

Output Generating System – A Tool for Creating Tables and Listings in Word

Jay Zhou, Quintiles, Inc., Kansas City, MO

Abstract

The Output Generating System (OGS) is a collection of 10 SAS® macros used to produce tables, listings, and their shells directly as Word table documents. A central concept in OGS is to link macros together to perform separate but inter-related tasks based on the layout and attributes of five sections (page header, title, column header, table body, and footnote) of a page. The user utilizes these macros like building blocks to produce customized tables and listings. The system provides the following main features: (a) creation of table and listing shells with dummy data, (b) ability to produce attractive output with a minimum of effort to control the appearance of the output by specifying style, page orientation, margins, font type, font size, and gridlines, (c) capacity to generate sophisticated tables or listings with less effort, (d) pagination that includes customizing features (18 available styles) and total number of pages, (e) management of a central ASCII file that allows rapid updating of external text such as page headers, titles, and footnotes, and (f) automation of MS Word to preview output in Word. The system conforms to SAS user standards, so it is easily learned by SAS programmers at different skill levels. This paper provides the general overview of the system.

Introduction

During drug development in the pharmaceutical industry, an important step is to conduct clinical trials that study the safety and efficacy of a drug for regulatory submissions. In most cases, data from the trials are analyzed with the SAS system. A major task of SAS programmers is to generate summary tables and listings programmatically with SAS for clinical study reports. The process itself can be labor intensive and time-consuming. The most common method practiced by programmers is to use PROC REPORT or a DATA _NULL_ step, which works together (prior to SAS Version 7) with PROC PRINTTO to create ASCII text files. However, the limitations of SAS ASCII files incorporating special symbols, sub/superscripts, fonts, font sizes, and margins prevent users from creating more attractive and usable deliverables, especially in this day of desktop document editors and publishing systems. Hence, more and more medical writers, biostatisticians, publishers, and expert reviewers now request SAS reports in Word document format. But page orientation, text alignment, and readability are lost if SAS ASCII output is directly opened in Word, and potentially time-consuming manual post-processing or manipulation of SAS ASCII outputs is required to maintain its character alignment and readability in Word.

For that reason, SAS programmers have been trying to develop innovative and effective ways of creating Word documents. Because of the formatting limitation of SAS, other formatting languages such as RTF (Rich Text Format), HTML (HyperText Markup Language), PostScript, PDF (Portable Document Format), and XML (Extensible Markup Language) are involved in the creation of output. Some programmers take advantage of interfacing SAS with Word via DDE (Dynamic Data Exchange) to post-process the output with VBA (Visual Basic for Applications).

Wehr (1996) introduced his powerful Print Driver (%print) that can generate text, RTF, PostScript, or HTML files. Because the %print macro can only perform a single format task with each invocation, many macro calls within a DATA _NULL_ step, involving substantial programming effort, are needed to generate even a simple table, which might result in time-consuming and difficult program maintenance.

Cunningham (1998) developed a SAS macro to insert delimiters between rows and columns and to convert the delimited text table to an MS Word table by interfacing with MS Word and running a VBA macro. This method works well with any fonts, including proportional fonts for small and simple tables. It may not be applicable for large and complex tables, as sometimes rows or columns cannot easily be separated, and some manual editing cannot be omitted.

Peszek, et al. (1999), presented two SAS macros, %rtf and %wrapup, to generate Word tables in RTF format. The first macro, used within a DATA _NULL_ step, provides full control over the table appearance with multiple calls. It can produce attractive Word tables with proportional or monospace fonts, but extensive experience in the DATA _NULL_ step with PUT statements, as well as a substantial amount of programming effort, are required. The second macro can generate an RTF table with a single call, but it may not be flexible enough to create a complex table or listing with multiple pages.

Yam (2000) described a method used to generate Word tables by transferring SAS data to Excel with DDE, which serves as an intermediary for repackaging SAS data with a VBA macro and interfacing with Word via OLE (Object Linking and Embedding) to transfer the Excel table into Word. With this method, 24 SAS variables must be derived in order for Excel to format a table appropriately. Therefore, the process may be time-consuming and difficult to maintain.

