

## Unknown Title



[ [новости](#) / [+++](#) | [форум](#) | [теги](#) | [🔗](#) ]

[Каталог документации](#) / [Раздел "Базы данных, SQL"](#) / [Оглавление документа](#)



### WebMoney:

WMZ Z294115950220    **YandexMoney:**    **E-gold:**  
WMR R409981405661    41001198119846    5128052  
WME E134003968233

Как только Вы распаковали исходный текст, изучите файлы INSTALL и README.

Сначала определите некоторые переменные среды. Например, пользователь оболочки bash может добавлять следующее в свой файл /etc/profile.

```
# TDS
#
SYBASE=/usr/local/freetds
LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${SYBASE/lib}
export SYBASE LD_LIBRARY_PATH
```

Опции configure могут просматриваться, выполняя ./configure --help. Главное, должна быть указана опция выбора конфигурации --with-tdsver. Теперь соберем пакет и установим его:

```
# ./configure --with-tdsver=4.2 --with-unixodbc=/usr/local
# make
# make install
```

Драйвер FreeTDS ODBC должен быть установлен в /usr/local/freetds/lib. Там должен появиться файл libtdsodbc.so.

## Настройка

### Регистрация SQL Server с TDS

FreeTDS должен знать несколько вещей относительно каждого сервера, с которым Вы хотите работать. Традиционно используется интерфейсный файл. FreeTDS может работать с файлом, называемым freetds.conf, но для упрощения примера применим старый метод. Файл интерфейсов может быть найден в каталоге /usr/local/freetds. Имеется пример для доступа к MS SQL Server:

```
P450
query tcp ether 192.168.1.25 1433
master tcp ether 192.168.1.25 1433
```

Под P450 здесь значится имя машины, на которой выполняется MS SQL Server. Имя может быть каким угодно. По умолчанию MS SQL Server слушает порт 1433.

### Регистрация ODBC-драйвера через unixODBC

unixODBC должен знать ряд данных относительно всех ODBC-драйверов, которые Вы предполагаете использовать. Самый лучший способ этого достичь: применить графическую программу ODBCConfig, которая приходит с unixODBC. Альтернативный метод состоит в том, чтобы использовать команду odbccinst, которая также приходит с unixODBC. Мы сосредоточимся на использовании команды odbccinst. Создайте файл с именем tds.driver.template с несколькими строками, описывающими нужный драйвер:

```
[TDS]
```

```
Description = v0.52 with protocol v4.2
```

```
Driver = /usr/local/freetds/lib/libtdsodbc.so
```

Выполните `odbcinst`, предписывая ему создать запись драйвера, используя файл `tds.driver.template`. Обратите внимание, что Вы должны оставить пробел между параметром `-f` и именем файла шаблона:

```
# odbcinst -i -d -f tds.driver.template
```

### Создание ODBC Data Source Name

Клиенты ODBC обычно будут работать с ODBC Data Source Names (DSN). Самый лучший способ создавать, редактировать и удалять DSN состоит в том, чтобы использовать инструмент `ODBCConfig`.

Снова мы используем команду `odbcinst` вместо него. Это делается потому, что программа `ODBCConfig` достаточно наглядна и понятна, но не все пользователи имеют установленные инструментальные средства `unixODBC GUI`. Снова начнем с создания файла-шаблона под именем `tds.datasource.template`, который содержит некоторые опции типа заданной по умолчанию базы данных и UID. Примечание: Вы будете использовать Ваш собственный `Servename` и `UID`:

```
[P450]
Driver = TDS
Description = Northwind sample database
Trace = No
Servename = 192.168.1.25
Database = Northwind
UID = sa
```

**Примечание:** `Servename` не означает DNS-имя или IP-адрес сервера. Это должно соответствовать серверу из интерфейсного файла (в случае `freets.conf` это значение между `[]`, заголовок желательного раздела сервера).

**Примечание:** предыдущие команды выполнялись как `root` (обозначенный символом `#`), но теперь команда выполняется от имени обычного пользователя. Это значительно. Все пользователи системы совместно используют `FreeTDS` и `ODBC`-драйверы, но каждый пользователь имеет собственный список DSN (имеются исключения, но здесь они не актуальны). Так что создайте DSN как тот пользователь, который собирается использовать это:

```
$ odbcinst -i -s -f tds.datasource.template
```

### Тестирование

`unixODBC` приходит с рядом инструментальных средств, которые позволяют Вам проверять работоспособность пакета. Используем инструмент командной строки `isql`. Он позволяет представлять на рассмотрение DSN команды (обычно это инструкции SQL) и принимать результаты:

```
$ isql -v P450 sa MyPassword
SQL> select contactname from customers
```

Вы должны видеть сообщение о соединении и подсказку SQL. Если этого не наблюдается, либо проблема с конфигурацией, либо задана неправильная пара значений ID и PWD.

Теперь попробуйте простую команду SQL:

```
SQL> SELECT contactname FROM customers
```

## Шаблон драйвера

Создан шаблон драйвера, чтобы позволить программистам легко разрабатывать новые драйверы. Это содержит некоторые файлы типа `make` и все функциональные заголовки, требуемые для совместимости с ODBC 3.5.1. Некоторые функции имеют код, но большая часть оставлена для конкретной реализации.

## Типовые драйверы

Двоичная переносимость драйверов для Linux ODBC когда-нибудь будет достигнута, но ODBC пока довольно плохо реализован под Linux в это время, и некоторые драйверы могут иметь проблемы. Это явление постепенно должно уменьшиться, поскольку ODBC становится более используемым на Linux.

Сертифицированные драйверы

## 8 ODBC и проект unixODBC

### Что такое проект unixODBC?

Цели проекта unixODBC состоят в том, чтобы разрабатывать и поддерживать unixODBC, который призван стать окончательным стандартом для ODBC на не-MS Windows системах. Это должно включить поддержку GUI для KDE и GNOME. Платформы Microsoft включают много расширений в спецификацию ODBC, эти расширения также обеспечиваются unixODBC.

Группа разработчиков unixODBC достигла лучших на сегодняшний день результатов под Linux. Весь проект unixODBC распространяется согласно лицензиям GPL или LGPL. Наличие LGPL для библиотек позволяет применять их в коммерческом программном обеспечении с закрытыми исходными текстами.

## ODBCConfig

Это приложение GUI, подобное MS ODBC Administrator. Это позволяет любому пользователю легко конфигурировать свои (пользовательские) источники данных. Пользователь root может также добавлять, удалять и настраивать общесистемные источники данных. Опции Driver меняются в зависимости от СУБД. ODBCConfig вызывает код Driver Config, чтобы позволить драйверу запрашивать пользователя относительно специфических опций.

Пользователь root может добавлять или удалять драйверы, используя ODBCConfig. Это скорее регистрация и отмена регистрации драйвера потому, что никакие файлы фактически не будут скопированы или перемещены, зато модифицируется информация системы ODBC.

Все зарегистрированные драйверы будут представлены пользователю на выбор при создании нового источника данных.

## DataManager

DataManager мог бы называться Data Explorer. Это приложение GUI, которое позволяет пользователю просматривать и управлять источниками данных. Более точно: оно позволяет пользователю просматривать источники данных, начиная с имени источника и заканчивая таблицами, столбцами и индексами. Это также включает SQL-редактор, который позволяет пользователю представлять на рассмотрение источнику данных SQL-запросы и просматривать результаты.

