

Re: DBIx::DBH – Perl extension for simplifying database connectio ns

Source: <http://coding.derkeiler.com/Archive/Perl/perl.dbi.users/2004-12/0120.html>

From: Jonathan Leffler (jonathan.leffler_at_gmail.com)

Date: 12/08/04

Date: Tue, 7 Dec 2004 22:26:22 -0800
To: "Orton, Yves" <yves.orton@de.mci.com>

On Tue, 7 Dec 2004 15:28:07 -0000, Orton, Yves <yves.orton@de.mci.com> wrote:

> *Henrik Tougaard wrote on 07 December 2004 10:59*

>

>

> > *Martyn J. Pearce skrev:*

> > > *On Fri, Dec 03, 2004 at 11:25:53AM +0000, Tim Bunce wrote:*

> > ...

> > > *The simplest fix is to standardize one set of driver DSN attribute*

> > > *names so that at least the host, port, and database (schema) can*

> > > *be specified in a portable way.*

> > >

> > > *Is that really the simplest? It occurs that the responses on this*

> > > *thread, and in my experience, many people are comfortable & familiar*

> > > *with the use of a hash (/ref) for this purpose.*

> >

> > *Maybe the number of responses on this thread come from people who*

> > *have this itch to scratch. I have heard Tim Bunce (DBI, DBD::Oracle*

> > *etc) and Jonathan Leffler (DBD::Informix) raise 'Beware, thing are*

> > *much more complex than you think' warnings.*

> >

> > *I (DBD::Ingres) have'nt pitched in as Jonathan voiced my concerns*

> > *quite precisely, saying:*

> > *Beware – DBMS are more different than anyone would like.*

> > *That's why DBI has the scheme it does have – it is flexible*

> > *but not easily codified.*

>

> *I just looked at the DBD::Informix docs. According to them Informix takes a*

> *connections string like:*

> *"dbi:Informix:\$database" where \$database is constructed like this:*

>

> *dbase # 'Local' database*

> *//machine1/dbase # Database on remote machine*

> *dbase@server1 # Database on (remote) server (as defined in*

> *sqlhosts)*

> *@server1 # Connection to (remote) server but no database*

perl.dbi.users: Re: DBIx::DBH – Perl extension for simplifying database connections

- > */some/where/dbase # Connect to local SE database*
- >
- > *DBD::Ingres does something similar. DBD::Oracle appears to be closer to*
- > *Sybase/MySQL:*
- > *"dbi:Oracle:host=myhost.com;sid=ORCL"*
- >
- > *It doesn't seem like a stretch of the imagination to see the common fields*
- > *"host" and "db" embedded in all three.*

Only one of the Informix notations has a host name in it, and even that's technically not a host name but a server name – the distinction is somewhat moot since the default server name in the environment where that notation is used defaults the server name to the FUNN (fully unqualified non-domain name – to coin an acronym as the antithesis of FQDN). None of the notations goes anywhere near a port — that information is buried in a separate configuration file (\$INFORMIXDIR/etc/sqlhosts by default, which in turn references another, system configuration file, /etc/services, to establish the port – service – name).

The server names referenced are entries in the sqlhosts file. They are not references to a machine unless you adopt a convention that a machine only has one server on it (often a machine runs several servers) and that server is named after the machine – which is bad practice.

What should DBD::Informix do if it is commanded to connect to a machine name and a port number? It won't work in the public connectivity layers — and DBD::Informix does not code to the internals because they're unreliable in the sense that they can and do change. Without a server name – distinct from a host name, remember – DBD::Informix cannot know how to connect to your remote server.

No – you are misunderstanding the DBD::Informix documentation. It doesn't spell out the details for non-Informix users; I still live under the illusion that people who use Perl to access a database know which database system they are connecting to and care about which system they are connecting to and sorting out the connection string is not a major problem.

Fundamentally, the DBI spec says: you connect to a database X by specifying 'dbi:X:whatever-X-chooses'. Trying to specify 'whatever-X-chooses' in some way that is independent of X is nonsense – and that's why the DBI spec does things the way it chooses to.

For the FOSS-ish databases, your 'whatever-FOSSish-chooses' bit can be substructured any way you like – but you need to recognize that other species of database have different views on life and that while the FOSSish views are fine, so are the alternatives, and they can — and do — and have for the 8 years that I've been working on DBD::Informix — successfully co-existed.

