CHAPTER 5

Working with Spatial Data

SAS Data Sets

A SAS data set is a collection of data values and their associated descriptive information that are arranged and presented in a form that can be recognized and processed by the SAS System. SAS data sets may be data files or views. A SAS data file contains the following elements:

- Data values that are organized into a rectangular structure of columns and rows.
- Descriptor information that identifies attributes of both the data set and the data values.

A SAS View contains the following elements:

- Instructions to build the table.
- Descriptor information that identifies attributes of both the data set and the data values.

A third element of the SAS data set is one or more indexes. A SAS index contains the data values of the key variables that are paired with a location identifier for the observation that contains the variable. The value/identifier pairs are ordered in a B-tree structure that enables the engine to search by value.

SAS data sets can be indexed by one or more variables, known as *key variables*. SAS indexes are classified as *simple* or *composite*, according to the number of key variables whose values make up the index.

SAS/GIS Data Sets

As a component of the SAS System, SAS/GIS stores all of its data in SAS data sets. The data sets for a SAS/GIS spatial database work together as one logical entity, although they are physically separated into multiple data sets. SAS/GIS separates the data into the following data sets:

Chains data set
: Contains coordinates for the polylines that are used to form line and polygon features. A polyline consists of either a single line segment of a series of connected line segments. A chain is a sequence of two or more points in the coordinate space. The end points, the first and last points of the chain, must be nodes. Each chain has a direction, from the first point toward the last point. The first point in the chain is the *from-node*, and the last point is the *to-node*. Relative to its direction, a chain has a left side and a right side. Points between the from-node and the to-node are *detail points*, which serve to trace the curvature of the feature that is represented by the chain. Detail points are not nodes.

  The chains data set also lists the from-node and to-node row numbers in the nodes data set, as well as the number of detail points and the corresponding details data set row number. The left and right side attribute values, for example, ZIP codes and FIPS codes, are also stored in the chains data set.

Nodes data set
: Contains the coordinates of the end points for the chains in the chains data set and the linkage information that is necessary to attach chains to the correct nodes. A node is a point in the spatial data with connections to one or more chains. Nodes can be discrete points or the end points of chains. A node definition may span multiple records in the nodes data set, so only the starting record number for a node is a node feature ID.

Details data set
: Stores curvature points of a chain between the two end nodes, which are also called the from-node and the to-node. That is, the details data set contains all the coordinates between the intersection points of the chain. The node coordinates are not duplicated in the details data set. Details data sets also contain the chains data set row number of the associated chain.

Polygonal index data set
: Contains one observation for each polygon that was successfully closed during the index creation process. It is called a polygonal index because each observation is literally an index to each polygon in the chains data set. That is, it points to the starting chain in the chains data set for each of the polygons.

Label data set
: Defines the attributes of labels to be displayed on the map. The attributes include all of the information that is applicable for each label, such as location, color, size, source of the text for a text label, as well as other behavioral and graphical attributes.

Managing Data Set Sizes

By their nature, spatial databases tend to be rather large. Users of spatial data want as much detail in the maps as they can get, which increases the demands on storage and processing capacity. Spatial data that are not carefully managed can become too large for easy use.

Following are five actions that you can take to manage the size of your spatial data sets. You need to perform most of these actions ***before*** importing your data into SAS/GIS.

Reduce the spatial extent of the data.
: Do not store a larger area than you need. If you need a map containing one state, do not store a map containing all the states for a region. For example, if you need to work with a map of Oregon, do not store a map containing all of the Pacific Northwest.

Store only the features that you need.
: If you do not need features such as rivers and lakes, do not store these features in your spatial data.

Limit the amount of detail to what is necessary for your application.
: If you are using a map for which you don't require highly detailed boundaries, reduce the detail level and save storage space. If you are using SAS/GRAPH data sets, you can use PROC GREDUCE to reduce the detail level. If you are using a data set from another source, you'll have to reduce the level of detail before importing the data set into SAS/GIS.