## DriverManager

ODBC-приложения делают ODBC-вызовы через DriverManager. DriverManager выполняет ряд задач для приложения:

- Обеспечение загрузки и выгрузки соответствующего драйвера
- Задачи проверки правильности
- Отображение данных и вызовов 3.5 на 3.0 и на 2.0

Большинство обращений к DriverManager передается на загруженный драйвер, чтобы он их дальше обработал сам.

Некоторые преимущества от использования ODBC DriverManager включают;

- Переносной (мобильный) код доступа к данным
- Связывание во время выполнения с источником данных (Data Source)
- Способность легко изменять источник данных

## Использование unixODBC с wine

Те, кто работает с wine (<http://www.winehq.org>), некоторое время назад добавили поддержку ODBC. Это выполнено через ODBC.DLL в коде wine, который направляет обращения из среды эмуляции к библиотеке libodbc.so в \*nix. Это работает через unixODBC с различными приложениями.

Когда Crystal Reports запущен под WINE и использует ODBC, появляется сообщение о том, что строка соединения недопустима (сообщение об ошибке: "Unable to connect: Incorrect log on parameters").

Дело в том, что Crystal Reports обращается к реестру Windows, чтобы найти там настройки драйвера для DSN в секции ODBC.INI реестра. Так как установка драйвера находится в файловой системе Linux, ничего хорошего он там не находит, что и приводит к ошибке.

Чтобы исправить положение надо ввести соответствующую запись в реестр. Новая запись может располагаться в дереве HKEY\_CURRENT\_USER или в HKEY\_LOCAL\_MACHINE и выглядит так: Software->ODBC->ODBC.INI-><MYDSN>-> Driver="not blank", где <MYDSN> определяет соответствующий DSN. Установка для драйвера не имеет значения, за исключением того, что это не должно быть пустой строкой. Такая схема применима и для других программ.

## Использование unixODBC с wine

Те, кто работает с wine (<http://www.winehq.org>), некоторое время назад добавили поддержку ODBC. Это выполнено через ODBC.DLL в коде wine, который направляет обращения из среды эмуляции к библиотеке libodbc.so в \*nix. Это работает через unixODBC с различными приложениями.

Когда Crystal Reports запущен под WINE и использует ODBC, появляется сообщение о том, что строка соединения недопустима (сообщение об ошибке: "Unable to connect: Incorrect log on parameters").

Дело в том, что Crystal Reports обращается к реестру Windows, чтобы найти там настройки драйвера для DSN в секции ODBC.INI реестра. Так как установка драйвера находится в файловой системе Linux, ничего хорошего он там не находит, что и приводит к ошибке.

Чтобы исправить положение надо ввести соответствующую запись в реестр. Новая запись может располагаться в дереве HKEY\_CURRENT\_USER или в HKEY\_LOCAL\_MACHINE и выглядит так: Software->ODBC->ODBC.INI-><MYDSN>-> Driver="not blank", где <MYDSN> определяет соответствующий DSN. Установка для драйвера не имеет значения, за исключением того, что это не должно быть пустой строкой. Такая схема применима и для других программ.

## Установка PHP с unixODBC

Эта процедура установки основана на apache 1.3.12 и PHP 3.0.16, но идентична по крайней мере для apache 1.3.6 и php 3.0.9. PHP4 имеет опцию выбора конфигурации, чтобы использовать unixODBC, но она, к сожалению, не всегда работает должным образом.

1. В каталоге Apache выполните следующую команду:

```
./configure --prefix=/www
```

Плюс любая другая локальная конфигурация, в которой Вы нуждаетесь.

2. Создайте файл odbc.h в /usr/local/include содержащий следующие строки:

```
#include <sql.h>
#include <sqlext.h>
#include <odbcinst.h>
```

Замените здесь /www на каталог, в который установлен apache.

3. Перейдите в каталог PHP и определите следующие переменные среды:

```
CFLAGS="-I/usr/local/include"
LDFLAGS=
CUSTOM_ODBC_LIBS="-L/usr/local/lib -lodbc"
```

Не забудьте экспортировать эти переменные!

```
export CFLAGS LDFLAGS CUSTOM_ODBC_LIBS
```

4. Настройте PHP командой:

```
./configure --with-apache=../apache_1.3.12 --with-custom-odbc=/usr/local \
--enable-track-vars
```

Плюс любая другая локальная конфигурация, в которой Вы нуждаетесь, затем:

```
make
make install
```

5. Вернитесь назад в каталог apache и введите:

```
./configure --prefix=/www --activate-module=src/modules/php3/libphp3.a
make
make install
```

6. Вернитесь в каталог PHP и скопонируйте:

```
cp php3.ini-dist /usr/local/lib/php3.ini
```

Вы можете редактировать файл /usr/local/lib/php3.ini, чтобы установить опции PHP. Отредактируйте Ваш файл httpd.conf или srm.conf и добавьте:

```
AddType application/x-httpd-php3 .php3
```

### Поддержка Apache DSO

Дополнительная проблема была найдена, когда Вы формируете PHP как динамический общедоступный объект с apache. Поскольку по умолчанию unixODBC формируется с поддержкой потоков, а вот apache нет, если они не связаны вместе в процессе построения, а встречаются только во время выполнения, такая комбинация будет взрывоопасной.

Имеются два способа обхода проблемы. Необходимо определить и экспортировать следующие переменные среды перед формированием apache:

```
export CFLAGS=-D_REENTRANT
export LDFLAGS=-lpthread
./configure --prefix=/www
```

Или пересоберите unixODBC без поддержки потоков, конфигурируя его так:

```
./configure --enable-threads=no
```

В обоих случаях не забывайте удалять файл config.cache из корневого каталога Apache или unixODBC

## UnixODBC и 2000 год

Из-за того, как спроектирован ODBC, не имеется никаких прямых ссылок к дате или времени, кроме как в структурах SQL\_DATE, SQL\_TIME и SQL\_TIMESTAMP. Любая информация о годе внутри них сохранена с четырьмя цифрами, включая столетие.

Поскольку ничего из вышеупомянутой информации не сгенерировано кодом внутри unixODBC, каких бы то ни было проблем с 2000 годом в рамках кода unixODBC быть не должно. Могут быть сложности у конкретной СУБД или операционной системы, но они не имеют отношения к пакету unixODBC.

## Библиотеки в unixODBC

INI-библиотека представляет собой библиотеку функций нужных, чтобы помочь в разработке кода для взаимодействия с INI-файлами. Будущая расширенная версия этой библиотеки скорее всего станет стандартной библиотекой Linux, что позволит рассматривать INI-файлы как стандартный формат новых файлов конфигурации в ОС Linux.

Log-библиотека представляет собой комплект функций нужных, чтобы создать средство регистрации событий. Это используется odbinst и unixODBC Driver Manager. Планируется поддержка системы регистрации событий в UNIX.

List-библиотека представляет собой набор маленьких и простых функций для создания, управления и удаления связанных списков. Эти функции служат, чтобы усилить код для работы связанных списков в unixODBC и обеспечить основу для поддержки курсора.

Библиотека odbinst обеспечивает приложения и другие компоненты unixODBC способностью к чтению и записи системной информации ODBC. Причем неважно, как или где эта информация сохранена. Это осуществляет основной API из MS и добавляет несколько функций для удобства. Одно из таких расширений: добавление нескольких API для работы с ODBC-программами Config в среде KDE/Qt.

