

QGIS Application - Bug report #474

crash zooming to a raster not matching the project's projection

2006-12-21 11:43 AM - Redmine Admin

Status: Closed	
Priority: Low	
Assignee: Gavin Macaulay -	
Category: Projection Support	
Affected QGIS version:	Regression?: No
Operating System: Debian	Easy fix?: No
Pull Request or Patch supplied:	Resolution: fixed
Crashes QGIS or corrupts data:	Copied to github as #: 10533

Description

QGIS SVN 6248:

1. added few shapefiles of unknown projection
2. set the projection to EPSG:4179, enabled on the fly
3. added a raster of a different projection, namely:

```
Driver: GTiff/GeoTIFF
Size is 3800, 2500
Coordinate System is:
PROJCS["TNTmips Georeference",
  GEOGCS["WGS 84",
    DATUM["WGS_1984",
      SPHEROID["WGS 84",6378137,298.2572235629972,
        AUTHORITY["EPSG""7030]],
      AUTHORITY["EPSG""6326]],
    PRIMEM["Greenwich"],
    UNIT["degree"],
    AUTHORITY["EPSG""4326]],
  PROJECTION["Transverse_Mercator"],
  PARAMETER["latitude_of_origin"],
  PARAMETER["central_meridian"],
  PARAMETER["scale_factor"],
  PARAMETER["false_easting"],
  PARAMETER["false_northing"],
  UNIT["metre",1,
    AUTHORITY["EPSG""9001]]
```

4. zoom to raster -> crash; the bt:

```
Program received signal SIGSEGV, Segmentation fault.
[Switching to Thread -1242003776 (LWP 19199)]
0xb657c9d5 in memcpy () from /lib/tls/i686/cmov/libc.so.6
(gdb) bt
#0 0xb657c9d5 in memcpy () from /lib/tls/i686/cmov/libc.so.6
#3905 0xb6dc34e2 in GDALRasterBand::IRasterIO ()
from /usr/local/lib/libgdal.so.1
```

```

#3906 0xb6dbaca4 in GDALRasterBand::RasterIO () from /usr/local/lib/libgdal.so.1
#3907 0xb7ed485d in [[QgsRasterLayer]::readData (this=0x83a4df8,
    gdalBand=0x833d820, viewPort=0x83b0cc8) at qgsrasterlayer.cpp:4434
#3908 0xb7edc8f5 in [[QgsRasterLayer]::drawPalettedMultiBandColor (this=0x83a4df8,
    theQPainter=0x228, theRasterViewPort=0x83b0cc8,
    theQgsMapToPixel=0x81e5b60, theBandNoInt=1) at qgsrasterlayer.cpp:2264
#3909 0xb7ee7ddb in [[QgsRasterLayer]::draw (this=0x83a4df8,
    theQPainter=0xbfa12488, theRasterViewPort=0x83b0cc8,
    theQgsMapToPixel=0x81e5b60) at qgsrasterlayer.cpp:1465
#3910 0xb7ef419c in [[QgsRasterLayer]::draw (this=0x83a4df8,
    theQPainter=0xbfa12488, theViewExtent=0xbfa12348,
    theQgsMapToPixel=0x81e5b60, drawingToEditingCanvas=true)
