



ABSTRACT

One of the main requirements in today's business world is the ability to present information in a format that can be understood by anyone from the most technically minded all the way up to the senior decision maker in the company. SAS® as we already know, provides us with the capability to aggregate, summarise and present data in a variety of different structures and formats.

This paper investigates a selection of new features available in SAS 9, which can be used to customise and improve the appearance of your reports using the Output Delivery System (ODS).

Using the Output Delivery System, we will look at the new ODS DOCUMENT destination that can be used to collect and store procedural output in a SAS Item Store. We will also look at how to manage the content of our documents and re-create outputs without physically needing to regenerate the results using a new base SAS procedure, PROC DOCUMENT.

Finally, we will take a look at some powerful new additions to the ODS RTF and ODS PDF destinations and will examine some of the benefits in report formatting that can be achieved. This paper will be relevant to programmers and analysts responsible for producing reports.

INTRODUCTION

The Output Delivery System provides us with a vast amount of flexibility when it comes to creating reports. In fact, it's hard to envisage how we as programmers and analysts actually survived without it. With the introduction of SAS V8 came the ability to harness the power of the Output Delivery System to stylise our outputs from within the SAS environment and a world of flexibility was opened up to SAS programmers and report consumers alike. And so following such a "giant step for style kind", it's hard to imagine how SAS can improve on this with the release of SAS 9 and so we find ourselves asking the question; what has SAS 9 got to offer?

On the surface, it would be very easy to dismiss some of the new features that have been introduced in SAS 9 ODS as minor changes and enhancements that support existing functionality. For example, the ability to direct outputs to a wider choice of output destinations. In order to appreciate some of the strides forward that have been made this paper will analyse a fairly comprehensive (and hopefully familiar) reporting scenario where a series of results need to be submitted and distributed to a number of different forums. Imagine for example results need to be submitted internally within the organisation and for publication in a journal. In each case the same data source will be analysed but a different set of results will be distributed and in different formats dependent on the audience.

The first problem to be addressed is how to manage both sets of results given that there will be a certain degree of overlap between them. We could save our code as two separate programs but that would mean maintaining duplicate versions of the same program, which could raise maintainability issues moving forward. Alternatively we could redirect selected outputs directly to the desired ODS destination, but what happens if further requirements surface in the future? For example, if we need to re-create outputs in a third or fourth format? There are also potential resource issues to be considered if we think about the time utilised generating new results every time we want to create a report in a different format. SAS 9 provides us with a much more efficient solution to this problem. The solution involves using an ODS destination called ODS DOCUMENT.

ORGANISING YOUR OUTPUT

ODS Documents provide the ability to store output objects produced by procedural code in a hierarchical file structure called a SAS Item Store. SAS Item Stores are not new to the SAS system and will be familiar to those who have ever worked with templates and styles.

The first thing of note is that an ODS DOCUMENT is in fact an ODS destination and so output objects can be re-directed to it in a similar fashion to any other destination, such as ODS RTF.

The key difference here is that ODS DOCUMENT provides the capability to save output objects in a raw format which is independent of other supported ODS formats. This has two advantages:

- (1) By storing the results in a raw format the programmer is not forced to make a decision regarding the format of the results until they are ready to produce the report. This means they can potentially make use of new formats supported by future releases of SAS. This is possible because ODS DOCUMENT effectively treats both the storage and reporting of our data as independent processes.
- (2) Any of the outputs stored within an ODS Document can be re-created in any available ODS destination without needing to re-submit the SAS procedure that created them in the first place. This can be achieved using a new procedure called PROC DOCUMENT. This is possible because all of the output objects contained within the Document are stored in a neutral format.

The following code displays how a series of output objects produced by a variety of different procedures can be re-directed to a "Document store" using the ODS DOCUMENT statement.

```
ods listing close;
ods document name=out.bank_account_details (write) catalog=out.gseg;

  title 'Cross Tabulation of Account Details';
  footnote "Report produced on: &sysdate. &systemtime.";
  proc tabulate data=acc_merge;

  title 'Summary of Account Details';
  proc summary data=acc_merge mean std min max print;

  title 'Frequency Table of Account Details';
  proc freq data=acc_merge;

  title 'Listing of distribution of account balance';
  proc univariate data=acc_merge;

  goptions device=activex gsfmode=replace ctext=red
           ftitle=simplex htitle=2 ftext=simplex;
  title 'Bar Chart of Account Balances';
  proc gchart data=acc_merge;

ods document close;
ods listing;
```

As with all ODS destinations both the name and location of the document need to be identified when directing output to that destination. A two level name should be used when working with Document stores, as they are a type of SAS object. It is also possible to define the level of access to be used when working with the Document store, which in this case will be WRITE as a new Document is being created.