Библиотека Tree была создана, чтобы хранить анализируемый код SQL. Она также должна использоваться в будущих версиях, чтобы сохранить синтаксис SQL в намного более реалистичном виде.

Библиотека SQL реализует сильно облегченный обработчик SQL, который использует текстовые файлы для хранения данных. Это дополнено ODBC-драйвером, чтобы создать очень открытую СУБД для маленьких наборов данных. Строго рекомендуется, чтобы все использование SQL было сделано через ODBC-драйвер: это позволит источнику данных превратиться в нормальную СУБД без того, чтобы перетранслировать исходный код.

Разработка SQL управлялась желанием включить поддержку ODBC в KDE и в соответствии с необходимостью создания полноценной СУБД, чтобы позволить использование. SQL для этого вполне достаточна.

Библиотека SQP представляет собой SQL-анализатор, основанный на lex/yacc. Это было выдрано из SQL так, чтобы общий анализатор SQL можно было бы обеспечивать для всех частей unixODBC, в частности, в драйверах и источниках данных, основанных не на SQL. Текущее понимание синтаксиса SQL очень ограничено, но это должно быть расширено в будущем.

## Утилиты командной строки

odbinst представляет собой инструмент, который был создан для помощи тем, кто разрабатывает скрипты и RPM-пакеты для драйверов, чтобы иметь возможность легко создавать и удалять записи в файлах odbc.ini и odbinst.ini. Этот инструмент командной строки связан с разделяемой библиотекой (libodbinst.so) и является частью компонента odbinst в unixODBC.

isql представляет собой инструмент, который позволяет пользователю выполнять SQL-команды в пакете или в интерактивном режиме. Это имеет некоторые интересные опции типа опции, чтобы генерировать вывод, в виде таблицы на HTML.

## Использование IBM DB2 с unixODBC

Этот короткий раздел детализирует то, как соединиться с IBM DB2 из Linux через unixODBC. Этот документ первоначально был ориентирован на DB2 версии 6.1, но все работает и с версией 7.1, хотя есть некоторые отличия, они описаны в конце раздела.

Сначала установите клиентский пакет доступа DB2 от IBM, это установит IBM CLI драйвер, который работает с unixODBC как ODBC-драйвер.

При использовании клиентского пакета доступа, определите клиентский доступ под неким именем, например, db2inst1, а затем создайте запись в файле odbinst.ini:

```
[DB2]
Description = DB2 Driver
Driver = /usr/IBMDB2/V6.1/lib/libdb2.so
FileUsage = 1
DontDLClose = 1
```

Параметр DontDLClose требуется, чтобы избежать проблемы в драйвере, когда он динамически загружен.

Теперь настройте файл odbc.ini примерно так:

```
[sample]
Description = Test to DB2
Driver = DB2
```

Затем, когда происходит соединение, Вы ДОЛЖНЫ ИМЕТЬ переменную среды DB2INSTANCE, установленную на корректный образец db2, например:

```
export DB2INSTANCE=db2inst1
isql -v sample db2inst1 ibmdb2
```

## Версия 7.1

Единственное различие сводится к имени пакета, используемого для доступа. CLI-драйвер теперь должен быть найден в DB2 Personal Developer's Edition V7.1.

Использование версии 7.1 формирует путь к библиотеке:

```
Driver = /usr/IBMdb2/V7.1/lib/libdb2.so
```

Вы также должны добавить "/usr/IBMdb2/V7.1/lib" в /etc/ld.so.conf.

Другая полезная подсказка. При использовании PHP+DB2 под AIX через unixODBC к IBM-серверу (4 процессора, мультитядра и т.д.), возникало сильное замедление работы. Имеется ошибка в стеке TCP/IP IBM. После изучения странички [http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/document.d2w/report?last\\_page=list.d2w&fn=1006553](http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/document.d2w/report?last_page=list.d2w&fn=1006553) и выполнения команд

```
$ su - db2inst1
$ db2set DB2TCPCONNMGERS=1
```

был достигнут поистине огромный рост производительности (в десятки раз!).

# unixODBC и Sybase/MS SQL Server

## Введение

Этот документ поможет пользователям unixODBC обратиться к MS SQL Server, а также к Sybase SQL Server из клиентов unixODBC.

Будем считать, что Вы имеете недавнюю версию unixODBC на своей системе и где-то выполняется сервер. SQL Server должен позволять удаленный доступ и поддерживать соединения, использующие TCP/IP. Конечно, нужны также идентификатор для входа в систему и пароль.

## Подготовка

MS SQL Server был порожден партнерством между Microsoft и Sybase. Впоследствии эти два сервера остались очень похожими. Фактически настолько похожими, что протокол связи, используемый клиентским программным обеспечением, чтобы связаться с этими серверами является фактически идентичным. Этот протокол называется Табличным Поток Данных (Tabular Data Stream, сокращенно TDS).

Brian Bruns создал проект открытых исходных текстов, призванный обеспечить реализацию протокола TDS для всех желающих. Проект называется FreeTDS и может быть найден на сайте <http://www.freetds.org>.

## Что с ним делать

FreeTDS приходит с несколькими интерфейсами TDS с ODBC. ODBC-драйвер, включенный в FreeTDS, работает хорошо для типичных запросов, типа запроса данных. Однако, этот ODBC-драйвер не поддерживает менее используемые функции каталога ODBC. FreeTDS хорошо работает с unixODBC, но Вы скачайте самую свежую версию, например, с <ftp://ibiblio.org/pub/Linux/ALPHA/freetds>.

## Установка

Наиболее вероятно, что Вы скачали исходный текст, и этой точки зрения я буду придерживаться. Обычно установка из исходных текстов представляет собой проблему по причине отсутствия необходимых инструментов, но почти версии Linux их имеют. Распакуйте дистрибутив следующими командами:

```
# tar zxvf freetds-0.52.tar.gz
# cd freetds-0.52
```

СУБД	Уровень ODBC	Устанавливается в odbcdrivers.ini	DriverConfig	Замечания
DB2 UDB для iSeries	<a href="http://www-1.ibm.com/servers/eserver/iseries/access/linux">http://www-1.ibm.com/servers/eserver/iseries/access/linux</a>	3.5	Да	Да
Microsoft SQL Server и Microsoft Access DB2	<a href="http://www.ibm.com/db2">http://www.ibm.com/db2</a>	3.5	No	Вы можете использовать Bridge для раф источником данных ODBC на Windows-т

