HyperSQL ODBC Driver

User Guide for the Hyper-SQL ODBC Driver, hsqlodbc

Edited by and Blaine Simpson

HyperSQL ODBC Driver: User Guide for the HyperSQL ODBC Driver, hsqlodbc

by and Blaine Simpson

$Revision: 2864 $

Publication date Dat

Table of Contents

List of Tables

Preface

ALPHA DOCUMENTATION

This documentation is not complete and is released to accompany an ALPHA version of HSQLDB 1.9 or 2.0. The latest, updated version of this documentation can be found at http://hsqldb.org/doc/2.0/

If you notice any mistakes in this document, please email the author listed at the beginning of the chapter. You can find the emails in the list of developers on the SourceForge project site via http://hsqldb.org. If you have problems with the procedures themselves, please use the HSQLDB support facilities which are listed at http://hsqldb.org/support

Available formats for this document

This document is available in several formats.

You may be reading this document right now at http://hsqldb.org/doc/2.0, or in a distribution somewhere else. I hereby call the document distribution from which you are reading this, your *current distro*.

http://hsqldb.org/doc/2.0 hosts the latest production versions of all available formats. If you want a different format of the same *version* of the document you are reading now, then you should try your current distro. If you want the latest production version, you should try http://hsqldb.org/doc/2.0.

Sometimes, distributions other than http://hsqldb.org/doc/2.0 do not host all available formats. So, if you can't access the format that you want in your current distro, you have no choice but to use the newest production version at http://hsqldb.org/doc/2.0.

Table 1. Available formats of this document

format	your distro	at http://hsqldb.org/doc/2.0
Chunked HTML	index.html	http://hsqldb.org/doc/2.0/hsqlodbc/
All-in-one HTML	hsqlodbc.html	http://hsqldb.org/doc/2.0/hsqlodbc/hsqlodbc.html
PDF	hsqlodbc.pdf	http://hsqldb.org/doc/2.0/hsqlodbc/hsqlodbc.pdf

If you are reading this document now with a standalone PDF reader, the your distro links may not work.

Purpose

This document is directed to users of a production ODBC Driver, not to those who wish to either modify the driver itself or make a custom driver build.

Those who wish to make custom driver builds should attend to the *.txt files in the doc directory of a source distribution or of a Subversion export or checkout.

License

The HyperSQL ODBC Driver, hsqlodbc, is licensed under LGPL v2. The file `license.txt`, included with all of our distributions, includes the full text of the license.

Credits

This software is the product of the work of many individuals and organizations, and also benefits from the use of several open source libraries and utilities. I will here just list the significant contributors to the current HyperSQL ODBC

Driver as well as its predecessors. Please see the file `notice.txt` which we supply with all of our distributions, for more details about these contributors.

Chapter 1. Driver configuration on UNIX

Blaine Simpson, The HSQL Development Group

$Revision: 3462 $
$Date: 2010-02-16 14:32:35 -0500 (Tue, 16 Feb 2010) $

Installation

Install the RPM or package, or extract the zip or tar file as root user from the root directory.

Configuration

See the Driver Settings chapter about individual driver runtime settings.

Our binary distributions contain drivers that work with unixODBC, and that is what this page assumes you are using. You can build drivers that work with iodbc instead. If you do that, see the iodbc docs about how to manage installed drivers and DSN definitions.

The unixODBC graphical program "ODBCConfig" just does not work for any driver I have ever tried to add. If the same applies to you, you will need to edit the files

- `/etc/unixODBC/odbc.ini` Driver definitions

- `/etc/unixODBC/odbcinst.ini` Global DSN definitions

- `$HOME/.odbc.ini` Personal DSN definitions

Depending on your UNIX or unixODBC distribution, your etc config files may be directly in `/etc/` instead of in the `unixODBC` subdirectory.

