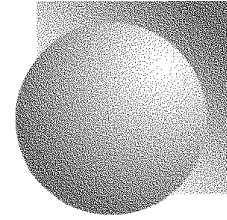


Encompass BLOB Format

May 30, 2002 NcBlob.doc



This paper describes the format of the BLOB (Binary Long Object) fields used by Geodesy's Encompass components for storing GDS-like map graphics. It is intended to be read by those who wish to understand more about the capabilities of the Encompass BLOB and those that wish to write software that directly reads and/or writes the BLOB.

About Encompass and BLOBs

Encompass components and applications store their geographic features and feature attribute data in relational databases. Each group or "class" of features is stored together in a table – parcels are stored together in one relational table, storm sewer manholes in another, for example. Each feature, a single entity like a parcel or manhole, is stored in one row of the feature class' table. The individual feature's graphic description is stored in a "BLOB" (Binary Long Object) column. Different relational databases use different datatypes for BLOB data – Microsoft Access, for example, uses its "OLE Object" data type, Oracle uses its "Long Raw" datatype.

For more information about Encompass, refer to Geodesy's "About Encompass" paper. For more information about Encompass' ability to read and write BLOBs, refer to the "Library (gxLib.dll)" paper.

BLOB Header

All Encompass BLOBs start with a 16-byte GUID (a Windows Globally Unique ID) and a count of the number of graphic primitives to follow.

<u>Byte Length</u>	<u>Contents</u>	<u>Value(s)</u>	<u>Notes</u>
16	GUID	{4550e5a0-6b58-11d3-9212-00a0cc412e25}	First byte of GUID indicates BLOB version number – starting with 1, the current version.
4	Primitive Count	long integer	Count of primitives to follow (at least 1; typically 1).

Data for each primitive follows. The primitive types are "PT", "LI", "LP", "LL", "TX", "TA", and "TD" and can be read as either a two-character string or a two-byte integer. At this time the obsolete "SY" (symbol) primitive is still read but is read as a "PT" (point) and the obsolete "LC" (circle) primitive is still read but is read as an "LI" (line).

Point (PT) Primitive Type

Used for symbol insertions. The depiction of the symbol is determined at map draw-time, not by the BLOB contents.

<u>Bytes</u>	<u>Contents</u>	<u>Value(s)</u>	<u>Notes</u>
2	Primitive Type	"PT" (21584)	The "SY" (22867) primitive type is now obsolete and interpreted as a Point.
8	Size	double-precision real	Symbol height in map units. Zero means default symbol height.
8	Angle	double-precision real	Angle in degrees. Zero for upright. Positive values spin symbol counter-clockwise.
8	X Position	double-precision real	X coordinate in map units.
8	Y Position	double-precision real	Y coordinate in map units.

The Line (LI) and Leader-Line (LL) Primitive Types

The color of the line is determined at map draw-time.

<u>Bytes</u>	<u>Contents</u>	<u>Value(s)</u>	<u>Notes</u>
2	Primitive Type	"LI" (18764) or "LL" (19532)	The "LC" (17228) circle primitive type is now obsolete and interpreted as a line.
2	Style	integer	Line pattern. Zero for solid. Patterns 1 thru 5 are reserved for Windows line types (see below).
8	Size	double-precision real	Line width in map units. Zero for hairline.
4	Vertex Count	long	2 or greater.
8	X Coordinate 1	double-precision real	Vertex 1 X coord. in map units.
8	Y Coordinate 2	double-precision real	Vertex 1 Y coord. in map units.
8	Bulge Factor 2	double-precision real	Bulge factor for line between Vertex 1 and 2. (See Bulge Factor notes below.)
8	X Coordinate 2	double-precision real	Vertex 2 X coordinate.
8	Y Coordinate 2	double-precision real	Vertex 2 Y coordinate.

(data for additional vertexes follow in Bulge-X-Y groups of 24 bytes...)

The windows line types are:

- 0 = Solid
- 1 = Dash

- 2 = Dot
- 3 = Dash-Dot
- 4 = Dash-Dot-Dot
- 5 = Transparent

Polygon (LP) Primitive Type

The color of the polygon's outline is determined at map draw-time.