Oracle	<a href="http://www.oracle.com">http://www.oracle.com</a>	3.5	Да	Да
Easysoft ODBC-Bridge	от 2.0 до 3.5	Да	Да	<a href="http://www.easysoft.com/products/oob/ma">http://www.easysoft.com/products/oob/ma</a> Easysoft's ODBC-ODBC Bridge (пробная бесплатная версия) обеспечивает удаленный доступ к любому ODBC-драйверу
DB2	<a href="http://www.starquest.com">http://www.starquest.com</a>	3.5	Нет данных	Нет данных
Sybase	<a href="http://www.sybase.com">http://www.sybase.com</a>	3.5	Да	Нет
Informix	<a href="http://www.informix.com">http://www.informix.com</a>	3.51	Нет	Нет
AdabasD	<a href="http://www.softwareag.com/adabasd">www.softwareag.com/adabasd</a>	3.5	Нет	Нет
Empress	<a href="http://www.empress.com">http://www.empress.com</a>	.	.	.
Postgres	<a href="http://www.postgresql.org">http://www.postgresql.org</a>	2.5	.	Да
MySQL	<a href="http://www.mysql.com">http://www.mysql.com</a>	.	.	.
MiniSQL (он же mSQL)	<a href="http://www.Hughes.com.au">http://www.Hughes.com.au</a>	Смесь из различных уровней. Борется за достижение 3.51. Безуспешно	Да	Да
SQI	Очень ограниченные версии INSERT, SELECT, UPDATE и DELETE	.	Да (используются значения по умолчанию для файла)	Text File Driver (alpha)
News Server	Реализована минимальная функциональность ODBC. Strongly ограниченное, но очень полезное подмножество SQL	.	Да	Расширенная версия Ke Jin's Internet News Server ODBC-драйвера. News-серверы представляют собой источники данных, группы новостей таблицы.
YARD SQL	3.5			
Interbase	3.5	Да	Да	Доступен для Windows, Linux Intel, Solaris, HP-UX, AIX и OS X
RDBMS Linter SQL	3.5	Да	Да	Оригинальный драйвер ODBC портирует UNIX
Ovrimos Webbase	<a href="http://www.ovrimos.gr/download.php">http://www.ovrimos.gr/download.php</a>	3.5	Да	Да
Mimer SQL	<a href="http://www.mimer.com">http://www.mimer.com</a>	3.51	Нет данных	Нет данных
Ingres	<a href="http://www.cai.com">http://www.cai.com</a>	3.51		
Sap DB	<a href="http://www.sapdb.org">http://www.sapdb.org</a>	3.51		

## Установка и настройка Cold Fusion и Interbase под Linux

Что было использовано:

Как и что делалось:

- 1. Под RH Linux 6.2 установлен UnixODBC 1.8.12 с применением для этой цели rpm-пакета с сайта Coding Apes ( <http://grapeape.codingapes.com/unixODBC.xml>). ODBC-диспетчер драйверов, обеспеченный ColdFusion, был заменен на диспетчер из UnixODBC:

```
mv /opt/coldfusion/lib/odbc/libodbc.so /opt/coldfusion/lib/odbc/libodbc.so.bak
ln -s /usr/lib/libodbc.so /opt/coldfusion/lib/odbc/libodbc.so
```

- 2. Под RH Linux 6.1 используйте исходный текст с сайта <http://www.unixodbc.org>. Если Вы не нуждаетесь в графических инструментальных средствах, а для серверов обычно их не ставят), выполните:

```
configure --enable-gui=no
make all
make install
```

Остальное подобно описанному выше.

- 3. Конфигурируйте Interbase ODBC Driver, загрузите Interbase Driver с сайта Easysoft (<http://www.easysoft.com/products/interbase>). Установите его, скопировав файлы в каталог /usr/local/lib. Теперь настройте Ваш файл odbcc.ini в /opt/coldfusion/odbc/odbc.ini примерно так:

```
[MyODBCSource]
Driver=/usr/local/lib/libib6odbc.so.1.0.0
Description=Interbase Driver
Database=192.168.0.1:/Database/MyDB.gdb
User=MyUser
Password=MyPassword
With_Schema=0
Dialect=3
```

## Получение ODBC-драйвера из Informix Client access или SDK, работающего с unixODBC

1. Установите переменную окружения INFORMIXDIR к корню Вашей установки Informix. Далее я считаю, что это /usr/local/informix.
2. Определите сервис online в файле /etc/services.
3. Создайте файл sqlhosts в каталоге \$INFORMIXDIR/etc, который определяет все Ваши серверы базы данных.
4. Удостоверьтесь, что все библиотеки Informix найдены ld. Они размещены в каталоге \$INFORMIXDIR/lib и его подкаталогах (это включает по крайней мере cli и esql). Обычно следует включить каталоги с библиотеками в файл /etc/ld.so.conf и выполнить команду ldconfig от имени root.
5. Определите переменную окружения ODBCINI, чтобы указать на Ваш системный файл odbcc.ini, например: export ODBCINI=/etc/odbc.ini. Дело в том, что эта переменная представляет собой единственный способ управлять тем, где драйвер Informix ищет сведения о конфигурации.
6. Добавьте драйвер Informix в odbccinst.ini без пробелов между именами параметров и их значениями, например:

```
[Informix]
Description=Informix IDS 2000
Driver=/usr/local/informix/lib/cli/libifcli.so
APILevel=1
ConnectFunctions=YYY
DriverODBCVer=03.00
FileUsage=0
SQLLevel=1
smProcessPerConnect=Y
```

7. Определите Informix DSN в odbcc.ini:

```
[miecr]
Driver=Informix
Server=BARNET
Database=miecr
CLIENT_LOCALE=en_us.8859-1
DB_LOCALE=en_us.8859-1
TRANSLATIONDLL=/usr/local/informix/lib/esql/igo4a304.so
```

8. Теперь попробуйте выполнить тестовую команду isql miecr.

Если Вы получаете ошибку "Server %s is not listed as a dbserver name in sqlhosts", это, вероятно, вызвано пробелами в определении драйвера Informix в файле odbccinst.ini или в записи DSN в файле odbcc.ini. Синтаксический анализатор драйвера очень легко ошибается.

Откуда и что можно скачать
<p><b>Формат дистрибутива</b></p> <p>unixODBC на сегодня доступен в формате tar+gzip. Это означает, что Вы должны:</p> <p>Скопировать файл unixODBC*.tar.gz в нужный Вам каталог.</p> <p>Распаковать его, например, командами:</p> <pre>gunzip unixODBC*.tar.gz tar xvf unixODBC*.tar</pre>



Выполнение этих команд создаст каталог unixODBC со всеми исходными файлами внутри.

На момент написания данной работы последней версией была 2.2.6 (архив назывался unixODBC-2.2.6.tar.gz).

### RPM-пакеты

В настоящее время на сайте CodingApes ( <http://grapeape.codingapes.com>) выложено несколько RPM-пакетов. Их также можно найти на <http://edas.visaci.cz/en/unixodbc>. RPM-пакеты для использования с версиями 7.0 и 8.0 Connectiva Linux (<http://www.conectiva.com> можно взять на <http://www.andrefelipemachado.hpg.com.br/linux/index.html>.

Есть также версия для QNX ( <http://www.qnx.com>), которую можно взять с <http://www.unixodbc.org/pub/beta/unixODBC/unixODBC-2.1.1-x86-public.qpr>.

### Версия для FreeBSD

Nick Sayer создал порт unixODBC для FreeBSD. Ее можно найти на сайте <http://www.freebsd.org/ports>.

### Solaris 8 PKG

Palle Haastrup создал пакет для Solaris 8 <http://www.unixodbc.org/pub/beta/unixODBC/binaries/unixodbc-2.1.1-sol8-sparc-local.gz> .

### CVS-доступ

Публичный доступ к CVS access также доступен для всех желающих работать с самыми последними исходными текстами. CVS-доступ есть на [http://sourceforge.net/cvs/?group\\_id=1544](http://sourceforge.net/cvs/?group_id=1544).

