SAS® Data Studio 2.1: User's Guide

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.

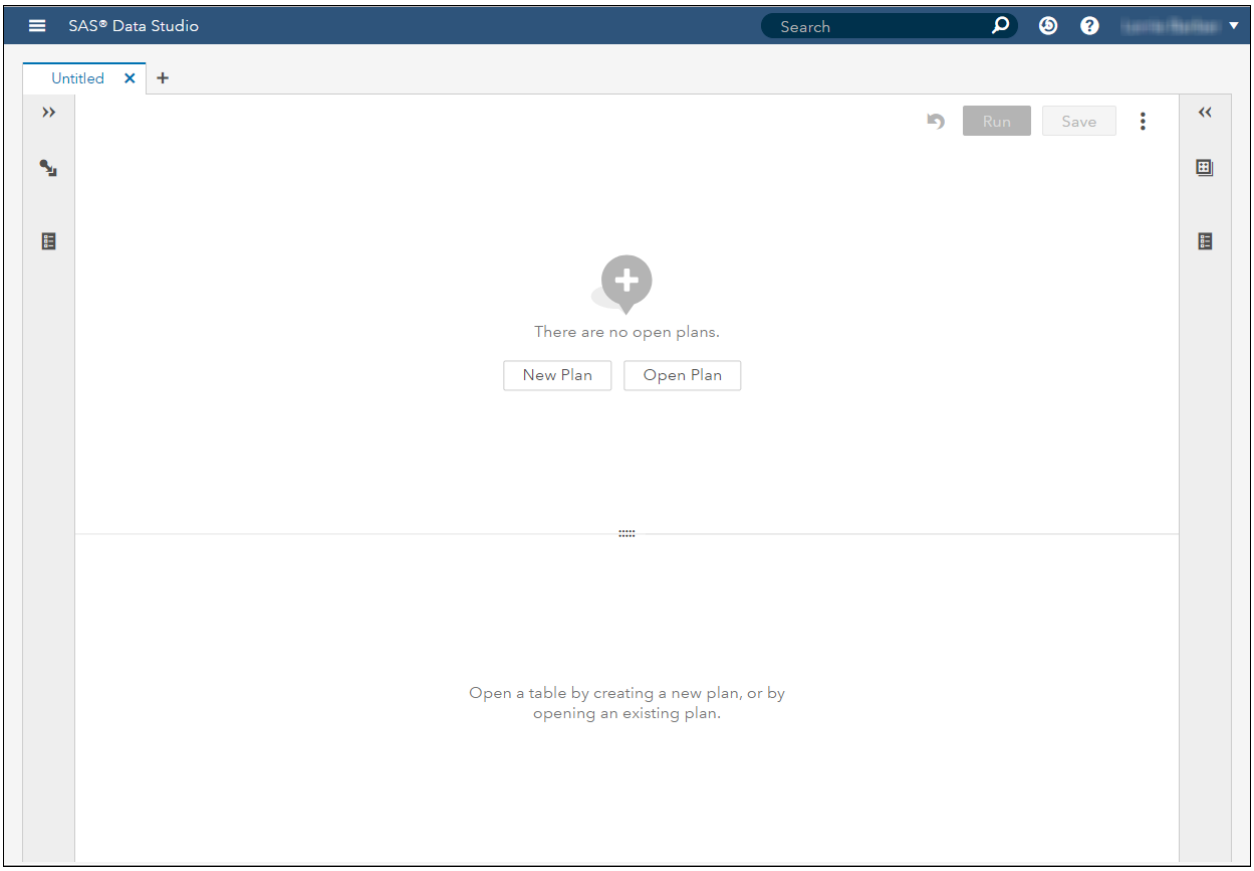
Create plans
: You can create a plan, which is a collection of actions or data transforms performed on a table.

View profiles
: You can 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.

Your First Look at the Interface

The SAS Data Studio interface enables you to prepare and view data. In general, data transformations are initiated from the left pane. Here are more details about the interface:



- The application bar at the top 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 "Modify SAS Data Studio Settings" on page 22. For more information about search and global settings, see *General Usage Help for SAS Viya Web Applications*.
- The toolbar enables you to run and save plans, change tables, and perform other menu options. The name of your plan is displayed in a tab directly above the toolbar.
- The workspace enables you to view and prepare data.
- The left pane enables you to add transforms and view properties for the source table.
- The right pane enables you to view plan actions and view properties for the result table.
- The bottom pane enables you to view the details about a table, including table data, profiles, and metrics.

