

KANSAS GEOLOGICAL SURVEY
OPEN-FILE REPORT 88-45b

The GIMMAP System

by

Charles G. Ross

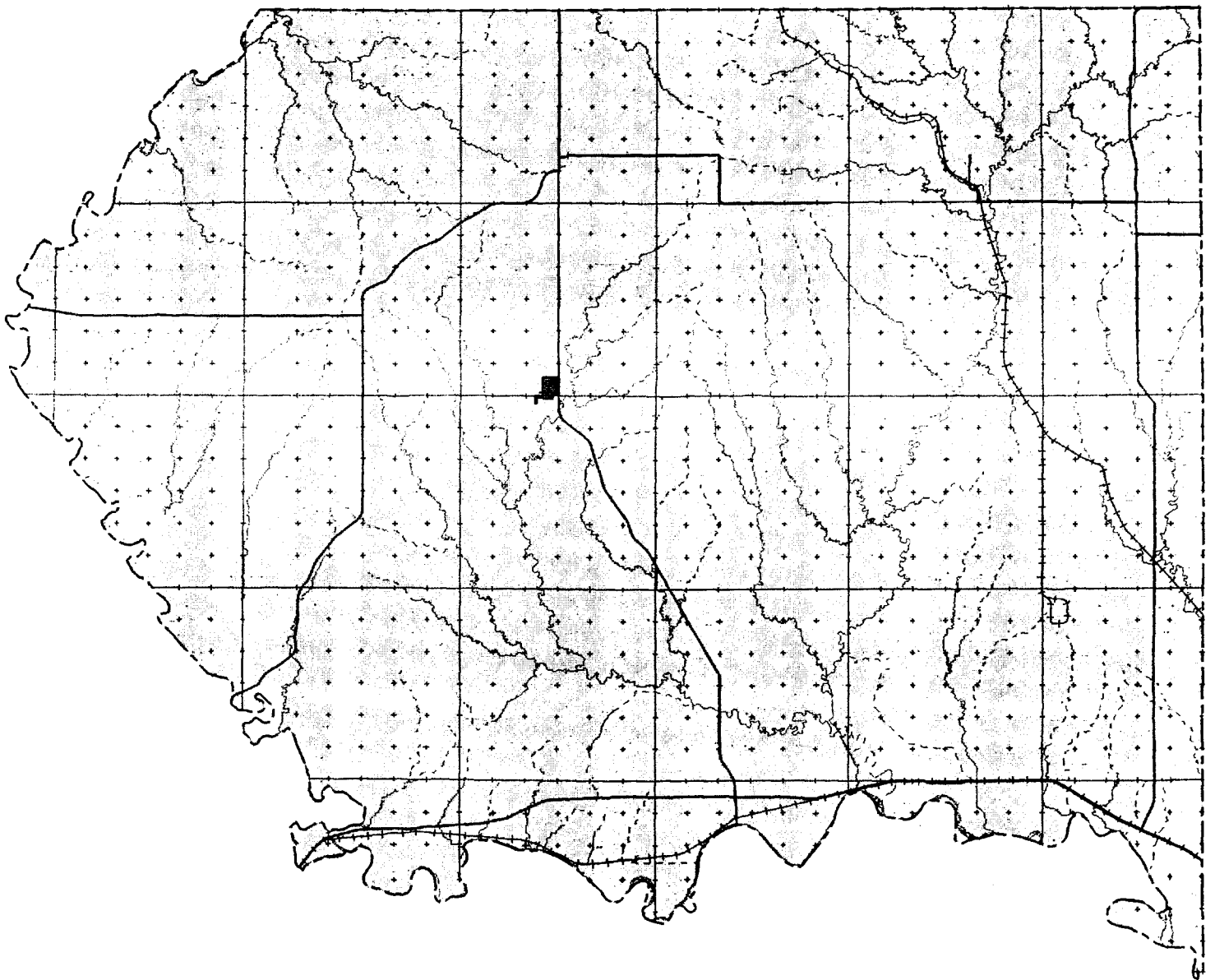
Disclaimer

The Kansas Geological Survey does not guarantee this document to be free from errors or inaccuracies and disclaims any responsibility or liability for interpretations based on data used in the production of this document or decisions based thereon. This report is intended to make results of research available at the earliest possible data, but is not intended to constitute final or formal publications.

KANSAS GEOLOGICAL SURVEY
1930 Constant Avenue
University of Kansas
Lawrence, KS 66047

The GIMMAP System

by Charles G Ross



KANSAS GEOLOGICAL SURVEY

TO:

Joe's mom, Gene, Van, and Bear

"the best part about writing is the finishing"

(quoted by Steven Spielberg)

"All good writing is swimming under water
and holding your breath."

F. Scott Fitzgerald

Foreword

The compilation of this information about the Geodata Interactive Management Map Analysis and Production or GIMMAP ("jim-map") system has been a long process, beginning with the first programs written in 1977. As with the conquest of the great mountains, there have been many different attempts made over the years. This set of separate volumes has finally resulted from the union of the efforts of those people involved directly in the creation and use of GIMMAP with the support of the administration of the Kansas Geological Survey.

First, I must acknowledge the people who have contributed to the completion of this work.

I thank my wife, Gina, and my children, Catherine and Patrick, who are my reason for living. And I must credit Gina, whose love and encouragement helped in many ways to bring me to the light at the end of this tunnel.

Trang Cao initiated GIMMAP in Orleans, France while he was working for the BRGM there. He brought ideas and the foundation for the system when he was Visiting Research Scientist in the 1977-78 academic year. His ideas and good will in working with me made the potential for an automated cartography system a reality at the Survey. His work is a shining example of what can be.

David Collins helped to provide momentum, direction and many, small packets of inspiration, and helped greatly to organize and edit the final effort.

Finally, this work would not have been completed without the commitment to the definition and prioritization of the project by Dr. Lee Gerhard, Director of the Kansas Geological Survey.

Creation of these four volumes of GIMMAP documentation has been a long and arduous process, marred by difficulties which might have been avoided. Important lessons have been learned about the process of creating software documentation and I would feel remiss if some of these were not passed on to. The foreword may not be the best place to air these thoughts, but I feel they are of a fundamental importance.

Of the lessons learned, these might be the most important:

1. The first step is to clearly define in writing the task which is to be done. This includes identifying the target audience, the purpose and function, and the form and scope of the documentation.
2. For the job to be well-defined and well done, a group of people must participate in overseeing and performing the task at all stages. This group should consist of administrators, people who use or benefit from the software, the software author(s), and experts in software, education, technical writing and marketing. Strong and generous support for this group and the documentation authors by the administration in organization, project definition and the provision of resources is required for success.
3. The software author is probably not the best nor even a good choice to be the author of the documentation. The skills required to write software documentation are not the same as those required to write computer software. Generally and historically, an experienced writer with knowledge of the software is the best choice. The software author is the best source of technical information on the software, prepared as draft material for the software documentation during implementation of the software. Comprehensive documentation of a complex software system such as GIMMAP can not be properly prepared in scattered pieces of time by software authors who have too many responsibilities in creating, expanding and maintaining the software.
4. The group of experts (above) should estimate the cost of the project if it is to be done right. Money, time, personnel and other resources should be discussed and consensus reached. To do this, clear definition of the priorities of all affected personnel must be made. Overlapping expectations coupled with improperly low estimations of the size of the task have led to disappointments in the past.
5. The reason(s) for writing the software in the first place should be well understood and accepted by the group as the foundation of the discussions implied above.

- D R A F T -

I. I N T R O D U C T O R Y I N F O R M A T I O N

I.2 THE GIMMAP VIEW OF MAP DATA

Map data may be viewed and modeled from varied perspectives, with many different objectives in mind. Maps present many different kinds of information in myriad forms. The hardware and software systems built for automated cartography amplify and expand the options for analysis and display of map data.

The elements of map data and their inter-relationships - the elemental features found on maps and the ways they relate to each other by spatial and non-spatial measures - describe or define a view of map data. The view of map data then fixes the foundation and framework of the cartographic database. This framework becomes the data structures (see I.5 on Data Structures) and hence the complete specification of the files comprising the cartographic database. Definition of the database then defines access to the database for the various components, their attributes and their relationships. Access then determines how all software gets, puts and changes the digital model or representation of the map data.

For GIMMAP, the view of map data is one which is intuitively consistent with the three dimensional objects of elementary geometry, but which also provides the topological structures required for analysis, networks and pathfinding, and area adjacency (neighbor) and parent/child (island) analysis. The GIMMAP view is limited to a simple set of feature classes and relations, supporting a rich assortment of map types, styles and objectives.

Map features are classified and distinguished essentially by the dimensionality of the feature, with three possible classes of objects: (isolated) points, lines (arcs), and areas (zones).

Isolated Points

A point (0-dimensional) feature is one which is generally represented on maps by a single symbol identifying a unique object or feature existing at a single location. In the GIMMAP view, such features are called **isolated points**. Each isolated point has a (generally) unique location and a primary **attribute** defining the point feature type such as well or building. This attribute is usually portrayed visually on the map by the type, color or size of the symbol used to represent the feature.

Other features which are isolated points are city centers, data points with locations, section or township corners, radio towers, benchmark locations, earthquake epicenters, locational

control points, mine shaft entrances, mountain peak elevation points and many other kinds of single point features.

Isolated point features may occasionally have additional attribute values associated with the location. These values may be posted on the map, may be incorporated into the type, color or size of the symbol representing the feature, or may reside elsewhere, even in a digital file. These extra attribute values are called **z-values**, because they often represent attribute values in a third dimension (Z), assuming the location (X and Y) to be the first two dimensions. They are often used for contouring or representing a 3-dimensional surface. In GIMMAP, there may be up to seven numerical z-values associated with any isolated point feature.

Feature Codes

The different types of isolated points, symbolized on the map by different shapes, sizes and colors of point symbols are distinguished in the cartographic database by the use of unique identifying numbers called **feature codes**. Each different kind of isolated point (or other) feature on the map is assigned a four digit feature code by which such feature may be singled out for some use or may be ignored for others. Feature codes are used to maximize retrieval, processing and display speeds and to minimize clutter in operations requiring good, fast views of data, such as interactive editing.

The feature codes later provide the ability to separate map features for assignment of pen widths, types and colors at the time of map production. Feature codes also allow restriction of analysis and construction operations (e.g. ZONGEN) to a specific desired subset of map features.

Although feature codes may generally be assigned arbitrarily to features, there are two basic restrictions which do apply in GIMMAP. First, valid feature codes must be in the range 1000 to 32,767, though codes above 9,999 will be treated in special ways by some programs. Secondly, codes for isolated point features must be in the range 1000 to 1999 (or 11000 to 11999 assuming the warning above) and linear arcs are coded 2000 to 9999 (or 12000 to 19999 as per above). Misuse of these restrictions will result in isolated points being joined as a line and lines being broken into disconnected points.

A special class of feature codes, in which the normal code is incremented by 10,000 is reserved by some programs to produce special effects. These "ten-thousand-plus" codes are intended to signal "edit" or "replacement" features which have been entered to replace existing features or to aid the process of editing in some other way. Such features are displayed with a different line type or color to allow clear comparison to older features so

that parts or all of the new features may be used to replace the older features. In ten-thousand-plus codes, the 11000 to 11999 codes are restricted to isolated point features and the 12000 to 19999 range is for linear arcs.

Arcs

Features which are linear (one-dimensional) in nature are called (linear) **arcs** in GIMMAP. Linear features on a map may end in intersection with the map edge, may terminate in the interior of the map, or may end as the result of intersection with another linear feature. Each linear feature has two endpoints and a path of (infinitely many) points between the endpoints. In the GIMMAP view, the endpoints of these features are called **nodes**, whether they are at the map edge, a dead-end in the interior, or at the intersection of two linear features.

Arcs are the parts of linear features between nodes. If the linear feature has only two nodes at its ends then the whole feature is a single arc. If the linear feature is broken by intersection to have three or more nodes, then each piece between nodes is a separate arc. The points lying on path of the linear feature which are selected (digitized) to represent the arc path are called the **interior points**. Unlike isolated points, interior points have importance only with respect to the arc which they describe. Furthermore, short, straight arcs may have no interior points since their paths are well described by their endpoints.

Many features on a map may be considered to be (linear) arcs. Base or general maps may contain roads and highways, city and county boundaries, streams and rivers and lakes, railroads, utility lines and pipelines, park and cemetery boundaries, hiking and biking trails, reference system lines (e.g. the township and range lines of the public lands survey systems), airport runways and boundaries, and countless other linear features or boundaries of areal features.

Linear arcs represent features which have a length along a linear path, but whose width is inconsequential at the scale of the map. However, it is necessary that some of these features are symbolized by line types which represent far greater width according to scale than the feature actually has. Railroads are usually represented with hashed lines and roads are double lines or wide solid lines. In both cases, the symbol has a width many times in excess of the features themselves. The assumption is that for the input (digitizing) side as well as the output side (plotting), the feature lies at the center of the map line.

Feature codes are once again used to distinguish the various types of arcs. Usually, a code between 2,000 and 9,999 is used to uniquely identify each kind of linear arc used on a map. A set of feature codes is usually kept for the whole of the system

rather than for each mapping project. For most installations, a single list provides consistency and familiarity, and is possible as long as the complete range of available feature codes is not required. The ten-thousand-plus feature codes may also be used for linear arcs to support "edit" or "replacement" operations.

Nodes

The endpoints of arcs where the arcs terminate at the map edge, in the map interior, or at an intersection with another linear feature are called **nodes**. The definition used in GIMMAP is further restricted by limiting which intersections are viewed as nodes and which are ignored. Such restriction is useful because not all intersections are of value to GIMMAP, and there is a price to pay (memory, disk space and time) for recognizing more intersections as nodes. The restriction may be imposed in any way advantageous to the user.

For this installation, intersections between lines of like or similar type (e.g. state with county lines or rivers with lakes) are recognized as nodes. Intersections of unlike type are not recognized as nodes unless needed for other operations. For example, intersections of state or county lines with any other lines are treated as nodes. This is because these lines, as well as quadrangle map boundaries, are used as the map boundaries for the most basic map unit (the **county patch**) in the system. Thus, any lines intersecting state or county lines are terminating at the map border.

This kind of operation also allows for rapid retrieval of map data based on these basic units as well as on individual counties and on a statewide basis in a larger system. Another example is that intersections between geologic contact lines and state lines would be treated as nodes because construction of a statewide geology map requires the state line to act as formation boundaries in the map production stage. Restriction of the node definition also results from the trade-off between the quantity of nodes and the value of the network containing them.

Isolated points in GIMMAP are treated as "collapsed" arcs with respect to nodes. That is, they may be seen as arcs with no interior points, collapsed so that both endpoints are the same. All isolated points must be treated as nodes so that they may be uniquely numbered, so operators may reference them. Generally, isolated points are unique arcs and unique nodes, but it is possible that the node also represents the intersection of two or more linear arcs along with the isolated point (e.g. a township corner and the intersection of a township and a range line).

The important aspect of the node is its relationship to other nodes in the network. Each node is connected to one or more other nodes by arcs, except for isolated point nodes which

may be connected only to themselves. Connections that do exist between nodes are represented by arcs, even for isolated points. The total number of node-arc connections for a single node is called the **degree** of the node, with a minimum value of one for isolated points, for linear arcs ending in the interior with no intersections, and for circular arcs beginning and ending at the same node. The degree of the node is the number of arcs which are incident at the node, with the restriction listed above.

Zones

Map features which are areal (2-dimensional) in nature are defined by a linear boundary which separates the interior of the area from the rest of the map (exterior). Areal features in the GIMMAP view of map data are called **zones**, which are the logical equivalent of polygons in other settings or other views of map data. The zone is essentially the interior area, not including the boundary line or any other zones which may exist in the interior of the zone.

The **zone border** or **zone boundary** consists of one or more (linear) arcs joined at node locations to define the boundary line between the interior and the exterior of the zone. Though the arcs for a given border may not all be represented in the counter-clockwise (in GIMMAP) representation of the border, the zone border contains appropriate information which allows GIMMAP to recognize each arc's direction with respect to the original direction of digitization. The entire border may then be produced as a continuous sequence of points from a beginning node along the boundary in a counter-clockwise fashion and returning back to the start node.

Important facets of zones include the boundary sequence of arcs, the neighbor zones on the "other" (exterior) sides of the arcs in the border, the major attribute for which the area or zone is defined, the color of the area as represented on the map, and the **zone color** which is a number identifying the major attribute of the zone and which defines the color or shading patterns used in map production when the area is selected.

It is possible that one or more zones may be completely surrounded by a zone. These zones are called **island zones** or **simple island zones**. Simple islands usually consist of a single arc for a border, and have the **surrounding zone** as their only neighbor. Hence, all zones have the potential of having other relationships beyond the (symmetric) neighbor relationship. A zone may be an island of a surrounding zone (and may be related to other islands of the same surrounding zone), or may be the surrounding zone to one or more islands. Island zones may also be surrounding zones to other islands.

Symbology

Maps usually contain written information in many different places for many different purposes. Maps usually have titles and often have legend boxes with labels identifying area colors or patterns, line types, and point symbols with values or feature type information. Paragraphs may exist to describe the making of the map or the content. Names are placed on the map to identify features such as rivers or towns. Labels may be used to better identify colored or shaded areas such as geologic formations. Information about the map scale and projection, directions and errors may be included. For many reasons, textual information is added to a map.

In the GIMMAP view of map data, such information is referred to as **symbology** and is not captured from original documents for inclusion on subsequent map products. Rather, individual pieces of text, called **symbols** are created on a graphics display screen interactively, directly on the image of the captured digital data for the map. A symbol is variably-sized, with options for the location, size (character height), font (character style), angle of orientation, content, and (left, right, center) justification with respect to the chosen point of location. Each symbol is associated with a string of points defining the path of plotting to produce the desired symbol. This set of points comprises the calligraphy of the symbol.

Miscellaneous

All printed maps of portions of the Earth's surface require a transformation of the real surface to a scaled, flat image of the map. The conversion of locations at scale on the surface of the Earth to the map is called a **map projection**. There are many different forms of projection used to make maps. In fact, there are infinitely many possible variations.

Map projections may be classified by the shape of the geometric feature onto which the projection occurs (usually sphere, plane, cone or cylinder) and the location of the light source for the projection. Many other basic types of projections are used. The size of the projection image (or map) will depend on the **scale factor**, a number by which real distances are divided to get the (scaled) map distances.

The effect of the projection and scale factor on the paper map or the computer-produced map is only partly visible to the map viewer. The choice of scale has the most obvious effect by determining the physical size of the chosen area for the map. The scale also determines what each unit of measure (inch or cm) represents on the map image.

The projection has a more subtle effect on the image of the map. The projection type determines the location and amount of distortion on the map image. Distortion is the inevitable result of the projection process for mapping, because the curved surface of the Earth can not be perfectly represented on a flat surface at all points. Instead, one or more of the map scale - distances (equidistance), directions or shapes of regions (conformality), and areas of regions (equivalency) must be in error in varying (but predictable) ways throughout every map.

When two nodes are connected by an arc, the nodes may be used to identify the arc. At some point, the two nodes will be uniquely numbered. Then, the arc may be identified as "the arc between node #1 and node #2", assuming their numbers are 1 and 2. However, it is possible that the linear features of a map create a situation in which two arcs of the same type exist between two nodes. This situation is called **bridging**, and if left alone, it would mean that arcs could not be uniquely referenced.

Beyond the interactive placement and editing of symbology as mentioned above, another kind of symbolization occurs within the GIMMAP system. Creation of point, line and areal images for the production of maps requires enhancement of the feature image from primitive, minimal information into publication-quality symbols. This enhancement is called feature **annotation**, a process which is performed automatically in GIMMAP.

In annotation, single point locations are converted into selected, meaningful point symbols of user-defined size. Lines are stylized to resemble (e.g. hashed for railroads) the features they represent, or distinctive line types may be selected for each linear feature. Areal features are annotated by shade lines in solid or dashed, at various angles for different attribute values. Point symbols, line types and area shading may also be executed in various colors, and areas may be annotated by filling them with solid colors. Annotation is what is done on the output side of GIMMAP to produce the kinds of point, line, and area symbols which appear on maps.

The geometric features known as zones all have certain properties because of their size and shape on the map. These physical attributes can be measured and are of some use in the world of automated cartography. Each has a perimeter - the distance around the boundary and an area. More useful are the extremes - the minimum and maximum x and y coordinates for all points in the boundary. Such points can assist in knowing very quickly which zones are to be displayed and which are not. They also speed analysis to determine which zones may be near a given zone.

Each (zone) geometric figure also has a **center-of-mass** point, where the zone could be balanced from below. This point is not obvious to the eye when inspecting a map, but may easily be calculated by GIMMAP. These points are then used as the first

approximation for the **zone marks**. A zone mark is a location in the interior of its associated zone, used for selection of that zone by GIMMAP programs and for other purposes.

Another special kind of island zone may also occur. It is possible that a group of two or more adjacent (neighboring each other) zones may, as a group, be completely surrounded by another zone. In this case, there is no true "simple" island zone since none of the zones in this island complex is completely surrounded by the surrounding zone of the complex. Yet, to properly produce the boundary of the surrounding zone, the boundary of the island complex must be recognized and used.

So, the outer boundary of the complex, the boundary arcs (of the zones in the complex) which border on the surrounding zone, are recognized as a special **complex island boundary**. This occurs naturally as a result of the GIMMAP technique in recognizing zone boundaries automatically from the Node and Arc information. The complex islands are generated in the opposite (clockwise) manner of all other zones. Hence, their interiors are actually exterior to their boundaries compared to simple islands. Thus, they may also be called **ghost islands** or **complex ghost islands**.

Many maps are bounded by a rectangular edge at which all the map features end. Maps which are not bounded by a neat rectangle at the edge of feature information sometimes have a rectangle outside the features such that all pertinent map data lies in the interior of the rectangle. Any rectangle at which all of the map features terminate is called a **neat line boundary**. Any rectangle which surrounds all the map data, whether it is a neat line or not, is a potential **map frame**.

In order for such a rectangle to be useful as a map frame, the corners of the rectangle must be at known locations in the common reference system for the mapping system. For GIMMAP, this reference system is the **geographic reference system** of latitude and longitude. If all four corners of the rectangle are at known points in latitude and longitude, then the rectangle can be used as the map frame. The four corners are called the **control points** for the map and they are used to correct all point locations of features digitized from the map. The control points define the transformations between their known geographic system locations and the cartesian coordinate system of the digitizing table.

The map reference system is not visible. Locations are viewed as directions and scaled distances between points. No absolute coordinate system is visible unless a locator grid is included for narrowing the search for objects to a rectangle whose coordinates are given in an index. For computer mapping, however, all locations must be given coordinates in a controlled reference system. In GIMMAP, the original reference system for map data is the cartesian coordinate system, provided by the digitizing table. For maps with control points defining a map frame, geographic coordinates may also be obtained.

Some maps may contain a grid of latitude and longitude lines or tick marks. Such lines are provided for easy reference to the geographic coordinates of points on the map. For GIMMAP mapping, these grids, like map frames and neat lines, may be produced for plotting product maps without the need to digitize them from the original source map or to keep any extra information beyond that needed to completely describe the projection and frame.

- D R A F T -

I. INTRODUCTORY INFORMATION

I.3 THE BASIC SEQUENCE FOR GIMMAP

In the GIMMAP automated cartography system there is a basic set of processes, represented by a set of programs, which is used to take map data for a single map from an initial graphical state into a digital state for construction of a **cartographic database**, to production of separations usable for four-color lithography or color printing. This **basic sequence** of GIMMAP programs provides the capability of capturing map data and processing it to create a digital model of the original map, from which a suitable copy may then be produced.

Fortunately, more benefit is derived from this process than the single production of a copy of a map. For one thing, the map in the computer may be used again and again, and may be used to produce many different maps, taken as subsets of the original or altered in their form (e.g. red lines instead of black ones). Still more advantages may be found through reprojection and other techniques used to combine digital maps and to enhance map form. These operations are discussed in detail later.

The basic sequence provides the ability to reproduce a map, and provides the cartographic database as a very important by-product. The database is set of four or more **random** or **direct access** files which model the structure and form of the map as it was digitized. Each file has its own name (Arc, Node, Point,...) because of the information it contains.

Each unit of information in a file is directly accessible, meaning that any program can get any unit of information (such as a node) without searching through the file to find it. This is possible because a single number, called the **record number**, works like a street address to allow the program to go directly to the desired node.

The cartographic database is the center for the activities of GIMMAP programs. It is the digital map to which all programs go to obtain information about the map for display or production, and it is the place where programs go to alter the computer-based model of the map.

The first program in the basic sequence is **MAPDIG** (programs will always be named in capital letters), for map digitizing. The MAPDIG program controls the entry of digitized data from the digitizer, communicates with the operator, and creates the **DDF** or digitized data format file, the raw digitized data describing the source map.

MAPDIG provides set-up information to help assure that the map and the digitizing table are set up properly. MAPDIG also

checks control point digitizing against a set of measures to see if the operator is following guidelines and is consistent. The set of format and order rules defining the DDF file is **syntax**, which is only partially checked by MAPDIG.

The DDF file created by MAPGEN is a sequential (not direct-access) file, containing a **map header** (alphanumeric information about the map and operator), digitized control points, feature code information for all digitized features, and the digitized points for isolated points and linear arcs in proper sequence as (x,y) coordinate pairs.

The second program in the basic sequence, **SYNEDT**, for syntax editing, takes the original DDF file as its input, and examines the content for compliance with the rules of the syntax. If the rules are broken, SYNEDT reports the occurrence and type of the error, and proceeds to modify the file to comply with the rules of the syntax.

SYNEDT does not actually correct the file, in the sense that the error is understood and the file is changed to make it as it would have been if the error had not occurred. Rather, SYNEDT is aware of all possible violations of syntax rules, because it uses a transition matrix (a table showing states, input possibilities, and appropriate actions) which has been built to take explicit steps to modify the file for all possible situations, regardless of the cause of the error.

The SYNEDT program produces a **syntactically-clean** DDF file as its output. This output file is then ready to be used for the next process, database generation, without fear that it will not be read incorrectly by the subsequent program due to errors in the syntax of the file. The output of SYNEDT is a DDF file with no syntax errors. Such a file is called **syntactically clean**.

Two options exist after the SYNEDT program has been run. The least likely of these is to use the **DIGPLT**, for digitized file plotting, to immediately produce a plot file from the raw DDF data. The advantage of this option is to have an instant check to see if the digitizer and operator have functioned in a proper manner on a gross scale. This option is seldom used, but may be helpful when new operators begin work, or when digitizing equipment is not operating properly.

The usual option following SYNEDT is to begin the process known as **database construction**. This process is the building of the four or more direct-access files which are the cartographic database. The first step in this process is the construction of the **Arc** and **Point** files, which is performed by the program called **PARGEN**, for point and arc generation.

The PARGEN program constructs the Arc and Point files from the DDF input strictly according to the rules of the syntax. The feature codes indicate when locations in the DDF file represent

individual isolated points. For such points, a separate record is set aside in the Arc file, and no interior points are stored in the Point file. Linear feature codes signal linear arcs which may have interior points to be stored in the Point file, with the Arc file record set to know where to find them. If the syntax rules are not followed, PARGEN will be forced to halt operation, because a broken syntax chain in the DDF file may be interpreted in many ways, only one of which would be correct.

The PARGEN program calculates the length of each linear arc and the extremes (minimum/maximum) in X and Y for each arc, and these values are put in the Arc file for each arc. Only the arc endpoints are kept in the Arc file, and these are used later to find the unique node locations. **Pointer** values (actually record numbers for the Point file) are kept to direct programs to the list of interior points for linear arcs.

The PARGEN program also generates a third file, the QPLOT, for quick plotting, file. The QPLOT file is used to provide a fast display of the whole map by the interactive editing programs GRFEDT and ZONEDT, described later. The QPLOT file parallels the Point file, record by record, except that the coordinates for the points are converted into direct screen display coordinates. These can be read and displayed directly while coordinates in the Point file require lengthy conversion.

Although the PARGEN program is used to generate the first blocks of the cartographic database, it can also be used to add blocks to an existing database. The PARGEN program operates in either the initialize mode or in the re-entrant mode. When the database is empty or when all data in the database is to be deleted, the initialize mode is selected. When the database has some data in it which is not to be deleted, the re-entrant mode is selected.

The PARGEN program performs one additional, important task. When the database and operator specify that a map projection is being used for the database, the control points in the beginning of the DDF file are used to rectify all points in the DDF file to the known projection. The control points represent locations of known longitude and latitude for the corners of the map frame. These corners are also the corners of the projection area. When the digitized locations of the control points are compared to the locations given by the mathematical model, transformations are set to map from the digitizer space onto the projection space.

Following PARGEN, the next step in database construction is the NODGEN, for node generation, program. The main goal of the NODGEN program is to construct the **Node** file, which describes the network of inter-connected arcs and nodes for the map. The Node file contains the location of all unique nodes, the arcs which intersect at each node, and the collections of nodes which are directly connected to each node. This information is part of the topology, or logical structure, of the map.

The NODGEN program determines the unique locations of the individual nodes by examining all the arc endpoints of linear arcs and isolated points. These endpoints are tested to find those which represent different locations. Whenever two or more endpoints lie within an operator selected **threshold** distance from each other, the endpoints are assumed to be intended to be the same node. In that case, a node is created and its location, as well as that of all common arc endpoints, is set to the average of all points involved.

This process is used to locate the complete set of unique node locations according to the way that the nodes were digitized by the operator (see above). Only digitized nodes are found. No splitting of arcs at non-digitized points occurs in GIMMAP. All isolated points translate into a node, some occurring at the same locations as other arc endpoints.

For each unique node found, NODGEN averages the locations of all the arc endpoints to find the (x,y) location for the node, finds the **degree** of the node as the total number of arcs incident at the node, and constructs a complete list of **node-arc pairs** for each node. A node-arc pair is a node which is directly connected by the arc of the pair. Thus the node-arc pairs are a complete list of nodes connected to a central node, and the arcs by which they are connected.