Reduce the number of attributes that are stored with the spatial data.
: If you don't need an attribute, and don't think you will ever need it again, delete it from your spatial data.

Reduce the size of variables that are stored in the spatial data
: Also, you might want to reduce the size of each variable, if possible. That is, examine the method that you use for storing your variables and determine if you can safely reduce the variable size that you use to store them.

  For example, if you have a numeric variable that contains a code that can be a maximum of two digits, perhaps it would be better to store it in a two-digit character variable rather than in an eight-byte numeric variable. Change the variables' defined types or lengths in a DATA step after you complete the import.

Of the five actions, reducing the number of attributes is the easiest to perform. Use the Import window that you can access by selecting [Modify Composites] from the GIS Spatial Data Importing window to remove and drop unneeded composite variables from your data set as it is imported.

Data Set Variables

The following sections describe the variables that are specific to each SAS/GIS data set.

Import Type Specific Variables

The following tables describe the composites and variables that are created for each of the import types. All of the variables are located in the chains data set except for the X and Y variables, which are in the nodes data set.

Table 5.1 Composites and Variables Specific to the ARC/INFO Import Type

Composite	Variable 1	Variable 2	Variable 3	Variable 4	Type1	Description
ARCID	ARCIDL	ARCIDR			A or C	ARCID from the ARC/INFO coverage. Maps made from line and point coverages will not have left and right variables.
ARCNUM					C	ARCNUM from the coverage.
'COVERAGE'	*'COVERAGE'_L*	*'COVERAGE'_R*			A or C	This variable is derived from the input filename. It's the last word preceding the file extension. For example, **/local/gisdata/montana.e00** would have a *'COVERAGE'*2 name of **montana**. The left variable would be **montanal**, the right variable would be **montanar**, and the composite type would be Area. Line and point coverages do not have left- and right-side variables, and the composite type would be Classification.
AREA	AREAL	AREAR			A	AREA from the coverage.
PERIMETE	PERIML	PERIMR			A	PERIMETER from the coverage.

Composite	Variable 1	Variable 2	Variable 3	Variable 4	Type1	Description
'ATTRIB'	'ATTRIB'L	'ATTRIB'R				All variables in the polygon, line, or point attribute tables are saved as composite variables. In the case of the polygon coverages, an L or an R is added to the end of the first five characters of the actual variable name.
COVER	_COVEL	_COVER			A or C	This variable contains the name stored in the *'COVERAGE'* variable.
SRC	_SRCL	_SRCR			C	Contains the string 'ARC'.
X	X				X	X coordinate.
Y	Y				Y	Y coordinate.

1
A = Area
C = Classification
X = X coordinate
Y = Y coordinate

2 Names in single quotation marks, such as 'COVERAGE' and 'ATTTRIB,' are GIS composite names.

Table 5.2 Composites and Variables Specific to the Digital Line Graph (DLG) Import Type

Composite	Variable 1	Variable 2	Variable 3	Variable 4	Type1	Description
LMAJOR(**n**)	LMAJOR(**n**)				C	Major line attribute code.
LMINOR(**n**)	LMINOR(**n**)				C	Minor line attribute code.
NMAJOR(**n**)	NMAJOR(**n**)				C	Major node attribute code.
NMINOR(**n**)	NMINOR(**n**)				C	Minor node attribute code.
MAJOR(**n**)	AMAJORR(**n**)	AMOJORL(**n**)			A	Major area attribute code.
MINOR (**n**)	AMINORL(**n**)	AMINORR(**n**)			A	Minor area attribute code.

X	X				X	X coordinate.
Y	Y				Y	Y coordinate.

