NAME

DBI::PurePerl -- a DBI emulation using pure perl (no C/XS compilation required)

SYNOPSIS

```
 BEGIN { $ENV{DBI_PUREPERL} = 2 }
 use DBI;
```

DESCRIPTION

This is a pure perl emulation of the DBI internals. In almost all cases you will be better off using standard DBI since the portions of the standard version written in C make it *much* faster.

However, if you are in a situation where it isn't possible to install a compiled version of standard DBI, and you're using pure-perl DBD drivers, then this module allows you to use most common features of DBI without needing any changes in your scripts.

EXPERIMENTAL STATUS

DBI::PurePerl is new so please treat it as experimental pending more extensive testing. So far it has passed all tests with DBD::CSV, DBD::AnyData, DBD::XBase, DBD::Sprite, DBD::mysqlPP. Please send bug reports to Jeff Zucker at <jeff@vpservices.com> with a cc to <dbi-dev@perl.org>.

USAGE

The usage is the same as for standard DBI with the exception that you need to set the environment variable DBI_PUREPERL if you want to use the PurePerl version.

```
 DBI_PUREPERL == 0 (the default) Always use compiled DBI, die
 if it isn't properly compiled & installed

 DBI_PUREPERL == 1 Use compiled DBI if it is properly compiled
 & installed, otherwise use PurePerl

 DBI_PUREPERL == 2 Always use PurePerl
```

You may set the environment variable in your shell (e.g. with set or setenv or export, etc) or else set it in your script like this:

```
 BEGIN { $ENV{DBI_PUREPERL}=2 }
```

before you `use DBI;`.

INSTALLATION

In most situations simply install DBI (see the DBI pod for details).

In the situation in which you can not install DBI itself, you may manually copy DBI.pm and PurePerl.pm into the appropriate directories.

For example:

```
 cp DBI.pm /usr/jdoe/mylibs/.
 cp PurePerl.pm /usr/jdoe/mylibs/DBI/.
```

Then add this to the top of scripts:

```
 BEGIN {
 $ENV{DBI_PUREPERL} = 1; # or =2
 unshift @INC, '/usr/jdoe/mylibs';
 }
```

(Or should we perhaps patch Makefile.PL so that if DBI_PUREPERL is set to 2 prior to make, the normal compile process is skipped and the files are installed automatically?)

DIFFERENCES BETWEEN DBI AND DBI::PurePerl

Attributes

Boolean attributes still return boolean values but the actual values used may be different, i.e., 0 or undef instead of an empty string.

Some handle attributes are either not supported or have very limited functionality:

```
ActiveKids
InactiveDestroy
AutoInactiveDestroy
Kids
Taint
TaintIn
TaintOut
```

(and probably others)

Tracing

Trace functionality is more limited and the code to handle tracing is only embedded into DBI:PurePerl if the DBI_TRACE environment variable is defined. To enable total tracing you can set the DBI_TRACE environment variable as usual. But to enable individual handle tracing using the *trace()* method you also need to set the DBI_TRACE environment variable, but set it to 0.

Parameter Usage Checking

The DBI does some basic parameter count checking on method calls. DBI::PurePerl doesn't.

Speed

DBI::PurePerl is slower. Although, with some drivers in some contexts this may not be very significant for you.

By way of example... the test.pl script in the DBI source distribution has a simple benchmark that just does:

```
my $null_dbh = DBI->connect('dbi:NullP:','','');
my $i = 10_000;
$null_dbh->prepare('') while $i--;
```

In other words just prepares a statement, creating and destroying a statement handle, over and over again. Using the real DBI this runs at ˜4550 handles per second whereas DBI::PurePerl manages ˜2800 per second on the same machine (not too bad really).

May not fully support *hash()*

If you want to use type 1 hash, i.e., `hash($string,1)` with DBI::PurePerl, you'll need version 1.56 or higher of Math::BigInt (available on CPAN).

Doesn't support *preparse()*

The DBI->*preparse()* method isn't supported in DBI::PurePerl.

Doesn't support DBD::Proxy

There's a subtle problem somewhere I've not been able to identify. DBI::ProxyServer seem to work fine with DBI::PurePerl but DBD::Proxy does not work 100% (which is sad because that would be far more useful :) Try re-enabling t/80proxy.t for DBI::PurePerl to see if the problem that remains will affect you're usage.

Others

```
can() - doesn't have any special behaviour
```

Please let us know if you find any other differences between DBI and DBI::PurePerl.

AUTHORS

Tim Bunce and Jeff Zucker.

Tim provided the direction and basis for the code. The original idea for the module and most of the brute force porting from C to Perl was by Jeff. Tim then reworked some core parts to boost the performance and accuracy of the emulation. Thanks also to Randal Schwartz and John Tobey for patches.

COPYRIGHT