FS X BGL File structure

This is a first attempt to understand the file structure of the FS X scenery files. It is still incomplete, since I do not understand all the features.

© Winfried Orthmann
eMail: winfriedorthmann@yahoo.com

BGL Files Overview

FS X BGL-files

File Name	Contents	Sections
APX*.BGL	Airports	including objects coded within the airport records thru VisualModel and TaxiwaySign subrecords
ATX*.BGL	Waypoints and boundaries	
BRX*.BGL	extrusion bridges	
NVX*.BGL	Navaids	
OBX*.BGL	Airport objects	including .mdl data
[city name].BGL	city objects	including .mdl data
cvx*.bgl	terrain vector data	
	files in the BASE subdirectory of the scenery directory	
	Object libraries in the Global subdirectory of the scenery directory	

In contrast to FS9, there are essentially no more files in the "old" file format.

Data types

Latitude and longitude are no longer represented as before. Each location on the earth is fixed in the LOD grid. Longitude and latitude are each represented by a 4 byte value (`DWORD`). The formula for obtaining the decimal values is as follows:

```
      (double) Lon = (DWORD) Lon * (360.0 / (3 * 0x10000000) – 180.0)
      (double) Lat = 90.0 - (DWORD) Lat * (180.0 / (2 * 0x10000000))
```

Altitude is given in 1/1000 m as DWORD.

Pitch, bank and heading: is given as ANGLE16 in form of a DWORD. The formula for obtaining the decimal value is as follows:

```
      (double) Pitch = (DWORD) Pitch * 360.0 / 0x10000
```

ICAO Identifiers and region codes are coded in a special format. Each number and letter has a value from 0 .. 37:

blank	00
digits 0 .. 9	02 .. 11
letters A .. Z	12 .. 37

The code is calculated by starting from left: the value of the first digit/letter is multiplied by 38, then the value of the next digit/letter to the right is added, the sum s multiplied by 38, and as long as there are more digits/letters this process is repeated.

The region codes have only 2 digits/letters and the result is used as such; for the ICAO identifiers for airports, ILS, VOR, NDB and waypoints there are up to 5 digits/letters, and the result is shifted left by 5 positions, i.e. multiplied by 0x20. Bits 0 .. 4 of the resulting DWORD are frequently used for other purposes.

The ICAO identifiers for primary and secondary ILS in a runway record are not shifted.

BGL file header

The BGL file header consists of a fixed part with the length of 0x38 (54) bytes and a variable number of section pointers.

The fixed part of the header has the following structure:

offset	length	format	description	contents
0	2	WORD	New bgl ID	0x0201
2	2	WORD	Probably version	0x1992
4	4	DWORD	size of header	0x0038
8	12	DWORD[3]	Unknown, possibly connected to compilation time	
20	4	DWORD	number of section pointers in header	
			rest are references to the geographical area covered by the contents of the file	

The section pointers are located immidiately after the fixed part of the header, i.e. starting at offset 0x0038.Each section pointer is 0x14 bytes long and has the following structure

offset	length	format	description	contents
0x00	4	DWORD	type of section. The following types have been identified: 0x0003: airport data 0x0013: VOR / ILS data 0x0017: NDB data 0x0018: markers 0x0020: boundary data 0x0022: waypoint data 0x0023: geopol data 0x0025: scenery objects 0x0027: namelist 0x002b: mdl data 0x002c: additional airport data 0x002e: exclusionRectangle	
0x04	4	DWORD	unknown	
0x08	4	DWORD	number of subsection pointers in section header	
0x0c	4	DWORD	offset from file start to section header	
0x10	4	DWORD	size of section header	

BGL section header

The section pointer records in the header point to the section header, which consist of 1 to n subsection pointer records. The number of subsection pointer records present is given in the section pointer record as mentioned above.
Each subsection pointer record is 0x10 bytes long and has the following structure:

offset	length	format	description	contents
0x00	4	DWORD	ID. Since many sections are subdivided into subsections according to the location of the objects in the LOD (or QMID) system, this ID is an index giving the location of the object in space.	
0x04	4	DWORD	number of records in the subsection	
0x08	4	DWORD	offset from file start to start of object records in this subsection	
0x0c	4	DWORD	size of subsection	

The section header for records of Boundary and Geopol type have a different structure. They consist of a 0x10 bytes long record for every subsection with the following structure:

offset	length	format	description	contents
0x00	4	DWORD	ID. As mentioned above this ID is an index indicating the location of the objects in the QMID space	
0x04	4	DWORD	Number of records in the subsection	
0x08	4	DWORD	Index into the list following these records	
0x0c	4	DWORD	unknown, seems always to contain	0x00000000

after this list follows a record for every subsection with the following structure, which is repeated for the number of records, i.e. if there are 2 records in the subsection, there will be two records of the following structure:

offset	length	format	description	contents
0x00	4	DWORD	offset from start of file to start of records	
0x04	4	DWORD	length of subsection	

BGL subsections

The subsections for each kind of objects (airports, sceneryObjects, ILS etc) consist of a list with the individual records following each other. Each record has at offset 2 a DWORD giving the total size of this record. Thus it is easy to find the start of the next record. Each section and thus each subsection contains records of the same general type. A number of records can contain subrecords, which in turn have a size field at offset 2 after a WORD identifying the type of subrecord.

Airport

Each airport record consists of a fixed part with the length of 0x38 bytes, followed by a variable part with 0..n subrecords of different types. The structure of the fixed part is as follows:

offset	length	format	description	contents
0x00	2	WORD	ID	0x003c
0x02	4	DWORD	size of airport record	
0x06	1	BYTE	number of runways subrecords	
0x07	1	BYTE	number of com subrecords	
0x08	1	BYTE	number of start subrecords	
0x09	1	BYTE	number of approach subrecords (?)	
0x0a	1	BYTE	Bit 0-6: number of aprons (?) Bit 7: flag for deleteAirport record	
0x0b	1	BYTE	number of helipad subrecords	
0x0c	4	DWORD	longitude	
0x10	4	DWORD	latitude	
0x14	4	long	altitude in m	
0x18	4	DWORD	longitude of tower (if present)	
0x1c	4	DWORD	latitude of tower (if present)	
0x20	4	DWORD	altitude of tower (if different from airport)	
0x24	4	float	magnetic variation	
0x28	4	DWORD	ICAO ident (special format)	
0x2c	4	DWORD	unknown	
0x30	4	DWORD	unknown	
0x34	1	BYTE	unknown	
0x35	1	BYTE	traffic scalar	
0x36	2	WORD	unknown	

The following subrecords can be present after the main **airport** record:

NAME

offset	length	format	description	contents
0x00	2	WORD	ID	0x0019
0x02	4	DWORD	Size of name subrecord	
0x06		STRING	airport name	

This subrecord seems to be present in every airport record, and it is always the first one immediately after the fixed part.