The CATALOG option is used to identify the location of all Graphical Segments entries (i.e. GRSEG entries or graphs) within the Document. This is important as ODS Documents will not store these output objects internally by default. What it will store are pointers to each GRSEG entry available within the

permanent catalog, identified using the CATALOG (CAT) option (Hence be careful when writing items to the Work library as they will only be available for the duration of the session).

There is an exception to the rule, namely if your graphs have been generated using a GOPTIONS device of JAVA, JAVAIMG, ACTIVEX or ACTXIMG within SAS 9.1.3 or a later release. If this is the case, all of your graph output objects WILL be embedded within the Document.

ODS Documents are capable of storing the following types of output object:

- Tables
- Graphs
- Equations
- Notes

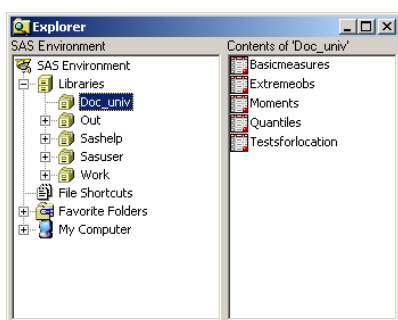
There are some caveats to the above. Firstly, not all SAS procedures are supported by ODS Document. PROC PRINT for example only has partial support in SAS 9.1 and PROC REPORT is not supported at all. Full support for these objects is planned for SAS 9.2. There are also issues to be aware of regarding metadata associated with each output object, which will be discussed later.

ODS Documents, as well as being able to contain output objects, may also contain sub-directories and links (or short-cuts) to other entries. The management of links will not be discussed within the scope of this paper but we will discuss how to influence both the structure of a Document and the output objects contained.

The first thing to be considered is how to access the output objects, which have just been directed to the new Document store. Each of those output objects will have been stored in a default directory structure.

You can assign a library to point to your new Document store using a LIBNAME statement, specifically using the SASDOC engine.

```
libname doc_univ sasdoc
        "\out.bank_account_details\Univariate#1\balance#1\C#1";
```



The only drawback to this is that you cannot assign a library to the top level of the Document store and hence explore the hierarchical structure. In order to use the SASDOC engine effectively, the LIBREF should reference a sub-directory within the Document store, which contains the actual output objects. In order to do this you obviously need to understand the default structure of the Document you've just created.

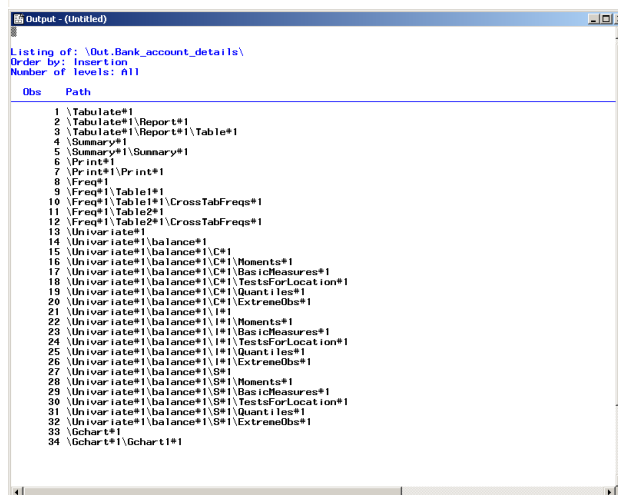
Perhaps a far more powerful method to explore and manipulate your new Document is PROC DOCUMENT.

PROC DOCUMENT can be used to perform a range of Document management functions. Its worthwhile knowing that the DOCUMENT procedure does have a Graphical User Interface, which can be used to perform exactly the same tasks as using the procedure itself but this paper will focus on using the procedural method.

