APPENDIX

1

Details of SAS/GIS Spatial Databases

The SAS/GIS Data Model

SAS/GIS software uses two basic types of data:

Spatial Data
    Describe the location, shape, and interrelationships of map features

Attribute Data
    Provide information that relates to the map features.

Spatial Data Features

SAS/GIS software uses spatial data to represent the following three types of map features:

Point Features
    Consist of individual locations that are shown as symbols, representing real-world locations of special points of interest.

Line Features
  Consist of sequences of two or more coordinates that form zero-width shapes, either closed or unclosed. Line features represent entities that either have no width, such as political boundaries, or those that can be represented as having no width, such as streets or water pipes.

Area Features
  Consist of sequences of three or more coordinates that form polygons (with single or multiple boundaries and with or without holes.) Area features represent two-dimensional entities such as geographic areas (countries, states, and so forth) or floorplans for buildings.

SAS/GIS Topology

To represent point, line, and area features in the map, SAS/GIS software defines the following topological features in the spatial data:

Chains
  Are sequences of two or more points in the coordinate space. The end points (that is, the first and last points of the chain) are 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, each 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.

Nodes
  Are points in the spatial data coordinate space that have connections to one or more chains.

Areas
  Are two-dimensional finite regions of the coordinate space. One or more chains, called *boundary chains*, separate two different areas. Chains that lie completely inside an area are called *internal chains* and are bounded on the left and right sides by the same area.

The spatial data coordinate space can be represented in any numeric units even those that include arbitrary values. Coordinates that are stored as longitude and latitude values have a maximum usable precision of about one centimeter.
  Representations of map features are implemented with one or more chains, as follows:

Point features
  Are implemented with one chain, one node (that is, the from-node and to-node for a point feature are the same node), and no detail points.

Line and Area Features
  Are implemented with one or more chains and one or more nodes.

Rules for Topological Correctness

SAS/GIS spatial data must obey the following rules in order for the topology to be correct. These rules are similar to the rules for TIGER/Line files from the U.S. Bureau of the Census. For more information on these rules, see Gerard Boudriault's 1987 article, "Topology in the TIGER File" in *AUTO-CARTA 8, Proceedings*, pages 258-263, published by the American Society for Photogrammetry and Remote Sensing and the American Congress on Surveying and Mapping.

Rule: Topological Completeness

All chains must be

☐ Bounded by two nodes, the from-node and the to-node

> *Note:* In chains for point features and for single-chain closed-loop line features or area boundaries, the from-node and the to-node are the same node, but both are still included in the chain definition. △

☐ Bounded by two areas, one on the left and one on the right.

These relationships must be complete, so the following two rules apply:

☐ The sides of all the chains incident to any given node must form a cycle. A *cycle* consists of one or more chains that start or end at the same node.

☐ The end points of chains that bound an area must form one or more disjoint cycles.

For each unique area ID or unique set of area IDs, all the boundary chains that have the ID value (either on the right or left, but not both) form one or more closed loops or cycles.

Rule: Topological-geometric Consistency

The collection of chains, nodes, and areas must have coordinates that make the collection a disjoint partitioning of the coordinate space. The following four conditions must be true to avoid problems with displaying the spatial data:

☐ No two points in the combined set of nodes and detail points share the same coordinate.

☐ No two line segment interiors share a common coordinate.

☐ No two areas share a common coordinate.

> *Note:* Graphically overlaid data may have overlapping polygons, chains, and nodes and have no topological interconnectivity △

☐ Polygons that form the boundaries of holes inside areas must fall completely within the enclosing areas.

Note: Edge-matched data share coordinates along the common boundaries, but each chain should have the proper polygonal ID values on the side that represents the outside edge of their respective physical coverages as well as the inside edges. △

Problems Resulting from Topological Errors

Topological errors in the spatial data cause the following types of problems:

☐ A polygonal index cannot be built for all the polygons for a particular area set.