Chapter 2. Driver configuration on Windows

Blaine Simpson, The HSQL Development Group

$Revision: 2890 $
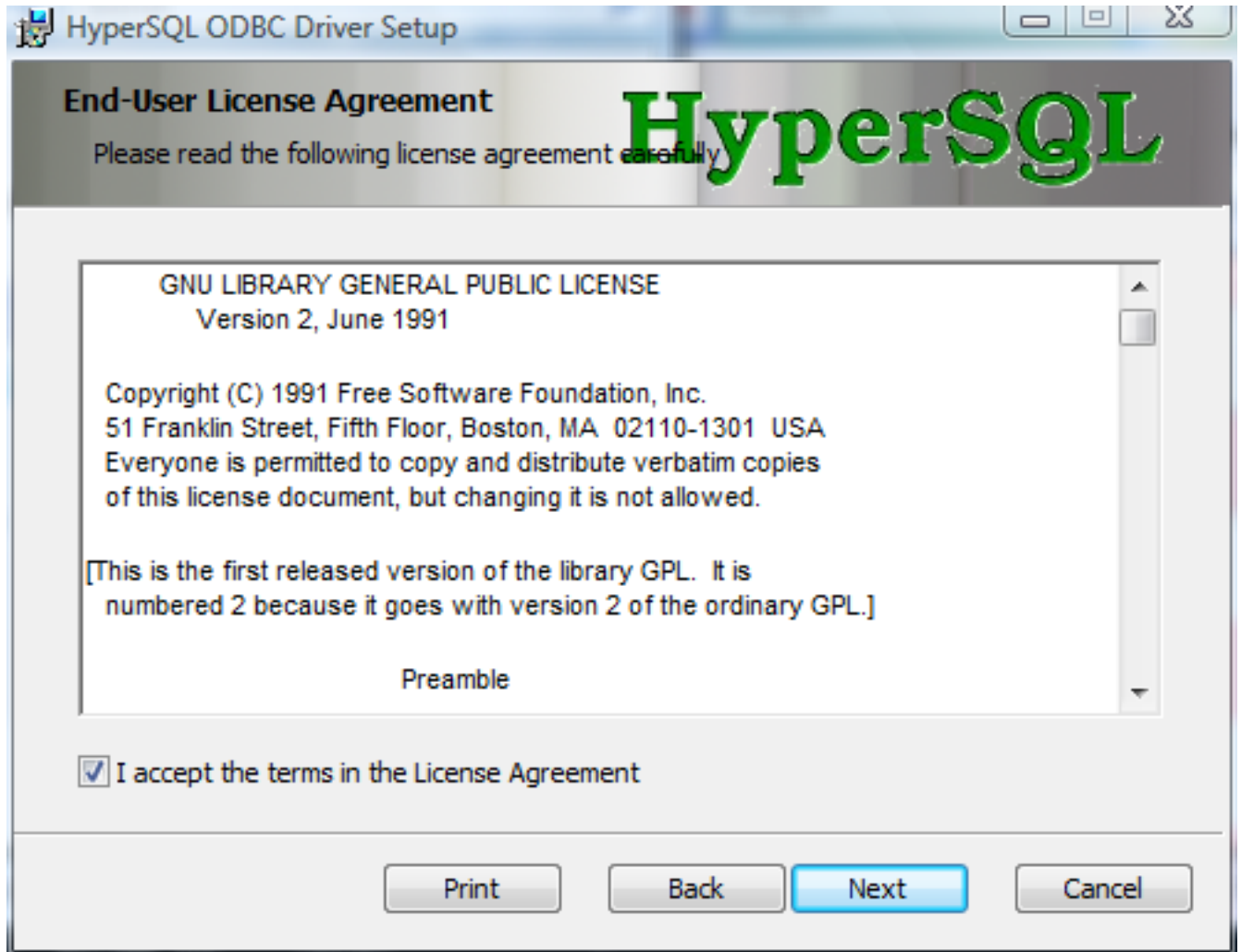$Date: 2009-03-06 20:37:59 -0500 (Fri, 06 Mar 2009) $