About the Toolbar

The following icons are available to help you prepare data:

Icon	Description
↺	enables you to undo a change to the plan.
⋮	displays a drop-down menu that enables you to save a table and save a plan.

About the Left Pane

Icon	Description
	enables you to add transforms.
	enables you to view properties for the source table.

About the Right Pane

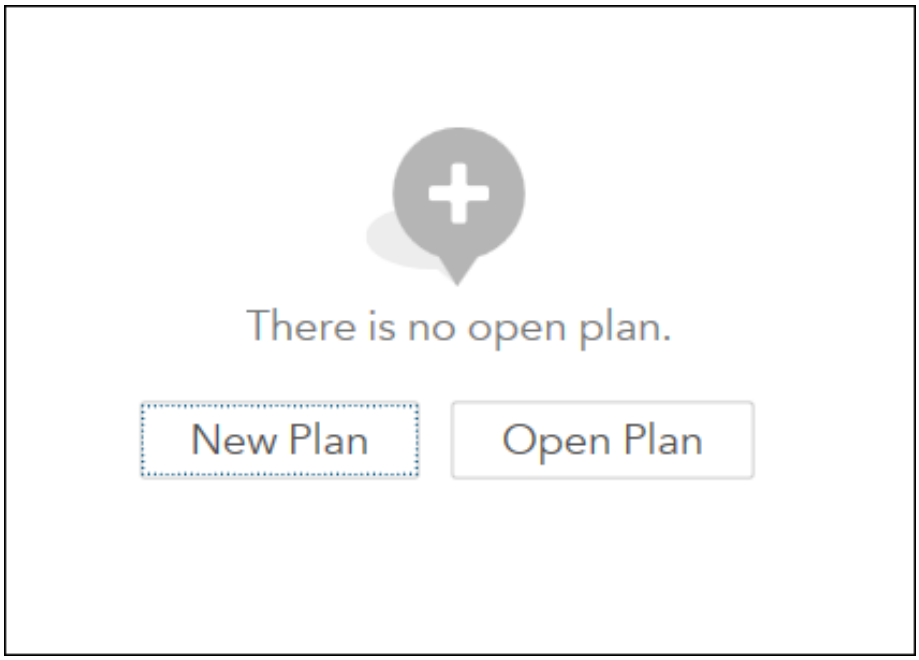
Icon	Description
	enables you to view plan actions.
	enables you to view properties for the result table.

Getting Started with SAS Data Studio

Opening a Data Source

To open a data source in SAS Data Studio:

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 "Getting Started with the Choose Data Window" 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.

Viewing Profiles

To view profile information, open a data source, click the **Profile** tab in the bottom pane, and then click **Run Profile**. If you have SAS Data Preparation licensed at your site, then you can view advanced column metrics, including pattern and frequency distributions. To view column metrics, click the **Profile** tab in the bottom pane, and then click the name of the column that you want to view.

Note: Clicking the **Run Profile** button refreshes 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 .

Viewing Table Metadata

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

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

Viewing 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.
- 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 will always be the same. This occurs because when you modify a table, a new target table is created.
- Encoding type

Viewing Plan Actions

To view a list of actions taken on a plan, click in the right pane.

Creating Jobs for Scheduling

To create a job:

1 Open a data source, and then click ⋮ in the toolbar.
2 Click **Create job**.

> **TIP** After you create a job, use SAS Job Monitor to schedule the job to run at intervals that you specify.

You can change the expiration time for a job using the **interactiveJobExpiresAfter** and **saveTableJobExpiresAfter** configuration properties in SAS Environment Manager. For more information, see "SAS Data Studio" in SAS Viya Administration: Configuration Properties .

Working with Columns

Changing 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 11.

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

1. Open a data source, and then click in the left pane.
2. Click **Change case** in the transforms list, and then click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu.
4. Select **Uppercase** or **Lowercase** from the **Case** drop-down menu.
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**.

Changing 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.

