

◀ **Step-by-Step Programming with Base SAS**

About This Book

What's New

Accessibility Features of Step-by-Step Programming with Base SAS 9.4

▶ [Introduction to the SAS System](#)

▶ [Getting Your Data into Shape](#)

◀ **Basic Programming**

▶ [Understanding DATA Step Processing](#)

▶ [Working with Numeric Variables](#)

▶ [Working with Character Variables](#)

▶ [Acting on Selected Observations](#)

▶ [Creating Subsets of Observations](#)

▶ [Working with Grouped or Sorted Observations](#)

▶ [Using More Than One Observation in a Calculation](#)

◀ **Finding Shortcuts in Programming**

[Introduction to Shortcuts](#)

[Input File and SAS Data Set](#)

**[Performing More Than One Action in an IF-THEN Statement](#)**

[Performing the Same Action for a Series of Variables](#)

[Summary](#)

[Learning More](#)

▶ [Working with Dates in the SAS System](#)

▶ [Combining SAS Data Sets](#)

▶ [Debugging SAS Programs](#)

▶ [Producing Reports](#)

▶ [Producing Plots and Charts](#)

▶ [Designing Your Own Output](#)

▶ [Storing and Managing Data in SAS Files](#)

▶ [Understanding Your SAS Environment](#)

▶ [Appendix](#)

▶ [SAS Programmer's Guide: Essentials](#)

[SAS Programming Glossary](#)

[Accessibility for Base](#)

[SAS Visual Analytics](#)

[SAS Studio](#)

[SAS Enterprise Guide](#)

[SAS 9.4 Administration](#)

[SAS Viya: Administration](#)


▶ [Other Resources](#)

[Step-by-Step Programming with Base SAS](#)

< >

# Performing More Than One Action in an IF-THEN Statement

Several changes are needed in the observations for Madrid and Amsterdam. One way to select those observations is to evaluate an IF condition in a series of IF-THEN statements, as follows:




```
/* multiple actions based on the same condition */
data updatedattractions;
  set mylib.attractions;
  if City = 'Madrid' then Museums = 3;
  if City = 'Madrid' then Other = 2;
  if City = 'Amsterdam' then TourGuide = 'Vandever';
  if City = 'Amsterdam' then YearsExperience = 4;
run;
```

To avoid writing the IF condition twice for each city, use a DO group in the THEN clause, for example:

```
IF condition THEN
    DO;
        ...more SAS statements...
    END;
```

The DO statement causes all statements following it to be treated as a unit until a matching END statement appears. A group of SAS statements that begin with DO and end with END is called a DO group.

The following DATA step replaces the multiple IF-THEN statements with DO groups:



```
/* a more efficient method */
data updatedattractions2;
  set mylib.attractions;
  if City = 'Madrid' then
    do;
      Museums = 3;
      Other = 2;
    end;
  else if City = 'Amsterdam' then
    do;
      TourGuide = 'Vandever';
      YearsExperience = 4;
    end;
run;

proc print data=updatedattractions2;
  title 'Data Set MYLIB.UPDATEDATTRACTIONS';
run;
```

The following output displays the results.

## Using DO Groups to Produce a Data Set

English

Privacy Statement Terms of Use Copyright © SAS Institute Inc. All Rights Reserved

https://documentation.sas.com/doc/en/pgmsascdc/9.4\_3.5/basess/p0pcj5ajwyngnron1wlsq0tet0hce.htm

1/1