Zhou (2001) introduced a SAS2WORD macro to automate the conversion of any ASCII file to a Word document by employing RTF language as a bridge and taking advantage of interfacing with Word via DDE. The macro functions as an RTF 'writer' to embed RTF control words and symbols in the text file. The macro invokes Word to insert the RTF file into Word and save it as Word document (if requested). The macro also gives users an opportunity to customize pagination, including the total number of pages, page numbering styles, and positions. However, a shortcoming of the macro is that monospace fonts should be used as it cannot convert an ASCII output into a readable Word document with proportional fonts.

SAS Institute has improved the creation of outputs formatted to be portable to other applications. Beginning with Version 7, the Output Delivery System (ODS) was introduced to overcome the limitations of traditional SAS output and to provide new formatting options to users. ODS is a method of delivering output in a variety of formats and of making the formatted output easy to access (SAS, 1999). With ODS, procedure outputs became much more flexible. Because ODS only provides a way for the user to choose individual output objects to send to ODS destinations, it still relies on other procedures to generate output. Hence, the output still cannot meet the high standards demanded in many pharmaceutical companies, even though ODS provides table definitions that define the structure of the output from procedures or a DATA step. Working together with ODS, the PROC TEMPLATE procedure allows the user to customize the definitions of a table template, but this procedure still cannot handle extremely sophisticated tables, is not necessarily easy to use, and can be time-consuming.

The Output Generating System (OGS) was originally developed as a result of a challenge to provide tables and listings with sub/superscripts and symbols for pharmacokinetic analyses. The system is a collection of SAS macros used to produce tables, listings, and their shells directly as Word table documents. A central concept in OGS is to link macros together to perform separate but inter-related tasks. The system conforms to SAS user standards, so it is easily learned by SAS programmers of different skill levels. It is designed for the Windows, Unix, and open VMS platforms, and can be called in batch or non-batch mode. The following features are provided:

- Creation of table and listing shells with code that can be recycled for production.
- Ability to control the appearance of output using a variety of different attributes, including style, page orientation, margins, proportional or non-proportional font type, and font size.

- Pagination that includes customizing features (18 available styles) and total number of pages.
- Creation of multipage tables or listings that may be divided into parts with different formats if too many columns exist for one page. Ability to interleaf or separate the parts (e.g., Part 1 of 3).
- Management of a central file that allows rapid updating of external text such as page headers, titles, and footnotes.
- Automation of MS Word to preview output in Word.
- Standardization of table and listing production to facilitate workload sharing across sites within a company.
- Ability to build a code library for common summary tables and listings.
- More efficient programming process.

The OGS Concept

The OGS macro system is designed using SAS macro language. However, in order to allow OGS output to be portable to word processors, the RTF language is utilized in formatting the output. The system also interfaces SAS with MS Word using WordBasic commands via DDE if used in the Windows environment.

The main logic of the OGS system design is based on the layout of a page. In general, a single page of a particular table or listing can be divided structurally into five sections from the top to the bottom: page header, title, column header, table body, and footnote (Figure 1). Understanding this concept is essential, because several macros and macro parameters in OGS are based on this concept.

Client Name		Page 1 of 1
Protocol Number		15AUG01:16:21
Table 14.1 Subject Disposition (Intent-to-Treat Population)		
	Treatment A (N=25)	Treatment B (N=25)
Randomized	25 (100%)	25 (100%)
Completed	24 (96%)	22 (88%)
Discontinued	1 (4%)	3 (12%)
Reason for Discontinuation		
Reason 1	1 (4%)	2 (8%)
Reason 2	0	1 (4%)
Reference: Listing 16.2.1 Program: c:\client\study\program\tsubdisp.sas		

Figure 1. Simplified table shows the page header, title, column header, table body, and footnote sections separated by dashes lines from top to bottom, respectively

The text and data presented in tables or listings of clinical study reports are displayed in the specific order. For a particular table or listing, the client name, drug name, protocol number, and/or protocol title are usually displayed in the page header section, and the text appearance is kept consistent across studies for an entire drug project. The table or listing number and titles are presented in the title section. The column headers, including column-spanning headers, in the column header section label the fields in the table body section where the data are presented. Any additional explanations to titles, column headers, values in the table body, and data references are listed in the footnote section in a customized order. Page number, program name and path, and a date/time stamp can be presented either in the page header or in the footnote section, depending on client preference. In most cases, only the data displayed in the table body section are from the internal SAS data source, and text in other sections is from an external source or defined by parameters in macros.