☐ A successfully indexed polygon does not close because

> ☐ the chains for a node do not form a cycle, which is sometimes the result of left- and right-side values being swapped for one or more of the connected chains

> ☐ a chain crosses another chain's interior coordinated space.

☐ Multiple features are selected when only one selection is desired because of overlapping features in a coordinate space.

☐ Select Like Connected processing fails to select apparently connected chains.

Attribute Data

Attribute data are all other data that are related to map features in some way, including the data that you want to analyze in the context of the map. Attribute data can be stored in the spatial database by the following methods:

    ☐ Directly with the spatial data as variables in the chains data set

    ☐ Indirectly in SAS data sets that are joined to the chains data set by a link that is composed of one or more variables.

Attribute data can be used as follows:

☐ As themes for map layers.

☐ By *actions* that display or manipulate the attribute data when features are selected in the map. Actions can be defined to display the attribute data, create new SAS data sets that contain subsets of the attribute data, or submit SAS programs to process the attribute data.

SAS/GIS Spatial Database Structure

A SAS/GIS spatial database consists of a set of SAS data sets that store the spatial data and a set of SAS catalog entries that define the functions of, and the relationships between, the spatial data elements.

Spatial Data Sets

As a component of the SAS System, SAS/GIS software stores all its spatial data in SAS data sets. The data sets for a SAS/GIS spatial database work together as one logical file, even though they are split into multiple physical files.

The spatial data sets implement a network data structure with links that connect chains to their two end nodes and each node to one or more chains. This structure is implemented by using direct pointers between the nodes and chains data sets. The details data set provides curvature points between nodes of chains, while the polygonal index data set provides an efficient method of determining the correct sequence of chains to represent polygons.

Common Spatial Data Set Variables

The following spatial data variables appear in the chains, nodes, and details data sets:

Variable	Description
ROW	row number (used as a link when the spatial data set is used as a keyed data set as well as for database protection)
DATE	SAS datetime value when the record was last modified
VERSION	data version number
ATOM	edit operation number
HISTORY	undo history record pointer

Variable Linkages in the Spatial Data

The following linkages exist between and within the spatial data sets:

Data Set	Variable	Links to...
chains	ROW1	self
	FRNODE	starting from-node record in the nodes data set

Data Set	Variable	Links to...
	TONODE	starting to-node record in the nodes data set
	D_ROW	starting detail record in the details data set
nodes	ROW	self
	C_ROW1-C_ROW5	chain records in the chains data set
	NC2	node record in the nodes data set used to store additional chain records
details	ROW	self
	C_ROW	parent chain record in the chains data set or next detail continuation record in the details data set
index3	C_ROW	starting chain record in the chains data set

1 The ROW variable is used as a link between records in the spatial data sets. The ROW variable value for the first record of a feature in the chains or nodes data sets is considered the *feature ID*. Because some records in the nodes data set are continuations of other records, not every row number in the nodes data set is a feature ID. As a result, node feature ID numbers are not necessarily sequential.The ROW variable also provides protection against corruption of the database that is caused by the accidental insertion or deletion of records. If records were linked by physical record number rather than by ROW value, an improper record insertion or deletion would throw off all linkages to subsequent records in the database. In the event the database is corrupted, the ROW variable can be used to move the records back into their proper locations with minimal data loss.

2 A negative value indicates that the variable points to a continuation record. The absolute value of the variable is the row number of the next record used to that feature's data. In newly imported spatial data, continuation records always point to the next record in the data set, but this is not required. New chains can be attached to existing nodes without having to insert records, which would require extensive printer reassignments.

3 The index data set has no ROW variable because it can be easily rebuilt from the chains, nodes, and details data sets from which it was originally constructed.

Because the data sets are linked together by row number, the chains, nodes, and details data sets must be radix-addressable and may not be compressed.

Chains Data Set

The 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 or a series of connected line segments.) The chains data set also contains the information that is necessary to implement nodes in the database.

