

## Using a Picture Format to Create Visit Windows

Richann Watson, Batavia, OH

### ABSTRACT

Creating visit windows is sometimes required for analysis of data. We need to make sure that we get the visit/day in the proper window so that the data can be analyzed properly. However, defining these visit windows can be quite cumbersome especially if they have to be defined in numerous programs. This task can be made easier by applying a picture format, which can save a lot of time and coding. A format is easier to maintain than a bunch of individual programs. If a change to the algorithm is required, the format can be updated instead of updating all of the individual programs containing the visit definition code.

### INTRODUCTION

Hypothetically, subjects are scheduled to come in for the first 6 days and then every week thereafter for the next four weeks, allowing +/- 3 days for each week visit. In order to see if a subject deviates from the protocol schedule, the actual week is calculated based on the study days and then compared to the expected week, with unscheduled visits allowed and designated by 99.XX, where '99' is used to represent unscheduled and '.XX' is used to represent the sequence number, starting with 01.

### THE DATA

For purposes of this paper, a dummy data set is created that contains a set of days using the DO statement with a list of items for the value of *start* where each item in the list uses the *start TO stop* syntax. Note that the DO statement can either have either all numeric or all character constants, with character constants enclosed in quotation marks. In addition, the DO statement can consist of variables. If TO or BY is used then the start and stop must be numeric. The DO loop is executed once for each value in the list.

### CREATE THE DUMMY DATA SET

To create a dummy data set that contains 38 records, the following code can be used.

```
data dummy;
  do DAYS = 1 to 28 by 1, 99.01 to 99.10 by .01;
    output;
  end;
run;
```

DAYS
1
2
3
4
5
6
7
8
9
10

DAYS
11
12
13
14
15
16
17
18
19
20

DAYS
21
22
23
24
25
26
27
28
99.01
99.02

DAYS
99.03
99.04
99.05
99.06
99.07
99.08
99.09
99.1

Display 1. Dummy Data

## CREATE THE VISIT WINDOWS

To create the visit windows, a series of IF-THEN-ELSE statements can be incorporated into each program that requires the visit windows, such as:

```

data viswin_if;
  set dummy;
  length VISIT $50.;

  if DAYS = 1 then VISIT = 'Screening';

  else if DAYS = 2 then VISIT = 'Baseline';
  else if 3 <= DAYS < 7 then VISIT = catx(" ", 'Day', put(DAYS, 3.));

  else if 7 <= DAYS < 99 then do;
    weeks = round(DAYS * .142857, 1);
    visit = catx(" ", 'Week', put(weeks, 3.));
  end;

  else if DAYS >= 99 then do;
    unsch = (DAYS - 99) * 100;
    visit = catx(" ", 'Unscheduled Visit', put(unsch, Z2.));
  end;

  drop weeks unsch;
run;

```

DAYS	VISIT
1	Screening
2	Baseline
3	Day 3
4	Day 4
5	Day 5
6	Day 6
7	Week 1
8	Week 1
9	Week 1
10	Week 1
11	Week 2
12	Week 2
13	Week 2

DAYS	VISIT
14	Week 2
15	Week 2
16	Week 2
17	Week 2
18	Week 3
19	Week 3
20	Week 3
21	Week 3
22	Week 3
23	Week 3
24	Week 3
25	Week 4
26	Week 4

DAYS	VISIT
27	Week 4
28	Week 4
99.01	Unscheduled Visit 01
99.02	Unscheduled Visit 02
99.03	Unscheduled Visit 03
99.04	Unscheduled Visit 04
99.05	Unscheduled Visit 05
99.06	Unscheduled Visit 06
99.07	Unscheduled Visit 07
99.08	Unscheduled Visit 08
99.09	Unscheduled Visit 09
99.1	Unscheduled Visit 10

### Display 2. Dummy Data With Visit Window Using If-Then-Else

To maintain this code for every program can be quite tedious. If a change to the algorithm is required, then every affected program needs to be updated accordingly.

