

Dictionary Tables And SASHELP Views: An Introductory Tutorial

Nancy J. Michal, ASG, Inc.

This tutorial gives an introduction to dictionary tables and SASHELP views which are new features of version 6.07 of the SAS system. Dictionary tables are read-only SQL tables, that are generated at run time. They contain information about the SAS data libraries, SAS data sets, SAS catalogs, SAS system options and external files associated with a SAS session. In addition, there are SQL views of the dictionary tables that are stored in the SASHELP library. The purpose of this paper is to describe the information contained in the dictionary tables and SASHELP views, discuss how it can be accessed, and give examples of how it can be utilized in various applications.

INTRODUCTION

There are 8 dictionary tables and 14 SASHELP views that are automatically part of the SAS environment at the beginning of a SAS session (Figure 1). These tables contain information about the SAS libraries and external files that are automatically defined at the beginning of a session (i.e. SASUSER, SASHELP, WORK), as well as any SAS libraries and external files that are defined

through the AUTOEXEC.SAS file. These tables are automatically updated every time there is a change in the SAS environment. If a SAS option is modified, a LIBNAME or FILENAME is assigned or deassigned, or a SAS file is created, updated, or deleted, the information in the dictionary tables will be immediately updated. Therefore, these tables always reflect the status of the SAS environment.

The Dictionary Tables are read-only objects and PROC SQL must be used to access the information that is contained in these tables. However, data in the SASHELP views can be accessed using PROC SQL, the data step, or a SAS procedure. The DESCRIBE statement can be used in the PROC SQL procedure to show how each dictionary table is defined. Similarly, PROC CONTENTS can be used on the SASHELP views.

```
proc sql;
  describe table dictionary.columns;

proc contents
  data=sashelp.vcolumn;
```

DESCRIPTION OF TABLES

The DICTIONARY.CATALOGS table contains information about the SAS catalogs in every SAS library that has been defined in the SAS session. There is one row in the table (or observation in the SASHELP view) for each entry in each of the catalogs. Figure 2 lists out the columns (variables) in this table. This data could be useful to a developer who may want to query this table by modification date or object type. This information is similar to the data that is generated from the CONTENTS options of PROC CATALOG. However, unlike the dictionary tables, the CATALOG procedure processes only one catalog at a time, and the results are not automatically kept current.

Figure 1
Dictionary Tables and SASHELP Views

Dictionary Tables	SASHELP Views
·CATALOGS	·VCATALOG
·COLUMNS	·VCOLUMN
·EXTFILES	·VEXTFL
·INDEXES	·VINDEK
·MEMBERS	·VMEMBER
·OPTIONS	·VOPTION
·TABLES	·VTABLE
·VIEWS	·VVIEW
	·VSLIB
	·VSTABLE
	·VSCATLG
	·VSACCES
	·VSVIEW
	·VSTABVW

Figure 2 DICTONARY.CATALOGS

libname	char(8)	label='Library Name'
memname	char(8)	label='Member Name'
memtype	char(8)	label='Member Type'
objname	char(8)	label='Object Name'
objtype	char(8)	label='Object Type'
objdesc	char(40)	label='Object Description'
modified	char(8)	label='Date Modified'
alias	char(8)	label='Object Alias'

The DICTONARY.COLUMNS table includes information about the variables and their attributes in the SAS data sets in the SAS libraries that have been defined. There is one row in the table for each variable in each of the data sets. Figure 3 lists out the columns in this table. Some of the same information that is generated from the OUT= option of PROC CONTENTS is included in this table. However, the CONTENTS procedure processes only 1 data library at a time.

Figure 3 DICTONARY.COLUMNS

libname	char(8)	label='Library Name'
memname	char(8)	label='Member Name'
memtype	char(8)	label='Member Type'
name	char(8)	label='Column Name'
type	char(4)	label='Column Type'
length	num	label='Column Length'
npos	num	label='Column Position'
varnum	num	label='Column Number in Table'
label	char(40)	label='Column Label'
format	char(16)	label='Column Format'
informat	char(16)	label='Column Informat'
idxusage	char(9)	label='Column Index Type'

The DICTONARY.EXTFILES table contains information about all external files, which includes all files that are not SAS files. There is one entry for each external file in each of the SAS libraries that have been defined and any files in the default directory where the SAS session started. The SAS system will automatically assign a sequential file reference for any file that has not been defined in a filename statement, or been given a logical name definition. Figure 4 lists out the information contained in the DICTONARY.EXTFILES table.