The arcs in node-arc pairs are signed positive or negative to indicate their direction of digitization with respect to the two nodes. If an arc begins (according to digitization) at the "central" node, then it will be positive in the node-arc pair. However, if it begins at the node in the node-arc pair, then it will be negative in the node-arc pair for the central node.

NODGEN also examines linear arcs to calculate statistics for length of each arc and the minimum/maximum X and Y coordinates of each arc, to aid in rapid display of data. These values are then stored in the Arc file. This process could not have been done in the PARGEN program, because the arc endpoints are moved when the locations are averaged by NODGEN.

Following the NODGEN program, the cartographic database is constructed in its most basic form (zones and symbology may be added later). Four direct access files, the Arc, Node, Point and Qplot, now provide a digital model of the map. At this point, numerous errors may be found in the database resulting from many different sources. These errors may be detected and some may be corrected by the **GRFCHK**, for graphical checking, program.

The errors in the database may arise from unusual methods in digitizing, from uncorrected errors in construction software (the MAPDIG, SYNETD, PARGEN, and NODGEN programs), or from naturally occurring situations in the original map data. Details of these errors and their treatment will be given later, but there are two basic kinds of errors found at this stage. The first is one in

which the error may only be detected and may be a serious error or only questionable. For these errors, GRFCHK can do nothing to attempt correction. For the second kind of error, GRFCHK will report the error and then perform a preset correction which will alter the database, such as deleting what is believed to be a duplicate arc.

GRFCHK is usually run whenever the NODGEN program is run, because that is when construction is complete on the basic files of the cartographic database, and these errors are likely to be found. However, GRFCHK checks for many kinds of inconsistencies and errors in the structure of the digital map, and may be used at any time program or operator error may be suspected.

The next operation in the basic sequence is to correct the graphical errors which have been introduced into the map data by the operator in the digitizing stage. This operation is provided by the **GRFEDT**, for graphical editing, program. GRFEDT, one of the largest modules in the GIMMAP system, provides many different edit functions in an interactive setting with continually updated graphic display of the edited map.

The GRFEDT program operates on a (monochrome - single color) graphics display terminal. The operator may view all or a part of the map to be edited on the graphics screen, on which one area is reserved for conversation between the operator and the GRFEDT program. The operator may display the command **menu** to see a list of commands to use, with information about necessary values to be included with each command.

Using commands from the menu, the operator may perform two basic kinds of operations. Modifying commands are used to change the contents of the cartographic database, to add, delete or to change map data in the database. Utility commands are used to change the environment of the editing session itself. Using the utility commands, the operator can better locate and operate on the graphical errors which are to be fixed.

The GRFEDT commands will be described in detail later, but a few general functions will be listed here to suggest the utility of the GRFEDT program. With the utility commands, the area to be viewed may be changed in scope (size) or in location. The set of features may be defined precisely by the selection of up to five ranges of feature codes. Features whose feature codes lie in at least one range are displayed, others are not. Numbers generated by the system (actually record numbers) may be displayed at each node for easy identification of nodes or arcs. A verification mode may be turned on (or off) to double-check modification of features before updating the database.

The modifying commands include the ability to move isolated points, nodes, and interior points to any desired location on the screen. Arcs may be deleted or may have their feature codes changed. Unneeded nodes (those occurring in the middle of an arc

but not at an intersection or termination) may be removed, a new arc may be created to bridge the gap between two existing nodes. Isolated point arcs (and possibly nodes) may be deleted as may one or more interior points of an arc. New nodes may be created where the digitizer operator forgot to create them, and nodes may be joined together by command. Several commands may be performed en masse on groups of arcs selected by feature codes or on nodes that are unneeded.

Following GRFEDT, several paths of program operation may be taken. Usually, the first path after GRFEDT is to create an **edit plot** to be used to check the digital image against the original map document. The edit plot is produced by the **PLTGEN** (plot generation) program to help the operator determine if graphical and other editing is complete.

The edit plot produced by the **PLTGEN** program reflects the current state of the digital database. It contains the complete set of all features in the database, with some stylized lines to identify certain features or colored lines for others. The plot shows where some map features have been overlooked, others may have been mis-digitized. Node numbers or degrees may be posted at the nodes to assist in identifying corrections in GRFEDT or to locate missing intersections.

The edit plot produced by **PLTGEN** may be created at any time after the database is created, and may thus be used to verify the state of the database at any point in **GIMMAP** processing. There is one fact about the edit plot which is important to correctness in editing. The edit plot is one in which the map features have all been rectified according to the control points and the known map projection. The paper map to which the edit map is compared has not been rectified, and retains distortions which occurred in the printing process and in subsequent storage. Thus the two maps may have differing scales or non-linear differences.

Production of the edit plot and comparison with the original map may result in additional digitizing and database construction or graphical editing. These operations in turn may again result in repeated cycles of editing and digitizing. When the cycle is completed and all (known) errors have been corrected, the database is considered to be **graphically clean**.

When graphical editing has been completed, there are two optional functions which may be performed in the basic sequence. These are the recognition and construction of zone boundaries for maps which have areas to be distinguished, and the creation and editing of map symbology directly on the display of the digital map. Both of these functions are considered to be optional since some maps do not involve areas or zones, while other maps do not require the addition of symbology. Therefore, neither, one, or both of these functions may be performed. And since neither is dependent on the other, they may be done one after the other in sequence or at the same time in parallel.

The creation of zones is considered to be a part of database construction, and is referred to as **extended construction**. The process involves examination of the qualified (by feature code range selections made by the operator) node-arc pairs in the Node file to construct pathways beginning at **start nodes**, following arcs counter-clockwise around a zone boundary, and returning to the start node. The program which performs this task is called **ZONGEN**, for zone generation.

The ZONGEN program creates the Zone and Border files as a result of the recognition and creation of zones. This process systematically uses all possible nodes (as qualified) as start nodes unless all (qualified) node-arc pairs for the node have been used in construction of previous zones. At each node in the boundary of a zone, a decision must be made as to which arc to take next. To produce the counter-clockwise boundary, the left-most, unused arc is selected. This **left-hand turn** procedure guarantees a counter-clockwise boundary with the interior on the left of each arc, and return to the start node, provided the node structure is correct and left-turn decisions are correct.

Of course, these conditions may not be correct in the course of running the ZONGEN program. One or more nodes may not have been digitized. Two nodes intended to be the same may have been digitized too far apart for automatic joining, and may not have been joined in graphical editing. An arc may have been missed in digitizing. The angle an arc makes at the node, determined by the first or last interior point, may be incorrect because of accidental "overshoot", causing the arc to appear to enter the node from the opposite side that it does. This error will result in a **wrong-turn** or a **dead-end** error at some point in construction of the Zone and Border files, thus terminating ZONGEN.

When extended construction is successful, both the Zone and the Border files are constructed. The complete content of these files is complex and will be described in detail later in I.5 Digital Representation of GIMMAP Data. However, in general, the Zone records (one per zone) each contain a number from 1 to 32767 called the **zone color**, the primary attribute of the zone. This zone color distinguishes the type of zone which is represented, as denoted by the color (or shading pattern...) assigned the zone on the original source map. This value is also used to determine the color or shading used in production of the map.

Each zone record also contains information for a reference mark and statistics about the zone. The reference or **zone mark** is specified by an (x,y) location and the **mark size**, a number to indicate the size of symbols placed at the zone mark location in production or display. The zone mark is used to provide a mark for the operator to point to to identify the zone, and a place to post numbers identifying the zone or its color. Initially, the zone mark is located at the center-of-mass of the zone by ZONGEN, but may be moved later.

The zone mark may also be used on a **peel coat** (a negative image in mylar of part of the map) in map production to show the interior of the zone which is to be peeled away for plate-making. For this reason, it is necessary that the mark size and location be adjusted so that the mark lies completely inside the zone. Similarly, production requires that the mark is not placed inside any island which may be in the interior of the zone.

The statistical information in the zone record includes the perimeter and area of the zone in digitizer and square digitizer units, easily convertible to appropriate units on the surface of the earth. Also, the extremes of the bounding rectangle for the zone (minimum/maximum X and Y coordinates) are kept in every zone record. These values assist in finding appropriate island zones, and improve the speed of all programs which display zones. These values quickly show when zones are complete inside or outside of a chosen display. Thus, only those which are neither must be carefully examined and **clipped** to cut off undisplayed arc parts while showing those inside the display.

Still other information in the zone records links together zones with their islands and their surrounding zones, allowing for rapid retrieval of the complete zone boundaries which must include the boundaries of all islands within. Other pointers in the zone record link the zones to their own chains of boundary records in the **Border** file and to **zone labels** (symbolology linked to the zones) in the **Label** file.

Information in the **Border** records reflects the fact that the arcs used for zone generation are all used as border arcs for two zones. Each is used positively, (in the direction in which it was digitized), for the zone to its left as digitized. Each was used negatively for the zone on the other side. The **Border** file records each represent a single arc and information about the two zones on either side of that arc. This information includes the zone numbers and colors and the pointers to the next records in the **Border** file for the boundaries of the two zones. Finally, the information needed to quickly draw the arc is included.

When zone generation has been performed and the **Zone** and **Border** files have been created, the next step (which can be performed in parallel to the symbol operations below), is to prepare the zones for production. This function is performed interactively by means of the **ZONEDT**, for zone editing, program. In zone editing, there are three goals to be met to prepare the zones for production. These are marking, coloring, and linking the islands.

Zone marking is aided greatly by the **ZONGEN** program in the original creation of the zones. At that time, **ZONGEN** calculates the center-of-mass for every zone and stores that location as the initial zone mark in the zone records. The size of the marks for the initial setting is arbitrarily selected by the operator and is uniformly set for all marks. The choice of size is usually

dependent on some combination of making all marks large enough to easily see on a plot while requiring marks to be in the interior of their zones. The ZONEDT program allows the operator to view marks in a controlled sequence and to modify the location or size or both in each pass.

The second goal of the ZONEDT program is to set the colors of the zones to appropriate values reflecting the major attribute or production color (or shade pattern). The ZONGEN program helps in this goal by initializing zone colors as instructed by the operator. Two options are provided. In one, the zone colors are set to be equal to the zone numbers (i.e. zone 4 has color 4...). This method is most useful when all areas of the map are to have unique values or when nearly all values are unique. The second option is to set all zone colors to a single value. This method is helpful when zones will all have one of several values. Then, all zones will be set the most common value and the others must be reset using the ZONEDT program.

The coloring function in ZONEDT is used to view one or more zones, find out the zone color for each, and then change the zone color to any desired value. Several different methods are used to accomplish this goal. As an example, one method allows the operator to select a color with which to "paint". The operator then selects zones by pointing to them on the screen, and each of these zones has its zone color set to the selected paint color. Another coloring method allows the operator to select an old and a new color. Then all zones having the old color are changed to have the new color.

The third goal of the ZONEDT program is **island linkage**, in which the operator must explicitly inform the ZONEDT program (which then incorporates the information in the database), which zones lie completely within which other zones as islands. Again, information generated by the ZONGEN program is used to assist the operator in this task. Island linkage is performed in ZONEDT by the operator selecting zones (by pointing or by number) which have islands which were detected by the operator through visual inspection. ZONEDT then produces a display of the surrounding zone with emphasis on islands (which ZONEDT detects) which may be inside the surrounding zone. The operator then points to those which are inside, and ZONEDT stores the information. Mistakes in this process may be removed just as easily.

Whether or not extended (zone) generation is performed, the process of **symbolization** may be performed whenever a map requires the addition of titles, scale bars, feature names, text or legend information. The interactive creation and editing of symbols on the display of the digital map is supported by the CYMBAL, for symbol generation, program.

The CYMBAL program provides a display of the map data which is similar to that of the graphical editing program, GRFEDT. The operator views all or part of the map area, with optional feature

masking by feature code, while selecting and placing symbology at chosen locations. All aspects of the symbols can be modified by commands in the menu of the CYMBAL program.

Generation of symbology using the CYMBAL program causes the creation of records in the **Symbol** file, (one per symbol), which contain information about selections made for the symbol. All symbols require a **reference point** location, specified by the (x,y) coordinates of the point. Each symbol requires selection of the **font** or **alphabet** set from which the characters are chosen to display the symbol. A **character height** must be selected as well as an angle of **orientation** for the symbol. The symbol must begin with its left or right end against the reference point or must be centered in its **justification**. And finally, the **symbol content** must be specified. All these factors may be changed by other editing commands from the CYMBAL menu.

Like arcs, symbols each have a feature code assigned to them to allow differentiation of symbols to parallel that of arcs. If the map is to contain only hydrology, then only symbols for the hydrologic features should be plotted. The symbol feature code is assigned at creation time, is stored in the symbol record, and may be changed with a CYMBAL command.

The **content** of the symbol (letters typed at the keyboard to produce the symbol) are not stored in the symbol file. Only the number of these characters is stored there. The actual content of the symbol is stored along with the sequence of ordered (x,y) locations defining the pen strokes needed to draw the symbol, in the **Calligraphy** file. Similar to the Point file, the Calligraphy file links one or more records by pointers to form a linked list containing all the pen strokes for the symbol. Each record in the Symbol file contains a pointer to its own list.

The records in the Calligraphy file contain the points that are needed to draw the symbols. The first record for a symbol contains the string of characters defining the content of the symbol. Beyond that (if there is room), the first record, like all the others, contains points defining the plotter pen strokes for production of the symbol. Each point has the (x,y) location, with the pen status (pen up or pen down) indicated by the sign of the coordinates (positive for down, negative for up). Since the end of the pen strokes may occur in the middle of a record, the special value of (-999.,-999.) is used to terminate the symbol.

The final operation in the basic sequence for GIMMAP is the **selective retrieval** of map data from the cartographic database constructed and edited in the above processes, and enhancement of the map data for creation of **plot files** to drive digital plotters into producing a map equivalent to the source original, as well as other map products. This process involves five or more GIMMAP programs in the basic sequence, two of which have been discussed above.

The basic map data, isolated points and linear arcs, in the cartographic database is selected and retrieved for plot file production by the **MAPGEN**, for map generation, program. In this program, the operator may select up to fifteen different ranges of feature codes to be selected, for each of as many as fifteen different plot files. The utility of separating features into separate files provides for different color assignments as well as creating maps containing many different combinations of map features without needing to re-run the program.

The **MAPGEN** program provides a number of point symbols which can be automatically generated when assigned to isolated points. Similarly, many different line annotations (dashed, hashed, double...) may be automatically generated from the default, solid line which is normally produced. Many other important functions are performed by the **MAPGEN** program which are outside the scope of the basic sequence. These functions will be described in some detail in the section on the Basic Sequence.

The **MAPGEN** program takes information from the Arc and Point files according to the selections made by the operator. Isolated point symbols are created at point locations by generating pen strokes relative to the selected location. Linear arcs trace the path of the endpoints in the Arc file and the interior points in the Point file. When these lines are annotated, extra pen strokes are generated for the plot file to provide the extra line for double lines, the hashes for hashed (railroad) lines and pen up commands for dashed lines. Many other considerations must be made to produce proper annotated lines for plot files.

The selective retrieval of zones and creation of plot files for zone information is performed by the **ZONEDT** program itself. The **PLOT** function provides for the cyclical creation of zone plot files containing the boundaries (in proper sequence) of zones of selected sets of colors. Some special options allow for the suppression of boundaries when the zone colors on both sides of the arc are the same, and when zone areas do not exceed a minimal threshold or do exceed a maximal one. Careful selection of color sets will provide a minimal collection of plots required to make printing plates for four-color publication.

Selective retrieval of symbols based on feature codes for the production of separate plot files is performed by use of the **PLOT** command in the **CYMBAL** program. As with zones in **ZONEDT**, the operator may select various ranges of feature codes for each of the plot files generated. All symbols in the specified ranges are extracted and converted to plot form, a minor transformation from the data in the Calligraphy file.

Use of the plot files in the basic sequence to create the final map products may follow many variations of two basic paths. The two paths are divided by the two different kinds of plotting devices which are supported in **GIMMAP** mapping. One is the vector plotter (Xynetics in our shop), the other is the raster plotter

(Calcomp for us). The vector plotter is used to produce the best line quality and highest accuracy on a variety of plot media with some limitations on line colors and widths, but with no ability to color areas in solid colors. The raster plotter has a reduced line quality and accuracy, but can produce solid areas of colors specified by the operator.

Consequently, the vector plotter is used for production of many high-quality line plots which are usable as finished map products directly off the vector plotter. More importantly, the vector plotter is used for direct production of color-separated materials (scribe coats and blacklines) used to create printing plates in the four-color lithography (printing process).

In this process, all boundaries of areas to be colored are placed in separate files depending on the percentages of primary colors (cyan, magenta, yellow) in the zone color. The various percentages of each primary are created through the application of **screens** to block given percentages and a single **printing plate** made of metal is created for each primary. With the addition of a black line plate, these plates comprise the four primary colors from which all others may be made. One pass of the map through the press for each plate and the maps are complete.

The raster plotter accepts vector (line-based) map data and converts it locally to raster format for plotting. This plotter specializes in coloring areas such as zones, but also plots lines and point symbols. Line plotting also means that the symbology of GIMMAP can be plotted as well. Production of such plots in the GIMMAP system is almost as easy as with the vector plotter. Line plots for the raster plotter are created in the same way as they would be for the vector plotter (by MAPGEN or CYMBAL) and then a program called **X2C**, for Xynetics to Calcomp, converts the vector plot into a raster plot. In this conversion process, line colors and thicknesses may be assigned to all groups of lines by feature codes.

When zones are to be produced for plotting in color on the raster plotter, a separate program must be used. The **CALPLOT**, for Calcomp plotting, program provides for the extraction of zone boundaries by zone colors to create plot files for color plotting on the raster plotter. The operator selects zones by ranges of zone colors, for each of numerous selected plot files. Selection of zone colors and mapping of the zone color to a plot color is assisted by the optional color dictionary, which may be reused to minimize typing for the CALPLOT program.

The raster plotter is used for production of intermediate and final working maps and research (one-of-a-kind) maps which emphasize colored, areal features or which have need for color to distinguish lines and/or areas. The raster plotter is also used for final production of moderately-high publication maps which are to be published in low to moderate quantity. To some extent, the raster plotter will produce near-production-quality for maps

which, in their digital form, might be regularly updated with new information to provide **on-demand** mapping.

I. INTRODUCTORY INFORMATION

I.4 HARDWARE USED BY GIMMAP

General

To perform the functions described for the basic sequence in GIMMAP, assuming that the color options are to be included, a set of hardware is needed which will perform virtually all the tasks that the system supports. That is, this set of hardware for the basic sequence will also provide all the hardware needed for the "extended" functions and the "additional processes and sequences" discussed later. Of course, if color processing is not included in the basic sequence, some hardware (as noted) is not required.

The central component of the needed hardware is a computer. The computer runs the software (FORTRAN77 programs for GIMMAP) that communicates with the operator, displays the maps, receives and interprets commands, constructs the database in disk files, changes the digital representations of the map features, defines colors, interprets selections and retrieves features, and every other function provided by the GIMMAP system.

The computer must perform all these tasks for one or more operators at a time in an environment in which automated mapping is only one of many important applications, and must provide such services with only minimal delays to the operators. That is, the verification process in the graphical editing program GRFEDT must be fast enough to provide the impression of a continuous form. When a fresh screen is drawn, the operator should not be tempted to start reading a book. To accomplish this, the computer must have a minimum level of speed and internal resources (primary and secondary memory...) comparable to today's super-minicomputer.

The GIMMAP system has been developed in an environment in which there has (nearly) always been a group of users competing for the computing resources. Competition has guided development of GIMMAP in more realistic ways than its absence would have done. Emphasis was placed on efficient, fast algorithms instead of on assuming that boundless resources would be available to solve the problems. At times, emphasis was placed on finding any way to do the job in a difficult environment, not on wasting resources. Having developed in this environment, GIMMAP could operate better in an environment in which the computing facility provides more resources for a single, or more for increased numbers of simultaneous mapping users.

The latter stages of development for GIMMAP occurred in a system with virtual memory. Thus, after the hard problems were solved as efficiently as possible, the new computer afforded us the ability to expand the size of allowable feature information

which can be processed on a single map in the basic sequence of the GIMMAP system. This "maximum" map contains enough features that its data segmentation (see below) might be considered for it.

The Computer

The computer used in a GIMMAP facility must not only run the GIMMAP programs described above, it must also support the other hardware discussed below. A super minicomputer does that and can support many other computing functions for many other users at the same time. That is why the Kansas Geological Survey has the Data General Eclipse MV/20000 Model 1 computer which performs all those tasks for the Survey. Today's 80386-based microcomputers might perform the tasks in GIMMAP, provided the peripheral devices were available and no other user demands had to be met.

While the MV/20000 is sufficient for running GIMMAP with two to eight operators at a time in the current mix of computing at the Kansas Geological Survey, it is not the only minicomputer which might do the job. To evaluate a computer for its ability to support such an environment, there are several aspects which must be considered.

In its component parts, the computer must be up to the job in at least three basic areas. The speed of the processor and of memories and communications must be up to a minimum level which is difficult to quantify. Certainly, the system must support up to forty or more simultaneous users in real time that appears to the users as acceptable interaction. (Interactive programming is also a requirement which should no longer need mention.)

Processor

The processor of the MV/20000 Model 1 has been rated at 6.4 MIPS or million instructions per second (a comparable VAX machine is the 8550 at approximately 6 MIPS). This number may be one useful way to compare computers for the purpose of determining if GIMMAP may be used on a system. Of course, other factors play a large role in that judgment even if a MIPS rating is available for a particular computer. The single most important factor may be the amount of computing power which is directed elsewhere. If too much processor time is given to others in the environment, the mapping application can not be supported. Or, if a computing system has memory or communications problems which reduce the real availability of the processor, interactive processes may be unacceptably slow.

Memory

The second area of the computer which must meet some minimum level of performance is the main memory of the computer. There is some amount of memory which is required to run the FORTRAN77

programs for the GIMMAP system. The exact amount of memory is not easily obtained from the MV/20000 system. In fact, the exact amount would not be meaningful in another system except in a very general sense. The local system has a "virtual memory" based on real memory of 8 megabytes.

The largest memory requirement in the GIMMAP system is for a program ZSHADE which creates shade lines for areas (zones). The current run time requirement for this program is more than one-and-a-half megabytes, more than a million and a half bytes. This figure is misleading in that it has become that large as a result of local needs and processing conditions, and because research has not yet been done to determine how large the arrays must be to shade a given map. At this size, ZSHADE can produce nearly solid areas for a map with many zones. With data segmentation, there should be no need for this program to require more than about 600 kilobytes of memory.

The next two largest programs in size, CYMBAL and GRFEDT, both require about one megabyte of memory. In both cases, the limits on features (ten thousand arcs and five thousand symbols) are set arbitrarily high to allow production on specific maps. If these limits were set lower and segmentation procedures were followed, the programs could operate with much smaller memory requirements (such as seven hundred kilobytes).

Secondary Storage

The third area of the computer in which minimum requirements must be met is the area of secondary storage. The creation and editing of cartographic databases, and the extraction of map data from these databases to create plot files for map production lead to a need for large amounts of high-speed, direct-access storage. This kind of storage must provide access from which map images can be generated fast enough to support interactive editing and retrieval for display operations.

The current allocation for all of the GIMMAP-related files on disk is approximately 250 megabytes. Of that, two thirds is used for storage of nearly 2,500 cartographic databases, most of which (2,379) are for USGS 7.5 minute quadrangle maps in which only a small subset of the map data on the original map has been captured. Of the remainder, less than half is used for storing all the GIMMAP software and for development and testing of new software or correction of existing software.

Requirements for secondary storage of this (disk) type can be reduced by following a careful system of archival of database files onto magnetic tapes. Base files which are no longer needed on a demand basis can be copied onto magnetic tape and then can be erased from the disk, freeing space for other, new bases. If the need for maps supports archival, very little disk space is actually required. However, local processing dictates the need for rapid turnaround of maps for all areas of the state. Hence,

statewide coverage in cartographic databases is necessary, and our need for disk space is as stated above.

Aside from the three areas of performance discussed above, the computer must also include some basic hardware and software in order to use the automated cartography software of the GIMMAP system on it. The computing system must include the adequate computer terminals and a high-speed line printer which might be taken for granted as part of all computing systems. The computer terminals provide for interactive communications to submit programs, check their progress and even examine their output. The printer must provide the means for the output of the computer to be realized in a hardcopy, paper form, where GIMMAP reports are sent to be examined by the operators.

Two other elements must be present in the computer to allow GIMMAP to function completely. One element, mentioned above, is the ability to produce, to store, and to reconstruct archived map data from the cartographic databases. Long-term archival storage can be supported best by use of magnetic tapes. Two tape drives, preferably with high-density capability, are suggested.

Last but not least on the list of necessities for the GIMMAP computing facility is a FORTRAN77 compiler. The GIMMAP system is written entirely in this language which seems to be very popular for scientific programming. While there are many issues involved in transporting software, the existence of a FORTRAN77 language standard should not be one for transporting GIMMAP. A minimal, hopefully portable, subset of the standard was adhered to where possible.

Digitizing Table

The capture of map data is made possible by the **digitizing table**, a large flat table which converts locations on its surface into cartesian (x,y) coordinates in computer or **digital** form. A digitizing table consists of the table, embedded with a grid of electrical wires at very small separation (in the kind of table used locally and at most mapping facilities - but, there are other kinds). Attached to the table is a cord with a small box on the end, and on the box are (usually) four or more buttons of different color or number.

In a round glass in the box, a set of crosshairs (**crosshair cursor**) indicates a single location when placed on a map on the table. The operator digitizes the map in general by locating the cursor on a desired point and pushing the correct button to tell the computer which point to use and what to do with it. There is a link, similar to that for a terminal, which is built into the digitizing table or is controlled by an interface (included with the table), which communicates with the computer.

The digitizing tables used locally (Altek) for GIMMAP are large, with the ability to digitize maps up to 42 by 60 inches. They are backlighted to improve readability while digitizing and the plane of the table may be rotated through 180 degrees while being moved vertically through a range of several feet. All of these features are helpful to digitizing operators, but are not necessary. The large digitizing area is also useful and closer to necessary than not. Large maps can be digitized in parts on small tables, but control points must exist in all parts.

Vector Graphics Terminals

Monochrome (single-color), vector graphics display devices are used for interactive graphical editing (GRFEDT), zone editing (ZONEDT), and symbology placement and editing (CYMBAL) in GIMMAP. These programs rely on a collection of graphics routines called the **GRFMAN**, for graphics management, library. The GRFMAN library is written in the FORTRAN77 language, and provides basic graphics functions for all vector-based displays in GIMMAP.

The monochrome, graphics terminals used by GIMMAP here have been the terminal on which a de facto standard had been built in recent years. The Tektronix 4014-1 vector terminals have been so successful that nearly all other manufacturers have found it in their interest to provide compatibility with the functionality of this terminal. It is for this terminal that the GRFMAN library was designed. For this reason, it is believed that conversion of vector graphics software to for GIMMAP would not be difficult. At least, it may be easier for systems compatible with the Tektronix 4014 de facto standard.

The basic function of the vector terminal in GIMMAP is to move an imaginary "cursor" (locator or pointer) to a specified point on the screen, then move it to a second point creating a visible line on the screen. When no line is drawn, it is called a **move**. When the line is drawn, it is called a **draw**. These two functions provide most of the operations performed on the vector display screens. With them, point, line and area features are constructed and annotated. Graphic editing functions use moves and draws for all but a few functions.

At times, it is advantageous to draw in special ways that are provided by the hardware of the vector display terminal to increase the speed of the operations from levels which would be unacceptable if done by the software. Included in these is the ability to draw wide or bold lines for emphasis and various line types such as dashed and dotted to distinguish some lines (by the feature codes for GIMMAP) from others.

The (Tektronix) vector graphics display terminal is based on a technology which allows very accurate placement of points and lines on a screen which can be drawn on anywhere, but can only be erased completely. This technology is called the **storage** tube technology. In this environment, features drawn on the screen

remain on the screen until an **erase** operation is performed. When this happens, all features on the screen are erased. Then, all desired features must be redrawn on a new, clean screen. Updates of map features can not be shown immediately. The screen may be **refreshed** only by explicit, feature by feature redrawing done by the software of GIMMAP. This is often time-consuming.

The (Tektronix) vector display terminal does provide a means to draw temporary lines on the screen which disappear after the drawing is completed. As easily as setting lines to be wide or bold or dashed or dotted or solid, the lines may be set to be temporary or **write-thru** (in Tektronix terminology). When such a line type is chosen (in GIMMAP, to identify a selected feature), the line is redrawn many times to allow the temporary flashing, dashed lines to be located by the operator.

Hardcopy Device

When a vector display terminal is used to perform the GIMMAP mapping functions it is advisable to have a device associated with the terminal which provides the ability to copy the entire contents of the screen of the terminal at the touch of a button. This monochrome device is called a **hardcopy** device, providing a "hard" or physical (versus the "soft" electronic image) copy of the image on the screen.

The hardcopy device is used to document the normal functions of graphical editing and to provide support documentation for the unusual editing situation which can be resolved only by others. The hardcopy device also provides copies of information taken on the spot when software or system conditions or errors cause any unusual termination of the program. Without such information, program errors may be more difficult to resolve.

Color Terminals

Color display terminals are used in several programs in the GIMMAP system. The basic function of the color terminal is to accept a string of points in (x,y) coordinates as the boundary of a zone, and to display that area with the color specified by the program (via the operator). For color terminals to do this, they must have a technology quite dissimilar to the technology of the vector display terminals.

Color terminals base their operations on the principles of the color television. Electrons are emitted from guns in the tube and under control of powerful magnetic fields are guided to selected dots on the terminal screen in a continuous, rapid zig-zag pattern covering the screen. The electrons strike only desired spots to produce colors based on the RGB (red green and blue additive mixture) system.