A simple solution is to create a format that contains this algorithm. This can be created using the 'PICTURE' statement instead of 'VALUE'.

```

libname library "location of format catalog";

proc format lib=library;
  picture visitwin (round)
    1 = 'Screening' (noedit)
    2 = 'Baseline' (noedit)
    3-6 = 9 (prefix='Day ')
    7-98 = 09 (prefix='Week ' mult=.142857)
    99.01 - 99.99 = 00.99 (prefix='Unscheduled Visit');
run;

```

Based on the series of IF-THEN-ELSE statements above, days 1 and 2 have a set value. A preset value that does not require any editing can be defined in the PICTURE format using the NOEDIT option. If a value is to be preceded with a specific label for a range of values, then PREFIX= allows the user to define what label they want to use. In addition, the MULT= option allows the user to specify a multiplier that will be used to convert the existing value. In this example, the MULT=.142857 allows for a visit window of +/- 3 days, so the week will be determined based on the number of days (i.e. days will be multiplied by the multiplier prior to formatting). The ROUND option allows the visit windows to be rounded to the nearest week; otherwise, the value generated by the multiplier will be truncated (i.e. decimal portion dropped) and the integer portion will be used to print the specified format.

```
data viswin_pic;
  set dummy;
  length VISIT $50.;
  VISIT = put(days, visitwin.);
run;
```

DAYS	VISIT
1	Screening
2	Baseline
3	Day 3
4	Day 4
5	Day 5
6	Day 6
7	Week 1
8	Week 1
9	Week 1
10	Week 1
11	Week 2
12	Week 2
13	Week 2

DAYS	VISIT
14	Week 2
15	Week 2
16	Week 2
17	Week 2
18	Week 3
19	Week 3
20	Week 3
21	Week 3
22	Week 3
23	Week 3
24	Week 3
25	Week 4
26	Week 4

DAYS	VISIT
27	Week 4
28	Week 4
99.01	Unscheduled Visit 99.01
99.02	Unscheduled Visit 99.02
99.03	Unscheduled Visit 99.03
99.04	Unscheduled Visit 99.04
99.05	Unscheduled Visit 99.05
99.06	Unscheduled Visit 99.06
99.07	Unscheduled Visit 99.07
99.08	Unscheduled Visit 99.08
99.09	Unscheduled Visit 99.09
99.1	Unscheduled Visit 99.10

Display 3. Dummy Data With Visit Window Using Picture Format

Note that the data is not quite in the desired format, but that can be fixed by using the TRANWRD, TRIM and LEFT functions.

```
data viswin_pic;
  set dummy;
  length VISIT $50.;
  VISIT = trim( left( tranwr( put(days, visitwin.), ' 99.', '' ) ) );
run;
```

DAYS	VISIT
1	Screening
2	Baseline
3	Day 3
4	Day 4
5	Day 5
6	Day 6
7	Week 1
8	Week 1
9	Week 1
10	Week 1
11	Week 2
12	Week 2
13	Week 2

DAYS	VISIT
14	Week 2
15	Week 2
16	Week 2
17	Week 2
18	Week 3
19	Week 3
20	Week 3
21	Week 3
22	Week 3
23	Week 3
24	Week 3
25	Week 4
26	Week 4

DAYS	VISIT
27	Week 4
28	Week 4
99.01	Unscheduled Visit 01
99.02	Unscheduled Visit 02
99.03	Unscheduled Visit 03
99.04	Unscheduled Visit 04
99.05	Unscheduled Visit 05
99.06	Unscheduled Visit 06
99.07	Unscheduled Visit 07
99.08	Unscheduled Visit 08
99.09	Unscheduled Visit 09
99.1	Unscheduled Visit 10

Display 4. Final Dummy Data With Visit Window Using Picture Format

## **CONCLUSION**

It is sometimes said a picture is worth a thousand words. In this case it may not be a thousand words but it is definitely worth something. For projects that require a consistent visit designation across all data sets or if the actual visit occurred according to a pre-defined schedule creating a PICTURE FORMAT to determine these visit windows can save programming time and lots of headaches especially if more visits are added or if the visit windows change.

## **REFERENCES**

Bilenas, Jonas. 2005. *The Power of PROC FORMAT*. Cary, NC: SAS Publishing

## **ACKNOWLEDGMENTS**

Thanks to my colleagues at PharmaNet/i3, who reviewed this paper and supported me while we worked through this process. In addition, I would like to thank Rick Langston for answering my question regarding the NOEDIT option.

## **CONTACT INFORMATION**

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

Richann Watson  
PAREXEL  
Richann.Watson@PAREXEL.com

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.