



## CHAPTER

## 4

# Batch Importing

<i>Overview</i>	45
<i>Implementation</i>	46
<i>Specifying the Input Parameters</i>	46
<i>The IMP_TYPE Macro Variable</i>	46
<i>INFILE Macro Variable or Required Filerefs</i>	47
<i>NIDVARS and IDVARn Macro Variables</i>	47
<i>Specifying the Output Parameters</i>	48
<i>Initiating the Batch Import</i>	49
<i>Examples</i>	49
<i>Example 1</i>	49
<i>Example 2</i>	50
<i>File Reference Table</i>	52
<i>Hints and Tips</i>	53

## Overview

The SAS/GIS Batch Import process allows you to use SCL code to import data into SAS/GIS without using the interactive GIS Spatial Data Importing window, or even invoking SAS/GIS. This feature can be useful when you have large amounts of data to import; it lets you set up a batch job to run overnight.

The SAS/GIS Batch Import process allows you to define the values that are needed for the import through macro variables and SAS filerefs. After you define the values, you then call an SCL entry to actually initiate the import. The process has three main steps:

- 1 Specify the input parameters. Include the definitions of the type of data to import, the location of the input spatial data, and any other specifications regarding identification variables (not necessary for all import types). Input parameters are defined either by setting the values of macro variables or by assigning filerefs (depending on the import type).
- 2 Specify the output parameters. Include the library in which the output spatial data sets and catalogs will be stored, name specifications for catalogs, data sets and catalog entries, and whether they will be created, replaced, or updated. All output parameters are defined by setting the values of macro variables.
- 3 Initiate the batch import by executing the SASHELP.GISIMP.BATCH.SCL entry. You do not pass any parameters directly to the SCL entry; the parameters must all have been defined through macro variables and filerefs before you call the SCL entry.

## Implementation

Parameters for the import are specified by assigning a value to a macro variable or by assigning a fileref, as indicated. There are several ways to assign a value to a macro variable, including the %LET statement, the SYMPUT SAS CALL Routine in the DATA step, and the SYMPUT/SYMPUTN function in SCL. As long as the value you want is stored in the macro variable along with the required name, it does not matter which method you use. However, all examples in this section use the %LET statement.

There are also several ways to assign a fileref, including the FILENAME statement, the FILENAME function in SCL, and host-specific file allocation mechanisms. However, all examples in this section use a FILENAME statement.

---

### Specifying the Input Parameters

The input parameters define the type of data to import, the location of the input spatial data, and other specifications regarding variables in your data set (not necessary for all import types).

#### The IMP\_TYPE Macro Variable

You must define a macro variable named IMP\_TYPE to indicate which type of data you are going to import. This parameter is required. Table 4.1 on page 46 contains valid values for IMP\_TYPE.

**Table 4.1** IMP\_TYPE Values and Descriptions

Value	Description
TIGER	Topologically Integrated Geographic Encoding and Referencing (TIGER) files from the U.S. Census Bureau and commercial data vendors
DYNAMAP	Dynamap files from Geographic Data Technology, Inc. (GDT)
DLG	Digital Line Graph (DLG) files from the U.S. Geological Survey and commercial data vendors
DXF	Drawing Interchange Files (DXF) produced by a variety of mapping and CAD software packages
ARC	Uncompressed ARC/INFO E00 transport files from Environmental Systems Research Institute, Inc. (ESRI)
SASGRAPH	SAS/GRAPH map data set format
GENLINE	Generic Line format SAS data set
GENPOINT	Generic Point format SAS data set
GENPOLY	Generic Polygon format SAS data set
MAPINFO	MapInfo Interchange format files from MapInfo Corporation

For example, to import a TIGER file, submit the following statement:

```
%let IMP_TYPE=TIGER;
```

## INFILE Macro Variable or Required Filerefs

You must specify where the input spatial data is located by using either the INFILE macro variable or the required filerefs for your import type.

For the Generic import types and the SASGRAPH import type, you accomplish this by assigning the name of a SAS data set to the INFILE macro variable. You can specify a one-level or a two-level name. One-level names are assumed to be located in the WORK library. For example:

```
/* The CUBA data set in the MAPS library. */
%let INFILE=MAPS.CUBA;
```

or

```
/* The NC data set in the WORK library. */
%let INFILE=NC;
```

For all other import types, you must allocate filerefs to point to the files that you want to import. See “File Reference Table” on page 52 for additional information about the filerefs for each import type. The following list contains the import types and the filerefs that you are required to assign for them.

