

## Using an Enhanced Numeric INFORMAT to Read a Combination of Character and Numeric Data

Ronald Cody, Ed.D.

### Introduction

Until the development of enhanced numeric informats, the only way to read a combination of character and numeric data was to read all data values as character data, check the value, and convert to numeric if necessary. We will demonstrate two programs that read a combination of character and numeric values, each using the "old-fashioned" method and using an enhanced numeric informat.

### Example 1 - "Old-fashioned" Method

For this example, we want to read some temperatures. However, instead of writing 98.6 every time a "normal" temperature was recorded, the letter 'N' was recorded instead. Here is a program that processes these data the "old-fashioned" way:

```
DATA TEMPER;
  INPUT DUMMY $ @@;
  IF DUMMY = 'N' THEN TEMP = 98.6;
  ELSE TEMP = INPUT(DUMMY,8.);
  DROP DUMMY;
DATALINES;
101.5 N 95.8 N 102.2
;
```

Notice that each data value is first read as character data. If the value is an 'N', the value of the temperature variable (TEMP) is set to 98.6. If the value is anything else, the INPUT function is used to perform a character to numeric conversion. Remember to drop the character variable from the resulting data set.

### Example 1 - Using an Enhanced Numeric Informat

Here is a program that accomplishes the same goal, using an enhanced numeric informat:

```
PROC FORMAT;
  INVALUE T 'N' = 98.6
  OTHER = _SAME_;
RUN;
```

```
DATA TEMPER;
  INFORMAT TEMP T.;
  INPUT TEMP @@;
DATALINES;
101.5 N 95.8 N 102.2
;
```

We use PROC FORMAT for create a user-defined informat using an INVALUE statement. Notice the values to the left of the equal sign can be either character values (placed in quotes), or numeric values or ranges.

### Example 2 - "Old-fashioned" Way

For this example, lead levels are entered from three different laboratories. Each of the three laboratories has a different non-detect value. For lab A, it is .05, for lab B, .1, and for lab C, .2. Each time a non-detect value is encountered, an 'A', 'B', or 'C' is entered. In addition, any value over 50 is considered a data error and the value is to be set to a SAS missing value. Here is the SAS code:

```
DATA LEAD;
  INPUT DUMMY $ @@;
  IF DUMMY = 'A' THEN X = .05;
  ELSE IF DUMMY = 'B' THEN X = .1;
  ELSE IF DUMMY = 'C' THEN X = .2;
  ELSE X = INPUT(DUMMY,8.);
  IF X GT 50 THEN X = .;
  DROP DUMMY;
DATALINES;
3 .09 A B 67 .5 C
;
```

### Example 2 - Using an Enhanced Numeric Informat

Here is the same program, using an enhanced numeric informat:

```
PROC FORMAT;
  INVALUE LEAD 'A' = .05
  'B' = .1
  'C' = .2
  0-50 = _SAME_
  OTHER = .;
RUN;

DATA LEAD;
  INPUT X : LEAD. @@;
DATALINES;
```

```
3 .09 A B 67 .5 C  
;
```

First notice that the informat was applied differently in the two examples. In Example 1, an INFORMAT statement was used to associate the user defined INFORMAT to the variable TEMP. In Example 2, the colon modifier was used to indicate that the informat LEAD was to be used to read the values of X.

## Conclusion

Use of a user-defined enhanced numeric informat can greatly simplify the task of reading combinations of character and numeric data.

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries, ® indicated USA registration.

Ronald P. Cody, Ed.D.  
Robert Wood Johnson Medical School  
Department of Environmental and Community Medicine  
675 Hoes Lane  
Piscataway, NJ 08822  
(732)235-4490  
cody@umdnj.edu