

Why not Picture Format?

Gaoyang LI, Bayer Healthcare co., ltd, Beijing

ABSTRACT

The importance and benefits of formats procedure are well-known. Nearly every SAS programmers know how to use SAS built-in FORMATS/INFORMAT or create costumed FORMATS/INFORMAT. However, it seems that picture format was excluded from their toolkit. Why not Picture format? The rationale of PICTURE statement is different from VALUE/INVALUE, at the same time it bring us some wonderful method to display or output data. In this paper, the logic of picture format is parsed step by step to make sure you are confident to use it without worrying error. A few examples with different options are followed to demonstrate its efficiency.

INTRODUCTION

There is a book named **the Healthy Programmer**, with a subtitle keeps fit, feel better, and keep coding. I am strongly agree with the author's idea about a healthy programmer from 2 aspects. On one hand, as a healthy SAS programmer, we need physically keep our body fit and mentally feel better about our work and life, and keep coding for all kinds of projects/studies. On the other hand, for our SAS programing or codes, we need to keep it fit, especially not to fat, and avoiding error-prone mess and wide structure. This would make us feel logically clear and emotionally better, therefore we could keep coding with confidence. For how to keep yourself fit as a SAS programmer, you may read the book. For how to keep your codes fit when programming, I recommend Format procedure, and focus on Picture Format since it is flexible and powerful, but under-used not only for entry level SAS programmers, but also experienced SAS programmers.

PICTURE FORMAT STEP BY STEP

Why not Picture Format? Perhaps because there is not build-in picture formats supplied in SAS, but there are other reasons. Let's have a whole picture of SAS format procedure and investigate the difference between Picture Formats and Formats. The FORMAT procedure enables you to define your own INFORMATS and formats for variables. INFORMATS determine how raw data values are read and stored. Formats determine how variable values are printed. There are 2 broad classes of Formats: VALUE Formats and PICTURE formats. A PICTURE Format could only associate with numeric variables and it draws a template rather than just calculate the value to display the numeric variables. Let's use following

example to illustrate how to build a PICTURE format and how it works step by step. With the same source data, 2 PICTURE formats are created to show the difference or diversity.

a) Picture template

The string in the quote on the right side of the equal sign is picture template. Picture templates are specified with three types of characters: digit selectors, message characters, and directives. It is not the exact value which would be output, but the exact layout the output looks like. Please note that picture template can only begin with digitals. Numbers in the template would be replaced by actual number in data. Non-digital characters on first position of the picture template would increase picture format length without error or warning in log, but it would not be output.

```

***Example a: Picture template;
proc format library=work;
  picture rtime_a
    low - high  ='99D99H99M'
    ;
  picture rtime_aa
    low - high  ='A999-999'
    ;
run;

data ae_raw;
  AETERM='HEADACHE'; AEPST=120320;      output;
  AETERM='HEADACHE'; AEPST=0;          output;
  AETERM='SICKNESS'; AEPST=1105;      output;
run;

data ae_output;
  set ae_raw;
  AEPSTC=put(AEPST,rtime_a.);
  AEPSTCC=put(AEPST,rtime_aa.);
run;

```

AETERM	AEPST	AEPSTC	AEPSTCC
HEADACHE	120320	12D03H20M	120-320
HEADACHE	0	00D00H00M	000-000
SICKNESS	1105	00D11H05M	001-105

Note: the character 'A' in the format `rtime_aa` was not output but replaced by a blank.

b) Digital selector

The numbers (0 through 9) in the picture template are digital selector by default. Zero digital selectors would let the format remove leading zeros. You can have a maximum of 16 digit selectors in a picture.

```

***Example b: digital selector;

proc format library=work;
  picture rtime_b
    low - high  ='00D00H00M'
  ;
  picture rtime_bb
    low - high  ='09D09H09M'
  ;
run;

```

AETERM	AEPST	AEPSTC	AEPSTCC
HEADACHE	120320	12D03H20M	12D03H20M
HEADACHE	0		0D00H00M
SICKNESS	1105	11H05M	0D11H05M

Note: if the data is zero, both numbers and characters in the picture template would not be output.

c) NOEDIT options

This option is tell SAS the numbers in the picture format for this range is not digital selectors, but the actual value would be output.

```

***Example c: noedit options;

proc format library=work;
  picture rtime_c
    0          ='0D0H0M'
    other     ='09D99H99M'
  ;
  picture rtime_cc
    0          ='0D0H0M' (noedit)
    other     ='09D99H99M'
  ;
run;