### TIGER

Assigns the filerefs TIGER1 and TIGER2 for file types 1 and 2, respectively. File types 4, 5, and 6 are optional, these are allocated to the filerefs TIGER4, TIGER5, and TIGER6, respectively. See “Hints and Tips” on page 53 for information about using the optional file types.

### DYNAMAP

Assigns the filerefs GDT1 and GDT2 for file types 1 and 2, respectively. File types 4, 5, and 6 are optional; these are allocated to the filerefs GDT4, GDT5, and GDT6, respectively. See “Hints and Tips” on page 53 for information about using the optional file types.

### DLG

Assigns the fileref DLGIN to the Digital Line Graph (DLG) file that you want to import.

### DXF

Assigns the fileref DXFIN to the Drawing Interchange File (DXF) that you want to import.

### ARC

Assigns the fileref ARCIN to the uncompressed ARC/INFO E00 transport file that you want to import.

### MAPINFO

Assigns the filerefs MID and MIF to the MapInfo Interchange format MID file and the MapInfo Interchange format MIF file, respectively.

## NIDVARS and IDVAR $n$ Macro Variables

*Note:* NIDVARS and IDVAR $n$  macro variables are for SASGRAPH and Generic Import Types only.  $\Delta$

For SASGRAPH and all of the Generic import types, you must provide additional information regarding variables in your data set.

For SASGRAPH and the Generic Polygon import types, you must also identify the number and names of the variables that uniquely identify unit areas in the map, in hierarchical order. The NIDVARS macro variable must be set to the number of

identification variables, and the IDVAR $n$  macro variables must specify, in order, the names of the identification variables. For example, for a SASGRAPH import of the MAPS.USCOUNTY data set (which contains State and County boundaries), you specify:

```
%let IMP_TYPE=SASGRAPH;
%let INFIL=MAPS.USCOUNTY;
%let NIDVARS=2;
%let IDVAR1=STATE;
%let IDVAR2=COUNTY;
```

The value of  $n$  in IDVAR $n$  ranges from 1 to the value that is specified for the NIDVARS macro variable.

For the Generic Line and the Generic Point import types, you can identify a single variable that will be used to generate layer definitions. You set NIDVARS=1 and IDVAR1=*layer variable*. A layer is created for each unique value of the specified layer variable. If you specify more than 16 layers, only the first 16 layers are added to the map. If you want all features to be added to a single layer, specify NIDVARS=0. For example, to add all points from a data set to a single layer, specify

```
%let NIDVARS=0;
```

However, if your points represent stores and you have a SIZE variable that indicates if the store is small, medium, or large, you can specify the following to create three separate layers, one for each value of SIZE:

```
%let NIDVARS=1;
%let IDVAR1=SIZE;
```

---

## Specifying the Output Parameters

The output parameters define the locations where the output data sets and catalog entries are stored and whether they will be created, replaced, or updated. These parameters are required for all import types. The following list contains the macro variables and a description of the information each specifies.

### MAPLIB

Specifies the libref in which the catalog is stored.

### MAPCAT

Specifies the catalog in which the entries are stored.

### MAPNAME

Specifies the name for the map and coverage entries.

### CATHOW

Specifies the action to be used for the catalog entries. Valid values are CREATE, REPLACE, or UPDATE.

### SPALIB

Specifies the libref in which the spatial data sets are stored.

### SPANAME

Specifies the name of the spatial entry and the prefix for the spatial data sets.

### SPAHOW

Specifies the action to be used for the spatial data sets. Valid values are CREATE, REPLACE, or APPEND.

## Initiating the Batch Import

You initiate the batch import by executing the SASHELP.GISIMP.BATCH.SCL entry. You do not pass any parameters directly to the SCL entry; the parameters must all have been defined through macro variables and filerefs before you call the SCL entry. The SASHELP.GISIMP.BATCH.SCL entry can be executed by using any one of the following methods:

Enter the following command from a DMS command line:

```
AF C=SASHELP.GISIMP.BATCH.SCL
```

Submit a DM statement that will issue the AF command:

```
DM 'AF C=SASHELP.GISIMP.BATCH.SCL';
```

Use the CALL DISPLAY routine in SCL:

```
CALL DISPLAY('SASHELP.GISIMP.BATCH.SCL');
```