INCLUDED TOWER SCENERY OBJECT

offset	length	format	description	contents
0x00	2	WORD	ID	0x0066
0x02	4	DWORD	Size of subrecord	
0x06	4	DWORD	Size of the included scenery object	

After this record we find an included scenery object with an internal structure identical to that of other scenery objects (see below) and including possible attachments. The BglComp compiler allows only one scenery object to be included at this point, but in some FS X scenery files we find more than one objects included here. If present, the subrecords of this type appear immediately after the Name subrecord.

RUNWAY

The runway subrecord consists of a fixed part with a length of 0x34 bytes and a variable number of sub-subrecords. The fixed part has the following structure;

offset	length	format	description	contents
0x00	2	WORD	ID	0x0004
0x02	4	DWORD	size of runway subrecord	

0x06	2	WORD	type of surface. The following numbers have been found:
			0x0000 CONCRETE; 0x0001 GRASS;
			0x0002 WATER; 0x0004 ASPHALT;
			0x0007 CLAY; 0x0008 SNOW;
			0x0009 ICE; 0x000c DIRT;
			0x000d CORAL; 0x000e GRAVEL ;
			0x000f OIL_TREATED;0x0010 STEEL_MATS;
			0x0011 BITUMINOUS; 0x0012 BRICK;
			0x0013 MACADAM; 0x0014 PLANKS;
			0x0015 SAND; 0x0016 SHALE;
			0x0017 TARMAC; 0x00fe UNKNOWN;	

0x08	1	BYTE	primary runway number (01 – 36, then 37ss. for NORTH, NORTHEAST, EAST, SOUTHEAST, SOUTH, SOUTHWEST, WEST, NORTHWEST	
0x09	1	BYTE	primary runway designator 0 = NONE, 1 = LEFT, 2 = RIGHT, 3 = CENTER, 4 = WATER, 5 = A, 6 = B	
0x0a	1	BYTE	secondary runway number	
0x0b	1	BYTE	secondary runway designator	
0x0c	4	DWORD	ICAO ident. for primary ILS (special format), 0x0000 if none	
0x10	4	DWORD	ICAO ident. for secondary ILS	
0x14	4	DWORD	longitude	
0x18	4	DWORD	latitude	
0x1c	4	long	elevation	
0x20	4	float	length in m	
0x24	4	float	width in m	
0x28	4	float	heading	
0x2c	4	float	pattern altitude	
0x30	2	WORD	marking flags: BIT 0: edges; BIT 1: threshold BIT 2: fixedDistance BIT 3: touchdown BIT 4: dashes BIT 5: ident BIT 6: precision BIT 7: edgePavement BIT 8: singleEnd Bit 9: primaryClosed BIT 10: second.Closed BIT 11: primaryStol BIT 12: secondaryStol Bit 13: alternateThreshold Bit 14: alternateFixed Distance Bit 15: alternateTouchdown	
0x32	1	BYTE	light flags: BIT 0-1: edge (00 none, 01 low, 10 medium, 11 high) BIT 2-3: center (as with edge) BIT 4: flag for centerRed marking flags BIT 5: alternatePrecision BIT 6: leadingZeroIdent BIT 7: noThresholdEndArrows	

0x33	1	BYTE	pattern flags: BIT 0: primaryTakeoff (0 = YES) BIT 1: primaryLanding (0 = YES) BIT 2: primaryPattern (0 = LEFT) BIT 3: secondaryTakeoff BIT 4: secondaryLanding BIT 5: secondaryPattern BIT 6-7: unused (?)	

The following sub-subrecords can be present within a runway subrecord:

OffsetThreshold

offset	length	format	description	contents
0x00	2	WORD	ID primary: secondary	0x0005 0x0006
0x02	4	DWORD	Size of sub-subrecord	0x0010
0x06	2	WORD	surface (same as in runway)	
0x08	4	float	length in m	
0x0c	4	float	width in m	

BlastPad

offset	length	format	description	contents
0x00	2	WORD	ID primary: secondary	0x0007 0x0008
0x02	4	DWORD	Size of sub-subrecord	0x0010
0x06	2	WORD	surface (same as in runway)	
0x08	4	float	length in m	
0x0c	4	float	width in m	

Overrun

offset	length	format	description	contents
0x00	2	WORD	ID primary: secondary	0x0009 0x000a
0x02	4	DWORD	Size of sub-subrecord	0x0010
0x06	2	WORD	surface (same as in runway)	
0x08	4	float	length in m	
0x0c	4	float	width in m	

VASI

offset	length	format	description	contents
0x00	2	WORD	ID primary left : primary right: secondary left: secondary right:	0x000b 0x000c 0x000d 0x000e
0x02	4	DWORD	Size of sub-subrecord	0x0018
0x06	2	WORD	type 0x01 = VASI21 0x02 = VASI31 0x03 = VASI22 0x04 = VASI32 0x05 = VASI23 0x06 = VASI33 0x07 = PAPI2 0x08 = PAPI4 0x09 = TRICOLOR 0x0a = PVASI 0x0b = TVASI 0x0c = BALL 0x0d = APAP/PANELS	
0x08	4	float	biasX	
0x0c	4	float	biasZ	
0x10	4	float	spacing	

20	4	float	pitch	

ApproachLights

offset	length	format	description	contents
0x00	2	WORD	ID primary: secondary	0x000f 0x0010
0x02	4	DWORD	Size of sub-subrecord	0x0008
0x06	1	BYTE	system 0x00 = NONE 0x01 = ODALS 0x02 = MALSF 0x03 = MALSR 0x04 = SSALF 0x05 = SSALR 0x06 = ALSF1 0x07 = ALSF2 0x08 = RAIL 0x09 = CALVERT 0x0a = CALVERT2 0x0b = MALS 0x0c = SALS 0x0e = SSALS	
0x07	1	BYTE	number of strobes	

(end of runway)

Helipad

offset	length	format	description	contents
0x00	2	WORD	ID	0x0026
0x02	4	DWORD	Size of helipad subrecord	0x0024
0x06	1	BYTE	surface (as with runway)	
0x07	1	BYTE	bit 0-3: type 0 = NONE 1 = H 2 = SQUARE 3 = CIRCLE 4 = MEDICAL bit 4: transparent bit 5: closed bit 6-7: unused	
0x08	4	BYTE[4]	color (cannot be set with bglcomp)	
0x0c	4	DWORD	longitude	
0x10	4	DWORD	latitude	
0x14	4	long	altitude * 1000	
0x18	4	float	length	
0x1c	4	float	width	
0x20	4	float	heading	