To change the data type for a column:

1. Open a data source, and then click in the left pane.
2. Click **Convert column** in the transforms list, and then click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu.
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**.

Changing the Data Format for a Column

To change the data format for a column:

1. Open a data source, and then click in the left pane.
2. Click **Convert column** in the transforms list, and then click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu.
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.
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**.
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.
9. Click **Run**.

Removing White Space in Columns

To remove white space in a column:

1. Open a data source, and then click in the left pane.
2. Click **Trim whitespace** in the transforms list, and then click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu.
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.

     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.
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.
6. Click **Run**.

Removing Columns

To remove a column, click **Remove** in the transforms list, and then click **Add Transform**. Select a column from the **Source column** drop-down menu, and then click **Run**.

You can remove multiple columns at the same time by clicking the plus sign, and selecting an additional column from the **Source column** drop-down menu.

Renaming Column Headings

To rename a column heading, click **Rename** in the transforms list, and then click **Add Transform**. 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.

Renaming a column heading does not change the label for the column. To change the column label, use the **Convert column** transform.

Splitting Columns

About Splitting Columns

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. The results of choosing the **On the delimiter** option include a new column that contains all of the characters to the left of the delimiter, and 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. The results of choosing the **Before a delimiter** option include a new column that contains all of the characters to the left of the delimiter, and 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. The results of choosing the **After a delimiter** option include a new column that contains all of the characters to the right of the delimiter, and 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.

Split a Column

To split a column:

1. Open a data source, and then click in the left pane.
2. Click **Split** in the transforms list, and then click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu.
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.

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).

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.

Creating Calculated Columns

To create a calculated column:

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

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

3 In the Calculated Column window, enter a DATA step expression in the **Expression** field. 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.

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.

5 Click **Run**.

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

Creating Custom Code

About Custom Code

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 will fail due to name changes. For more information about the variables that are available, see Step 4 on page 9.

Create Custom Code

To create custom code:

1. Open a data source, and then click in the left pane.
2. Click **Code** in the transforms list, and then click **Add Transform**.
3. Select the code language from the drop-down menu. The following code languages are available: **CASL** and **DATA step**.

   For more information about CASL, see SAS® Cloud Analytic Services 3.3: 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 will 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 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. 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

Click **Code** in the transforms list, and then click **Add Transform**.