Installation
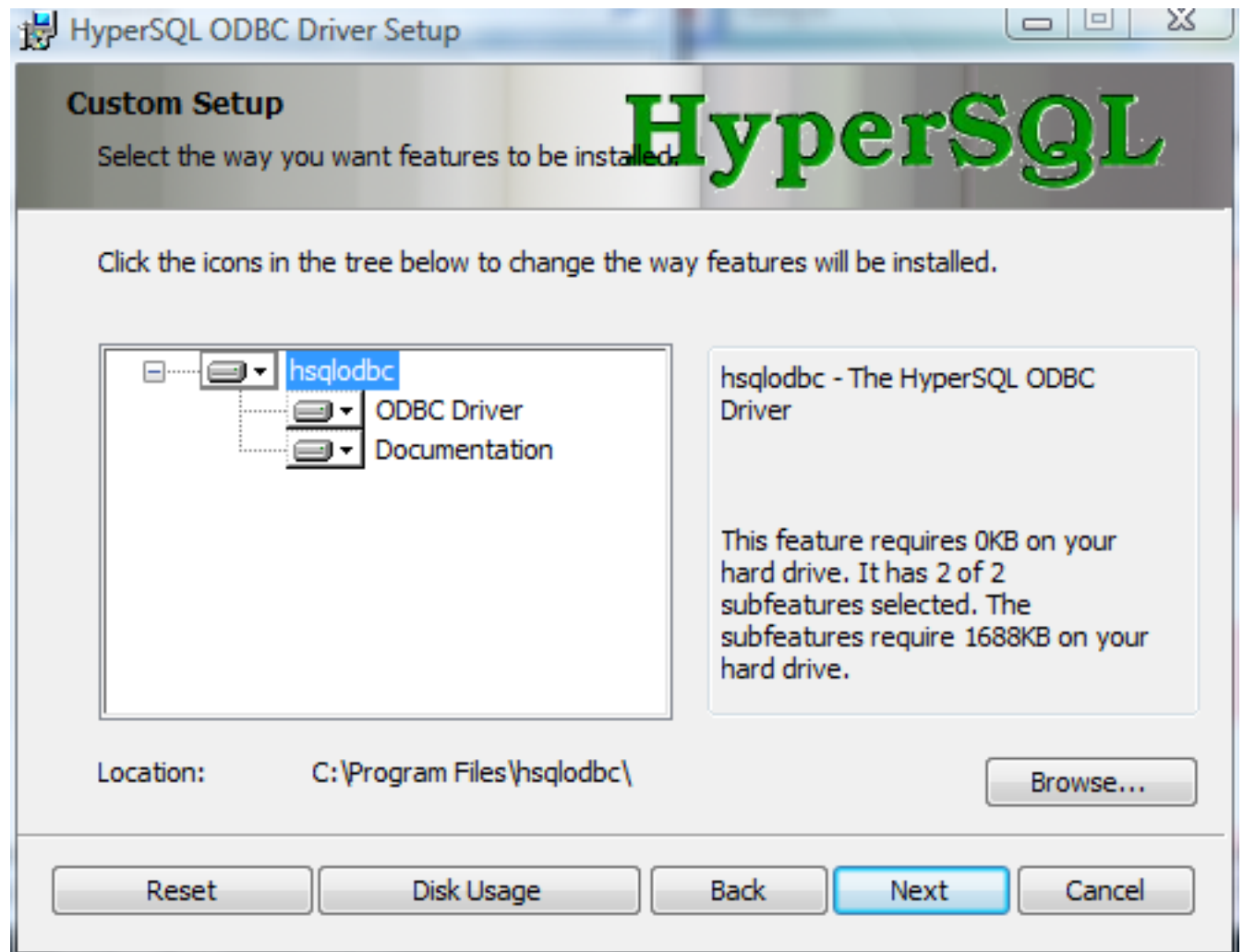
Procedure 2.1. Windows Installer