1
A = Area
C = Classification
X = X coordinate
Y = Y coordinate

Table 5.3 Composites and Variables Specific to the DXF Import Type

Composite	Variable 1	Variable 2	Variable 3	Variable 4	Type1	Description
’ATTRIB’	’ATTRIB’L	’ATTRIB’R			A or C	All polygon, line, or point attributes are saved as composite variables. In the case of polygon maps, an ’L’ or ’R’ is added to the end of the first seven characters of the actual variable name.

1
A = Area
C = Classification

Table 5.4 Composites and Variables Specific to the Genline Import Type

Composite	Variable 1	Variable 2	Variable 3	Variable 4	Type1	Description
ID	ID				C	The ID variable from the data set.
’ATTRIB’	’ATTRIB’	’ATTRIB’			C	Any other variable in the data set is saved as a classification composite.
X	X				X	X coordinate.
Y	Y				Y	Y coordinate.

1
C = Classification
X = X coordinate
Y = Y coordinate

Table 5.5 Composites and Variables Specific to the Genpoint Import Type

Composite	Variable 1	Variable 2	Variable 3	Variable 4	Type1	Description
ID	ID				C	The ID variable from the data set.
’ATTRIB’	’ATTRIB’	’ATTRIB’			C	Any other variable in the data set is saved as a classification composite.

Composite	Variable 1	Variable 2	Variable 3	Variable 4	Type1	Description
X	X				X	X coordinate.
Y	Y				Y	Y coordinate.

1
C = Classification
X = X coordinate
Y = Y coordinate

Table 5.6 Composites and Variables Specific to the MapInfo Import Type

Composite	Variable 1	Variable 2	Variable 3	Variable 4	Type1	Description
'ATTRIB'	'ATTRIB'L	'ATTRIB'R			A or C	All polygon, line, or point attributes are saved as composite variables. In the case of polygon maps, an 'L' or 'R' is added to the end of the first seven characters of the actual variable name.
LINELYR					C	This variable is derived from the input filename. It's the last word preceding the file extension. For example, **/local/gisdata/montana.mif** would have a LINELYR name of **montana**.
PTLYR					C	This variable is derived from the input filename. It's the last word preceding the file extension. For example, **/local/gisdata/montana.mif** would have a PTLYR name of **montana**.

Composite	Variable 1	Variable 2	Variable 3	Variable 4	Type1	Description
POLYLYR					A	This variable is derived from the input filename. It's the last word preceding the file extension. For example, **/local/gisdata/montana.mif** would have a POLYLYR name of **montana**.
'MAP'	'MAP'				A or C	This variable is derived from the input filename. It's the last word preceding the file extension. For example, **/local/gisdata/usa.mif**, would have a 'MAP' name of **usa**. The left variable would be **usal**, and the right variable would be **usar** and, in this case, the composite type would be **Area**. Line and point maps do not have left- and right-side variables, and the composite would be **Classification**.

1
A = Area

C = Classification

Table 5.7 Composites and Variables Specific to the SAS/GRAPH and Genpoly Import Types

Composite	Variable 1	Variable 2	Variable 3	Variable 4	Type1	Description
'IDVAR'n	'IDVAR'L	'IDVAR'R			A	An area composite variable is created for each ID variable (IDVAR) selected by the user in the ID vars list box. In the case of polygon maps, an 'L' or 'R' is added to the end of the first seven characters of the actual variable name.

1
A = Area

Table 5.8 Composites and Variables Specific to the TIGER and DYNAMAP Import Types

Composite	Variable 1	Variable 2	Variable 3	Variable 4	Type1	Description
ADDR	FRADDL	FRADDR	TOADDL	TOADDR	ADDR	Address range.
BLOCK	BLOCKL	BLOCKR			A	Block number.
CFCC	CFCC				C	Feature classification code.
COUNTY	COUNTYL	COUNTYR			A	County FIPS code.
DIRPRE	DIRPRE				ADDRP	Feature direction prefix.
DIRSUF	DIRSUF				ADDRS	Feature direction suffix.
FEANAME	FEANAME				C	Feature name.
MCD	MCDL	MCDR			A	Minor civil division.
PLACE	PLACEL	PLACER			A	Incorporated place code.
RECTYPE	RECTYPE				C	Record type.
STATE	STATEL	STATER			A	State FIPS code.
TRACT	TRACTL	TRACTR			A	Census tract.
ZIP	ZIPL	ZIPR			A	ZIP code.
BG	BGL	BGR			A	Block group.
LONGITUDE	X				X	Longitude.
LATITUDE	Y				Y	Latitude.