The LIST statement can be used to view the contents of the Document store by querying the structure either from the root directory down or from any specified point.

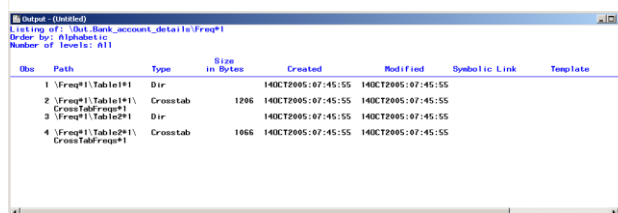
The LEVELS=ALL option can also be used to view the entire contents of the Document Store working off the root directory.

```
proc document name=out.bank_account_details;
    list/levels=all;
run;
quit;
```



The LIST statement can also be used to view a specific sub-directory tree and the output objects contained. The use of additional options can also help to control the order and content of the resulting output.

```
proc document name=out.bank_account_details;
    list \out.bank_account_details\Freq#1,
        \out.bank_account_details\Univariate#1
        / levels=all order=ALPHA details;
run;
quit;
```

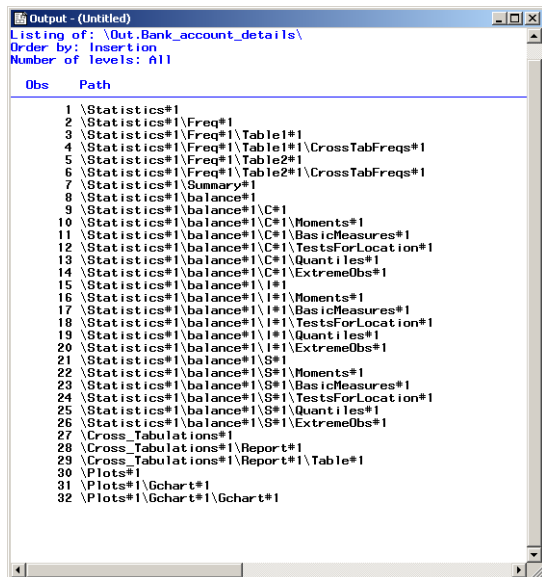


PROC DOCUMENT can also be used to change the default directory structure used within the Document store. It is possible, for example, to create sub-directories at any level under the root as demonstrated in the following example:

```
proc document name=out.bank_account_details(update);
    make Statistics;
    make Cross_Tabulations;
    make Plots;
run;
setlabel Statistics      'PROC FREQ and SUMMARY output';
setlabel Cross_Tabulations 'PROC TABULATE output';
setlabel Plots          'PROC GCHART output';
run;
```

Furthermore, once the desired directory structure has been created the programmer can then choose to copy, move, rename, duplicate and delete entries in order to update the structure and contents of the new Document store. The DIR statement provides the ability to change directory within the Document Store in order to manipulate the entries contained.

```
proc document name=out.bank_account_details(update);
  dir Statistics;
  move \Freq#1 to ^ /levels=ALL;
  move \Summary#1\Summary#1 to ^ /levels=ALL;
  copy \Univariate#1\balance#1 to ^ /levels=ALL;
  dir ^^;
  dir Cross_Tabulations;
  move \Tabulate#1\Report#1 to ^;
  dir ^^;
  dir Plots;
  move \Gchart#1 to ^;
  dir ^^;
run;
  dir ^^;
  delete \Tabulate#1;
  delete \Summary#1;
  delete \Univariate#1;
run;
quit;
```



It is worth knowing that you can actually create entries in a Document (whether they be output objects, directories or links) with exactly the same name within the same level of the Document store. This is possible because each entry is identifiable by its sequence number, which will be unique within a given directory.

Once outputs have been stored in a structure which best suits our particular business needs we are now ready to create our outputs in any of the available ODS destinations.

REPLAYING ISN'T JUST FOR GRAPHS

Now that the output has been stored in a Document it can now be re-created in a destination of our choice. Here in lies the power of the ODS DOCUMENT destination.

In the following example selected outputs as stored in the Document are replayed to an RTF destination. Note: In doing so the procedures that created these outputs will not be re-submitted thus eliminating the overhead of repeatedly processing the same data.

