



Summary of Lesson 3: Accessing Data

This summary contains topic summaries, syntax, and sample programs.

Topic Summaries

To go to the movie where you learned a task or concept, select a link.

Accessing SAS Libraries

SAS data sets are stored in SAS libraries. A [SAS library](#) is a collection of one or more SAS files that are recognized by SAS. SAS automatically provides one temporary and at least one permanent SAS library in every SAS session.

Work is a temporary library that is used to store and access SAS data sets for the duration of the session. **Sasuser** and **sashelp** are permanent libraries that are available in every SAS session.

You refer to a SAS library by a library reference name, or [libref](#). A libref is a shortcut to the physical location of the SAS files.

All SAS data sets have a [two-level name](#) that consists of the libref and the data set name, separated by a period. Data sets in the **work** library can be referenced with a one-level name, consisting of only the data set name, because **work** is the default library. Data sets in permanent libraries must be referenced with a two-level name.

You can create and access your own [SAS libraries](#). User-defined libraries are permanent but are not automatically available in a SAS session. You must assign a libref to a user-created library to make it available. You use a [LIBNAME statement](#) to associate the libref with the physical location of the library, that is, the physical location of your data. You can submit the LIBNAME statement alone at the start of a SAS session, or you can store it in a SAS program so that the SAS library is defined each time the program runs. If your program needs to reference data sets in multiple locations, you can use multiple LIBNAME statements.

```
LIBNAME libref 'SAS-library' <options>;
```

Use [PROC CONTENTS](#) with `libref._ALL_` to display the contents of a SAS library. The report will list all the SAS files contained in the library, as well as the descriptor portion of each data set in the library. Use the NODS option in the PROC CONTENTS statement to suppress the descriptor information for each data set.

```
PROC CONTENTS DATA=libref._ALL_ NODS;  
RUN;
```

After associating a libref with a permanent library, you can write a [PROC PRINT](#) step to display a SAS data set within the library.

```
PROC PRINT DATA=libref.SAS-data-set;  
RUN;
```

In an interactive SAS session, a libref remains in effect until you cancel it, change it, or end your SAS session. To [cancel a libref](#), you submit a LIBNAME statement with the CLEAR option. This clears or disassociates a libref that was previously assigned. To specify a different physical location, you submit a LIBNAME statement with the same libref name but with a different filepath.

```
LIBNAME libref CLEAR;
```

When a SAS session ends, everything in the **work** library is deleted. The librefs are also deleted. Remember that the contents of permanent libraries still exist in the operating environment, but each time

you start a new SAS session, you must resubmit the LIBNAME statement to redefine a libref for each user-created library that you want to access.

Examining SAS Data Sets

[SAS data sets](#) are specially structured data files that SAS creates and that only SAS can read. A SAS data set is displayed as a table composed of variables and observations. A SAS data set contains a descriptor portion and a data portion.

The descriptor portion contains general information about the data set (such as the data set name and the number of observations) and information about the variable attributes (such as name, type, and length). There are two types of variables: character and numeric. A character variable can store any value and can be up to 32,767 characters long. Numeric variables store numeric values in floating point or binary representation in 8 bytes of storage by default. Other attributes include formats, informats, and labels. You can use PROC CONTENTS to browse the descriptor portion of a data set.

```
PROC CONTENTS DATA=libref.SAS-data-set;  
RUN;
```

The data portion contains the data values. Data values are either character or numeric. A valid value must exist for every variable in every observation in a SAS data set. A [missing value](#) is a valid value in SAS. A missing character value is displayed as a blank, and a missing numeric value is displayed as a period. You can specify an alternate character to print for missing numeric values using the MISSING= SAS system option. You can use PROC PRINT to display the data portion of a SAS data set.

[SAS variable and data set names](#) must be 1 to 32 characters in length and start with a letter or underscore, followed by letters, underscores, and numbers. Variable names are not case sensitive.

Sample Programs

Accessing a SAS Library

```
/*Replace filepath with the physical location of your practice files.*/  
%let path=filepath;  
libname orion "&path";
```

Browsing a Library

```
proc contents data=orion._all_;  
run;  
  
proc contents data=orion._all_ nodes;  
run;
```

Viewing a Data Set with PROC PRINT

```
proc print data=orion.country;  
run;
```

Viewing the Descriptor Portion of a Data Set

```
proc contents data=orion.sales;  
run;
```

Viewing the Data Portion of a SAS Data Set

```
proc print data=orion.sales;  
run;
```

Close