The color terminals operate in **raster** graphics mode in which the image is viewed (always) in the whole as a kind of matrix or

grid covering the screen, in which each element or node is always assigned a value which determines the color to be displayed at that element. The image elements are called "picture elements" or **pixels**, and can each be addressed by row and column, and their colors can be changed by program. Raster processing, unlike the vector-storage technology, allows for each of the elements to be updated or erased independent from others.

The color terminal supports this advantage by providing a bank of memory which parallels the color image on the screen. The **video memory** provides enough storage to allow the hardware of the terminal to continuously update the entire screen (updating every pixel 60 times each second) from the contents of the video memory, which may be structured in various ways. The options allow for varied numbers of display colors and different techniques of determining them. The set of colors available to the terminal is called the **palette**, but only a subset of these (many) colors can usually be displayed at a single time.

The resolution of the color screen has been notoriously low in comparison to the vector terminals, but this has been changing in recent years. A pixel resolution of 640 by 480 or better will minimize resolution problems like the "jaggies" (the staircase or other rough line appearance of some lines) on the color terminal. High pixel resolution in conjunction with very large palettes and large simultaneous color selections help to make color terminals an acceptable alternative or even a replacement to vector storage terminals. Consider that color operations may require much less resolution and color becomes the first choice.

Some limitations on color terminals may prohibit or diminish their usefulness in the setting of the GIMMAP system. The limits on functionality are generally the result of the limits placed on the amount of local memory the color terminal has or on the architecture used for the color memory. Such limits may affect color options and reduce the number of usable colors to levels which are unacceptable. The memory limitation may restrict the ability to segment (keep entire images in local memory for local manipulation...zoom etc.). Insufficiency of the local memory may even reduce the number of vertices allowed to represent a single polygon to an unacceptable level in which some maps with complex area borders could not be displayed at all.

Color Hardcopy Device

For much the same reasons as the monochrome, vector display terminal, operation of the color terminal can be supplemented by addition of a hardcopy unit. For a color terminal, the hardcopy unit may produce full color, a subset palette of colors with all displayed colors mapped into it, or it may produce only levels of gray. Full color is usually more expensive, but is the option of choice. Hardcopies may assist all editing operations as well as verification of color selections. Hardcopies may also be used to record any unusual structures (potential errors) found in the map

database and in details surrounding the occurrence of errors of different kinds. The hardcopy may also serve as a preliminary output device and may provide map products of sufficient quality to be included as final copy in single or limited copy, in-house reports. Color hardcopy devices are expensive (relative to the monochrome, vector devices), but offer a dimension to automated cartography which may prove vital to some applications.

Vector Plotters

The principal output device of the vector-based cartography system has traditionally been the vector plotter. Prior to the recent advent of the high-resolution (400 dots per inch) color, raster plotters, the vector plotter provided the primary source of conversion from digital data to map products. Vector plotters all perform the task of accepting a stream of locations in (x,y) coordinates with a command value which says "pen up", "pen down" or some special function. For the two basic commands, the vector plotter responds by moving the pen to the location specified with the pen up (not drawing) or down (drawing).

Vector plotters come in many different forms, depending in part on the quality and cost of the plotter. Most use pens which move up and down, writing much as we would with a similar if not identical instrument. Some move the pen to each new location, others move the paper to the pen which is then moved only along one axis of the plotter. Some use ball point pens, some use liquid ink pens for a more professional look, others use both. Some plotters allow the direct cutting of scribe material (color separations are made with scribe material, providing access to the four-color, traditional printing process for the automated cartography system). Still others use optical techniques to directly expose film negatives under control of the computer, also providing direct access to the printing process.

The vector plotter in GIMMAP is used for a wide range of map production functions. The primary, edit-completion plotting to display the complete map in multiple color at digitizing scale for comparison to the original input document is performed with the vector plotter and the PLTGEN program in the basic sequence. Finished plots of a single map area or of multiple areas which have been quilted together by the MAPGEN reprojection facility may be plotted on the vector plotter. Done on paper for a report in-house or on a limited basis, such maps may act as the final form for the map products for a project. Other maps may be done on mylar to act as the final report or as a controlled document for addition and digitization of other map data. Such maps must be made on the vector plotter.

The top end of map products are those which are plotted on the vector plotter as scribes or as blacklines (for creation of "contact" negatives) to be used in creation of materials for the process of four-color lithography (printing) for the traditional publication process. Linework is **scribed** (cut through the opaque

layer of a two-layer mylar material called **scribecoat**, creating a negative image of the lines) using a diamond- or sapphire-tipped stylus, with variable width to produce the appropriate line width for each map feature.

Areas to be colored are similarly scribed with complete boundaries, after being separated according to desired colors. The area scribes are used to create a positive image of the areas by exposing **peelcoat** material to burn boundary lines where they occurred on the scribecoat. The opaque layer of the peelcoat may then be "peeled" away to expose the whole area to be colored, and may be **screened** to diminish the amount of color to be allowed through by controlled amounts.

The scribecoats and the peelcoats, with desired screening, are then combined to create the metal printing plates which are used for the passes through the printing press. If colors are few, each pass will use one plate and a single pre-mixed color. Whether or not many colors are required, all map colors may be reduced to combinations of percentages of primary colors (cyan, magenta, yellow, and black). Then, with all the percentages of each of the primaries combined on separate plates, at most four passes (one for each primary) must be made.

The completed map is of the highest quality, rivaling any map made by traditional manual methods, but with all the added advantages of digital mapping (see above, Advantages of Mapping By Computer, topic I.1.1.5). This map will greatly exceed the quality and aesthetics of any map made by today's color plotters, though costs may be comparable. This difference is not likely to change in the near future, but other factors (see below) have made the addition of the color plotter highly desirable.

Color Plotter

The color plotter is a recent addition to the automated cartography shop. It brings the ability to produce full-size plots of maps in color, as they are displayed on the color screen of the raster graphics display terminal, and as they have been printed in the past only by the traditional printing process from color-separation materials produced by the vector plotters above.

Now, we control the finished map in its colors and its forms from the original capture of the map data from source documents to the mixture of primary colors for all elements of the final color map. With this control we can define every aspect of the map product, and we have the responsibility to define all aspects of the map and to redefine them until we have the best possible product.

The color plotter operates in the world of raster graphics, but arrives knowing that it will (most likely) be placed in an environment full of vector graphics users. The color plotter is ready. It converts vector-based files to its own inner form and

produces perfect color, raster plots. In fact, it is designed as much for the vector form as for its raster form. Even raster-based operations result in counts of vector-like objects.

The color plotter may also come in many forms, but the most common is one in which paper is moved across special heads which use **electrostatic** processes to place electric charges on small areas (dots) of the paper to attract ink to the selected areas. The small size of these areas (current technology is 400 dots per inch) defines the resolution of the plotter, and now makes the image so well detailed that color plotter output is now becoming acceptable as final form output for maps.

Color plotters are expensive and their technology is still lacking in comparison to that of their vector counterparts. But, the function performed in automated cartography by color plotters can not really be performed by vector plotters. To some extent, the reverse is also true, but the large majority of use in an automated cartography system (edit plots and quality plots in small quantity for local or less formal can be performed.

- D R A F T -

I. I N T R O D U C T O R Y I N F O R M A T I O N

I.5 DIGITAL REPRESENTATION OF GIMMAP DATA

The foundation for any complex computer system involving large data sets is the fundamental structure of the data. The structure of map data in GIMMAP is comprised of the content and interrelationship of records in several types of files, each file representing different spatial entities inherent in map data. These entities and relationships are embodied in the **data structures** which define the content and order of the records of files in the cartographic database.

In the GIMMAP system, each map area is represented by a cartographic database comprised of at least four and as many and nine separate, unformatted files, each a **direct** or **random access** file (may access by record directly without first accessing many other records such as on a tape). These files represent the physical embodiment of parts of the GIMMAP view of map data described above.

The four basic files for a database are the Arc, Point, Qplot and Node. If zones are to be generated and used, then three additional files, the Zone, Border and Label files, are generated. Whenever symbology is added to a map, the symbol and calligraphy files are created. In general, these files contain the following:

- Arc - Information about line segments or arcs, which connect nodes and isolated point features such as well locations or section corners.
- Point- Contains the sequences of interior points which define the paths of linear features.
- Qplot- Parallels the Point file but in a special form for quick display on the graphics terminal.
- Node - Contains the topological relationships between arcs and arc intersections (nodes) providing an identification method for arcs, automatic generation of zone boundaries,
- Zone - Contains information about zones, areas to be delineated based on their primary attribute values. Includes statistics and access to boundary information in the Border file.

Border- Contains information on all arcs which have been used to create zones, including adjoining information and access to the point file records needed to draw the arc.

Symbol- Contains all defining information for symbols, the unit of symbology which may be a letter, a symbol or a title.

Calligraphy - Contains the points defining the pen strokes for the symbols.

Label-

Complete descriptions of each of these files are given below. Each description includes a type designator, a letter used in each file name to indicate the file type, the length of the file in bytes, and the contents of all data records in the file. ??** All these files and values may be examined via RAFPRN (references made in the files records to (other) records in (the same or) other files in the database.** In some files, certain records are set aside for special purposes, and their contents are included.

All files in a GIMMAP database are controlled by the Random Access File Management (RAFMAN) system (described in detail later). This means that the first record in a file is reserved as the accounting record for RAFMAN and that all unused (non-data) records are linked through special marker values in the records.

The items listed as contents of each file will be labelled as to their programming type.

I2 = 2-buffer integer
I4 = 4 byte integer
R = real (4 byte)
An = n-character Alpha-Numeric

ARC FILE

Type Designator: A
Record Length: 42 Bytes
References to Database Files: Point, Qplot

There are two types of records in the Arc files. One represents a true line segment arc (usually) with two distinct end points and zero or more interior points. The second type of arc record represents an isolated point arc which is a single point feature on the map. The two features may be distinguished by restrictions placed on the feature codes.

For the line segment type of arc (feature code 2000 or greater) each record contains:

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I2	FCODE	Feature code of the arc
R	XF,YF	Coordinates of first point (node)
R	XL,YL	Coordinates of last point (node)
I4	PFPR	Pointer to first point record
R	ALEN	Length of arc in digitizer units
R	XMIN,YMIN	Minimum X and Y coordinates
R	XMAX,YMAX	Maximum X and Y coordinates

The feature code of the arc must be 1000 or greater, or the record may be ignored by GIMMAP programs. Since the feature code is only 2 bytes, the maximum feature code is 32,767. The first and last (end) points of the arc represent where digitizing began and ended. These points are not necessarily the maximum/minimum x or y coordinate points, but they may be. These may be the only points for the arc. These two points are the endpoints of the arc and are identical if the arc represents a closed loop. As endpoints, they also represent nodes, and thus are the locations at which identifying numbers are posted to select the arc. These point locations are not repeated in the Point file.

The pointer into the point file is the path to the string of points representing the path of the arc as it was digitized or entered. This string of points will be contained in one or more records in the Point file in the order the points were digitized. When more than one such record is required, the multiple Point records are linked by a string of values in the records, each pointing to (by record number) the next record in the linked list.

It is also possible for a legitimate (straight-line) arc to

contain no interior points. In such a case, PFPR, the pointer to the Point file is set to zero. For unused records in the freelist, the pointer is set to -1. For all used records, the pointer must be positive. Additional information about the structure of the Point file is given below.

The arc length is provided in digitizer units from initial entry into the system. This value may be converted via the scale factor to more useful units. It is used to calculate the perimeter of any zones for which it acts as part of the boundary. The arc length may be altered by GIMMAP when its points are changed in editing.

The minimum and maximum X and Y coordinates (extremes) of the arc are also generated o input and are thus in digitizer coordinates*. These may change if points on the arc are moved outside the original extremes during editing. The box defined by these extremes is used for the clipping process to quickly determine what arcs must be displayed or plotted and which may be ignored for a windowed subset of data. The extremes are also used to calculate extremes for zones and may be used in other operations.

The isolated point form of arc record contains:

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I2	FCODE	Feature code
R	X, Y	Location of the point
I4	NUMZ	Number of used Z-values
R	Z1,...,Z7	Up to 7 Z-values

The feature code is the same as for line segment type arcs, but restricted in value to 1000-1999. The location of the point is the same as XF,YF and XL, YL but condensed since the two points are identical. There are no interior and so no pointer to the Point file. There is no length and the extremes are all the same as X, Y.

Thus the rest of the record is used to store information which isolated points may have, but which line arcs don't have. Many isolated points have one or more values associated with the X, Y location. These values are often numerical and are called Z-values (representing a third dimension). As many as 7 Z-values may be stored in the arc record for an isolated point. The Z-values may be positive or negative, but their number must be indicated in NUMZ. If no Z-values are stored, NUMZ must be zero.

POINT FILE

Type Designator: P
Record Length: 68 Bytes
References to Database Files: None

The Point file contains two records called the first and second information records which contain numerous facts about the map and the database. These are described below.

There is only one type of data record in the Point file. This record consists of the following:

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I4	NEXT	Pointer to the next Point Record
R	X1,Y1	First point
R	X2,Y2	Second point
:	:	:
:	:	:
R	X8,Y8	Eighth point

The NEXT pointer is the link in the list of possibly multiple Point records for an arc. The value of NEXT is zero for the last record in the linked list for an arc. Such a record may also be the first when the arc requires eight or fewer interior points. The value of NEXT is negative only for unused records in the freelist.

When there are two or more records in the linked list for an arc, the first record points to the second via NEXT. That is, the value of NEXT is the first record is the record number of the second Point file record for the arc. This second record for the arc may be anywhere in the Point file. It may even be before the first record of that arc, if certain editing operations have occurred.

As an example, let an arc have 20 interior points labeled X1,Y1, X2,Y2 . . . X20,Y20 as they were digitized. Then one possible arrangement is as follows. The PFPR pointer in the arc record points to record 17 in the Point file.

Then, the arc record may contain...

FCODE, XF, YF, XL, YL, 17, length, ...

And in the Point file we have. . .

<u>Point Record</u>	<u>NEXT</u>	<u>Record Content</u>
17	18	X1,Y1...X8,Y8
18	19	X9,Y9...X16,Y16
19	0	X17,Y17...X20,Y20...

If some editing has been done prior to the entry of this arc, the point records may not be in the same order as the points were digitized. They may in fact be out of order by Point file record numbers with respect to the order in which they are to be used (as implied by order of digitization and entry). But no matter what their order by Point file record numbers, their correct order of use is preserved by the NEXT pointers.

<u>Point Record</u>	<u>NEXT</u>	<u>Record Content</u>
6	43	X9,Y9...X16,Y16
17	6	X1,Y1...X8,Y8
43	0	X17,Y17...X20,Y20

The points in the last record for an arc may number exactly eight and thus perfectly fill the record. More likely, there will be less than eight. In this case, part of the record will be unused. To indicate this fact, all unused positions will be filled with negative values since all digitized points are required to be positive. Thus, the last record in the second example would contain:

<u>Point Record</u>	<u>NEXT</u>	<u>Content</u>
43	0	X17,Y17,...,X20,Y20,-1.,-1.,..., -1.-1.

So, when the last record is encountered (NEXT=0) a simple test for negative coordinates will find the end of the good interior points for the arc.

First Information Record

The first information record is physically the record following the accounting record (which is usually #1). This record contains two values. The first is a two-byte "flag" with a value of -1. This guarantees that the record will never accidentally be used a part of the linked list for an arc. The second value is the record (in A64) is the Map Header for the database. This header may contain anything but usually contains the base name for the cartographic database, the projection and scale for the database, and the date and initials of the operator who last entered data into the database. This header is displayed by editing programs.

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I2	FLAG	Negative value (-1)
A64	HEADER	Alpha-numeric general information about maps and construction.

Second Information Record

The second information record contains information required by the system to properly maintain the coordinate system defined for the database, to perform appropriate transformations of unprojected data;, and to guide proper display of the map as a whole. The contents are:

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I2	FLAG	Negative Value (-1) to exclude from list.
I2	NPRO	Number to indicate projection type
R	SCALE	Scale Factor
R	XMIN,YMIN	Minimum X and Y coordinates
R	XMAX,YMAX	Maximum X and Y
I2	ROW,COL	Row and Column Numbers
R	CPTS(8)	Control Points

The FLAG value guarantees that this record will not be used in a linked list for an arc. The Row and Column method applies (see below). Basic projection choices indicated by NPRO are:

0	=	None
-1,1	=	Modified Polyconic
-2,2	=	Lambert Conformal Conic

Non-projected maps (NPRO=0) are not maps using the latitude-longitude reference system. They are maps for which projection information is unknown or for which; projection is ignored and the features are treated as a fundamentally correct drawing with no connection to a known mathematical model.

The **scale factor** is the factor by which coordinates must be multiplied to be equivalent to the original surface (usually the Earth) which they represent on the map or coded in the database. Thus it is the factor by which lengths and heights must be divided to construct the original map. The **scale** of the map is represented as 1:scale factor, such as 1:50,000 ("one to 50,000") meaning that an inch on the map represents 50,000 inches on the surface of the Earth.

The extremes (XMIN,YMIN,XMAX,YMAX) represent bounds for the map or the database within which all points lie. For projected map data these bounds are fixed by the projection, scale and choice of area. for non-projected map data, these values are determined by what enters the system and may be changed. In projected map data, any point lying outside the bounds of the projection area will be moved to the edge of that area and thus within these fixed extremes.

The row and column numbers represent an alternate method for specifying the latitude and longitude extremes for a map area (see below). The row and column numbers specify a unique map in a fixed grid of maps, thus removing the need for further specifications. For example (used at the KGS), the grid may be the grid of 7.5' (USGS) quadrangle maps at 1:24,000 which covers the state of Kansas.

An arbitrarily imposed numbering system fixes a unique row and column number for each map, and provides direct translation from the row and column numbers for each map to the latitude and longitude limits of the bounding rectangle for each map. The row and column numbering system numbers rows from 1 (south) to 26 (north) and columns from 1 (west) to 62 (east) covering all maps which contain any part of the state. Using this particular scheme, the bounding rectangle map be calculated in degrees by:

```
SOUTH = 36.875 + (ROW-1) * 0.125
NORTH  = SOUTH + 0.125
WEST   = (-102.125) + (COL-1) * 0.125
EAST   = WEST + 0.125
```

The row and column values are used only for maps which belong to the special grid system. Only maps with a positive projection type (1, 2, 3,...) will be defined this way. Maps with projection type of zero or negative will not use row and column, and for these maps the row and column should be zero.

Following the row and column, are eight values (CPTS) which are used differently for projected and non-projected maps. For non-projected maps, these eight values are the digitizer coordinates of the four corners (SW,SE,NE,NW) or control points of the map. These points are used in comparison to subsequent additions of data to effect appropriate translation, scaling and rotation so as to match the two sets. For this case, these points are not extremes of the map, and the XMIN,YMIN,XMAX.YMAX must be used to store that information.

When there is no map projection (NPRO=0), CPTS defines the bounding rectangle:

CPTS (1) = Minimum latitude (SOUTH)
CPTS (2) = Maximum latitude (NORTH)
CPTS (3) = Minimum longitude (WEST)
CPTS (4) = Maximum longitude (EAST)

If the projection type requires additional information, then the remaining four values may be used. For example, the Lambert Conformal Conic projection requires two parallels (intersection circles of the cone with the earth):

CPTS (5) = South Standard Parallel in degrees
CPTS (6) = North Standard Parallel in degrees

QPLOT File

Type Designator:	Q
Record Length:	36 Bytes
References to Database Files:	Self

The Qplot file is a special file built for a single purpose. This purpose is to provide the fastest possible display ("quick plot") of the full view (only) of all map data for the given database. This view is defined as the largest image of the complete given area which may be displayed on the device. This single view is the most useful, but also the most time-consuming view of the map, and hence this special file has been created to minimize the display time by storing a converted, ready-to-draw image, requiring only the reading of the file and the direct writing of the display-ready data.

All other displays are generated in real-time and are governed by feature and areal selections which tend to reduce their conversion time. However, some overhead is encountered in the clipping of this data to fit the selected window and conversion to display form. Unfortunately, the infinite number of such views preclude the possibility of pre-conversion as has been done with the Qplot file.

The Qplot file parallels the Point file in structure with each record containing a NEXT pointer followed by up to 8 points defining the path of the arc. As with the Point file, the NEXT pointers link together the records for a single arc, and the last record contains NEXT=0. In fact, the Qplot records are made as duplicates of the one Point file, but with the points converted to display terminal coordinates. All unused points in the last record for an arc are set to -1,-1.

Records in the Qplot file are directly from the (PFPR) Point file pointers in the Arc file. It is not necessary to use the point file pointers to access the Qplot file. The Qplot records corresponding to the information records (2 and 3) in the Point file are not used. The freelist structure is intact and parallels the freelist in the Point file.

NODE File

Type Designation: N
Record Length: 28 Bytes
References to Database Files: Arc, Node

Records in the Node file represent the arc endpoints and arc intersection points, known as nodes. These points are significant for arc identification, zone boundary generation and potential applications like path-finding in networks. Each node may be represented by one or more records in the Node file, depending on the number of arc incident (intersecting) at the node. This number is the degree. Each record contains information for up to four intersections. Thus a node with 14 arcs intersecting would require 4 records. The last record would be only half-used. In general the node record contains:

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I2	NEXT	Pointer to next node file record
I2	DEG	Degree of the node
R	X,Y	Location of the node
I2	N1,A1	Node (N1) connected by arc A1
.	.	
.	.	
.	.	
I2	N4,A4	Node (N4) connected by arc A4

The NEXT pointers are used the same as in the Point file. If the record is the last for a node then NEXT=0. If there are more records for the node, then NEXT= the record number of the next Node record for the node. As with the Point records, these numbers are not necessarily in increasing order. Unlike the point records, the order of records for a node does not matter, only the list of connected nodes and arcs is of importance. All used node records must have a non-negative NEXT value or they will be ignored by GIMMAP.

The degree of the node is the number of arcs intersecting at the node. This number includes all isolated point arcs which coincide with the node location. There is no limit on this number (practically speaking). The degree is stored as the second value of the Node record. However, when a node degree exceeds four, more than one record is required for the node. In these cases, the degree is not stored in records beyond the first, called the anchor record.

All subsequent records replace the degree value by a pointer to (the record number of) the anchor record, but with a negative value. Thus, if the first (anchor) record is 143, then all other

records in the linked list will have a value of -143 instead of the degree. This pointer could help to reconstruct the Node file in the unlikely event of a catastrophic error, and can be helpful during examination by RAFPRN following smaller errors.

The location (X,Y) of the node is the averaged position of all arc endpoints which fall within a **threshold** distance selected by the operator in the execution of the NODGEN program when the Node file is built. This location is also made to be identical to all these locations by NODGEN. It may be altered during editing, but is maintained consistent for all arcs through the topological information kept in the remainder of the record.

The remainder of the Node record contains the node-arc pairs for up to 4 connecting arcs. These pairs define the **topology** of the arc endpoint or node network. Each pair consists of the record number of a node in the Node file which is connected to "this node" (i.e. the node which is represented by the Node record containing the node-arc pairs) by one or more arcs.

Associated with each node (for the node-arc pairs) is the record number of the arc by which the two nodes are connected. Furthermore, each Arc record number in the node-arc pairs is signed positive or negative to indicate the direction of the arc. If the arc was digitized from this node to the other node, then the arc is positive. If the arc was digitized from the other node to this node, then the arc number in the node-arc pair is negative.

For example, consider node 50 which is connected to three other nodes as follows. Arc 45 begins at node 26 and ends at node 50. Arc 23 begins at node 98 and ends at node 50. Finally, arc 46 begins at node 50 and ends at node 42. The degree of node 50 is then 3. The node-arc pairs for node 50 are as follows:

<u>Node</u>	<u>Arc</u>
26	-45
42	46
98	-23

Arc 45 and 23 are given as negative, because they begin at other nodes and end at node 50. Arc 46 begins at node 50. Remember that all arcs appear twice in the node-arc pairs of the Node file. Each appears once as a positive for the "start" node of the arc and once as a negative in the Node records for the "end" node (in the Node records too). Thus, for example, the node-arc pairs for node 98 above will contain the entry:

<u>Node</u>	<u>Arc</u>
50	23

The node-arc pairs provide identification of arcs by operators entering the node numbers of the endpoints. This topological information also provides for automatic generation of zone boundaries (described later). All unused node-arc pairs (in the last record) are set to zero.

ZONE FILE

Type Designator: Z
Record Length: 48 bytes
References to Database Files: Zone, Border, Label

The Zone file contains a record for each separate area of color on the map. These areas are called zones in GIMMAP and correspond to the "Polygons" referred to elsewhere. These areas are the units which must be recognized as individuals for proper color separation for printing of color maps or creation of color or hash pattern area files for in-house map plotting.

The information in the Zone file includes a link to the Border file which contains information about the chains of "border areas" which define the boundaries of the zones. The Zone file also contains "zone color"- a number identifying the (printed) color or major attribute value (e.g. geologic formation, average rainfall...) which is to be displayed on the map. Zone records contain information about "island relationships" which define when one zone lies completely inside (is an island of another (the "surrounding") zone. Since any number of levels of islands may occur and any number of islands may exist within a single surrounding zone, three separate values are kept in zone records.

Zone records also contain a link to the chain of label records associated with each zone. Labels are identical to symbols, but are linked together and to the zone for which they are created. Finally the perimeter (created from the lengths of the area file), area (in square digitizer units which may easily be converted) and the extremes of the zone. The extremes define a rectangle which is used for fast display when windowing (as for areas before) and also used to hasten the determination of islands which may or may not belong to a selected surrounding zone.

The content of each zone file record is:

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I2	COLOR	Zone color or major attribute
R	XM, YM	Location of zone mark
R	SIZE	Size of the zone mark
I2	PFBR	First border record
I2	ISLWI	First island within
I2	ISLOF	Surrounding zone
I2	NEXTI	Next island
I2	PFLR	First label record
R	PERI	Perimeter in digitizer units

R	AREA	Area in square digitizer units
R	XMIN, YMIN	Minimum X and Y coordinates
R	XMAX, YMAX	Maximum X and Y coordinates

The COLOR value in each zone record is the single indicator of what each zone represents. For example, in a surface geology map these values would identify the different types of formations which are to be distinguished by different colors (or other technique) on the finished product. The values in COLOR may simply reflect the indices to a table of values for this major attribute or they may actually be the value of the attribute. The former is represented in the geology map described above. The latter case may be illustrated by considering a map in which annual rainfall in inches is to be displayed by regions. In such a map, the values in COLOR may be the actual rainfall values for each area.

A third technique for the values in COLOR is to completely define printing colors for each area as a result of values associated as described above. Then, each color may be assigned an index number from a list of the actual percent color may be encoded into a single value. Then, either the color index or the color code is used for the value of COLOR in the zone record.

The COLOR values should be positive and are restricted to a maximum of 32767 since only 2 bytes are used. Values greater than 30000 may be given special significance by the ZONEDT program and are usually reserved for "background" zones and "complex" or "ghost" islands. A record with a negative COLOR.

The location of the zone mark is originally generated by ZONGEN as the center-of-mass for the zone. The location of the zone mark may be changed in ZONEDT since the object is to have a single mark within the zone boundary identification and selection of the zone. for some zones, the center-of-mass does not even lie inside the zone boundary.

Each zone mark is assigned a default size for ZONGEN. This size is chosen so as to make the mark visible in plotting as well as in graphic display in ZONEDT. The size may be diminished in ZONGEN for zones whose areas are so small that the default mark can not fit. Zone marks are also changed in ZONEDT to cause the marks to lie completely inside the zone boundaries or to be made more visible.

Three pointer values (numbers of zone records) exist to provide complete information on island relationships. The first ISLWI points to another zone which lies within the zone described by the zone record. If the zone has islands within it then ISLWI is zero. If there are islands within this zone then they will be linked into a list by their NEXTI values, much like Point and Node records are linked. Then, ISLWI will point to the first island zone in the list. For example, if island zone 53 lies

within zone 100 and is the only island in zone 100 (or is the first island in the linked list) then the island value in zone 100 will be 53.

The primary importance of this connection is that whenever the boundary of a zone is produced, the boundaries of all of its islands must also be produced. To properly color a zone, one must cease coloring at the boundaries of the zone as well as at the boundaries of all the islands. The islands must (usually) be of a different color or they would not have been included.

The second pointer value is ISLOF which indicates whether or not the zone is an island of some surrounding zone. In fact, ISLOF indicates whether the zone is an island, a complex or ghost island, or not an island. These three states are determined by ZONGEN which set ISLOF to zero for non-islands, to the (positive) zone number itself if it is an island and to the negative zone number itself if it is a complex or ghost island.

In ZONEDT, ISLOF values are reset to identify the actual zone numbers of the surrounding zones. Since there can be only one surrounding zone, there is no linked list associated with ISLOF values. As an example, in the previous example island zone 53 was inside surrounding zone 100. For zone 100, ISLWI was set to 53. For the island zone 53 then, ISLOF would be 100.

The third value in the zone record used for island relationship is NEXTI for next island. The value is the zone record number of another island zone which has the same surrounding zone as this island zone. The two island zone records are a part of a linked list containing the islands which are within the same surrounding zone. In this linked list, all the island zones linked have the same ISLOF value.

Unlike the linked lists in the Point file and the Node file, the NEXTI linked list is **circular**. That is, the NEXTI value in the last island in the list is not zero, instead, it is a pointer back to the first island in the list. It is identical to ISLWI for the surrounding zone. As an example, let island zone 18, 52 and 23 be islands of surrounding zone 226. If island 23 was the first in the list then these zones may have the following values:

<u>Zone</u>	<u>ISLWI</u>	<u>ISLOF</u>	<u>NEXTI</u>
:			
:			
18	0	226	23
:			
:			
23	0	226	52
:			
:			
52	0	226	18
:			

:
226

23

0

0

This example assumes that islands 18, 23 and 52 do not also have islands within them and that surrounding zone 226 is not also an island of some other zone. Note that, when they are unused, both ISLWI and NEXTI are zero. Also, NEXTI values are initially set to zero by ZONGEN, but are reset in ZONEDT as island relationships are defined.