In general, the presentation of the title, column header, and footnote sections are not changed for the same table or listing across multiple pages. However, when there are too many columns to be presented on a page, a table or listing may need to be divided into different parts (e.g., Part 1 of 3). In this case, the text in the title, column header, and footnote sections, and the table appearance may be different among the parts, but should be the same within a part.

The OGS System

The OGS macro system consists of a family of 10 SAS macros that streamline report programming. The user utilizes these macros like building blocks to produce customized tables and listings. These macros and their overall purposes are listed in Table 1, followed by text further describing the function of each macro. Details of the macro syntax and individual parameters are beyond the scope of this paper and will be presented in subsequent papers.

Table 1. OGS macro descriptions

Name	Description
ZINITIAL	Initialize global macro variables used for the rest of OGS
ZDEFINE	Produce an individual column definition
ZCS	Initialize a column-spanning header
ZCSEND	Terminate a column-spanning header
ZINSERT	Insert internal data and external text to the output
ZSHELL	Make a dummy SAS data set used to create a table or listing shell
ZPAGENUM	Prepare the data for presentation by sorting and paginating them
ZRTF	Embed RTF language code
ZTITLE	Import external text from a text file to create macro variables for page header, titles and footnotes
ZREPORT	Assemble output into the final product

The ZINITIAL macro is a required macro used to initialize the OGS by defining global macro variables fundamental to the system. These macro variables are established to

control the functions of other components of the system and are designed to keep track of the number of columns and spanning headers, the program name and path, output destination, page layout, and text appearance. Particular values for global macro variables such as program name may later be defined with a **%let** statement within each specific program. Ideally, this macro should be called only once per study in order to maintain consistency for the entire study. It is recommended this macro be called in the study setup or autoexec program.

The ZDEFINE macro is designed to specify attributes such as variable, format, column header and width, decimal alignment, justification, and the presence of a vertical line in the column field of a table or listing. The function of this macro is very similar to the DEFINE statement in PROC REPORT. Each call to the macro will define one column. To facilitate the creation of spanning headers, two macros, ZCS (CS stands for column spanning) and ZCSEND, are designed in the system. The ZCS macro initializes the spanning while the ZCSEND macro terminates the spanning. These two macros are only used for specifying column-spanning headers and must be called in conjunction with the ZDEFINE macro. Therefore, these three macros work together to control the column header section.

Placing internal data or external text in a specific section of a table or listing can be very challenging. Like the DEFINE statement in PROC REPORT, the ZDEFINE macro only defines one variable for one column per call, so it cannot display more than one variable in the same column without some level of pre-processing (e.g., concatenation of two fields). For example, it is not easy in an adverse event table or listing for the ZDEFINE macro to print the body system and preferred term in the same column without concatenation. In other cases, data values need to be displayed in the title or column header section in order to group the data based on treatment, investigator site, or sub-population. To meet these challenges, the ZINSERT macro was designed to increase reporting capabilities by allowing the user to insert a blank line, external text, or internal data into the page header, title, column header, table body, and footnote sections.

Due to regulations for the pharmaceutical industry and increased pressures to produce tables and listings quickly after database closure, shells outlining the content and format of desired tables and listings are commonly developed before real data is available. Along with documentation such as Data Analysis and Reporting Plan (DARP) and Statistical Analysis Plan (SAP), shells have become an important component of the package reviewed and approved by clients since tables and listings are produced based on the shells. Often effort is spent to produce shells manually, followed by additional effort to program the tables and listings. New programming code

developed to populate the shells with real data often inefficiently uses external text (titles and footnotes) from a different source than used for the shell production.

One of the advantages of the OGS system is the ability to programmatically create a table or listing shell in the absence of real data. The ZSHELL macro provides an easy way to create a dummy SAS data set that can be used for generating the table or listing shell. It gives the user an opportunity to start report programming even before data are available. Once the study data are available, the code can be recycled with little modification, so that the appearance of the outputs will remain the same as the shells.

