Creating a Compact Columnar Output with PROC REPORT

Walter R. Young, Wyeth, Collegeville, PA

ABSTRACT

In a compact columnar output, the maximum number of observations should be made to attractively fit either on the width of a single page or on the width of a minimum number of pages without losing any information. This is the neatest most heuristic way to present a data set. Various techniques using PROC REPORT will be shown to accomplish this. Everything was done in SAS® 8.2 batch using the NO WINDOWS (NOWD) option.

INTRODUCTION

You may wish to display a data set to:

- use yourself.
- show others at a meeting
- put in a standardized regulatory report.
- put in a publication

These are ranked in roughly increasing effort required to create the report. The first two reasons and proofreading can be adequately handled using a browser. If a hard copy is required, one would have to use either (1) PROC PRINT; (2) PROC REPORT with or without the Output Delivery System (ODS) STYLE option[2]; (3) ODS as a stand alone; or (4) PUT statement formatting through the use of a DATA _NULL_. PROC PRINT cannot be sufficiently customized. DATA _NULL_ can do anything but it is unduly complex to program. ODS, recommended for publications, allows one to use proportional fonts and color. But, for a standardized report, PROC REPORT, with a mastery of all applicable options, can rival a serious DATA _NULL_ step. The main thing that PROC REPORT cannot do is to wrap observations neatly within a page as the WRAP and NAMED options create a messy output. Compacting columns reduces the need for this desirable feature.

OBJECTIVE

The objective is to display wide data sets, given a specified line size, in a neat and heuristic manner. It is not to save paper. PROC REPORT options[1] not needed to accomplish this limited objective are not discussed. A recent clinical reason for making the report compact, is the FDA guideline of using 12 point fonts with 1" margins for tables in electronic submissions. This restricts you to 90 columns and 43 lines. For a narrow data set, you can use the BOX option to draw lines between columns or around cells. You can also use the number of PANELs and PSPACE (space between panels) options to minimize paper use by putting the data in newspaper like columns (see SPECIAL AND UNPRINTABLE CHARACTERS). Do not use these options for a wide data set. It is assumed, you know how to write a basic PROC REPORT with its options and the COLUMN (COL), DEFINE and BREAK statements

PROC PRINT

With a little trial and error, you can calculate the page size (PS) system option needed to create a columnar output by having it wrap at the number of rows in the data set and then edit this output by adding page breaks. This is not viable for a data set having variables with more observations than will fit on a single page, but it will create neat even columns. But, (1) there are no beautification options other than the ID and VAR statements; (2) wide columns are truncated at 128 for a Wyeth production standard line size of 132; and (3) one cannot control the space between columns.

DEFAULT PROC REPORT

This will create a columnar report and a limited number of variables will fit neatly on a page width. However, (1) rows are not identified for multiple pages; (2) spacing between columns is uneven; (3) column order is not optimum; and (4) columns with a width greater than 113 with a line size of 132 will result in an error message and no output will be generated. PROC REPORT was designed to run interactively and thus has some odd defaults.

- Labels are used as column headers and label words are split.
- Variables are output in position order.
- Space between columns is 2 including before the first column.
- There is no option to print observation numbers[3].
- MISSING option should always be used to print all data rows.
- If all variables are numeric and none are specified as DISPLAY, they are summed *instead* of listed.
- If a variable name, not in the data set, is listed in both the COL and DEFINE statements, there is no error as it is assumed that it was created in a COMPUTE statement.

Thus one cannot create a useful compact columnar output without using a fair number of PROC REPORT options as well as having a good understanding of these options. Note in PC SAS, you can make a report compact using the Windows Page Setup options[4].

JUSTIFICATION RULES WITHIN COLUMNS

- Default is right for numeric and left for character.
- Numerical values are right justified within their specified formats and these are justified within their specified widths.
- Character values are justified in width with leading blanks retained and trailing blanks eliminated.

SEPARATELY JUSTIFYING LABELS AND VALUES

Justification rules apply simultaneously to values and their labels. Thus, it requires some effort to separately justify the two. You usually wish to left or right justify the value and center the label. You can do this by putting a SPLIT character followed by the requisite number of blanks at the beginning and/or a macro variable containing an appropriate unprintable character(s) at the end of the label (e.g., %sysfunc(byte(160)) works). Also, you can put label characters over the space before the column by using RIGHT and SPACING=0 in the DEFINE statement for the variable.

SPLIT CHARACTER

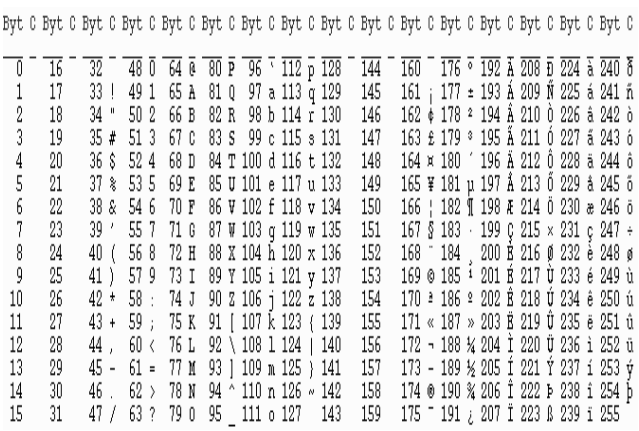
This splits both labels and variables having the FLOW option. You should pick a keyboard printable character less common than the default "/" since "/" is quite common in many entered texts. Possible choices are Wyeth's standard of |, ~, \ or `. If you are paranoid about using any keyboard character as the possibility always exists that it may occur in your data, you may use an obscure non-keyboard printable character as shown in the next section. You may wish to indent concatenated flowed text by inserting a split character plus the desired number of spaces. The indentation won't be correct unless one also puts a split character at the end of the text. Normally, if a word's length in a flowed variable is greater than the variable's width, the word will split at that width. However, if there is a split character in the text, words at the end of the field will split randomly due to a SAS bug that will be fixed in a future SAS version. To fix this, you either widen the field to eliminate non-indented flow or use the macro in this paper that inserts SPLIT characters based on knowledge of the field's width.

