

Contents

1	Transformations	1
1.1	wkb2geos	1
1.2	geos2wkb	1
1.3	wkbFROMSTR	2
1.4	wkbTOSTR	2
1.5	wkbFromText	2
2	Basic Methods on geometric objects (OGC)	3
2.1	wkbDimension	3
2.2	wkbGeometryType	3
2.3	wkbGetSRID	3
2.4	Envelope	3
2.5	wkbAsText	3
2.6	AsBinary	3
2.7	wkbIsEmpty	3
2.8	wkbIsSimple	3
2.9	Is3D	3
2.10	IsMeasured	3
2.11	wkbBoundary	3
3	Notes on sql level	3
3.1	int atom_cast(atom *a, sql_subtype *tp)	3

1 Transformations

1.1 wkb2geos

GEOSGeom wkb2geos(wkb* geomWKB)

The functions gets a pointer to a wkb struct and uses GEOS library to create a GEOSGeometry. In case `geomWKB` is **NULL** `wkb2geos` returns **NULL**.

1.2 geos2wkb

wkb* geos2wkb(const GEOSGeometry* geosGeometry)

The function gets a pointer to a GEOSGeometry and uses GEOS library to create a wkb struct. In case `geosGeometry` is **NULL**, `malloc` cannot reserve the needed memory or GEOS `GEOSGeomToWKB_buf` function returns **NULL** the function also returns a pointer to a **NULL wkb struct**.

1.3 wkbFROMSTR

```
int wkbFROMSTR(char* geomWKT, int* len, wkb **geomWKB, int srid)
```

The function is used when reading data from disk. It gets a WKT geometry representation `geomWKT` and the `srid` of it and creates the wkb struct `geomWKB`. In case `geomWKT` is `str_nil` or there is some error when creating the GEOSGeometry or the wkb struct from it the function creates a `wkb_nil`. It returns the number of characters in `geomWKT` or 0 if an error occurred. `*len` is set to the size of the `geomWKB` but I am not sure why this is needed. The function returns the size of the created `geomWKT` and it is important in order to return the correct value.

1.4 wkbTOSTR

```
int wkbTOSTR(char **geomWKT, int* len, wkb *geomWKB)
```

The functions is used when writing data to disk. It gets a wkb struct `geomWKB` and creates the WKT representation of it `geomWKT`. If an error occurs when creating the GEOSGeometry from the `geomWKB` or when creating the WKT from the GEOSGeometry "nil" is returned as `geomWKT`. `*len` has the length of `geomWKT` including quotes and `'\0'` character, and is very important for the correct storage of the data.

1.5 wkbFromText

```
str wkbFromText(wkb **geomWKB, str *geomWKT, int* srid, int *tpe)
```

The function uses the `wkbFROMSTR` to create a `geomWKB` out of `geomWKT`. The function is used by the SQL functions

- `ST_GeomFromText`
- `ST_GeometryFromText`
- `ST_PointFromText`
- `ST_LineFromText`
- `ST_PolygonFromText`
- `ST_MPointFromText`
- `ST_MLineFromText`
- `ST_MPolyFromText`
- `ST_GeomCollFromText`

`*tpe` is used to indicate the geometry type that should be created. Although all SQL functions use the same `wkbFromText` c function a check is performed whether the `geomWKT` created a `geomWKB` of the same type as `*tpe`.

2 Basic Methods on geometric objects (OGC)

2.1 wkbDimension

```
str wkbDimension(int *dimension, wkb **geomWKB)
```

2.2 wkbGeometryType

```
str wkbGeometryType(char** out, wkb** geomWKB, int* flag)
```

2.3 wkbGetSRID

```
str wkbGetSRID(int *out, wkb **geomWKB)
```

2.4 Envelope

2.5 wkbAsText

```
str wkbAsText(str *txt, wkb **geomWKB)
```

The function uses the `wkbFROMSTR` to create a `geomWKT` from `geomWKB`. The function is used by the SQL function `ST_AsText`.

2.6 AsBinary

2.7 wkbIsEmpty

```
str wkbIsEmpty(bit *out, wkb **geomWKB)
```

2.8 wkbIsSimple

```
str wkbIsSimple(bit *out, wkb **geomWKB)
```

2.9 Is3D

2.10 IsMeasured

2.11 wkbBoundary

```
str wkbBoundary(wkb **boundaryWKB, wkb **geomWKB)
```

3 Notes on sql level

3.1 int atom_cast(atom *a, sql_subtype *tp)

It is called to check the arguments of a function (once for each argument) `sql_subtype *tp` has the type of the expected argument and `sql_subtype *at = &a->tp` has the type of the provided argument.