```

AETERM	AEPST	AEPSTC	AEPSTCC
HEADACHE	120320	12D03H20M	12D03H20M
HEADACHE	0		0D0H0M
SICKNESS	1105	0D11H05M	0D11H05M

Note: with the NOEDIT option, zero was output exactly as picture template for its range, '0D0H0M', rather than blanks, nor '0D00H00M'.

d) PREFIX options

Though non-digital characters or symbols could not occupy the first position of the picture format, non-digital characters or symbols could be added by prefix options for its range. It is recommended using default length options whenever you use prefix options, especially the prefix are special characters. If no default length was specified, the default length would ignore the prefix if the length is not enough.

```

***Example d: prefix options;
proc format library=work;
  picture rtime_d
    low - <0      = '99D99H99M'      (prefix="BEFORE ")
    0             = '0D0H0M'        (noedit)
    0< - high    = '99D99H99M'      (prefix="AFTER ")
  ;
  picture rtime_dd (default=11)
    low - <0      = '99D99H99M) '    (prefix="( ")
    0             = '0D0H0M) '      (prefix="( ")
    0< - high    = '99D99H99M) '    (prefix="( ")
  ;
run;

```

AETERM	AEPST	AEPSTC	AEPSTCC
HEADACHE	-120320	BEFORE 12D03H20M	(12D03H20M)
HEADACHE	0	0D0H0M	
SICKNESS	1105	AFTER 00D11H05M	(00D11H05M)

Note: similar to message characters in the picture template, prefix would be ignored for zero.

e) FILL options

Fill option is only for zero digital selector, rather than non-zero. Fill options replace the leading zeros with the specified characters, instead filling blanks. If there is not leading zeros, there is no characters are filled even if you have specified a character. If you use the FILL= and PREFIX= options in the same picture, then the format places the prefix and then the fill characters.

```

***Example e: fill options with different digital selector;
proc format library=work;
  picture rtime_e
    low - <0    ='99D99H99M' ( fill='*' prefix="BEFORE ")
    0          =' 0D00H00M' ( fill='*')
    0< - high  ='99D99H99M ' ( fill='*' prefix="AFTER ")
  ;
  picture rtime_ee
    low - <0    ='00D00H00M' ( fill='*' prefix="BEFORE ")
    0          ='00D00H00M' ( fill='*')
    0< - high  ='00D00H00M ' ( fill='*' prefix="AFTER " )
  ;
run;

```

AETERM	AEPST	AEPSTC	AEPSTCC
HEADACHE	-120320	BEFORE 12D03H20M	BEFORE 12D03H20M
HEADACHE	0	*****	*****
SICKNESS	1115	AFTER 00D11H15M	*AFTER 11H15M

Note: 'AFTER 00D11H15M' was not filled with '*' in column AEPSTCC due to no leading zero would be replaced since 9 is the digital selector.

f) DEFAULT options

Default length of the picture format could be specified by default options. The maximum or minimum length could be specified by Max or Min options. If the length is not enough, the right side of the picture format would be truncated rather than left in value format.

```

***Example f: default options;
proc format library=work;
  picture rtime_f (default=40)
    low - <0      ='99D99H99M' ( fill='*' prefix="BEFORE ")
    0             = '0D00H00M' ( fill='*' )
    0< - high    ='99D99H99M' ( fill='*' prefix="AFTER ")
  ;
  picture rtime_ff (default=10)
    low - <0      ='99D99H99M' ( fill='*' prefix="BEFORE ")
    0             = '0D00H00M' ( fill='*' )
    0< - high    ='99D99H99M' ( fill='*' prefix="AFTER ")
  ;
run;

```

AETERM	AEPST	AEPSTC	AEPSTCC
HEADACHE	-120320	*BEFORE 12D03H20M	12D03H20M
HEADACHE	0	*****	*****
SICKNESS	1115	*AFTER 00D11H15M	00D11H15M

Note: asterisk (*) is not filled due to limited length. Only the blank at the end of prefix charters was added.

g) MULT and ROUND options

The output of the picture format could be multiplied by specifying mult options for its range.