SPECIAL AND UNPRINTABLE CHARACTERS

Hexadecimal A's and C's are line and page feeds and must be removed from flowed text to prevent unwanted splitting. You should also remove any other unprintable printer control characters that may exist in flowed variables. This is an uncommon problem that can happen with data coming from many sources (investigator comments). To find what characters should be removed for your host environment, run the following code in display manager:

```
data a;
     length f $1;
     do byt=0 to 255;
       f=byte(byt);
        output;
     end;
run;
proc report spacing=1 pspace=1 panels=16 ls=95 ps=20 nowd;
     col byt f;
     define byt/width=3 'Byt/__' spacing=0;
     define f/width=1 'C_';
run;
```

This will give you the SAS mapping for the 256 characters in a byte as can be seen in the following screen print of the output window:



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Byt	C	Byt	C	Byt	C	Byt	C	Byt	C	Byt	C	Byt	C	Byt	C	Byt	C	Byt	C	Byt	C	Byt	C	Byt	C	Byt	C	Byt	C
0		16		32		48	0	64	@	80	P	96	`	112	p	128		144		160		176	°	192	À	208	Ð	224	à	240	ð
1		17		33	!	49	1	65	A	81	Q	97	a	113	q	129		145		161	¡	177	±	193	Á	209	Ñ	225	á	241	ñ
2		18		34	"	50	2	66	B	82	R	98	b	114	r	130		146		162	¢	178	²	194	Â	210	Ò	226	â	242	ò
3		19		35	#	51	3	67	C	83	S	99	c	115	s	131		147		163	£	179	³	195	Ã	211	Ó	227	ã	243	ó
4		20		36	$	52	4	68	D	84	T	100	d	116	t	132		148		164	¤	180	´	196	Ä	212	Ô	228	ä	244	ô
5		21		37	%	53	5	69	E	85	U	101	e	117	u	133		149		165	¥	181	µ	197	Å	213	Õ	229	å	245	õ
6		22		38	&	54	6	70	F	86	V	102	f	118	v	134		150		166	¦	182	¶	198	Æ	214	Ö	230	æ	246	ö
7		23		39	'	55	7	71	G	87	W	103	g	119	w	135		151		167	§	183	·	199	Ç	215	×	231	ç	247	÷
8		24		40	(	56	8	72	H	88	X	104	h	120	x	136		152		168	¨	184	¸	200	È	216	Ø	232	è	248	ø
9		25		41	)	57	9	73	I	89	Y	105	i	121	y	137		153		169	©	185	¹	201	É	217	Ù	233	é	249	ù
10		26		42	*	58	:	74	J	90	Z	106	j	122	z	138		154		170	ª	186	º	202	Ê	218	Ú	234	ê	250	ú
11		27		43	+	59	;	75	K	91	[	107	k	123	{	139		155		171	«	187	»	203	Ë	219	Û	235	ë	251	û
12		28		44	,	60	<	76	L	92	\	108	l	124	|	140		156		172	¬	188	¼	204	Ì	220	Ü	236	ì	252	ü
13		29		45	-	61	=	77	M	93	]	109	m	125	}	141		157		173	-	189	½	205	Í	221	Ý	237	í	253	ý
14		30		46	.	62	>	78	N	94	^	110	n	126	~	142		158		174	®	190	¾	206	Î	222	Þ	238	î	254	þ
15		31		47	/	63	?	79	O	95	_	111	o	127		143		159		175	¯	191	¿	207	Ï	223	ß	239	ï	255	

Note than the LS, PS (which override the system options) and PANELS options were used to create a balanced square output. The 16 by 16 format is useful as FORMCHAR is expressed in hexadecimal. All of the non-blank characters are treated as printable by SAS, including the non-keyboard characters with byt>126. Save and print this output window. The keyboard characters will be the same for the screen and paper printouts. The characters to be excluded will cause page and line breaks, tabs and font changes. You may see many odd characters, replacing the above blanks. Note that while these blank replacing characters will print as the 12 FORMCHAR characters that PROC REPORT uses, they will be treated as blanks in SAS functions. In Wyeth's system, there are surprisingly few unprintable characters (blank values of C) in the second printout. You can get smiley and frowning faces, the suits on a deck of cards, arrows in four directions, check marks, as well as other odd symbols. The only 3 relevant to this paper are the (1st) BOX separator between columns; (2nd) HEADLINE character; and (13th) double overlining and underlining in BREAK lines. Figure 1 shows a HEADLINE of ± using formchar(2)='B1'x where B1 is the hexadecimal equivalent of 177 in the above table (column B row 1). Changing FORMCHAR from its default values and/or using the LINE statement in a COMPUTE block allow you to get interesting effects using both printable and unprintable characters in macro variables and/or text. It does not affect the SAS FORMCHAR system option. The below code gives the ¿ line in Figure 2: Note that your printer may not support all of these characters.

```
%let longline=%sysfunc(repeat(%sysfunc(byte(191)),131));
title7"&longline";
```

MINIMAL OPTIONS REQUIRED

- COL statement specifying the desired order of the variables (Analogous to a PROC PRINT VAR statement).
- BREAK statement to insert spaces between observations.
- HEADLINE, HEADSKIP or BREAK BEFORE <first order variable> statement to separate labels from the observations.
- Constant SPACING between the columns.
- Customize labels in the DEFINE statement or use variable names through the use of the system NOLABEL option.

The above gives you most of the features of using a PUT statement formatting through the use of a DATA _NULL_.

ID OPTION

If the variables do not fit in the designated line size, you must specify the ID option in the DEFINE statement for the last variable in the COL statement required to fully identify all observations. This will cause that variable as well as the preceding variables in the COL statement to print on the left of all pages of the output. This will not work if the width of any variable (1) exceeds the inherent PROC REPORT limit or (2) plus the width of the ID variables plus the spaces between columns exceed the line size. In this case the FLOW option must be used.

TO MAKE THE REPORT COMPACT

- Make every reasonable effort to limit the page width to one. (Many users find a width of more than one page undesirable.)
- Reduce the SPACING between columns to one.
- Remove space before first column (SPACING=0 in DEFINE).
- Do not alter any variable if proofreading.
- Drop unnecessary variables from COL statement (e.g. Meta variables such as the source case report form number).
- Drop variables having the same or missing values for all rows and consider putting them in a title, footnote or legend.
- For all columns, use the minimum possible width that will not truncate any data. Do not use the variable's default width.
- Sort the data by sensible variables having a fair number of rows for each combination in the BY statement. Use this BY in PROC REPORT.
- For data sets wider than a single page, pick the minimum number of ID variables to adequately identify all observations. Balance the width of the non-ID variables across pages so that each page has about the same width. This gives you room to put lines or multiple spaces between columns.
- Use PROC FREQ (or my %legend) to determine whether long variables can be coded and explain the code in a legend.
- Do not use the FLOW option unless necessary as it increases the number of lines per observation.
- Judiciously use the SPLIT character in a flowed variable to eliminate the random word splitting SAS error.
- Eliminate all but one of a group of variables that have a one to one relationship with each other.
- Since formats can alter variable widths, apply them prior to calculating column widths.
- Sensibly condense character variables.
- Edit variables without altering their meaning.
- Transfer meaning from a variable to its label.
- Use width to truncate a SAS format (e.g., width=13 and f=datetime18. will remove seconds from a datetime variable)
- Use the STYLE attribute, some of the six font parameters and ODS to eliminate the unattractiveness of monospace fonts. Decimal points may not line up and this is not the best approach for reports that must be standardized over all drugs.

