

Paper 066-2009

%LibDoc: A library documentation macro

Richard Koopmann Jr., Capella University, Minneapolis, MN

ABSTRACT

%LibDoc is a simple macro that produces an Excel workbook containing meta data for a given library. The metadata is pulled from dictionary.* tables via proc sql. The macro is also implemented as a SAS Explorer Action.

Code was developed with SAS 9.1.3 SP4 and HTML Help Workshop 4.74 running under Windows XP Professional.

INTRODUCTION

Codebooks are often desired for documenting data sets. The contents procedure provides a simple way to get useful information for a single data set, but what about documenting an entire library of data sets? What about documenting format catalogs? Table views? Indexes? To address these additional documentation requirements, we'll exploit the dictionary tables. (While there are corresponding views in the SASHELP library which could be accessed in the DATA STEP, we'll access the dictionary tables via the SQL procedure. Querying the dictionary tables is considerably faster than going against the SASHELP views.)

The dictionary tables contain a wealth of information on various aspects of the current SAS session including library metadata, options, external files, libname engines, macro variables, titles, and footnotes. The LibDoc macro accesses the following dictionary tables containing library-specific metadata:

dictionary.	Metadata returned
libnames	Library name, engine, path, concatenation level, format, read only flag, sequential flag, system description/name/value.
tables	Library name, member name/type/label, data set type, creation/modification date, number of observations, observation length, number of variables, password protection, compression, encryption, number of pages, etc.
views	Library name, member name/type, engine.
indexes	Library name, member name/type, column name, index type/name/position, nomiss/unique flags.
columns	Library name, member name/type, variable name,
catalogs	Library name, member name/type, object name/type/description, creation/modification date, alias, concatenation level.
members	Library name, member name/type, engine, presence of indexes, path.
formats	Library name, member name, path, object name, format name/type, source, minimum/maximum/default width and decimal width.

The *Summary of SAS Dictionary Tables and Views* tip sheet is a very useful and well-organized reference.

While `dictionary.formats` provides metadata on what informats and formats are applied to a given variable, often it is desirable to know how these informats and formats are defined. Fortunately, the `FORMAT` procedure offers an option for extracting definitions from a specific library into a SAS data set. The resulting control data set is then processed along with the other dictionary tables.

MACRO PARAMETERS

Parameter	Description	Default
LIBNAME	Identifies the SAS library to be processed.	<i>Required parameter</i>
PATH	Location for writing the output files.	C:\TEMP
FILE	Name of output file (Excel and PDF). Prefix of output files (HTML).	&LIBNAME
CHM	Generate a compiled help file. Requires HTML Help Workshop to be installed.	NO

Parameter	Description	Default
HTML	Generate HTML files for each table. Each filename will begin with the value of FILE parameter.	&CHM
PDF	Generate a PDF version of the documentation.	NO
STYLE	Set style for ODS outputs.	MINIMAL

Usage notes: Submitting %LibDoc(); or %LibDoc(HELP); will print help information in the log window. Submitting %LibDoc(EDIT); will open the macro source file in the editor window for revisions. Thus, the LibDoc macro will not process SAS libraries that are named EDIT or HELP.

SUPPORTED DESTINATIONS

The LibDoc macro uses the Excel libname engine and the HTML and PDF ODS Destinations. The macro may be modified to use the TAGSETS.EXCELXP ODS destination in lieu of the Excel libname engine if SAS/ACCESS Interface to PC Files is unavailable.

GENERATING A COMPILED HELP FILE

With HTML Help Workshop (HHW) installed, the LibDoc macro can generate a compiled help (CHM) file automatically. HHW compiles a set of HTML files into a single CHM file. To achieve this, the macro generates the project files needed by HHW; these project files include references to the HTML versions of the dictionary tables. The macro then uses the x command to generate the CHM file.

```
x ""C:\Program Files\HTML Help Workshop\hhc" ""&PATH.\&FILE..hhp"";
```

IMPLEMENTING LIBDOC AS A SAS EXPLORER ACTION

If the LibDoc macro is compiled at startup or is included in the autocall macro directory, the macro call can be implemented as a SAS Explorer Action. While users cannot customize explorer actions associated with SASLibrary metadata entries, we can add actions to SAS Library members (e.g., tables, views, & catalogs). Once installed, users can execute the macro by simply right-clicking on the table and selecting the LibDoc action.