    at qgsrasterlayer.cpp:1377
#3911 0xb68c42f4 in [[QgsMapRender]::render (this=0x81f9470, painter=0xbfa12488)
    at qgsmaprender.cpp:261
#3912 0xb68a64fa in [[QgsMapCanvasMap]::render (this=0x81f63f0)
    at qgsmapcanvasmap.cpp:81
#3913 0xb68a22ab in [[QgsMapCanvas]::render (this=0x81bfb88) at qgsmapcanvas.cpp:305
#10 0xb7eb78ab in [[QgsLegend]::zoomToLayerExtent (this=0x82ea208)
    at qgslegend.cpp:1745
#3914 0xb7ebf04d in [[QgsLegend]::qt_metacall (this=0x82ea208,
    _c=QMetaObject::InvokeMetaMethod, _id=23, _a=0xbfa12b8c)
    at qgslegend.moc.cpp:131
#3915 0xb7d9f0a2 in QMetaObject::activate (sender=0x83c5b08,
---Type <return> to continue, or q <return> to quit---
    from_signal_index=5, to_signal_index=6, argv=0x228)
    at kernel/qobject.cpp:2940
#3916 0xb7d9f379 in QMetaObject::activate (sender=0x228, m=0x228,
    from_local_signal_index=1, to_local_signal_index=2, argv=0x780)
    at kernel/qobject.cpp:2992
#3917 0xb7405e09 in QAction::triggered (this=0x228, _t1=false)
    at moc_qaction.cpp:208
#3918 0xb7406a6c in QAction::activate (this=0x83c5b08, event=QAction::Trigger)
    at kernel/qaction.cpp:1070
#3919 0xb7706731 in QMenuPrivate::activateAction (this=0x83e40c8,
    action=0x83c5b08, action_e=QAction::Trigger) at widgets/qmenu.cpp:751
#3920 0xb77076e0 in QMenu::mousePressEvent (this=0xbfa13828, e=0xbfa13254)
    at widgets/qmenu.cpp:1828
#3921 0xb744fa20 in QWidget::event (this=0xbfa13828, event=0xbfa13254)
    at kernel/qwidget.cpp:5698
#3922 0xb770379b in QMenu::event (this=0xbfa13828, e=0xbfa13254)
    at widgets/qmenu.cpp:1896
#3923 0xb740bad3 in QApplicationPrivate::notify_helper (this=0x80661a8,
    receiver=0xbfa13828, e=0xbfa13254) at kernel/qapplication.cpp:3434
#3924 0xb740ee4d in QApplication::notify (this=0xbfa14798, receiver=0xbfa13828,
    e=0xbfa13254) at kernel/qapplication.cpp:3133
#3925 0xb746636c in QETWidget::translateMouseEvent (this=0xbfa13828,
    event=0xbfa134ec) at qcoreapplication.h:186
#3926 0xb7464ed8 in QApplication::x11ProcessEvent (this=0xbfa14798,
    event=0xbfa134ec) at kernel/qapplication_x11.cpp:2764
#3927 0xb74883e3 in x11EventSourceDispatch (s=0x8086450, callback=0,
    user_data=0x228) at kernel/qguieventdispatcher_glib.cpp:122
---Type <return> to continue, or q <return> to quit---