Так что Вы можете использовать следующие команды. Когда будет запрошен пароль для anonymous, просто нажмите клавишу Enter:

```
cvs -d:pserver:anonymous@cvs.unixODBC.sourceforge.net:/cvsroot/unixodbc
login
cvs -z3 -d:pserver:anonymous@cvs.unixODBC.sourceforge.net:/cvsroot/unixodbc
co modulename
```

Чтобы сформировать пакет из архива CVS, выполните следующую команду:

```
make -f Makefile.cvs
```

Это будет формировать скрипт configure.

Чтобы сформировать пакет из CVS, Вы нуждаетесь в некоторых инструментальных средствах:

- automake 1.5
- libtool 1.4.2

Они могут быть найдены на web-сайте проекта GNU <http://www.gnu.org> или на его многочисленных зеркалах.

### Построение библиотек и программ

Установка теперь использует стандартный процесс GNU autoconf. Скомандуйте:

```
./configure
>make
make install
```

По умолчанию файлы будут установлены в каталог /usr/local. Как и положено с configure, это расположение может быть изменено опцией prefix:

```
./configure --prefix=/usr/local/unixODBC
```

Это установит каталоги в /usr/local/unixODBC.

Чтобы соответствовать руководящим принципам GNU, файл odbcinstd теперь установлен по умолчанию в {prefix}/etc, что может быть изменено, используя опцию --sysconfdir в вызове configure. Чтобы устанавливать файлы в старое место, надо указать:

```
./configure --sysconfdir=/etc
```

Скрипт будет пытаться находить каталоги, которые необходимы для компонентов GUI, и если они не найдены (то есть, Вы не установили QT), построение пакета продолжится, но без частей GUI.

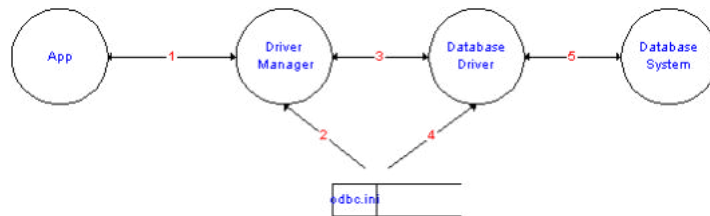
### Зависимости

Компоненты GUI (ODBCConfig и DataManager) требуют Qt не ниже 2.2 ( <http://www.troll.no/dl/qtfree-dl.html>).

Библиотека SQL требует flex и yacc, чтобы анализировать команды SQL.

Дистрибутив в целом зависит от наличия многих из стандартных инструментальных средств разработки, типа компилятора C/C++ (в настоящее время протестирован egcs).

## Начинка unixODBC



Data Flow Diagram (DFD) описывает все ключевые компоненты в окружении Linux ODBC. DFD пронумерована так, чтобы показать, что происходит при **соединении** с системой базы данных.

1. Приложение вызывает функцию (SQLAllocEnv, SQLAllocConnect, SQLConnect) из DriverManager.
2. DriverManager отвечает на SQLConnect драйвером базы данных, который соответствует заданному источнику данных (Data Source Name, DSN).
3. DriverManager загружает драйвер базы данных и просит, чтобы он установил связь с СУБД.
4. Драйвер базы данных может быть должен получить дополнительные опции из DSN, что выполняется им по ходу дела.
5. Драйвер базы данных использует специфическое для конкретной СУБД обращение, чтобы соединиться с базой данных.

## Файл odbc.ini

**ОБРАТИТЕ ВНИМАНИЕ:** файлы odbc.ini и odbcinst.ini являются системной информацией ODBC. Весь доступ (чтение или запись) к этой информации должен проходить через DriverManager или разделяемую библиотеку ODBCINST. Не обращайтесь к этим INI-файлам непосредственно!

Файл odbc.ini хранит список источников данных и любых параметров для каждого из них. Имеются два типа файлов odbc.ini: системный (/etc/odbc.ini) и пользовательский (~/.odbc.ini). По умолчанию, DriverManager будет искать источники данных, используя следующую схему;

1. Используя переменную окружения, чтобы указать путь к пользовательскому файлу odbc.ini. Например, ODBCINI+/.odbc.ini.
2. Используя домашний каталог, заданный в файле /etc/passwd, чтобы найти путь для .odbc.ini.
3. Используя системный файл odbc.ini, который расположен в /etc/odbc.ini.

Файлы odbc.ini предоставляют те же самые функциональные возможности за исключением того, что системные сервисы могут принимать более высокий уровень защиты на системном odbc.ini. Во всем остальном пользовательские и системные файлы совершенно равны.

ini-файлы используют те же самые требования к синтаксису. Имена источников данных охвачены квадратными скобками. Параметры следуют в последующих строках и используют формат `PropertyName=PropertyValue`. Например;

```
[TEST]
driver = /usr/lib/libsybase.so.11
tracefile = /home/pharvey/sql.log
trace = off
database = dbAccounting
```

Каждое имя источника данных должно иметь, по крайней мере, определенный параметр driver. Это позволит DriverManager загружать драйвер, когда сделан запрос на соединение. Единственные параметры, с которыми работает сам DriverManager: DRIVER, TRACEFILE и TRACE. Другие параметры будут интерпретироваться уже драйвером. Секция [ODBC Data Sources] поддерживается в каждом файле odbc.ini.

## Файл odbcinst.ini

Этот ini-файл просто перечисляет все установленные драйверы. Синтаксис прост: имя, сопровождаемое параметром, который сообщает имя файла драйвера:

```
[Sybase 11]
Comment = Super Sybase Server
Driver = /usr/lib/libsybase.so.11
Setup = /usr/lib/libsybaseS.so.11
FileUsage = 1
```

Имя файла драйвера (в данном случае /usr/lib/libsybase.so.11) должно быть уникально. Дружественное имя (то есть Sybase 11) также должно быть уникально.

Параметр Setup указывает на общедоступную (разделяемую) библиотеку, содержащую функции, которые нужно вызвать ODBC Config. ODBC Config вызовет эту библиотеку, чтобы получить специфические для конкретного драйвера имена параметров в ходе конфигурации источника данных. Если программа ODBC Config не может найти или использовать этот файл, она примет некоторые значения по умолчанию. Этот файл должен изменяться только разделяемой библиотекой ODBCINST или утилитой командной строки odbccinst.

## Графическое наполнение

Спецификации ODBC включают два случая, в которых GUI был бы особенно полезен. Проект LinuxODBC находится в процессе реализации GUI для обоих случаев. Случаи это такие:

### 1. Настройка

ODBC Config должен позволить пользователю настроить параметры, специфичные для конкретного драйвера. В MS это обработано через ODBC Admin, вызывающий драйвер и передающий ему дальнейшую инициативу. LinuxODBC не будет вынуждать драйвер делать такую работу. Вместо этого, ODBC Config вызовет библиотеку Drivers Setup, чтобы получить простой список реквизитов. ODBC Config обработает представление и сохранение этих реквизитов. Этот подход делает создание библиотек Driver Setup ОЧЕНЬ простым потому, что они должны обеспечить только список опций.

### 2. Ввод логина

DriverManager, вероятно, придется запрашивать пользователя относительно большего количества информации для входа в систему. С поддержкой графики это вполне реально.

