QGIS Application - Bug report #4604
WFS option "Only request features overlapping the current view extent" disappeared from QGIS master

2011-12-04 03:10 AM - Giovanni Manghi

Status:	Closed		
Priority:	High		
Assignee:			
Category:	Web Services clients/WFS		
Affected QGIS version:	master	**Regression?:**	No
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	Yes	**Resolution:**	fixed/implemented
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	14509

Description

Still available in 1.7.2

Related issues:

Related to QGIS Application - Bug report # 14876: WFS client broken in QGIS m...	**Closed**	**2016-05-24**

History

#1 - 2011-12-10 11:30 AM - Bill Clay

Looks like this was intentionally disabled by developer "MD" in providers/qgswfssourceselect.cpp at lines 54 and 266-274. Perhaps this was done pending implementation of feature request #4280? For my work, it's better to have that capability (even with the flaw described at feature request #4280) than not have it at all.

#2 - 2011-12-16 02:09 PM - Giovanni Manghi

- Target version set to Version 1.7.4

#3 - 2011-12-19 08:26 AM - Bill Clay

- File incremental-wsf.patch added

The attached patch reimplements the disappeared capability described above, thanks to Marco Hugentobler, who supplied much of the approach and code.

Differences/limitations:

1. The former per-WFS-source "only request features overlapping ..." checkbox has become a per-layer "cache features" checkbox. The new checkbox is the inverse of the old option. It is checked by default; the entire WFS layer is fetched and cached at layer instantiation unless user UNchecks it.
2. If "cache features" is NOT checked, WFS layers will be incrementally fetched to cover new areas exposed by pans and zooms. These fetches will cover the combined extent of the previous WFS layer extent and the new canvas extent. (This causes large WFS fetches on large diagonal pans.)
3. The incremental fetches described above work with and without "on the fly" canvas reprojection.
4. If "cache features" is NOT checked and the layer contains no features that intersect the canvas extent at layer instantiation, layer creation fails. Subsequent pans and zooms will NOT retrieve features, even if they intersect the newly-rendered areas. (This bug appears to independent of this patch.)

This patch has NOT BEEN TESTED with editable WFS layers; I do not have access to an updateable WFS repository.

I intend to address the issues in items 2 and 4 above ASAP, but the restored functionality may be useful to others even with these limitations.

#4 - 2011-12-19 08:28 AM - Giovanni Manghi

- Pull Request or Patch supplied changed from No to Yes

#5 - 2011-12-22 07:03 AM - Bill Clay

- File incremental-wsf-rev2.patch added

The attached patch completely **replaces** the patch attached at update nr. 3. It contains the following improvements:

1. correctly creates layers even if their initially retrieved extents contain no features (e.g., due to BBOX or FILTER)
2. eliminates double fetches in some cases and removes unreliable logic for mismatchs between project and saved layer extents by refactoring initialization logic (DescribeFeatureType) out of feature retrieval logic (GetFeature)
3. limits incremental growth of fetched extents when "Cache Features" is not selected

When "Cache Features" is NOT selected, the following flaws are present:

A. per OGC WFS spec, BBOX and FILTER are mutually exclusive; BBOX must be merged into the filter, server's capabilities permitting

B. when fetched extents are incrementally expanded, the previously fetched area should not be re-fetched

I hope to correct these flaws within year 2011.

This code **has not been tested with editable WFS providers!**

#6 - 2011-12-27 01:07 AM - Marco Hugentobler

- Affected QGIS version set to master

- Status changed from Open to Closed

- Crashes QGIS or corrupts data set to No

Patch is pushed to master (commit:9957ee58f05892c3251216dc5dee421405a6a412). Thank you very much for these nice improvements!

The approach to read the feature type from describeFeatureType is better than the current one. However, in some situations, there is no feature type declared (e.g.Giovannis test layer for WFS-T). If the feature type is still WKBUnknown after describeFeatureType, it would be good to additionally get the type from the first fetched feature.

In future, it would be cool if there was a possibility to limit memory for really large layers (e.g. fetch layer in several parts if too big, keep already fetched features in cache)

#7 - 2011-12-27 09:28 AM - Paolo Cavallini

In a large, real use case, option #3 is most probably the best, as it will happen often that a layer is quite large, with many large features. Avoiding refetching (based on an ID/PK) will greatly increase the usability of the plugin.
Probably the local cache could be stored in SpatiaLite, to take advantage of spatial and alphanumeric indexing, etc.