There are a few things to be aware of here. As we know an output object consists of a Data component and a Template component. The purpose of the Document procedure is to primarily store the data component. The presentation of that output object isn't entirely within the domain of the Document destination.

The good news is that if certain attributes like titles, footnotes, notes and by-grouping were in effect at the time when the output object was added to the Document, they will be automatically retained. This doesn't necessarily mean that this functionality is restrictive. You can for example choose to adopt current attributes active in your SAS session at the point where you

render your outputs. Not only that, using PROC DOCUMENT you can define attributes in conjunction with using the REPLAY statement as demonstrated in the following example.

The OBTITLE statement allows us to dynamically define titles at the point of reporting whilst still making use of TITLES stored with the output object itself. If OBTITLE, OBSTITLE or OBFOOTN redefines information currently stored with the output object then this information will be used at the point of reporting. The ACTIVETITLE or ACTIVEFOOTN options can also be used on the REPLAY statement to use titles and footnotes defined in the current session.

```
ods rtf file='D:\Papers\ODS Reporting In For Business\Results.rtf';
proc document name=out.bank_account_details(read);
  obtitle2 \Statistics#1\Freq#1\Table1#1\CrossTabFreqs#1
  'Title for First Frequency table';
  replay \Statistics#1\Freq#1\Table1#1\CrossTabFreqs#1;
run;
  replay \Statistics#1\Freq#1\Table2#1\CrossTabFreqs#1;
  obtitle2 \Statistics#1\Freq#1\Table2#1\CrossTabFreqs#1
  'Title for Second Frequency table';
run;
quit;
ods rtf close;
```

Frequency Table of Account Details
Title for First Frequency table

(The FREQ Procedure)

Table of account_type by demog_cat		demog_cat			Total
		A	B	C	
Current Account	Frequency	5	4	1	10
	Percent	20.00	16.00	4.00	40.00
	Row Pct	30.00	40.00	10.00	.
	Col Pct	41.67	50.00	20.00	.
High Interest Account	Frequency	4	2	1	7
	Percent	16.00	8.00	4.00	28.00
	Row Pct	57.14	28.57	14.29	.
	Col Pct	33.33	25.00	20.00	.
Savings Account	Frequency	3	2	3	8
	Percent	12.00	8.00	12.00	32.00
	Row Pct	37.50	25.00	37.50	.
	Col Pct	25.00	25.00	60.00	.
Total	Frequency	12	8	5	25
	Percent	48.00	32.00	20.00	100.00

Frequency Table of Account Details
Title for Second Frequency table

(The FREQ Procedure)

Table of account_type by gender		gender		Total
		Female	Male	
Current Account	Frequency	3	7	10
	Percent	12.00	28.00	40.00
	Row Pct	30.00	70.00	.
	Col Pct	30.00	46.67	.
High Interest Account	Frequency	2	5	7
	Percent	8.00	20.00	28.00
	Row Pct	28.57	71.43	.
	Col Pct	20.00	33.33	.
Savings Account	Frequency	5	3	8
	Percent	20.00	12.00	32.00
	Row Pct	62.50	37.50	.
	Col Pct	50.00	20.00	.
Total	Frequency	10	15	25
	Percent	40.00	60.00	100.00

It follows that if it is possible to render output objects in one particular format using PROC DOCUMENT, they can be rendered in any number of available formats. The following code demonstrates replaying output objects in both RTF and PDF format.



```
proc document name=out.bank_account_details(read);
ods rtf file='D:\Papers\ODS Reporting In For Business\Results_2.rtf';
replay \Statistics#1\Freq#1\Table1#1\CrossTabFreqs#1;
replay \Statistics#1\Freq#1\Table2#1\CrossTabFreqs#1;
run;
ods rtf close;
run;
ods pdf file='D:\Papers\ODS Reporting In For Business\Results_2.pdf';
replay \Statistics#1\Summary#1;
run;
ods pdf close;
quit;
```

```
ods noproctitle;
ods usegopt;

ods html body='D:\Papers\ODS Reporting In For Business\Results.html'
style=statdoc;

proc document name=out.bank_account_details(read);
replay \Statistics#1\Freq#1\Table1#1\CrossTabFreqs#1;
run;
title 'PROC UNIVARIATE listing for Account Balance';
replay \Statistics#1\balance#1\C#1 / activetitle;
run;
replay \Plots#1\Gchart#1\Gchart#1;
run;
quit;

ods html close;
```