Start

(the keywords “Start” and “RunwayStart” produce identical subrecords)

offset	length	format	description	contents
0x00	2	WORD	ID	0x0011
0x02	4	DWORD	Size of start subrecord	0x0018
0x06	1	BYTE	runway number	
0x07	1	BYTE	bit 0-3:runway designator (as with runway subrecord) bit 4-7: start type 1 = RUNWAY 2 = WATER 3 = HELIPAD	
0x08	4	DWORD	longitude	
0x0c	4	DWORD	latitude	
0x10	4	long	elevation	
0x14	4	float	heading	

Com

offset	length	format	description	contents
0x00	2	WORD	ID	0x0012
0x02	4	DWORD	Size of subrecord: variable	
0x06	2	WORD	type. The following numbers have been identified: 0x0001 ATIS 0x0002 MULTICOM 0x0003 UNICOM 0x0004 CTAF 0x0005 GROUND 0x0006 TOWER 0x0007 CLEARANCE 0x0008 APPROACH 0x0009 DEPARTURE 0x000a CENTER 0x000b FSS 0x000c AWOS 0x000d ASOS 0x000e CLEARANCE_PRE_TAXI 0x000f REMOTE_CLEARANCE_DELIVERY	
0x08	4	DWORD	frequency	
0x0c	variable	STRINGZ	name	

DeleteAirport

The DeleteAirport subrecord has a fixed and a variable part. The fixed part has the following structure:

offset	length	format	description	contents
0x00	2	WORD	ID	0x0033
0x02	4	DWORD	Size of subrecord: variable	
0x06	2	WORD	delete flags BIT 0: allApproaches BIT 1: allApronLights BIT 2: allAprons BIT 3: allFrequencies BIT 4: allHelipads BIT 5: allRunways BIT 6: allStarts BIT 7: allTaxiways Bit 8: allBlastFences Bit 9: allBoundaryFences Bit 10: allJetways Bit 11: allControlTowers	
0x08	1	BYTE	number of individual runways to delete	
0x09	1	BYTE	number of individual starts to delete	
0x0a	1	BYTE	number of frequencies to delete	
0x0b	1	BYTE	unused (?)	

according to the number of individual features to delete there are the following parts of the record added:

for runways:

offset	length	format	description	contents
0x00	1	BYTE	surface (as in runway subrecord)	
0x01	1	BYTE	runway number primary	
0x02	1	BYTE	runway number secondary	
0x03	1	BYTE	bit 0-3: runway designator primary bit 4-7: runway designator secondary	

for starts:

offset	length	format	description	contents
0x00	1	BYTE	runway number	
0x01	1	BYTE	runway designator	
0x02	1	BYTE	type of start 1 = RUNWAY, 2 = WATER, 3 = HELIPAD	
0x03	1	BYTE	unused (?)	0x00

for frequencies

offset	length	format	description	contents
0	4	DWORD	bit 28-31: type (as with COM records) bit 0-27: frequency * 1000000	

Apron

There are 2 subrecords for each apron which follow each other. Both have variable length. First record:

offset	length	format	description	contents
0x00	2	WORD	ID	0x0037
0x02	4	DWORD	size	
0x06	1	BYTE	surface (as with runway subrecord)	

0x07	2	WORD	number of vertices	
			and then for each vertex:	
	4	DWORD	longitude	
	4	DWORD	latitude	
			and then	
			zero-fill to next DWORD boundary	

second record:

offset	length	format	description	contents
0x00	2	WORD	ID	0x0030
0x02	4	DWORD	size	
0x06	1	BYTE	surface (as in first record)	
0x07	1	BYTE	flags: bit 0: drawSurface bit 1: drawDetail	
0x08	2	WORD	number of vertices	
0x0c	2	WORD	number of triangles to draw	
			and then for each vertex	
	4	DWORD	longitude	
	4	DWORD	latitude	
			and then for each triangle to draw	
	2	WORD	index of first point	
	2	WORD	index of second point	
	2	WORD	index of third point	

ApronEdgeLights

offset	length	format	description	contents
0x00	2	WORD	ID	0x0031
0x02	4	DWORD	size	
0x06	2	WORD	unknown	
0x08	2	WORD	number of vertices	
0x0a	2	WORD	number of edges	
0x0c	4	DWORD	unknown, probably color of lights	0xff0000ff
0x10	4	float	unknown (value 1)	0x3f800000
0x14	4	float	unknown (value 800)	0x44480000
			and then for each vertex	
	4	DWORD	longitude	
	4	DWORD	latitude	
			end then for each edge	
	4	float	unknown (value 60.96)	
	2	WORD	index of start vertex	
	2	WORD	index of end vertex	

Fences

offset	length	format	description	contents
0x00	2	WORD	ID: BlastFence BoundaryFence	0x0038 0x0039
0x02	4	DWORD	size	
0x06	2	WORD	vertex count	
0x08	16	GUID	instanceId	
0x18	16	GUID	profile	
			and then for each vertex	
	4	DWORD	longitude	
	4	DWORD	latitude	

Unknown record

Every (?) airport in the FS X scenery files contains a subrecord with the ID of 0x3b. This record contains as usual a DWORD length field at offset 0x02. It cannot be reproduced with the BglComp compiler, and it has no apparent function. It concists of a long list of vertices along the perimeter of the airport and a list of indices for triangles to be drawn (similar to the second Apron record), but in fact the sim apparently does not use this list for drawing.

TaxiwayPoint

All taxiway points are joined in one record, which has a fixed part of 8 bytes and a variable part with 12 bytes for each point. Structure of the fixed part:

offset	length	format	description	contents
0x00	2	WORD	ID	0x001A
0x02	4	DWORD	size : variable	
0x06	2	WORD	number of taxiway points present	

and for each taxipoint:

0x00	1	BYTE	type: 1 = NORMAL, 2 = HOLD_SHORT 4 = ILS_HOLD_SHORT 5 = HOLD_SHORT_NO_DRAW 6 = ILS_HOLD_SHORT_NO_DRAW	
0x01	1	BYTE	flag: 0 = FORWARD, 1 = REVERSE	
0x02	1	WORD	unknown	0x0000
0x04	4	DWORD	longitude	
0x08	4	DWORD	latitude	

TaxiwayParking

This record type has a short fixed part for all TaxiwayParking records together and a longer variable part with sections for each TaxiwayParking. The fixed part is 8 bytes long:

offset	length	format	description	contents
0x00	2	WORD	ID	0x001B
0x02	4	DWORD	size : variable	
0x06	2	WORD	number of taxiway parking records present	

The record sections for each TaxiwayParking are again of variable length, depending on the number of airlineCodes present:.