1

A = Area

C = Classification

ADDR = Address

ADDRP = Address Prefix

ADDRS = Address Suffix

X = Longitude

Y = Latitude

Data Set and Catalog Entry Interactions

SAS/GIS software uses SAS catalog entries to store *metadata* for the spatial database-that is, information about the spatial data values in the spatial data sets. SAS/GIS spatial databases use the following entry types.

Spatial Entries

A *spatial entry* is a SAS catalog entry of type GISSPA that identifies the spatial data sets for a given spatial database and defines relationships between the variables in those data sets.

SAS/GIS software supports simple spatial entries and merged spatial entries as follows:

Simple spatial entries contain

- References to the chains, nodes, and details data sets that contain spatial information.
- References to any polygonal index data sets that define the boundaries of area features in the spatial data.
- Definitions for composites that specify how the variables in the spatial data sets are used. See "Composites" on page 67 for more information on composites.
- The definition for a lattice hierarchy that specifies which area features in the spatial data enclose or are enclosed by other features.
- Parameters for the projection system that is used to interpret the spatial information that is stored in the spatial data sets.
- The accumulated bounding extents of the spatial data coordinates of its underlying child spatial data sets, consisting of the minimum and maximum X and Y coordinate values and the ranges of X and Y values.

Merged spatial entries

- Consist of multiple SAS/GIS spatial databases that are linked together hierarchically in a tree structure.
- Contain logical references to two or more other child spatial entries. A child spatial entry is a dependent spatial entry beneath the merged spatial entry in the hierarchy.
- Contain specifications of how the entries were merged (by overlapping or edgematching).
- Do not have their own spatial data sets.
- Reference the spatial data sets that belong to the child spatial entries beneath them on the hierarchy.
- Do not have references to any polygonal index data sets that define the boundaries of area features in the spatial data.
- Do not have definitions for composites that specify how the variables in the spatial data sets are used. See "Composites" on page 67 for more information on composites.
- Do not have the definition for a lattice hierarchy that specifies which area features in the spatial data enclose or are enclosed by other features.
- Do not have parameters for the projection system that is used to interpret the spatial information stored in the spatial data sets.
- Contain the accumulated bounding extents of the spatial data coordinates of their underlying child spatial entries, consisting of the minimum and maximum X and Y coordinate values and the ranges of X and Y values.

For example, you may have two spatial databases that contain the county boundaries of adjoining states. You can build a merged spatial entry that references both states and view a single map that contains both states' counties. Otherwise, you would have to import a new map containing the two states' counties. This new map would double your spatial data storage requirements.

Spatial entries are created and modified using the SPATIAL statement in the GIS procedure.

Note: You can also create a new spatial entry by selecting the following from the GIS Map window's menu bar:
File → Save As → Spatial △

The following additional statements in the GIS procedure also update the information in the spatial entry:

COMPOSITE statement
: Creates or modifies composites that define the relation and function of variables in the spatial data sets. The composite definition is stored in the spatial entry. See "COMPOSITE Statement" on page 89 for details about creating or modifying composites.

POLYGONAL INDEX statement
: Updates the list of available index names stored in the spatial entry. See "POLYGONAL INDEX Statement" on page 95 for details about creating or modifying polygonal indexes.

LATTICE statement
: Updates the lattice hierarchy stored in the spatial entry. See "LATTICE Statement" on page 98 for details about defining lattice hierarchies.