Frequency Table of Account Details

The FREQ Procedure

Table of account_type by demog_cat		demog_cat			
		A	B	C	Total
Current Account	Frequency	5	4	1	10
	Percent	20.00	16.00	4.00	40.00
	Row Pct	50.00	40.00	10.00	.
	Col Pct	41.67	50.00	20.00	.
High Interest Account	Frequency	4	2	1	7
	Percent	16.00	8.00	4.00	28.00
	Row Pct	57.14	28.57	14.29	.
	Col Pct	33.33	25.00	20.00	.
Savings Account	Frequency	3	2	3	8
	Percent	12.00	8.00	12.00	32.00
	Row Pct	37.50	25.00	37.50	.
	Col Pct	25.00	25.00	60.00	.
Total	Frequency	12	8	5	25
	Percent	48.00	32.00	20.00	100.00

Frequency Table of Account Details

Table of account_type by demog_cat		demog_cat			Total
		A	B	C	
Current Account	Frequency	5	4	1	10
	Percent	20.00	16.00	4.00	40.00
	Row Pct	50.00	40.00	10.00	.
	Col Pct	41.67	50.00	20.00	.
High Interest Account	Frequency	4	2	1	7
	Percent	16.00	8.00	4.00	28.00
	Row Pct	57.14	28.57	14.29	.
	Col Pct	33.33	25.00	20.00	.
Savings Account	Frequency	3	2	3	8
	Percent	12.00	8.00	12.00	32.00
	Row Pct	37.50	25.00	37.50	.
	Col Pct	25.00	25.00	60.00	.
Total	Frequency	12	8	5	25
	Percent	48.00	32.00	20.00	100.00

Report produced on:14OCT05 10:32

Summary of Account Details

The SUMMARY Procedure

Analysis Variable = balance

account_type	demog_cat	N	Mean	Std Dev	Minimum	Maximum
Current Account	A	5	4550.00	4507.49	150.0000000	12000.00
	B	4	2242.50	1243.34	700.0000000	3450.00
	C	1	1100.00	.	1100.00	1100.00
High Interest Account	A	4	5250.00	4682.95	1800.00	12000.00
	B	2	350.0000000	424.2640687	50.0000000	650.0000000
	C	1	240.0000000	.	240.0000000	240.0000000
Savings Account	A	3	643.3333333	329.5957119	350.0000000	1000.00
	B	2	127.5000000	144.9568901	25.0000000	230.0000000
	C	3	1113.33	941.5589909	540.0000000	2200.00

STANDARDISING YOUR OUTPUT

We've seen how we can use the ODS Document destination to manage, store and ultimately create our outputs using a centralised Document store. What other features are available in SAS 9 that will help to standardise the reports that we generate?

Let us start by considering the use of another destination HTML, which may be appropriate to use to distribute results internally via an intranet. In the following example the output generated by PROC FREQ, PROC UNIVARIATE as well as a vertical bar chart created using PROC GCHART needs to be published. The ODS USEGOPT can be used to standardise the appearance of the text used within the graph with the text used within the titles and footnotes of the statistical report generated. This is particularly useful as SAS/GRAPH has a wide range of software and hardware fonts available and allows us to customise the fonts characteristics, which can now be used to provide a consistent and professional presentation of results.

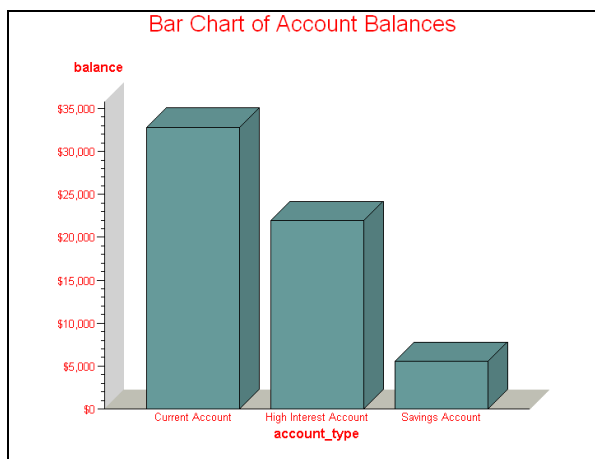
Note the use of ODS NOPROCTITLE to suppress automatic headings generated by certain SAS procedures within their output.

