**PharmaSUG 2014 - Paper CC25**

# SCAN and FIND “CALL SCAN”

Usha Kumar, inVentiv Health Clinical, Pune, Maharashtra, INDIA

## ABSTRACT

Let’s explore the unexplored. Most of us are frequent users of SCAN and FIND/INDEX function. We use these when it comes to parsing a string. If we were to figure out the nth word from a character string, we would do so by using SCAN to solve this. If we are to figure out the starting position of the nth word extracted, we would try and use INDEX/FIND for this purpose. What if we come up with a situation where we are to find out the position of the nth word that is repeated multiple times in a string?  It gets more complicated. We have definitely come across this situation and solved it too. But let’s find out if it is as easy as solving using the CALL SCAN routine available in SAS® v9.1 and above.

## INTRODUCTION

We have frequently used certain character functions like SCAN, FIND to help us locate the first occurrence of a word in the string in either direction. SCAN can be used to extract the nth word quickly. FIND can be used to find the position of the nth word. We will however see how CALL SCAN is better over the other ways of solving specific problem when it comes to locating the position of the nth word in the string.

## WHEN DO I USE CALL SCAN?

Let us see a simple example of a fixed-text format string where we want to extract parts of the character string. The example string used below has a fixed format with name of the medical symptom as the first part, date as the second part and severity of the adverse event (AE) as the third part. All of these parts are separated by the delimiter ‘-‘.  Note that the second part is always a date and the third part of the text takes only fixed set of values. We need to extract the string in such a way that the date and text are read separately into different variables i.e. we want to separate the name of the AE and the date.

The problem here is that the first part of the text can also have the delimiter as part of its value.

**Example 1:**

String ’HEAD –ACHE – 24 JAN 2013–SEVERE’;

***Problem Statement***

Separate out the date and the name of AE into 2 different variables

Let’s see the way of solving this using the most commonly used functions SCAN and INDEX.

***Solution using SCAN and INDEX:***

```
DATA _null_;
LENGTH string _string str1 str2 $200;
    string=’HEAD-ACHE -24 JAN 2013-SEVERE’;
    _string=string;
    Pos=0;
    Curr=0;
    prev=0;

    DO WHILE (INDEX(_string,“-”)>0);
       Prev=prev+pos;
       Pos=INDEX(_string,“-”);
       _string=SUBSTR(_string,pos+1,LENGTH(_string)-pos);
       Curr=curr+pos;
       PUT _string “*” curr “*” prev;
    END;

    str1=SUBSTR(string, 1,prev-1);
    str2=SCAN(string,-2,"-");
```

```
        PUT str1 “,” str2;
    RUN;
```

We could also use FIND in place of INDEX function. We see that, we need one step of DO LOOP to find the last and the second last position of the delimiter and another step to separate out the strings as required.

***Solution using CALL SCAN routine:***

Syntax of CALL SCAN –

**CALL SCAN**(<*string*>, *count*, *position*, *length* <, <*charlist*> <, <*modifier(s)*>>>);

The first parameter *string* is searched for the nth word mentioned in the parameter *count* where words in the string are separated by delimiter specified in the *charlist* parameter. This routine extracts the nth word from the string and returns it in the variable *position* along with the length of the string in the variable parameter *length*. Modifiers are used to modify the search process.

```
DATA _null_;
LENGTH string str1 str2 $200.;
    string=’HEAD-ACHE – 24 JAN 2013-SEVERE’;
    CALL SCAN (string, -2, position, length, ‘-’);
    str1=SUBSTR(string, 1,position-2);
    str2= SUBSTR(string,position,length);
    PUT str1 “,” str2;
RUN;
```

From the above, we see that the complex DO LOOP has been replaced with a simple CALL SCAN routine which returns the position of the start of the nth word from a string where words are separated by the delimiter ‘-‘. It not only returns the length of the nth word but also its position in the string.

**<u>Example 2:</u>**

String ’HEADACHE -24 JAN 2013-SEVERE-HEADACHE-04 FEB 2013 - MODERATE’

***Problem Statement***

The above example gets a little complicated with the AE repeating. We are to figure out the date of the last recorded AE ‘HEADACHE’. We assume the format of the text to be fixed and the AE text does not contain any delimiter.

***Solution using SCAN and INDEX:***

It’s very clear from the last example that we would need to track the position of every word using an additional step as we loop and extract the word using SCAN on similar lines as explained above refer “Solution using SCAN and INDEX”.

***Solution using CALL SCAN routine:***

We use CALL SCAN to read each word one by one and compare with the text “HEADACHE”. We continue comparing till we get the last match in the string, that’s the solution; we can now extract the next word immediately as we know the position of the last occurrence of the word “HEADACHE”.

```
DATA RESULT;
LENGTH text final $200;
    string="HEADACHE-24 JAN 2013-SEVERE-HEADACHE-04 FEB 2013 - MODERATE";
    counter=1;      length=0;     position=-1; text="”; /* Initialise */
    DO UNTIL (length=0);
       PUT "Enter:” counter position length text;

       CALL SCAN (string, counter, position, length, "-”);
       IF length NE 0 THEN text=SUBSTR(string,position,length);
       ELSE text="";
       IF text="HEADACHE” THEN DO;
          CALL SCAN(string, counter+1, position, length, "-”);
          final=SUBSTR(string,position,length);
          counter=counter+1;
       END;
       ELSE counter=counter+1;

       PUT "Exit:” counter position length text;
    END;

RUN;
```

So, from the above, we see that for certain specific problems involving parsing of string, it’s a lot easier to use CALL SCAN to arrive at the required result. CALL SCAN comes with lot of modifiers, like SCAN, that modifies the search process.

For e.g.    Modifier ‘A’ instructs to treat all alphabets as delimiters.

Modifier I ignores the case of the characters.

I shall limit the examples of modifiers here. A complete list can be found in the SAS documentation guide.

## CONCLUSION

We clearly see from the above examples that when it comes to parsing the string with words separated by certain delimiters,  in a specific way where knowing the position is important, it’s easier to do with CALL SCAN than using other character functions to resolve the same.

## REFERENCES

Complete information on CALL SCAN, available at

http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a002255934.htm

## ACKNOWLEDGMENTS

Thanks to the management and all my colleagues for their support and guidance.

## CONTACT INFORMATION

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

Usha Kumar
inVentiv Health Clinical
6th Floor, Building No.4, Commerzone, Survey No. 144/ 145, Airport Road, Yerwada
Pune – 411006, INDIA
+91 20-30569112
usha_cool@hotmail.com

www.inventivhealth.com