

QGIS Application - Bug report #15978

QgsRasterDataProvider::cStringList2Q_ without UTF8 support

2016-12-13 03:23 AM - Mark Johnson

Status:	Closed	
Priority:	Normal	
Assignee:		
Category:	Rasters	
Affected QGIS version:	2.18.0	Regression?: No
Operating System:		Easy fix?: No
Pull Request or Patch supplied:	No	Resolution:
Crashes QGIS or corrupts data:	No	Copied to github as #: 23894
Description		
While (again) wondering why non-Latin characters are being garbled when metadata from TIFFTAGS are being read, I looked for the cause of this in QgsGdalProvider::metadata(). This uses the function QgsRasterDataProvider::cStringList2Q_ to read a c-stringlist. There is a missing QString::fromUtf8() when the string is being read: <pre>strings.append(QString::fromUtf8(stringList[i]));</pre> Before: TIFFTAG_IMAGEDESCRIPTION[1811 Grundriss von Berlin / von neuem aufgenommen und mit Genehmigung der Königl. Akademie der Wissenschaften hrsg. von J. C. Selter. Gestochen von C. Mare] After: TIFFTAG_IMAGEDESCRIPTION[1811 Grundriss von Berlin / von neuem aufgenommen und mit Genehmigung der Königl. Akademie der Wissenschaften hrsg. von J. C. Selter. Gestochen von C. Mare] This seems to be the only function inside QgsRasterDataProvider that needs such handling. QgsOgrProvider uses a similar logic - in the FROM8 and TO8 macros, where QString::fromUtf8() is also used Since this is mostly metadata that is being read - one can assume that non-latin characters will often be used		

Associated revisions

Revision 5ab8e717 - 2017-04-10 09:36 PM - Mark Johnson

UTF8 support for QgsRasterDataProvider::cStringList2Q_ (#4345)

UTF8 support for QgsRasterDataProvider::cStringList2Q_.

Fix #15978

History

#1 - 2017-03-20 03:20 AM - Mark Johnson

- *Target version changed from Future Release - Nice to have to Version 3.0*

This matter needs to be resolved for QGIS 3, since it creates rubbish from any UTF8 string.

#2 - 2017-04-10 12:28 PM - Mark Johnson

- *Status changed from Open to Closed*

Fixed in changeset commit:"5ab8e7176b9ca470863c125ede4183a65df1ab68".