You can view a formatted report of the contents of a spatial entry by submitting a SPATIAL CONTENTS statement in the GIS procedure.

See "SPATIAL Statement" on page 84 for details about using the GIS procedure to create, modify, or view the contents of spatial entries.

Coverage Entries

A *coverage entry* is a SAS catalog entry of type GISCOVER that defines the subset, or *coverage*, of the spatial data that are available to a map. SAS/GIS maps refer to coverages rather than directly to the spatial data.

A coverage entry contains the following elements:

- A reference to the root spatial entry.
- A WHERE clause that describes the logical subset of the spatial data that are available for display in a map.

  Note: The clause WHERE='1' can be used to define a coverage that includes all the data that are in the spatial database. WHERE='1' is called a *universal coverage*. △

  The WHERE clause binds the coverage entry to the spatial data sets that it subsets. The WHERE clause is checked for compatibility with the spatial data when the coverage entry is created and also whenever a map that uses the coverage entry is opened.
- The maximum and minimum X and Y coordinates in the portion of the spatial data that meets the WHERE clause criteria for the coverage.

  These maximum and minimum coordinates are evaluated when the coverage is created. The GIS procedure's COVERAGE CREATE statement reads the matching chains and determines the extents from the chains, the XMIN, YMIN, XMAX, and YMAX variables. If you make changes to the chains, nodes, and details data sets that affect the coverage extents, use the COVERAGE UPDATE statement to update the bounding extent values.

Multiple coverage entries can refer to the same spatial entry to create different subsets of the spatial data for different maps. For example, you could define a series of coverages to subset a county into multiple sales regions according to the block groups that are contained in each of the regions. The spatial data for the entire county would still be in a single spatial database that is represented by the chains, nodes, and details data sets and by the controlling spatial entry.

Coverage entries are created and modified by using the COVERAGE statement in the GIS procedure. You can view a formatted report of the contents of a coverage entry by submitting a COVERAGE CONTENTS statement in the GIS procedure. (The contents report for a coverage entry also includes all the contents information for the root spatial entry as well.)

See “COVERAGE Statement” on page 100 for more information about creating, modifying, or viewing the contents of coverage entries.

Layer Entries

A *layer entry* is a SAS catalog entry of type GISLAYER that defines the set of features that compose a layer in the map. A layer entry contains the following elements:

- A WHERE clause that describes the common characteristic of features in the layer.

  The WHERE clause binds the layer entry to the spatial data even though the WHERE clause is stored in the layer entry. The layer is not bound to a specific spatial entry, just to those entries that represent the same type of data. Therefore, a layer that is created for use with data that are imported from a TIGER file can be used with data that are imported from any TIGER file; however, not all filetypes can take advantage of this behavior. The WHERE clause is checked for compatibility with spatial data when the layer entry is created and also whenever a map that uses the layer entry is opened.

  Note: When you define area layers, you can specify a composite as an alternative to specifying an explicit WHERE clause. However, the layer entry stores the WHERE clause that is implied by the composite. For example, if you specify STATE as the defining composite for a layer, and the STATE composite specifies the following variables: VAR=(LEFT=STATEL,RIGHT=STATER), then the implied WHERE clause that is stored in the layer entry is 'STATEL NE STATER'. △
- Option settings for the layer such as the layer type (point, line, or area), whether the layer is static or thematic, whether it is initially displayed or hidden, whether detail points are drawn for the layer, and the scales at which the layer is automatically turned on or off.
- The graphical attributes that are necessary to draw the layer when it is displayed as a static layer.
- The attribute links, theme range breaks, and graphical attributes if the layer contains themes.

See “LAYER Statement” on page 103 for more information about creating, modifying, or viewing the contents of layer entries.

