revert to their original configurations.