#8 - 2011-12-31 07:46 AM - Bill Clay

- File incremental-wsf-rev3.patch added

The attached patch is intended to replace my 2nd patch attached at update 5 above.

Although this patch has no known bugs, it has two undesirable (but infrequent) interactions. Perhaps a slightly-improved version of the 2nd patch is what should be released, even though it will usually run slower. Please let me know which the team prefers.

The major enhancement of this patch over patch nr. 2:

- "Incremental get:" when "cache all features" is not selected, the WFS provider fetches features only for the NEW map canvas areas exposed by pans and zoom-outs, adding them to cached features for the areas previously fetched. This improves response time for areas with many features. A user who pans and zooms in a localized area rapidly builds a cache that avoids delays for new WFS feature retrieval.

Undesirable interactions:

- The map canvas extent is occasionally expanded or shifted slightly without the user (or this patch) explicitly changing it. This can trigger a WFS "incremental get," with resulting delay. This is not a bug, but it may surprise the user.

- Testing revealed one WFS provider (ArcGIS on Windows) that does not process filters with BBOXes as produced by this patch. Rather than returning an exception, the service returns ALL features contained in the layer. This causes slow response on each "incremental" fetch, but QGIS continues to function correctly. I have found no simple & efficient method to reliably detect WFS providers that have this problem (I omitted a moderately complex and inefficient one from this patch). Work-around: user deletes the layer and recreates it with the "cache all features" option.

>> *Does anyone know a reliable method to identify WFS providers that do NOT support filter syntax v. 1.0.0, but that do not signal an*

exception when they receive it? (Methods that do NOT work: "version=1.0.0" in the GetCapabilities URL; looking at version number in the WFS_Capabilities xsi:schemaLocation attribute -- it contains http://schemas.opengis.net/filter/1.0.0/filter.xsd!)

Responses to updates 6 & 7 above:

Marco, I encountered the layer creation problem with Giovanni's TinyOWS test server and added the fix you suggested.

Paolo, I'm willing to take a swing at a SpatiaLite local cache, but I'd need some guidance on what you and Marco have in mind. It presents some interesting design challenges that the team has probably already resolved in similar contexts. (I've also never used SpatiaLite, but I have used SQLite a bit in Python.)

#9 - 2012-01-08 03:25 AM - Marco Hugentobler

BBOX support:

Normally, a WFS server lists BBOX as a spatial capability (see below). Is this also the case for ArcGIS WFS?

```
<ogc:Filter_Capabilities><ogc:Spatial_Capabilities><ogc:Spatial_Operators><ogc:Disjoint/><ogc:Equals/><ogc:DWithin/><ogc:Beyond/><ogc:In
tersect/><ogc:Touches/><ogc:Crosses/><ogc:Within/><ogc:Contains/><ogc:Overlaps/><ogc:BBOX/></ogc:Spatial_Operators></ogc:Spatial_Capabilities>
pabilities>
```

#10 - 2012-01-08 12:07 PM - Bill Clay

Marco, my nr. 3 patch uses its "incremental get" method only with WFS providers that advertise filtering capability with BBOX and logical operators. The ArcGIS system in question advertises all three in response to WFS GetCapabilities, both version 1.0.0 or 1.1.0. Unfortunately, it does not provide the service it advertises, at least with a version 1.0.0 GetFeatures. It ignores my <filter> and provides ALL the features in the queried layer.

How do you want to handle that? I need direction from someone who knows better than I QGIS architecture, practice, and plans regarding the 2nd bullet item in update 8 above.

Meanwhile, I have resolved the first "undesirable interaction" listed in update 8 with the patch attached to issue #4734, which does not touch WFS code.

#11 - 2012-01-09 02:17 AM - Giovanni Manghi

- Status changed from Closed to In Progress

- % Done changed from 0 to 50

#12 - 2012-01-10 07:44 PM - Jeremy Palmer

Hi Bill,

I Just came across this now. Very interesting indeed. This work is quite funny because I was just talking to colleagues before Xmas about options for improving the WFS provider, including the possibility of using a spatialite cache!