Reports typically include page numbering, which is readily available using SAS' NUMBER and PAGENO options to specify pagination and to reset page numbering, respectively. However, there is no option that provides pagination that includes any customizing features or the total number of pages. In the OGS system, the ZPAGENUM macro is designed to control the pagination by sorting and paginating the input file. The macro prepares a field containing page numbers in one of 18 available pagination styles (Table 2), which will replace the token PAGEXOFY (or pagexofy) present in a page header, title, or footnote line.

Table 2. Pagination Style and Example

Style	Example	Style	Example
pagexofy	Page 1 of 10	pagex	Page 1
pagexofyp	(Page 1 of 10)	pagexp	(Page 1)
xofy	1 of 10	pagen	1
xofyp	(1 of 10)	pagenp	(1)
pxofy	p. 1 of 10	pagexpy	Page 1/10
pxofyp	(p. 1 of 10)	pagexpyp	(Page 1/10)
px	p. 1	pxpy	p. 1/10
pxp	(p. 1)	pxpyp	(p. 1/10)
xpy	1/10	xpyp	(1/10)

The ZPAGENUM macro is also used to manipulate the input data set specified by the **dsin** parameter for the ZREPORT macro to report. It can split the data into parts and flag the places to insert a page break or a blank line, if necessary. The macro can handle an empty data set or a dummy data set created by the ZSHELL macro. If ZSHELL is called, ZPAGENUM will process the dummy data set, even though real data may be available. Hence, the ZSHELL macro should be suppressed when the shell program is recycled for production.

For a clinical study report, a particular table or listing usually presents both internal SAS data and external text such as page headers, titles, and footnotes. In order for the OGS system to work appropriately, a special ASCII text file, TITLE.TXT, must be created prior to the call to the system. This file centralizes all external text, which allows for easy maintenance of the text. The ZTITLE macro is designed to provide the user a tool for retrieving these page

headers, titles, and footnotes from this text file. The special structure and preparation of the TITLE.TXT file will be a topic of future papers.

All macros discussed to this point are used to initialize and define modules for building an output (table or listing). ZREPORT, the capstone macro, is the last macro in the OGS system to be called and must be used at the end of the program. ZREPORT serves as the assembly center to gather the parts for table creation by invoking the ZTITLE and ZRTF macros. The ZRTF macro as the backbone of the OGS system is designed specifically for the ZREPORT macro and functions as an RTF 'writer' to embed RTF control words and symbols that format the column and row attributes of the output.

In many cases, the user only needs to program calls to ZINITIAL, ZDEFINE, ZPAGENUM, and ZREPORT to produce the desired output. These four macros follow the four basic steps defined below, which constitute a call cycle to the OGS system. Each call cycle will generate a table or listing (Figure 2).

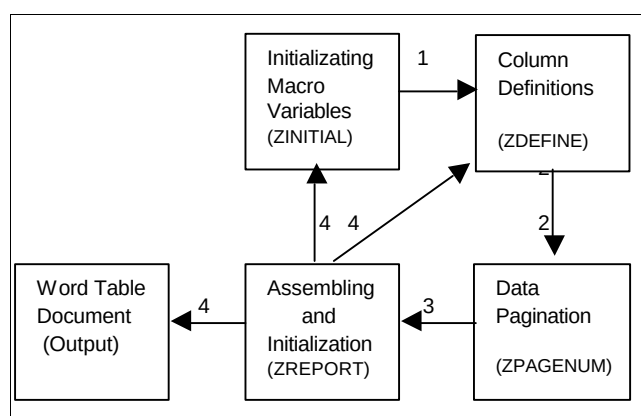


Figure 2. A typical OGS call cycle to create a Word document

A typical OGS call cycle will:

1. Initialize basic global macro variables to establish an environment for the other macros in OGS to work properly across tables and listings for the entire study. The variables for the first call cycle must be initialized by the ZINITIAL macro. The ZINITIAL macro is not required in other call cycles if the basic settings, such as the page orientation, margins, font, and font size are the same.
2. Define specific column attributes and structures of a table or listing. In this step, ZDEFINE is the primary macro to define the structures and properties for the column header and table body sections. Other supplemental macros involved in this step (i.e., ZCS, ZCSEND, and ZINSERT) perform additional tasks for customizing output.
3. Define and manipulate source data for the pagination of reporting, based on information provided in previous

steps. The ZPAGENUM macro performs this function to prepare the data of reporting.

