SAS® Data Studio 2.4: 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.

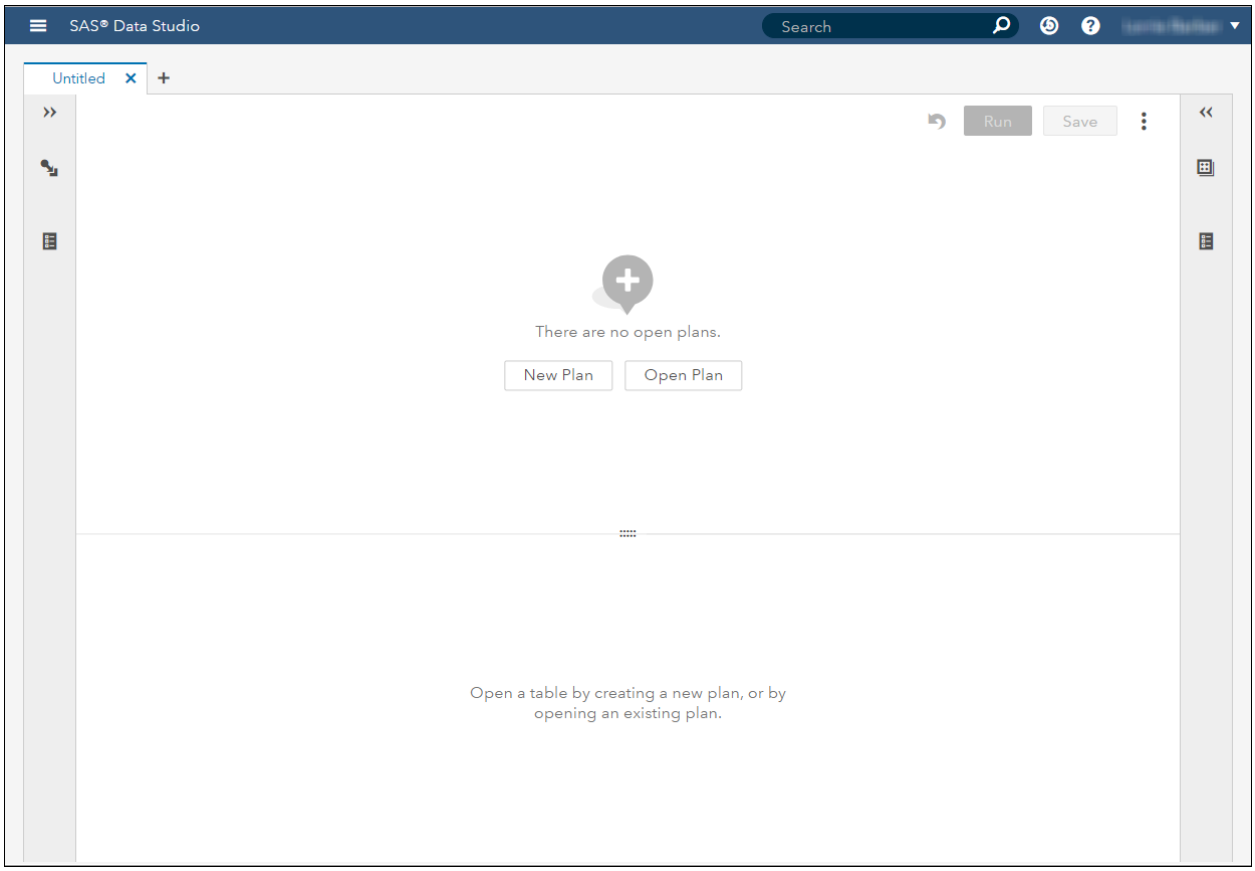
Create plans
: You can create a plan, which is a collection of actions or data transforms performed on a table.

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 "Modifying SAS Data Studio Settings" on page 35. For more information about search and global settings, see *SAS Viya Web Applications: General Usage Help*.
- 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 Left Pane

Table 1 *Left Pane Controls*

Icon	Description
	enables you to add transforms.