You may be wondering that convert decimal fraction to percentage, multiply 100 is enough, why 1000 here? The tricky is at all circumstance there are how many digitals behind the decimal in the picture template, the picture format multiply how many power of 10 even the input numbers are equal to the output number .if no other multiply calculation, the picture format would multiply it according to PICTURE TEMPLATE if there is no explicit mult options. In this case, we need to multiply the decimal fraction by another 100, therefore it should be mult=1000.

```

***Example g: mult and round options;
proc format library=work;
  picture compl_g
    low - high  ='009.9%' ( mult=1000 )
  ;
  picture compl_gg ( round)
    low - high  ='009.9%'
  ;
run;

data da_raw;
  SUBJID='540101'; TAKEN=30; EXDAY=30; COMPL=TAKEN/EXDAY;
  output;
  SUBJID='540102'; TAKEN=0;  EXDAY=30; COMPL=TAKEN/EXDAY;
  output;
  SUBJID='540103'; TAKEN=26; EXDAY=30; COMPL=TAKEN/EXDAY;
  output;
  SUBJID='540104'; TAKEN=37; EXDAY=30; COMPL=TAKEN/EXDAY;
  output;
run;

data da_output;
  set da_raw;
  COMPLC=put (      COMPL, compl_g.);
  COMPLCC=put (100*COMPL, compl_gg.);
run;

```

SUBJID	TAKEN	EXDAY	COMPL	COMPLC	COMPLCC
540101	30	30	1	100.0%	100.0%
540102	0	30	0	0.0%	0.0%
540103	26	30	0.8666666667	86.6%	86.7%
540104	37	30	1.2333333333	123.3%	123.3%

Note: The difference is round 0.8666666667 into 86.7% rather than 86.6%

h) Directive

Directive is special characters that you can use in the picture to format date, time, or DATETIME values. Directive is for DATE, TIME or DATETIME, these 3 datatype, therefore datatype options is required. With directive in picture template, numbers are not as digital selector and characters other than the directive keywords could be added at the first position of picture template. If you format a numeric missing value, then the resulting label will be ERROR.

```

***Example h: Directive and datatype options;

proc format library=work;
  picture dt_h (default=11)
    low - high = '%b-%0d-%Y' ( datatype=date)
    ;
  picture dt_hh (default=21)
    low - high = '%Y-%0m-%0d-T%0I:%0M:%0S'
    ( datatype=datetime)
    ;

run;

data lb_raw;
length SUBJID $6 LBTESTCD $8;
format LBDT date9. LBSTDTM datetime19.;
  SUBJID='540101'; LBTESTCD='AST'; LBDT='08JUL2014'd;
LBSTDTM='08JUL2014:8:45:08'dt; output;
  SUBJID='540102'; LBTESTCD='WBC'; LBDT='09AUG2014'd;
LBSTDTM='09AUG2014:9:00:16'dt; output;
  SUBJID='540103'; LBTESTCD='HDL'; LBDT='10SEP2014'd;
LBSTDTM='10SEP2014:10:15:24'dt; output;
  SUBJID='540104'; LBTESTCD='LDL'; LBDT='12OCT2014'd;
LBSTDTM='12OCT2014:12:30:30'dt; output;
run;

data lb_output;
  set lb_raw;
  LBDTC=put( LBDT, dt_h.);
  LBSTDTC=put(LBSTDTM, dt_hh.);
run;

```

SUBJID	LBTESTCD	LBDT	LBSTDTM	LBDTC	LBSTDTC
540101	AST	08JUL2014	08JUL2014:08:45:08	JUL-08-2014	2014-07-08-T08:45:08
540102	WBC	09AUG2014	09AUG2014:09:00:16	AUG-09-2014	2014-08-09-T09:00:16
540103	HDL	10SEP2014	10SEP2014:10:15:24	SEP-10-2014	2014-09-10-T10:15:24
540104	LDL	12OCT2014	12OCT2014:12:30:30	OCT-12-2014	2014-10-12-T12:30:30

Note: The date format in LBDTC is similar to date9. format but with different order. The colons (:) in LBSTDTM between date part and time part are replaced by 'T' in LBSTDTC. The output form of DATE, TIME, and DATETIME could be changed or combined in any order according to your needs.

PICTURE FORMAT PROCESS

The major pieces of the picture format have been illustrated. In order to create picture format with confidence, we need understand its process. Let's use following example from SAS support website to get a whole picture.