0x00	4	DWORD	bit 31-24: count of airlineCodes present bit 23-12: number bit 11-8: type 0x1 = RAMP_GA 0x2 = RAMP_GA_SMALL 0x3 = RAMP_GA_MEDIUM 0x4 = RAMP_GA_LARGE 0x5 = RAMP_CARGO 0x6 = RAMP_MIL_CARGO 0x7 = RAMP_MIL_COMBAT 0x8 = GATE_SMALL 0x9 = GATE_MEDIUM 0xa = GATE_HEAVY 0xb = DOCK_GA bit 7-6: pushback (00 = none, 01 = left, 10 = right, 11 = both)

			bit 5-0: name 0x00 = NONE, 0x01 = PARKING, 0x02 = N_PARKING, 0x03 = NE_PARKING 0x04 = E_PARKING, 0x05 = SE_PARKING 0x06 = S_PARKING, 0x07 = SW_PARKING 0x08 = W_PARKING, 0x09 = NW_PARKING 0x0a = GATE, 0x0b = DOCK, 0x0c = GATE_A, 0x0d = GATE_B, 0x0e = GATE_C 0x0f = GATE_D 0x10 = GATE_E .. .. 0x25 = GATE_Z
0x04	4	float	radius
0x08	4	float	heading
0x0c	4	float	teeOffset1
0x10	4	float	teeOffset2
0x14	4	float	teeOffset3
0x18	4	float	teeOffset 4
0x1c	4	DWORD	longitude
0x20	4	DWORD	latitude
..	4	STRING	airline designator (0..n times repeated)

TaxiwayPath

This record has a fixed length of 8 byte and a variable part with records for each path. It has the following structure:

offset	length	format	description	contents
0x00	2	WORD	ID	0x001C
0x02	4	DWORD	size	
0x06	2	WORD	number of paths defined	
			and then for each path:	
0x00	2	WORD	index of start point NB: for type TAXI, the index of the start and of the end must both refer to a TaxiPoint. For type PARKING the start index must refer to a TaxiPoint, the end index must refer to a TaxiwayParking.	
0x02	2	WORD	Bit 0-11: index of end point	
			Bit 12-15: runway designator	
0x04	1	BYTE	type 1 = TAXI 2 = RUNWAY 3 = PARKING 4 = PATH 5 = CLOSED 6 = VEHICLE	
0x05	1	BYTE	runway number / index into TaxiName	
0x06	1	BYTE	bitfield BIT 0: centerline BIT 1: centerLineLighted BIT 2-3: leftEdge (00 = NONE, 01 = SOLID, 10 = DASHED, 11 = SOLID_DASHED) BIT 4: leftEdgeLighted BIT 5-6: rightEdge BIT 7: rightEdgeLighted	
0x07	1	BYTE	surface	
0x08	4	float	width	
0x0c	4	DWORD	weightLimit	
0x10	4	DWORD	??	

TAXINAME

This record has variable length, it consist of 8 bytes as a fixed part and then 8 bytes for each Name

offset	length	format	description	contents
0x00	2	WORD	ID	0x001D
0x02	4	DWORD	size : variable	
0x06	2	WORD	number of name entries	
			and then for each name	
	8	STRING	taxiName	

TAXIWAYSIGN

These record are coded in the section for scenery objects (0x25) with a separate type of entry. All Taxiway signs for one airport are coded together in one record. There is no apparent coordination of this record with the airport record to which it belongs. The main structure of the record is identical with that of other scenery objects

offset	length	format	description	contents
0x00	2	WORD	ID	0x000e
0x02	2	WORD	size : variable	
0x04	4	DWORD	longitude	
0x08	4	DWORD	latitude	
0x0c	4	long	altitude (?) cannot be coded with the compiler	
0x10	2	WORD	altitudeIsAGL cannot be coded	0x0001
0x12	2	WORD	pitch (?) cannot be coded	
0x14	2	WORD	bank (?) cannot be coded	
0x16	2	WORD	(heading) (?) cannot be coded	
0x18	2	WORD	imageComplexity (?) cannot be coded	
0x1a	2	WORD	unknown	
0x1c	16	GUID	instanceId (cannot be coded)	
0x2c	4	DWORD	number of taxiway signs for this airport	
			and then for each sign	
0x00	4	float	longitude offset from value in main record	
0x04	4	float	latitude offset from value in main record	
0x08	2	WORD	heading as coded	
0x0a	1	BYTE	Size (SIZE1 .. SIZE5)	
0x0b	1	BYTE	justification (1 = left, 2 = right)	
0x0c	var	STRINGZ	label (zero filled to next WORD address)	

JETWAY

offset	length	format	description	contents
0x00	2	WORD	ID	0x003a
0x02	4	DWORD	size : variable	
0x06	2	WORD	parking Number (refers to an existing parking)	
0x08	4	WORD	gate name	
0x0a	4	DWORD	unknown	
			after this follows a normal scenery object record starting with an ID of 0x0b. This record refers to an appropriate scenery object like {BFCDF52B-9142-415C-8318-03C1B92CA9D9}	

Approach

offset	length	format	description	contents
0x00	2	WORD	ID for Approach	0x0024
0x02	4	DWORD	size : variable	
0x06	1	BYTE	suffix	
0x07	1	BYTE	runway number	
0x08	1	BYTE	bit 0-3: type 0x01 = GPS 0x02 = VOR 0x03 = NDB 0x04 = ILS 0x05 = LOCALIZER 0x06 = SDF 0x07 = LDA 0x08 = VORDME 0x09 = NDBDME 0x0a = RNAV 0x0b = LOCALIZER_BACKCOURSE bit 4-6: runway designator bit 7: gpsOverlay flag	
0x09	1	BYTE	number of transitions ?	
0x0a	1	BYTE	number of approach legs	
0x0b	1	BYTE	number of missedApproach legs ?	
0x0c	4	DWORD	fixIdent BIT 0-4: fixType 02 = VOR 03 = NDB 04 = TERMINAL_NDB 05 = WAYPOINT 06 = TERMINAL_WAYPOINT 09 = RUNWAY BIT 5-31 fixIdent	
0x10	4	DWORD	bit 0-10: fixRegion bit 11-31: ICAO Id of relevant airport	
0x14	4	float	altitude	
0x18	4	float	heading	
0x1c	4	float	missedAltitude	

after this the following record can occur

offset	length	format	description	contents
0x00	2	WORD	ID for ApproachLegs	0x002D
0x02	4	DWORD	size : variable	
0x06	2	WORD	number of legs to follow	