<u>Bytes</u>	<u>Contents</u>	<u>Value(s)</u>	<u>Notes</u>
2	Primitive Type	"LP" (20556)	
2	Style	integer	Fill color and hatch pattern. -1 to 6100 (see below).
8	Size	double-precision real	Outline width in map units. Zero for hairline.
4	Vertex Count	long	4 or greater.
8	X Coordinate 1	double-precision real	Vertex 1 X coord. in map units.
8	Y Coordinate 2	double-precision real	Vertex 1 Y coord. in map units.
8	Bulge Factor 2	double-precision real	Bulge factor for line between Vertex 1 and 2. (See Bulge Factor notes below.)
8	X Coordinate 2	double-precision real	Vertex 2 X coordinate.
8	Y Coordinate 2	double-precision real	Vertex 2 Y coordinate.

(data for additional vertexes follow in Bulge-X-Y groups of 24 bytes...)

The last vertex must have the same X and Y values as the first vertex.

The values of -1 to 6100 for fill color and hatch patterns are used as follows:

- -1 means transparent (no fill).
- x000 means opaque, background color.
- x050 means halfway from background to outline color.
- x100 means same color as polygon outline.
- 0xxx means solid fill with interpolated color (between background and polygon outline).
- 1xxx means *horizontal hatch* with interpolated color.
- 2xxx means *vertical hatch* with interpolated color.
- 3xxx means *diagonal hatch f* with interpolated color.
- 4xxx means *diagonal hatch b* with interpolated color.
- 5xxx means *cross hatch* with interpolated color.
- 6xxx means *diagonal cross hatch* with interpolated color.

Text (TX), Annotation (TA), and Data-Annotation (TD) Primitive Types

The color of the text (and the meaning of the font integer for that matter) is determined at draw-time.

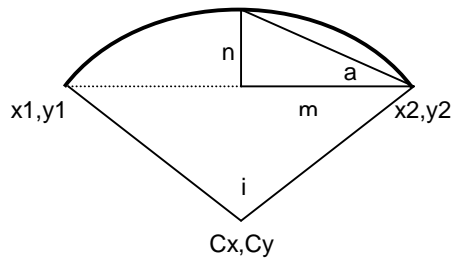
<u>Bytes</u>	<u>Contents</u>	<u>Value(s)</u>	<u>Notes</u>
2	Primitive Type	"PT" (21584)	The "SY" (22867) primitive type is now obsolete and interpreted as a Point.
2	Style	integer	1=Arial, 2=Courier New, 3=Times New Roman. Higher numbers (4..2046) are for application-specific fonts. Bit 14 is set for italic, bit 15 is set for bold. For example, &H2001 means Arial Italic, &H4001 means Arial Bold, and &H6001 means Arial Bold Italic.
8	Size	double-precision real	Symbol height in map units.
2	Justification	"TL", "TC", "TR", "CL", "CC", "CR", "BL", "BC", "BR"	
8	Angle	double-precision real	Angle in degrees. Zero for upright. Positive values spin symbol counter-clockwise.
2	Text Length	integer	Length of following text string in characters.
<i>varies</i>	Text String	string	No terminator ("\0") required.
8	X Position	double-precision real	X coordinate in map units of the text justification point.
8	Y Position	double-precision real	Y coordinate in map units of the text justification point.

The text string used by Data-Annotation primitives is the name of the column (in the same table as the BLOB) to annotate. If the column name is not found in the table, the text string itself is drawn to the map.

Bulge Factors

The concept of a bulge factor is used by GDS and AutoCAD in place of a radius value. It expresses the degree of curvature between two points in a line or polygon (a "segment") in terms of the height of the curve over the distance between a segment's end points.

The following formulas apply:



$B = \text{Bulge Factor}$

$B = n/m$

$B = \tan(a)$

$i = 4 * \text{atan}(B)$

$f = 0.5 * (B - (1/B))$

$Cx = (x1+x2+(f*(y2-y1))) / 2$

$Cy = (y1+y2-(f*(x2-x1))) / 2$

A bulge factor of zero means the line is straight between the two vertices. A bulge factor of one means the line is a semi-circle. An infinite bulge factor (if there were such a thing) would be a circle.

A positive bulge factor means the arc is counter-clockwise from the vertex preceding the bulge factor to the vertex with the bulge factor.