Figure 4 DICTONARY.EXTFILES

fileref	char(8)	label='Fileref'
xpath	char(80)	label='Path Name'
xengine	char(8)	label='Engine Name'

The DICTONARY.INDEXES table contains information about the indexes that are used in the SAS data files in each defined SAS data library. There is one row for each SAS variable that is used as an index in any SAS data file. Figure 5 lists out the information in DICTONARY.INDEXES. This information would otherwise have to be obtained from running a PROC CONTENTS on the data.

Figure 5 DICTONARY.INDEXES

libname	char(8)	label='Library Name'
memname	char(8)	label='Member Name'
memtype	char(8)	label='Member Type'
name	char(8)	label='Column Name'
idxusage	char(9)	label='Column Index Type'
inxname	char(8)	label='Index Name'
indxpos	num	label='Position of Column in Concatenated Key'
nomiss	char(3)	label='Nomiss Options'
unique	char(3)	label='Unique Option'

The DICTONARY.MEMBERS table provides information about the SAS files that are in the SAS data libraries. There is one entry in this table for each SAS file. SAS files include SAS data sets (data files or views), as well as access descriptors, catalogs, and stored programs. Figure 6 lists out the information contained in this table. It is important to note that the path name of a user's SAS WORK directory is given for any files stored in the WORK area. This information may be useful to have in certain applications in which users output files to the WORK area.

Figure 6 DICTONARY.MEMBERS

libname	char(8)	label='Library Name'
memname	char(8)	label='Member Name'
memtype	char(8)	label='Member Type'
engine	char(8)	label='Engine Name'
index	char(8)	label='Indexes'
path	char(80)	label='Path Name'

The DICTONARY.OPTIONS table contains data about the current option settings in a SAS session. There is one row for every option in the SAS session. Information included in this table is option name, option setting, and option description (Figure 7). The only other way this information can be obtained is by running a PROC OPTIONS. It is useful to review the data in this table in order to understand more about the SAS system options that are available.

Figure 7 DICTONARY.OPTIONS

optname	char(16)	label='Session Option Name'
setting	char(200)	label='Session Option Setting'
optdesc	char(80)	label='Option Description'

The DICTONARY.TABLES table provides information about the SAS data files in the SAS data libraries. There is one entry for each data file in each of the SAS data libraries. This would also include SQL tables, since they are considered SAS data files. Figure 8 lists out the columns contained in this table.

Figure 8 DICTONARY.TABLES

libname	char(8)	label='Library Name'
memname	char(8)	label='Member Name'
memtype	char(8)	label='Member Type'
memlabel	char(40)	label='Dataset Label'
typemem	char(8)	label='Dataset Type'
crdate	num	label='Date Created'
modate	num	label='Date Modified'
nobs	num	label='Number of Observations'
obslen	num	label='Observation Length'
nvar	num	label='Number of Variables'
protect	char(3)	label='Type of Password Protection'
compress	char(8)	label='Compression Routine'
reuse	char(8)	label='Reuse Space'
busize	num	label='Bufsize'
delobs	num	label='Number of Deleted Observations'
indxtype	char(9)	label='Type of Indexes'

The DICTONARY.VIEWS table provides information about the SAS data views in the current SAS session. There is one entry for each view that is defined in the SAS data libraries. The columns in this table are listed out in Figure 9. SAS data views can include PROC SQL views, Data Set views, and SAS/ACCESS view descriptors. This table is especially informative, since the engine name is given for each view. This is useful since the type of view can not be determined by looking at the file extension under VMS.