The following system variables are unique to the chains data set:

Variable	Description
FRNODE1	starting from-node record.
TONODE	starting to-node record.
D_ROW	starting detail point record.
ND	number of detail points in the chain.

Variable	Description
RANK	sorting key used to sort all the chains around an arbitrary node by their angle, starting from 0 and proceeding counter-clockwise.
	RANK values have the form *ffffff.tttttt*, where the *ffffff* component is used to sort the chain around its from-node and the *tttttt* component is used to sort the chain around its to-node. The *ffffff* and *tttttt* components are calculated using the following formula:

$$R=1E5*((Q-1)=\tan(A/2))$$

where

	R	is the calculated ranking factor.
	Q	is the quadrant (1 to 4) that contains the angle α for the chain. For the *ffffff* component, α is defined by the vector $F \rightarrow D_0$, where F is the from-node and D_0 is the first detail coordinate. For chains with no detail coordinates, D_0 is the to-node. For the *tttttt* component, α is defined by the vector $T \rightarrow D_L$, where T is the to-node and D_L is the last detail coordinate. For chains with no detail coordinates, D_L is the form-node.
	A	is $\alpha - (Q-1)*\pi/2$.

The tangent term is called the half-angle tangent. Since the angle A/2 can never exceed $\pi/4$ (45 degrees), the half-angle tangent has values from 0 to 1. The (Q-1) multiplier adjusts the range of values to 0 to 4. The values 0, 1, 2, 3, and 4-ε represent angles of 0, 90, 180, 270, and just under 360 degrees, respectively.

The 1E5 multiplier is used to transform decimal rank values to integers. Thus the rank values for a chain have six significant digits.

Variable	Description
XMIN	mininum X coordinate of chain.
XMAX	maximum X coordinate of chain.
YMIN	minimum Y coordinate of chain.
YMAX	maximum y coordinate of chain.

1 The TONODE and FRNODE variables can point to the same record.

The XMIN, YMIN, XMAX, and YMAX variables define a bounding box for the chain. These variables are included in the chains data set to make it possible to select all the chains in a given X-Y region by looking only at the chains data set.

In addition to the system variables, the chains data set may also contain any number of attribute variables, some of which may be polygon IDs. Since the chains have sides, there are typically paired variables for bilateral data such as polygon areas or address values. The names of the paired variables typically end with **L** or **R** for the left and right sides, respectively. For example, the data set may contain COUNTYL and COUNTYR variables with the codes for the county areas on the left and right sides of the chain, respectively. However, this naming convention is not required.

Nodes Data Set

The nodes data set contains the coordinates of the nodes for the chains in the chains data set and the linkage information that is necessary to attach chains to the correct nodes. 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.

The following system variables are unique to the nodes data set:

Variable	Description
C_ROW1- C_ROW5	Chain records for the first five chains connected to the node. If fewer than five chains are connected to the node, the unused variables are set to 0.
NC	number of chain pointers (if five or fewer chains are connected to the node) or the negative of the next continuation node record number (if more than five chains are connected to the node). See "Variable Linkages in the Spatial Data" on page 138 for more information about how NC is used to string continuation node records.
X	X coordinate of node.
Y	Y coordinate of node.

Details Data Set

The details data set stores curvature points of a chain between the two end nodes. Therefore, it contains all the coordinates between the intersection points of the chains. The node coordinates are not duplicated in the details data set.

The following system variables are unique to the details data set:

Variable	Description
C_ROW	parent chain record (if the chain has ten or fewer detail points) or the negative of the next continuation detail record (if the chain has more than ten detail points). See "Variable Linkages in the Spatial Data" on page 138 for a description of how C_ROW is used to string continuation detail records.
X1-X10	X coordinates of up to 10 detail points.
Y1-Y10	Y coordinates of up to 10 detail points.

Detail coordinate pairs (X2, Y2) through (X10, Y10) contain missing values if they are not used. The missing values ensure that the unused coordinate pairs are never used in any coordinate range calculation. The various importing methods set unused detail coordinates to missing as a precautionary measure.