PROC UNIVARIATE listing for Account Balance

Variable: balance
account_type = C

Moments			
N	10	Sum Weights	10
Mean	3262	Sum Observations	32620
Std Deviation	3383.45911	Variance	11447795.6
Skewness	2.17204221	Kurtosis	5.57589891
Uncorrected SS	210745400	Corrected SS	103030160
Coeff Variation	103.091381	Std Error Mean	1069.94372

Basic Statistical Measures			
Location		Variability	
Mean	3262.000	Std Deviation	3383
Median	2800.000	Variance	11447796
Mode	.	Range	11850
		Interquartile Range	2350



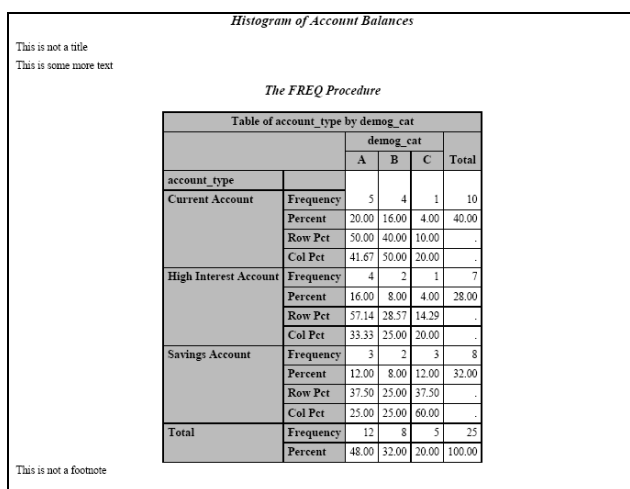
There are drawbacks to using this approach. If for example you use the OBTITLE, OBSTITLE or OBFOTN statements within PROC DOCUMENT, any GOPTIONS defined will not be applied to this information. They will only be applied to either active or stored titles or footnotes.

ENHANCING YOUR OUTPUT

Let us have a look at a few further techniques for enhancing the appearance of our outputs.

The ODS PDF and ODS RTF destinations now support the ability to embed text around procedural output. This is obviously a very powerful feature as it provides an additional mechanism for adding descriptive text outside of the boundaries of titles and footnotes.

```
ods listing close;
ods pdf file='C:\ian\Results.pdf';
ods pdf text='This is not a title' startpage=no ;
ods pdf text=' ' ;
ods pdf text='This is some more text';
proc document name=out.bank_account_details(read);
  replay \Statistics#1\Freq#1\Table1#1\CrossTabFreqs#1;
run;
quit;
ods pdf text='This is not a footnote';
ods pdf close;
ods listing;
```



The good news is there does not appear to be a limit to the amount of text that can be added using this technique. The capability to justify this text or to specify the font characteristics is not available using the text= option in this release but should be possible using in-line formatting together with escape characters.

ODS RTF and ODS PDF also support a more convenient means

for defining cell padding within reports. Previously this was only achievable by using existing styles or by creating and applying custom style definitions.

Using STYLE options, which are supported in PROC PRINT, PROC TABULATE, PROC REPORT, it is now possible to influence the indentation depth applied to values within table cells using the INDENT option. The LEFTMARGIN, RIGHTMARGIN, TOPMARGIN or BOTTOMMARGIN options can also be used to control the size of any of the available margins.

```
ods pdf file='D:\Papers\ODS Reporting In For Business\Results.pdf';
title 'Listing of Accounts data on the Book';
proc print data=acc_sort label noobs;
  var demog_cat account_no balance;
  / style=(font_weight=bold rightmargin=12pt
  indent=5pt topmargin=3pt);
  by account_type;
  id account_type;
  sum balance;
  label account_type = 'Account Type'
  demog_cat = 'Demog Category'
  account_no = 'Acc No.'
  balance = 'Balance (£)';
  format account_type $ac_type. gender $gender. balance pound.;
run;
ods pdf close;
```

Notice the indentation visible in both the character and numeric columns in the resulting output.