ALPHANUMERIC VARIABLES

- Determine their maximum width in the data set.
- If a format increases this width, use that width.
- Consider removing any invariant prefixes or suffixes (e.g., leading zeroes in a patient number).
- If FLOW is required, consider the line size constraint, calculate the width plus spacing of all other variables and:

(1) For one FLOW variable, use its maximum calculated width.
(2) For multiple FLOW variables, determine how to best allocate their widths to minimize lines per observation.
(3) See if other data can be put on the added line(s) per observation (e.g., concatenate two variables with the SPLIT character between them and use the FLOW option to make them print under one another. If necessary, indent them.
(4) Do not make the field's width less than the width of the largest word in the text.

NUMERIC VARIABLES

- Determine their range, minimum and maximum values and whether they are integers, and then specify an appropriate format and decimal point. Avoid the use of the BEST format.
- Use the DEFINE statement NOZERO option to suppress printing of a column whose values are zero or missing.
- For date time variables, specify an appropriate compact format (e.g., MMDDYY6.). Separate date and time with the DATEPART and TIMEPART functions. If the time is missing for all observations, remove it from the report.
- Use ORDER=INTERNAL in the DEFINE statement to prevent dates from being sorted alphabetically as default is formatted.
- If integer with trailing zeroes (common in lab tests) change the units to reduce the field width.
- If its format makes it an alphanumeric variable, apply that format and treat it as though it was an alphanumeric variable.

TO MAKE A COMPACT REPORT MORE ATTRACTIVE

- Use ODS and a proportional font.
- Appropriately order the COL statement variables.
- Use neat and informative titles, footnotes and/or legends.
- Appropriately specify the ORDER option in the DEFINE statements for the initial variables in the COL statement.
- Consider using NOPRINT in a DEFINE statement to order the observations by a variable which is not printed.
- Use the BREAK BEFORE statement to put blank or solid lines between ID observations and separate the labels from the values. With the FORMCHAR option you can make a line of any character (e.g., ~) available in your host system.
- A LINE statement in a COMPUTE BLOCK will allow you to customize spacing lines.
- Use informative labels neatly spanned in the COL statement and appropriately split in the DEFINE statement. They should be designed with the column width in mind.
- Separately justify values and their labels as discussed.
- End all labels with a SPLIT character followed by two underline characters (i.e., __) to separately underline labels.
- If necessary, expand a label's meaning in a legend page.

ADVERSE EVENT LISTING EXAMPLE

Wyeth has highly sophisticated validated modules, with built in error checks, that create data sets for listing and summarization. The project programmer for the specified drug chooses the options for the module and then writes a report program. Alternatively, he could copy and modify a report from a template. The following is such a PROC REPORT program with the data having been sorted BY INVTEXT TPNAME (investigator and treatment name).

```
proc report nowd data=_report_ split='|' headline missing spacing=1 formchar(2)='B1'x;
    by invtext tpname;
    col patient agesex priryn incryn tese eventcls toxgr sev rel actn
        outc strtdate strttime stopdate stoptime fsaeyn abs_rel dai;
    define patient/left width=7 order order=internal "&sbpat" spacing=0;
    define agesex/left width=7 order order=internal "AGE(Y)/|SEX';
    define priryn/width=2 "P|R|I|O|R?";
    define incryn/width=2 "W|O|R|S|E?";
    define tese/width=3 "T.|E.?";
  define eventcls/left flow width=20 "BODY SYSTEM|  ADVERSE EVENT|    VERBATIM";
    define toxgr/left width=4 "NCI|TOX";
    define sev/left width=4 "SEV";
    define rel/left width=4 "DRUG|REL.";
    define actn/left flow width=4 "AC-|TION";
    define outc/left width=4 "OUT-|COME";
    define strtdate/left format=date9. width=9 "START|DATE";
    define strttime/left format=time5. width=7 "TIME|(24 hr)";
    define stopdate/left format=date9. width=9 "STOP|DATE";
    define stoptime/left format=time5. Width=7 "TIME|(24 hr)";
    define fsaeyn/left width=4 "SAE";
    define abs_rel/left width=6  "STUDY|DAY";
    define dai/left flow width=8 "DAI";
    break after patient/skip;
run;
```

At first glance, the first page of this output (with the titles, which are the same as Figure 2, stripped) in Figure 1 seems acceptable. The options SPACING=1 and SPACING=0 for the first column should yield a compact report and the above adds up exactly to a usage of 132 columns. There is a BY statement for investigator and treatment name and a DEFINE statement for every variable in the column statement. However, you should always carefully scrutinize an entire PROC REPORT output for problems.

(1) Labels are not neat. The space allocated for most columns was determined by their label length rather than by their data.
(2) The SAS error of random word splitting caused by the SPLIT character in the text occurs in MUSCULOSKELETAL SYST.
(3) The lines per observation are excessive and the indented variable flows to the beginning of the line. The times are blank
(4) Using DATE7 and no width would remove the date's century.
(5) SEV with only four different values, while sparse, has more than four characters. It should either be flowed or coded.
(6) ACTN was truncated in a data step to 3 characters, which limits you to one code as it had a space after the comma.
(7) OUTC is truncated to 3 characters but is not in the footnotes.
(8) An incorrect template caused DAI to be inadvertently left out of the call in the module. No error resulted as it was in the COL and DEFINE statements and thus was blank.
(9) The template also caused "VERBATIM" to be wrongly indented since its label had no split character at its end.
(10) There are only 15 different COSTART body system codes, 12 of which are used in this report. It thus makes sense to code them by their unique first two characters and put the code in a footnote rather than listing them for each observation.

INDENTATION MACRO

With reasonable values of the parameters, the following macro will (1) eliminate the SAS error; (2) give the correct indentation when an indented line flows; and (3) handle the largest word's width being larger than the field width as well as blank indented lines:

```
%macro ind(name,v=,indent=2,width=,o=,len=150,split=/);
%let i=0;
%do %while(%length(%scan(&v,%eval(&i+1),' ')));
  %let i=%eval(&i+1);
  %let var&i=%scan(&v,&i,' ');
  %let w&i=%eval(&width-&indent*%eval(&i-1));
%end;
data &name(drop=w1-w30 line i j k l s);
      length &o $&len;
      set &name;
      array w{30} $40;
      length i $7;
      &o=' ';
      i="&split";
      %do m=1 %to &i;
        j=0;
        do while(scan(&&var&m,j+1,' ')^=' ' & j<30);
          j=j+1;
          w{j}=scan(&&var&m,j,' ');
        end;
        line=0;
        do k=1 to j;
          s=&&w&m-line;
          l=length(w{k});
          if l+line<&&w&m then do;
            if line=0 then &o=trim(left(&o))||left(w{k});
            else &o=trim(left(&o))||' '||left(w{k});
            line=l+line+1;
          end;
          else if l<=&&w&m then do;
            if line=0 then &o=trim(left(&o))||left(w{k});
            else &o=trim(left(&o))||trim(i)||left(w{k});
            line=l;
          end;
          else do;
            if line=0 then &o=trim(left(&o))||left(substr(w{k},1,s))||i;
            else &o=trim(left(&o))||' '||left(substr(w{k},1,s))||i;
            &o=trim(left(&o))||substr(w{k},s+1);
            line=line+1+l-&&w&m;
          end;
        end;
        %do n=1 %to &indent;
          i=trim(i)||byte(174);
        %end;
        if j then &o=trim(left(&o))||i;
      %end;
      &o=translate(&o,' ',"%sysfunc(byte(174))");
run;
%mend ind;
```

