QGIS Application - Feature request #11860
getLegendGraphic called for WMS layer while legend is not visible

2014-12-13 02:17 PM - Sandro Santilli

Status:	Closed			
Priority:	Normal			
Assignee:	Sandro Santilli			
Category:	Map Legend			
Pull Request or Patch supplied:	No		**Resolution:**	fixed/implemented
Easy fix?:	No		**Copied to github as #:**	20075

Description

The getLegendGraphic method of WMS provider is called when map content legend filtering is active, even if the legend is not visible in the layertree. Right now such call is not very expensive because there's no support for map-content dependent WMS legend (see #11859 for that) but in any case there's a call to a function that could be skipped.

It is not clear to me how to determine if the legend would be visible or not in order to decide on the call.

History

#1 - 2014-12-13 02:33 PM - Sandro Santilli

Hint:

    if (!testFlag(ShowLegend))

#2 - 2014-12-13 02:42 PM - Sandro Santilli

Taking it back, showLegend flag is always 1 when the legend is "collapsed"

#3 - 2014-12-13 02:44 PM - Sandro Santilli

I've found that the call also happen if the whole layer visibility is off, not only if the legend is invisible, but even the layer itself...

#4 - 2014-12-13 02:45 PM - Sandro Santilli

- Assignee set to Martin Dobias

#5 - 2014-12-13 03:02 PM - Sandro Santilli

BTW, I tried to find code setting the ShowLegend flag but all I found was it being set in the constructor for QgsLayerTreeModel and never toggled. It is a model-wide setting anyway, nor a per-node one.

#6 - 2014-12-13 03:31 PM - Sandro Santilli

isExpanded() could be used to check for legend being visible, but we can't just skip to create legend node as otherwise the collapse/expand icon would be removed (in absence of child). So we need a kind of placeholder and also we need to react on expand event.

#7 - 2014-12-13 05:27 PM - Martin Dobias

- Assignee deleted (Martin Dobias)

This looks to me like a new feature rather than a bug :)

I would suggest to have a new QgsMapLayerLegend implementation specific for WMS raster layers. That one would return legend node that would be lazily initialized - it wouldn't fetch getLegendGraphics from WMS unless asked in QgsLayerTreeModelLegendNode::data(). I would even suggest to download the legend graphics asynchronously so that GUI is not blocked: on first data() request it would issue a request and show a temporary "loading" icon in the legend, once the legend graphics is downloaded, the real image would be shown.

I don't think it would be good to use isExpanded() and expanded event. The Qt model/view framework supports the lazy approach - it does not request data it does not need (e.g. when an item is in a collapsed part of tree).

#8 - 2014-12-14 02:09 AM - Giovanni Manghi

- Tracker changed from Bug report to Feature request

#9 - 2014-12-15 02:49 AM - Sandro Santilli

Thanks for the insights Martin, I'll make this enhancement part of the work for #11859 then. Work is in progress

#10 - 2014-12-15 11:00 AM - Sandro Santilli

For the record, I've pushed an experimental version of the suggested approach here:
https://github.com/strk/QGIS/tree/wmslegend-node

The lazy initialization is triggering a segfault for some reason I cannot understand.

#11 - 2014-12-17 06:51 AM - Sandro Santilli

The experiments to lazily load the legend from Model::data() ended up being full of peril in that the implementation for getLegendGraphic in QgsWmsProvider uses an internal QEventLoop which seem to process further dataChanged event and eventually corrupts the process memory.

The code in current PR seems to be more stable, by only fetching legend when ::data() is called with DecorationRole.
But calling it also for SizeHintRole starts the hell. The async load idea could fix this problem, but would need implementing support for that in the WmsProvider class. Any help with that ?

#12 - 2014-12-17 09:07 AM - Sandro Santilli

Finally I've started refactoring legend downloading to allow for async loading. For now it's only used internally by the blocking (but dangerous) getLegendGraphic. But I plan to use the async model directly from the model later.

See https://github.com/strk/QGIS/commit/cc208f6d22e4d41e83c66f82ede0f02de930b86e

#13 - 2014-12-18 07:33 AM - Sandro Santilli

I've the async downloader used by WMSLegendNode and it works very nicely.
This required exposing a new virtual method in QgsRasterDataProvider:

virtual QgsImageDownloader* getLegendGraphicDownloader(const QgsMapSettings* mapSettings)

Now I'd need to know if the RasterDataProvider supports or does not support map-based wms legend because if it doesn't we'd better not get a new downloader. Rather, we could reuse the cache.

Handling the cache fully within the QgsRasterDataProvider could be possible but then we'd still need to provide a QgsImageDownloader implementation for the sake of returning a cached image.

What do you think ?

\\cc Martin Dobias, Nyall Dawson

#14 - 2014-12-18 08:52 AM - Sandro Santilli

- Assignee set to Sandro Santilli

I've taken the QgsWmsRasterProvider internal caching road for now.
It will emit a QgsCachedImageFetcher object when a cache is available for the given scale/extent.
See it here: https://github.com/strk/QGIS/commit/0c744c2a3d56e96c6e48b743f33aa4987ae595c0

#15 - 2015-01-07 04:32 AM - Sandro Santilli

- % Done changed from 0 to 100

- Status changed from Open to Closed

- Resolution set to fixed/implemented

The commit is now in master, not-visible WMS legends are not fetched anymore.