4. Assemble the output with ZREPORT, which calls ZRTF and ZTITLE to format the output with the data prepared by ZPAGENUM and external text from a text file. In this step, the system will communicate with MS Word, if output is successfully generated in the Windows environment, to insert the output to Word and save it as Word document when requested. At the end of the ZREPORT execution, the global macro variables are re-initialized to set up the environment for the next call cycle.

Example of a call cycle to OGS

The example provided in this section illustrates one OGS call cycle to the ZINITIAL, ZDEFINE, ZINSERT, ZCS, ZCSEND, ZPAGENUM, and ZREPORT macros. The code below is written to create a statistical summary table for PK parameter data, and the output is shown in Figure 3. Not all parameters with the default values are shown in the calls to the macros. Since the **textfile** parameter is specified in the call to ZINITIAL, the ZTITLE macro will be invoked by the ZREPORT macro.

There are 7 calls to the ZDEFINE macro defining seven columns. The first two columns are left justified as specified in the calls with **j=1**. For illustration, the decimal position is purposely not aligned for the third column as the **align** parameter is defined with 0. The decimal alignment is corrected in other columns that have numeric variables with the **align** parameter defined appropriately. Shifting the position from right to left is demonstrated as the value of **align** increases from 1 for the 4th column to 6 for the 7th column. The 5th column is a blank column, separating the Analyte One group from the Analyte Two group, defined by a call without specifying a variable or a column header, but with **width=1** to minimize the width of the column.

A call to the ZINSERT macro is given in the example to demonstrate the flexibility of the OGS system. The first column of the table contains the data from two fields: the PK parameters and the associated units (Figure 3). The first call to ZDEFINE is followed by the ZINSERT macro in order to insert the unit data into the column after the parameters are displayed. In order to perform this function correctly, the **where** parameter in the call to the ZINSERT macro must be specified with a logic condition (e.g., **where=%str(stat=2)**) is given in the example, using the value of the next column in the same row as an indicator to specify the location.

A call to the ZPAGENUM macro is required for each table or listing, even if no page numbers are needed in the output. In order for the macro to work properly, the variables specified in the **byvar** parameter must include those defined in the **pagevar** and **skipvar** parameters of ZPAGENUM

and in the **var** parameter of the ZDEFINE macro with **id=y**. Other variables, not defined in any macro, but used for the purpose of sorting, can be included in the call. The parameter **rows** specified with 20 in the macro call indicates that there are 20 lines available for data presentation in the table body section.

Example of a call cycle to the OGS system:

```
*** Data preparation & analysis steps omitted ***;

*** OGS macros for data presentation ***;
*** Initialize the reporting process ***;
%let _path=c:\client\study\program;
%zinitial(progpath=_path, progname=tpksum,
          textfile=_path\title.txt, font=arial)

*** define columns ***;
%zdefine(var=pkpar, header=Parameter#(unit),
          id=y, format=pkpar., j=1, width=7)
%zinsert(var=unit,where=%str(stat=2))
%zdefine(var=stat, header=Statistics, j=1,
          format=stat., width=7)
%zcs(text=Analyte One)
  %zdefine(var=t1, header=Treatment A, align=0)
  %zdefine(var=t3, header=Combination, align=1)
%zcsend;
%zdefine(width=1); *** insert a blank column ***;
%zcs(text=Analyte Two)
  %zdefine(var=t2, header=Treatment B, align=2)
  %zdefine(var=t3, header=Combination, align=6)
%zcsend

*** get the page number ***;
%zpagenum(dsin=pkdata, byvar=pkpar stat,
          pagevar=pkpar, skipvar=pkpar, rows=20)

*** report the table ***;
%zreport(show=y)
```