The call to the adverse events data set is as follows:

```
%ind(_report_,v=aescx aeptx aex,width=20,o=eventcls,split=|)
```

Note byte(174), ®, which is a non-keyboard printable character, is used as a placeholder. One must specify the **name** of the data set, the order of the variables to be indented, **v**, the number of characters of the indentation, **indent**, the **width** of the PROC REPORT column, the name of the final variable, **o**, its length, **len**,

and the **split** character used. The macro creates another data step but the code could be changed so that it is inserted in an existing data step. This macro corrects the indentation and splitting of the 4th and 5th Figure 1 observations.

```
RESPIRATORY SYSTEM                RESPIRATORY SYSTEM
  COUGH INCREASED                   COUGH INCREASED
    Productive                        Productive
Cough                             Cough
MUSCULOSKELETAL SYST              MUSCULOSKELETAL
EM                                SYSTEM
  ARTHRALGIA                        ARTHRALGIA
    Pain - Hips                       Pain - Hips
```

REVISED ADVERSE EVENT LISTING

```
proc report nowd data=_report split='|' missing spacing=1;
     by invtext tpname;
     col pat_age visit('_Date_' visitdt strtdate stopdate)
          priryn incryn tese aescx v toxgr sev rel actn outc fsaeyn abs_rel dai;
    define pat_age/width=4 order "SUB-JECT||AGE|(Y)|/SEX|__"spacing=0 flow;
     define visit/width=2'VISIT#__'order;
     define visitdt/f=date7."Visit|__"center order order=internal;
     define strtdate/f=date7."Start|__"center order order=internal;
     define stopdate/f=date7."Stop|__"center;
     define priryn/width=1"PRIOR?_";
     define incryn/width=1"WORSE?_";
     define tese/width=1"T.E.?_";
     define bs/width=2"BODY||SYSTEM__";
     define v/flow width=54"ADVERSE EVENT| VERBATIM|__";
     define toxgr/width=1"NCITOX_;
     define sev/flow width=4"SEVERITY|__";
     define rel/width=4"DRUG||RELATIONSHIP|__";
     define actn/flow width=3"ACTION|__";
     define outc/width=3"OUTCOME|__"center;
     define fsaeyn/width=1"SAE_";
     define abs_rel/width=5 left"STUDY| DAY|__";
     define dai/flow width=8"Data|Analysis|Interval|__"center:;
     break after pat_age/skip;
run;
```

The new variables created and/or used for PROC REPORT are:

(1) v: adverse event concatenated with an indented verbatim.
(2) pat_age, which is the first two zeroes, stripped from patient that is then concatenated with agesex. This did not result in any extra output rows as v was already flowed.
(3) bs: body system - the first two characters of aescx.

Note from Figure 2 and the above code you can see:

(1) Labels are individually underlined by using a SPLIT character followed by 2 underlines (1 underline for a width of 1).
(2) COL statement labels are spanned and made more explicit.
(3) The maximum possible width was used for the verbatim This corrected all indenting problems and the SAS word split error.
(4) STUDY DAY value was left oriented. Putting a space after the SPLIT after STUDY centered the label.
(5) Observations were logically ordered by visit number and date.
(6) OUTC was added to the legend page.
(7) The order option was not used for stopdate and subsequent variables as it would be difficult to distinguish missing values from those that were identical to the previous observation.
(8) Superfluous label SPLIT character and options were removed
(9) Footnotes were replaced by a dynamic legend page

This reduced the output from 45 to 19 pages. Figure 2 shows data for a different patient to illustrate that the action code for an observation can have more than one value. Note that the first and third observations in Figure 1 are essentially replicates. While clinical data should not be edited, these replicates could be collapsed by (1) removing punctuation marks and case; (2) sorting the words in verbatim; (3) sorting the observations with the NODUPKEY option; and (4) outputting the first original verbatim.

DYNAMIC LEGEND PAGE

This does not directly reduce the width of a report but it reduces its page count over the use of footnotes repeated on every page. It must be used to explain (1) cryptic column headings created as a result of compacting the report and (2) the coded contents of a column. Creating the legend page dynamically (i.e., only showing the codes used in the report) allows you to check if any codes were inadvertently excluded from the legend page as well as eliminating codes which are extraneous to the study. The following is the dynamic legend page for Figure 2:

```
Action         C    96 Concomitant Medication
               H     2 Hospitalized
               N   189 None
               O     1 Other Action(s) Taken
Body System    **   14 Unknown
               BO   70 Body as a Whole
               CA    4 Cardiovascular System
               DI   75 Digestive System
               HE   16 Hemic & Lymphatic System
               ME   30 Metabolic & Nutritional
               MU   26 Musculoskeletal System
               NE   25 Nervous System
               RE   21 Respiratory System
               SK    7 Skin & Appendages
               SP    3 Special Senses
               UR    6 Urogenital System
Outcome        Dea   2 Death
               Per 176 Persisted
               Res 118 Resolved
Relationship   DEF   1 Definitely
               DNOT 186 Definitely Not
               PNOT  46 Probably Not
               POS   53 Possibly Not
               PRB   11 Probably
Severity       Life   1 Life Threatening
               Mild  28 Mild Threatening
               Mod    2 Moderate
               Sev    1 Severe
```

Normally, you would not put counts into a legend. However, since they automatically come out of PROC FREQ, you can see that they give some useful information such as two patients died and the severity data was quite sparse. The above dynamic legend page was created from the following master legend data set:

```
data legend;
     input category $ 1-12 abbrev $ 13-16 text $ 17-53 count 54;
     cards;
Action      C   Concomitant Medication
Action      D   Discontinued Test Article Permanently
Action      H   Hospitalized
Action      N   None
Action      O   Other Action(s) Taken
Action      P   Primary Reason for Study Withdrawal
Action      R   Reduced Test Article Dose
Action      T   Temporarily Stopped Test Article
Body System ** Unknown
Body System AD  Adverse Event Assoc with Misc Factors
Body System BO  Body as a Whole
Body System CA  Cardiovascular System
Body System DI  Digestive System
Body System EN  Endocrine System
Body System HE  Hemic & Lymphatic System
Body System ME  Metabolic & Nutritional
Body System MU  Musculoskeletal System
Body System NE  Nervous System
Body System RE  Respiratory System
Body System SK  Skin & Appendages
Body System SP  Special Senses
Body System TE  Terms Not Classifiable
Body System UR  Urogenital System
Outcome     Dea Death
Outcome     Per Persisted
Outcome     Res Resolved
RelationshipDEF Definitely
RelationshipDNOTDefinitely Not
RelationshipPNOTProbably Not
RelationshipPOS Possibly Not
RelationshipPRB Probably
Severity    LifeLife Threatening
```

```
Severity    MildMild Threatening
Severity    Mod Moderate
Severity    Sev Severe
run;
```