To give you some background we (LINZ) provide our data via WFS to the public using geoserver on http://data.linz.govt.nz. Many of our vector layers have millions of features in them (e.g http://data.linz.govt.nz/#/layer/772-nz-primary-parcels/webservices) and we would like users to be able to add these layers to a "fresh" QGIS canvas and then zoom in to their smaller area of interest - much like you can do with shapefiles or PostGIS layers by adding the layer, canceling the rendering, and then zooming in. This way a general user would not have to understand caching options to get going.

Not that I've looked at the full WFS code that much, but it seems that this might be possible to do if the WFS server has the necessary capabilities (as you have already worked on). Our early thinking before Xmas is quite close to what you have in patch 3, but has a few subtitle differences:

- Don't load the WFS data on the construction of the object
- QgsWFSProvider::nextFeature would be changed. This method will do 2 things, first check a cache to see if the current select extents are available. If the extent is not it will get the data incrementally using the QgsWFSData reader (which is setup to get the faetures in batches), put the data into the cache and finally returning false once the reader has run out features. Also once the reader has finished it will set the cache to flag the extents as complete.
- The cache would have an index of extents that had been visited and would dissolved extents that are superseded by larger covering extents. Ideally the cache is configurable to be memory based (map of features by fid with a spatial index, discarding features once memory limits are reached) or a spatialite database backend storing the features which are indexed by fid and geometry.
- The cache is invalided if the WFS filter changes.

BTW. I'm more that willing to help with testing or any of the WFS stuff you are working on. I've already testing out your WFS changes so far, and it's a big improvement with large WFS datasets!

Cheers,
Jeremy

#13 - 2012-01-11 12:41 AM - aperi2007 -

> *QgsWFSProvider::nextFeature would be changed. This method will do 2 things, first check*
> *a cache to see if the current select extents are available. If the extent is not it will*
> *get the data incrementally using the QgsWFSData reader (which is setup to get the*
> *faetures in batches), put the data into the cache and finally returning false once the*
> *reader has run out features. Also once the reader has finished it will set the cache to*
> *flag the extents as complete.*

Hi, i like give my opinion on the cache system for a WFS server.

I guess that unfortunately, use only the extent is not sufficient to understand if the WFS client has cached all the feature of that extent.

Infact there is some WFS Server (tinyows for example) that allow to limit the maximun features send.
So if an extent there is really 10.000 feature, but the WFS Server is set to send maximum 5000 features. You wrongly understand that the extend is fully received, instead there are again 5000 feature you don't know there are in that extent.

To avoid this , when the user do a zoom into a new extent that is inside the old extent, the WFS client should again send a request to the Server WFS.
So the server WFS send the feature available in this new extent. Again it send max 5.000 features.
The cache system should analyze every feature received to understand if it is a new feature or a just cached feature. To do this I guess the better method is use the MD5 of the feature: If a new feature received has the same MD5 of a just cached feature (this kind of search are simple in a spatialite) otherwise add to the cache this new feature received.

Regards,
Andrea.

#14 - 2012-01-12 04:58 PM - Jeremy Palmer

Hi Andrea,

Hmm this is bad news. I didn't realise that WFS servers could do that. Is this advertised in the server capabilities, or as a vendor extension?

If this really is the case, then caching is almost a waste of time because each extent refresh will cause a server request, causing server processing and full transfer of the GML payload.

If only tinyows does this then we could test for the server type and disable caching. I know the WFS ogr driver does server testing like this.

Cheers
Jeremy

#15 - 2012-02-02 04:29 AM - Paolo Cavallini

Any hope to have the option back soon? Thanks a lot.

#16 - 2012-02-02 06:50 AM - Bill Clay

Sorry, all, for the long silence. I'm in the US on personal business and vacation Jan. 17 - Mar. 19, and will not have much time to work on this. However, my nr. 3 patch restores the desired "fetch map extent only" functionality despite its limitations (fairly minor, I believe). Since Marco Hugentobler is the WFS original developer, and surely knows this code MUCH better than I, I guess he's the one to decide whether it's worth releasing my patch as it currently exists.