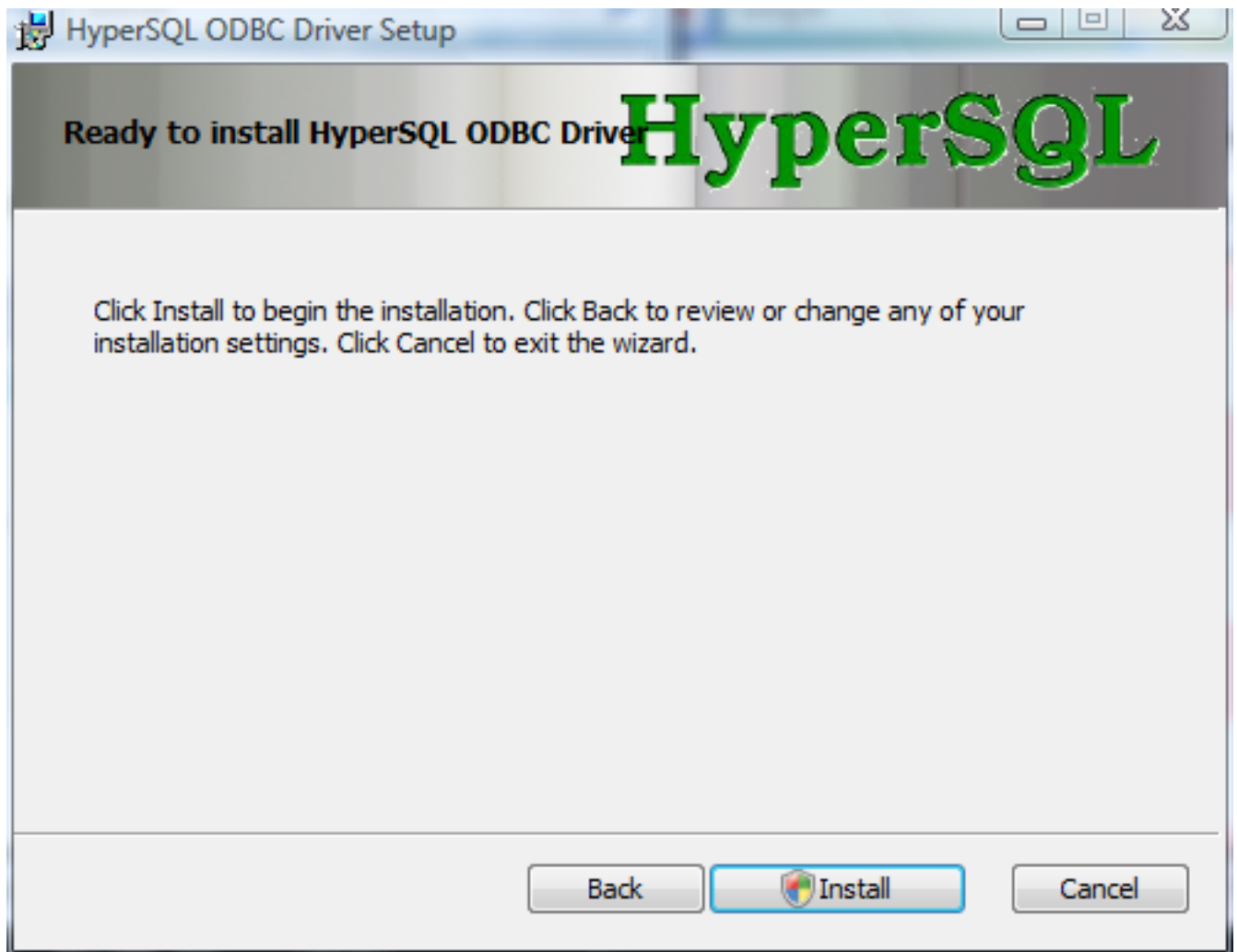1. Execute the provided *.msi file

2. Click Next.

3. Accept the license agreement, then click Next.

4. De-select the Documentation if you don't want it. Select a different installation