Then use %legend, specifying the data set and coded variables:

```
%macro legend(data,v=);
%let i=0;
%do %while(%length(%scan(&v,%eval(&i+1),' ')));
  %let i=%eval(&i+1);
  %let c&i=%scan(&v,&i,' ');
%end;
data a;
     set legend(obs=0 keep=abbrev count);
     length name $8;
run;
%do k=1 %to &i;
  proc freq data=&data(keep=&&c&k) noprint;
       tables &&c&k/out=b(drop=percent rename=(&&c&k=abbrev));
  run;
  data b;
       set b;
       length name $8;
       name="&&c&k";
  run;
  proc append base=a data=b force;
  run;
%end;
data a(keep=abbrev count where=(abbrev^=''));
     set a(where=(abbrev^=' '));
     j=abbrev;
     i=1;
     do until(scan(j,i,', ')='');
        abbrev=scan(j,i,', ');
        output;
        i=i+1;
     end;
run;
proc sort data=a nodupkey;
     by abbrev;
run;
proc sort data=legend nodupkey;
     by abbrev;
run;
data legend;
     merge legend a(in=i);
     by abbrev;
     if i;
     if category=' ' then category=trim(name)||'?';
run;
%mend legend;
```

Note that %legend will identify codes that are not in the master legend data set with the name of the variable in which they were found. Thus by using a null master legend data set with an appropriate **abbrev** width, you can make a decision as to whether or not it makes sense to code a variable as well as determine what its codes should be. Also, it handles the observation in Figure 2 that has two action codes as long as the width of **abbrev** in the master legend data set is as big as the largest concatenated actn. The width of the codes can be different as **FORCE** is used in PROC APPEND. Note a width need only be specified for count as the others are defaulted from the data step. This macro will handle any master data set if specified as above.

```
%legend(_report_,v=sev rel bs actn outc)
proc report nowd data=legend nocenter spacing=1;
     col category abbrev count text;
     define category/order 'Category/__'spacing=0;
     define abbrev/'Abbreviation/__'center;
     define count/' #/__'width=3;
     define text/'Text/__';
run;
```

GENERALIZED ADVERSE EVENTS LISTING

Considering Figure 2, there is some justification for writing a validated listing program that would handle all drugs and protocols. Other projects may require additional variables (e.g. visit name, dose of drug at onset, duration of event, study analysis interval, etc.) and may require different sort orders and BY variables. To write a general program you must determine every possible variable that can be used for any project and give the user the ability to specify variables and their order. For each variable:

(1) Specify its label attractively with SPLIT characters.
(2) Calculate its width if not constant over drugs or else specify it.
(3) Check if it is blank (e.g., time) or missing and, if so, exclude it.
(4) Maximize the width of the verbatim variable.
(5) Decide whether it should be flowed (e.g. ACTN and DAI).
(6) Change to alphanumeric, concatenate, SPLIT and flow the 3 dates and times, first changing the time to blank if it is zero. These would have to be ordered by the NOPRINT datetime variable. Then the times would print under the ordered dates.
(7) From Figure 2, you can see that this report should be able to fit on a single page width with perhaps a narrowed verbatim.

Let the user specify a predetermined list of sort orders within PROC REPORT and/or BY variables after sorting the data

ECGTEST EXAMPLE

Wyeth has standardized SAS views for many different types of data for all drugs and protocols. ECGTEST contains 57 variables. A drug and protocol was chosen that had 29 ECG observations with the objective of printing this view without losing any information in the width of a single page. First, the data was carefully examined. 37 of the variables (1) had either the same value for each observation; (2) were either blank or missing; or (3) were meta variables which contained no useful information. A compressed PROC CONTENTS of the remaining is given below:

Variable	Format	W	Label	Comment
ABS_REL		5	Prim Relative Day	Integer
COUNTRY	$3.	3	Country	Abbreviated
CPENM	$20.	16	Planned Event Name	1 to1 to visit
DAI	$30.	9	Data Analysis Interval	1 to 1 to dai_ord
DAI_ORD		1	DAI Order	1 to1 to dai
INVEST	$10.	5	Investigator	Uninformative
LVALC	$200.	16	answer char	related sasname
MILESTN	$30.	1	milestone name	7th char different.
PATIENT	$10.	6	patient id	4 leading 0's
QLVALUE	$70.	3	Label type qualifying	1:1 to visit
REGIMEN	$30.	5	Regimen	blank or tpname
RELDAY		5	time from milestone	Integer
SASNAME	$8.	6	sas name	1 to 1 to test
STDYSITE	$10.	3	Study site	Uninformative
STINT	$30.	28	planned study interval	
TEST	$20.	12	Question name	1 to 1 to sasname
TPCODE	$10.	1	Therapy code	1 to 1 to tpname
TPNAME	$30.	28	Therapy text	4th & 5th char diff
VISIT	6.	2	WAR number	1 to 1 to cpenm
VISITDT	DATETIME20.	6	Visit date/time	No times entered

(As an aside, note that PROC PRINT with a PS of 165, after adding page breaks, will yield a neat columnar output of these variables.) The maximum width, W, was calculated for each variable. While this is appreciably less that that of the formats, the total width is still too big to fit on a single page. The code for investigator and site is not informative. These two variables must be translated via a merge with a patient information view to the actual names. COUNTRY is abbreviated to DEU for "Deutchland" and ESP for "España". Replacing these with Germany and Spain would be more informative. There were only 3 investigators and two sites, each with rather long names, so I concatenated these variables with country into INV. Since times were not entered, the date can be reduced to a width of 6 using MMDDYY6. TPNAME has values of "EKB XXmg+FU/LV/IRINOT 180 mg" where XX is the only thing that varies from observation to observation. REGIMEN is either identical to this or blank. Thus, these two variables could be reduced to 2 characters plus an explanation in a footnote. I chose instead to (1) put "EKB" in the label, (2) leave the rest of the characters in the variable; and (3) let REGIMEN be either "Same" or blank. TPCODE, which had a one to one relationship to TPNAME, was dropped. The first 4 zeroes were stripped out of PATIENT, which was renamed to PT. To better understand this, look at some of the data from the first 4 patients:

P	LVALC	SASNAME	TEST	MILESTN	VI	CPENM
2	Normal	OVEVAL	OVER_EVAL	Cycle 1 Start	1	SCREENING
3	Normal	OVEVAL	OVER_EVAL	Cycle 4 Start	46	FINAL EVALUATION
3	No	CHGYN	CHANGE_YN	Cycle 4 Start	46	FINAL EVALUATION
4	NCS Abnormality	OVEVAL	OVER_EVAL	Cycle 1 Start	1	SCREENING
5	NCS Abnormality	OVEVAL	OVER_EVAL	Cycle 1 Start	1	FINAL EVALUATION
5	NCS Abnormality	OVEVAL	OVER_EVAL	Cycle 1 Start	46	FINAL EVALUATION
5	No	CHGYN	CHANGE_YN	Cycle 1 Start	46	SCREENING

