

## Determining the Maximum Length of a Column

The **pg2.np\_unstructured\_codes** table contains a single column whose contents include location codes and names. Create a table that efficiently stores the location code and location name.

1. Open the **p203p06.sas** program from the **practices** folder. Submit the program.

- Examine the output report. Notice that the **Column1** column contains raw data with values separated by various symbols. The SCAN function is used to extract the **ParkCode** and **ParkName** values.
- Examine the PROC CONTENTS report. Notice that **ParkCode** and **ParkName** have a length of 200, which is the same as **Column1**.

**Note:** When the SCAN function creates a new column, the new column will have the same length as the column listed as the first argument.

2. Modify the program.

- The **ParkCode** column should include only the first four characters in the string.
- Add a LENGTH statement to define the length of **ParkCode** as 4.
- The length for the **ParkName** column can be optimized by determining the longest string and setting an appropriate length. Create a new column named **NameLength** that uses the LENGTH function to return the position of the last non-blank character for each value of **ParkName**.
- Use a RETAIN statement to create a new column named **MaxLength** that has an initial value of zero.
- Use an assignment statement and the MAX function to set the value of **MaxLength** to the current value of either **NameLength** or **MaxLength**, whichever is larger.
- Use the END= option in the SET statement to create a temporary variable in the PDV named **LastRow**. **LastRow** will be zero for all rows until the last row of the table, when it will be 1. Add an IF-THEN statement to write the value of **MaxLength** to the log if the value of **LastRow** is 1.

```
data parklookup;
    set pg2.np_unstructured_codes end=LastRow;
    ...
    if LastRow=1 then putlog MaxLength=;
run;
```

**Note:** The statements added to determine the maximum length can be deleted or commented.

```
data parklookup;
    set pg2.np_unstructured_codes end=lastrow;
    length ParkCode $ 4;
    ParkCode=scan(Column1, 2, '{}:',"()-'");
    ParkName=scan(Column1, 4, '{}:',"() '");
    retain MaxLength 0;
    NameLength=length(ParkName);
    MaxLength=max(NameLength,MaxLength);
    if lastrow=1 then putlog MaxLength=;
run;

proc print data=parklookup(obs=10);
run;
```

```
proc contents data=parklookup;
run;
```

3. Submit the DATA step. Examine the output data to confirm that the **MaxLength** column sequentially stores the maximum value for **NameLength**. View the log to determine the last value of **MaxLength**.
4. What is the final value of **MaxLength**?

The final value of **MaxLength** is 83.

5. Modify the LENGTH statement to set the length of **ParkName** to the maximum length. Submit the program and confirm in the PROC CONTENTS report that the lengths of the new columns are optimized.

**Note:** The statements added to determine the maximum length can be deleted or commented.

```
data parklookup;
    set pg2.np_unstructured_codes end=lastrow;
    length ParkCode $ 4 ParkName $ 83;
    ParkCode=scan(Column1, 2, '{ }:', "()-'");
    ParkName=scan(Column1, 4, '{ }:', "() '');
    /*      retain MaxLength 0; */
    /*      NameLength=length(ParkName); */
    /*      MaxLength=max(NameLength, MaxLength); */
    /*      if lastrow=1 then putlog MaxLength=; */
run;

proc print data=parklookup(obs=10);
run;

proc contents data=parklookup;
run;
```

6. Examine the PROC CONTENTS output. What is the length of each character column?
  - **Column1** has a length of 200.
  - **ParkCode** has a length of 4.
  - **ParkName** has a length of 83.