directory if you don't like the default. Click Next.
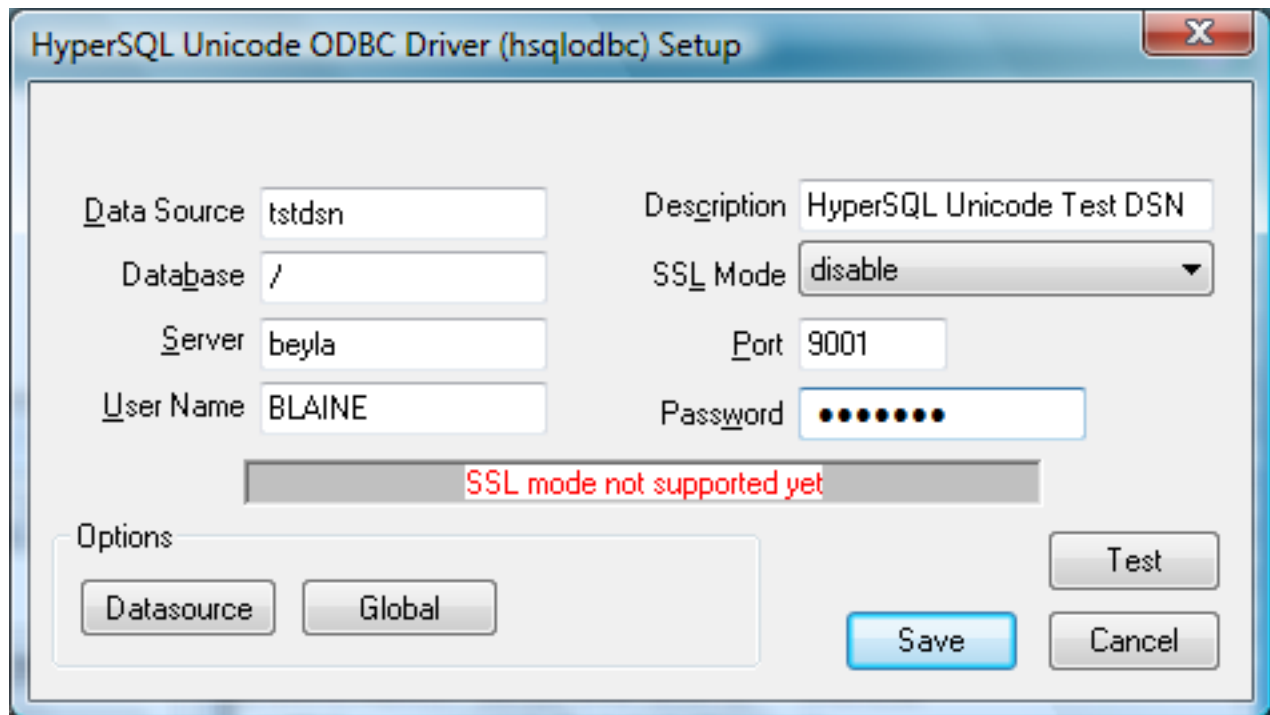
5. Click Install.

6. Click Finish.

Configuration

See the Driver Settings chapter about individual driver runtime settings.