To add the LibDoc action to tables, create a text file at C:\LibDoc_Action.txt containing:

```
[CORE\EXPLORER\MENUES\MEMBERS\TABLE]
"LibDoc"="gsubmit '%LibDoc(%8b)';";
```

The action is imported with the REGISTRY procedure as follows:

```
proc registry import='C:\LibDoc_Action.txt'; run;
```

The new explorer action can also be added via the GUI. With the Explorer pane active, select Tools > Options > Explorer. Under the Members tab, select TABLE member and click Edit followed by Add. Set the Action field to LibDoc and the Action Command field to gsubmit '%LibDoc(%8b)';. Press OK, OK, OK. Right-click on a table and see the LibDoc action command.

Optional macro parameters can be added as desired. For example, gsubmit '%LibDoc(%8b, PDF=yes)';.

CONCLUSION

The LibDoc macro provides a quick method for compiling metadata on a specific SAS library and exporting this to three common destinations (Excel, PDF, and CHM).

REFERENCES

CodeCrafter's, Inc. (2007). Summary of SAS Dictionary Tables and Views.

<http://www.codecraftersinc.com/pdf/DictionaryTablesRefCard.pdf>

Microsoft Corporation. (1999). Extend Office Help with Your Own Additional Resources.

<http://office.microsoft.com/en-us/ork2000/HA011384521033.aspx>

RECOMMENDED READING

For more information on SAS Explorer Actions, see Richard A. DeVenzia's website:

<http://www.devenzia.com/downloads/sas/actions/>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Richard Koopmann
 Capella University
 225 S 6th St, 9th Fl
 Minneapolis, MN 55402
 richard [dot] koopmann [at] capella [dot] edu
 www.capella.edu