It is obvious that SASNAME and TEST have a one to one relationship as do VISIT and CPENM (visit name). A close scrutiny of TEST shows that it either tells you whether the EKG was normal or whether there was a change from the baseline VISIT 1. Thus, the 15 character LVALC can be reduced to a one character Y or N. TEST is thus labeled EKG and given the six-character value of "Normal" or "Change". MILESTN can be reduced to, MILES, its seventh character. CPENM can be abbreviated to "Screen" and "Final" without losing meaning Figure 3, with 132 characters, was generated by the following data step and PROC REPORT code:

```
data test(drop=patient visitdt sasname lvalc cpenm regimen tpname milestn);
     set test(keep=country stdysite invest patient visitdt cpenm dai sasname
                   lvalc stint tpname abs_rel relday regimen milestn);
     length pt $2 ecg visit $6 yn miles $1 reg $4 tp $24 land $7;
     pt=substr(patient,5);
     tp=substr(tpname,5);
     miles=substr(milestn,7,1);
     date=datepart(visitdt);
     if sasname=:'O' then ecg='Normal';
     else ecg='Change';
     if lvalc=:'Nor' then yn='Y';
     else yn='N';
     if cpenm=:'S' then visit='Screen';
     else visit='Final';
     if regimen^=' ' then reg='Same';
     else reg=' ';
     if country='DEU' then land='Germany';
     else land='Spain';
run;

proc report data=test nowd spacing=1 missing split='~';
     col country inv pt ('Visit' date visit) ecg yn ('Relative Day' abs_rel relday)
          ('EKB Therapy          Regimen' tp reg) miles dai stint;
     break after inv/skip;
     define country/order noprint;
     define inv/order 'Investigator~__' width=22 flow;
     define pt/order 'Patient~__' width=4 center;
     define date/order 'Date~__' spacing=0 format=mmddyy6.;
     define visit/order 'Name~__';
     define ecg/'ECG~__';
     define yn/width=1 '?~_';
     define abs_rel/width=6 spacing=0 'Abs~__';
     define relday/width=6 spacing=0 'Miles~__';
     define tp/width=24 '__';
     define reg/width=4 '__';
     define miles/width=2 'Milestone~__' spacing=0 right;
     define dai/width=9 'Data~Analysis~Interval~__' center;
     define stint/width=28 'Planned Study Interval~__';
run;
```

The first variable in the COL and DEFINE statements is COUNTRY with the ORDER and NOPRINT options. This causes the data to be sorted by COUNTRY rather than alphabetically by investigator. Since, the first column is not printed, the SPACING=0 option is not needed as long as the ID option is not used. As was the case in the previous example, the ORDER variables are only printed if their value changes. TP and REG are labeled in the COL statement rather than in the DEFINE. For, illustrative purposes, SPACING=0 was used to label MILES over the space between columns as was the spanned "Relative Day". The 2 character PT was given a 4 character label.

GENERALIZED ECGTEST COMPACT REPORT

It should be reiterated that this example, unlike the previous one, is presented only for illustrative purposes. Still, many of the considerations for generalizing the adverse events listing are valid. To create code for listing ECGTEST for all drugs and protocols:

- Decide what variables to always exclude.
- Decide on which constant variables belong in the title.
- Decide on what one to one variables to include.
- Count and remove applicable leading zeroes.
- Determine which data condensing tricks work for all cases.
- Predetermine which variables need FLOW.
- Determine which variables have a fixed width.
- Calculate the width of all remaining variables.

It is unlikely that the data can be squeezed into the width of a single page for all ECGTEST views. You can get more space to list the variables by either using the ID option or using a BY statement for the investigator. In this example, the ID option would list all data on two page widths while the BY statement would require three page widths. Since the BY statement was already illustrated in the adverse events example, let us consider the ID option. If you (1) add the NOCENTER option to the PROC REPORT; (2) increase the width of INV to 31 (so that investigator only prints on two lines) and add the SPACING=0 option; and (3) add the ID option to VISIT, you get the output in Figure 4. (The title and labels, which are identical to those in Figure 3, were stripped and only the first ten observations of the second page included to save space). This output is not balanced in that the second page has a lesser width than the first page. However, it is assumed that the reason for adding the ID option was to widen and add variables with the output still fitting neatly on two pages. There is sufficient room to do this and still put additional spaces between the columns to balance the two page widths.

CONCLUSION

It does not take an undue amount of effort to create an attractive compact columnar output using PROC REPORT. It does take a reasonable knowledge of the options, but this knowledge is not particularly difficult to acquire and definitely worth the effort. You should never generate a report by copying it from a template without understanding the basic principles of PROC REPORT.

REFERENCES

(1) SAS Procedures Guide, Chapter 32, The Report Procedure; pp 859 -1004 SAS OnlineDOC® Version 8 with PDF Files.

(2) Pass, Ray and McNeill, Sandy SUGI 15-28 (2003) "PROC REPORT: Doin' It in Style!"

(3) Pass, Ray, NESUG (1999) "PROC REPORT - Land of the Missing OBS Column".

(4) Hjelle, Dean, support.sas.com (12/00) "Quick Tip: Fitting More Variables on a Page"

CONTACT INFORMATION

Walter Young
Wyeth
PO Box 42528
Philadelphia, PA 19101-2528
(484) 865-5640
youngw@wyeth.com

