

## QGIS Application - Bug report #10134

### Credential Dialog, disabling default Realm information messages that discloses password

2014-04-29 04:38 AM - F B

<b>Status:</b>	Rejected	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	Jürgen Fischer	
<b>Category:</b>		
<b>Affected QGIS version:</b>	2.2.0	<b>Regression?:</b> No
<b>Operating System:</b>		<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b>	No	<b>Resolution:</b> invalid
<b>Crashes QGIS or corrupts data:</b>	No	<b>Copied to github as #:</b> 18593
<b>Description</b>		
<p>I'm using the QgsCredentialDialog for determining user access to the database. When a user is failing to log in correctly the log in credentials are shown in the dialog (see attached screenshot).</p> <pre>(success, uname, passwd) = QgsCredentialDialog.instance().get(connInfo, None, None, "Log in")</pre> <p>Despite the credentials being faulty there's the case where username is misspelled but the password is correct. As a minimum, I'd like it if the entered password was not shown in the "Realm" message at the top of the dialog (see attached screenshot).</p>		

#### History

##### #1 - 2014-04-30 02:29 AM - Jürgen Fischer

- Target version deleted (Version 2.2)
- Status changed from Open to Rejected
- Category deleted (Data Provider/PostGIS)
- Resolution set to invalid
- Priority changed from High to Normal

If you don't want to have the password visible you shouldn't store it.

QgsCredentialDialog isn't supposed to be run directly, use QgsCredentials.instance().

QgsCredentials caches the result based on the give realm, so it shouldn't change between get calls.

put the credentials into the cache once you verified that username and password are correct (maybe using an updated copy of connInfo), so that the providers can fetch them if they need it.

##### #2 - 2014-04-30 02:40 AM - F B

In the example above I do use QgsCredentials.instance().

I think I do most of the things that you propose. Here is a bit more of my code to perhaps make it more clear.

```
self._uri.setConnection(self.getHostName(), self.getPortNr(), self.getDbName(), None, None)
connInfo = self._uri.connectionInfo()
(connection_success, uname, passwd) = QgsCredentialDialog.instance().get(connInfo, None, None, "Log in")
.
.
.
self._uri.setUsername(self.getUname())
self._uri.setPassword(self.getPassWd())
```

What do you mean by not storing the password, can you give an example? (is it what I do when I set the password in the uri object?)

#3 - 2014-04-30 03:26 AM - Jürgen Fischer

F B wrote:

| *What do you mean by not storing the password, can you give an example? (is it what I do when I set the password in the uri object?)*

You shouldn't put the password into the connInfo you pass to get() at least. Everything else depends on what you actually do with the credentials.

If you want to create a layer with it later the uri passed to it should also not contain the password. Instead the credentials for should be QgsCredentials.put() into the cache. The provider will then run QgsCredentials.get() and fetch the credentials from the cache (without a new prompt) and the datasource uri of the layer will still not contain the password.

Files

credentials1.png	10.7 KB	2014-04-29	F B
------------------	---------	------------	-----