Here are the DSN definition screens captured from a Vista PC. First, the DSN definition screen.
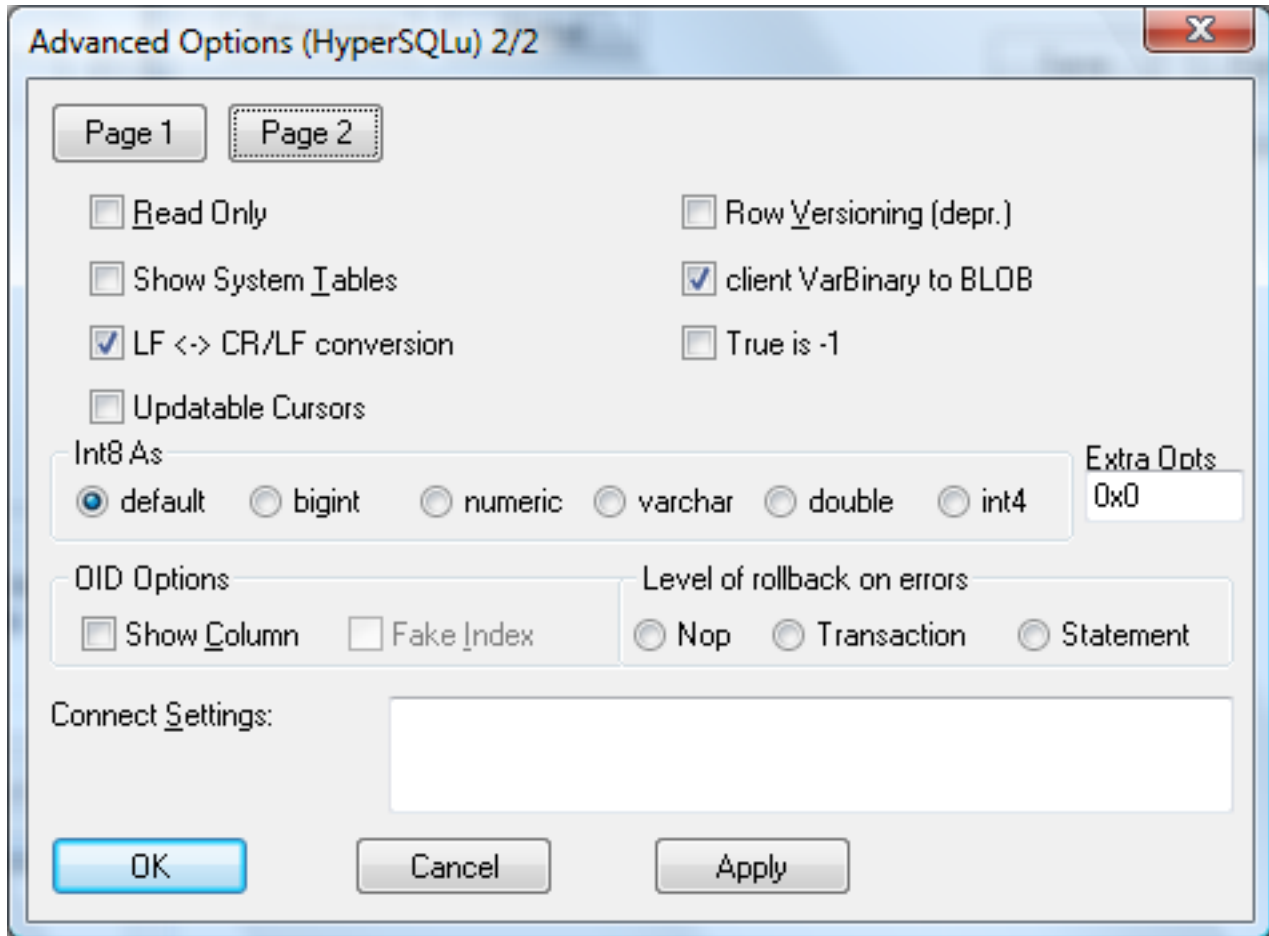
Then option screen 1 of 2.

... and 2 of 2.

Our Windows binary distributions include MSI installers. This page assumes that you are installing with our MSI installer. If you get your HyperSQL ODBC driver with another application, much of what is written here may apply to your distribution.

Our driver is designed to facilitate incorporation into installers for other products as a *Merge Module*.

Replacing Installed Driver

The "Date" column in the ODBC Data Source Administrator shows the driver install date.

Can uninstall by rerunning the same *.msi used to install; or run Control Panel / Programs and Features (Vista) Control Panel / Add/Remove Programs (other)

Fortunately, DSNs for removed drivers are retained! Uh... unfortunately, DSNs can only be removed or edited if a compatible driver is not currently installed. Therefore if you are not replacing a driver with a compatible one, you have better remove your DSNs first.

I see no benefit to using our upgrade batch script, which just passes params to "msiexec".

Just make sure no "running" programs are using the driver, and you do not need to reboot in order to pick up a newly installed driver. Use Control Panel / Administrative Tools / Data Sources (ODBC) to configure your DSNs. (This is the Windows ODBC Data Source Administrator).

ODBC does not allow an empty string for a DSM database name. Therefore, you must specify "/" (without the quotes) to indicate the default database. This setting is on the main Configure page for the DSN.

You may want to turn on one or more of the log files at the Datasource or Global level of the DSN Configure screen.

The logging destination files shown in the Data Source Administrator are wrong, at least in some common cases. The MyLog file gets written in your home directory, if you have one.