each leg is a structure with a fixed length of 44 bytes

offset	length	format	description	contents
0x00	1	BYTE	ID of the leg types found: 0x01 = AF 0x02 = CA 0x03 = CD 0x04 = CF 0x05 = CI 0x06 = CR 0x07 = DF 0x08 = FA 0x09 = FC 0x0a = FD 0x0b = FM 0x0c = HA 0x0d = HF 0x0e = HM 0x0f = IF 0x10 = PI 0x11 = RF 0x12 = TF 0x13 = VA 0x14 = VD 0x15 = VI 0x16 = VM 0x17 = VR	
0x01	1	BYTE	altitudeDescriptor 01 = A 02 = + 03 = - 04 = B	
0x02	2	WORD	flags: bit 0: turnDirection = L bit 1: turnDirection = R	

			bit 8: magneticCourse (0) trueCourse (1) bit 9: distance (0) or time (1) bit 10: flyover false (0) true (1)	
0x04	4	DWORD	bit 5-31: fixIdent bit 0-4: fixType	
0x08	4	DWORD	bit 0-10: fixRegion bit 11-32: ICAO Id of relevant airport	
0x0c	4	DWORD	bit 5-31: recommendedIdent bit 0-4: recommendedType	
0x10	4	DWORD	recommendedRegion	
0x14	4	float	theta	
0x18	4	float	rho	
0x1c	4	float	trueCourse / magneticCourse (depending on flag)	
0x20	4	float	distance / time	
0x24	4	float	Altitude1	
0x28	4	float	Altitude2	

offset	length	format	description	contents
0x00	2	WORD	ID for missedApproachLegs	0x002E
0x02	4	DWORD	size : variable	
0x06	2	WORD	number of legs to follow	

offset	length	format	description	contents
0x00	2	WORD	ID for Transition	0x002C
0x02	4	DWORD	size : variable	
0x06	1	BYTE	transitionType 1 = FULL, 2 = DME	
0x07	1	BYTE	number of TransitionLegs (?)	
0x08	4	DWORD	bit 0-4: fixType 2 = VOR 3 = NDB 4 = TERMINAL_NDB 5 = WAYPOINT 6 = TERMINAL_WAYPOINT bit 5-31: fixIdent (spezial format)	
0x0c	4	DWORD	bit 0-10: fixRegion bit 11-31 : airportID of relevant airport	
0x10	4	float	altitude	
			if transitionType = DME and DmeArc record exists, then the following 16 bytes are present	
0x14	4	DWORD	dmeIdent	
0x18	4	DWORD	bit 0-10: dmeRegion bit 11-31: airportID of relevant airport	
0x1c	4	DWORD	radial	
0x20	4	float	distance	

offset	length	format	description	contents
0x00	2	WORD	ID for TransitionLegs (can follow only after transition	0x002F
0x02	4	DWORD	size : variable	
0x06	2	WORD	number of legs to follow	

WAYPOINT

The waypoint record can be part of the Airport group or can be entered independently. In both cases the output for the BGL is the same but for the DWORD at offset 0x18

offset	length	format	description	contents

0x00	2	WORD	ID for Waypoint	0x0022
0x02	4	DWORD	size : variable	
0x06	1	BYTE	type 1 = NAMED, 2 = UNNAMED, 3 = VOR 4 = NDB, 5 = OFF_ROUTE, 6 = IAF 7 = FAF	
0x07	1	BYTE	number of Route entries to follow	
0x08	4	DWORD	longitude	
0x0c	4	DWORD	latitude	
0x10	4	float	magvar	
0x14	4	DWORD	waypointIdent (special format)	
0x18	4	DWORD	bit 0-10: waypointRegion (special format) bi1 11-31: ICAO ident of the relevant airport, if it is a terminal waypoint, defined within an airport record	
			optional, if Route is given:	
0x1c	1	BYTE	routeType (1 = VICTOR, 2 = JET, 3 = BOTH	
0x1d	8	char[8]	name (zero padded), name cannot be longer than 8 characters	
			for Next:	
0x25	4	DWORD	BIT 0-2: type 2 = VOR, 3 = NDB, 5 = all other BIT 5-31: waypointIdent (special format)	
0x29	4	DWORD	Bit 0-10 waypointRegion (special format) BIT 11-31 airportId if terminal waypoint	
0x2d	4	float	altitudeMinimum	
			for Previous:	
0x31	4	DWORD	type + waypointIdent (as for Next)	
0x35	4	DWORD	Bit 0-10 waypointRegion (special format) BIT 11-31 airportId if terminal waypoint	
0x39	4	float	altitudeMinimum	
			Note: it is not necessary for any route to have both previous and next defined, in that case the fields for this part of the record are all zero	

ILS / VOR

The records for ILS and VOR are in the same section and they are identical for the fixed section.
ILS records can have an additional subrecord
The fixed part is 40 bytes long and has the following structure:

offset	length	format	description	contents
0x00	2	WORD	ID	0x0013
0x02	4	DWORD	size	
0x06	1	BYTE	type. The following numbers have been found: 0x0001 VOR TERMINAL 0x0002 VOR LOW 0x0003 VOR HIGH 0x0004 ILS 0x0005 VOR VOT	
0x07	1	BYTE	flags. The following bits have been recognized: bit 0: if 0 then DME only bit 2: backcourse bit 3: glideslope present bit 4: DME present bit 5: NAV true	
0x08	4	DWORD	longitude	
0x0c	4	DWORD	latitude	
0x10	4	DWORD	elevation	
0x14	4	DWORD	frequency	
0x18	4	float	range in m	
0x1c	4	float	magnetic variation	
0x20	4	DWORD	ICAO ident (special format)	
0x24	4	DWORD	bit 0-10 regionId bit 11-31 airportId (for ILS)	

The following subrecords can follow:
(for ILS)

offset	length	format	description	contents
0x00	2	WORD	ID localizer	0x0014
0x02	4	DWORD	size	0x0010
0x06	2	WORD	unknown	
0x08	4	float	heading	
0x0c	4	float	width	

:
(for ILS)

offset	length	format	description	contents
0x00	2	WORD	ID glideslope	0x0015
0x02	4	DWORD	size	0x001c
0x06	2	word	unknown	
0x08	4	DWORD	longitude	
0x0c	4	DWORD	latitude	
0x10	4	DWORD	elevation	
0x14	4	float	range	
0x18	4	float	pitch	

(for ILS/VOR)

offset	length	format	description	contents
0x00	2	WORD	ID DME	0x0016
0x02	4	DWORD	size	0x0018
0x06	2	WORD	unknown	
0x08	4	DWORD	longitude	

0x0c	4	DWORD	latitude	
0x10	4	DWORD	elevation	
0x14	4	float	range	

After these subsections, a name subsection is added:

offset	length	format	description	contents
0x00	2	WORD	ID	0x0019
0x02	4	DWORD	size	
0x06		STRING	Name (max. 48 characters)	

if VisualModel is added in the source file, the compiler adds another section to the file with a record of type 0x0025 (SceneryxObject) with the GUID for the object referenced. The coordinates for this objects are taken from the ILS/VOR and adjusted, if BiasXYZ is added to the VisualModel.