Conclusion

The OGS system presented in this paper provides a new, simple, flexible, and fast way to create attractive Word documents in MS Word directly from SAS data set. It is page-oriented, meaning it is based on the characteristics of five sections of a page: page header, title, column header, table body, and footnote. Many options are given to users to control the appearance of the document with page orientation, paragraph style, margins, font, font size, sub/superscript, and gridlines (style and thickness). It also provides programmers with 18 pagination style options. With the flexibility of the ZINSERT macro, the OGS system can create very sophisticated tables or listings. The OGS system is not only able to create a table or listing shell, but is also able to recycle the code used for shell creation to generate a table or listing in production with the same appearance as the shell, since all external text is from the same source file. In conclusion, the OGS system greatly simplifies the task of creating tables and listings directly in Word.

Client Name		Protocol Title		Page 1 of 3	
Protocol Number					
Table 14.10					
Pharmacokinetic Parameters for Each Analyte by Treatment (Pharmacokinetic Population)					
Parameter (Unit)	Statistics	Analyte One		Analyte Two	
		Treatment A	Combination ^a	Treatment B	Combination ^a
C _{max} (ng/mL)	N	20	18	20	18
	Mean	36.09	30.28	31.80	28.57
	Std Dev	10.75	8.09	14.56	10.54
	%CV	25.6	24.2	41.6	34.9
	Median	35.90	29.90	27.20	25.60
	Minimum	21.80	17.60	17.40	13.50
	Maximum	55.10	44.90	55.90	48.40
AUC _(0-∞) (hr*ng/mL)	N	20	18	20	18
	Mean	348.72	336.46	323.85	357.51
	Std Dev	147.61	69.53	115.12	96.67
	%CV	49.1	29.4	32.4	34.1
	Median	356.52	327.25	317.97	376.81
	Minimum	174.00	252.75	177.76	215.16
	Maximum	514.30	425.29	491.35	460.98
^a Subjects 0005 and 0019 were not included in the summary due to missing data.					
Reference: Listing 14.2.30					
Program: c:\client\study\program\tpksum.sas					
15AUG2001:11:27					

Figure 3. A Word table created with OGS showing the text appearance and column attributes

References

- Cunningham, G. (1998). An Easy-to-Use SAS Macro for Use with Microsoft Windows That Converts Existing Text Tables to Microsoft Word Tables. Proceedings of the PharmaSUG 98 Conference, 304-308.
- Microsoft Press (1997), Microsoft MS-DOS, Windows, Windows NT, and Apple Macintosh Applications: Rich Text Format (RTF) Specification and Sample RTF Reader Program, Microsoft Technical Support Application Note, Version 1.5. 4/97-GC0165.
- Peszek, I., Song, C., and Kuznetsova, O. (1999), Producing Tabular Reports in SAS® System in the Form of MS Word® Tables. Proceedings of the PharmaSUG 99 Conference, 298-293.
- SAS Institute (1999), The Complete Guide to the SAS Output Delivery System. SAS V8 Online Documentation.
- Wehr, P. (1996), %Print Drivers: Teaching SAS to Speak the Many Languages of Document Publication. Proceedings of the 21st Annual SAS® User's Group International Conference.

Yam, A. L. (2000), SAS® Software and Microsoft Office Visual Basic for Applications Make Beautiful Reports Together. Proceedings of the 25th Annual SAS® User's Group International Conference.

Zhou, J. (2001), From SAS® ASCII File to Word Document – A SAS Macro Approach. Proceedings of the PharmaSUG 2001 Conference.

Acknowledgments

The author would like to take the opportunity to give his special thanks to Monica Mattson for her encouragement and support during the course of the OGS development. Many thanks are also extended to Susan Kenny, Michael Litzsinger, Thang Tran, and Jasmin Fredette for testing the OGS and providing valuable feedback. The author would like to acknowledge Dawn DuBois, Lori Griffin, Monica Mattson, and John Morrill for reviewing this manuscript and providing valuable comments.

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

Contact Information

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

Jay Zhou
Quintiles, Inc.
10245 Hickman Mills Drive
Kansas City, MO 64137
Work Phone: 816-767-6768
Fax: 816-767-7347
Email: jay.zhou@quintiles.com
zhou_jay@yahoo.com