Chapter 3. Driver Settings

Blaine Simpson, The HSQL Development Group

$Revision: 2871 $
$Date: 2009-02-25 19:04:38 -0500 (Wed, 25 Feb 2009) $

For some reason, attempts to screen-shot the DSN config screens fail with a Windows system error on Vista. Perhaps security-related. TODO: Try on XP.

This chapter applies to both UNIX and Windows. The setting heading includes the descriptive name as shown by the Windows ODBC DS Administrator, as well as the real keyword names that UNIX users will use.

The HyperSQL ODBC Driver product consists of two driver variants. You should try to use the *Unicode* variant first, since it supports the later and better ODBC protocol. Use the *ANSI* variant if the Unicode variant won't work for your application. The way you select the driver variant for a DSN is platform-specific. For UNIX, set the DSN setting `Driver` to the key as defined in the uniXODBC config file `/etc/unixODBC/odbcinst.ini`. For UNIX, select the driver after you click Add on the User DSN screen, or switch it afterwards with the DSN's Manage DSN button.

Driver settings can also be made at connection time by just appending keyword abbreviation/value assignment pairs to the ODBC connection string, delimiting each such pair with a semicolon. Base connection Strings are language-dependent, but you always append a String in this form

```
;A0=0;B9=1
```

See the Settings List appendix for a concise list of the abbreviations you may use. The appendix also shows the default values for settings (this is really only necessary for UNIX users, since the Winodws DSN manager always shows the current effective value).

Runtime Driver Settings

Database	ODBC does not allow an empty string for a DSM database name. Therefore, you must specify DSN database name of "/" (without the quotes) to indicate the default database
Recognize Unique Indexes	
Cancel as FreeeStmt	*Find out what this experimental feature is for.*
MyLog	Enables fairly verbose runtime logging to the indicated file. With value 1 will write coded mylog() messages to the MyLog file. With value 2 will write both mylog() and inolog() messages to MyLog file.
CommLog	Enables runtime communiction logging to the indicated file. With value 1, will write coded qlog() messages to the CommLog.
Unknown Sizes	This controls what SQLDescribeCol and SQLColAttributes will return as to precision for the *variable* data types when the precision (for example for a column) is unspecified. For the recommended `sql_enforce_strict_size` mode, this stting will have no effect.

- Maximum: Always return the maximum precision of the data type.

- Dont Know: Return "Don't Know" value and let application decide.

- Longest: Return the longest string length of the column of any row. Beware of this setting when using cursors because the cache size may not be a good representation of the longest column in the cache.

MS Access: Seems to handle Maximum setting ok, as well as all the others. Borland: If sizes are large and lots of columns, Borland may crash badly (it doesn't seem to handle memory allocation well) if using Maximum size.

Max Varchar	Use this setting only as a work-around for client app idiocy. Generally, the database should enforce your data constraints.
	The maximum precision of the VARCHAR and CHAR types (perhaps others). Set to 1 larger than the value you need, to allow for null terminator characters. *The default is 255 right now. 0 really means max of 0, and we need to change this ASAP so that 0 will mean unlimited.*
	If you set this value higher than 254, Access will not let you index on varchar columns!
Cache Size	When using cursors, this is the row size of the tuple cache. If not using cursors, this is how many tuples to allocate memory for at any given time. The default is 100 rows for either case.
Max LongVarChar	The maximum precision of the LongVarChar type. The default is 4094 which actually means 4095 with the null terminator. You can even specify (-4) for this size, which is the odbc SQL_NO_TOTAL value.
ReadOnly	Whether the datasource will allow updates.
Show System Tables	The driver will treat system tables as regular tables in SQLTables. This is good for Access so you can see system tables.
LF <-> CR/LF conversion	Convert Unix style line endings to DOS style.
Updatable Cursors	Enable updateable cursor emulation in the driver. *Fred will be implementing real Updatable ResultSets.*
Row Versioning	*Will turn on MVRCC currency control mode, once we implement this.*
True is -1	Represent TRUE as -1 for compatibility with some applications.
Int8 As	Define what datatype to report int8 columns as.
Extra Opts	Extra Opts: combination of the following bits.