NDB

The NDB records are stored in a separate section. The have a 40 bytes long fixed section and a name section of variable length. The fixed section has the following structure:

offset	length	format	description	contents
0x00	2	WORD	ID	0x0017
0x02	4	DWORD	size variable	
0x06	2	WORD	Type 0 = COMPASS_POINT 1 = MH 2 = H 3 = HH	
0x08	4	DWORD	frequency	
0x0c	4	DWORD	longitude	
0x10	4	DWORD	latitude	
0x14	4	long	elevation	
0x18	4	float	range	
0x1c	4	float	magnetic variation	
0x20	4	DWORD	ICAO ident (special format)	
0x24	4	DWORD	bit 0-10: region bit 11-31: ICAO id of airport, if it was defined with an airport (terminal NDB)	

The name subsection has the following structure

offset	length	format	description	contents
0x00	2	WORD	ID	0x0019
0x02	4	DWORD	size	
0x06		STRING	name	

SceneryObject

LibraryObject

The record has a fixed length of 0x40 bytes with the following structure:

offset	length	format	description	contents
0x00	2	WORD	ID	0x000b
0x02	2	WORD	size	0x0040
0x04	4	DWORD	longitude	
0x08	4	DWORD	latitude	
0x0c	4	DWORD	altitude	
0x10	2	WORD	flag: 1 = isAboveAGL	
0x12	2	WORD	pitch	
0x14	2	WORD	bank	
0x16	2	WORD	heading	
0x18	2	WORD	imageComplexity 0 = VERYSPARSE 1 = SPARSE 2 = NORMAL 3 = DENSE 4 = VERYDENSE	
0x1a	2	WORD	unknown	
0x1c	16	GUID	instance ID	
0x2c	16	GUID	name	
0x3c	4	float	scale	

if an AttachedObject exists, there are three other records following:

offset	length	format	description	contents
0x00	2	WORD	ID	0x1002
0x02	2	WORD	size	0x0004
			and then 2nd record	
0x00	2	WORD	ID depending on the kind of attached object. It is possible toattach beacons, effects and other library objects	
0x02	2	WORD	size	
0x04	2	WORD	offset of attach point string	
0x06	2	WORD	pitch	
0x08	2	WORD	bank	
0x0a	2	WORD	heading	
0x0c	4	float	bias X	
0x10	4	float	bias Y	
0x14	4	float	bias Z	
0x18	16	GUID	instance ID	
0x28	2	WORD	probability	
0x2a	2	WORD	randomness	
			the following part of the record depends on the type of attached object and corresponds to the code of this type of object	
			and then the 3rd record	
0	2	WORD	ID	0x1003
2	2	WORD	size (?)	0x0004

In theory, there can be several attachments with one library object (if an adequate number of attchment points exists)

Effect

The record has a fixed part of 108 byte and a variable part. The fixed part has the following structure:

offset	length	format	description	contents
0x00	2	WORD	ID	0x000d
0x02	2	WORD	size : variable	
0x04	4	DWORD	longitude	
0x08	4	DWORD	latitude	
0x0c	4	DWORD	altitude	
0x10	2	WORD	flag: 1 = isAboveAGL	
0x12	2	WORD	pitch	
0x14	2	WORD	bank	
0x16	2	WORD	heading	
0x18	2	WORD	imageComplexity 0 = VERYSPARSE 1 = SPARSE 2 = NORMAL 3 = DENSE 4 = VERYDENSE	
0x1a	2	WORD	unknown	
0x1c	16	GUID	instance id	
0x2c	80	STRINGZ	effectName	
0x7c	variable	STRINGZ	effectParams	

GenericBuilding

offset	length	format	description	contents
0x00	2	WORD	ID	0x000a
0x02	2	WORD	size : variable	
0x04	4	DWORD	longitude	
0x08	4	DWORD	latitude	
0x0c	4	DWORD	altitude	
0x10	2	WORD	flag: 1 = isAboveAGL	
0x12	2	WORD	pitch	
0x14	2	WORD	bank	
0x16	2	WORD	heading	
0x18	2	WORD	imageComplexity 0 = VERYSPARSE 1 = SPARSE 2 = NORMAL 3 = DENSE 4 = VERYDENSE	
0x1a	2	WORD	unknown	
0x1c	16	GUID	instance id	
0x2c	4	float	scale	
0x30	2	WORD	type: 0x00a0 generic building	
0x32	2	WORD	size of record	
0x34	2	WORD	subtype. The following numbers have been identified: 0x0004 rectangular with roofType FLAT 0x0006 rectangular with roofType RIDGE 0x0007 rectangular with roofType PEAKED 0x0008 rectangular with roofType SLANT 0x0009 pyramidal building 0x000a multisidedBuilding	

for all rectangular buildings:

0x36	2	WORD	sizeX	0
0x38	2	WORD	sizeZ	1
0x3a	2	WORD	bottomTexture	2
0x3c	2	WORD	sizeBottomY	3
0x3e	2	WORD	textureIndexBottomX	4
0x40	2	WORD	textureIndexBottomZ	5
0x42	2	WORD	windowTexture	6
0x44	2	WORD	sizeWindowY	7
0x46	2	WORD	textureIndexWindowX	8

0x48	2	WORD	textureIndexWindowY	9
0x4a	2	WORD	textureIndexWindowZ	10
0x4c	2	WORD	topTexture	11
0x4e	2	WORD	sizeTopY	12
0x50	2	WORD	textureIndexTopX	13
0x52	2	WORD	textureIndexTopZ	14
0x54	2	WORD	roofTexture	15
0x56	2	WORD	textureIndexRoofX	16
0x58	2	WORD	textureIndexRoofZ	17

end for rectangular buildings with rooftype FLAT

for rectangular buildings with roofType RIDGE or SLANTED

0x5a	2	WORD	sizeRoofY	18
0x5c	2	WORD	textureIndexGableY	19
0x5e	2	WORD	gableTexture	20
0x60	2	WORD	textureIndexGableZ	21

for roofType SLANTED only

0x62	2	WORD	faceTexture	22
0x64	2	WORD	textureIndexFaceX	23
0x66	2	WORD	textureIndexFaceY	24

for rectangular buildings with roofType PEAKED

0x5a	2	WORD	sizeRoofY	18
0x5c	2	WORD	textureIndexRoofY	19

for multisided buildings:

0x36	2	WORD	buildingSides. Note: The Argument for smoothing is required by the compiler, but it has no effect on the BGL-file	0
0x38	2	WORD	sizeX	1
0x3a	2	WORD	sizeZ	2
0x3c	2	WORD	bottomTexture	3
0x3e	2	WORD	sizeBottomY	4
0x40	2	WORD	textureIndexBottomX	5
0x42	2	WORD	windowTexture	6
0x44	2	WORD	sizeWindowY	7
0x46	2	WORD	textureIndexWindoxX	8
0x48	2	WORD	textureIndexWindowY	9
0x4a	2	WORD	topTexture	10
0x4c	2	WORD	sizeTopY	11
0x4e	2	WORD	textureIndexTopX	12
0x50	2	WORD	roofTexture	13
0x52	2	WORD	sizeRoofY	14
0x54	2	WORD	textureIndexRoofX	15
0x56	2	WORD	textureIndexRoofZ	16
			Note: textureIndexRoofY is required by the compiler, but it has no effect on the bgl file !	

for pyramidal buildings

0x36	2	WORD	sizeX	0
0x38	2	WORD	sizeZ	1
0x3a	2	WORD	sizeTopX	2
0x3c	2	WORD	sizeTopZ	3
0x3e	2	WORD	bottomTexture	4
0x40	2	WORD	sizeBottomY	5
0x42	2	WORD	textureIndexBottomX	6
0x44	2	WORD	textureIndexBottomZ	7

0x46	2	WORD	windowTexture	8
0x48	2	WORD	sizeWindowY	9
0x4a	2	WORD	textureIndexWindowX	10
0x4c	2	WORD	textureIndexWindowY	11
0x4e	2	WORD	textureIndexWindowZ	12
0x50	2	WORD	topTexture	13
0x52	2	WORD	sizeTopY	14
0x54	2	WORD	textureIndexTopX	15
0x56	2	WORD	textureIndexTopZ	16
0x58	2	WORD	roofTexture	17
0x5a	2	WORD	textureIndexRoofX	18
0x5c	2	WORD	textureIndexRoofZ	19

WINDSOCK

Record with fixed length of 46 byte

offset	length	format	description	contents
0x00	2	WORD	ID	0x000c
0x02	2	WORD	size	0x003e
0x04	4	DWORD	longitude	
0x08	4	DWORD	latitude	
0x0c	4	long	altitude	
0x10	2	WORD	flags (unused)	
0x12	2	WORD	pitch	
0x14	2	WORD	bank	
0x16	2	WORD	heading	
0x18	2	WORD	imageComplexity	
0x1a	2	WORD	unknown	
0x1c	16	GUID	instance id	
0x2c	4	float	poleHeight	
0x30	4	float	sockLength	
0x34	1	BYTE	PoleColor: blue	
0x35	1	BYTE	PoleColor:green	
0x36	1	BYTE	PoleColor:red	
0x37	1	BYTE	PoleColor ?	0xff
0x38	4	BYTE[4]	SockColor	
0x3c	2	WORD	flag: lighted (TRUE = 0x0001)	

EXTRUSION BRIDGE

offset	length	format	description	contents
0x00	2	WORD	ID	0x0012
0x02	2	WORD	size	
0x04	4	DWORD	longitude	
0x08	4	long	latitude	
0x0c	4	DWORD	altitude	
0x10	2	WORD	flags	
0x12	2	WORD	pitch	
0x14	2	WORD	bank	
0x16	2	WORD	heading	
0x18	2	WORD	imageComplexity	
0x1a	2	WORD	unknown	
0x1c	16	GUID	instance id	
0x2c	16	GUID	profile	
0x3c	16	GUID	material set	
0x4c	12	DWORD[3]	altitude sample location 1	
0x58	12	DWORD[3]	altitude sample location 2	

0x64	4	float	road width	
0x68	4	float	probability	
0x6c	1	BYTE	suppress	
0x6d	1	BYTE	placement count	
0x6e	2	WORD	point count	
			and then for each polyline objct placement	
	16	GUID	placement id	
			and then for each polyline point	
	4	DWORD	longitude	
	4	DWORD	latitude	
	4	long	elevation	

TRIGGER

The record consists of a fixed part and a variable part. The fixed part is 34 byte long and has the following structure:

offset	length	format	description	contents
0x00	2	WORD	ID	0x0010
0x02	2	WORD	size : variable	
0x04	4	DWORD	longitude	
0x08	4	DWORD	latitude	
0x0c	4	DWORD	altitude	
0x10	2	WORD	altitudeIsAGL (0x00001 = TRUE)	
0x12	2	WORD	pitch	
0x14	2	WORD	bank	
0x16	2	WORD	heading	
0x18	2	WORD	imageComplexity	
0x1a	2	WORD	unknown	
0c1c	16	GUID	instance id	
0x2c	2	WORD	type (0x0000 = REFUEL_REPAIR, 0x0001 = WEATHER	
0x2e	4	float	triggerHeight	
in case of WEATHER the variable part has the following structure				
0x32	2	WORD	type 0x0001 = RIDGE_LIFT 0x0002 = UNIDIRECTIONAL_TURBULENCE note: in bglcomp.xsd this keyword is spelled NONDIRECTIONAL_TURBULENCE, but the compiles does not understand it. If you change the keyword in bglcomp.xsd compilation is ok. 0x0003 = DIRECTIONAL_TURBULENCE 0x0004 = THERMAL	
0x34	4	float	heading	
0x38	4	float	scalar	
0x3c	4	DWORD	number of vertices	
			and then for each vertex:	
	4	float	BiasX	
	4	float	BiasZ	
in case of FUEL_REPAIR the variable part has the following structure				
0x32	4	DWORD	fuel type and availability BITFIELD: bit 0-1: type 73 bit 2-3: type 87 bit 4-5: type 100 bit 6-7: type 130 bit 8-9: type 145 bit 10-11: type MOGAS bit 12-13: type JET	

			bit 14-15: type JETA bit 16-17: type JETA1 bit 18-19: type JETAP bit 20-21: type JETB bit 22-23: type JET4 bit 24-25: type JET5 bit 26-29 : unused bit 30 : piston type bit 31 : jet type for all except last two : 0 = NO; 1 = UNKNOWN; 2 = PRIOR_REQUEST; 3 = YES when type=UNKNOWN and availability = YES then type=100 and type = JETA both are set to availability=YES	
0x36	4	DWORD	number of vertices	
			and then for each vertex	
	4	float	BiasX	
	4	float	BiasZ	