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

%LIBDOC CODE

Note: Updates to the LibDoc macro will be freely available at <http://sas-quatch.googlecode.com>.

```
%macro LibDoc(
  libname
, path=C:\temp
, file=&LIBNAME.
, chm=NO
, html=&CHM
, pdf=NO
, style=minimal
);

%let libname = %upcase(&LIBNAME);

%if &LIBNAME eq HELP or &LIBNAME eq %str( ) %then %do;
%put *****;
%put * LIBDOC documents a SAS library to an Excel workbook. *;
%put *****;
%put * Positional Parameters (in this order): *;
%put * LIBRARY The LIBRARY to document. *;
%put * *;
%put * Optional Keyword Parameters (in any order): *;
%put * PATH Path to save the Excel workbook. Defaults to H drive of user. *;
%put * This MUST be a fully qualified network directory--no local drives. *;
%put * FILE File to save the Excel workbook. Defaults to H drive of user. *;
%put * This MUST be a fully qualified network directory--no local drives. *;
%put * CHM Optionally compile CHM file from HTML files. Defaults to NO. *;
%put * HTML Optionally create set of HTML files. Defaults to &CHM. *;
%put * PDF Optionally a PDF file. Defaults to NO. *;
%put * STYLE Specify a style for optional output destinations. Defaults to minimal*;
%put *****;
%put * Example macro call *;
%put * LibDoc( WORK, path=C:\temp, file=myWork) *;
%put *****;
%goto ByeBye;
%end;

%else %if &LIBNAME eq EDIT %then %do; /* location of LibDoc macro source file */
dm wedit 'whostedit "\\mspfile02\MPLSData\sas\macros\LibDoc.sas" ';
%goto ByeBye;
%end;

x "if exist "&PATH.\&FILE..xls" del "&PATH.\&FILE..xls"";
libname ld excel "&PATH.\&FILE..xls";
```

```

/*****
fetch library metadata.
*****/;
proc sql;
  create table ld.libnames(drop=libname) as select *
  from dictionary.libnames where libname=&LIBNAME";
  create table ld.tables(drop=libname) as select *
  from dictionary.tables where libname=&LIBNAME";
  create table ld.views(drop=libname) as select *
  from dictionary.views where libname=&LIBNAME";
  create table ld.indexes(drop=libname) as select *
  from dictionary.indexes where libname=&LIBNAME";
  create table ld.columns(drop=libname) as select *
  from dictionary.columns where libname=&LIBNAME";
  create table ld.catalogs(drop=libname) as select *
  from dictionary.catalogs where libname=&LIBNAME";
  create table ld.members(drop=libname) as select *
  from dictionary.members where libname=&LIBNAME";
  create table ld.formats(drop=libname) as select *
  from dictionary.formats where libname=&LIBNAME";
quit;
proc format cntlout=fmtdefs library=&LIBNAME; run;
/*the excel engine doesn't appreciate numeric variables with length < 8*/
proc sql;
  create table ld.format_definitions as
  select FMTNAME, START, END, LABEL,
  MIN length=8
  , MAX length=8
  , DEFAULT length=8
  , LENGTH length=8
  , FUZZ, PREFIX, MULT, FILL
  , NOEDIT length=8
  , TYPE, SEXCL, EEXCL, HLO, DECSEP, DIG3SEP, DATATYPE, LANGUAGE
  from fmtdefs;
quit;

%if "&HTML" ne "NO" OR "&PDF" ne "NO" %then %do;
/*****
if a table has 0 observations, insert 1; otherwise, 1:1 copy.
*****/;
data ld_libnames; if _obs_=0 then output; set ld.libnames nobs=_obs_; output; run;
data ld_tables; if _obs_=0 then output; set ld.tables nobs=_obs_; output; run;
data ld_views; if _obs_=0 then output; set ld.views nobs=_obs_; output; run;
data ld_indexes; if _obs_=0 then output; set ld.indexes nobs=_obs_; output; run;
data ld_columns; if _obs_=0 then output; set ld.columns nobs=_obs_; output; run;
data ld_catalogs; if _obs_=0 then output; set ld.catalogs nobs=_obs_; output; run;
data ld_members; if _obs_=0 then output; set ld.members nobs=_obs_; output; run;
data ld_formats; if _obs_=0 then output; set ld.formats nobs=_obs_; output; run;
data ld_format_definitions;
  if _obs_=0 then output; set ld.format_definitions
  nobs=_obs_; output; run;

/*****
generate the desired output.
variables listed on the ID statement help uniquely identify rows.
*****/;
ods listing close;
ods noproctitle;
option orientation=landscape;
%if "&PDF" ne "NO" %then ods pdf file="&PATH.\&FILE..pdf" style=&style uniform
fontscale=80;;

ods html file="&PATH.\&FILE.libnames.html" (title="libnames") style=&style;
ods proclabel 'Libnames';
proc print data=ld_libnames contents=""; id path level; run;
ods html close;