From the step by step table, there are 3 tricky points:

- 1) picture format ignore minus symbol “-”, due to it only process the data based on its absolute value **(step 2)**;
- 2) It would multiply the absolute value according to the number of digitals behind the decimal specified in the picture template, if no decimal specified in the picture template and the data for this range have decimals, you need to multiply it manually by MULT options **(step 3)**.
- 3) It ignores decimals and it only draws decimals according the position of the range and the format template. Therefore you need to draw a “-” before the number with prefix options if is a negative number. Correspondingly, you need to draw a minus symbol and a decimal symbol before the number with prefix options if it does not begin with digitals. That is “-” for negative numbers or /and “.” for decimal fraction **(step 7)**.

<pre> data sample; input Amount; datalines; -2.051 -.05 -.017 0 .093 .54 .556 6.6 14.63 0.996 -0.999 -45.00 ; run; </pre>	<pre> libname library 'SAS-library'; proc format; picture nozerosR (round fuzz=0) low - -1 = '000.00' (prefix='-') -1 < - < -.99 = '0.99' (prefix='- ' mult=100) -0.99 <-< 0 = '99' (prefix='-.' mult=100) 0 = '9.99' 0 < -< .99 = '99' (prefix='.' mult=100) 0.99 - < 1 = '0.99' (mult=100) 1 - high = '09.99'; picture nozeros (fuzz=0) low - -1 = '000.00' (prefix='-') -1 < - < -.99 = '0.99' (prefix='-.' mult=100) -0.99 <-< 0 = '99' (prefix='-.' mult=100) 0 = '9.99' 0 < -< .99 = '99' (prefix='.' mult=100) 0.99 - < 1 = '0.99' (prefix='.' mult=100) 1 - high = '09.99'; run; </pre>
---	---

Step	Action	-2.051	-.05	.556	.996	6.6
1	Range	low --1	-0.99 < -- < 0	0 < -- < .99	0.99 -- < 1	1 -- high
	Picture	000.00	99	99	0.99	09.99
2	Absolute value	2.051	.05	.556	.996	6.6
3	MULT=	2.051×10 ² =205.1	.05×100=5	.556×100=55.6	.996×100=99.6	6.6×10 ² =660
4	Round	205	5	56	100	660
4a	No Rounding	205	5	55	99	660
5	Character string, rounding	00205	05	56	100	0660
5a	Character string, no rounding	00205	05	55	099	0660
6	Template, rounding	2.05	05	56	1.00	6.60
6a	Template, no rounding	2.05	05	55	99	6.60
7	Prefix, rounding	prefix='-'	prefix='-'	prefix=''	none	none
7a	Prefix, no rounding	prefix='-'	prefix='-'	prefix=''	prefix=''	none
	Formatted result, rounding	-2.05	-.05	.56	1.00	6.60
	Formatted results, no rounding	-2.05	-.05	.55	.99	6.60

CONCLUSION

PICTURE format is powerful and flexible when mapping data, analysis data and report data. First, like value format, it could be served as global or project code list, at least study level code list in a centralized way, rather than deal with it in each SAS program or domains or tables. Second, it only change layout for numeric variables when output/display with a viewable template, avoiding error-prone and time-

consuming pre-manipulation of the data. Third, it could conduct simple calculation by multiplying and rounding data, without touch the date itself. At the same time, you must be very careful about the decimal and negative symbol of your data.

REFERENCES

SAS Institute Inc., “Base SAS® 9.4 Procedures Guide, Third Edition”. Available at:

<http://support.sas.com/documentation/cdl/en/proc/67327/HTML/default/viewer.htm#titlepage.htm>

Andrew H. Karp, “Getting into Picture (format)”. Proceedings of SUGI 31. Available at:

<http://www2.sas.com/proceedings/sugi31/243-31.pdf>

Christiana Williams, “SAS Formats Top Ten”. *Proceedings of SAS Global Forum 2015* . Available at:

<http://support.sas.com/resources/papers/proceedings15/3155-2015.pdf>

Arthur L. Carpenter, “Building and Using User-Defined Formats”. Proceedings of SUGI 29. Available at

<http://www2.sas.com/proceedings/sugi29/236-29.pdf>

ACKNOWLEDGMENTS

Thanks XXXX for reviewing and commenting on this paper.

CONTACT INFORMATION

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

Name: Gaoyang LI

Enterprise: Bayer Healthcare co., ltd, Beijing

Address: Bayer Center, No.27, Dong San Huan North Road Chaoyang District, Beijing, 100020

Work phone: (86-10) 6536-0717

E-mail: Gaoyang.li@bayer.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.