The next value in the Zone record is PFLR, the pointer into the linked list of Label records associated with the Zone. These labels parallel symbology records and have in turn pointers into the Calligraphy (pen stroke) file. Labels differ from symbology in that they are associated with and may be selected via the zone. If there are no labels associated with the zone, then PFLR will be zero.

The perimeter is the length of the zone boundary in digitizer units, basically as it was digitized. It may be easily converted to ground or other units. The area is represented as square digitizer units (usually inches or centimeters) and may likewise be converted to more useful units based on the scale or other projection parameters.

The extremes are essentially in digitizer coordinates but may have been altered on input to force the data to fit a mathematical model based on projection information. At any rate, they define a box within which the entire zone lies. This box is used to quickly distinguish between zones which are completely in, completely out or partially in a selected window. Using this approach, the overhead involved in the process of clipping for a new view is greatly reduced. And finally, the extremes are similarly used for selection of potential islands in ZONEDT.

BORDER FILE

File Designator: B
Record Length: 34 bytes
References to Database Files: Border, Zone, Point

The Border file provides access to the boundaries of zones. The boundary of each zone consists of a sequence of one or more "border arcs" which are distinguished from arcs by the fact that border arcs are a selected (by feature code) subset of all arcs to be used for zone boundary generation in ZONGEN.

The Border file contains this set of arcs along with information about the zones on either side of each arc, pointers into the Point file for drawing the arc and linking information which allows for the correct sequencing of arcs around a border and points with an arc. This provides the means to sequence all arcs and points to construct a continuous (counter-clockwise) boundary completely around each zone.

Each record of the Border file contains:

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I2	NARC	A r c n u m b e r
I2	LZONE	Left zone
I2	LCODE	Color of left zone
I2	NEXTL	Next left border record
I2	RZONE	R i g h t z o n e
I2	RCODE	R i g h t c o d e
I2	NEXTR	Next Right Border record
I4	PFPR	First Point File record
R	XF,YF	First endpoint coordinates (node)
R	XL,YL	Last endpoint coordinates (node)

The arc number, NARC, is stored to reduce waste which would occur if only the records of the border arcs were used (and many other arcs were not). The arc number is the record number from the arc file of the arc which forms part of the boundary for two adjacent zones. Thus it is always a positive value.

The next three values in the Border record all refer to the zone on the left of the border arc, as that arc is stored in the database. This will usually be as it was digitized unless editing operation in GRFEDT might have reversed it. Thus, if traveling along the border arc, LZONE is the Zone file record for the zone on the left-hand side of the arc. Further, LCODE is the zone color of that Left-hand zone.

The NEXTL pointer is the linked list connection to the next border arc in the boundary of the zone on the left-hand side of this border arc. The next border arc is the next one in the sequence of border arcs which describe the zone on the left-hand side in a counter-clockwise manner. This convention is necessary since there will always be two ways to go from a given border arc. The counter-clockwise tradition was initiated in ZONGEN and is maintained here.

The NEXTL value is the number of the record in the Border file which represents the border arc which follows this border arc in a clockwise following of the zone boundary. It should be noted that this second border arc may be stored (was digitized) in the proper direction for describing the zone on the left of the first border arc. Or, it may be stored in the reverse order. Thus in it's record the zone on the left of the first border arc will instead be on the right. Information about that common zone will be stored in the second triplet (RZONE, RCODE, NEXTR) for that second border arc.

Whether the next border arc will reference the information about the zone on the left of this border arc is indicated by the value of NEXTL in this border arc record. If the next border arc has this common zone on the left (as it was digitized), then NEXTL will be positive. If the next border arc has the common zone on its right, then NEXTL in this Border record will be negative.

Thus the NEXTL values (with NEXTR values) from linked lists which tie the border arcs of each zone together for a quick retrieval of a complete and coherent boundary for each zone. As with the NEXTI values in the zone file, these lists are circular. The last Border record for a zone will point to the first in the list. A simple island zone (with only a single border arc) will have a single Border record with NEXTL pointing to the single Border record itself.

The second triplet of pointers apply similarly to the zone on the right of the border arc. The values of RZONE, RCODE AND NEXTR are defined as above for the first triplet of values, but with respect to the zone on the right-hand side.

The last five values in the Border record provide direct plotting or access of the border arc points which define the path of the line. Inclusion of PFPR, the Point file pointer and the arc endpoints (XF,YF and XL,YL) in the Border record eliminate the need for access through the Arc file. This access would otherwise be required to plot or display every border arc.

Simple island zones which have not been associated with surrounding zones in ZONEDT will have Zone, Color and Next values of zero on the right-hand side. (If ZONGEN does not automatically reverse simple islands which were digitized in a clockwise manner, these zero values may be on the left-hand side). For complex or ghost islands which have not been

associated with surrounding zones, the border records will be linked so as to define the island in a clockwise manner. The Zone, Color and Next values on the "inside" will reflect the non-island zones which are interior to the complex island.

SYMBOL File

File Designation: S
Record Length: 34
References to Database Files: Calligraphy

The Symbol file contains a record for each unit of symbology created with the CYMBAL program. Each unit may represent a single character (letter, digit) or a special symbol (highway or well symbol) or a unit may be as many as 64 characters long (such as a title). These units are called "symbols" in general and each symbol is associated with a record in the Symbol file. A symbol is then defined by the information in it's record in the Symbol file and by the sequence of pen strokes required to draw the symbol. The pen strokes are stored in the Calligraphy, in circular limited lists similar to NEXTI in the Zone file and the NEXT pointers in the Border file.

The Symbol records contain the following:

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I2	CODE	Feature code of the symbol
R	XR,YR	Fixed reference point
I2	JUST	Just if is action of symbol
R	SLENG	Length of symbol
R	CHGT	Character height
I2	SET	Set or Alphabet
R	ORI	Orientation (rotation)
I4	PFCR	Pointer to Calligraphy record
I2	NEXTS	Next Symbol record
I2	NCHRS	Number of characters in symbol

Each symbol is assigned a feature code as are arcs and isolated points. These codes are then used to select subsets of the symbols for plotting in different colors or pen widths. As with arc feature codes, these feature codes must be at least 1000 and may not exceed 32,767 because of the 2-byte size of CODE in the Symbol record. If the CODE in a Symbol record is less than 1000, the record will be ignored. The CYMBAL program also allows for selective display of symbols based on these codes.

The reference point for the symbol serves to locate and identify the symbol and to translate the points defining the symbol. Translation and rotation are based on justification of the symbol. If left-justified, the reference point is assumed to be the lower-left corner of the symbol prior to rotation. In center or right justification, the reference point is the bottom-center or bottom-right of the symbol prior to rotation. The reference point is also the point about which the symbol is rotated, if that is selected.

The symbol length represents the distance from the lower-left to the lower-right corner (prior to rotation) of the symbol. This distance is in map units or digitizer units. This length and the character height, selected to determine the size of the characters and of the symbol, combine to provide an indirect method to speed clipping for windowed views.

The character set number or alphabet number is a selection of one of the character fonts or special character sets from a dictionary, the available Hershey There are twenty-three available sets (1- 23). The Hershey fonts (A.V. Hershey, U.S. Naval Weapons Laboratory, Dalgren, Virginia, 1967) provide templates of pen strokes which produce the various characters for the 23 alphabets. This subsystem will be described in detail in a later section. Each symbol may be rotated about its reference point. The orientation of a symbol is defined by a positive angle specified as a value from zero to 360 degrees.

Each symbol record is associated with at least one Calligraphy record. The first such record contains the content of the symbol, up to 64 characters, as it was entered (in ASCII characters) on the graphics terminal keyboard. The PFCR value in each Symbol record is the record number of the first Calligraphy record associated with the symbol, and containing the content of the symbol. If the symbol requires more than one Calligraphy record (as do most), they will be linked together as described in the Calligraphy file section below.

Two or more symbols may be linked together to form a complex symbol which allows for editing operations on this complex symbol to be performed a single time rather than once for each simple symbol. An example is to join the interstate highway symbol from one character set with the number that identifies the route of the highway from a different character set. Then, a rotation, scaling, translation or copy command will operate on the numbered highway symbol as a unit. Both simple symbols, the interstate symbol and the highway number, exist as symbols and may be edited separately. Simple symbols are joined to form complex symbols through the NEXTS pointers which form a circular linked list, with NEXTS for the last symbol pointing to the first. Simple symbols which are not part of a complex symbol have NEXTS set to zero.

The number of character typed at the keyboard to specify the content of the symbol is the last value in the Symbol record. it should be noted that in most cases the characters typed at the Keyboard are those which form the content of the symbol. However, when special alphabets are used, the content of the symbol will be greatly different from the keys typed. In this case, the keys typed and the characters stored (in the first Calligraphy record) are the same, but they are only indices to the pen strokes in the Hershey alphabet. The number of characters in the symbol is kept to indicate where in the first

possibly second) Calligraphy file the pen strokes for the symbol begin.

It should be noted that the values stored in the Symbol records are used in conjunction with the pen strokes of the Hershey alphabet files to create the proper pen strokes for the symbol. These values are then stored in the Symbol record so that future editing operations may be performed to modify any aspect of the symbol separately, without having to completely re-create the symbol and its pen strokes again.

CALLIGRAPHY File

Type Designator: C
Record Length: 68 bytes
References to Database Files: Calligraphy

The Calligraphy file is to the Symbol file as the Point file is to the Arc file. The sequences of points which define a path for display or plotting, both provide for multiple records of these points or pen strokes linked together in a circular list. Both have eight points per record with a NEXT pointer at the beginning to join multiple records.

There are a few basic differences in the two. The pen strokes in symbols require a pen status (pen up, pen down) value since, unlike an arc, the symbol may have many unconnected lines. The first Calligraphy record for a symbol or label contains the (ASCII) characters typed at the Keyboard to define the content of the Label/Symbol. The number of these characters is a part of the Symbol or Label record and is used to determine where the pen strokes begin in the first Calligraphy record. Finally, the unused points in the last Calligraphy record to each symbol are set to -999., -999.

The Calligraphy record contains:

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I4	NEXT	Pointer to next Calligraphy record
R	X1,Y1	First point
:	:	:
:	:	:
R	X8,Y8	Last point

As described above, the NEXT values form a circular, linked list of all Calligraphy records for a symbol in the order they were created. Being circular, the NEXT value in the last record of the list points to the first Calligraphy record in the list for the symbol. Thus, NEXT in the last record is equal to PFCR in the Symbol record.

The pen stroke coordinates and stores eight to a record except for the first and possibly the last records in a linked list for a symbol. They are used in the order stored, from point 1 to point 8, and in the order of records defined by the NEXT pointers. If used otherwise, the result would generally not be the denied symbol.

Pen status values are coded in the points themselves. If the coordinators are positive, the pen status is down ("draw"). Otherwise, the pen status is up ("move"). Unused points in the last record are set to -999, -999. This should not be confused as pen up to 999,999 because the reference area is (assumed to be) smaller in size than 999 by 999. Hence, there can be no legitimate coordinates of that magnitude.

The first record for each symbol contains a string of characters representing the content of the symbol. This string follows the NEXT pointer. If there is enough space following the string of characters, there will be points for pen strokes to begin the symbol. Each point requires X and Y and thus 8 bytes. The string of characters requires 1 byte for each character, and the NEXT value uses 4 bytes. Since the record length is 68 bytes, there are 64 left for the string of characters and any points (X,Y) for pen strokes.

So, if the symbol used the maximum of 64 characters then no points could be put in the first record. In fact, since each point requires 8 bytes, any symbol using more than 56 characters would have no points in the first Calligraphy record. For each symbol whose content becomes 8 characters smaller, one more point may be kept in the first Calligraphy record. It is possible in special cases (the BOX and LINE commands in CYMBAL) to create symbols for which no characters are required. In such cases, the entire first record may be used for pen strokes.

The number of pen strokes N in the first record may be calculated, given the number of characters NCHRS in the symbol:

$$N = \frac{68 - NCHRS - 4}{8}$$

This assumes integer arithmetic, in which fractions are dropped. Finally, the unused space (if any) between the content character string and the first (X,Y) point in the first record is filled with blanks.

LABEL File

File Designator: L
Record Length: 28 bytes
References to Database Files: Label, Calligraphy

The Label file is similar to the Symbol file, but is designed for applying identifying labels to zones. The best example is that of applying the 1-3 letter symbols identifying formation type on the (formation) zones of a surface geology map. These symbols identify the geologic formation in as many places as necessary whether the formations are color-coded or shaded or neither. As such, labels are symbols which are associated to specific zones.

Label records keep almost the source information about their labels as symbols do: height, length, content, alphabet, location of the reference point, and a pointer to the same Calligraphy file as the Symbol record, and the content character string is again stored in the first Calligraphy record for the symbol. Multiple Calligraphy records are linked in a circular list.

Labels are also different from symbols. Labels are not connected to form complex symbols, but are all linked in circular lists for each zone. Labels are always oriented horizontally (0 degrees) and centered about the reference point. Finally, since some zones may be too small for a label, an optional "hook" or line from the reference point (outside the zone) may be generated to the zone mark.

The contents of the Label record are:

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I2	NEXTL	Next Label record
I4	PFCR	First Calligraphy record
R	XL,YL	Label reference point
I2	HOOK	Hook option
I2	FONT	Hershey alphabet
R	SLENG	Symbol Length
R	CHGT	Character height
I2	NCHRS	Number of characters

All labels associated with a single zone are linked in a circular list by the NEXTL pointers. The last record in the list for a zone will have a NEXTL value equal to the PFLR in that Zone record. Thus, when a subset of zones is selected, only the associated labels will be displayed or plotted.

The pointer to the first Calligraphy record for the label is the entry point to the pen strokes which make the label. The circular list in the Calligraphy file which is reached through this PFCR value is the same as that described above.

The reference point for the label is used to locate (and identify) the label, which will be centered at the reference point and horizontally oriented. The label is tied to a zone, and in the case where the zone is too small a hook may be created. The hook is simply a line from the reference point of the label to the zone mark to connect the label to the zone to which it applies. When the hook is selected, the value of HOOK in the label record is set to 1. When not used, HOOK is zero.

As in the Symbol file, the selected Hershey alphabet is indicated in the Label record in FONT, and has a value from 1 to 23. The Hershey fonts are shown in the appendices. The length of the generated label is kept in stored (usually digitizer) units in SLENG. The height used to generate the Label (basically the character height) is kept in CHGT in similar units. The string of characters defining the content is kept in the first Calligraphy record. The number of characters, NCHRS, provides a means to know where points begin in the first Calligraphy record as described above. As with symbols, the maximum number of characters is 64.

These values describing the label are saved after the creation of the Calligraphy records for the label so that editing operation may be performed more efficiently as described above for Symbols. In the case of labels, these operations would be performed with the ZONEDT program rather than with CYMBAL.

FREELIST STRUCTURE

All files in a GIMMAP digital cartographic database are built around a fundamental structure which provides basic file operations and maintenance. This fundamental structure is called the Freelist structure, and consists of three parts. One part is the set of "used" records, where used means they have been requested for use, have been filled with information and have been in some way marked so as to distinguish them as such. The second part of the Freelist structure are "free" or "unused" records which are not used records and are not marked as such. Such records may have been used in the past or may never have been used, but currently do not belong to the used set of records. These records have also been marked to distinguish them as part of the free or unused records.

The third part of the Freelist structure is a single record known as the Accounting record, to be used for management of the free and used parts of the file. This record is marked to distinguish it from the used records and is not a part of the free records. It is known uniquely by its location to be what it is.

This basic Freelist structure is then imposed on all cartographic database files in GIMMAP by all programs accessing the database through a sub-system known as RAFMAN for random-access file Management. The RAFMAN subsystem is described in detail in another section but the two primary functions of the subsystem are to:

1. Remove or GET a free record (number) from the free list part of the file, for a GIMMAP program.
2. Return or PUT a used record identified by a GIMMAP program to the free list part.

All records in all GIMMAP files must be obtained from the freelist by the GET function prior to being used, and must be PUT when deleted. The advantages of using RAFMAN are conservation of file space, avoidance of file compaction (to reduce wasted space) and immediate availability of needed records. The uniformity of the RAFMAN operations simplifies access for all programs.

The accounting record is used to keep track of the free list (Free records) part of the file, which is maintained as a doubly-linked list. Records in the freelist are linked by Next or Successor pointers and also by Last or Predecessor pointers.

The content of a freelist record is:

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I2	FLAG	Indicates a Freelist record
I4	PRED	Predecessor in the Freelist
I4	SUCC	Successor in the Freelist

The FLAG is set to -1 as an indication that the record is not used, and with the exception of the Accounting record, is a member of the freelist. All GIMMAP programs are responsible for marking used records to distinguish them from free list records. This means that GIMMAP programs must put a positive value in the first 2 bytes. This value may represent anything (feature code in the Arc file, or COLOR in the Zone file), as long as it is positive.

The other two values in the Freelist record, PRED and SUCC are used to form a doubly-linked list of Freelist records. All Freelist records are linked together in these two lists. The PRED pointer points to the previous or predecessor record and SUCC points to the next or successor record. Since the first record in the list has no predecessor, its PRED value points to the first record itself. Similarly, the last record has no successor and its SUCC value points to the last record itself.

The Freelist Accounting record contains:

<u>Type</u>	<u>Name</u>	<u>Meaning</u>
I2	FLAG	Marks the record as not available
I4	FIRST	First record in Freelist
I4	LAST	Last record in Freelist
I4	MAX	Maximum number of records
I4	USED	Number of used records
I2	LENG	Length of records in bytes

The FLAG marks the record to indicate it is not a part of the Used records. This is actually redundant because the Accounting record is uniquely identified and separated from both the Used and the Freelist records by its location. As the first record in the file, the Accounting record is (generally) not accessed by GIMMAP programs in their search for used records. On rare occasions, some programs need to know some of the values in the Accounting record.

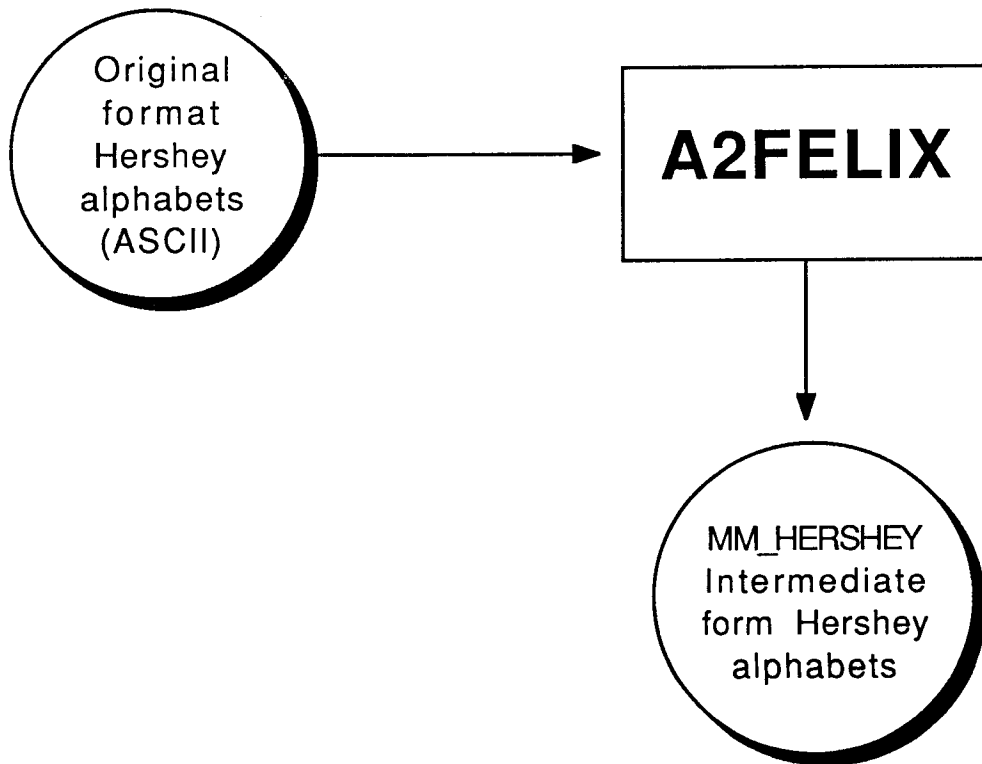
The size of the information in the Accounting record (20 bytes) determines the minimum record length of any file to be built for the GIMMAP system. All current files exceed that minimum by at least eight bytes. Any smaller size would require removal of some values from the Accounting record. Throughout the system, a change that would be possible but expensive.

I. INTRODUCTORY INFORMATION

I.6 SOFTWARE OVERVIEW

A2FELIX (Convert Hershey Sets From ASCII to Felix Form)

The occidental Hershey graphic alphabets (A. V. Hershey), used by Gimmap, exist originally in a special ASCII format. This format is used for ease in transferring the data between systems, but is not an efficient structure or form for production of map annotation. A2FELIX converts the original form Hershey file (HERSHEYA) into an intermediate form used originally by the FELIX system (M. J. McCullagh). For use in GIMMAP, the Hershey data must be further converted into a third form, consisting of two random-access files. This conversion is performed by the HERSH program described below).

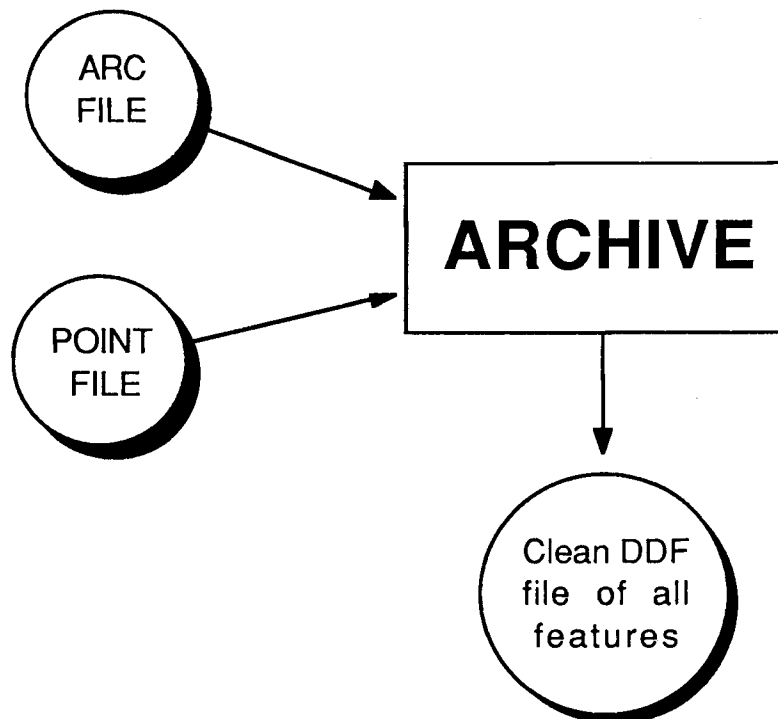


Primary conversion of Hershey alphabet sets

ARCHIVE

(Archival of Map Data)

At various stages in the life of a GIMMAP cartographic database it is desirable to convert the information in the binary, direct-access files of the database an easy-to-read and use ASCII form which may be stored as a back-up or transferred to other sites. This ASCII form is precisely the standard DDF form used by GIMMAP for the input of original digitized data, but has both syntactical and graphical errors removed if the database has been edited. ARCHIVE creates this DDF file by grouping features of each type to reduce headers in the file. The result is suitable for archival, transfer or direct reconstruction of the database in the event of a catastrophic event.

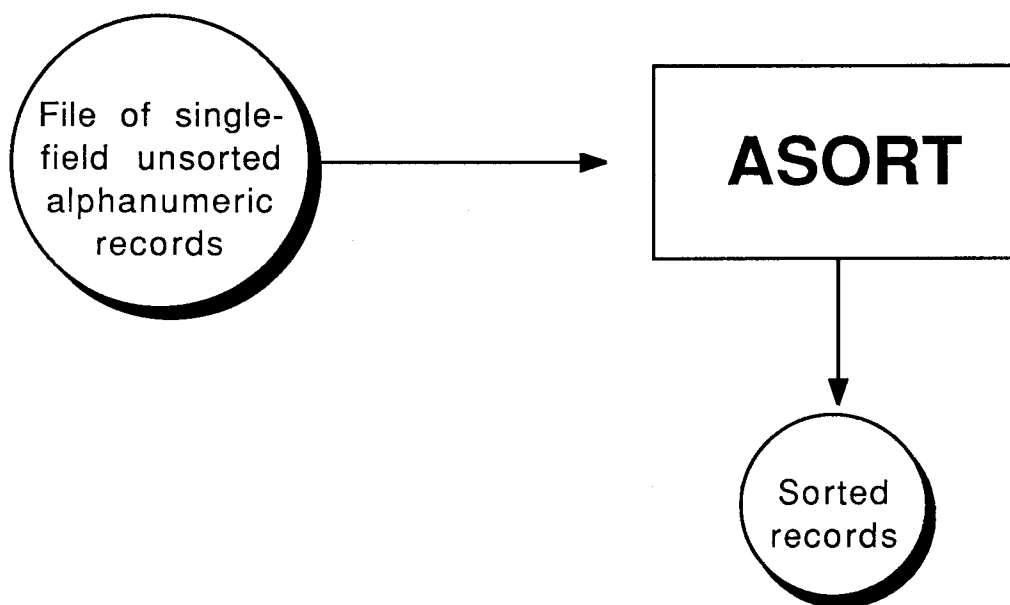


Creation of backup DDF (Digitized Data Format) file of syntactically and graphically clean features from a GIMMAP database.

ASORT

(Alphabetic Sorting)

This program serves the single purpose of performing rapid sorting of lists of elements which are alphabetic or alpha-numeric. It is used to sort file names for input to GIMMAP programs (such as MAPGEN primarily as a convenience to operators of the programs. Processing of files in sorted lists allows a simple determination of the progress of the program toward completion. In the case of creating counties from county patch databases, it may reduce pen movement in plots since county patches will be sorted into contiguous, sequential blocks.

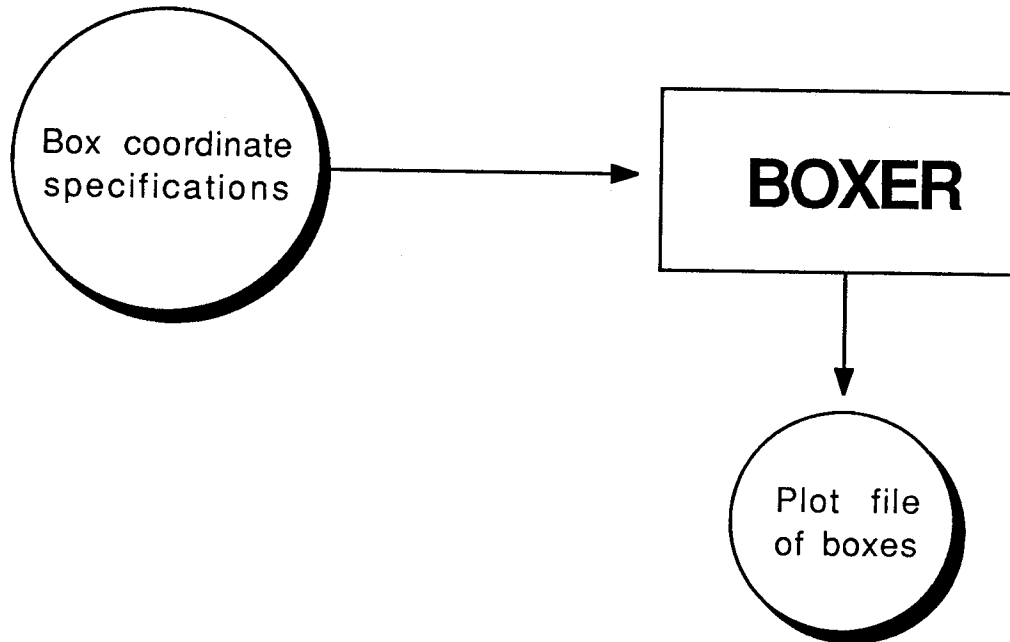


Quicksort algorithm applied to sort an alphanumeric list

BOXER

(Creation of Box or Rectangle plot)

Boxes (rectangles) are required in map production for many purposes. Among these are the map border - a single or double rectangle which surrounds the map and legend, boxes containing the legend or other textual information, and boxes containing shade lines or color for definition of map feature classes. BOXER provides direct creation of rectangles in a plot file format.



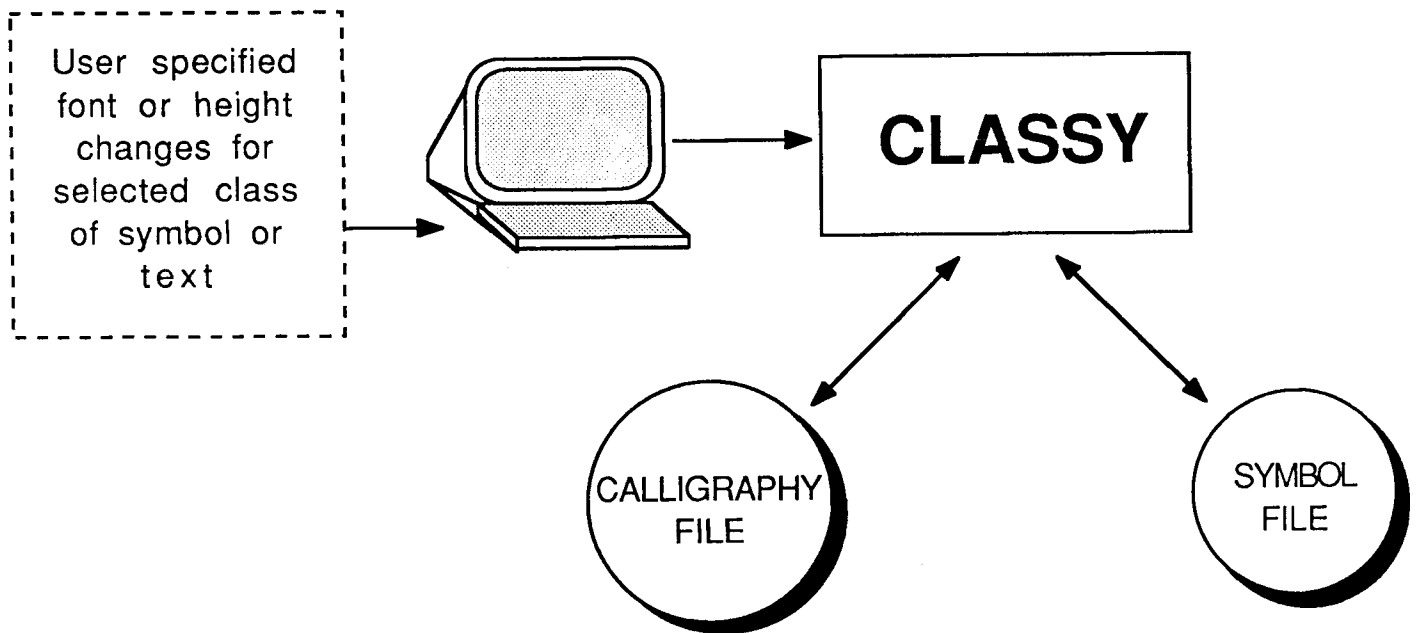
Creation of plot files of boxes for frames, legends, etc.