Map Entries

A *map entry* is a SAS catalog entry of type GISMAP. Map entries are the controlling entries for SAS/GIS maps because they tie together all the information that is needed to display a map. A map entry contains the following elements:

- A reference to the coverage entry that defines the subset of the spatial data that are available to the map. Note that the map entry refers to a particular coverage of the spatial data rather than directly to the spatial entry.
- References to the layer entries for all layers that are included in the map.
- References to any attribute data sets that are associated with the map, for example, the data sets that are used for the map actions, along with definitions of how the attribute data sets are linked to the spatial data.
- A reference to the SAS data set that contains labels for map features.
- Definitions for the actions that can be performed.

- Definitions for map legends.
- Parameters for the projection system that is used to project spatial data coordinates for display.
- Option settings for the map, including the following:
  - The units and mode for the map scale
  - Whether coordinate, distance, and attribute feedback are displayed
  - Whether detail points are read
  - Whether the tool palette is active.

Map entries are created by using the MAP CREATE statement in the GIS procedure. However, much of the information that is stored in the map entry is specified interactively in the GIS Map window.

You can view a formatted report of the contents of a map entry by submitting a MAP CONTENTS statement in the GIS procedure. (The contents report for a map entry includes all the contents information for the spatial, coverage, and layer entries as well.)

See “MAP Statement” on page 119 for details on using the MAP statement.

Composites

For most operations that involve the spatial database, you refer to composites of the spatial data variables rather than directly to the variables in the spatial data sets. A composite consists of the following elements:

- A *variable association* that identifies which variable or variables in the spatial database comprise the association. The variable association can specify a single variable, a pair of variables that define a bilateral (left-right) association, or two pairs of variables that define the start and end of a directional (from-to) bilateral association.
- A *class attribute* that identifies the role of the composite in the spatial database.

For example, if the chains data set has a variable named FEANAME that contains feature names, you can create a composite for the FEANAME variable that assigns the class attribute NAME to indicate that the association represents feature names. Or, if the chains data set has COUNTYL and COUNTYR variables that contain the codes for the counties on the left and right sides of the chains, you can create a composite named COUNTY. The composite identifies the bilateral relationship between these two variables and assigns the class attribute AREA to indicate that the association defines county areas in the spatial data.

Composites are created and modified using the COMPOSITE statement in the GIS procedure. Composite definitions are stored in the spatial entry.

See “COMPOSITE Statement” on page 89 for more information about creating or modifying composites.

Merging Spatial Data

Overview

MERGE is an option of the GIS procedure’s SPATIAL statement that lets you build a new spatial entry by referencing two or more existing spatial entries. The dependent data sets for the spatial entries are not actually combined when you use the MERGE argument; the new spatial entry includes them by reference.

Syntax

The MERGE option syntax follows.

MERGE=(*<libref.catalog.>spatial-entry-1* <, ..., *<libref.catalog.>spatial-entry-n*>)
<EDGEMATCH | OVERLAP>

Note: Keep in mind that MERGE is specified as an option on a SPATIAL statement. △

If you specify a one-level name for any of the entries to be merged, the spatial entry is assumed to be in the catalog that is specified in the CATALOG= argument with the PROC GIS statement or in the most recently issued CATALOG statement. An error occurs if you have not specified a catalog prior to specifying the names of the entries you want to merge.

Types of Merge Operations

The MERGE option accepts the following arguments:

EDGEMATCH
: Locates common boundaries between the merged spatial entries and updates missing left- or right-side composite variable values in the chains data that lie on the boundaries.

In other words, the EDGEMATCH operation compares the chains in the different data sets and finds those chains that map the same feature. When it finds the same chain in both data sets, it replaces any missing left- or right-side composite values in either chain with the valid values from the other data set. EDGEMATCH also creates a merged spatial entry that references other spatial entries (either merged or simple) that you specified with the MERGE option.

EDGEMATCH rewrites the specified chains data sets. You cannot reverse this operation.