APPENDIX

***Example a: Picture template;

```
proc format library=work;
    picture rtime_a
        low - high  = '99D99H99M'
        ;
    picture rtime_aa
        low - high  = 'A999-999'
        ;
run;
```

***Note: picture template can only begin with digitals.

**Non-digital characters on first position of the picture template would increase picture format length without error or warning in log.;

```
data ae_raw;
    AETERM='HEADACHE'; AEPST=120320;    output;
    AETERM='HEADACHE'; AEPST=0;        output;
    AETERM='SICKNESS'; AEPST=1105;     output;
run;
```

```
data ae_output;
    set ae_raw;
    AEPSTC=put(AEPST,rtime_a.);
    AEPSTCC=put(AEPST,rtime_aa.);
run;
```

***Example b: digital selector;

```
proc format library=work;
    picture rtime_b
        low - high  = '00D00H00M'
        ;
    picture rtime_bb
        low - high  = '09D09H09M'
        ;
run;
```

```
data ae_raw;
    AETERM='HEADACHE'; AEPST=120320;    output;
    AETERM='HEADACHE'; AEPST=0;        output;
    AETERM='SICKNESS'; AEPST=1105;     output;
run;
```

***Note: the numbers (0 through 9) in the picture template are digital selector by default.;

***Note: Zero digital selector would let the format remove leading zeros. ;

```
data ae_output;
    set ae_raw;
    AEPSTC=put(AEPST,rtime_b.);
    AEPSTCC=put(AEPST,rtime_bb.);
run;
```

***Example c: noedit options;

```
proc format library=work;
  picture rtime_c
    0          = '0D0H0M'
    other     = '09D99H99M'
  ;
  picture rtime_cc
    0          = '0D0H0M' (noedit)
    other     = '09D99H99M'
  ;
run;

data ae_raw;
  AETERM='HEADACHE'; AEPST=120320;   output;
  AETERM='HEADACHE'; AEPST=0;       output;
  AETERM='SICKNESS'; AEPST=1105;    output;
run;

data ae_output;
  set ae_raw;
  AEPSTC=put(AEPST,rtime_c.);
  AEPSTCC=put(AEPST,rtime_cc.);
run;
```

***Example d: prefix options;

```
proc format library=work;
  picture rtime_d
    low - <0   = '99D99H99M'      (prefix="BEFORE ")
    0          = '0D0H0M'        (noedit)
    0< - high = '99D99H99M'      (prefix="AFTER ")
  ;
  picture rtime_dd (default=11)
    low - <0   = '99D99H99M) '    (prefix="( ")
    0          = '0D0H0M) '      (prefix="( ")
    0< - high = '99D99H99M) '    (prefix="( ")
  ;
run;
```

*Note: It is recommend to use default length options when ever you use prefix options, espically the prefix are speical characters.;

```
data ae_raw;
  AETERM='HEADACHE'; AEPST=-120320; output;
  AETERM='HEADACHE'; AEPST=0;       output;
  AETERM='SICKNESS'; AEPST=1105;    output;
run;

data ae_output;
  set ae_raw;
  AEPSTC=put(AEPST,rtime_d.);
  AEPSTCC=put(AEPST,rtime_dd.);
run;
```

***Example e: fill options;

**Note: Fill options are only for 0 digital selector, rather than 9;
**Note: Fill options are filling the specified chacter to instead filling blanks. ;

****Note:** If the picture template is fully replaced by data, fill options would not work.;

```
proc format library=work;
  picture rtime_e
    low - <0    = '99D99H99M' ( fill='*' prefix="BEFORE ")
    0          = '0D00H00M' ( fill='*' )
    0< - high  = '99D99H99M ' ( fill='*' prefix="AFTER ")
    ;
  picture rtime_ee
    low - <0    = '00D00H00M' ( fill='*' prefix="BEFORE ")
    0          = '00D00H00M' ( fill='*' )
    0< - high  = '00D00H00M ' ( fill='*' prefix="AFTER ")
    ;
run;

data ae_raw;
  AETERM='HEADACHE'; AEPST=-120320;   output;
  AETERM='HEADACHE'; AEPST=0;         output;
  AETERM='SICKNESS'; AEPST=1115;      output;
run;

data ae_output;
  set ae_raw;
  AEPSTC=put(AEPST,rtime_e.);
  AEPSTCC=put(AEPST,rtime_ee.);
run;
```

