QGIS Application - Bug report #4607
Do not show not pertinent fields in the "data defined settings" of the labelling tool

2011-12-04 04:24 AM - Giovanni Manghi

Status:	Closed		
Priority:	Normal		
Assignee:	Larry Shaffer		
Category:	Labelling		
Affected QGIS version:	master	**Regression?:**	No
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	fixed
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	14512

Description

For example, why show non integer columns for the "rotation" field? Like this all the others should accept just text or integer fields.

History

#1 - 2011-12-05 08:09 AM - Paolo Cavallini

See also #4613

#2 - 2011-12-16 10:55 AM - Giovanni Manghi

- Target version changed from Version 1.7.3 to Version 1.7.4

#3 - 2012-04-16 06:32 AM - Paolo Cavallini

- Crashes QGIS or corrupts data set to No

- Target version changed from Version 1.7.4 to Version 1.8.0

- Affected QGIS version set to master

#4 - 2012-09-04 11:56 AM - Paolo Cavallini

- Target version changed from Version 1.8.0 to Version 2.0.0

#5 - 2012-12-11 05:52 AM - Giovanni Manghi

- Assignee set to Larry Shaffer

seems still an issue in master, text fields are shown for parameters that need to be just numbers.

#6 - 2013-05-17 04:31 PM - Larry Shaffer

Hi Giovanni,

I have implemented this in the new QgsDataDefinedButton. However, and considering basic types, restricting options that require an Int or Double input value to only data source fields that support those types in a bit limiting.

This is because the resultant value, either from a direct feature field-mapped attribute query, or from an expression, can often be successfully converted to the appropriate type. For example, almost all results are QVariants, which means the result needs be be converted from QVariant to the expected type before it can be used.

Specifically:

> ***double QVariant::toDouble (bool * ok = 0) const***
>
> *Returns the variant as a double if the variant has type() Double, QMetaType::Float, Bool, ByteArray, Int, LongLong, **String**, UInt, or ULongLong; otherwise returns 0.0.*

> ***int QVariant::toInt (bool * ok = 0) const***
>
> *Returns the variant as an int if the variant has type() Int, Bool, ByteArray, Char, Double, LongLong, **String**, UInt, or ULongLong; otherwise returns 0.*

> ***QString QVariant::toString () const***
>
> *Returns the variant as a QString if the variant has type() **String**, Bool, ByteArray, Char, Date, DateTime, **Double**, **Int**, LongLong, StringList, Time, UInt, or ULongLong; otherwise returns an empty string.*

So you can see that conversion to Int or Double can accept a String, and vic-versa. So there is really no need to limit the data source field type, unless you know it can not be successfully converted to what's needed. For example, if a data defined value might be the text 'Half' then limiting to only String type fields makes sense because Int and Double fields could not possibly produce such a result.

#7 - 2013-05-19 04:23 AM - Giovanni Manghi

- Status changed from Open to Feedback

Hi Larry,

thanks for the detailed explanation.

So if you think that the original issue make no sense especially after your labelling overhaul, please close this ticket.

cheers!

Larry Shaffer wrote:

> *Hi Giovanni,*
>
> *I have implemented this in the new QgsDataDefinedButton. However, and considering basic types, restricting options that require an Int or Double input value to only data source fields that support those types in a bit limiting.*
>
> *This is because the resultant value, either from a direct feature field-mapped attribute query, or from an expression, can often be successfully converted to the appropriate type. For example, almost all results are QVariants, which means the result needs be be converted from QVariant to the expected type before it can be used.*
>
> *Specifically:*
>
>> ***double QVariant::toDouble (bool * ok = 0) const***
>>
>> *Returns the variant as a double if the variant has type() Double, QMetaType::Float, Bool, ByteArray, Int, LongLong, **String**, UInt, or ULongLong; otherwise returns 0.0.*
>
> ***int QVariant::toInt (bool * ok = 0) const***

> > *Returns the variant as an int if the variant has type() Int, Bool, ByteArray, Char, Double, LongLong, **String**, UInt, or ULongLong; otherwise returns 0.*
>
> > ***QString QVariant::toString () const***
> >
> > *Returns the variant as a QString if the variant has type() **String**, Bool, ByteArray, Char, Date, DateTime, **Double**, **Int**, LongLong, StringList, Time, UInt, or ULongLong; otherwise returns an empty string.*
>
> *So you can see that conversion to Int or Double can accept a String, and vic-versa. So there is really no need to limit the data source field type, unless you know it can not be successfully converted to what's needed. For example, if a data defined value might be the text 'Half' then limiting to only String type fields makes sense because Int and Double fields could not possibly produce such a result.*

#8 - 2013-05-29 01:42 PM - Larry Shaffer

- Status changed from Feedback to Resolved

- Resolution set to fixed

Should be fixed with commit commit:45f374f4

Reopen if necessary

#9 - 2014-02-13 12:31 PM - Jürgen Fischer

- Status changed from Resolved to Closed