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

About the Right Pane

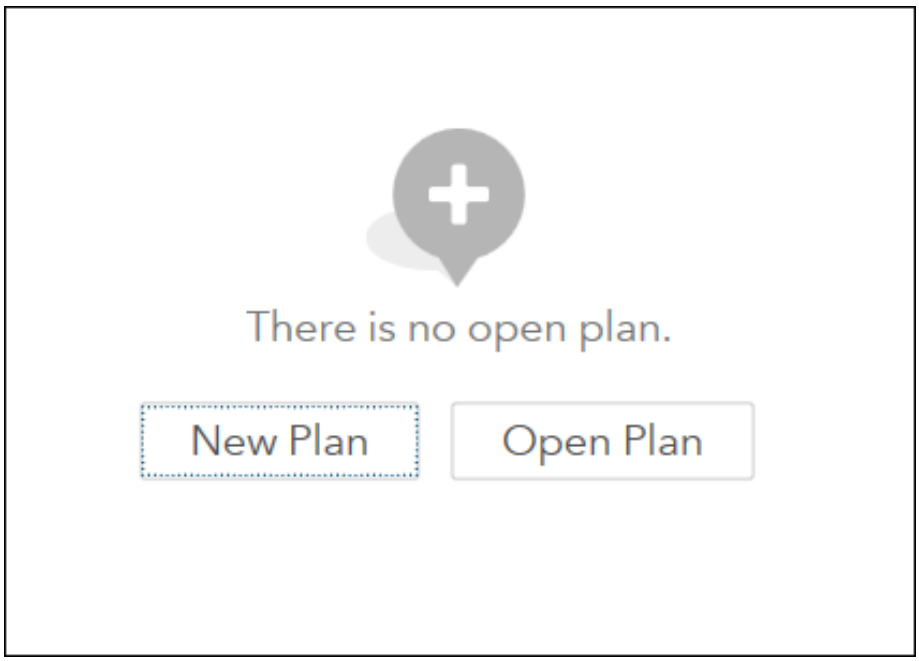
Table 2 Right Pane Controls

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

Getting Started with SAS Data Studio

Open a Data Source

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.

Run and View Profiles

1 Open a data source.

2 Click the **Profile** tab in the bottom pane.

3 Click **Run Profile**.

4 Click the **Profile** tab in the bottom pane, 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** 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.

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

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.

View Plan Actions

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

Working with Columns

Access Columns

The fields that you use to select the columns for transforms in SAS Data Studio support all of the columns included in the tables.

1. Click ▾ in column field. Up to 20 columns in the table are displayed.
2. Click **View all columns** at the bottom of the columns list to display the Select a Source Column window. This link is available only when the table contains more than 20 columns.
3. Search the list of available columns and select the one you need.
4. Click **OK** to close the window, and enter the column into the field.

Note: You use the Manage columns window to hide and display columns in the **Table** tab. Click ⊞▾ at the top right corner of the table to access 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 13.

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 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.

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**.

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`. 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.
9. Click **Run**.

Remove White Space in Columns

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**.

Remove Columns

1. To remove a column, click **Remove** in the transforms list, and then click **Add Transform**.
2. Select a column from the **Source column** drop-down menu, and then click **Run**.
3. (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. To rename a column heading, click **Rename** in the transforms list, and then click **Add Transform**.
2. 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.
3. (Optional) 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

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

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. 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.

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. 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

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 11.

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 3.4: 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 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.

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

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 only displayed and available in SAS Data Studio 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 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 data source, and then click in the left pane.
3. Select **Casing** in the transforms list, and then click **Add Transform**.
4. Select a source column from the **Source column** drop-down menu.
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.
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 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**.

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 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**.

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 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. 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.
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 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**.

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$$HHIO$$

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 11.

1. Open a data source, and then click in the left pane.
2. Select **Matchcodes** 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 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 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**.

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 IDs.

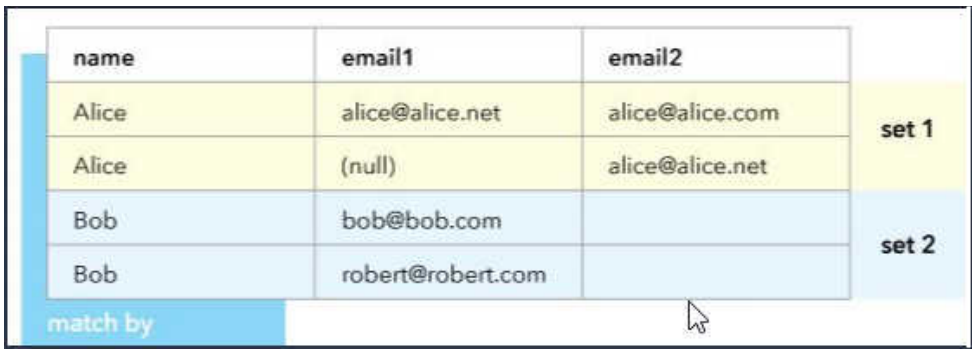
1. Open a data source, and then click in the left pane.
2. Select **Match and cluster** in the transforms list, and then click **Add Transform**.
3. Keep the default value in the **New column name** field, or enter an appropriate column name. The **Match and cluster** transform creates this new column 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, changes that you make to the **Length** or the **Format** fields are not saved when you close the Options for new columns window.
5. Create one or more match rules. For more information, see "Creating Match Rules" on page 21.
6. (Optional) Set advanced options for the **Match and cluster** transform. For more information, see "Setting Advanced Options" on page 22.