OVERLAP
: Merges spatial entries without attempting to match boundaries. OVERLAP is the default behavior of the MERGE argument. The OVERLAP argument creates a merged spatial entry that references the specified spatials entries (either merged or simple) .

OVERLAP does not rewrite the specified chains data sets.

For more information, see “SPATIAL Statement” on page 84.

Benefits of Merging Data

Merging data allows you to construct maps that show larger geographic areas without the overhead of storing duplicate spatial data sets. For example, you may have a chains, nodes, and details data set for each U. S. state. If you want to create a map of New England, you do not have to physically combine and duplicate the individual data sets for the six states composing the region. Instead, you can create a merged spatial entry named `New_England` that references the individual states’ simple spatial entries.

Edgematching provides a mechanism to update adjoining spatial data sets to replace missing left or right values in the chains data sets. Using the New England example, the chains in the chains data set for New Hampshire that lie along the Vermont border contain the FIPS code of 33 on one side of each chain. The other side of each chain has a missing value. The corresponding chains in the Vermont chains data set contain the Vermont FIPS code of 50 on one side and a missing value on the other side. An edgematch merge of the two data sets locates these common boundary chains in each data set and replaces the missing values with the correct FIPS code for the adjoining state. It will also create a merged spatial entry that references the New Hampshire and Vermont simple spatial entries.

The EDGEMATCH operation creates a single-merged spatial entry by which you can create a map of the two states. It also adds the Vermont FIPS code to the appropriate chains in the New Hampshire data set, and adds the New Hampshire FIPS code to the corresponding chains in the Vermont data set.

Sample SAS/GIS Spatial Database

SAS/GIS offers a code sample that creates a fully functional SAS/GIS spatial database. This sample is available in the online help. You can access it by following this path:

Help ▶ SAS System Help

The SAS System Help window opens. From this window select the following:

Sample SAS Programs and Applications ▶ SAS/GIS

▶ Create the Gis Entries for a Simple Demo Map

The SAS Technical Support home page at the following URL http://www.sas.com/service/techsup/intro.html contains a link to the sample map code. You can access the sample map code by following these links from the Technical Support Web page:

Find Your Answer Here ▶ SAS Sample Programs ▶ SAS/GIS

▶ Create the GIS Entries for a Demo Map

You can use this sample map with the SAS/GIS interface and the GIS procedure.

Hints and Tips

1 When SAS/GIS uses a nonuniversal coverage, that is, one in which the value of the WHERE clause is not '1', to subset a map, all of the layers in the map must also satisfy this WHERE clause. If any of the layers do not satisfy this WHERE clause, some features of the map may not be displayed, and the reason may not be apparent.

For example, if you have a map of the United States and you want to create a subset map containing just North Carolina and Virginia, you could use the following COVERAGE statement to create the subset map:

```
COVERAGE CREATE NCVA /
     where='STATEL IN(37 51) OR
              STATER IN(37 51)';
```

Any points or lines that do not have 37 or 51 as the STATEL or STATER value will not display on the map.

2 Defining a layer with WHERE='1' displays all of the features in the underlying spatial data that have that type. For example, if you have a map with a point layer that contains capital cities, and you add a new point layer for grocery store locations by using WHERE='1' for the layer definition, the grocery store layer will display all of the point features in the spatial data. This layer includes capital cities, grocery stores, and all other point features in the spatial data.

You may find this confusing if you are not aware that all point features are being displayed when you intend to display only one layer.

You may encounter this situation because the GENPOINT import, by default, defines all point layers with a WHERE='1' clause. You can use the Modify layers button on the GIS Spatial Data Importing window to redefine the layer definition to be a WHERE clause that uniquely identifies the set of points in the layer. If the layer already exists on the map, you can use the LAYER statement in PROC GIS to redefine the layer with a WHERE clause that defines only those points in the layer.

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

ISBN 1–58025–519–1

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

1st printing, October 1999

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