```

```

ods proclabel 'Members';
ods html file="%PATH.\&FILE.members.html" (title="members") style=&style;
proc print data=ld_members contents=""; id memname memtype; run;
ods html close;

ods proclabel 'Tables';
ods html file="%PATH.\&FILE.tables.html" (title="tables") style=&style;
proc print data=ld_tables contents=""; id memname memtype; run;
ods html close;

ods proclabel 'Columns';
ods html file="%PATH.\&FILE.columns.html" (title="columns") style=&style;
proc print data=ld_columns contents=""; id memname memtype name; run;
ods html close;

ods proclabel 'Indexes';
ods html file="%PATH.\&FILE.indexes.html" (title="indexes") style=&style;
proc print data=ld_indexes contents=""; id memname name indxname; run;
ods html close;

ods proclabel 'Views';
ods html file="%PATH.\&FILE.views.html" (title="views") style=&style;
proc print data=ld_views contents=""; id memname; run;
ods html close;

ods proclabel 'Catalogs';
ods html file="%PATH.\&FILE.catalogs.html" (title="catalogs") style=&style;
proc print data=ld_catalogs contents=""; id memname objname objtype; run;
ods html close;

ods proclabel 'Formats';
ods html file="%PATH.\&FILE.formats.html" (title="formats") style=&style;
proc print data=ld_formats contents=""; id memname path objname fmtname fmttype; run;
ods html close;

ods proclabel 'Format Definitions';
ods html file="%PATH.\&FILE.format_definitions.html" (title="format definitions")
style=&style;
proc print data=ld_format_definitions contents=""; id fmtname; run;
ods html close;

%if "&PDF" ne "NO" %then ods pdf close;;
option orientation=portrait;
ods proctitle;
ods listing;
%end;

libname ld excel clear;

%if "&CHM" ne "NO" %then %do;
data _null_;
file "%PATH.\&FILE..hnp";
put '[OPTIONS]';
put 'Compatibility=1.1 or later';
put "Compiled file=&FILE..chm";
put "Contents file=&FILE..hhc";
put "Default topic=&FILE.libnames.HTML";
put 'Display compile progress=No';
put 'Language=0x409 English (United States)';
put "Title=&LIBNAME as of &SYSDATE9 - LibDoc";
put ;
put '[FILES]';
put "&FILE.catalogs.HTML";
put "&FILE.columns.HTML";
put "&FILE.formats.HTML";
put "&FILE.format_definitions.HTML";
put "&FILE.indexes.HTML";
put "&FILE.libnames.HTML";
put "&FILE.members.HTML";
put "&FILE.tables.HTML";

```

```

put "&FILE.views.HTML";
put ;
put '[INFOTYPES]';
put ;
run;

data _null_;
file "&PATH.\&FILE..hhc";
put "<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">";
put "<HTML><HEAD>";
put "<!-- Sitemap 1.0 -->";
put "</HEAD><BODY>";
put "<OBJECT type=\"text/site properties\">";
put "<param name=\"ImageType\" value=\"Folder\">";
put "</OBJECT>";
put "<UL>";
put "<LI><OBJECT type=\"text/sitemap\"><param name=\"Name\" value=\"Library\">";
put "    <param name=\"Local\" value=\"&FILE.libnames.HTML\"></OBJECT>";
put "<LI><OBJECT type=\"text/sitemap\"><param name=\"Name\" value=\"Members\">";
put "    <param name=\"Local\" value=\"&FILE.members.HTML\"></OBJECT>";
put "<LI><OBJECT type=\"text/sitemap\"><param name=\"Name\" value=\"Tables\">";
put "    <param name=\"Local\" value=\"&FILE.tables.HTML\"></OBJECT>";
put "</UL>";
put "<LI><OBJECT type=\"text/sitemap\"><param name=\"Name\" value=\"Columns\">";
put "    <param name=\"Local\" value=\"&FILE.columns.HTML\"></OBJECT>";
put "<LI><OBJECT type=\"text/sitemap\"><param name=\"Name\" value=\"Indexes\">";
put "    <param name=\"Local\" value=\"&FILE.indexes.HTML\"></OBJECT>";
put "</UL>";
put "<LI><OBJECT type=\"text/sitemap\"><param name=\"Name\" value=\"Views\">";
put "    <param name=\"Local\" value=\"&FILE.views.HTML\"></OBJECT>";
put "<LI><OBJECT type=\"text/sitemap\"><param name=\"Name\" value=\"Catalogs\">";
put "    <param name=\"Local\" value=\"&FILE.catalogs.HTML\"></OBJECT>";
put "<LI><OBJECT type=\"text/sitemap\"><param name=\"Name\" value=\"Formats\">";
put "    <param name=\"Local\" value=\"&FILE.formats.HTML\"></OBJECT>";
put "</UL>";
put "<LI><OBJECT type=\"text/sitemap\"><param name=\"Name\" value=\"Definitions\">";
put "    <param name=\"Local\" value=\"&FILE.format_definitions.HTML\"></OBJECT>";
put "</UL>";
put "</UL>";
put "</BODY></HTML>";
run;

x "C:\Program Files\HTML Help Workshop\hhc" "&PATH.\&FILE..hhp";
%end;

proc datasets library=work nodetails nolist; delete ld_; run; quit;

%ByeBye;
%mend;

```