## unixODBC без GUI

### Цель

Много людей используют unixODBC, но по ряду причин не формируют конфигурацию GUI и соответствующие утилиты (ODBCConfig и DataManager).

### Что такое ini-файл?

ODBC сначала появился внутри Windows 3.0. В это время Windows использовала .ini-файлы, чтобы хранить информацию о конфигурации. Это текстовые файлы, содержащие данные в следующем формате:

```
[section1]
entry1 = value
entry2 = value

[section2]
entry1 = value
entry2 = value
...
```

С появлением Windows NT, эти ini-файлы были заменены реестром, но API, чтобы обратиться к ним из ODBC, остался тем же самым. Windows имеет две функции в odbccinst.dll, которые позволяют приложениям и драйверам сделать запрос и изменять эти файлы: SQLGetPrivateProfileString и SQLPutPrivateProfileString.

Как часть репродуцирования ODBC-среды на платформах не-Windows, unixODBC обеспечивает тот же самый формат ini-файлов и функциональные возможности с помощью библиотеки libodbcinst.

### Система и пользователь

ODBC делает различие между двумя типами ini-файлов. Системные ini-файлы разработаны, чтобы быть доступными, но не изменяемыми любым пользователем, а ini-файлы пользователя относятся только к какому-то специфическому пользователю и могут изменяться им.

Системные ini-файлы: odbccinst.ini и odbcc.ini (внимание: точек в начале имени нет!), а пользовательский: ~/.odbc.ini в домашнем каталоге пользователя (вот этот уже с точкой: скрытый).

Системный файл odbccinst.ini хранит информацию относительно ODBC-драйверов, доступных всем пользователям, а файл odbcc.ini содержит информацию относительно DSN, доступных всем пользователям. Эти системные DSN полезны для приложения типа web-серверов, которые не могут выполняться как реальный пользователь и соответственно не будут иметь домашнего каталога, чтобы хранить в нем файл .odbc.ini.

Хороший пример этого: Apache и PHP с поддержкой ODBC. Когда http-сервер запущен, он от имени root вызывает SQLAllocEnv. Позже он меняет свою принадлежность на иного пользователя и вызывает SQLConnect. Если DSN не является системным, сервер потерпит неудачу.

## FILEDSN

ODBC 3 также имеет третий сорт DSN, файловый DSN. Они сохраняют информацию соединения в файле, который может быть доступен любому пользователю. UnixODBC в настоящее время не поддерживает файловый DSN, но такая поддержка планируется в ближайшее время, поскольку такой тип DSN весьма полезен. Он больше распространен в NT, чем в UNIX по причине того, такие файлы позволяют хранить данные на центральном сервере, который является доступным всем рабочим станциям, что избавляет от настройки всех систем сразу.

## Почему не vi?

Все файлы конфигурации, необходимые unixODBC представляют собой простые текстовые файлы, так что не имеется никакой причины, из-за которой Вы не можете использовать Ваш любимый текстовый редактор для правки файлы.

Однако, начиная с версии beta 1.6, расположение системных файлов odbccinst.ini и odbcc.ini определено скриптом выбора конфигурации. Заданное по умолчанию расположение: /usr/local/etc, и если префикс определен, расположение будет {prefix}/etc. Расположение системных файлов можно поменять и радикально указанием опции --sysconfdir=DIR, например, для обратной совместимости с версиями до pre 1.6 попробуйте:

```
./configure --sysconfdir=/etc
```

## Что входит в эти файлы?

### odbcinst.ini

Он содержит заголовок раздела, который обеспечивает имя для драйвера, в примере ниже это будет PostgreSQL для драйвера Postgres. Следующие строки содержат описание и затем важные биты. Пути Driver и Setup указывает на ODBC-драйвер и установку lib. Установка lib используется, когда Вы нажимаете на Add в ODBCConfig, чтобы добавить новый DSN. Наиболее интересна запись Driver. Это библиотека, которую диспетчер драйверов будет загружать динамически, когда SQLConnect или SQLDriverConnect запрашивают этот DSN. Если это указывает на неправильное место, DSN не будет работать. Если dlopen() не сработает, DSN также не будет работать. Запись fileusage дописывается программой odbcinst, так что в текстовом редакторе добавьте ее сами.

```
[PostgreSQL]
Description = PostgreSQL driver for Linux & Win32
Driver = /usr/local/lib/libodbcpsql.so
Setup = /usr/local/lib/libodbcpsqlS.so
FileUsage = 1
```

## Шаблоны

odbcinst должен быть обеспечен файлом шаблона. Если Вы добавляете драйвер для вышеупомянутой записи, файл шаблона содержал бы следующее:

```
[PostgreSQL]
Description = PostgreSQL driver for Linux & Win32
Driver = /usr/local/lib/libodbcpsql.so
Setup = /usr/local/lib/libodbcpsqlS.so
```

Вы вызвали бы odbcinst со следующими параметрами:

```
odbcinst -i -d -f файл_шаблона
```

Параметры odbcinst имеют такой смысл:

- i Установить
- d драйвер
- f имя файла с шаблоном

## Потоки

Начиная с версии 1.6, если диспетчер драйверов был сформирован с поддержкой потоков, Вы можете добавлять другую запись к каждой записи о драйвере:

```
[PostgreSQL]
Description = PostgreSQL driver for Linux & Win32
```

```
Driver = /usr/local/lib/libodbcpsql.so
Setup = /usr/local/lib/libodbcpsqlS.so
Threading = 2
```

Эта запись изменяет заданный по умолчанию уровень преобразования в последовательную форму потока.

#### [.]odbc.ini

Содержание файлов odbc.ini немного сложнее, но они все равно следуют формату файла odbcinst.ini. Они усложнены каждым драйвером, требующим различных записей. Записи могут быть добавлены таким же образом, используя odbcinst или текстовый редактор. Пример типичной записи:

```
[PostgreSQL]
Description = Test to Postgres
Driver = PostgreSQL
Trace = Yes
TraceFile = sql.log
Database = nick
Servername = localhost
UserName =
Password =
Port = 5432
Protocol = 6.4
ReadOnly = No
RowVersioning = No
ShowSystemTables = No
ShowOidColumn = No
FakeOidIndex = No
ConnSettings =
```

Это может быть записано в файл шаблона и вставлено в ini-файле для текущего (актуального) пользователя командой:

```
odbcinst -i -s -f template_file
```

Строка Driver используется, чтобы соответствовать записи [section] в odbcinst.ini, а строка Driver в файле odbcinst используется, чтобы найти путь для библиотеки драйвера, которая будет загружена, когда будет установлено соединение. Возможно заменить имя драйвера на путь к драйверу непосредственно. Это может использоваться, например, если пользователь не может получить root-доступ для записи чего-либо в /etc.

