

QGIS Application - Bug report #17351

Link failure on OS without thread local storage

2017-10-29 12:46 AM - landry Landry Breuil

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	Nyall Dawson	
<b>Category:</b>	Projection Support	
<b>Affected QGIS version:</b>	master	<b>Regression?:</b> Yes
<b>Operating System:</b>	OpenBSD	<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b>	No	<b>Resolution:</b>
<b>Crashes QGIS or corrupts data:</b>	No	<b>Copied to github as #:</b> 25248
<b>Description</b>		
OpenBSD doesnt have support (yet) for thread local storage, so since <a href="https://github.com/qgis/QGIS/pull/4561">https://github.com/qgis/QGIS/pull/4561</a> the linking will fail.		
Is there an alternative to use thread_local here ?		

Associated revisions

Revision 265be41d - 2017-11-06 12:20 AM - Nyall Dawson

Don't use thread\_local on mingw or OpenBSD builds

MingW has broken support for thread\_local, so force disabling it  
see

<https://sourceforge.net/p/mingw-w64/bugs/445/>  
<https://sourceforge.net/p/mingw-w64/bugs/527/>  
[https://gcc.gnu.org/bugzilla/show\\_bug.cgi?id=80816](https://gcc.gnu.org/bugzilla/show_bug.cgi?id=80816)

also OpenBSD has no thread\_local support, see <https://issues.qgis.org/issues/17351>

So on these platforms we fall back to using QThreadStorage.

Fixes #17351

History

#1 - 2017-10-29 08:39 AM - Nyall Dawson

You may have luck by ifdefing in QThreadStorage for bsd. This is what's used in the 2.18 backport - see  
commit:af3370d03e7e7502e8a738d99cd99e38df23768d

#2 - 2017-10-29 06:20 PM - landry Landry Breuil

I have a bit of a hard time diffing the commits in 2.18 and master for that feature - you mean just replacing thread\_local by QThreadStorage ? ie  
thread\_local QgsProjContextStore replaced by QThreadStorage< QgsProjContextStore\* > ?

#3 - 2017-10-29 11:58 PM - Nyall Dawson

I can't recall if that was the only change required, but I'd start with that and see if it compiles OK and the transform related unit tests pass.

#### #4 - 2017-10-30 05:38 PM - landry Landry Breuil

I probably need a bit more:

```
- static thread_local QgsProjContextStore mProjContext;  
+ static QThreadStorage<QgsProjContextStore* > mProjContext;
```

Yields:

```
src/core/qgscoordinatetransform_p.cpp:188:139: error: no member named 'get' in 'QThreadStorage<QgsProjContextStore *>'  
    QMap < uintptr_t, QPair< projPJ, projPJ > ::const_iterator it = mProjProjections.constFind( reinterpret_cast< uintptr_t>( mProjContext.get() ) );
```

I'll try to look a bit more at the 2.18 code.

Edit: hah, seems QThreadStorage needs to wrap the calls into .localData() to access the stored pointer. Right ?

#### #5 - 2017-10-30 11:20 PM - landry Landry Breuil

Alas, with the following blind patch

```
Index: src/core/qgscoordinatetransform_p.h  
--- src/core/qgscoordinatetransform_p.h.orig  
+++ src/core/qgscoordinatetransform_p.h  
@@ -31,6 +31,7 @@  
//  
  
#include <QSharedData>  
+#include <QThreadStorage>  
#include "qgscoordinatereferencesystem.h"  
  
typedef void *projPJ;  
@@ -100,7 +101,7 @@ class QgsCoordinateTransformPrivate : public QSharedDa  
    * Thread local proj context storage. A new proj context will be created  
    * for every thread.  
    */  
- static thread_local QgsProjContextStore mProjContext;  
+ static QThreadStorage<QgsProjContextStore* > mProjContext;  
  
    QReadWriteLock mProjLock;  
    QMap < uintptr_t, QPair< projPJ, projPJ > > mProjProjections;  
  
Index: src/core/qgscoordinatetransform_p.cpp  
--- src/core/qgscoordinatetransform_p.cpp.orig  
+++ src/core/qgscoordinatetransform_p.cpp  
@@ -29,7 +29,7 @@ extern "C"
```

```
/// @cond PRIVATE
```

```
-thread_local QgsProjContextStore QgsCoordinateTransformPrivate::mProjContext;
+QThreadStorage< QgsProjContextStore* > QgsCoordinateTransformPrivate::mProjContext;

QgsProjContextStore::QgsProjContextStore()
{
@@ -185,7 +185,7 @@ QPair<projPJ, projPJ> QgsCoordinateTransformPrivate::t
{
    mProjLock.lockForRead();

- QMap < uintptr_t, QPair< projPJ, projPJ > >::const_iterator it = mProjProjections.constFind( reinterpret_cast< uintptr_t>( mProjContext.get() ) );
+ QMap < uintptr_t, QPair< projPJ, projPJ > >::const_iterator it = mProjProjections.constFind( reinterpret_cast< uintptr_t>(
mProjContext.localData()->get() ) );
    if ( it != mProjProjections.constEnd() )
    {
        QPair<projPJ, projPJ> res = it.value();
@@ -196,9 +196,9 @@ QPair<projPJ, projPJ> QgsCoordinateTransformPrivate::t
    // proj projections don't exist yet, so we need to create
    mProjLock.unlock();
    mProjLock.lockForWrite();
- QPair<projPJ, projPJ> res = qMakePair( pj_init_plus_ctx( mProjContext.get(), mSourceProjString.toUtf8() ),
-
        pj_init_plus_ctx( mProjContext.get(), mDestProjString.toUtf8() ) );
- mProjProjections.insert( reinterpret_cast< uintptr_t>( mProjContext.get() ), res );
+ QPair<projPJ, projPJ> res = qMakePair( pj_init_plus_ctx( mProjContext.localData()->get(), mSourceProjString.toUtf8() ),
+
        pj_init_plus_ctx( mProjContext.localData()->get(), mDestProjString.toUtf8() ) );
+ mProjProjections.insert( reinterpret_cast< uintptr_t>( mProjContext.localData()->get() ), res );
    mProjLock.unlock();
    return res;
}
```