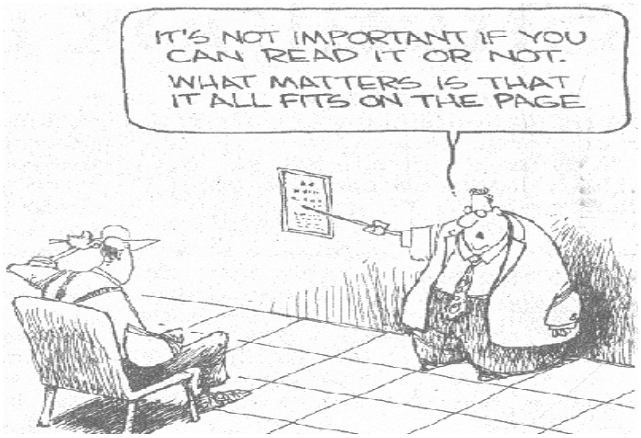


Figure 1: Original Adverse Event Listing Stripped of Titles

SUBJECT	AGE(Y)/ SEX	PR?	WOR?	ST?	T.E.?	BODY SYSTEM ADVERSE EVENT VERBATIM	NCI TOX	SEV	DRUG REL.	ACTION	OUTCOME	START DATE	TIME (24 hr)	STOP DATE	TIME (24 hr)	SAE	STUDY DAY	DAI
001001	69/M	Y	N	N		DIGESTIVE SYSTEM PAIN RECTAL Pain - Rectal	1		DNOT	C	Per	01APR2002					-180	
		Y	N	N		DIGESTIVE SYSTEM PAIN RECTAL Pain - rectal	1		DNOT	C	Res	01APR2002		03JAN2003			-180	
		Y	N	N		DIGESTIVE SYSTEM PAIN RECTAL Rectal pain	1		DNOT	C	Per	01APR2002					-180	
		Y	N	N		RESPIRATORY SYSTEM COUGH INCREASED Productive cough	1		DNOT	N	Res	03SEP2002		06OCT2002			-25	
		Y	N	N		MUSCULOSKELETAL SYSTEM ARTHRALGIA Pain - Hips	1		DNOT	C	Per	13SEP2002					-15	
		Y	N	N		MUSCULOSKELETAL SYSTEM ARTHRALGIA Pain - hips	1		DNOT	C	Res	13SEP2002		03JAN2003			-15	
		Y	N	N		MUSCULOSKELETAL SYSTEM ARTHRALGIA Pain right hip	1		DNOT	C	Per	13SEP2002					-15	
		Y	N	N		HEMIC AND LYMPHATIC SYSTEM ANEMIA Anemia	1		DNOT	N	Per	24SEP2002					-4	

ACTION CODES:
N - NONE
T - TEMPORARILY STOPPED TEST ARTICLE
R - REDUCED TEST ARTICLE DOSE
D - DISCONTINUED TEST ARTICLE PERMANENTLY
H - HOSPITALIZED
C - CONCOMITANT MEDICATION
P - PRIMARY REASON FOR STUDY WITHDRAWAL
O - OTHER ACTION(S) TAKEN

DRUG REL.:
DEF - DEFINITELY
PNOT - PROBABLY NOT
DNOT - DEFINITELY NOT
PRB - PROBABLY
POS - POSSIBLY

Figure 2: Final Adverse Event Listing

REPORT AE1 LISTING OF ADVERSE EVENTS

INVESTIGATOR: 001, USA, Doe, John TREATMENT: Ascending Dose 37mg/m2

SUBJECT	AGE (Y) /SEX	VI SI T#	Visit	Date Start	Date Stop	PR?	WOR?	ST?	BODY SYSTEM	ADVERSE EVENT VERBATIM	NCI TOX	SEV	DRUG REL.	ACTION	OUTCOME	SAE	STUDY DAY	Data Analysis Interval
1006	63/M	2	04DEC02	01MAR02		Y	N	N	BO	ABDOMINAL PAIN Abdominal pain	1		DNOT	C	Per		-279	baseline
		9	26DEC02	04DEC02	04DEC02	N		N	DI	ESOPHAGITIS Burning of alimentary tract	1		DEF	N	Res		-1	baseline
			21DEC02	22DEC02		N		Y	UR	URINARY TRACT DISORDER Ureteral Inflammation	3		DNOT	C	Res		17	cycle 1
			26DEC02			N		Y	BO	BACK PAIN Low back pain	2		DNOT	C	Per		22	cycle 1
		16	16JAN03	01SEP02		Y	N	N	DI	ANOREXIA Early Satiety	1		DNOT	N	Per		-95	baseline
					11JAN03	Y	N	N	DI	FLATULENCE Bloating	1		DNOT	N	Res		-95	baseline
			26DEC02	28DEC02		N		Y	BO	BACK PAIN Low back pain / right flank	2		DNOT	C	Res		22	cycle 1
			28DEC02	29DEC02		N		Y	BO	BACK PAIN Low back pain / right flank	3		DNOT	C	Res		24	cycle 2
			29DEC02	05JAN03		N		Y	BO	BACK PAIN Low back pain / right flank	2		DNOT	C	Res		25	cycle 2
			03JAN03	11JAN03		N		Y	DI	CONSTIPATION Constipation	2		POS	C	Res		30	cycle 2
			11JAN03	12JAN03		N		Y	DI	FECAL IMPACTION Fecal Impaction	3		DNOT	H,O	Res	Y	38	cycle 2
					12JAN03	N		Y	DI	FECAL IMPACTION Fecal impaction	3		PNOT	C	Res		38	cycle 2
			12JAN03	15JAN03		N		Y	DI	CONSTIPATION Constipation	2		POS	C	Res		39	cycle 2
		19	06FEB03	16DEC02	25JAN03	N		Y	BO	ASTHENIA Fatigue	1		PRB	N	Res		12	cycle 1
			18DEC02			N		Y	HE	ANEMIA Anemia	1		PRB	N	Per		14	cycle 1
			17JAN03	17JAN03		N		Y	DI	NAUSEA Nausea	1		POS	C	Res		44	cycle 3
					17JAN03	N		Y	DI	VOMITING Vomiting	1		POS	C	Res		44	cycle 3
			20JAN03	20JAN03		N		Y	**	**Classification Unknown** Shortness of breath with anxiety	2		PNOT	C	Res		47	cycle 3
						N		Y	NE	ANXIETY Anxiety attacks	1		DNOT	C	Per		47	cycle 3
			24JAN03			N		Y	DI	ANOREXIA Anorexia	1		POS	N	Per		51	cycle 3