CALPLOT(T) (Create color plot of zone information)

Zones in a database which have been colored via the ZONEDT program may be converted to plot form for the color plotter in either an online or and offline form. Zones may be selected by groups of zone colors to be mapped into plotter colors which are defined by the operator interactively or in a selected dictionary file mapping zone colors to percentages of primary colors or to pre-mixed plotter colors.

CLASSY (Class Editing Operations on Map Symbolology)

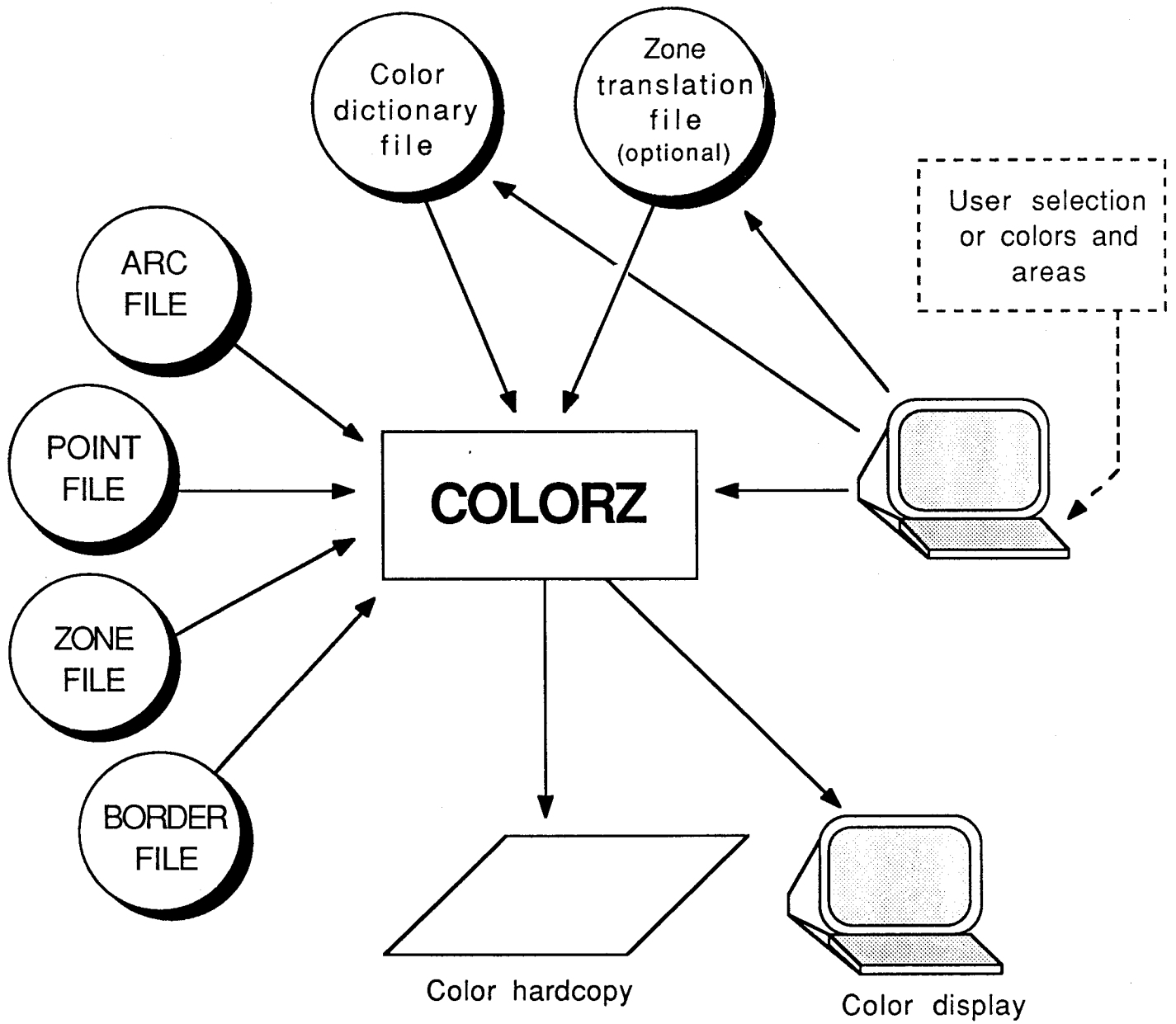
Annotation generated with the CYMBAL or MAPTTL programs may include many classes of symbols, such as highway type and number symbols, and names of perennial streams each distinguished by a unique feature code. When wholesale changes are desired for entire classes of symbols, a great deal of operation time is required to accomplish the task. CLASSY performs two such operations on (feature code) classes of symbols without operator intervention. Given the map base name, CLASSY modifies the font (Hershey Alphabet) or height of selected classes of symbols.



Editing entire classes (feature codes) of symbology

COLORZ (Display Zones by color selections)

Areal (zone) features such as geologic formations and petroleum fields are best examined in a color display in which the different colors each represent the feature types (oil field, gas field). COLORZ provides selection of zones for display based on subsets of feature types (zone colors), and superimposes on the colored areas the user-selected subset of point and line base data such as highways and rivers. Display colors may be mixed from primary colors and selected directly or by use of a dictionary, mapping feature types (zone colors) to display colors of selected (%cyan-%magenta-%yellow) mixtures. The dictionary entries may be edited and saved for future use. Optionally, "zooming" in or out may be performed to examine small areas of the map in detail. Paper maps may be obtained by "hard-copying" the graphics screen containing the map image.



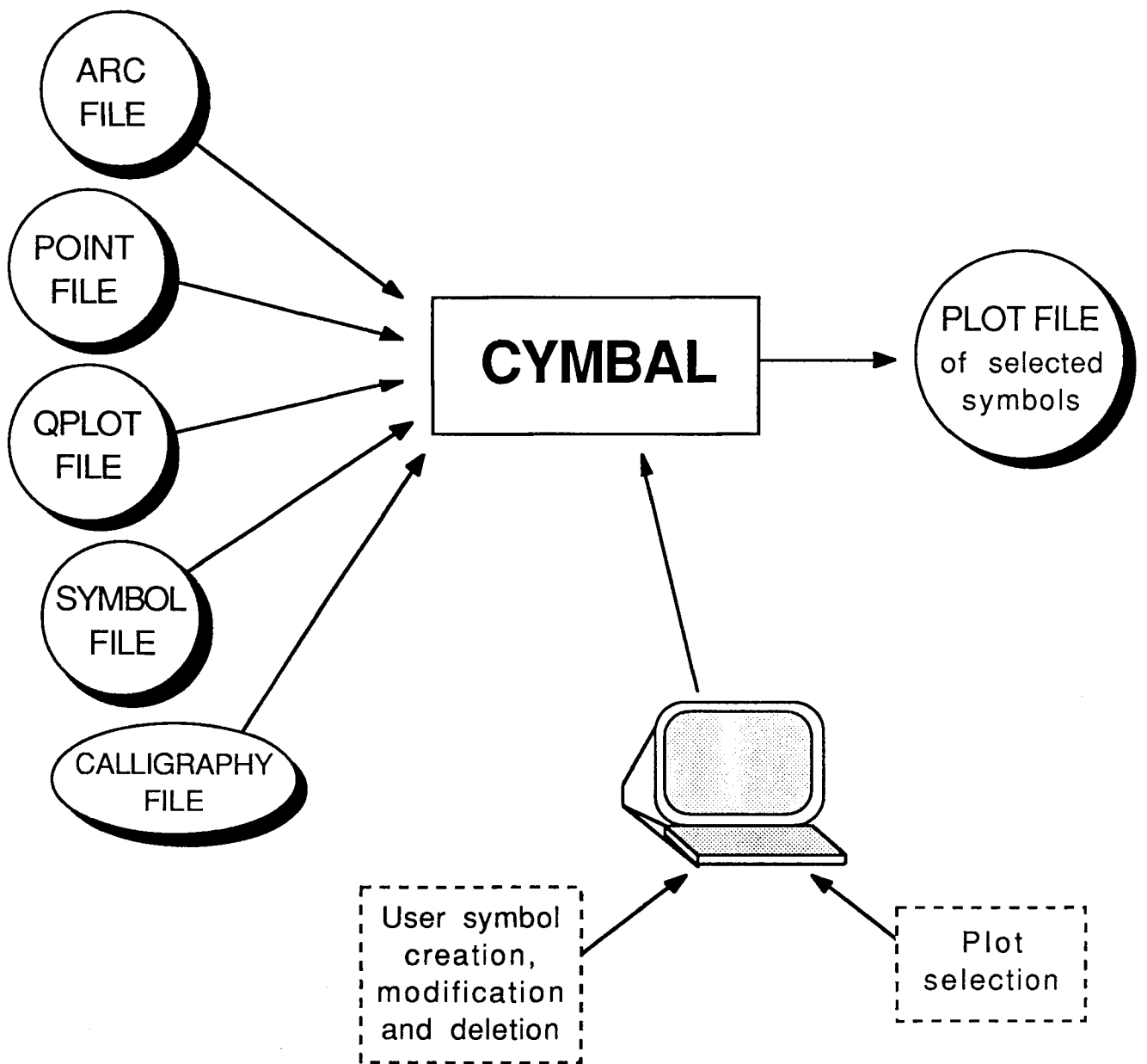
Color display and hardcopy production of selected zones

CYMBAL

(Symbology Editing)

Symbology editing is the creation, placement and modification of symbology on a map. Map symbology may include the title, legend, scale bar, north arrow, map frame and labelling of map features such as river names and highway numbers. CYMBAL provides for the creation, modification and deletion of map symbology of all these types. Map symbology is placed directly on the display of the map being edited by use of a graphics display terminal. Each symbol is drawn to (the display) scale, and its relationships with all map features and other symbology may be directly observed and changed if desired. With the exception of content, all six aspects (height, content, Hershey alphabet, justification, reference point, feature code) defining a symbol may be modified.

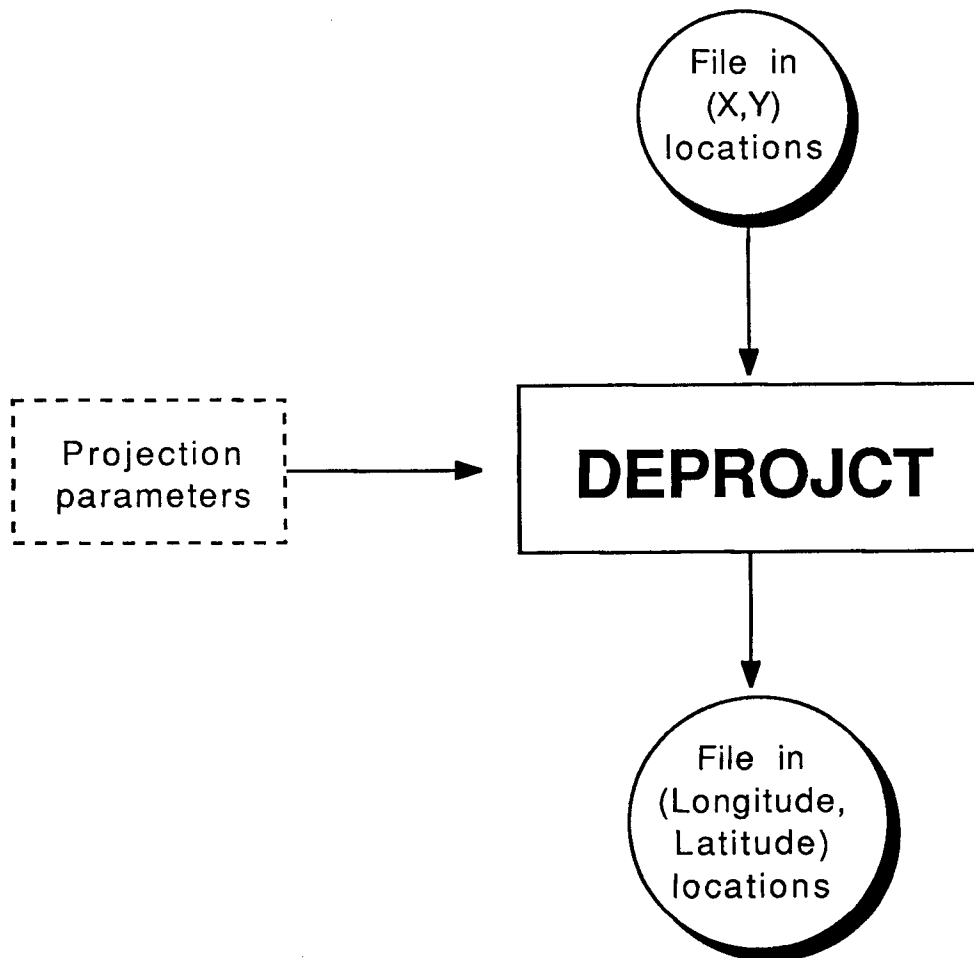
All aspects except content and reference point may also be defaulted in a template when many similar symbols are to be created. Values set in the template need not be entered for subsequent symbols. Scale bars, boxes, lines and north arrows may be created, and existing symbols may be joined together to form more complex ones. The Hershey fonts are drawn upon for symbol creation, and may be easily displayed. Symbols may be duplicated or deleted, and a reference grid may be called upon to guide placement of symbols. Many other utility functions control appropriate graphic display of selected features and symbols, including windows and masking by feature codes.



Creation and editing of symbology information

DEPROJCT (Deprojection: conversion to Latitude/Longitude)

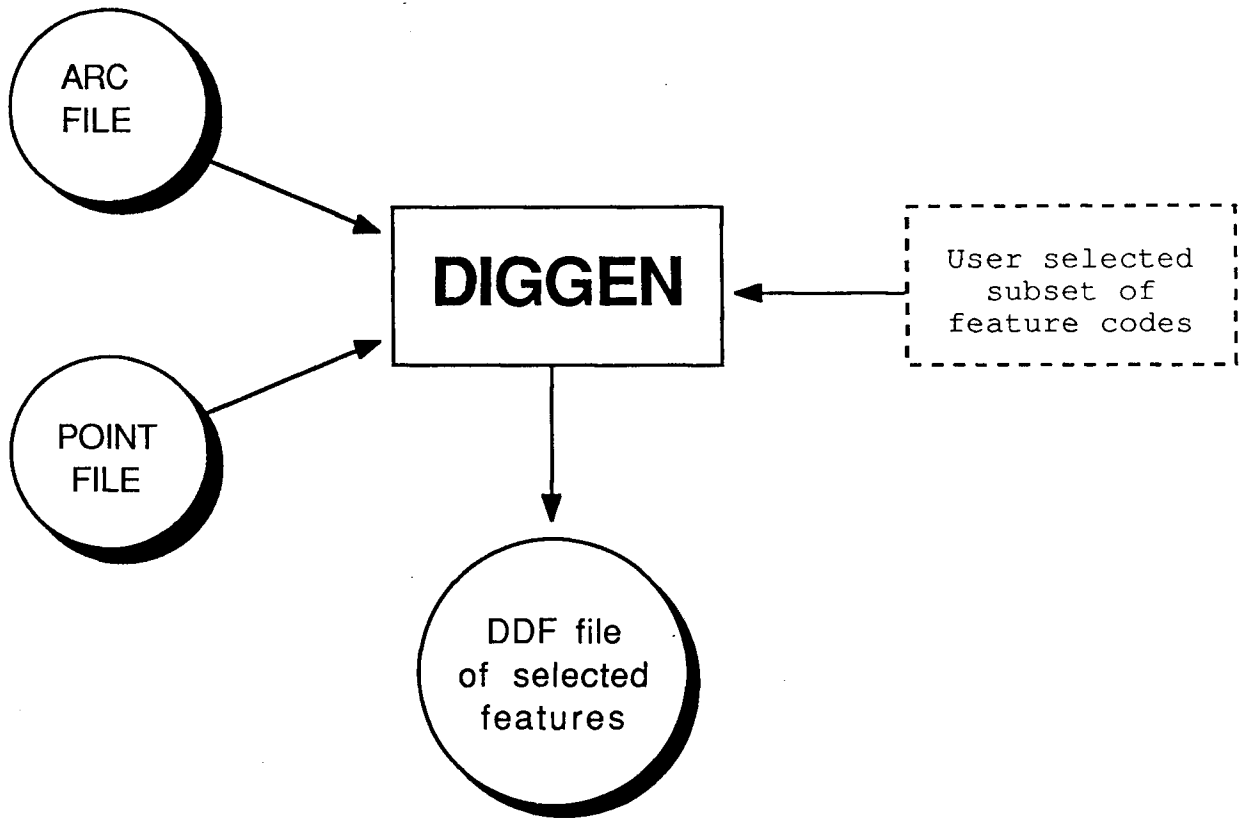
Projected map coordinates, in X-Y, are useful only with respect to other map data projected in exactly the same manner. Map data is projected from longitude-latitude coordinates to X-Y for the purpose of providing a realistic and controlled representation of the map area on a flat piece of paper (or graphics screen). DEPROJCT converts X-Y coordinates back to longitude and latitude, based on the correct description of the parameters of the original projection. Longitude and latitude are universal, so deprojected data can be transferred anywhere, and may then be reprojected to fit other projected map data. The X-Y data to be deprojected may reside in a file or may be directly entered at the terminal.



Conversion from projected (X,Y) locations to (longitude,latitude)

DIGGEN (Clean, Digitized Data Format Generation)

Map data stored in a GIMMAP cartographic database is kept in a form useful only to GIMMAP software. To transfer or save copies of this data or to get it in a form which is readable by the user, it must be converted into a GIMMAP DDF (Digitized Data Format) file. DIGGEN performs this function with selection of the subsets of features to be converted under control of the user - not all features will be converted as in ARCHIVE. A single DDF file is created by this process. If the database has already been graphically edited, then this will be reflected in the output file.

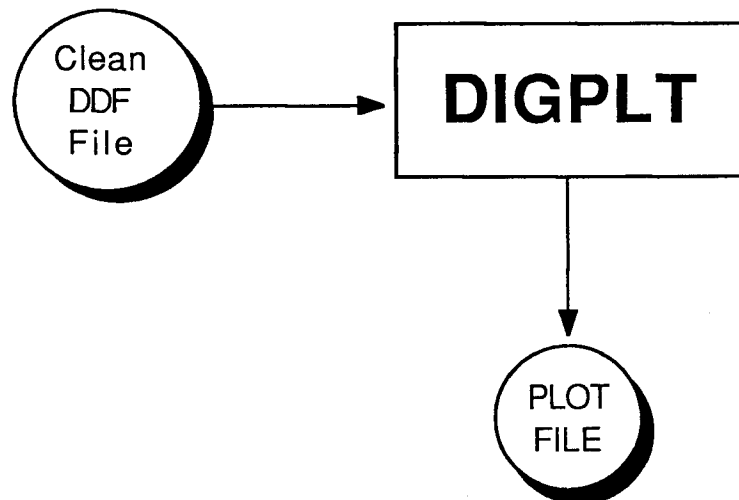


Creation of a DDF file of selected features from a GIMMAP database

DIGPLT

(Digitized Data Plot)

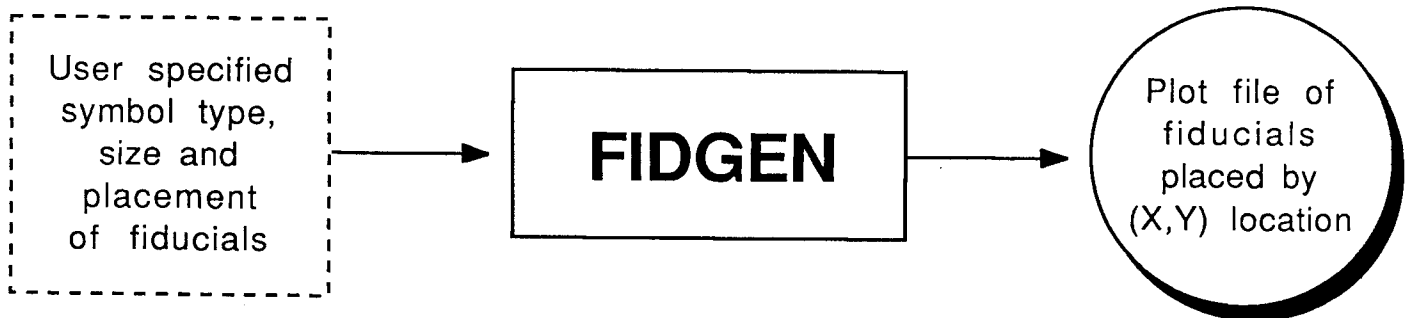
A digitized data format (DDF) file which has been created and cleaned by SYNEDT, or any other DDF file may be converted directly to plot form DIGPLT. Such a plot allows a quick check for gross set-up or procedural errors which may occur for digitizing. It also provides for a quick look at map data that has been archived or transferred. DIGPLT provides such a plot which, in both cases, may be generated directly, without the expense of creating the GIMMAP database from the DDF file.



Creation of PLOT file directly from a DDF file

FIDGEN**(Fiducial Generation)**

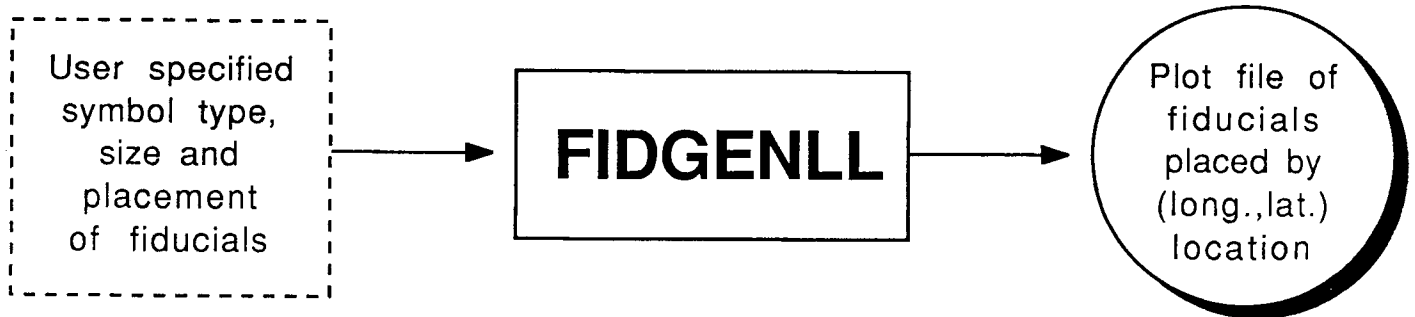
Registration marks (fiducials) serve many functions in the use of maps in automated cartography. Multiple plots may be properly overlaid when registration marks, placed at fixed X-Y or longitude-latitude locations, are aligned. Printers may register the many separate overlays by registration marks in the four-color printing process used for published maps. Finally, properly located registration marks may be used in the digitization process to accurately control locations for the entire map. FIDGEN offers creation of a plot file containing either a standard set of 4 fiducials placed at the corners of a specified X-Y or latitude-longitude rectangle, or any number of fiducials placed at locations individually specified by the user.



Creation of PLOT file for fiducials specified at (X,Y) locations

FIDGENLL (Fiducial Generation in Latitude-Longitude)

See FIDGEN above for uses of fiducials. FIDGENLL provides the necessary control for placing fiducials at exact points of longitude and latitude. All projections supported by GIMMAP are available for placement of fiducials. The parameters of the selected projection must be specified by the user, including the scale and the area of interest, indicated as a bounding rectangle in longitude and latitude. Both the fiducial size and symbol type may be selected by the user. Four fiducials are then placed at the corners of the bounding rectangle in plot format.



Creation of PLOT file for fiducials specified at (longitude,latitude) locations

GIMMAP.PROJECT

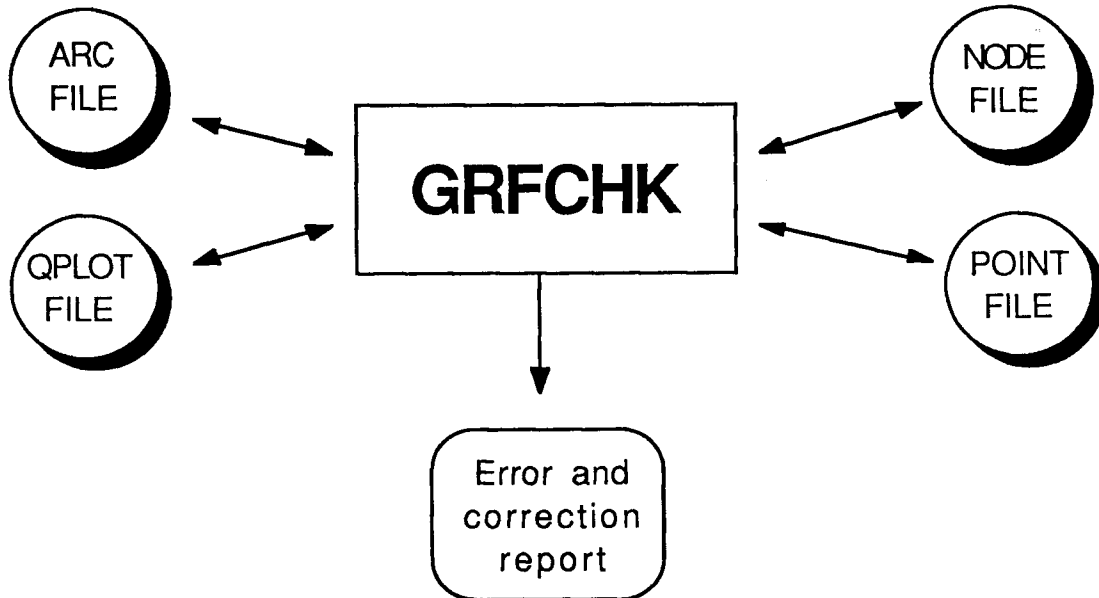
(Public Projection Library)

A projection and deprojection library exists for general use outside the framework of gimmap software. This library supports projection of points from geographic (longitude-latitude) coordinates into rectangular (X,Y) coordinates, and deprojection from (X,Y) back to geographic coordinates given the description of the original projection. This library is based directly on the PROMAN library used by GIMMAP programs, but with additional high-level functions built-ins. Routines exist to set up both input (deprojection) and output (projection) projection descriptions. Additional routines allow direct projection, deprojection and reprojection (deprojection using the input projection, followed by projection using the output projection).

GRFCHK

(Graphical Error Checking)

An unedited GIMMAP database may contain an array of errors arising from numerous sources. Of these errors, some may violate the GIMMAP view of map data and the many rules logically stemming from that view. GRFCHK analyzes the cartographic database to report errors in (1) feature locations and statistics, (2) the Bridging problem (ambiguous paths), (3) internal file structures and inter-file consistencies, (4) the node-arc graph for the map, (5) duplicated features, (6) map projection specifications and (7) zero-length lines. Many of these reported problems are automatically corrected by GRFCHK.



Analysis, correction and reporting of graphical and other errors in a GIMMAP database

GRFEDT**(Interactive Graphical Editing)**

From the source document to the process of digitizing to the execution of GIMMAP programs which create the cartographic database, numerous errors creep into the map data. Some are removed by the GRFCHK program, some by new digitization and others may not be discovered until the treatment of areas (zones) at a later stage. Many errors may be repaired by a process called interactive, graphical editing. In this process, an operator uses a graphics terminal to view the map data and to request and direct modifications to the displayed map data.

For this purpose, GRFEDT provides a set of modifying commands which change the contents of the cartographic database, and a second set of utility commands which do not modify the data, but which assist the operator to find and correct errors. The modifying commands include creation, moving, merging, deletion, insertion and splitting. Utility commands include view selection, feature masking (to select feature subsets), correction verification, input prompting and feature location. The final result of interactive graphical editing is a cartographic database which is ready for map production, zone creation (color maps), and data archival or transfer.

GRFMAN**(Graphics Management Library)**

The graphics library serves to support graphics operations on the devices for which the library is designed. GRFMAN supports graphics operations on the Tektronix 4014 vector terminal and the Xynetics 1100 flatbed plotter. Use of this library with other devices would require modification. The operations supported for the 4014 include direct line, point and special symbol and line type displays, and buffered graphics for a subset of these functions. Direct or buffered graphics allow for use of either user or screen coordinate reference systems. Buffered graphics increases speed for support of large data sets such as those encountered in cartographic applications. Support for the plotter includes initialization and termination routines, solid and dashed lines, and some point symbols. GRFMAN also provides for number posting and alphabetic symbology placement for both the graphics terminal at the plotter, using the Hershey alphabets.