The following is a list of the remaining open issues, as I understand them:

1. There appear to be some WFS servers that advertise WFS v. 1.0 filter capabilities, but then ignore the filter and send the whole layer. (I have found only one such instance, but it may be common to ArcGIS of certain versions.)
2. I understand TinyOWS can be configured NOT to provide its internal FeatureIDs; Andrea Peri's test server is the only example I know. Andrea suggests computing our own internal "FeatureID" by running an MD5 digest on the feature text. This would be expensive, but we could add logic to do it only when required. If Tiny OWS is the only package that can be configured NOT provide Feature IDs, I'm not sure it's worth the extra logic. Maybe we should simply require that a WFS service provide Feature IDs if one wants to use this feature.
3. I opened issue #4734 for the "undesired canvas movement" problem (first "undesirable interaction" in my update nr. 8 above) and supplied a patch. Alexander Bruy closed that issue yesterday as "fixed." HOWEVER, looking at the text of his commit, it appears he fixed a similar issue on "map scale" edits, but the commit does not seem to include my patch for "map center" edits. Unless I've missed something (perhaps "map scale" and "map center" are in a union and logic that fixes one fixes both?), #4734 should be reopened and my patch (or an improved version like Alexander's "map scale" edit patch)

actually applied to master. I have not (yet) contacted Alexander about this, nor have I looked deeper than the text of the commit posted yesterday, so this issue may really be fixed.

Between preparing for my trip and actually leaving, I have not had the opportunity to examine Jeremy Palmer's suggestions in any depth nor responded to Andrea Peri's concerns regarding Tiny OWS's omitting Feature IDs. I apologize to both, and hope to be able to respond more constructively within two weeks. I would like to continue to improve the WFS caching code once I return to Italy, but if someone is able to do so sooner or better, please don't hesitate!

#17 - 2012-04-16 06:32 AM - Paolo Cavallini

- Target version changed from Version 1.7.4 to Version 1.8.0

#18 - 2012-09-04 11:56 AM - Paolo Cavallini

- Target version changed from Version 1.8.0 to Version 2.0.0

#19 - 2012-12-13 01:50 PM - Giovanni Manghi

what is the status of this in progress ticket? as it was a feature available in the past in QGIS should we tag this as regression?

#20 - 2013-04-21 06:12 PM - Nathan Woodrow

- Priority changed from Normal to Severe/Regression

It would be good to have this back as a lot of people in Australia use this feature.

#21 - 2013-07-12 04:50 AM - Nathan Woodrow

- Priority changed from Severe/Regression to High

This seems to have been missing from 1.8 too. While it would be good to have back there seems to be a lot involved in adding this feature back in. Downgrading for now.

#22 - 2014-01-30 11:43 PM - Paolo Cavallini

- Target version changed from Version 2.0.0 to Future Release - High Priority

#23 - 2014-02-07 05:15 AM - Giovanni Manghi

- Priority changed from High to Normal

There is no crash or data corruption, so I'm lowering the priority.

#24 - 2014-06-20 12:23 PM - Giovanni Manghi

- Status changed from In Progress to Open

- Priority changed from Normal to High

this does not seems a work in progress anymore.

#25 - 2014-11-12 07:36 AM - Burak Katipoglu

Still no progress about this feature/issue in 2.6 version.

#26 - 2015-10-07 02:28 AM - Nicolas Boisteault

This feature is not present in 2.8.3 neither but is back in 2.10.1.
Though I looked in Fiddler and saw that there is no BBOX parameter added to the request with this option checked.
The strangest is that I have to cancel once to get the right query sent.
I'll test on master and report back.

#27 - 2015-10-07 05:23 AM - Nicolas Boisteault
- File WFS_issue.jpg added

I tested master (2.11 rev. 890dce8). To have WFS features in my current view extent, there are two possible methods :

1. To uncheck "Cache Feature", click add button and cancel first query. A second query with bbox is sent after.
2. To check "Only request features overlapping the current view extent", click add button and cancel first query. A second query with bbox is sent after.

In both cases if I move the view extent there are no new request sent. I can't see any other difference between these two methods.
In screenshot attached I put in red two '&&'. Maybe BBOX parameter should be between those.

#28 - 2016-05-24 04:01 AM - Giovanni Manghi
- Status changed from Open to Feedback

The wfs client received several improvements along last weeks, but it also seems broken right now. #14876

#29 - 2016-05-24 01:26 PM - Giovanni Manghi
- Status changed from Feedback to Closed
- Resolution set to fixed/implemented

implemented in master.

Files

incremental-wsf.patch	15.7 KB	2011-12-19	Bill Clay
incremental-wsf-rev2.patch	28.4 KB	2011-12-22	Bill Clay
incremental-wsf-rev3.patch	43 KB	2011-12-31	Bill Clay
WFS_issue.jpg	148 KB	2015-10-07	Nicolas Boisteault