Figure 3: Compact ECGTEST Output

ECGTEST for Protocol 104 11:09 Wednesday, March 19, 2003 1

Investigator	Patient	Visit Date	Visit Name	ECG	Relative Day ?	Relative Day Abs	Relative Day Miles	EKB Therapy	Regimen	Milestone	Data Analysis Interval	Planned Study Interval
Koehne, Klaus Medizinische Klinik und Poliklinik I Germany	02	041702	Screen	Normal	Y	-6	-6	10mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
	03	091802	Final	Normal	Y	141	43	10mg+FU/LV/IRINOT 180 mg	Same	4	Cycle 3-4	Cycle7(+ after) treat period
				Change	N	141	43	10mg+FU/LV/IRINOT 180 mg	Same	4	Cycle 3-4	Cycle7(+ after) treat period
	04	052802	Screen	Normal	N	-2	-2	25mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
	05	060502	Screen	Normal	N	-6	-6	25mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
		080502	Final	Normal	N	56	56	25mg+FU/LV/IRINOT 180 mg	Same	1	Cycle 1-2	Cycle7(+ after) treat period
				Change	N	56	56	25mg+FU/LV/IRINOT 180 mg	Same	1	Cycle 1-2	Cycle7(+ after) treat period
	06	070902	Screen	Normal	Y	37445	37445	50mg+FU/LV/IRINOT 180 mg		1	Cycle 1-2	Pre-study Screening
		082602	Final	Normal	Y	37493	37493	50mg+FU/LV/IRINOT 180 mg	Same	1	Cycle 1-2	Cycle7(+ after) treat period
				Change	N	37493	37493	50mg+FU/LV/IRINOT 180 mg	Same	1	Cycle 1-2	Cycle7(+ after) treat period
	07	071502	Screen	Normal	Y	-3	-3	50mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
		111302	Final	Normal	Y	119	28	50mg+FU/LV/IRINOT 180 mg	Same	4	Cycle 3-4	Cycle7(+ after) treat period
				Change	N	119	28	50mg+FU/LV/IRINOT 180 mg	Same	4	Cycle 3-4	Cycle7(+ after) treat period
	08	073102	Screen	Normal	Y	-6	-6	50mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
	09	102202	Screen	Normal	N	-2	-2	75mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
Cortes- Funes, Hernan Servicio de Oncologia Spain	31	052902	Screen	Normal	Y	-10	-10	25mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
	33	070902	Screen	Normal	Y	-4	-4	50mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
	36	100802	Screen	Normal	Y	-3	-3	75mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
Tabernero, Josep Servicio de Oncologia Spain	41	041902	Screen	Normal	Y	-5	-5	10mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
	42	041602	Screen	Normal	Y	-8	-8	10mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
	43	052802	Screen	Normal	Y	37403	37403	25mg+FU/LV/IRINOT 180 mg		1	Cycle 1-2	Pre-study Screening
	44	053102	Screen	Normal	Y	-6	-6	25mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
		082602	Final	Normal	Y	82	50	25mg+FU/LV/IRINOT 180 mg	Same	2	Cycle 1-2	Cycle7(+ after) treat period
				Change	N	82	50	25mg+FU/LV/IRINOT 180 mg	Same	2	Cycle 1-2	Cycle7(+ after) treat period
	46	073102	Screen	Normal	Y	37467	37467	50mg+FU/LV/IRINOT 180 mg		1	Cycle 1-2	Pre-study Screening
		100702	Final	Normal	Y	37535	42	50mg+FU/LV/IRINOT 180 mg	Same	2	Cycle 1-2	Cycle7(+ after) treat period
				Change	N	37535	42	50mg+FU/LV/IRINOT 180 mg	Same	2	Cycle 1-2	Cycle7(+ after) treat period
	47	082802	Screen	Normal	Y	-8	-8	50mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening
	48	101002	Screen	Normal	Y	-6	-6	75mg+FU/LV/IRINOT 180 mg		1	Baseline	Pre-study Screening

Figure 4: Compact ECGTEST Output With ID Option for Visit Name

Investigator	Patient	Visit Date	Visit Name	ECG	Relative Day ?	Relative Day Abs	Relative Day Miles	EKB Therapy	Regimen	Milestone	Data Analysis Interval
Koehne, Klaus Medizinische Klinik und Poliklinik I Germany	02	041702	Screen	Normal	Y	-6	-6	10mg+FU/LV/IRINOT 180 mg		1	Baseline
	03	091802	Final	Normal	Y	141	43	10mg+FU/LV/IRINOT 180 mg	Same	4	Cycle 3-4
				Change	N	141	43	10mg+FU/LV/IRINOT 180 mg	Same	4	Cycle 3-4
	04	052802	Screen	Normal	N	-2	-2	25mg+FU/LV/IRINOT 180 mg		1	Baseline
	05	060502	Screen	Normal	N	-6	-6	25mg+FU/LV/IRINOT 180 mg		1	Baseline
		080502	Final	Normal	N	56	56	25mg+FU/LV/IRINOT 180 mg	Same	1	Cycle 1-2
				Change	N	56	56	25mg+FU/LV/IRINOT 180 mg	Same	1	Cycle 1-2
	06	070902	Screen	Normal	Y	37445	37445	50mg+FU/LV/IRINOT 180 mg		1	Cycle 1-2
		082602	Final	Normal	Y	37493	37493	50mg+FU/LV/IRINOT 180 mg	Same	1	Cycle 1-2
				Change	N	37493	37493	50mg+FU/LV/IRINOT 180 mg	Same	1	Cycle 1-2
	07	071502	Screen	Normal	Y	-3	-3	50mg+FU/LV/IRINOT 180 mg		1	Baseline
		111302	Final	Normal	Y	119	28	50mg+FU/LV/IRINOT 180 mg	Same	4	Cycle 3-4
				Change	N	119	28	50mg+FU/LV/IRINOT 180 mg	Same	4	Cycle 3-4
	08	073102	Screen	Normal	Y	-6	-6	50mg+FU/LV/IRINOT 180 mg		1	Baseline
	09	102202	Screen	Normal	N	-2	-2	75mg+FU/LV/IRINOT 180 mg		1	Baseline
Cortes- Funes, Hernan Servicio de Oncologia Spain	31	052902	Screen	Normal	Y	-10	-10	25mg+FU/LV/IRINOT 180 mg		1	Baseline
	33	070902	Screen	Normal	Y	-4	-4	50mg+FU/LV/IRINOT 180 mg		1	Baseline
	36	100802	Screen	Normal	Y	-3	-3	75mg+FU/LV/IRINOT 180 mg		1	Baseline
Tabernero, Josep Servicio de Oncologia Spain	41	041902	Screen	Normal	Y	-5	-5	10mg+FU/LV/IRINOT 180 mg		1	Baseline
	42	041602	Screen	Normal	Y	-8	-8	10mg+FU/LV/IRINOT 180 mg		1	Baseline
	43	052802	Screen	Normal	Y	37403	37403	25mg+FU/LV/IRINOT 180 mg		1	Cycle 1-2
	44	053102	Screen	Normal	Y	-6	-6	25mg+FU/LV/IRINOT 180 mg		1	Baseline
		082602	Final	Normal	Y	82	50	25mg+FU/LV/IRINOT 180 mg	Same	2	Cycle 1-2
				Change	N	82	50	25mg+FU/LV/IRINOT 180 mg	Same	2	Cycle 1-2
	46	073102	Screen	Normal	Y	37467	37467	50mg+FU/LV/IRINOT 180 mg		1	Cycle 1-2
		100702	Final	Normal	Y	37535	42	50mg+FU/LV/IRINOT 180 mg	Same	2	Cycle 1-2
				Change	N	37535	42	50mg+FU/LV/IRINOT 180 mg	Same	2	Cycle 1-2
	47	082802	Screen	Normal	Y	-8	-8	50mg+FU/LV/IRINOT 180 mg		1	Baseline
	48	101002	Screen	Normal	Y	-6	-6	75mg+FU/LV/IRINOT 180 mg		1	Baseline

Investigator	Patient	Visit Date	Visit Name	Planned Study Interval
Koehne, Klaus Medizinische Klinik und Poliklinik I Germany	02	041702	Screen	Pre-study Screening
	03	091802	Final	Cycle7(+ after) treat period
				Cycle7(+ after) treat period
	04	052802	Screen	Pre-study Screening
	05	060502	Screen	Pre-study Screening
		080502	Final	Cycle7(+ after) treat period
				Cycle7(+ after) treat period
	06	070902	Screen	Pre-study Screening
		082602	Final	Cycle7(+ after) treat period
				Cycle7(+ after) treat period