HERSH (Conversion of Hershey sets from Felix to GIMMAP)

The Hershey alphabets, converted to an intermediate (FELIX) form by the A2FELIX program, must be further converted into a HERSHEY and an ALPTRS file for GIMMAP. HERSH takes the FELIX format and produces the alphabet pointers and Hershey coordinates files. These files are created to facilitate direct access to each character in each Hershey alphabet. The HERSH and A2FELIX programs are run only one time.

HERSHEXAM**(Examine Hershey Files)**

Certain programming or operational concerns may require occasional examination of the Hershey alphabet pointers (ALPTRS) and coordinates (HERSHEY) files. HERSHEXAM provides access to examine the sequence of raw coordinates which the Hershey alphabets contain for selected characters. Any character from any alphabet may be so displayed by selection of the alphabet number and entry of the desired character or keyboard equivalent.

HERSHPLOT (Hershey Alphabets Plot)

The twenty-three Hershey alphabets (fonts) are available in GIMMAP for the placement of symbology on maps. These sets include options in boldness (simplex, duplex, triplex) of numerous fonts, as well as a number of special, non-alphabetic, character sets. Selection of the appropriate alphabet is made possible only through visual examination of the available sets. HERSHPLOT provides a plot file of a compact chart of the Hershey alphabets, arranged by number for easy reference. This chart is essential in selecting symbols from the special character sets, since these characters do not appear on the keyboard. Instead, the keyboard equivalent (character on the keyboard which will select the desired special character) for these characters is directly evident from the chart.

KONTIKI (BOX)

(Display Oil and Gas Fields in Color)

The first of the color programs for GIMMAP, the KONTIKI and KONTIKIBOX programs are used to create a color display of the Oil and Gas fields in the state of Kansas on a Tektronix terminal. The fields may be selected by type (gas, oil, oil & gas, storage areas, shallow gas) and any combination of types is allowed. The BOX program allows the operator to select a rectangular sub-area of the state in latitude/longitude or projected x-y coordinates while the KONTIKI program always displays the entire state. Both provide legend information, a title, a date, and a north arrow.

LEO

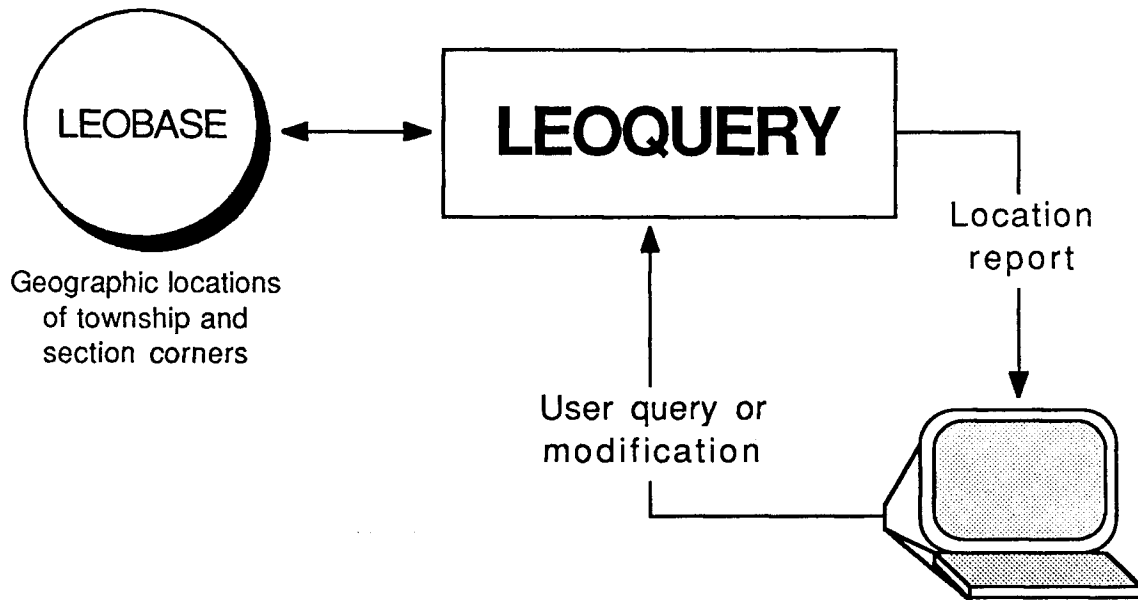
(Legal-Geographic Conversion)

Conversion between the legal and geographic reference systems in Kansas is essential to making graphic displays and maps of cartographic data, cross-referencing data in different systems, transferring data and locating secondary data. The process of conversion frees different data sets for many new applications. The LEO system, comprised of the FORTRAN77 subroutine LEOCVT and the database (LEOBASE) of spatially ordered township and section corners, performs conversion from either system to the other. The legal system used by LEO is based strictly on that for Kansas, and relies on the digitized township and section corners in Kansas, as represented on the USGS 7.5' quadrangle map series.

LEOQUERY

(Query the LEOBASE)

The LEOBASE database of township and section corners exists to support conversion between legal and geographic coordinates via the LEOCVT routine. The coordinates stored in LEOBASE are the longitude and latitude of the section and township corners of Kansas. LEOQUERY provides the means to examine these coordinates and if necessary to modify them. When a township and range designation is made, LEOQUERY displays the geographic coordinates of the four (if available) corners of the sections are displayed as longitude-latitude pairs. If modification has been selected, any of the four corners may be changed. Additional sections may be so examined, or a new township may be selected. When modifications are performed, duplicate copies of the changed point, existing in adjacent sections or townships will be changed as well.

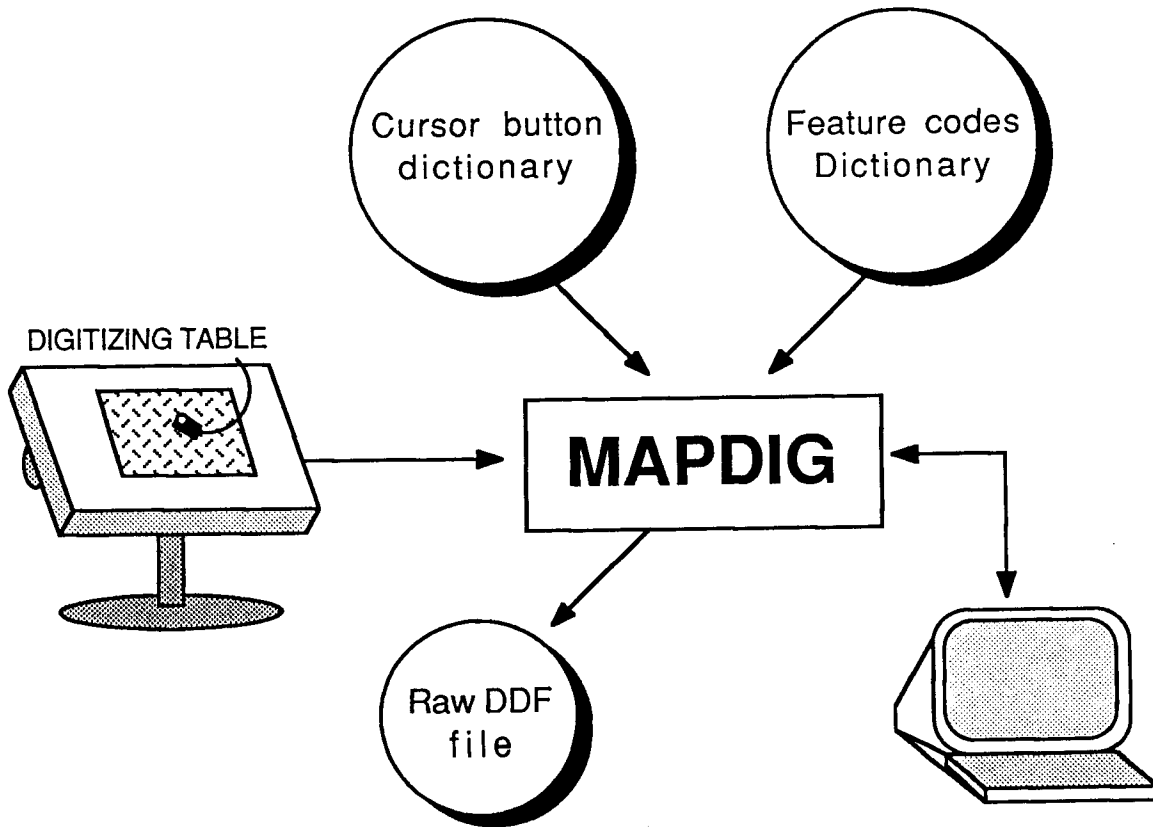


Direct examination and modification of the legal land-grid system database

MAPDIG

(Map Digitizing)

The first process in creating a GIMMAP cartographic database to support automated mapping operations is the conversion of map data in graphic form into digital form (i.e. usable by a digital computer) This process is called digitization, and is performed in numerous ways. The usual manner for GIMMAP digitizing is by means of an operator-guided pointing device (cursor) on a specially wired digitizing table, which converts selected points into computer-usable X-Y values, and transmits the points to the computer under the guidance of a digitizing program. MAPDIG guides the operator to digitize in the proper form and sequence, prompting for necessary information such as map title and projection information and feature type identification. The operator is prompted with a bell and questions on the terminal, and selects different buttons on the digitizing cursor to perform different functions.



Map digitization control program

MAPGEN

(Map Generation)

When one or more GIMMAP cartographic databases have been created and edited, the primary function of the system is to produce maps from them. This task translates into producing a set of plot files which contain features selected by type and by area, and are prepared graphically to produce the desired image on the map. MAPGEN collects map data from multiple databases, each with its own projection and collates the features into a single map by reprojection to the common output projection.

MAPGEN does not edit and splice the lines and edges, it merely reprojects to a common scale area and projection. If the pieces are appropriately digitized and edited, they will fit together properly. MAPGEN allows for multiple plot files (to provide for different colors, pen widths, etc.,) each of which will be sent multiple features selected from the input files. The selected output projection may be any which is supported by GIMMAP, including the standard projection for Kansas - requiring no further projection information. If USGS 7.5' quadrangle bases are listed as the input databases, MAPGEN will calculate the projection area directly from the list of Arc file names.

As an option, DDF files may be produced instead of plot files. This option is useful for archival and data transfer. It is also useful for creating new GIMMAP cartographic databases from collated areas. The new database may then serve to place and edit symbology properly for the map area. The DDF files may be produced in DDF format or may be created in geographic (longitude-latitude) coordinates for universal transfer.

Translation of data in X and in Y may be applied to plot or DDF output. Also applied to either is rectangular clipping in either X-Y or longitude-latitude to restrict the output map or DDF data + a windowed area. This clipping provides for neat borders for which exact (rectangular in X-Y or latitude-longitude boundaries may be produced. Clipping facilitates extremely good matching between adjacent sub-areas, and provides excellent control for digitization of features drawn on maps thus produced (with appropriate control markers). Large areas may be sub-divided for more efficient editing and placement of symbology. Also, separate data sets may be merged neatly by use of clipping during reprojection.

For plot files, both points and lines may be automatically annotated for enhanced effect. Nine general and six special (geosciences) point symbols may be selected, with symbol size and other parameters also selected by the user. Seven special line types with variable parameters may be used to enhance lines. Other options include a map frame; a GIMMAP label with projection, scale, date and time; and registration marks with longitude-latitude labels.

MAPREQUEST (Map Project Request)

A request for map services requires a large number of specifications of the map area, content, form and every other aspect. As implemented, the map request not only identifies the requestor and his/her needs, but also serves to guide and document the production process. MAPREQUEST prompts the user for many details of the map preparation process for the desired map. This generally forces the requestor to contemplate important aspects of the production which had not been considered before. The requestor finalizes the request after entering and editing for as many sessions as desired, often with the assistance of the production staff. The finalized request causes the generation of a report to all affected parties. The production staff then attempts to clarify any remaining questions, and proceeds to produce the cartographic product based on the report. Selections made by the production staff may also be recorded with the original request for future reference. All information may be displayed and printed at any time.

MAPTTL

(Map Titling)

The placement of symbology on a map may be done even after the plot files for the map already exist. Usually the CYMBAL program is used to perform this task on the GIMMAP database. MAPTTL is capable of producing symbology to fit the image in a selected plot file. The usual GIMMAP symbology files are built in this process, but some advantages of the traditional CYMBAL method are lost. The primary difference lies in the display of the map data as a reference for placement of the symbology. With a plot file, only one option exists - to plot everything in the file. This may be time consuming and may be difficult to view. Fortunately, windowing may be performed to enlarge small parts of the image for clearer view.

MAPTTL provides for creation, modification, duplication and deletion of labels whose content, Hershey alphabet, reference location, character height, angle of orientation and justification are specified by the user. Additionally, rectangles and scale bars may be generated. Plot output goes to a single file, but does allow for the inclusion of offsets for translation. The best use for MAPTTL is for small, uncomplicated plots created outside GIMMAP, which need simple symbology like titles, frames and scales.

NODGEN**(Node Generation)**

The third major step in the creation of a GIMMAP cartographic database is the recognition of unique "nodes," and generation of the node file containing information about the nodes. Basically, the nodes are the intersections of lines with each other, termination of lines in the interior of the map, and termination of lines at the edges of the map. For each node, it is important to maintain information about the location of the node, the lines connecting at the node, and the number of lines intersecting there. NODGEN inspects the endpoints of all arcs and "isolated" point features to generate a list of unique nodes. The information for each node is collected into a record, which is then stored in the node file. This information makes it easy to identify nodes connected to each other and the arcs by which they are connected. It is used for identification of map features for editing, automatic generation of zone boundaries by ZONGEN, and could be used for general network analysis operations. NODGEN averages the locations of nodes and updates arc endpoints to be consistent.

PARGEN

(Point and Arc File Generation)

The Digitized Data Format or DDF file contains the raw digital image of the map as captured by the operator in the digitizing session. From the DDF file(s), the cartographic database is built. PARGEN accepts syntactically-clean DDF file input and generates the Arc, Point and Q-lot files for the map. The DDF file and the digitizing syntax (rules of structure and format) tell PARGEN where arcs begin and end, what feature types are included, where nodes are indicated, what distortion of scale occurred in the input document, how to rotate and translate the data to fit the mathematical model for the projection and many other factors.

Each arc generates a unique record in the arc file. Interior points for each arc are segregated, linked together and connected to the associated arc record. A display file, parallel to the point file, is generated for quick plotting of the total map view. Statistics are reported on the number of arcs and points, file usage, and spatial extremes. PARGEN may be used to add data to an existing base or to overwrite (initialization)

PLTGEN

(Plot Generation)

Plot files may be produced from an existing GIMMAP cartographic database to create a working or publishable map. The plotted map may contain any desired subsets of map features, and may be used to compare the state of the database with the original source map. This **edit plot** may be created for plotting with separate colors for the chosen features. PLTGEN creates the plot with a map border and a title at the top of the map. Also plotted are fiducial or registration marks at the corners of the map area. State and county lines are annotated as long-short-short and long-short dashed lines. Node numbers may be plotted to assist in location of missing or unjoined nodes. Node degrees may be posted instead of node numbers to assist in locating node errors. Both sets of numbers are also helpful in locating errors during zone generation.

PROJCT**(Map Projection)**

Map features which are represented by points in geographic coordinates (longitude-latitude) may be converted to rectangular (X,Y) coordinates for creation of a GIMMAP cartographic database or for plotting. PROJCT accepts points from the terminal or a disk file and projects according to the specified projection parameters. The projections supported by the PROMAN library may be used, including the (GIMMAP) standard projection for the state of Kansas. The input points may optionally include Z-values which will be carried over to the output file. The projected output may be in the simple form used for input or may be converted to the Digitized Data Format (DDF) used in GIMMAP for database construction. For the DDF files, feature codes may be selected or optionally they may be generated automatically from Z-values.

PROMAN**(Projection Management)**

Various GIMMAP programs require the ability to perform projection and deprojection of points (conversion between geographic or longitude- latitude and rectangular X-Y coordinate systems). For this reason, a common set of routines to perform projection initialization, point projection and deprojection is collected into a single library, accessible to any program. PROMAN includes routines for set-up of the Modified Polyconic and Lambert Conformal Conic projections and for projection and deprojection in both. The calling program is responsible for setting parameters such as the scale and area of interest.

QSORT**(Quick Sort)**

Occasionally, the system manager may have need for the ability to sort a list of integers. QSORT performs a rapid sort of up to 10000 integers taken from an unordered file.

RAFMAN (Random Access File Management Library)

All random-access files in a GIMMAP cartographic database (may include Arc, Point, Node, Qplot, Zone, Border, Label, Symbol, Calligraphy) are structured according to set rules and managed by a library of routines. The structure includes an accounting record and a doubly-linked "freelist" chain of unused or reclaimed records. RAFMAN provides routines to initialize the structure, set file codes (channels or unit numbers) for the generic base files, get a record for use, return a record which will no longer be used, and report various errors found in these structures. RAFMAN is used by every GIMMAP program which accesses a cartographic database. The calling programs are responsible for such actions as overwriting the "flag" of each "gotten" (new) record and returning deleted records to the freelist.

RAFPRN (Random Access File Examine, Print, and Modify)

The direct-access, binary files of the database may be examined by use of the RAFPRN program. For each of the files in the database, there exists a file which reads the contents of the specified records of the desired file and prints them on the terminal screen for the operator to examine. The commands for printing are available on a menu listing; printing the file whose first letter is X may be performed by using the PX command (e.g. printing the arc file is performed by the PA command).

Selection of records is by a list of numbers or ranges which are separated by commas. The numbers represent single records and the ranges specify contiguous blocks of records from the first number to the second. All records so specified are printed with unused records displayed as the freelist record with a flag, a predecessor record and a successor record.

To be able to modify the content of a record, (available for most files), the operator uses the command which begins with M and ends with the letter of the file to be modified. When this option is selected, the operator must select a record to select and then selects one or more fields within the record to modify. The record content is displayed before any changes are made. The changes are made immediately as the operator selects them.

The RAFPRN program may be run in two modes, one to display the database content at the terminal or to a disk file.

SHADEZ

(Zone Shading)

Area shading may be a substitute or an enhancement to coloring of areas. In either case, it provides the necessary distinctions between the different types of areas in a map, when done properly. SHADEZ may be used to produce shade lines on a graphics terminal as well as in plot format. As many as 100 different zone colors may be extracted from a GIMMAP cartographic database. Each zone color may be represented by shade lines on a graphics terminal as well as in plot format. As many as 100 different zone colors may be extracted from a GIMMAP cartographic database.

Each zone color may be represented by shade lines of one or two separate angles. The selected zone colors may also be grouped into as many as twenty separate plot files, each of which may be plotted in a different color. The zone color, plot file and angle selections may be conveniently specified in a dictionary file, and need be typed only once. Plotting of zone boundaries is optional, and selection of zones may also be restricted by use of minimum/maximum zone area criteria.

SUMMIT**(Summary of Feature Codes)**

When an operator wishes to know in a gross fashion what groups of features have been included in a GIMMAP cartographic database, or in a group of databases, a summary of the quantity of each feature by type may be reported. SUMMIT will examine a single database or a collection of databases, count the numbers of features of all types, and report the counts to the operator. Such an evaluation may be made of an entire county to make certain that all required coverages have been made. Such counts will not guarantee complete or correct coverage, but only indicate the presence or absence of features in general.

SYMGEN**(Symbology Generation)**

Map symbology such as titles, feature names or numbers, scale bars and north arrows which have been created and edited with the CYMBAL program or the MAPTTL program may be transferred from one or more 'input' cartographic databases to a single 'output' database. SYMGEN performs a reprojection from the input bases to the output base, which may create a new set of files or simply append to existing records in the output database files. The reprojected information is transferred directly to the symbol and calligraphy files of the output base. The operation may result in the concatenation of multiple pieces into a single map, the splitting of an area into multiple sub-maps (through multiple runs of the program) or simply the re-scaling of a map, all in terms of the symbology existing in symbol and calligraphy files. Re-scaling of the symbology may be performed using three methods, selection of symbology may be controlled by feature codes, and optional outside margins guarantee no inadvertent loss of symbology when symbology has been placed outside the map edge.

SYMPLT**(Symbology Plotting)**

When an operator is not using the CYMBAL program or is stationed at a non-graphics terminal and wishes to create a plot file of selected symbology, such a plot file may be created directly and quickly. SYMPLT provides this direct production without a graphics terminal. The function of SYMPLT is the same as the PLOT command of the CYMBAL program. Symbols are selected by feature codes and sent to a selected plot file with optional translation in X and Y, and optional scaling. Options also include the addition of feature codes to a plot, and creation of multiple plot files in this manner. Some attempt is made to order the symbology in X and Y to reduce wasteful pen travel between symbols.

SYNEDT

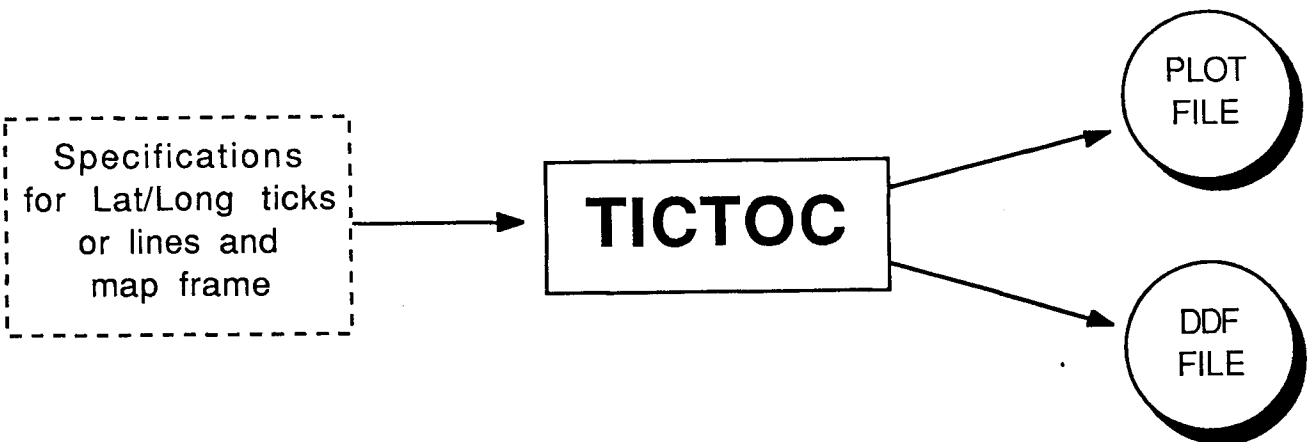
(Syntax Editing)

In order for the PARGEN program to properly build the Point and Arc files from the digitized data format (DDF) file, the data in that file must be error free. That is, the file must be in the correct form according to the rules which describe digitizing and the DDF file. These rules and the implied form of the file are called the **digitizing syntax** or **syntax**. The operator may violate the syntax when digitizing, and PARGEN is designed to expect only syntactically-correct input. SYNEDT receives the raw digitized data, finds and corrects errors in the syntax, and produces a syntactically-clean DDF file and a report of found errors. Corrections made automatically by SYNEDT are to conform to the syntax only, they may result in the creation of nonsensical graphical entities which must later be graphically edited via the GRFEDT program. SYNEDT provides optional translation in X and Y, and setting of default Z-values for isolated point features.

TICTOC

(Tick and Grid Creation in Lat/Long)

Lines of longitude and latitude or tick marks placed at points of longitude and latitude may be produced as plot information or in a DDF file for entry in a GIMMAP cartographic database. In the latter case, the lines/ticks are to be labelled using the CYMBAL program. TICTOC generates these lines/ticks beginning at specified longitudes and latitudes and ending at a second set of specified longitudes and latitudes. Lines/ticks are spaced according to specified separations. All choices are made in (decimal) degrees. Lines/ticks end at their outer edges at a neat line which may be rectangular in projected X-Y or in longitude and latitude. This border may be selected or may be determined automatically from the chosen projection and area. In fact, a projection boundary frame as the only output is one option. Fiducial marks of optional size may also be added.



Create DDF or PLOT files of latitude/longitude tick marks or lines to fit a specified map frame

X2C(T)

(Convert a Vector Line Plot to Color Plot)

Line plots made for the vector plotter may be segregated so that different line features are plotted in different colored pens on the vector plotter to produce color maps. Better results may be generated by converting the plot data for use on the color plotter. The X2C program performs this conversion of plot files which are in their original, binary, sequential form (x,y,pen) to create color plots of the lines. Each file may be converted to a specified color and line weight, and may be assigned a level to screen certain features by overlays. Excessive points may be weeded by the X2C program with a report generated to report the counts of points kept, weeded, and discarded as improper (the good, the bad and the ugly).

ZMARKS**(Zone Mark Reprojection)**

In GIMMAP, zones (polygons) may not be reprojected to new scales or to concatenate areas into new zones. Instead, this is accomplished by reprojecting the arcs and creating a new cartographic database on which zone generation and editing may be performed. However, at times it is desirable to put together several neighboring maps of zones to examine them as a unit and to compare the boundary match-ups and coloration across borders. Again the arcs may be reprojected directly into plot files to form the boundaries of the zones. ZMARKS may be used to reproject the zone marks from one or more input databases to the common projection for the output map. With the reprojected zone marks, the zone numbers or zone color numbers may be posted.

ZONEDT

(Zone Editing)

There are three basic editing functions which must be performed on zones, assuming that ZONGEN has successfully generated the complete set of zone and border records for the GIMMAP cartographic database. First, is the proper placement and signing of the zone mark for each zone. Zone marks, used to identify and label zones, are generated by ZONGEN at the center of mass of each zone. Each mark is created at a specified size, and may or may not lie within the zone boundary. Since one function of the zone mark is to mark the area to be peeled in the production stage, the entire mark must lie within the zone.

ZONEDT provides the MARK function in which zone marks may be quickly examined and accepted or modified as needed. The second editing function is to properly set the zone colors for all zones. Each unique color to be differentiated in the final product must have a unique color assigned to the zones of that type. Several different techniques are supplied by the COLOR function in ZONEDT to check, properly set and display these colors.

The third function to be performed in ZONEDT is to identify zones which are completely surrounded by other zones to the system. These island zones must be identified so that when the outer or surrounding zone is to be colored, the boundaries of its islands will also be drawn so they will not be colored. The ISLAND function in ZONEDT provides several techniques for assuring that island linkage is performed quickly and completely. Additional commands provide for selecting areas and subsets of zones for display and editing, posting of zone number or colors, deleting certain island zones, and converting between the CMY coloring system and unique zone colors.

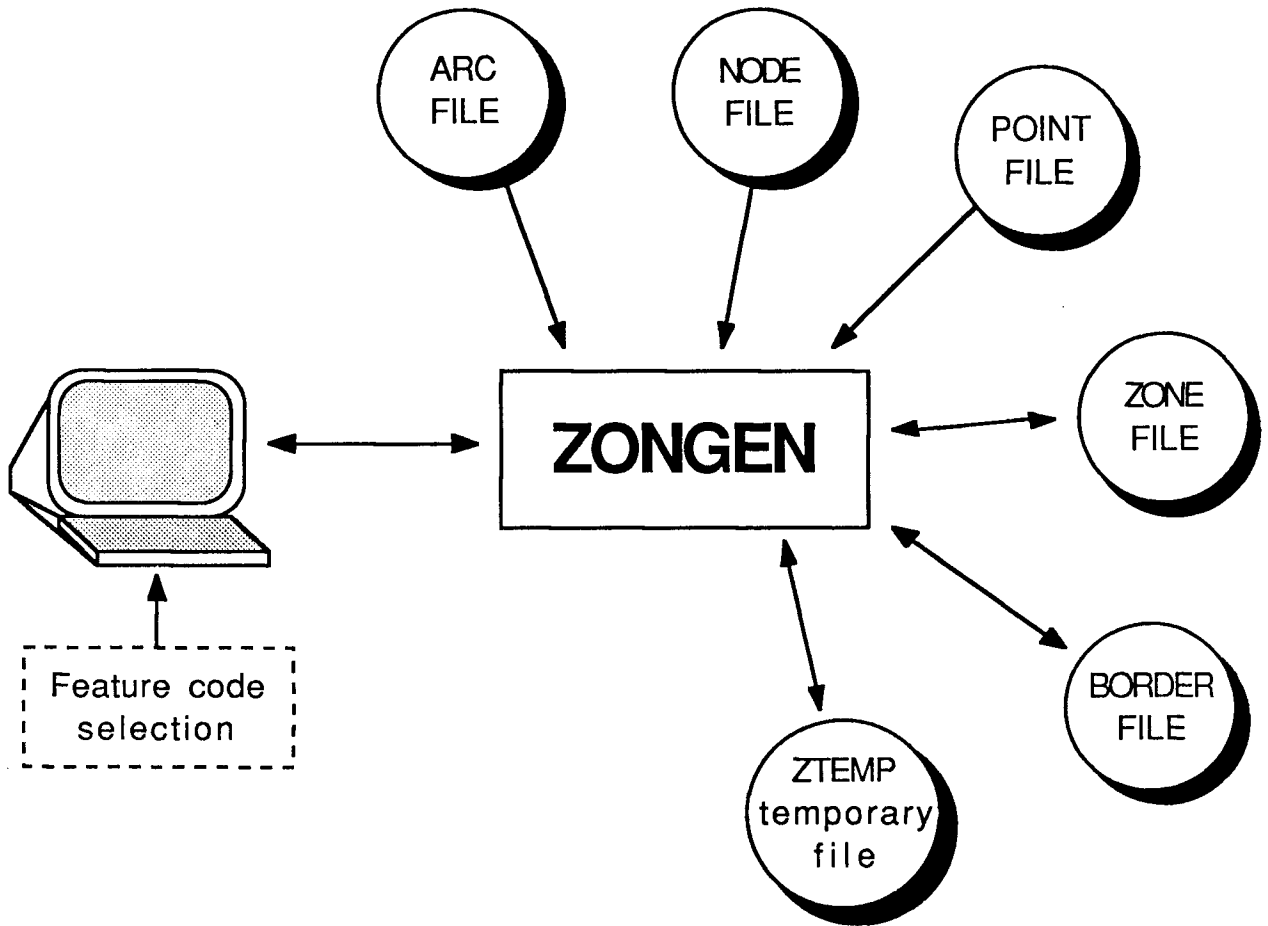
ZONGEN

(Zone Generation)

A zone boundary is the sequence of arcs which bounds the zone, separating the interior from the exterior. This boundary is ordered so that it follows as if the pen were drawing the boundary, necessitating the reversal (logically if not physically) of some arcs in the border. The arcs making up the boundary are called border arcs, and are referred to in the Border file. ZONGEN generates these unique boundaries, creating one record in the Zone file and one or more in the Border file for each zone.

