Deciphering the _TYPE_ Variable in MEANS and SUMMARY Output Data Sets

Arthur L. Carpenter
California Occidental Consultants

ABSTRACT
Output data sets generated using the procedures MEANS and SUMMARY automatically contain the variable _TYPE_. This variable can prove to be very valuable when creating or working with subsets of the summary data set, but knowing what values of _TYPE_ to select for is difficult if you do not know how its values are assigned.

Fortunately the values assigned to _TYPE_ depend on the presence of the CLASS statement and the number and order of the variables that it contains. Understanding the relationship between the CLASS statement and _TYPE_ will allow you to accurately predict the value of _TYPE_ that will be associated with a particular combination of variables in the CLASS statement. This value can then be used to select data subsets based on this relationship.

As it turns out this relationship is fairly simple to understand and this paper will show you how to determine the resulting value of _TYPE_ based on any given combination of CLASS variables.

KEY WORDS
means, summary, _type_, class statement, output statement

INTRODUCTION
The variables _FREQ_ and _TYPE_ are automatically added to the summary data set when the OUTPUT statement is used. _FREQ_ is the count of the number of observations available for use and _TYPE_ is a numeric flag which indicates the subgroup of the CLASS variables summarized by that observation in the output data set. When no CLASS statement is present the resulting data set will have one observation and _TYPE _ will equal 0.

Example 1 creates a data set, STATS, which contains the mean and n for the variables SALINITY and PH. The data set is then printed using PROC PRINT.

```
****** ex 1;
proc means data=sasclass.h2oqual
           noprint;
var salinity;
output out=stats n=nsalin mean=msalin;
run;


proc print data=stats;
title1 'Example 1';
title2 'output a summary data set';
run;
```

The variable _FREQ_ has been included by PROC MEANS. It counts the number of available observations and because of missing values, may not be the same as the number of observations used to generate the statistics.

```
Example 1
output a summary data set

OBS    _TYPE_    _FREQ_    NSALIN     MSALIN

 1        0        215       213     19.3508
```

USING THE CLASS STATEMENT
When the CLASS statement is present _TYPE_ indicates whether a particular observation in the summary data set is 'for' or 'across' values of the CLASS variable(s). In Example 1there is no CLASS statement so the summary is 'across' all observations (_TYPE_=0). In other words when we say that we are summarizing 'for' a CLASS variable, we are using its values to determine the summary subsets.

Example 2 repeats Example 1 and adds the CLASS statement for STATION. Notice that all the information generated from Example 1 is still available in the output data set.

```
****** ex 2;
proc means data=sasclass.h2oqual
           noprint;
class station;
var salinity;
output out=stats n=nsalin mean=msalin;
run;

proc print data=stats;
title1 'Example 2';
title2 'output a summary data set';
title3 'STATION is the class variable';
run;
```

TYPE=0 indicates that the statistics summarize 'across' all levels of the classification variable (values of STATION are not used). Notice that STATION has been automatically added to the output data set and its value is missing when the summary is 'across' STATION. _TYPE_=1 is used for observations summarizing 'for' STATION and the value of STATION in the summary data set designates the level of STATION which is being summarized.

```
Example 2
output a summary data set
STATION is the class variable

OBS    STATION    _TYPE_    _FREQ_   NSALIN    MSALIN

 1                   0        215      213    19.3508
 2       TS3         1        107      105    18.9908
 3       TS6         1        108      108    19.7009
```

The following table shows that an internal flag is set which indicates how a particular observation in the previous data set summarizes the CLASS variable. This flag is used to calculate the value for _TYPE_.

	CLASS VARIABLE		
Observations	**STATION**	**Internal Flag**	**_TYPE_**
1	'ACROSS'	0	0
2 - 3	'FOR'	1	1

Example 3 adds a second classification variable (DEPTH) to the CLASS statement. When two or more classification variables are present, statistics for all possible combinations of all the levels of these variables are added to the summary data set.

```
****** ex 3;
proc means data=sasclass.h2oqual
           noprint;
class depth station;
var salinity;
output out=stats(drop=_freq_)
       n=nsalin mean=msalin;
run;

proc print data=stats;
title1 'Example 3';
title2 'output a summary data set';
title3 'STATION and DEPTH are the class
variables';
run;
```