Account Type	Demog Category	Acc No.	Balance (£)
Current Account	A	000A001	£5,000.00
	A	000A003	£12,000.00
	A	000A007	£150.00
	A	000C002	£2,600.00
	A	000C006	£3,000.00
	B	000A004	£3,020.00
High Interest Account	B	000A012	£700.00
	B	000B002	£3,450.00
	B	000B005	£1,800.00
Savings Account	C	000C005	£1,100.00
			£32,820.00
High Interest Account	A	000B001	£12,000.00
	A	000C001	£2,400.00
	A	000C003	£1,800.00
	A	000C007	£4,800.00
	B	000A011	£50.00
	B	000C008	£650.00
	C	000A006	£240.00
Savings Account			£21,940.00
	A	000A002	£350.00
	A	000A009	£1,000.00
Savings Account	A	000B003	£580.00
			£1,930.00
			£56,690.00

RICHER FORMATTED RTF FILES

There are some useful new options available with the ODS RTF destination in SAS 9. Using STYLE options, as mentioned above, with the main Base SAS reporting procedures it is now possible to align numeric columns based on the position of the decimal point.

```
ods rtf file='D:\Papers\ODS Reporting In For Business\results.rtf';
proc print data=values;
  var class value /style(COLUMN)={just=d};
  format value best14.6;
run;
ods rtf close;
```



Obs	class	value
1	A	12.3
2	B	34.245
3	C	3.22
4	D	0.1036
5	E	136262
6	F	3636.3
7	G	60

The next option is similarly popular because it addresses a problem that every SAS programmer has inevitably had to overcome in the process of constructing reports. The PAGEOF option provides us with a solution to the problem of generating page numbers within our RTF files, which could only previously be addressed in earlier versions of SAS by embedding complex RTF code directly into our SAS statements.

By including PAGEOF in conjunction with an escape character (which identifies the instruction that follows as being an RTF command) we can now easily include Page n of N information in our titles and footnotes.

```
ods rtf file='D:\Papers\ODS Reporting In For Business\pageno.rtf';
ods escapechar='\';

title 'Listing with Page Numbers' justify=right ' Page \{pageof}';
proc print data=acc_sort noobs label;
var demog_cat account_no balance;
format account_type $ac_type. gender $gender. balance pound.;
label account_type = 'Account Type'
demog_cat = 'Demog Category'
account_no = 'Acc No.'
balance = 'Balance (£)';

run;
ods rtf close;
```

Listing with Page Numbers			Page 1 of 1
Demog Category	Acc No.	Balance (£)	
A	000A001	£5,000.00	
A	000A003	£12,000.00	
A	000A007	£150.00	
A	000C002	£2,600.00	
A	000C006	£3,000.00	
B	000A004	£3,020.00	
B	000A012	£700.00	
B	000B002	£3,450.00	
B	000B005	£1,800.00	
C	000C005	£1,100.00	
A	000B001	£12,000.00	
A	000C001	£2,400.00	
A	000C003	£1,800.00	
A	000C007	£4,800.00	
B	000A011	£50.00	
B	000C008	£650.00	
C	000A006	£240.00	
A	000A002	£350.00	
A	000A009	£1,000.00	
A	000B003	£580.00	

CONCLUSION

SAS 9 provides a flexible and powerful suite of new options and destinations that go further than merely supplementing existing functionality.

The use of the new ODS Document destination will help to support a standard way of working by allowing a programmer to create outputs once and store them in a central Document item store which can be managed using a new procedure PROC DOCUMENT. Output objects can then be easily rendered in any available ODS destination without the additional overhead of re-submitting procedural code.

Additional benefits to the programmer are available through a wealth of new and exciting formatting features that have been introduced to ease the production of high quality reports.

The changes and enhancements that have been made in SAS 9 represent a confident stride forward in the world of report formatting. This is SAS 9 reporting in for business!

REFERENCES

SAS Online Documentation for SAS 9
<http://support.sas.com> ODS Document Resources

CONTACT INFORMATION

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

Author Name: Ian Amaranayake
Company: Amadeus Software Limited
Address: The Old School Hall, 11 Wesley Walk, Witney, Oxon OX28 6ZJ
Work Phone: +44 (0) 1993 848010
Email: Ian.Amaranayake@amadeus.co.uk
Web: www.amadeus.co.uk

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.