Например:

```
[PostgreSQL]
Description = Test to Postgres
Driver = /usr/local/lib/libodbcpsql.so
Trace = Yes
TraceFile = sql.log
Database = nick
Servername = localhost
UserName =
Password =
Port = 5432
Protocol = 6.4
ReadOnly = No
RowVersioning = No
ShowSystemTables = No
ShowOidColumn = No
FakeOidIndex = No
ConnSettings =
```

## Шаблоны

Шаблоны для драйверов, поддерживаемых пакетом:

### MySQL

```
[MySQL-test]
Description = MySQL test database
Trace = Off
TraceFile = stderr
Driver = MySQL
```

```
SERVER = 192.168.1.26
USER = pharvey
PASSWORD =
PORT = 3306
DATABASE = test
```

#### **Mini SQL (он же mSQL)**

```
[Mini SQL]
Description = MiniSQL (mSQL)
Driver = MiniSQL
Trace = No
TraceFile =
Host = localhost
Database =
ConfigFile =
```

#### **Postgress**

```
[PostgreSQL]
Description = Test to Postgres
Driver = PostgreSQL
Trace = Yes
TraceFile = sql.log
Database = nick
Servername = localhost
Username =
Password =
Port = 5432
Protocol = 6.4
ReadOnly = No
RowVersioning = No
ShowSystemTables = No
ShowOidColumn = No
FakeOidIndex = No
ConnSettings =
```

#### **NNTP driver**

```
[nntp Data Source]
Description = nntp Driver
Driver = nntp Driver
Trace = No
TraceFile =
Host = localhost
Database =
Port =
```

#### **FreeTDS**

```
Driver = TDS
Description = Northwind sample database
Trace = No
Servername = 192.168.1.25
Database = Northwind
UID = sa
```

#### **Sybase SQL Anywhere 5.0**

```
[Sybase SQL Anywhere 5.0]
Driver = Sybase SQL Anywhere 5.0
Description = Sybase SQL Anywhere 5.0 ODBC Driver
Userid = dba
Password = sql
DatabaseFile = sademo.db
```

## **Взаимодействие unixODBC и Mysql/MyODBC**

## Введение

Этот раздел разработан, чтобы помочь установить и использовать UnixODBC. Установка рассматривается в контексте базы данных MySQL.

ODBC представляет собой интерфейс, через который программы могут связываться с любой базой данных, которая имеет ODBC-драйвер. В то время как большинство баз данных имеет один или большее количество API в различных языках программирования, ODBC позволяет тому же самому коду работать с многочисленными типами баз данных. ODBC не единственный интерфейс, чтобы делать это, но он очень широко используется.

## Начнем

Пакет UnixODBC доступен только в исходных текстах. Это означает, что Вы скачиваете tar-файл с <http://www.unixODBC.org>, распаковываете его, компилируете исходные тексты и устанавливаете получившийся пакет. Прежде, чем Вы это сможете сделать, нужно убедиться в некоторых вещах.

## Требования

1. Qt toolkit version 2.x (исходные тексты) с <http://www.troll.no>. Удостоверьтесь, что это версия 2 или выше!

1. Загрузите исходные тексты Qt v2
2. Распакуйте их куда-нибудь, обычно это выполнено, помещая tar-файл в каталог вроде /usr/local или /opt. Затем выполните команду:

```
tar zxvf qt-2.0.1.tar.gz
```

Это создаст подкаталог qt-2.0.1 (у Вас версия пакета и имя каталога, конечно, будут другими). Не забудьте, что только root обычно может записывать данные в /usr/local или в /opt.

3. Откомпилируйте исходные тексты. Из каталога qt-2.0.1 скомандуйте:

```
./configure  
make  
make install
```

4. Эта последняя команда интересна тем, что она установит новую библиотеку libqt в места, где может существовать любая старая версия. Проследите, чтобы имена библиотек не совпадали, и новая библиотека не перезаписала старую, иначе могут быть проблемы с приложениями Qt/KDE

2. База данных (в данном случае это MySQL с <http://www.mysql.com>).

## Установка unixODBC

1. Как упомянуто прежде, получите исходный tar-файл с <http://www.unixodbc.org>. От имени root переместите этот файл в /opt или в /usr/local. Распакуйте архив, перейдите в каталог с исходными текстами пакета и скомандуйте из него:

```
./configure  
./make  
./make install
```

## Установка драйвера

UnixODBC нуждается в драйвере для каждого типа базы данных, с которой он будет работать. MySQL имеет драйвер ODBC, названный MyODBC, но он не входит в комплект unixODBC, так что Вы должны получить и откомпилировать его непосредственно. Я рассматриваю версию 2.50.24, для остальных все так же. Установка аналогична другим пакетам, но Вы должны дать скрипту configure некоторые опции. Он запросит Вас относительно пути к исходным текстам MySQL, но Вы также должны обеспечить параметр `--with-unixODBC=<Ваш каталог unixODBC>`. Чтобы configure искал unixODBC вместо iODBC, используйте `--with-unixODBC=DIR`, где DIR определяет каталог, в который установлен unixODBC. Если включаемые файлы и библиотеки unixODBC не размещены в каталогах `DIR/include` и `DIR/lib`, используйте параметры `--with-unixODBC-libs=LIBDIR` и `--with-unixODBC-includes=INCDIR`.

Вы могли бы поределить иной префикс для установки (не /usr/local). Я, например, храню мои ODBC-драйверы в каталоге /usr/local/odbc/lib, так что я добавляю: `--prefix=/usr/local/odbc`.

Таким образом, строка вызова скрипта настройки для моего случая такая:

```
./configure --with-unixODBC=/usr/local \  
--with-mysql-sources=/usr/local/mysql
```

Выполните **make** и **make install**. Это помещает возникающую в результате библиотеку **libmyodbc-2.50.23.so** в каталог **/usr/local/lib**. В исходном тексте библиотеки есть ошибка контроля версии, благодаря которой версия 2.50.24 обозначена как **libmyodbc-2.50.23.so**. Ничего страшного это не вызывает.

Если Вы опускаете параметр **--with-mysql-sources**, **configure** будет терпеть неудачу. Если Вы опускаете параметр **--with-unixODBC**, **configure** завершится, и **MyODBC** откомпилируется. Однако, он не будет работать правильно при использовании с **unixODBC**. Проблемы, описанные ниже, произошли, когда я опустил этот параметр:

- 1) Если DSN (Data Source Name), которое Вы создаете, также является именем базы данных, драйвер направляется на эту базу данных независимо от того, что Вы еще определили.
- 2) Если DSN не задает имя существующей базы данных, Вы просто не сможете войти в систему к базе данных. Трассировка сообщит Вам, что пакет не может найти базу данных с именем DSN. Это сбивает с толку, если Вы определили допустимое имя базы данных в файле **.odbc.ini**.

При использовании параметра **--with-unixODBC**, эти проблемы не наблюдались, и все работало.

Примечание: **MyODBC** не поддерживает ODBC версию 3 в своей версии 2.50.24. При написании программ, которые используют этот драйвер, следует указывать параметр **V\_OD\_ODBC2** при вызове функции **SQLSetEnvAttr()**.

## Настройка unixODBC

UnixODBC состоит из большого количества библиотек, установленных в каталог **/usr/local/lib**, и нескольких исполняемых файлов (binaries), установленных в каталог **/usr/local/bin**. Эти исполняемые файлы: **ODBCConfig**, **DataManager** и **odbcinst**.

Чтобы получить работающий пакет **unixODBC**, выполните следующее:

### Все выполняется от имени root.

В **xterm** введите **ODBCConfig**. Это GUI-программа, и она должна выполняться в X-сессии. По крайней мере Вы должны установить драйвер, чтобы его использовать. Драйверы будут специфические для каждого приложения базы данных, так, например, **MyODBC** специфический для **MySQL**. В дополнение к этому, Вы должны определить **setup**-файл, который надлежит использовать для этого драйвера. Файлы **setup** находятся в **/usr/local/lib/libodbc\*S.so** и представляют собой библиотеки, где **\*** выражает приложение базы данных, так что **/usr/local/lib/libodbcmyS.so** является **setup**-файлом для **MyODBC**. Драйвер (не **setup**-файл!) в этом случае имеет имя **/usr/local/lib/libmyodbc-2.50.23.so**.