at runtime it blows.

```
#0 QgsProjContextStore::get (this=0x0) at
/usr/obj/ports/qgis-2.99pre0/QGIS-a843df8947b3597118ad162eb0c7263469e2f1b2/src/core/qgscoordinatetransform_p.h:52
#1 QgsCoordinateTransformPrivate::threadLocalProjData (this=0x1dc74ecacb80)
    at /usr/obj/ports/qgis-2.99pre0/QGIS-a843df8947b3597118ad162eb0c7263469e2f1b2/src/core/qgscoordinatetransform_p.cpp:199
#2 0x00001dc735cce7cc in QgsCoordinateTransformPrivate::initialize (this=0x1dc74ecacb80)
    at /usr/obj/ports/qgis-2.99pre0/QGIS-a843df8947b3597118ad162eb0c7263469e2f1b2/src/core/qgscoordinatetransform_p.cpp:131
#3 0x00001dc7288c9e53 in QgsMeasureDialog::updateSettings (this=0x1dc7d6613a00)
    at /usr/obj/ports/qgis-2.99pre0/QGIS-a843df8947b3597118ad162eb0c7263469e2f1b2/src/app/qgsmeasuredialog.cpp:108
#4 0x00001dc7288d02fc in QgsMeasureTool::updateSettings (this=0x1dc7fe5fd700)
    at /usr/obj/ports/qgis-2.99pre0/QGIS-a843df8947b3597118ad162eb0c7263469e2f1b2/src/app/qgsmeasuretool.cpp:163
#5 0x00001dc7246971e4 in QMetaObject::activate(QObject*, int, int, void**) () from /usr/local/lib/qt5/.libQt5Core.so.2.2
```

**#6 - 2017-10-30 11:46 PM - Nyal Dawson**

Does 2.18.13 work ok? Maybe QThreadStorage doesn't work on bsd either...

**#7 - 2017-10-31 03:27 AM - Nyal Dawson**

Actually you're missing bits:

```
QPair<projPJ, projPJ> QgsCoordinateTransform::threadLocalProjData() const{
    mProjLock.lockForRead();
    projCtx pContext = nullptr;
    if ( mProjContext.hasLocalData() )
        pContext = mProjContext.localData()->get();
    else {
        mProjContext.setLocalData( new QgsProjContextStore() );
        pContext = mProjContext.localData()->get();
    }

    QMap < uintptr_t, QPair< projPJ, projPJ > ::const_iterator it = mProjProjections.constFind( reinterpret_cast< uintptr_t>( pContext ) );
    ...
    QPair<projPJ, projPJ> res = qMakePair( pj_init_plus_ctx( pContext, mSourceProjString.toUtf8() ),
        pj_init_plus_ctx( pContext, mDestProjString.toUtf8() ) );
    mProjProjections.insert( reinterpret_cast< uintptr_t>( pContext ), res );
}
```

(untested - may be some other changes required)

#### #8 - 2017-11-01 07:04 AM - Nyall Dawson

PR and discussion at <https://github.com/qgis/QGIS/pull/5504>

#### #9 - 2017-11-01 12:08 PM - landry Landry Breuil

Nyall Dawson wrote:

| Does 2.18.13 work ok? Maybe QThreadStorage doesn't work on bsd either...

2.18.13 & 14 work fine on OpenBSD.

#### #10 - 2017-11-06 12:45 AM - Nyall Dawson

Updated PR at <https://github.com/qgis/QGIS/pull/5541>

#### #11 - 2017-11-07 02:21 AM - Nyall Dawson

- % Done changed from 0 to 100
- Status changed from Open to Closed

Applied in changeset commit:qgis|265be41d7c1cb0bb9d39aac3285c2c8fb40ffa47.