Marker

The marker record has a fixed length of 28 byte with the following structure:

offset	length	format	description	contents
0x00	2	WORD	ID	0x0018
0x02	4	DWORD	size	0x0000001c
0x06	1	BYTE	heading	
0x07	1	BYTE	Type 0 = INNER; 1 = MIDDLE; 2 = OUTER 3 = BACKCOURSE	
0x08	4	DWORD	longitude	
0x0c	4	DWORD	latitude	
0x10	4	DWORD	altitude	
0x14	4	DWORD	ident (special format)	
0x18	2	WORD	region (special format)	
0x1a	2	word	unknown	0x0000

Boundary

offset	length	format	description	contents
0	2	WORD	ID	0x0020
2	4	DWORD	size : varying	
6	1	BYTE	type 00 = NONE 01 = CENTER 02 = CLASS_A 03 = CLASS_B 04 = CLASS_C 05 = CLASS_D 06 = CLASS_E 07 = CLASS_F 08 = CLASS_G 09 = TOWER 0a = CLEARANCE 0b = GROUND 0c = DEPARTURE 0d = APPROACH 0e = MOA 0f = RESTRICTED 10 = PROHIBITED 11 = WARNING 12 = ALERT 13 = DANGER 14 = NATIONAL_PARK 15 = MODEC 16 = RADAR 17 = TRAINING	
7	1	BYTE	BIT 0-3: maximumAltitudeType BIT 4-7: minimumAltitudeType 1 = MEAN_SEA_LEVEL (= UNKNOWN) 2 = ABOVE_GROUND_LEVEL 3 = UNLIMITED	
8	4	DWORD	minimum longitude of area covered	
12	4	DWORD	minimum latitude of area covered	
16	4	DWORD	minimumAltitude * 1000	
20	4	DWORD	maximum longitude of area covered	
24	4	DWORD	maximum latitude of area covered	
28	4	DWORD	maximumAltitude	
32	2	WORD	type field of name record	0x19
34	4	DWORD	size of name record	
36	size-6	STRING	name	

on this follows a record describing the drawing of the lines

offset	length	format	description	contents
0	2	WORD	ID	0x0021
2	4	DWORD	size : varying	
6	2	WORD	number of points to follow	
			for each point 10 bytes	
0	2	WORD	type of point 1 = START 2 = LINE 3 = ORIGIN 4 = ARC clockwise 5 = arc counter-clockwise 6 = circle NB: in case of circle, the entries for minimumAltitude and maximumAltitude override the values in start if both are given. the start entry is in case of circle not needed at all Note: there is a bug in the new version of bglcomp.xsd: the word BoundaryStart in grpBoundaryChildren has to be replaced by Start, otherwise the compiler does not accept it!	
2	4	DWORD	latitude of point (in case of circle: unknown, = 0x0000)	
6	4	DWORD	longitude of point (in case of circle:	

			float: radius	

Geopol

fixed part:

offset	length	format	description	contents
0	2	WORD	ID	0x0023
2	4	DWORD	size : varying	
6	2	WORD	Bit 0-13: number of vertices number of vertices BIT 14-15: type (0x40 = BOUNDARY, 0x80 = COASTLINE)	
8	4	DWORD	minimum longitude	
12	4	DWORD	minimum latitude	
16	4	DWORD	maximum longitude	
20	4	DWORD	maximum latitude	

variable part: for each vertex

0	4	DWORD	longitude	
4	4	DWORD	latitude	

Model data

The model data structure has a fixed length of 24 bytes

offset	length	format	description	contents
0	16	GUID	name	
16	4	DWORD	mdl file offset from the start of this subsection	
20	4	DWORD	mdl file length	

ExclusionRectangle

This record has a fixed length record of 20 bytes

offset	length	format	description	contents
0	2	WORD	exclusion type 0x0008 = excludeAll otherwise: bit 4 = BeaconObjects bit 5 = EffectObjects bit 6 = GenericBuildingObjects bit 7 = LibraryObjects bit 8 = TaxiwaySignObjects bit 9 = TriggerObjects bit 10 = WindsockObjects bit 11 = ExtrusionBridges	
2	2	WORD	size (unused)	0x0000
4	4	DWORD	longitude of NW corner	
8	4	DWORD	latitude of NW corner	
12	4	DWORD	longitude of SE corner	
16	4	DWORD	latitude of SE corner	

Namelist

The namelist contains only one record of variable length. It consists of a fixed part and a variable part. The fixed part is 42 bytes long and has the following structure:

offset	length	format	description	contents
0x00	2	WORD	ID	0x0027
0x02	4	DWORD	size (?) seems always to be 0x00000000	
0x06	2	WORD	number of region names	
0x08	2	WORD	number of country names	
0x0a	2	WORD	number of state names	
0x0c	2	WORD	number of city names	
0x0e	2	WORD	number of airport names	
0x10	2	WORD	number of ICAO ident.	
0x12	4	DWORD	offset of region list (from start of record)	
0x16	4	DWORD	offset of country list	
0x1a	4	DWORD	offset of state list	
0x1e	4	DWORD	offset of city list	
0x22	4	DWORD	offset of airport list	
0x26	4	DWORD	offset of ICAO ident list	

The lists for region, country, state, city and airport names have all the same structure:

```
an index with 1 DWORD for each entry in the list, containing the offset of the
nth name from the beginning of the names part (i.e. after the index)
followed by the names in form of zero-terminated strings
```

The ICAO list has a different structure. It contains n entries (one for each ICAO name), each of them 20 bytes long, with the following structure;:

offset	length	format	description
0x00	1	BYTE	region name index (all indexes start with 0 for the first name in the relevant list)
0x01	1	BYTE	country name index
0x02	2	WORD	bit 4-15 : state name index bit 0-3 : unknown
0x04	2	WORD	city name index
0x06	2	WORD	airport name index
0x08	4	DWORD	ICAO identifier (special format)
0x0c	4	DWORD	unknown
0x10	4	DWORD	unknown

Vector data in cvx files

The cvx files contain terrain data in vetor format which can be produced with the Shp2Vec program. They are organized as the other bgl files, i.e. the data are contained in records grouped according to the QMID region they belong to. The vector data in the single records (or subsections) are compressed. Since the SDK does not contain a program which would allow to obtain the shape files back from the BGL files and since Microsoft has not published the algorithm with which the compression is achieved, it is at present impossible to analyze the internal structire of the cvx files. The only way to obtain information about their contents is to load them into the TmfViewer program provided with the SDK.