---

## Examples

---

### Example 1

This example imports the data for Wake County, North Carolina, from a TIGER file, and appends the data for neighboring Durham County, North Carolina, from a separate TIGER file.

```
/* Define the input parameters for Wake County,
North Carolina. */

/* Define the import type. */
%let IMP_TYPE = TIGER;

/* Specify where the TIGER files for Wake County
are located, using the TIGER1 and TIGER2
required filerefs. */
filename TIGER1 '/tgr37183.f51';
filename TIGER2 '/tgr37183.f52';

/* Define the output parameters for Wake County,
North Carolina. */
%let MAPLIB = SASUSER;
%let MAPCAT = TIGER;
%let MAPNAME = COUNTIES;
%let CATHOW = CREATE;
%let SPALIB = SASUSER;
%let SPANAME = COUNTIES;
%let SPAHOW = CREATE;

/* Initiate the batch import by executing the SCL entry. */
DM 'AF C=SASHELP.GISIMP.BATCH.SCL';

/* Define the input parameters for Durham County,
North Carolina. */
```

```

/* IMP_TYPE value stays the same, so you just
   need to reallocate the filerefs to point
   to the spatial data for Durham County. */
filename TIGER1 'tgr37063.f51';
filename TIGER2 'tgr37063.f52';

/* Define the output parameters for Durham County,
   North Carolina. */

/* You want the locations to stay the same,
   so you only need to redefine CATHOW and
   SPAHOW to update the catalog entries and
   append to the spatial data sets for the
   second import. */
%let CATHOW = UPDATE;
%let SPAHOW = APPEND;

/* Initiate the batch import by executing the SCL
   entry a second time, this time to add the Durham
   County data to the Wake County data. */
DM 'AF C=SASHELP.GISIMP.BATCH.SCL';

```

When the import completes, you can open the map named SASUSER.TIGER.COUNTIES. This map displays Wake and Durham counties.

---

## Example 2

This example creates a map of North Carolina with the state and county boundaries and then adds points at city locations. The state and county boundaries are imported by using the SASGRAPH import type, and the points are appended using the GENPOINT import type.

```

/* Construct the data sets to be imported into
   SAS/GIS. The North Carolina state and county
   boundaries are obtained from the MAPS.USCOUNTY
   data set and the North Carolina city locations
   are obtained from the MAPS.USCITY data set.
   Both data sets are supplied with SAS/GRAPH
   software. */

/* Subset just the boundaries for the state of
   North Carolina. */
data sasuser.nc;
  set maps.uscounty;
  /* 37 is the FIPS code for North Carolina.*/
  where state=37;
run;

/* Subset just the cities in North Carolina. */
data sasuser.nccities;
  set maps.uscity;
  /* 37 is the FIPS code for North Carolina. */
  where state=37;
run;

```

```

/* Define the input parameters for the SASGRAPH
import of the boundaries. */

/* Define the import type. */
%let IMP_TYPE = SASGRAPH;

/* Specify where map data set is located. */
%let INFILE=SASUSER.NC;

/* Specify the identification variables, in
hierarchical order. */
%let NIDVARS=2;
%let IDVAR1=STATE;
%let IDVAR2=COUNTY;

/* Define the output parameters for the boundaries. */
%let MAPLIB = SASUSER;
%let MAPCAT = NC;
%let MAPNAME = NC;
%let CATHOW = CREATE;
%let SPALIB = SASUSER;
%let SPANAME = NC;
%let SPAHOW = CREATE;

/* Initiate the batch import by executing the SCL entry. */
DM 'AF C=SASHELP.GISIMP.BATCH.SCL';

/* Define the input parameters for the GENPOINT
import of the cities. */

/* The import type has changed, so redefine the
IMP_TYPE macro variable. */
%let IMP_TYPE=GENPOINT;

/* Specify where the generic point data are located. */
%let INFILE=SASUSER.NCCITIES;

/* Define the number of identification
variables. If you want all of the cities to
be contained in one layer, don't define any. */
%let NIDVARS=0;

/* Define the output parameters for the cities. */

/* You want the locations to stay the same, so
you only need to redefine CATHOW and SPAHOW to
update the catalog entries and append to the
spatial data sets for the second import. */
%let CATHOW = UPDATE;
%let SPAHOW = APPEND;

/* Initiate the batch import by executing the SCL
entry a second time, this time to add the points

```

```
to the boundaries. */
DM 'AF C=SASHELP.GISIMP.BATCH.SCL';
```

When the import completes, you can open the map named SASUSER.NC.NC. This map displays the state and county boundaries for North Carolina. You can select to display the city points on the map.

---

## File Reference Table

Table 4.2 on page 52 lists the reserved filerefs for each of the different import types, a brief description of the file, and whether the fileref is required or optional when using that import type. For example, to import a Digital Line Graph file, you must allocate the file with a fileref of DLGIN.