- 0x1: Force the output of short-length formatted connection string. Check this bit when you use MFC CDatabase class.

- 0x2: Fake MS SQL Server so that MS Access recognizes PostgreSQL's serial type as AutoNumber type.

- 0x4: Reply ANSI (not Unicode) char types for the inquiries from applications. Try to check this bit when your applications don't seem to be good at handling Unicode data.

OID Options	- Show Column: Includes the OID in SQLColumns. This is good for using as a unique identifier to update records if no good key exists OR if the key has many parts, which blows up the backend.

- Fake Index: This option fakes a unique index on OID. This is useful when there is not a real unique index on OID and for apps which can't ask what the unique identifier should be (i.e, Access 2.0).

OID Options

Level of rollback on errors: Specifies what to rollback should an error occur.

- Nop(0): Don't rollback anything and let the application handle the error.

- Transaction(1): Rollback the entire transaction.

- Statement(2): Rollback the statement.
default value is a sentence unit (it is a transaction unit before 8.0).

Connection Settings

The driver sends these commands to the backend upon a successful connection. It sends these settings AFTER it sends the driver "Connect Settings". Use a semi-colon (;) to separate commands. This can now handle any query, even if it returns results. The results will be thrown away however!

Appendix A. Settings List

Blaine Simpson, The HSQL Development Group
$Revision: 2871 $
$Date: 2009-02-25 19:04:38 -0500 (Wed, 25 Feb 2009) $

See the Driver Settings chapter for descriptions and usage details. This appendix just contains a list of the available settings.

Table A.1. Settings List

Keyword	Abbrev.	Default Val.	Purpose
Description	N/A		Data source description
Servername	N/A	[required]	Name of Server
Port	N/A	9001	HyperSQL Server Listen Port
Username	N/A	[required]	User Name
Password	N/A	[required]	Password
Debug	B2	0	MyLog logging level
Fetch	A7	100	Fetch Max Count *Test to test if this applies to EXECDIRECT and/or prepared queries*
Socket	A8	4096	Socket buffer size
ReadOnly	A0	No/0	Read Only
CommLog	B3	0	Log communications to log file
UniqueIndex	N/A	1	Recognize unique indexes
UnknownSizes	A9	0 [= max prec. for type]	Unknown result set sizes
CancelAsFreeStmt	C1	0	Cancel as FreeStmt
UnknownsAsLongVarchar	B8	0	Unknowns as LongVarchar
BoolsAsChar	B9	0	Bools as Char
MaxVarcharSize	B0	255	Max Varchar size. Value of 0 will break everything. We will be changing 0 to mean *unlimited* and will then change the default to 0.
MaxLongVarcharSize	B1	8190	Max LongVarchar size
RowVersioning	A4	0	Row Versioning
ShowSystemTables	A5	0	Show System Tables
DisallowPremature	C3	0	Disallow Premature
UpdatableCursors	C4	0	Updatable Cursors
LFConversion	C5	1 Windows, 0 UNIX	LF <-> CR/LF conversion
TrueIsMinus1	C6	0	True is -1

Keyword	Abbrev.	Default Val.	Purpose
BI	N/A	0	Datatype to report BIGINT columns as
LowerCaseIdentifier	C9	0	Lower case identifier
SSLmode	CA	disable	SSL mode
AB	N/A		Connection string suffix options

Abbreviations are for use in connection strings.

Appendix B. Code Samples

Code examples of ODBC clients which access a HyperSQL database
Blaine Simpson, The HSQL Development Group
$Revision: 2861 $
$Date: 2009-02-24 19:39:50 -0500 (Tue, 24 Feb 2009) $

The HyperSQL Engine distribution contains these same ODBC client code examples in the `sample` subdirectory.

- Python pyodbc sample   [../verbatim/sample/sample.py]

- PHP ODBC sample   [../verbatim/sample/sample.php]

- Perl DBI/DBD sample   [../verbatim/sample/sample.pl]

- C client sample   [../verbatim/sample/sample.c]

General Index