Examining a selected subset of arcs, ZONGEN uses a left-turn algorithm to find each zone boundary (providing no graphical errors prevent completion). In this algorithm, the next node (intersection of 3 or more zone boundary arcs) on a list is chosen as the start node. The path from the start node to its next neighbor is taken. A left-turn is taken along the arc which is left-most to the incoming path, to the next node. This process is repeated until the start node is returned to.

The center-of-mass of each zone is calculated and stored as the zone mark location. The color assigned is a selected default or equal to the record number of the zone. Both island and non-island zone boundaries are generated. Area and perimeter statistics are also generated for each zone.



Creation of Zone and Border records from arcs with selected feature codes

- D R A F T -

I. INTRODUCTORY INFORMATION

I.7 LOGICAL AND PHYSICAL DATA SEGMENTATION

Using the principle of **data segmentation**, it is possible for automated cartography to process and produce maps of much greater size (in terms of the amount of data contained) than would be possible otherwise. Consequently, the GIMMAP system can be used to provide cartographic functions for an area the size of the state of Kansas by various forms of data segmentation, while such a proposition on the whole (all the state at once) is prohibited by limitations on the computing system (primary storage for data and program), possible limitations on physical devices (plotter and digitizer), time constraints on display and plotting, human limitations on comprehension of large sets of data, and others.

Data segmentation is basically the time-honored principle of **divide-and-conquer**. The use of data segmentation for automated cartography is one of dividing a very large set of map data into a number of smaller, disjoint data sets such that re-combining these small sets will produce the original large set. The small sets are designed to be of a size which is readily processed by the software and the operator, without exceeding any limitations or constraints discussed above.

This subdivision into parts, or segmentation, may also be referred to as a partitioning of the data set into a collection of disjoint or mutually exclusive subsets. This more formal definition, coming from the mathematics, is perhaps more rigid than is useful. For example, the restriction to disjoint or mutually exclusive subsets, meaning that no two subsets contain a common element, is not always adhered to in our use of the data segmentation principle. For example, it is possible to include a common border between two sub-areas in the map databases for both of the sub-areas. Several strategies for retrieval of map data and map production avoid the problems which might arise from such duplication.

There are two basic kinds of data segmentation employed by the GIMMAP system and in its implementation. The first of these is **physical data segmentation**, consisting largely of the physical subdivisions of data suggested in the preceding paragraphs. In physical data segmentation, the area of interest is divided into sub-regions which cover the original area. The map data for each of the sub-regions is digitized and a cartographic database is created for each area. Graphical editing, creation and editing of symbology and creation and coloring of areas may be performed and the pieces may be re-assembled into the original area (at an appropriate map scale).

All computing facilities and mapping systems are restricted at some point by limitations which in turn restrict the size of the map data set which can be processed. The use of segmentation extends these limits and allows maps of larger data sets to be processed. By the process of physical data segmentation, map data may be subdivided into smaller and smaller sub-areas to the point where very many sub-areas are combined to make maps which represent very large data sets.

The process of physical data segmentation requires the physical separation of map data into mutually exclusive subsets, but this separation may be performed on any logically consistent basis. The primary basis, listed above, is that of dividing the principle map into disjoint areas, usually of the same size. All data for a given sub-area is directed to the database for that sub-area only. Construction of the whole area is accomplished by placing the sub-areas (assuming a common scale and projection) in proper juxtaposition to one another.

Any other possible basis may be used to separate map data in a physical way. Given a single area with a great collection of various map features such that the map becomes dense or crowded with features, a simple form of physical data segmentation could be based on feature types. A separate database could be created for each type of feature or for groups of feature types based on their quantity or other aspect. The separate databases could be processed with less difficulty, and without exceeding limits or constraints. Overlays of separate features could be produced from the separate databases to construct the complete map.

Another kind of physical data segmentation is inherent in the way that the GIMMAP system structures and stores digital data in the cartographic database. The separation of the key graphic elements (arcs, nodes, points, zones,...) and the collection and division of the critical information for each element stored in a separate file of the database helps in many ways to extend the size of the maximum map data set.

The second, equally important, form of data segmentation is **logical data segmentation**. In this form of segmentation, there is no physical separation of data into sub-groups of smaller size than the original. Instead, sub-groups of data are defined by some form of logical segmentation such as using the value of some attribute of all the map features.

Precisely this type of logical data segmentation is used to distinguish features by their type, as defined by the feature code which all map features have. Using features codes, programs such as those for graphical, symbology, area and color editing can restrict display and processing to the subset of map features specified by the operator for logical segmentation.

One advantage of logical segmentation is that map features are not physically separated. Rather, all the sub-groups which

can be defined are all available to the operator at all times. Thus, the set of sub-groups defining the whole is not fixed as it is in physical segmentation. The operator may redefine subsets of map features dynamically, as they are desired. This has the effect of increasing the number of subsets available to all the possible sets, as compared to the fixed number of subsets that are available in physical segmentation.

The restriction of physical segmentation to a separation of the subsets of map data by physical means (in different files...) does not apply to logical segmentation. Thus, the technique of physical separation of map data into different databases by means of the feature types, described above as physical segmentation, may also be viewed as a form of logical segmentation which has been used to physically segregate features by type. This kind of segmentation separates map data into discernable **layers** which may be drawn one upon the other to reconstruct the original map, or which may be used separately to create a useful map product.

The limitation, loosely referred to as a "maximum map size" in automated cartography does not refer to a maximum area on the surface of the earth which can be mapped. No, with a change of map scale, maps all become the same size and fit either on your digitizer or on your plotter. Assuming the software is able to handle it, all the world can be made to fit on the devices used in the automated cartography shop. Of course, the display of all this information may overly clutter the editing screen if it is not done properly.

The maximum map size is not a reference to the digitizing or plotting hardware, for exactly the same reason. Change of scale can allow any area to fit on these devices. However, a better solution to this problem lies in the data segmentation techniques discussed above. Source documents for digitizing and entry to the database come in the scales that are available. They must be used at those scale, and application of data segmentation will help with what results from that original data.

The most probable limitation on the maximum map size arises from the limitations of hardware and software in the automated cartography system and its environment. With a virtual memory system, there are still limits which are those reached by the operator in the accurate perception of map data displayed on the screen or drawn on the paper. But with any system, the careful use of physical and logical data segmentation will increase the practical size of a set of map data which can be processed and from which maps may be produced.

- D R A F T -

I. I N T R O D U C T O R Y I N F O R M A T I O N

I.8 Additional Processes and Sequences

As described above, the basic sequence of programs is used to capture map data, create and maintain a digital cartographic database and then to produce plot files from the database which will effectively reproduce the original map. Other possible map operations and functions arise from the creation of the database as described above.

This sequence accounts for the most important subset of programs comprising the GIMMAP system, but less than half of the programs in GIMMAP are a part of the basic sequence. The rest exist for other processes and sequences not included in the basic sequence to support reproduction of a single document map. Many other functions exist based on the assumption that the digital cartographic database exists.

The functions described here are divided into groups of programs according to function or to some aspect of function. The groupings include one for programs which reproject map data taken from one or more databases, programs which produce plot files of map annotation, additional programs for symbolization, programs producing line and area plots for the color plotter, utility programs for assorted functions, programs and libraries for map projection and deprojection, programs and libraries for conversion between legal and geographic reference systems, programs to set up and examine the Hershey annotation database, programs for sorting data, and special programs to display the Oil and Gas field data of Kansas.

1. Map Production - Reprojection

The single most important function beyond the basic sequence is the ability to join multiple maps together through the process called **quilting**. In this process, a collection of maps covering adjoining areas may be brought together to form a single map database or to be plotted together as a single coherent map.

In order to accomplish this goal, the point locations representing the features from all the maps are brought together into a common reference system (geographic) through **deprojection**, and then they are **projection** in a unified manner to create either a new database or to plot a quilted map. Together, deprojection followed by projection is referred to as **reprojection** of the map data, a process common to the programs in the first group.

The program used to perform quilting of multiple areas into a single area is called **MAPGEN**, for map generation. (Use of the MAPGEN program for basic sequence functions has been described above in Processes: The Basic Sequence For GIMMAP). As suggested

above, there are two basic functions or modes of operation with the MAPGEN program. One is the creation of plot files to create a map based on the quilted areas. The second is to create a new database representing the quilted areas.

For either operation, the operator selects the databases to be quilted, usually by naming a file which contains the names of the arc files for all the databases. The operator also selects the subsets of map features which are to be quilted for either operation by specifying the groups of feature codes which are to be selected. For plot output, each set of groups is assigned to a separate plot file. For database creation, features may be assigned to separate DDF files.

The separation of features for plot files is very important for the creation of the final plotted map. Features should be sent to separate files to accomplish two goals. First, for lines or point symbols to have unique colors or line widths on a vector or a color plotter they must be segregated at this time. Map features separated into different plot files can then be plotted on a vector plotter in any selected color with various pen widths as selected on demand by the operator. Color plots may also be produced from the same files, with similar options.

Second, if only one map product is to be produced then no further feature separation is necessary. However, if two or more different map products are to be made, even if at a future time, then features may be separated in such a way as to provide for the exclusion of any unwanted features or groups of features to produce the desired maps. Similarly, features which are to be represented in different ways (via color etc.) for different products must be separated into different files.

The ability to reproject map data is useful to two other programs outside the basic sequence of GIMMAP processing. These two programs allow the reprojection of map data which can not be done by the MAPGEN program. The two types of map data which can be reprojected as a result of these programs are symbology data (as that produced by the CYMBAL program) and map zone marks with posting of zone numbers or colors.

The quilting of map data from multiple map databases to form a new database of larger scope (covering a larger area) requires the creation of new symbology to fit the new map. If useful (for the quilted map) symbology exists for the source databases, then it may be possible to transfer some symbology from the source databases to the new database.

This transfer is not always possible due to the usual change in map scale between the source maps and the quilted map. Change of scale may make the height of the symbology inappropriate on the new map. Height will be altered in the transfer, but it can only be set to a constant size or scaled by a constant factor for all symbols being transferred.

Scale change and reprojection may cause the symbology to be translated to undesirable locations. These new locations may be too distant from the object which the symbology labels or names. The new locations may also cause the symbology to lie on top of map features or other symbology or to lose symmetry or centering with respect to map features. The inclusion of additional map features from other sources will exacerbate this problem.

Other problems exist for the reprojection of symbology for quilted maps. For example, along the seams of the joined areas, symbology may be duplicated in the databases on the two sides of the boundaries. Other examples can be generated, but it must be noted that most, if not all, of these problems can be corrected in the quilted database by use of the CYMBAL program. If many symbols exist and can be reprojected by the SYMGEN program, then many of them might be close and of value in the quilted database.

The character height of the reprojected symbols in the new database can be easily modified to fit the new scale. Locations and orientations may also be changed easily. All these things can probably be done much faster by using the CYMBAL program for all the reprojected symbols than to create an entire set of new symbology from the beginning. Symbols which can not be corrected to fit the new map may be deleted easily.

Thus, the SYMGEN program may be used to reproject symbology from the databases of adjoining areas to be joined into a single database for the quilted area. Symbols to be reprojected may be selected by feature codes. Reprojection is performed only on the reference point. All points in the symbol are then translated in parallel to the reference point. The output symbols are created as Symbol and Calligraphy records in parallel to the records in the original files.

The character height of each symbol is adjusted according to the selection of the operator for (1) height set by a fixed scale factor applied to the original height, (2) height scaled by the ratio of the map scale factors (source/quilted), and (3) a fixed height for all symbols. These options are provided with very little experience in their use, in the hope that each will have a valid use and together, they will provide for many different applications.

The third use of reprojection outside the basic sequence is the reprojection of zone marks from multiple source databases to fit a common (quilted) database. The program which performs this task is the ZMARKS, for zone marks, program. The reprojection of zone marks is a facility which complements the reprojection of the zone boundaries by the MAPGEN program. It allows maps of areas to be processed for smaller, adjoining areas before they are joined together. This is the principle of segmentation which increases efficiency and effectiveness.

The reprojection of zone marks allows the creation of a map of quilted zones along with their zone marks which are annotated with either the zone marks or the zone colors. The result is a map plot which may be used to compare the boundaries of the areal features being represented (e.g. geologic formations) across (source) map boundaries and colors assigned to the areas as the join across these boundaries.

2. Map Production - Supplemental Annotation

There are three functions which fit into category of adding annotation to a map product in the form of a plot file. Two of the three forms of annotation are aesthetic and remain as a part of the map product. The third is used in the production process when a map is to be printed, when overlays must be registered, or when control point locations (for digitizing data which is to be added to the map by hand) must be marked. Registration marks do not remain with the map unless they are to be control points for later digitizing.

The first program in this group is the **BOXER** program, used to generate plot files of multiple rectangles (boxes) which are defined by the operator. The extremes (minima and maxima) of the rectangles are requested by the **BOXER** program, which places the plot commands in a specified file. The operator may select as many boxes as are desired for the map, and may place rectangles in separate files where different colors or pen types and sizes are needed.

The creation of boxes at operator-selected locations and sizes fulfills various needs in the creation of map products. The most common use of such a box is the map frame, a bold line surrounding all parts of the map, the map legend and any other items included as part of the map. The map frame may also be drawn as a double line or double rectangle, created with two boxes of nearly the same size. (It should be noted that the **CYMBAL** program also provides a command specifically for creating the map frame with double line as an option).

Many other uses exist for creating boxes to compliment the map product. (Some of the uses listed here may be provided by other parts of the **GIMMAP** system). The shaded or colored areas on the map will require similarly shaded or colored boxes in the legend at which text will be placed to define or explain the feature which is represented by each. Some special text areas may be highlighted or separated from features by enclosing the text in a box. Inset maps may be set inside a box, and the map area may be separated from the legend, map title and other map features by a "neat-line" box at which all map lines end.

The second function in the annotation group is the creation of **registration marks** which represent locations in the geographic reference system (latitude and longitude). And, given that these

(four) points also represent fixed locations in the digitizing reference system when the map is to be digitized, these four points represent a transformation system which allows for proper conversion from digitizer coordinates into a system of correct projected coordinates based on knowledge of the map's projection parameters.

Registration marks are also known as **fiducial marks**, and are used to properly align multiple overlays in the printing process, to create printing plates for four-color lithography. To serve only for the purpose of registering multiple overlays together for printing of composite images, fiducial marks could be placed anywhere outside the map image, including legend, title, frame and other materials. For this purpose, the locations of fiducial marks in either reference system would be unimportant.

It is only when digitizing of new data placed on the map is planned for the future that the fiducial marks must identify known locations in geographic reference. This information will later support the conversion of digitized locations to geographic coordinates, also allowing for reprojection for other functions.

Two programs support the creation of plot files of fiducial marks (in GIMMAP the fiducial mark is made by drawing a square, a plus, and an X, all centered on the selected point). The **FIDGEN**, for fiducial generation, program creates a set of fiducial marks at locations selected by the GIMMAP operator using projected (x,y) coordinates and places the results in a plot file for the vector plotter. Using **FIDGEN**, the operator can produce fiducial marks which are useful only for combining overlays, but are not useful for creating marks of known geographic location to be used for digitizing added data.

The **FIDGENLL**, for fiducial generation in latitude-longitude, program is used to create fiducial marks at known geographic locations, selected by the operator. Using the **FIDGENLL** program, fiducial marks may be properly located at the corners of the map projection, as is expected when they are to be used as control points in the digitizing process.

As with the **FIDGEN** program, **FIDGENLL** produces a set of four fiducials in response to the operator specifying the extremes of a rectangle in projected cartesian or geographic coordinates. The **FIDGEN** program provides the option of specifying locations for any number of fiducial marks, and **FIDGENLL** provides optional symbols (plus, square, star, or fiducial) to be placed at the selected four locations.

The final annotation function in this group is that of the creation of a grid of latitude and longitude lines or tick marks (short lines at the outer edge of the map sometimes bounded by a neatline rectangle) in the correct geographic locations according to the projection information selected by the operator.

The presence of a latitude and longitude grid provides the map reader with the ability to locate map features in geographic coordinates which are universally used and understood. The grid of lines or ticks is created by the TICTOC program in the GIMMAP system. Using the TICTOC program, the operator may create either a plot file or a DDF (digitized data format) file (for addition to the map database) containing the ticks or lines selected by the operator.

The plot form is used when direct annotation is to be added to a set of existing plot files for a map. The DDF form is used to add the ticks or lines to a map database. One primary reason for doing this is to provide the opportunity for the operator to add annotation to the ticks or lines in the form of labels to number the latitudes and longitudes. Addition of the ticks or lines to a database (any one will do) and labeling them with the CYMBAL program is the only way to properly annotate the lines.

Whether ticks or lines are selected, a bounding rectangle specified in either projected (x,y) or geographic (latitude and longitude) coordinates is used to determine where the lines or ticks end at the outside edge of the map. If lines are chosen, they will completely traverse the map to the opposite edge. If ticks are chosen, the operator selects the length of the ticks as measured from the bounding rectangle.

The operator may select the bounding rectangle from one of five options depending on the choice of the neatline for the map and other aesthetic options selected for the map. The options include specification by the operator in either the projected or geographic coordinate systems, the boundaries implied by the map projection in projected or geographic coordinates, and finally the geographic boundary implied by the tick and line selections.

The selection of tick or line locations is restricted to a regular spacing of lines or ticks from the starting and ending latitude and longitude locations specified in decimal degrees by the operator. Thus, all lines or ticks in the grid are equally spaced in latitude and longitude by distances set by the GIMMAP operator.

The operator may select the map projection type and other parameters defining the mathematical projection. Using this information, the operator may select an option in the TICTOC program which produces a special grid of latitude and longitude lines. The operator may select to create only those lines which lie at the edges of the map projection area, forming a neatline frame around the map area.

3. Map Production - Symbology

The third group of functions is related to the creation, editing and plot creation of map symbology information. The

CYMBAL program provides for these functions in basic ways in the basic sequence, but there are other ways in which these functions are needed in different environments, and they are provided by other programs outside the basic sequence. In the case of one program, new functions are provided by applying existing CYMBAL functions on entire classes of symbols.

The use of the CYMBAL program requires access to a vector graphics display terminal. However, if the operator is only using the program to create a plot file of existing, correct map symbology then there should be no need to use the vector graphics terminal, which may be in demand or may produce unwanted displays which slow the operation.

Selection of map symbology from the database for creation of a plot may be accomplished using the SYMPLT, for symbology plotting, program. This program allows the operator to produce the symbology plot using any terminal whether it is a graphics display terminal or not. It provides the same selection options for the creation of plots as does the CYMBAL program, but without the graphics displays.

The SYMPLT program allows the operator to create multiple plots of symbols with different feature codes in each file. The operator selects the single feature code to be retrieved and then names the plot file where all symbology for the code is converted to the plotting format for the vector plotter. When all of the symbology for that code is processed, the program cycles to allow the operator to select additional symbology.

The second function (outside the basic sequence) involving map symbology meets the need to create map symbology for pre-existing plot files of map data. This need parallels that of the original functions of the CYMBAL program. For that program, the map database must exist to support and compliment the symbology files, while here the plot files must exist for the same purpose.

The display of the supporting map data is different in the two cases. For the CYMBAL program the display is created by taking map data from the map database and converting to a display form for the vector display terminal, exactly as is done for the graphical editing program GRFEDT. For this operation, the plot file is interpreted and converted to display form with no concern for feature codes (as can be used in CYMBAL for screening data).

But the goal of attaching symbology at correct locations is common to both the CYMBAL program in the basic sequence and the MAPTTL, for map titling, program used to add symbology to match an existing plot file of map data. In this process, the map data is displayed directly from the plot file and the operator creates and places symbology in proper locations just as was done in the CYMBAL program of the basic sequence.

The functions available in the MAPTTL program cover those basic commands available in the CYMBAL program. In the initial phase of operation, the MAPTTL program, like CYMBAL, offers the option of file initialization (available only at the start) and then executes recognized commands from a menu of symbol modifying and utility commands.

The modifying commands are a subset of those used in CYMBAL, paralleling the primary functions. Symbols are created with the specification of Hershey alphabet (font), reference point, angle of orientation, justification (left, right, or center), feature code, and content. All specifications may be changed and symbols may be deleted by pointing to the symbol and giving the correct information to the desired command.

Rectangles may be generated by selecting the corner points (extremes) in various ways, including selection by pointing and selection by entering coordinates. The rectangles are used for many things (neatline, legend or text boxes, etc.) as described above. Map **scale bars** may be created by menu command and placed anywhere on the plot. Labeling of the scale bars is separate from the command that creates them, and, unlike the command for creating scale bars in the CYMBAL program, the scale information can not be obtained automatically by the program.

As in the CYMBAL program, symbols are all associated with feature codes which can be used to assist in the display of the symbols as well as for the selection of symbols to create plot files. With the MAPTTL program, the display of created symbols is strictly under the control of the operator, with a specific command required to cause the display of symbology (unlike the CYMBAL program).

Additional utility functions with the MAPTTL program include a command to display entire sets of Hershey alphabets, creation and elimination of a default set of symbol specifications which define a **template** for symbols, and the usual set of windowing and display functions common to GIMMAP programs for graphical feature and symbology editing.

Map symbology created and edited with the MAPTTL program can be extracted for plotting using a command in the menu. Symbols are selected by single feature codes, converted to plot format for the vector plotter, and sent to the plot file selected by the operator. The MAPTTL program then cycles to allow for additional selections to be sent to other plot files.

The last function in the symbology section is the **CLASSY** program in which two important modification operations on symbols are applied to classes of map symbols by feature code, rather than to a single symbol at a time as the CYMBAL program does. The application of symbol modification functions to whole groups of symbols saves the selection and verification process and the redundant entry of specification information.

The two modification functions offered by the CLASSY program are the modification of the height of a class of symbols and of the font (Hershey alphabet) for a class of symbols. In the use of this program, the operator selects the option of height or font, and then must specify the new value for that which is to be changed.

In order to execute the requested changes, the operator must also select the range of feature codes that specify which symbols are to be modified. Only the symbols whose feature codes lie in the specified range are selected and modified according to the operator selections. After each class modification is completed, the program cycles to offer the operator new selections for as many modifications as are desired.

The modifications made to the groups of map symbology by the CLASSY program are made directly to the Symbol and Calligraphy files of the database. Any changes made are made immediately and can not be undone as such. Of course, any change in height or in font can be changed back by further operations using the CLASSY program.

4. Map Production - Color and Shading

The addition of a color plotter to an automated cartography system provides a rich outlet for new map products. Although the current set of color plotters operates in raster mode, they are all designed to accept vector input to define areal boundaries. They generally require minimal conversion of line plots made for vector plotters to the pre-raster form which the color plotters will convert into colored lines of varying width.

Given the existing GIMMAP programs, database structures, and philosophy of operations, a set of programs has been developed to draw on the GIMMAP databases and software to create color plots of linework and of areas. Additionally, one program provides for color display of areas created in ZONGEN and edited by ZONEDT. A final program in this area provides the ability to **shade** areas in a map database with parallel, straight lines (solid or dashed) at angles set by the operator. This capability provides a method of distinguishing areas by their types with or without color output.

Two basic abilities must be developed to support mapping by color in the GIMMAP system. First, the linework which is plotted on vector plotters and may have colors assigned at plot time by changing pen colors must be translated into an equivalent form on the color plotter. Second, the areas generated and edited in the database for separation onto scribecoat material for printing or shaded, perhaps with colored pens, on the vector plotter must be sent to produce solid-color areas on the color plotter.

The first function, plotting lines on the color plotter, is supported by the X2C program which takes a raw vector plot file

as input and produces a plot file for the color plotter which sets all the lines to a single color and line width as selected by the operator. Using this program, the lines that are to be used in the color plot of the map are assigned to appropriate colors and widths, and each type is sent to its own plot file for the vector plotter. This separation would probably have occurred for the vector plotter as well unless the output was restricted to blackline for various reasons.

The different line types will require one additional piece of information for most color plotters. Beyond the color and the width, each line must be assigned to a layer or level relative to other data in the map. This level is one of (possibly) many, and is used to determine what is plotted at places where more than one feature or object exists. For example, if a line exists at the same place as a background fill color for an area then which is plotted? Options usually include the line, the area, or both. In some systems, this technique can be used to blank (white) out an area.

The use of levels is usually to set levels of area fills of solid color at a level which is replaced by the level used for the linework. Where lines intersect, both may be displayed if they are at the same level. Background area around the map may be filled with a low level so that all the map features replace it, and background area within the map area may have a level set to provide a separate color which is replaced by everything which lies in the map area.

The X2C program offers a further function which may reduce the time required for plotting on the color plotter significantly when lines are defined with overly dense points or when points become dense as a result of scale change. The weeding of points which lie within an operator-selected distance of one another may reduce the number of points defining a line by as much as fifty percent or more. The X2C program counts and reports the numbers of points kept, weeded, or unrecognized (the Good, the Bad, and the Ugly).

The X2C program converts raw plot files from MAPGEN, PLTGEN, CYMBAL and related programs into plot files ready to transmit to the color plotter in an online form. The companion program X2CT performs the same function, but creates output which is designed to be transmitted to the color plotter via magnetic tape in the offline form.

The second function listed above for the basic abilities in color mapping is the ability to extract the boundaries of areas and send them in a form to be plotted as areas of solid-color fill on the color plotter. This function is supplied in GIMMAP by the CALPLOT program. The CALPLOT program examines the zones in the Zone file to find those which are selected. The Border and Point files are then used to produce the sequences of points which define the boundaries of the zones.

Zones are selected for production of the color plot based on choices made by the operator. The operator must select zones in the database on the basis of the color or primary attribute value of the zones. For each zone color selected, the operator must also define the plot color which is to be applied to the areas of that zone color. The plot color definitions consist of primary color (cyan, magenta, yellow) mixtures taken from color charts or arrived at by experience with the color plotter.

To save the operator repetitious typing and trouble when the set of plot colors is repeated, CALPLOT is set up to allow the operator to make selections from a file called a dictionary. The color dictionary entries each contain the zone color, the plot color and the primary mixture values to define the plot color. If the operator selects a dictionary, the entries are read from the dictionary and used instead of requesting values from the operator. Only zone colors appearing in the dictionary will be selected and plotted.

The entries in the dictionary file define the plot colors and assign them to zone colors. The definitions of the plot colors may occur in one of two forms. The first is the use of percentages of the primary colors which comprise the color. The second form is to use the default colors which are available in the palette of the color plotter. For this form, only a single number is required to select the desired color.

Normally, the zone colors in the database are mapped through the dictionary or operator definitions to plot colors and then the operator selects which subset is to be plotted. There are cases in which the zone colors undergo an intermediate mapping to a subset of possible plot colors, before assignment to defined colors is completed. This process uses a Translation file which maps zone colors to plot colors.

One such example would be the mapping of counties, each with a unique color number, into the plot colors which represent some production statistics on a county basis. The Translation file allows you to make the desired map without editing the database to have the proper zone colors assigned to each (county) zone. This approach allows the creation of multiple Translation files for multiple maps made from the same database without performing unnecessary zone editing.

When plot colors have been defined and zone colors assigned, the operator selects the subset of zones which are to be plotted. The selections are made on the basis of the zone colors (after they may have been modified by translation), and all qualified zones are extracted from the Border and Point files and produced in the color plotter format for the appropriate output file.

The CALPLOT program requires the operator to specify the level (described above) at which the non-island zones are to be

plotted. This "starting" level is the lowest level at which zones will be plotted. Islands within these zones will have the next level, and islands within the first islands will have the next, and so on. This guarantees that selected islands will be visible on the final map, and they will replace the appropriate parts of the surrounding zones rather than having both appear.

The operator also has the option of selecting a background color for the map. This is accomplished by selecting a rectangle (via the extremes) to be filled with color, the level at which the background rectangle is to be plotted (this should be lower than all the map features), and a color with which the rectangle is to be filled. The color selection may use a default color of the color plotter or may be made as a mixture of primary colors.

With the color plotter, the boundary lines of the areas to be filled with color may also be drawn by the color plotter. If this is true and the lines are not desired, they may be made to be invisible by the CALPLOT program. To do this, CALPLOT must be given the number of a pen with which to draw the invisible lines. This usually means that the line type normally associated with the chosen pen number will not be otherwise available.

Like other programs dealing with the color plotter, CALPLOT has two forms. The CALPLOT program produces output to be sent directly to the color plotter from the host computer. This is the online version of the program. A second form of the program is CALPLOT, which generates plotter output for transfer to the host computer via a magnetic tape. This form is referred to as the offline version.

The X2C and CALPLOT programs provide the basic forms of converting linear and areal map data for plotting on the color plotter. A third program, COLORZ, is used to display both kinds of map data on the color display terminal. To some degree, this program parallels the approach taken by the CALPLOT program in that both select zones by color, translating colors once or twice into plot colors, extract zone boundaries from the database and convert coordinates to a display/plot form.

The COLORZ program also provides the optional Translation file to convert zone colors from the database into a subset of plot or display colors. Translation files are used to allow one database to be displayed as many different maps without editing to change the zone colors for each map. Translation files may be created with much less work than editing to change zone colors.

Like CALPLOT, the COLORZ program allows the optional use of a dictionary file to define the colors to be used in the display. The dictionary may be created within the program, new entries may be added to an existing dictionary or an existing dictionary may just be used by COLORZ. Entries in the dictionary include a zone color mapped to a display color number and the definition of that display color as a mixture of primary colors.