- > *Clearly any DBD driver that can connect to providers on a different host*
- > *will have to in some way allow the user to specify which host that is. The*
- > *fact that in some particular RDBMS's culture this isn't called the "host"*
- > *and that "port" is for some reason unnecessary is IMO a bit irrelevant.*

IMO, your opinion is irrelevant. I've diagnosed that port doesn't exist in Informix; and neither does host in the normal sense of meaning the name for an i/p address. neither concept applies to standard Informix connections. You thinking that it is irrelevant is itself irrelevant. I'm sorry, you're WRONG.

- > *The*
- > *fact still remains that the generic "Host" slot could be used for this*
- > *purpose quite easily, as could the "DB" slot.*

Well, only if you know enough about the database to know how to write that information. And even that's debatable at best.

- > *Those parameter that make no*
- > *sense could either be ignored, or somehow usefully overloaded.*

You should not need to ignore what the user tells you. And no, the port number cannot be usefully overloaded. You mis-understand the problem, as a number of people have told you reasonably politely up until now. Sorry, you (collectively) have overstepped the bounds --- you speak of what you do not understand.

OK – I've probably also over-stepped the boundaries too. Sorry.

Deep breath. Calm, gentle typing... Relax.

- > *This would enable the establishment of a baseline set of connection details*
- > *that all DBD drivers should know how to more or less deal with. At bare*
- > *minimum this would mean one less trivial piece of knowledge to remember when*
- > *working with multiple providers.*

We already have that. 'dbi:Driver:Driver-specific-information'.

This is good. DBI handles the dbi bit. It also ensures the correct driver is invoked. The drivers already work.

Get with it. This is a successful, sane, safe, operational system that caters to the whims and whimsies of a diverse set of systems without causing any real irritation except to people who think they need to generalize that which does not need generalizing.

More deep breaths.

- >> *Ingres, like Informix and (I think) Oracle, doesn't have the concept*
- >> *of 'host' or 'port', using other ways of addressing remote databases.*
- >>

> > *It seems to me that you are trying to force an extension onto the DBI
> > based on what a small number of RDBMSs accept. The people who
> > want this seem to use only a few DBDs – perhaps it could be added to
> those?*
>
> *Coming up with common set of parameters that most DB's are going to require
> and then providing standardized names for them would seem to be useful in
> general.*

Not really. The connection string is system-specific -- end of sob story.

> *So far I havent seen anyone provide something that a given driver
> Has To Have that doesn't fit into the proposal. (Ie, Host,DB,Port).*

Informix does NOT have a port parameter.

> *Which
> _mandatory_ parameter does Informix need that can't be shoehorned into one
> of those?*

How are you planning to deal with the paths?

And shoehorning square pegs into round holes makes an uncomfortable fit.

In a separate email, Christopher Hicks commented:

> *I really really object to the DB slot being called DB. Oracle's term
> "tablespace" is much less overused and confusing than database.*

DB2 has table spaces; they're different from Informix's table spaces; and different again from Oracle's table spaces (though they all have this much in common – they are not something you connect to). Using 'tablespace' would not be a good idea – regardless of whether you choose to rename the 'database' to something more generic and somehow common to things that 'database management systems' permit you to connect to.

Fundamentally, the status quo works. You have a generic bit to the connection string, and you have a DBMS-specific bit. If you want to connect to a specific DBMS, you have to supply the correct notation for that DBMS. AppConfig might be a way to do it, but the factorization needs to be at the DBI-Driver-Name and DBI-Driver-Connection-String level -- leaving the internal structure of the DBI-Driver-Connection-String undefined. Obviously, a good configuration system would allow you to define DBI-Driver-Connection-String in terms of other parameters (which, for a given Driver, might include host and port information), but the rest should be left alone.

Grump. I'd better hit send before I get myself into any more trouble.

--

perl.dbi.users: Re: DBIx::DBH – Perl extension for simplifying database connectio ns

Jonathan Leffler <jonathan.leffler@gmail.com> #include <disclaimer.h>
Guardian of DBD::Informix - v2003.04 - <http://dbi.perl.org>
"I don't suffer from insanity - I enjoy every minute of it."