The variable _FREQ_ was dropped from the summary data set (STATS) to conserve space for the display of this listing in this paper. With this exception notice that the first three lines of the resulting listing (shown below) are the same as those produced by Example 2.

```
Example 3
output a summary data set
STATION and DEPTH are the class variables

OBS    DEPTH    STATION    _TYPE_   NSALIN    MSALIN

  1      .                    0       213    19.3508
  2      .        TS3         1       105    18.9908
  3      .        TS6         1       108    19.7009
  4    0.0                    2        46    18.5989
  5    1.0                    2        28    13.2621
  6    2.0                    2        45    21.0060
  7    2.5                    2         1     0.4300
  8    3.0                    2        25    15.5532
  9    3.5                    2         3    11.8833
 10    4.0                    2        34    22.1762
 11    5.0                    2        12    19.7567
 12    6.0                    2        16    28.9056
 13    7.0                    2         1    17.3000
 14    7.5                    2         1    18.9000
 15    8.0                    2         1    34.9000
 16    0.0        TS3         3        23    19.2935
 17    0.0        TS6         3        23    17.9043
 18    1.0        TS3         3        10     5.8230
 19    1.0        TS6         3        18    17.3950
 20    2.0        TS3         3        23    20.7861
 21    2.0        TS6         3        22    21.2359
 22    2.5        TS6         3         1     0.4300
 23    3.0        TS3         3         9     8.5767
 24    3.0        TS6         3        16    19.4775
 25    3.5        TS6         3         3    11.8833
 26    4.0        TS3         3        22    21.0855
 27    4.0        TS6         3        12    24.1758
 28    5.0        TS3         3         5    16.7400
 29    5.0        TS6         3         7    21.9114
 30    6.0        TS3         3        13    29.9385
 31    6.0        TS6         3         3    24.4300
 32    7.0        TS6         3         1    17.3000
 33    7.5        TS6         3         1    18.9000
 34    8.0        TS6         3         1    34.9000
```

The internal flag (always 0 or 1) is shown in the following table for the various combinations of these **two** CLASS variables. Together these flags can be used to create a binary value which becomes _TYPE_ when it is converted to decimal.

	CLASS VARIABLES			
Observations	**DEPTH**	**STATION**	**Binary Value**	**_TYPE_**
1	0	0	00	0
2 - 3	0	1	01	1
4 - 15	1	0	10	2
16 - 34	1	1	11	3

Conversion from binary to decimal is fairly easy when you know the process. Consider the binary number 1101, this is 13 in decimal. The following table shows the conversion process.

Binary Place value	2^3	2^2	2^1	2^0
Decimal Place value	8	4	2	1
Binary number	1	1	0	1
Binary value	1*8	1*4	0*2	1*1
Decimal values	8	4	0	1

The Decimal value of 1101 is therefore 8+4+0+1=13. When more complicated CLASS statements are required, the _TYPE_ value can still be calculated. Consider the following CLASS statement:

```
CLASS STUDY DRUG DOSE CLINIC;
```

Since DRUG and CLINIC are the second and fourth variables in the CLASS statement, a summary of all combinations of DRUG and CLINIC ('across' STUDY and DOSE) would have a binary value of 0101=(0*8)+(1*4)+(0*2)+(1*1)=4+1=5. This indicates that _TYPE_ would be 5 for this combination. The highest value of _TYPE_ for this CLASS statement would be 15 (1111 in binary).

SUMMARY

The variable _TYPE_ can be used to identify summary subsets from the data set produced by the procedures MEANS and SUMMARY. The value of _TYPE_ is easily calculated by creating a binary value based on combinations of variables in the CLASS statement and then by converting the binary value into a decimal value.

ABOUT THE AUTHOR

Art Carpenter's publications list includes two chapters in *Reporting from the Field*, the two books *Quick Results with SAS/GRAPH® Software*, and *Carpenter's Complete Guide to the SAS® Macro Language*, and over two dozen papers and posters presented at SUGI, PharmaSUG, and WUSS. Art has been using SAS since 1976 and has served as a steering committee chairperson of both the Southern California SAS User's Group, SoCalSUG, and the San Diego SAS Users Group, SANDS; a conference cochair of the Western Users of SAS Software regional conference, WUSS; and Section Chair at the SAS User's Group International conference, SUGI.

Art is a SAS Quality Partner™ and through California Occidental Consultants he teaches SAS courses and provides contract SAS programming support nationwide.

Art Carpenter
California Occidental Consultants
P.O. Box 6199
Oceanside, CA 92058-6199

(760) 945-0613
art@caloxy.com
www.caloxy.com

TRADEMARK INFORMATION