*****Example f:** default options;
****MAX=length** specifies a maximum length for the format. **MIN=length** specifies a minimum length for the format. ;

```
proc format library=work;
  picture rtime_f (default=40)
    low - <0    = '99D99H99M' ( fill='*' prefix="BEFORE ")
    0          = '0D00H00M' ( fill='*' )
    0< - high  = '99D99H99M ' ( fill='*' prefix="AFTER ")
    ;
  picture rtime_ff (default=10)
    low - <0    = '99D99H99M' ( fill='*' prefix="BEFORE ")
    0          = '0D00H00M' ( fill='*' )
    0< - high  = '99D99H99M ' ( fill='*' prefix="AFTER ")
    ;
run;
```

*****Note:** The length of picture format is defined 10, and the digitals and characters number is 9. ;

*****Note:** There is a blank in the prefix string, so asterisk (*) is not added due to limited length;

*****Note:** If the length is not enough, the right side of the picture format would be truncated rather than left in value format. ;

*****Interaction:** If you use the FILL= and PREFIX= options in the same picture, then the format places the prefix and then the fill characters.;

```

proc format library=work;

run;

data ae_raw;
  AETERM='HEADACHE'; AEPST=-120320;   output;
  AETERM='HEADACHE'; AEPST=0;         output;
  AETERM='SICKNESS'; AEPST=1115;      output;
run;

data ae_output;
  set ae_raw;
  AEPSTC=put(AEPST, rtime_f.);
  AEPSTCC=put(AEPST, rtime_ff.);
run;

***Example g: mult and round options;
**The difference is round 0.8666666667 into 86.7 rather than 86.6.;
**CAUTION: The picture must be wide enough for an additional digit if
rounding a number adds a digit to the number.99.99 -->100.0;
proc format library=work;
  picture compl_g
    low - high = '009.9%' ( mult=1000 )
    ;
  picture compl_gg ( round)
    low - high = '009.9%'
    ;
run;

data da_raw;
  SUBJID='540101'; TAKEN=30; EXDAY=30; COMPL=TAKEN/EXDAY;
  output;
  SUBJID='540102'; TAKEN=0;  EXDAY=30; COMPL=TAKEN/EXDAY;
  output;
  SUBJID='540103'; TAKEN=26; EXDAY=30; COMPL=TAKEN/EXDAY;
  output;
  SUBJID='540104'; TAKEN=37; EXDAY=30; COMPL=TAKEN/EXDAY;
  output;
run;

data da_output;
  set da_raw;
  COMPLC=put( COMPL, compl_g.);
  COMPLCC=put(100*COMPL, compl_gg.);
run;

***Example h: Directive and datatype options;

proc format library=work;
  picture dt_h (default=11)
    low - high = '%b-%0d-%Y' ( datatype=date)
    ;
  picture dt_hh (default=21)
    low - high = '%Y-%0m-%0d-T%0I:%0M:%0S' ( datatype=datetime)
    ;
run;

```

**Tip: If you format a numeric missing value, then the resulting label will be ERROR.

Adding a clause to your program that checks for missing values can eliminate the

ERROR label.;

```
data lb_raw;
```

```
length SUBJID $6 LBTESTCD $8;
```

```
format LBDT date9. LBSTDTM datetime19.;
```

```
    SUBJID='540101'; LBTESTCD='AST'; LBDT='08JUL2014'd;
```

```
LBSTDTM='08JUL2014:8:45:08'dt; output;
```

```
    SUBJID='540102'; LBTESTCD='WBC'; LBDT='09AUG2014'd;
```

```
LBSTDTM='09AUG2014:9:00:16'dt; output;
```

```
    SUBJID='540103'; LBTESTCD='HDL'; LBDT='10SEP2014'd;
```

```
LBSTDTM='10SEP2014:10:15:24'dt; output;
```

```
    SUBJID='540104'; LBTESTCD='LDL'; LBDT='12OCT2014'd;
```

```
LBSTDTM='12OCT2014:12:30:30'dt; output;
```

```
run;
```

**Note: Directive is for date, time or datetime, therefore datatype options is required.

**With directive in picture template, digitals, is not as digital selector and characters other than the directive keywords could be added at the first position of picture template.;

```
data lb_output;
```

```
    set lb_raw;
```

```
    LBSTC=put(    LBDT, dt_h.);
```

```
    LBSTDTC=put(LBSTDTM, dt_hh.);
```

```
run;
```