

How to Easily Create an Empty SAS Dataset

Authored by
stats writer

November 22, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Create an Empty SAS Dataset*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=99697>

Creating an empty **dataset** in **SAS** is a fundamental step in data preparation, often used when the structure (schema) needs to be defined before data input begins. This process is surprisingly straightforward, primarily relying on the **DATA step** and specific control statements.

An empty dataset serves as a template, establishing essential **metadata** like variable names, data types, lengths, and formats. Once defined, this shell can be populated with observations later, ensuring all incoming data adheres to the required structure. We will explore two reliable methods for accomplishing this task, both resulting in a dataset that possesses attributes but contains zero records.

Two Primary Methods for Empty Dataset Creation

There are two common and highly effective methods for generating an empty dataset structure within a **SAS** library. The choice between them depends entirely on whether you are designing a structure from scratch or mirroring an existing schema.

Method 1: Creating from Scratch (Defining the Schema): This technique involves using the **DATA** and **ATTRIB** statements to manually define every variable attribute, which is ideal for building entirely new data structures where no template exists.

Method 2: Creating from an Existing Dataset (Copying Metadata): This approach uses the **SET** statement combined with the **STOP** statement to clone the structure (but not the data) of an already existing dataset, providing a rapid way to replicate complex schemas.

Method 1: Create Empty Dataset from Scratch

The first approach allows for granular control over every aspect of the dataset's design. This technique utilizes the **ATTRIB** statement to specify critical properties for each variable, such as **LENGTH**, **FORMAT**, and **LABEL**. Crucially, the **STOP** statement is employed to prevent the **DATA** step from processing any observations, ensuring the resulting dataset remains empty.

```
data empty_data;
attrib
var1 length=8 format=best12. label="var1"
var2 length=$30 format=$30. label="var2"
var3 length=8 format=best12. label="var3"
stop;
run;
```

Method 2: Create Empty Dataset from Existing Dataset

When the desired structure already exists in another dataset, the fastest way to create an empty shell is using the `SET` statement. This method copies all variable definitions, including attributes like formats and labels, from the source dataset into the new one, bypassing the need for extensive `ATTRIB` statements.

```
data empty_data;  
set existing_data;  
stop;  
run;
```

Understanding the Critical Role of the STOP Statement

In both creation methods, the inclusion of the `STOP` statement is paramount to achieving a truly empty dataset. When **SAS** executes a **DATA step**, it typically loops through the input data until the end-of-file is reached, creating an observation during each iteration. The `STOP` statement immediately halts the execution of the **DATA step** before the first observation is written to the output dataset.

If the `STOP` statement were omitted in Method 1 (Creating from Scratch), the **DATA step** would execute once, creating a single observation containing missing values for all defined variables, thus resulting in a dataset that is technically not empty. In Method 2, omitting `STOP` would cause the entire source dataset to be copied, defeating the purpose of creating an empty schema.

The result of utilizing the `STOP` statement is a dataset object containing defined variable names and **metadata**, but with a record count of zero observations. The following practical examples demonstrate the application and verification of both methods.

Example 1: Defining a New Dataset Schema from Scratch

This example demonstrates Method 1 by defining a new dataset named `empty_data`. We specify four variables relevant to employee sales tracking, ensuring we establish the correct data type (numeric or character) and presentation (**format**) for each column using the `ATTRIB` statement.

Notice the detailed assignment of attributes:

`employee_ID` and `sales` are defined as `LENGTH=8` (standard numeric).

`employee_Name` is defined as `LENGTH=$30` (a character variable of length 30).

A specific date **format** (`date9.`) is applied to `sales_date`.

```
/*create empty dataset*/  
data empty_data;
```

```

attrib
employee_ID length=8 format=best12. label="Employee ID"
employee_Name length=$30 format=$30. label="Employee Name"
sales length=8 format=best12. label="Sales"
sales_date length=8 format=date9. label="Sales Date";
stop;
run;

```

Verifying the Empty Dataset Structure

To confirm that the dataset was created successfully with the required **metadata** but without any records, we use the **PROC CONTENTS** procedure. This procedure reads the descriptor portion of the dataset, providing key information about its structure and attributes.

```

/*view contents of dataset*/
proc contents data=empty_data;

```

Data Set Name	WORK.EMPTY_DATA	Observations	0
Member Type	DATA	Variables	4
Engine	V9	Indexes	0
Created	11/17/2022 10:30:43	Observation Length	56
Last Modified	11/17/2022 10:30:43	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

The resulting output clearly shows that the dataset possesses the necessary structural elements (four variables defined), but the crucial metric, **Observations**, is listed as zero. This confirms the dataset is truly empty of data rows.

Furthermore, reviewing the variable listing confirms the successful application of the **ATTRIB** statement, detailing the variable name, type, length, and assigned label for all columns:

Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Label
1	employee_ID	Num	8	BEST12.	Employee ID
2	employee_Name	Char	30	\$30.	Employee Name
3	sales	Num	8	BEST12.	Sales
4	sales_date	Num	8	DATE9.	Sales Date

Example 2: Leveraging Existing Structure for Metadata Cloning

Method 2 is often preferred for efficiency, especially when dealing with complex data structures or when ensuring compatibility with an existing system. Here, we create an empty dataset named `empty_dat` by referencing the structure of `sashelp.Comet`, which is a predefined dataset built into the **SAS** system library.

The `SET sashelp.Comet;` line instructs the **DATA step** to inherit all structural properties of the source dataset. The subsequent `STOP;` statement prevents the reading of any actual data records, ensuring only the structure is copied.

```
/*create empty dataset from existing dataset*/  
data empty_dat;  
set sashelp.Comet;  
stop;  
run;
```

Validating the Copied Structure

Just as in the previous example, we use `PROC CONTENTS` to inspect the newly created dataset `empty_dat` and verify that the structural copy was successful without including any observations.

```
/*view contents of dataset*/  
proc contents data=empty_data;
```

The CONTENTS Procedure

Data Set Name	WORK.EMPTY_DAT	Observations	0
Member Type	DATA	Variables	4
Engine	V9	Indexes	0
Created	11/17/2022 10:41:24	Observation Length	32
Last Modified	11/17/2022 10:41:24	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

The output confirms that `empty_dat` successfully adopted the schema from `sasHELP.Comet`, listing all its variables. Crucially, the observation count remains at zero, confirming the successful use of the `SET` and `STOP` combination for metadata cloning.

A closer look at the variables section shows the inherited variable names and their predefined attributes, ready for data insertion:

Alphabetic List of Variables and Attributes				
#	Variable	Type	Len	Label
1	Dose	Num	8	1,2 Dimethylhydrazine dihydrochloride Dose Level
4	Length	Num	8	Tail Length of the Comet
2	Rat	Num	8	Rat Index
3	Sample	Num	8	Slide Index of Grouped Cells from a Rat

Conclusion and Next Steps

Mastering the creation of empty datasets is a crucial skill for any **SAS** programmer, enabling robust data management and streamlined ETL processes. Whether you manually define the structure using `ATTRIB` or leverage existing definitions using `SET`, the key is the efficient use of the `STOP` statement to preserve the schema while ensuring zero observations.

Once the empty structure is established, the next logical step involves techniques for population, appending data, or merging multiple sources into this new template. Understanding how to manage the dataset **metadata** is foundational for more advanced data manipulation tasks.

The following tutorials explain how to perform other common tasks in **SAS**:

ARABPSYCHOLOGY.COM