Other functions are provided in COLORZ to enhance the view of the map data. The operator may select whether the background area around the map area is black or white. The choice of color for the background may be dependent on the intended use of the display. If hardcopies are to be made, the background color will be set white to save ink on the paper. If a photograph is to be taken, the background is black. Other such considerations may decide the background color selection.

The area to be viewed in the display is initially set to be the maximum view needed to include only the complete map area as defined in the database for viewing with other GIMMAP programs. The boundaries of this area may be reset by the operator in the setup portion of the COLORZ program to alter the scope of the map area to be displayed. This option provides a program zoom on the map to magnify the image to the largest possible size prior to hardcopying, which is useful when segments are not selected and zooming by the terminal is not possible. However, over-zooming to less than the map image may result in undesirable effects.

The boundary lines which define the borders of the selected zones may or may not be displayed in the final map display. This option is provided to allow the operator to decide which form of the map is most pleasing and most effective in presenting the map information to the viewer. The boundary lines are drawn in both cases and is simply covered by the area fill in the option with no lines. Thus, no time is saved by selecting no lines over the lines included option.

Similarly, the operator has the option of selecting a subset of the base data (non-zone information) in the map database for addition to the map display. The inclusion of the base data will increase recognition of locations of the zone data, enhancing the value of the map display. Subsets of base data are selected from the database by specifying feature code ranges.

If either the boundary lines of the zones or base data from the map database is selected, the lines which are drawn will be colored according to the operator's selection for the color of the background. The boundary lines or base data will be drawn in the opposite color as the background, so that these lines will be readily visible on the display as well as on a hardcopy.

Once the possible plot colors have been defined and zone colors have been translated and mapped to defined plot colors, the operator may select the set of zone (or translated via the Translation file) colors which will be displayed. The operator may select all colors at once or may select single colors until all desired colors are named.

The operator may select to have the selected map data saved in the local memory of the color terminal in a form we shall call segments here. If segments are chosen, the finished display can

be altered with the zoom, pan or other windowing hardware in the terminal, but the display will be slower and the upper limit on that memory will restrict the amount of data so displayed. If segments are not chosen, the map display will proceed much faster than with segments, but no hardware windowing using the terminal functions may be performed on the completed display.

When the map display has been completed by COLORZ, the same database may be used again to create another, different display. The operator may select to cycle the program, resetting values so that a second, independent display may be created. The operator does have the option to re-use the same dictionary to define the plot color definitions described above, or may opt to reset these definitions in the ways listed above. The only fixed constant is the selection of the map database, which remains the same unless the whole process (macro and all) is terminated and re-started.

One problem of note concerning the use of the COLORZ program is that there exist limits within the software and hardware of the color display terminal. One problem which may occur is due to the limit on the number of points which may define an area to be colored in the display. When zones have more points than this limit, the display terminal can not display them. Attempting to perform the color filling by software may fail due to problems such as incomplete filling of areas where boundaries overlap and halt the filling algorithm. Dividing such areas into two or more areas takes effort and time, but can solve the problem.

Selection of plot colors for area filling and line colors and widths is made possible only by providing the operator with the a set of possible options to choose from. This function is performed by the PALETTE program.

The PALETTE program provides plotted displays of area and line colors from the default color settings for the color plotter and from a systematic mixing of primary colors using the Cyan Magenta and Yellow primary color mixing system. The program offers four options for basic operation.

The first two options produce a complete set of colors in increments of ten percent of the three primary colors. Including zero percent through one hundred percent of each primary, there are eleven separate tables (0 percent cyan through 100 percent cyan) each with 11 columns (0 through 100 percent yellow) and 11 rows (0 through 100 percent magenta). Due to the limit of the color plotter, so many colors are produced in this chart that it is divided into two parts, which comprise the first two options in the PALETTE program.

The second two options in PALETTE provide the area and line color options in the default palettes of the color plotter. The first option provides a plot file to display the entire set of area filling colors, each labeled with the color number by which it may be selected. The entire set of area colors is displayed

as boxes of colors in rows and columns, with each box labeled by the appropriate color number.

The fourth option creates a display of the complete set of line colors and widths. Each line color option is displayed as a sample of the line which is created as a result of the selection. These lines are each numbered with the appropriate selection number for the color plotter selection and arranged in columns. It should be noted that new plots of the four options should be made whenever changes alter the functioning of the plotter.

The final program in this section is not geared to produce color output on a plotter or on a display terminal. Rather, it is designed to distinguish areas by the process of shading, which uses parallel lines at varying angles to differentiate one zone type from another. The output from the SHADEZ program is one or more plot files for the vector plotter, each with shade lines for one or more different areas and possibly the boundary lines for the zones selected for that file.

All shade lines begin at the edge (on the border) of the zone being shaded and end at the zone edger on the opposite side of the zone. If the zone boundary snakes in and crosses the path of the shade line prior to the opposite side (which it must do an even number of times), the shade line will stop and start again as necessary to only have the shade line inside the zone, and to have it everywhere that the path of the shade line lies inside the zone.

The general approach of the SHADEZ program is similar to that of the COLORZ and the CALPLOT program. Zone boundaries are to be extracted from the amap database on the basis of the colors assigned to the zones. Each selected color is mapped to a set of information which defines the shade lines which are to be drawn in the plot for the map.

The plot output of the SHADEZ program is directed to a set of plot files selected by the operator. Within each plot file, the operator may direct shade lines from one or more sets of zone boundaries, including the zone boundary lines as an option. As another option, the operator may select a single plot file in which all boundary lines for the map database are placed in the form ready for plotting on the vector plotter. Conversely, the operator may choose to include no zone boundaries in any file.

As with other similar programs, SHADEZ allows the operator to define the shade lines associated with each selected color in a dictionary file named by the operator. The entries residing in the dictionary file or given to the program by the operator each contain the zone color to be shaded, an angle at which the first set of shade lines is drawn, an angle at which an optional second set of shade lines (cross-hatching) is drawn, and the line type (solid or dashed) with specifications. Any number of selections may be assigned to each output file.

Two other factors affect the function of the SHADEZ program. First is the ability of the operator to restrict the selection of zones based on the area of the zones. At times, it is convenient to exclude very small or very large zones from the process of shading zones. The operator is allowed to specify a minimum and a maximum area for restriction of zones to be shaded. The second factor affecting the function is that the operator may choose a single directory in which all plot files are assumed by SHADEZ to reside. This reduces unnecessary typing by the operator.

The use of zone shading plots to produce color plots may be accomplished in two ways. First, the separations discussed above allow the plot operator of the vector plotter to apply different color pens in the plotting of different plot files. Thus, all the different zone colors may be represented as different angles of shade lines as well as different colors.

Alternatively, the plot files of shade lines, all produced for the vector plotter, can be converted by the X2C program (see above) into the form for the color plotter. The operator must assign line color/width numbers for each plot and select a level at which the lines are to plot, and a color plot of the shade lines will be made. Though the quality of color in these plots may be superior to that of the pens in the color vector plots, changing colors is clearly easier on the vector side since no program must be run.

5. Special Utilities

There is a set of programs in the GIMMAP system that provide unique functions required for processing map data in an automated cartography system. These programs support a variety of map data processing functions not easily categorized other than as utility programs.

The first program in this set is the **ARCHIVE** program which is used to create an archive of map data stored in a DDF format, usually on magnetic tape. The **ARCHIVE** program operates similarly to the **DIGGEN** program in the basic sequence to produce an ASCII file containing x and y coordinates of the selected features in the standard DDF form of clean digitized data files.

The **ARCHIVE** program differs from **DIGGEN** in that the archive is assumed to require all map data available in the map database. Thus, the program does not have to ask the operator to select the feature codes for the features to be sent to the DDF file. In addition, the **ARCHIVE** program can be set up to process a number of map databases, given a file of arc file names for each of the databases.

The second utility program is **RAFPRN**, for random access file printing, a program designed to directly examine and modify the individual records in the files of the map database. With **RAFPRN**

the operator may select any of the files of the database, select a range of record numbers, and have displayed the contents of the selected records from the chosen file. With the modify function, individual values in the selected records may be changed.

The display option in RAFPRN allows the operator to examine the contents of all the available map database files for the map area chosen. This will include the Arc, Node, and Point files for all databases. For those with zones, the Zone and Border files will be available. Those with symbols will have Symbol and Calligraphy files for inspection.

Inspection of the Arc, Node, Point, Zone and Symbol files is direct. The record number of the desired record is given by the operator and the contents of the record are displayed. However, the Border and Symbology files are examined indirectly through the Zone and Symbol files. A chain of Border records describing a zone is listed by selecting the zone. Similarly, selection of a record number for a symbol is used to examine the linked list of calligraphy records for the symbol. Finally, the operator can examine the freelist (linked list of unused records) of any of the database files.

Modification functions can only be performed on a subset of the database files. Only the Arc, Node, Point, Symbol and the Calligraphy files may be modified. The freelist structures and sometimes the "information" records (containing information such as the projection parameters or the map title) can be modified. Modification of values in the records of the database files is made following selection and display of the record contents. The operator then selects the value to be changed and enters a new value for the record. The operator may change more than one value in the selected record.

One other function is provided by the RAFPRN program to provide a general overview of some basic information about the map database. The FSURVEY command prints basic information about the numbers of features and records in the database, map extremes in X and Y coordinates, the map header, map projection parameters for projection type and area.

A production shop using the automated cartography facilities like GIMMAP will be required to record information central to the proper production of each mapping project. Such information is useful to the organization of the mapping projects as they are being worked on. Operators have a source for the specifications of many aspects of the map which must be used again and again in the various programs. This same information also helps protect against misunderstanding and poor communication among production staff, map authors, and editors.

The GIMMAP program which is responsible for this information is the MAPREQUEST, for map project request, program. Using this program, the operator or map requestor will respond to questions

which detail production information for the map project to create a set of records in the **Map Request Database** for the project. A unique number, called the project identifier (PID), is assigned to every project request, and future access to the information is restricted to those with the PID and operator assigned password.

A block of records is built for each map request to contain the information about the map project. Within these records is kept information including the requestor name and group; funding information; the project and map name; the project password; the dates of the request, the desired completion, and the absolute completion; a description of the area of coverage; a list of the features to be included; source information for obtaining feature information; general information on symbology, map title and map scale information; general legend information; specification of the map projection parameters; translation offset values; border location and spacing; line and point symbol type information for all features; registration mark locations and size; latitude and longitude line or tick grid selections; and area labeling.

The MAPREQUEST program provides a number of functions for the operator to exercise with respect to the entries in the map request database described above. The operator or user may use the creation function to begin the initial entry of information to a new project record, with information being solicited by the program for all the topics listed above, and many more.

Once the map request has been entered for the first time, the operator or user may change any value or add values which were unspecified in the first entry using the Modify command in the MAPREQUEST menu. In this process, the operator references the listing of the record content for map projects in the map project request database. The item to be modified is referenced by the record and the item number according to the listing, and the operator may alter the value of the item.

To determine which items in which records must be modified, the operator may view the values in the records by specifying the record and item numbers for the Examine command in the menu. The listing of record content and item numbers must be referenced for the operator to access the desired information. Entire records may be listed by number, or single items may be displayed.

At various times, the operator may wish to produce a listing of the map project entry in the map request database. This may be done by selecting the Transmit command in the menu. With the entry of the PID and password, a complete listing of the items in the records of the project entry in the database. The resulting printout may be sent to all affected parties from the requestor to the editor to the map production supervisor. In addition, a summary of all map projects may be printed for inspection.

The information in the records of a map request entry in the map request database is divided into different parts according to

the type of information. One division of information in these records reserves certain material for the map production operator and prevents the requestor from being aware of it. The material involved is that which is of direct concern only to the operation of GIMMAP programs during the production process, and not of any interest to the requestor. To enter information in this section the operator must use the Supercreate command in the menu.

The information in the map request entry may be examined and modified as often as desired. At a certain point, a commitment is made to proceed on the map project according to specifications as they exist. At this point, the values in the request become "frozen" and are printed to all parties to check for agreement. This process puts the request into "final form", and is initiated by the Final Form command in the menu. Once final form is set, the requestor can not change values. The operator may override the restriction and modify all values.

Finally, the records of the map request for a map project which has been completed may reside in the database as long-term record of the selections made for the project. This may have the advantage that these selections may be consulted for future map projects with some similarities. When the map request serves no further purpose, it may be released from the database by use of the Delete command in the MAPREQUEST menu.

Only one of the utility programs in this section produces plot file output. The POSTER program performs a function which is similar to one part of the function of the MAPGEN program by producing plot data for special point symbols created for point locations in a file, rather than from a map database like the MAPGEN program.

The POSTER program accepts a file of point locations given as projected (x,y) or geographic (longitude,latitude) locations. If geographic coordinates are used, the operator must specify the parameters of the map projection, and the points are projected on input prior to point symbol plotting. The form of the input file may be one of three alternatives: (1) fixed-length, formatted; (2) free format, data sensitive; or (3) fixed-length binary.

The selection of symbol types and sizes may be made in two different ways. The symbol type and size information may be included with the location as part of the record in the input file, or this may be specified by the operator. When the size is specified by the operator, the one value is applied to all points in the input file. However, the code values in the input file may be used to select a different type of symbol for each point.

The symbols available for the point locations are geared to the special symbols used for the Oil and Gas industry, but most standard symbols (such as those available in the MAPGEN program) are also available. The oil and gas well symbols and the dry hole symbol are all available as are other symbols from the Oil

and Gas industry. In addition, the basic symbols for general map applications are available and include the plus mark, the square, the triangle, the X symbol, the circle and other general symbols.

For certain symbols (such as the solid circle), additional information (such as the separation distance between concentric circles) must be provided by the operator. One final function of the POSTER program is the automatic clipping provided to remove any point symbols which lie outside the rectangle defining the map area. If only part of the symbol lies outside, then only that part will be removed. The rest of the symbol will appear on the plot.

One special utility program extracts data from the database to produce (currently only) plot files of lines for which the set of points representing the selected line has been reduced in size (number of points) by a process called **line generalization**. The program is **SIMPLE**, for simplification of lines through selective reduction of points. This program has not reached a production stage in its current form, but could easily be adapted to produce DDF output or useful plot files, or to generalize directly in the map database files.

Two different methods of generalization are available. One is the JENKS method, named after the distinguished professor of Geography at the University of Kansas. This method uses three criteria to examine points "on the fly" in making a single pass along the line to be generalized. Setting the three values to effective levels requires substantial experience with the results of the JENKS method.

The second generalization method is the DOUGLAS technique, once known as the Douglas-Peucker (Poiker) method. The name has since been changed to reflect Dr. Poiker's modesty. The DOUGLAS method views the line as a whole, dividing it into two pieces of varying length based on a single criteria. The two pieces are then each treated as separate lines for which the same process of dividing is repeated. The divide-and-conquer process is repeated on ever-smaller segments until the dividing process fails to find a point at which to divide for all segments.

The dividing process is simply to assume a base line exists between the two endpoints of the segment. Then, the distances between the other points of the segment and the base line are calculated and the point of division is that point for which the distance to the base line is the greatest.

However, the single criteria mentioned above is a minimum distance criteria. If the distance from the potential division point to the base line (the greatest for the intermediate points) is less than the criteria value, no subdivision of the segment occurs. The intermediate points are discarded, and only the two endpoints are kept. If the point with the greatest distance to the base line has a distance greater than the minimum distance

criteria specified by the operator then the segment is divided at that point and the two new segments are analyzed similarly.

When one segment has been analyzed to the point where the remaining intermediate points are discarded and no new segments are found, analysis returns to the matching (other half) segment of the segment just completed, or to the "parent" segment from which those two came. In this unbalanced, binary subdivision of the original line, excellent generalization provides very good representation of the original line with great reduction in the number of points remaining.

One of the special utility programs operates directly on the map database to correct a serious situation in which the display (Qplot) file may contain numerous incorrect values or may have become completely erroneous. This condition may result from many different causes, but usually will be the direct result of errors in the programs which access the display file.

The purpose of the **RENEWQ**, for renew Qplot, program is to reconstruct the values in the display file in part or in whole, from values (assumed to be correct) in the parallel Point file. The result of the **RENEWQ** program repairing the Qplot file is to make the special, full-map view provided by the **ALL** command in the (vector terminal) editing programs of **GIMMAP**.

The **RENEWQ** program allows the operator to select the range of records in the display file which are to be regenerated from the values in the Point file. This range is selected indirectly by the selection of a start and an ending arc whose associated Qplot records set the range of display file records which are to be regenerated. This range may be a single record or the entire collection of records in the file.

From the first to the last of the indicated records, **RENEWQ** reads the parallel record from the Point file, converts all the projected (x,y) values to the full view, screen coordinates of the display file, and places these values in the corresponding Qplot record along with the proper pointers to maintain the same structure as in the Point file.

The final program in this special utility section is one used to survey one or more map databases to determine the basic content so that planning for future operator work may be based on knowledge of the presence and absence of map features.

The **SUMMIT** program produces a summary of the features which exist in the set of map databases selected by the user. The list of databases is entered by specifying a file which contains the names of the Arc files of all the databases. The result of the program analysis is a report of the total counts of all features found in all the databases by feature type, listed in increasing order according to the feature codes of the features found in all the databases.

6. Map Projection Software

There is a set of programs and libraries (software which must be "called" by other programs and can not "run" by itself) which provide functions for projecting map data from **geographic** (latitude and longitude) coordinates to cartesian coordinates (x,y). Geographic coordinates represent locations on the surface of the earth, while cartesian or **projected** coordinates represent locations as found in the digitizer and on the vector plotter.

The surface of the earth is approximated by an **ellipsoid**, a three-dimensional solid created by rotating an ellipse around the major axis. The conversion of locations on the surface of this model to the projection surface of a tangent cone, cylinder or a plane is a mathematical exercise which can become complex. The calculation of distances from the equator, for example, is a very complex matter.

Map projection is this process of converting point locations on the surface of the earth to projected locations as used in the map databases and elsewhere throughout GIMMAP. Deprojection is the inverse of projection. It is the conversion of projected map locations back in to geographic coordinates, representing points on the surface of the earth.

The **PROJCT** program provides the operator with the ability to project points from a file or interactively at the terminal. If the interactive mode is selected, then points are entered at the terminal, projected and listed at the terminal. If input points come from a file, then the projected output points are sent to a file selected by the operator.

The projection process requires the operator to specify the parameters of the projection. These will include the projection type, the projection area, and the map scale among others. The projection type is one of those available in the libraries of map projection (described below) which currently include the Lambert Conformal Conic and the Modified Polyconic projections. There is one special instance of the Lambert projection which represents a de facto standard for maps of the entire state of Kansas, called the "(GIMMAP) standard map projection for Kansas".

Selection of the projection area fixes the reference system of the projection and sets the origin to the location selected by the operator. This rectangle, defined in latitude and longitude, is also used to window map data to clip (remove) data outside the selected area. The projection software is designed to place the southwest corner of the area at (1.,1.) and the southeast corner with Y = 1.

The output of the **PROJCT** program may be a projected mirror of the input, or may be formatted as a DDF file, ready for input to create a new map database, add to an existing one, to create a new database or add to an existing one. The optional z-values in

the input file may be used to create feature codes in the output of the DDF file. Flags will be entered in the output of the DDF file, but are not necessary in the input file.

The inverse of projection is deprojection, the conversion of projected (x,y) points in cartesian coordinates to locations in geographic coordinates of longitude and latitude. The advantage of using the geographic reference system is that it is the most universally accepted system for locations. Using the geographic system, map data may be transferred between installations with a high likelihood that it will be useful.

As with projection, deprojection may occur interactively at the terminal or with files. The deprojection process, which is provided by the DEPROJCT program, requires the operator to set up the projection information (type, area, scale, etc.) required by the projection libraries. This information is the same as the information required for PROJCT even though deprojection is the inverse process of projection.

Both PROJCT and DEPROJCT, along with other programs, rely on a group of commonly used projection functions which are collected in a GIMMAP library called PROMAN, for projection management. Of the functions provided by PROMAN, half support one map projection while the other half support a second. The first projection may be the "input" map projection while the second is the "output" projection. The first used for deprojection and the second used for projection to a new projection common to all output. This is the process used by the MAPGEN program in quilting maps together to form a larger area map at a different scale.

The PROMAN library includes functions for defining both in and out projections by type, area, scale, and other parameters; projecting either in or out by the selected type of projection; converting degrees to radians; normalizing the map data to lie within the map area; and finding the distance from the equator to a point along a line of longitude.

Another version of the PROMAN library is designed for more public use of the projection software. The GIMMAP.PROJCT library provides a slightly higher level of use for programs outside the GIMMAP system. For example, the process of deprojecting a point in one projection and then projecting it in a new one is called **reprojection**, a function obtained in GIMMAP by calls to the two routines indicated. However, in the GIMMAP.PROJCT library, only the single REPROJCT routine must be called to achieve the desired result, provided the projections have already been set up.

7. Legal Reference System Conversion

The **Legal reference** system in Kansas is a system of location based on an imperfect grid of once-surveyed, almost-mile-square rectangles called **sections**. Furthermore, sections are grouped in

six-by-six rectangles called **townships**. This system was set up to provide a convenient and consistent method to define property boundaries and other locations in the state.

As it turned out, the curvature of the earth and less than perfect surveying techniques (such as trying to make two separate surveys come together properly from opposite directions) left the legal reference grid as a collection of points, which could only approximate a rectangular grid. Many of the corners were marked on the ground, but townships progressively south are shifted out of position with those to the north. And sections are not a mile square, but do generally approximate that.

Thus, there exists a need to convert between the legal and the geographic reference systems because many important locations (including oil and gas wells and property boundaries) are still recorded in legal reference. Furthermore, the state of the legal grid, as a collection of semi-random points, should indicate the difficulty of performing this needed conversion.

Conversion between locations in the legal and the geographic reference systems in the GIMMAP system is made possible by the creation of a database of section and township corners taken from the complete set of digitized corners in the **Kansas Cartographic Database (KCD)**. The KCD is actually a collection of more than two thousand map databases covering the state of Kansas. The KCD contains map data for many features, including the township and section corners, as digitized from the series of 7.5' topographic maps published by the U.S. Geological Survey.

The database of section and township corners created for the conversion between the **LEGAL** and **gEOGRAPHIC** reference systems is called **LEOBASE**. The database contains an entry (record) for each township in the state. Each township record contains geographic locations of all the section corners for the township in a fixed order.

Converting from legal to geographic coordinates is a matter of going directly to the township record, finding the corners of the section, and subdividing the section into quarters or halves as described. Conversion from geographic to legal coordinates is performed by making an initial guess via a rough conversion from latitude to township and from longitude to range. Following the guess, searches are made from township to township and section to section until the correct section is found. Subdivision is then generated to the finest level.

Conversion between legal and geographic coordinates in the GIMMAP system is supported by a library called **LEOCVT**, for **LEO** convert. The library consists of a single, large routine of the same name (**LEOCVT**) which converts locations in Kansas from either reference system to the other by use of the **LEOBASE** database.

The LEOCVT routine accepts a switch indicating the direction of the desired conversion. If the switch indicates a conversion from geographic coordinates, then LEOCVT uses the longitude and latitude to find the legal location of the point. All the values required to completely describe a legal location are set to the converted location. These include the township, range, section and subdivision descriptions.

If the switch indicates a conversion from legal coordinates, then LEOCVT uses several supplied values including the township, range, section and subdivision designations. Subdivisions may be specified by different methods, and either the center of the area or a corner of the section may be selected. Either success or a description of the failure of a conversion in either direction is indicated by the LEOCVT routine.

A second version of the LEOCVT library exists to support the public need for legal and geographic conversions. This library is called the **GIMMAP.LEOCVT** library, and differs very little from the LEOCVT library. Differences may increase as increased public use requires additional communication to resolve user errors and to improve comprehension.

The information contained in the LEOBASE database can be converted to a readable form and put on a tape for transfer to other facilities. In this form, the information does not present the ability to convert between the two reference systems, but such capability can be generated using the file.

Creation of this readable (ASCII) file is the function of the **LEOASCII** program. The output of the **LEOASCII** program is a readable file which contains the records of the LEOBASE database in a set order. Generally, township record information occurs from north to south and from west to east across the state. The section corners within a township are similarly ordered.

The inclusion of whole townships in the **LEOASCII** file means that township corners are repeated up to four times and section corners on the edges of the townships are duplicated in the file. This small redundancy is accepted for the convenience of easy access to complete sets of corners by township.

The township and section corner information in the LEOBASE database can be interrogated and modified interactively by use of the **LEOQUERY** program. With this program, an operator may inspect the locations of the township and section corners in LEOBASE by specifying the township and range designations of the township, and then selecting the section by number.

The locations in longitude and latitude of the four corners of the chosen section are displayed at the appropriate corners of a rectangular area of the display screen on the terminal where they may be examined or copied at will. When ready to continue, the operator must select the next section to inspect within the

chosen township. The operator may select any other section in the township and a similar display of the four corners will be posted. When done with that township, the operator may easily select a non-existent (number zero) section to select a different township.

There are places around the state where the section corners in a township are partly non-existent. This occurs at the state edges and especially in the northeast where the boundary of the Kansas river acts as the state boundary. Where such values are non-existent, the LEOBASE database stores negative values for the longitude and the latitude (usually -9.99,-9.99).

An important function of the LEOQUERY program is to modify the contents of the LEOBASE. This function is required to make corrections to the section and township corner locations which may have occurred for numerous reasons. The primary causes for errors in the locations are (1) the location and digitization of the point from the USGS map and (2) the process of sorting and organizing the nearly 80,000 points covering the state.

Regardless of the cause for the errors in the locations, the most likely source by which the errors become known is the use of the LEOBASE to convert locations between reference systems. The correction of these discovered errors is then accomplished by the direct inspection and modification of locations in the database.

8. Hershey Alphabet Sets

The Hershey alphabets (from A. V. Hershey,...) are sets of points representing pen strokes required for a plotter to draw the various characters of the alphabet, the digits, and other so called "special characters" for a number (23 here) of different fonts such as Gothic and Italic. Different versions of some of the fonts (simplex, duplex,...) use multiple strokes to provide the appearance of varying thicknesses of pen width or boldness.

The Hershey alphabets are used by GIMMAP programs (CYMBAL and MAPTTL primarily) to create symbology for map annotation. This use of the Hershey files is made possible by construction of the GIMMAP Hershey database from the raw data supplied with the original form of the Hershey information.

The original Hershey data passes through two GIMMAP programs before the GIMMAP Hershey database is constructed. The first of these programs is the **A2FELIX**, for ASCII to Felix form, program which converts the Hershey data in its original, ASCII form into an intermediate state defined by the **FELIX** system (a contouring package for minicomputers, written by M. J. McCullagh,...). The Felix form is a sequential structure, which is not as efficient as the final form in GIMMAP.

The Felix form of the Hershey data is further refined into the final GIMMAP form for the Hershey database by means of the **HERSH** program. In the final form, the Hershey database consists of two direct access files with one containing alphabet pointers for direct access from a selected character in a chosen alphabet to the record in the other, pen stroke file for strokes to draw the selected character.

The two file GIMMAP Hershey database may be examined by the operator interactively at a terminal. The actual numbers stored to represent the pen strokes for any selected character of any alphabet may be viewed on demand by use of the **HERSHHEXAM** program, a function used occasionally to check the Hershey set for errors, to analyze possible additions or changes in the file, or to help debug programs accessing the database.

A more useful tool for examining the GIMMAP Hershey database is the **HERSHPLOT** program, used to generate a plot for the vector plotter to produce an orderly display of all the Hershey alphabet sets. The resulting plot is produced at a size which is large enough to clearly display all the characters of all the alphabets in the Hershey sets, but is small enough to handle easily. All the characters are displayed in the order of their existence in the database, which corresponds to the sequence of keyboard keys which must be typed to produce the characters. Currently, there are twenty-three Hershey sets available.

9. Sorting

Occasionally it is advantageous to have a list of numbers or names sorted into increasing (or decreasing) order for entry into a program. One example is to feed such a list of Arc file names to the **MAPGEN** program so that progress of the program through a large set of files can be more easily monitored.

Two programs exist for the purpose of sorting numbers and names into ascending order as quickly as possible. The program which sorts names or other alphanumeric information in which the entries may have up to sixty characters, is called **ASORT**, for alphanumeric sorting. Up to one thousand such entries may be sorted at one time by placing them in a file and naming the file to the **ASORT** program. A file must also be named to receive the sorted entries.

The program which sorts the numeric information in the form of (up to twelve-digit?) integer values stored in a file named by the operator as the input file to be sorted. Up to ten thousand such integers may be included in the file for sorting. Sorting of integers is performed by the **QSORT**, for quicker sort, program.

The algorithm (method or procedure) used in both the **QSORT** and **ASORT** programs was devised by R. C. Singleton as a variation on the Quicksort algorithm which in turn was a variation on the

more famous Quicksort algorithm of ? Hoare. The three approaches collectively represent perhaps the three most efficient and the fastest sorting algorithms for the kinds of sorting they desired here.

10. Oil and Gas Fields of Kansas

Two programs provide very specific application functions for display of Oil and Gas data in the state of Kansas. The first is the **KONTIKI** program which produces a color display of the oil and gas fields of the state on the color terminal. The second is the **OGNAMES** program designed to reproject the field names from the single base in which they reside to fit other maps of different projections.

The **KONTIKI** program creates a display of the state of Kansas as a collection of counties forming the state in the foreground with a background selected by the operator either for hardcopying (grey) or for taking pictures (deep blue). With county lines for reference and optional title with date and colored boxes, the map is covered with the fields selected by the operator.

The choices include all combinations of the available units which are the oil fields, gas fields, oil and gas fields, shallow gas fields, and gas storage fields. Each type is displayed in a unique, appropriate color, with the storage areas being a shade pattern of cross-hatched lines since this type may lie on top of other fields.

The function of the **OGNAMES** program is to extract the field names from the existing database for the Oil and Gas field names and to convert the names to fit another map. This process is similar to that of the **SYMGEN** program which also reprojects the symbology information.