```

#3928 0xb649b8d6 in g_main_context_dispatch () from /usr/lib/libglib-2.0.so.0
#3929 0xb649e996 in g_main_context_check () from /usr/lib/libglib-2.0.so.0
#3930 0xb649ee1e in g_main_context_iteration () from /usr/lib/libglib-2.0.so.0
#3931 0xb7db0e11 in QEventDispatcherGlib::processEvents (this=0x8084b90, flags=@0xbfa136d8) at kernel/qeventdispatcher_glib.cpp:363
#3932 0xb7488131 in QGuiEventDispatcherGlib::processEvents (this=0x8084b90, flags=@0x0) at kernel/qguieventdispatcher_glib.cpp:178
#3933 0xb7d8e3e3 in QEventLoop::processEvents (this=0x228, flags=@0x228) at kernel/qeventloop.cpp:126
#3934 0xb7d8e522 in QEventLoop::exec (this=0xbfa1376c, flags=@0xbfa13778) at kernel/qeventloop.cpp:172
#3935 0xb7707953 in QMenu::exec (this=0xbfa13828, p=@0x228, action=0x0) at widgets/qmenu.cpp:1601
#3936 0xb7eb193a in [[QgsLegend]]::handleRightClickEvent (this=0x82ea208, item=0x840e6b8, position=@0xbfa1414c) at qgslegend.cpp:473
#3937 0xb7eb2e59 in [[QgsLegend]]::mousePressEvent (this=0x82ea208, e=0xbfa14134) at qgslegend.cpp:178
#3938 0xb744f8da in QWidget::event (this=0x82ea208, event=0xbfa14134) at kernel/qwidget.cpp:5694
#36 0xb76d7674 in QFrame::event (this=0x228, e=0x228) at widgets/qframe.cpp:633
#37 0xb7747fd5 in QAbstractScrollArea::viewportEvent (this=0x0, e=0x780) at widgets/qabstractscrollarea.cpp:854
#38 0xb77a4900 in QAbstractItemView::viewportEvent (this=0x82ea208, event=0xbfa14134) at itemviews/qabstractitemview.cpp:1277
#39 0xb774a964 in QAbstractScrollAreaFilter::eventFilter (this=0x228, o=0x82e6538, e=0x228) at qabstractscrollarea_p.h:78
#40 0xb740ba87 in QApplicationPrivate::notify_helper (this=0x80661a8, ---Type <return> to continue, or q <return> to quit--- receiver=0x82e6538, e=0xbfa14134) at kernel/qapplication.cpp:3426
#41 0xb740ee4d in QApplication::notify (this=0xbfa14798, receiver=0x82e6538, e=0xbfa14134) at kernel/qapplication.cpp:3133
#42 0xb7465582 in QETWidget::translateMouseEvent (this=0x82e6538, event=0xbfa143cc) at qcoreapplication.h:186
#43 0xb7464ed8 in QApplication::x11ProcessEvent (this=0xbfa14798, event=0xbfa143cc) at kernel/qapplication_x11.cpp:2764
#44 0xb74883e3 in x11EventSourceDispatch (s=0x8086450, callback=0, user_data=0x228) at kernel/qguieventdispatcher_glib.cpp:122
#45 0xb649b8d6 in g_main_context_dispatch () from /usr/lib/libglib-2.0.so.0
#46 0xb649e996 in g_main_context_check () from /usr/lib/libglib-2.0.so.0
#47 0xb649ee1e in g_main_context_iteration () from /usr/lib/libglib-2.0.so.0
#48 0xb7db0e11 in QEventDispatcherGlib::processEvents (this=0x8084b90, flags=@0xbfa145b8) at kernel/qeventdispatcher_glib.cpp:363
#49 0xb7488131 in QGuiEventDispatcherGlib::processEvents (this=0x8084b90, flags=@0x0) at kernel/qguieventdispatcher_glib.cpp:178
#50 0xb7d8e3e3 in QEventLoop::processEvents (this=0x228, flags=@0x228) at kernel/qeventloop.cpp:126
#51 0xb7d8e522 in QEventLoop::exec (this=0xbfa14650, flags=@0xbfa14658) at kernel/qeventloop.cpp:172
#52 0xb7d9090b in QCoreApplication::exec () at kernel/qcoreapplication.cpp:727
#53 0xb740b64f in QApplication::exec () at kernel/qapplication.cpp:2927
#54 0x0804bf3f in main (argc=1, argv=0xbfa14954) at main.cpp:608

I know the projection I declared and raster's projection don't match, but this I learned in the process and QGIS shouldn't crash in the first place.

Maciek

History

#1 - 2006-12-28 11:57 AM - Magnus Homann

Is it possible to attach raster and/or the dataset?

#2 - 2006-12-28 01:45 PM - Redmine Admin

Replying to [comment:1 homann]:

| *Is it possible to attach raster and/or the dataset?*

Sorry I don't have it anymore. And I can't reproduce with other dataset at will. The procedure in general should be:

1. add a shapfile without *.prj
2. set some projection, enable on the fly
3. add a [[GeoTIFF]] with a projection different than set in step 2.
4. zoom to raster -> crash

Maciek

#3 - 2006-12-28 10:11 PM - Gary Sherman

this can be duplicated by loading any shapefile with a projection different from the raster, setting the projection to a third projection, then adding the raster.

Zoom to raster causes crash.

#4 - 2006-12-29 01:07 AM - anonymous -

Partial fix in svn commit:546b8e21 (SVN r6342). Qgis shouldn't crash now, but it doesn't quite yet zoom to layer extent correctly for all combinations. Note that the zoom to layer extent button in the toolbar does work correctly - the zoom to layer in the legend is the problem

#5 - 2006-12-29 12:24 PM - Gavin Macaulay -

- Status changed from Open to Closed
- Resolution set to fixed

Fixed in svn commit:c226ee18 (SVN r6346). No longer crashes. Better behaviour when projecting between incompatible coordinate systems could be an enhancement for later versions of qgis.

#6 - 2009-08-22 12:46 AM - Anonymous

Milestone Version 0.8 deleted