QGIS Application - Bug report #14204
QgsGeometry::fromWkb fails if WKB is different endian representation

2016-01-31 11:38 AM - David Adler

Status:	Closed		
Priority:	High		
Assignee:			
Category:	Geometry		
Affected QGIS version:	master	**Regression?:**	No
Operating System:	Windows	**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	fixed/implemented
Crashes QGIS or corrupts data:	Yes	**Copied to github as #:**	22206

Description

Geometry creation fails when processing WKB from an IBM DB2 z/OS database which uses big-endian on an Windows system with a little-endian architecture.

Some of the QGIS geometry code is set up to handle the difference but it does not seem to be working correctly.

The first byte of the WKB is x'01' for little-endian and x'00' for big-endian. The next 4 bytes are an integer representing the WKB type (point, line, etc) per the WKB specification.

The first problem arises in QgsGeometryFactory::geomFromWkb which uses the following logic to get the WKB type:

```
    int type;
    memcpy( &type, wkb + 1, sizeof( int ) );
```

which just grabs 4 bytes without taking consideration of the endian-ness.
This can be handled correctly by using QgsConstWkbPtr::readHeader() which returns the WKB type in the WKB header and (should) handle the endian-ness.

```
   QgsWKBTypes::Type wkbType = wkbPtr.readHeader();
```

However, there is a problem in QgsConstWkbPtr::readHeader() which corrupts the WKB type in the logic:

```
    ( *this ) >> wkbType;
    if ( mEndianSwap )
    {
      QgsApplication::endian_swap( wkbType );
    }
```

The >> operator handles the endian-ness and swaps the bytes appropriately when setting wkbType here.
However, the if statement checks mEndianSwap and swaps the bytes back again to the wrong order.
Just taking out the if statement seems to fix the problem.

There is another major problem. The original wkb is saved in QgsGeometry which is later accessed in numerous places where the endian-ness is not checked. (In particular, the drawing simplification logic).
I think the solution is to delete the original wkb and re-create it in QgsGeometry::fromWKB() as follows:

```
   void QgsGeometry::fromWkb( unsigned char *wkb, int length )
```

```
{
  Q_UNUSED( length );

  detach( false );

  if ( d->geometry )
  {
    delete d->geometry;
    removeWkbGeos();
  }
  d->geometry = QgsGeometryFactory::geomFromWkb( wkb );
  if  ( *wkb != QgsApplication::endian() ) // rebuild wkb if different endian
  {
    delete wkb;
    d->mWkb = d->geometry->asWkb( d->mWkbSize );
  } else
  {
    d->mWkb = wkb;
  }
  d->mWkbSize = length;
}
```

History

#1 - 2016-02-01 02:28 AM - Jürgen Fischer

Could you check if https://github.com/qgis/QGIS/pull/2748 helps with this issue?

#2 - 2016-02-01 03:39 AM - Sandro Santilli

- Tag set to wkb

See also #14182

#3 - 2016-02-11 09:08 AM - Jürgen Fischer

- Resolution set to fixed/implemented

- Status changed from Open to Closed

fixed in commit:b9726d7