*Note:* This information can also be found in the SASHELP.GISIMP data set. See “The SASHELP.GISIMP Data Set” on page 40 for more information.  $\Delta$

**Table 4.2** File Reference Table

IMP_TYPE Value	Fileref	File Description	Status
TIGER	TIGER1	TIGER basic data record	Required
	TIGER2	TIGER shape coordinate points	Required
	TIGER4	TIGER index to alternate feature names	Optional
	TIGER5	TIGER feature name list	Optional
	TIGER6	TIGER additional address and ZIP code	Optional
	DYNAMAP	GDT1	Dynamap basic data record
GDT2		Dynamap shape coordinate points	Required
GDT4		Dynamap index to alternate feature names	Optional
GDT5		Dynamap feature name list	Optional
GDT6		Dynamap additional address and ZIPcode data	Optional
DLG		DLGIN	Digital Line Graph file
DXF	DXFIN	DXF file	Required
ARC	ARCIN	Uncompressed ARC/INFO e00 transport file	Required
MAPINFO	MIF	MapInfo Interchange Format MIF file	Required
	MID	MapInfo Interchange Format MID file	Required

For the SASGRAPH and Generic import types, which require that the spatial data be stored in a SAS data set, do not allocate a reserved fileref to indicate the location of the data set. Instead, assign the name of the SAS data set to the INFILE macro variable.



## Hints and Tips

- The SAS/GIS Batch Import process provides less error checking than the SAS/GIS Spatial Data Importing window. Defining invalid values for the input parameters will cause the import to fail.
- The Import interface lets you modify the default composites and the default layer definitions before you proceed with the import. The batch import process does not provide this functionality. If you want to modify the composites and layers before the import occurs, you will need to use the Import interface. However, whether using the Import interface or the Batch Import process, you can always use PROC GIS after the import is complete to make changes to your map and its underlying components.
- The batch import for the TIGER and DYNAMAP import types does not automatically import all of the files, only the required file types 1 and 2. To use the batch import process to import any or all of the optional file types, you need to perform the following steps:

- 1 Copy the SASHELP.GISIMP data set to a location where you have WRITE access. For example, copy the SASHELP.GISIMP data set into the SASUSER library:

```
proc copy in=sashelp out=sasuser;
  select gisimp / mt=data;
run;
```

- 2 Edit the SASUSER.GISIMP data set and change the value of the REQ variable for the TIGER or DYNAMAP file type from 0 (zero) to 1. For example, if you have the TIGER4 and TIGER5 filerefs allocated to the TIGER file types 4 and 5, and you want them to be imported, you could run the following DATA step to change the value of the REQ variable:

```
data sasuser.gisimp;
  set sasuser.gisimp;
  /* Make sure that the values of the
  FILEREF variable are in all upper
  case. */
  if fileref in ('TIGER4' 'TIGER5')
  then req=1;
run;
```

However, you could also use FSBROWSE, FSVIEW, VIEWTABLE, or any other method that you are familiar with to change the value of the REQ variable. Just remember that for the import type that you choose, it will import only from the filerefs for which REQ=1 in the data set.

- 3 For SAS/GIS to use the new SASUSER.GISIMP data set, you must define the USER\_FIL macro variable to point to the name of the new data set. For example:

```
%let USER_FIL=SASUSER.GISIMP;
```

As long as the USER\_FIL macro variable is defined when the SASHELP.GISIMP.BATCH.SCL entry is executed, it will use the current values of REQ to specify which files are imported.

- The SASHELP.GISIMP data set also contains the default librefs for the output catalog entries and spatial data sets. You can modify these defaults by making a copy of the SASHELP.GISIMP data set and changing the values for the DEFMLIB

and DEFSLIB variables to a valid, assigned libref. You then need to define the macro variable USER\_FIL to point to your modified copy of the data set.

- The data set that is specified by the USER\_FIL macro variable is used by both the batch and interactive imports. If you have modified a copy of the SASHELP.GISIMP data set and assigned the data set name to the USER\_FIL macro variable, but you do not want to have that data set used for a specific import, redefine the USER\_FIL macro variable to point to the default data set, SASHELP.GISIMP, before performing an import of either type.
- Throughout this chapter, all of the macro variable names, their values, and all filerefs have been shown in all uppercase for clarity. However, their names and values are not case-sensitive. For example, the following four statements are equivalent:

```
%let imp_type=sasgraph;  
%let imp_type=SASGRAPH;  
%let IMP_TYPE=sasgraph;  
%let IMP_TYPE=SASGRAPH;
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *Working with Spatial Data Using SAS/GIS® Software, Version 8*, Cary, NC: SAS Institute Inc., 1999.

**Working with Spatial Data Using SAS/GIS® Software, Version 8**

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-519-1

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, by any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute, Inc.

**U.S. Government Restricted Rights Notice.** Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

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 registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.