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 1 *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 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**.

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 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! 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 24 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.

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.

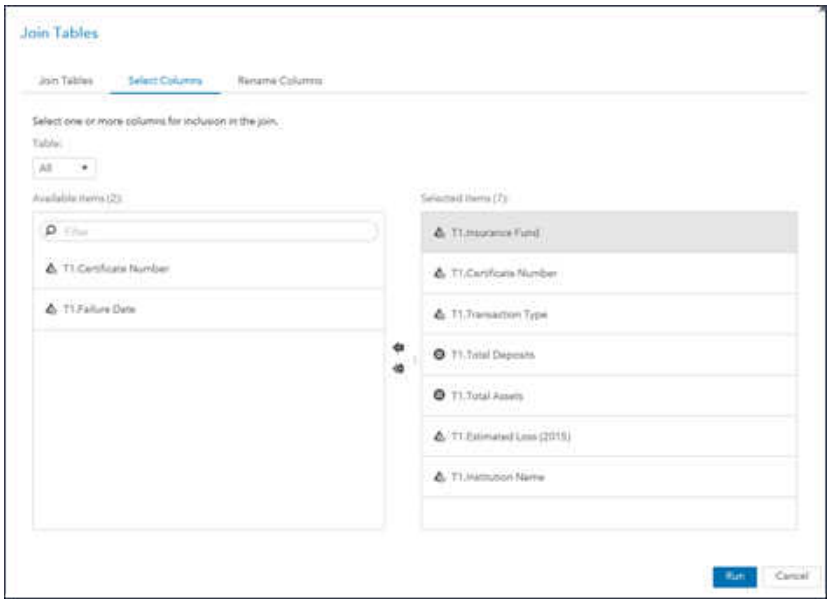
7 Click **Run**.

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

Select or Rename Join Columns

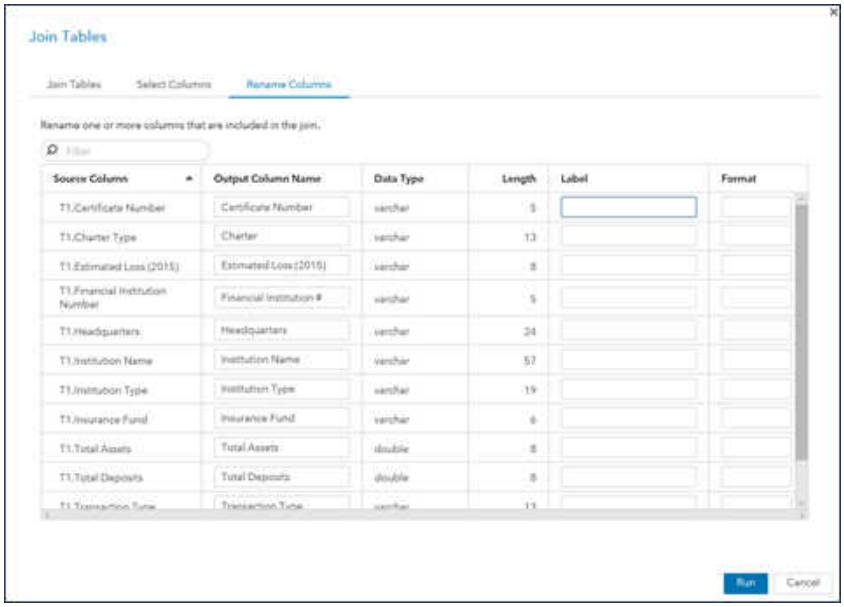
You can use **Select Columns** and **Rename Columns** tabs when you need to select or rename the columns included in the output of your join. Selecting columns enables you to base your join on a subset of the columns available in the input tables for the join. Renaming columns enables you to change the names, labels, and formats of the columns to meet your needs.

Figure 2 *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 3 *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. 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 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 26.
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

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.

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.

1. Open a data source, and then click in the left pane.
2. Select **Analytic Partitioning** in the transforms list, and then click **Add Transform**.
3. Review the default value in the **New column name** field. Change as needed.
4. Move up to four columns from the **Available columns** pane to the **Selected columns** pane. These columns are used as stratification values.
5. Review the default values in the **Training**, **Validation**, and **Test** fields. Change as needed.
6. Click **Run**.
7. (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 data source, and then click in the left pane.
2. Select **Unique identifier** in the transforms list, and 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. 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`.

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 added 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 33.
- 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:

Table 3 Plan Tasks

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. 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**.	
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: Content Management*.
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**.	

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.
- Tables are always saved in the original source format unless you save them into a library with a different format. You can choose any library that is available on the CAS server.
- Table names cannot exceed 247 characters.

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.

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.

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 data source.

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: How To" in *SAS Viya Administration: Jobs*.

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 will fail. 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.