Quassel IRC - Feature #826

SQL backends cleanup

10/12/2009 09:57 PM - lupine

Status:	New	**Start date:**	10/12/2009
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Quassel Core	**Estimated time:**	0.00 hour
Target version:			
OS:	Any		

Description

Right now, Quassel has support for SQLite and Postgresql; it implements that through three classes (an ABC and a subclass of that per-database). I was looking into adding MySQL support (and other backends too), but the current state of the backends makes me want to stop, clean it all up, and then implement a 'good' MySQL backend - rather than doing a copy 'n paste of the one backend and hacking it to work with MySQL.

A summary of stuff: * The abstract base class isn't... specifically, it doesn't define the database state manipulation interface (addUser, etc). Instead, each backend implements it instead, and the whole thing only works because of convention * Shared code isn't. Specifically query construction is again handled separately in each subclass - this could be moved into the ABC and would significantly cut the 1.6kloc size of the subclasses - lowering the barrier to implementing new backends. * Locking seems to be applied erratically and doesn't always do what you'd expect it to. Quassel can't always recover from another process accessing its sqlite database, for instance. Again, moving the code doing this into the ABC simplifies the overall pictures and reduces bug count * No tests... * No support for ODBC * The schema migrations and query construction framework seems, again, to have considerable (unnecessary) duplication, which I suspect would go away with some work. In current git master, sqlite is on schema version 15, but pgsql is on 14, for instance * Is there any reason there isn't just an ODBC backend? * The whole thing is procedural in nature, but it would probably make more sense for it to be built around model objects (possibly using Qt's table models? That's not a feature I know much about)

It sounds like one long gripe, but I'm interested in writing code to clean this lot up ;) - and here's as good a place as any to document what I'd like to do, and get some feedback...

Ideally, I'd like to, first: implement some tests for the functionality provided by the various sqlstorage classes; second, move as much common code into the abstract class as possible (keep all the tests passing, of course), keeping the two current sql backends up to date; third, implement a mysql backend; fourth, an ODBC backend. Finally, investigate moving to a database model architecture - that's the only point that touches other subsystems significantly ;)

Comments all welcome, of course.

Related issues:

Related to Quassel IRC - Feature #1046: MySQL support	**New**	**01/31/2011**

History

#1 - 03/09/2010 08:00 PM - johu

- Tracker changed from Bug to Feature

#2 - 03/09/2010 08:01 PM - johu

- Target version deleted (0.6.0)