Figure 9 DICTONARY.VIEWS

libname	char(8)	label='Library Name'
memname	char(8)	label='Member Name'
memtype	char(8)	label='Member Type'
engine	char(8)	label='Engine Name'

Figure 10 is a diagram of the files that can be present during a SAS session, along with the dictionary table that describes the files. In contrast to the data output from PROC CONTENTS and PROC CATALOG, the data in the dictionary tables is normalized. Since there is referential integrity in this data, these tables form a relational database of meta-data.

Figure 10 Overview of Files in a SAS Session

SAS Session (Options)**SAS data library**•External files (*Extfiles*)•SAS files (*Members*)

<SAS Data Sets (Columns)	•Data (Table, Indexes) •View (Views)
-----------------------------	---

<Other SAS files	•Access •Catalog (<i>Catalogs</i>) •Program
------------------	---

SASHELP VIEWS

There are 14 PROC SQL views in the SASHELP library. Information can be retrieved from these views using PROC SQL, a Data Step, or a SAS Procedure. Eight of these views are copies of the dictionary tables. The remaining 6 views are subsets of the Dictionary.Members table. Figure 11 lists the names of the views that are based on the Dictionary.Members table along with the types of data included in the view.

Figure 11
SASHELP Views of DICTONARY.MEMBERS

View Name	Includes information on:
•VSLIB	SAS libraries
•VSTABLE	SAS data files
•VSCATG	SAS catalogs
•VSACCESS	Access Descriptors
•VSVIEWS	Views
•VSTABLW	Sas Data Sets

APPLICATIONS

Dictionary tables and SASHELP views can be used during both batch and interactive SAS sessions. When retrieving information from a dictionary table, column names can be used in a select statement and row values can be used in a where statement to extract a subset of the information. The code below illustrates how a data set label can be extracted from the dictionary tables and then used in the title for a PROC PRINT of the data set.

```
%macro printit(libname,memname);
  proc sql;
    select memlabel into :label
    from dictionary.tables
    where libname="%&libname" and
           memname="%&memname";
  quit;

  proc print data=%&libname..%&memname;
    title "%&label";
  run;
%mend printit;

%printit(sasuser,veneer)
%printit(sasuser,retail)
```

The code below illustrates how information is extracted from the SASHELP views. Although the code accomplishes the same task, it takes less CPU time to extract from the Dictionary Tables than from the SASHELP Views.

```
%macro printit(libname,memname);
  data _null_;
    set sashelp.vtable;
    where libname="%&libname" and
           memname="%&memname";
    call symput("label",memlabel);
  run;

  proc print data=%&libname..%&memname;
    title "%&label";
  run;
%mend printit;

%printit(sasuser,veneer)
%printit(sasuser,retail)
```

The SASHELP views can also be browsed interactively. When the Display Manager is used, the LIBNAME statement can be entered on the Command Line. The SASHELP library can then be selected, and a view can be browsed. A Where clause can be used when browsing to filter the data, until the appropriate information is retrieved. The SASHELP views can be very useful to a developer who needs to learn more about the files in the SAS session.

There are numerous ways in which the dictionary tables and SASHELP views can be utilized. However one of the most helpful things that can be done with this information is to document a system. Reports can be generated that list out the variables from numerous data sets. Information can also be gathered from a number of catalogs to report on the entries used in an interactive system.

Since the dictionary tables are dynamic, they can also be a good tool for determining what is in the SAS environment during an interactive application. Code that would previously have to be executed at the operating system level (or be operating-system dependent), can now be written more generically in SAS, since information about path names, and creation and modification dates, is stored in the tables and views.

CONCLUSION

The dictionary tables and SASHELP views that are available with version 6.07 of the SAS system are a very useful tool for SAS users of all levels. Information from the tables and views can be used in report generation and system documentation. When applications are developed using this meta-data, they can be more generic and less operating-system dependent. It is also very easy to do ad-hoc queries of this data when using the Display Manager. Furthermore, one of the best ways to learn about the SAS system is to examine the meta-data that is contained in these tables.

ACKNOWLEDGEMENTS

I would like to thank Frank DiIorio for introducing me to dictionary tables and SASHELP views, and for assisting me with this paper.