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

Enter the following code in the text box:

```
data {{_dp_outputTable}} (caslib={{_dp_outputCaslib}}); set {{_dp_inputTable}} (caslib={{_dp_outputCaslib}});
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 with SAS Data Preparation only. If SAS Data Preparation is licensed at your site, then you can access the data quality transforms.

The data quality transforms use SAS Quality Knowledge Base (QKB), which 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.

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 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.

Casing

Sometimes it can be desirable to change the case of a text string without making more complex changes, such as those involved in a standardization operation. Casing provides a way to change a text string to uppercase, lowercase, or proper case, without altering word order or punctuation. Proper casing applies locale-specific rules. The rules preserve common capitalization, as used in acronyms and names. Example inputs and outputs of Casing are as follows:

Input	Output
mcdonald, ronald	McDonald, Ronald
SAS INSTITUTE	SAS Institute

To change case using the **Casing** transform:

1. Open a data source, and then click in the left pane.
2. Select **Casing** in the transforms list, and then click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu.
4. Select a locale from the **Locale** drop-down menu.
5. Select a definition from the **Casing** drop-down menu.
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.

8 Click **Run**.

Parsing

It is often valuable to break up a string into its constituent semantic components, and then analyze the individual components. For example, you can analyze customer addresses using only the cities and postal codes. You can parse your address data as follows:

Input	Output	
100 Main St, Cary, NC 27513	Street	100 Main St
	City	Cary
	State	NC
	Postal Code	27513

To parse data:

1 Open a data source, and then click in the left pane.
2 Select **Parsing** in the transforms list, and then click **Add Transform**.
3 Select a source column from the **Source column** drop-down menu.
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.
8 Click **Run**.

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

To perform field extraction:

1 Open a data source, and then click in the left pane.
2 Select **Field extraction** in the transforms list, and then click **Add Transform**.
3 Select a source column from the **Source column** drop-down menu.
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.
8 Click **Run**.

Gender Analysis

For some applications, you might want to know the gender associated with the name of an individual. For example, gender information can be useful in marketing analyses. Here are example inputs and outputs of Gender Analysis, where U indicates unknown:

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

To perform gender analysis:

1. Open a data source, and then click in the left pane.
2. Select **Gender analysis** in the transforms list, and then click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu.
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.
8. Click **Run**.

Identification Analysis

In order to take advantage of the value of your data, you need to know what type of data you have. Identification Analysis reads text values and determines the semantic type of those values. For example, some inputs and outputs of identification analysis might be as follows:

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

To perform identification analysis:

1. Open a data source, and then click in the left pane.
2. Select **Identification analysis** in the transforms list, and then click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu.
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.
8. Click **Run**.

Match Codes

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 are said to match. For example, the following records constitute a match:

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

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 9.

To use match codes:

1. Open a data source, and then click in the left pane.
2. Select **Match codes** in the transforms list, and then click **Add Transform**.
3. Select a source column from the **Source column** drop-down menu.
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.
9. Click **Run**.

Standardize

Standardization updates source strings so that they conform to a preferred, expected format. Standardization is a way to eliminate noise and get more accurate results from frequency counts and other more advanced analytics. Standardization also increases the readability of data in reports by rendering data in easy-to-read formats. Here are example inputs and outputs of a Standardization operation:

Input	Output
n.c.	NC
N Carolina	NC
North carolina	NC

To standardize data:

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

2 Select **Standardize** in the transforms list, and then click **Add Transform**.
3 Select a source column from the **Source column** drop-down menu.
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.
8 Click **Run**.

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.

To append data to a table:
1 Open a data source, and then click in the left pane. This table is used as the base table.
2 Select **Append** in the transforms list, and 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**.

Joining Tables

To join tables:
1 Open a data source, and then click in the left pane.
2 Select **Join** in the transforms list, and then click **Add Transform**. This table is used as the base table.

   CAUTION! It is not recommended that you join tables that contain more than 250 columns.
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**.

You can modify the steps that you took to join the tables by clicking **Edit 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 data source, and then click in the left pane.
2 Select **Filter** in the transforms list, and 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 17.
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 (/), for 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 (/), for 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

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

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

2 Click **Transpose** in the transforms list, and 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 will become the new column headings. The column headings of the columns that are transposed will be 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.
5 (Optional) On the **Group By Columns** tab, specify the columns by which the rows of the newly transposed columns will be 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

Click **Transpose** in the transforms list, and then click **Add Transform**. 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.

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, 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:

- Opening a table automatically adds the table to the current plan or creates a new plan if you do not have one open.
- 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 Tables” on page 22.
- You can add multiple transforms to a plan. To add multiple transforms, you must run the plan after you make changes to the current transform before you can add another one.
- 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.

Work with Plans

You can perform the following tasks with plans:

Task	Steps	Additional Information
Create a new plan	Click the **New Plan** button in the workspace, and open a table.	
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, but 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.	
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**.	
Close a plan	Click ✕ on the tab for the plan that you want to close.	
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**.	

Saving Tables

About 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.
- Tables are always saved in SASHDAT format.
- When you save a table that was loaded in a format other than SASHDAT, the file extension will change to `.sashdat`, even if you save the table using the original table name and file extension.
- You cannot save a table in a format that would enable you to share the table outside of the SAS Viya environment (for example, in the CSV or XLSX format).
- Table names cannot exceed 247 characters.

Save a Table

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

1. Click ⋮ .
2. Click **Save as**.
3. Specify information about the plan, including the folder where you want to save it and the plan name.
4. Make sure that the **Save table** check box is selected.
5. 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.
6. (Optional) Review the library in the **Library** field, and make changes if necessary.
7. Click **Save**.

If the table that you specified already exists, a window appears, asking if you want to overwrite the table. To overwrite the table, click **Yes**. Otherwise, click **No**, and specify a different table name.

Modify 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. To modify settings:

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

- Click **General** to select a default locale for SAS Data Quality transforms. The SAS Data Quality transforms are available with SAS Data Preparation.
- Click **Geographic Mapping** to accept the terms and conditions for Esri ArcGIS Online Services.

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

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