Polygonal Index Data Set

Polygonal indexes are indexes to chains data sets. The index contains a record for each boundary of each polygon that was successfully closed in the index creation process. The same rules that are used to construct polygons are also used to construct polygonal indexes.

The following system variables are unique to polygonal index data sets:

Variable	Description
C_ROW	starting chain from which a polygon can be dynamically traversed and closed. This chain is sometimes referred to as the *seed chain* for the polygon. Any chain on a polygon's boundary can be the seed chain.
FLAGS	control flag for polygons.
NC	number of chains in the polygon boundary.

Polygonal index data sets are created with the POLYGONAL INDEX statement in the GIS procedure. See "POLYGONAL INDEX Statement" on page 95 for more information about using the GIS procedure to create polygonal index data sets.

Catalog Entries

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 composite associations that specify how the variables in the spatial data sets are used. See "Composites" on page 145 for more information.
- □ The definition for a lattice hierarchy that specifies which area features in the spatial data enclose or are enclosed by other features.
- □ The parameters for the projection system that are used to interpret the spatial information that is stored in the spatial data sets.
- □ The bounding extents of the spatial data coordinates, 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 is a dependent spatial entry beneath the merged spatial 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 about 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 that is stored in the spatial data sets.
- □ Contain the accumulated bounding extents of the spatial data coordinates of their underlying child spatials, 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 then you can view a single map containing both states' counties. Otherwise, you would have to import a new map that contains the two states' counties. This new map would double your spatial data storage requirements.

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

Note: You can also create a new spatial entry by making the following selections from the GIS Map Window's menu bar:

| File | ► | Save As | ► | Spatial |

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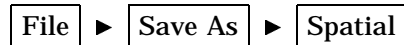
The following other statements in the GIS procedure also update the information in the spatial entry:

COMPOSITE statement
Creates or modifies composites that define the relation and function of variables in the spatial data sets. See "COMPOSITE Statement" on page 89 for details on using the GIS procedure to create or modify composites.

POLYGONAL INDEX statement
Updates the list of available index names that are stored in the spatial entry. See "POLYGONAL INDEX Statement" on page 95 for details about using the GIS procedure to create or modify polygonal indexes.

LATTICE statement
Updates the lattice hierarchy that is stored in the spatial entry. See "LATTICE Statement" on page 98 for details about using the GIS procedure to define 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 more information 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 is 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. (The clause WHERE=1 can be used to define a coverage that includes all the data in the spatial database.)

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 meet 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' XMIN, YMIN, XMAX, and YMAX variables. If you make changes to the chains, nodes, and details data sets

that affect the coverage extents, you should 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 county would still be in a single spatial database that is represented by the chains, nodes, and details data sets and 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 using the GIS procedure to create, modify, or view 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 it is stored in a separate entry. The layer is not bound to a specific spatial entry, just to those entries representing the same type of data. Therefore, a layer that is created for use with data that were imported from a TIGER file can be used with data that were imported from any TIGER file. 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 defining area layers, you can specify a composite association as an alternative to specifying an explicit WHERE clause. However, the layer entry stores the WHERE clause that is implied by the composite association. For example, if you specify STATE as the defining composite association for a layer, and the STATE composite association has the variable association VAR=(LEFT=STATEL,RIGHT=STATER), then the implied WHERE clause that is stored in the layer entry is WHERE 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 if it is 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 using the GIS procedure to create, modify, or view 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, 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 when map features are selected.

□ 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 about the items that can be specified with the GIS procedure. See Chapter 10, "SAS/GIS Windows" in the *SAS/GIS Software: Usage and Reference, Version 6* for details about the items that can be specified interactively in the GIS Map window.

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 compose 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 that is 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 composite represents feature names. 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 that identifies the bilateral relationship between these two variables and assigns the class attribute AREA to indicate that the composite defines county areas in the spatial data.

Composites are created and modified by 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 using the GIS procedure to create or modify composites.