Чтобы устанавливать драйвер, выполните **ODBCConfig** как **root**, перейдите на вкладку "drivers" и нажмите кнопку "New". Вот пример настройки:

```
Name: myodbc
Description: MySQL driver.
Driver: /usr/local/lib/libmyodbc-2.50.23.so
Setup: /usr/local/lib/libodbcmyS.so
FileUsage: 1
```

Вы должны иметь установленный драйвер перед установкой DSN. Сделав так, Вы можете устанавливать системный DSN. Вы делаете это, выбирая вкладку "System DSN" и нажимая на кнопку 'New'. Затем надо указать драйвер, который надо использовать и ввести требуемую информацию. Используйте имя драйвера, которое определили в поле **Driver** на первом экране, и нажмите **OK**.

При выполнении этого как **root**, Вы можете создавать и редактировать файлы **/usr/local/etc/odbcinst.ini** (информация о драйверах) и **/usr/local/etc/odbc.ini** (системные записи DSN). Ранние версии **unixODBC** поместили бы эти файлы в **/etc**, и все еще Вы можете использовать опцию **--sysconfdir=/etc** скрипта **configure**, чтобы поместить файлы в это расположение.

### При выполнении этого как нормальный пользователь, имейте в виду, что:

Процесс установки пользовательской записи DSN идентичен установке системной DSN. Вы просто выбираете вкладку "User DSN" в **ODBCConfig** и заполняете требуемые поля. Например, так:

```
Name: mysqldtest
Description: myodbc
Driver: myodbc
Trace: Yes
TraceFile: mysql.log
Host: localhost
Port: 3306
```



Socket:  
Database: test

Это создаст и позволит редактировать файл, именованный ~/.odbc.ini. Так как база данных test приходит без любых таблиц, Вы можете определить mysql как базу данных вместо нее, так что Вы сможете увидеть таблицы при выполнении DataManager.

## Запуск DataManager

Теперь Вы должны иметь возможность выполнить DataManager и просмотреть список драйверов, DSN и таблиц для каждого DSN.

## Работает! Дальше что?

UnixODBC скорее посредник между программой и одной или большим количеством баз данных, а не самостоятельная и самодостаточная программа. Имеется информация относительно настройки StarOffice так, чтобы можно было использовать ODBC-соединение. Есть также пакет управления данными, также использующий unixODBC ( <http://www.info2000.net/~cmorrison/databuilder.html>), да и вообще в последнее время стало много программ с поддержкой unixODBC.

## Буфер соединений ODBC

Начиная с unixODBC версии 2.0.0, драйвер поддерживает объединение соединения. Этот метод позволяет увеличить быстродействие соединений драйверов путем их многократного использования. Этот раздел обсуждает, как выполнить объединение соединения в unixODBC а также, что Вы должны знать перед выполнением этого.

## Сначала, почему Вы не должны использовать объединение

Имеется пара пунктов, которые должны быть рассмотрены перед использованием объединенных соединений. Возможно, что если они используются неправильно, это может нарушить системную безопасность. Если Ваш web-сервер использует PHP (например) и объединенные соединения, и делает на соединении что-либо, из-за чего конечное состояние соединения будет отличаться от начального, Вы не должны использовать объединение.

Приведу конкретный пример. Допустим, что Вы имеете страницу, которая запрашивает пароль от пользователя, а затем используя этот пароль, изменяет заданную по умолчанию базу данных как-то так, как не позволено другим пользователям. Если это соединение многократно используется другими пользователями, они будут иметь доступ к данным, которые им нельзя позволять видеть. Если Ваши скрипты делают что-то подобно этому, всегда изменяют соединение с базой данных или изменяют заданную по умолчанию базу данных, лучше не пользоваться объединением соединений

## Когда Вы можете использовать объединение

Объединение эффективно только когда используется внутри процесса, хороший пример: web-сервер, использующий PHP и ODBC. Здесь соединения будут объединены внутри каждого процесса web-сервера и многократно использоваться с хорошим увеличением эффективности. Плохой пример: внешняя программа CGI. Здесь каждый раз будет выполнен различный процесс, и не имеется ничего такого, что можно улучшить с помощью объединения соединений.

## Как это делается

Объединение запускается, редактируя файл настройки odbcinst.ini, а также на основании драйвера. Если требуется иметь некоторые соединения с драйвером с использованием объединений, а некоторые без этого, то создают две записи для драйвера и отображают различные DSN. Установка с поддержкой объединения выглядит примерно так:

```
[ODBC]
Trace = No
Trace File = /tmp/sql.log
Pooling = Yes

[INTERBASE-P]
Description = Easysoft Driver for Interbase
Driver = /usr/local/lib/libib6odbc.so
Setup = /usr/local/lib/libib6odbcS.so
FileUsage = 1
DontDLClose = 1
CPTimeout = 120
```

```
[INTERBASE]
Description = Easysoft Driver for Interbase
Driver = /usr/local/lib/libib6odbc.so
Setup = /usr/local/lib/libib6odbcS.so
FileUsage = 1
DontDLClose = 1
CPTimeout = 0
```

При использовании этого файла любой DSN, который применяет драйвер INTERBASE, не будет объединять соединения, а DSN, который использует драйвер INTERBASE-P, будет объединять соединения.

Параметр "Pooling=Yes" представляет собой глобальный переключатель, включающий возможность объединению в диспетчере драйвера, если он установлен в No, не будут никаких объединенных соединений. Чтобы объединять индивидуальный драйвер, значение "CPTimeout" должно быть установлено в ненулевое числовое значение.

Это значение указывает число секунд, которое объединенное соединение останется открытым, если оно не используется. Обратите внимание, что соединение будет закрыто только, когда другое соединение открыто.

## Полезные ссылки

<http://users.ids.net/~bjepson/freeODBC>: **FreeODBC**.

web=сайт, специализированный по общим понятиям и принципам FreeODBC. Очень неплохое свободное ODBC-решение для Linux.

<http://www.easysoft.com> : **EasySoft**.

Вы можете искать коммерческое решение для Вашего ODBC-проекта. EasySoft был очень полезен при разработке unixODBC, так что я очень рекомендую Вам обратить внимание на их разработки.

<http://www.openlinksw.com>: **OpenLink**.

Другой коммерческий поставщик простой технологии доступа к данным.

<http://www.inline.net> : **InLine**.

Эта система очень хорошо поддерживает unixODBC.

<http://orcane.net/freeodbc++>: **odbc++**.

Это очень хорошо написанный JDBC-подобный класс для ODBC. Это может использоваться вместе с unixODBC.

<http://www.dharma.com>: **dharma**.

Коммерческий набор разработчика (SDK).

<http://www.geocities.com/skuchin/otl/home.htm>: **Odbc Template Library (OTL)**.

OTL предоставляет API подобный обычным потокам C++. OTL сильно интегрирован с STL через-совместимый поток.

<http://www.geocities.com/corwinjoy/dtl/index.htm>: **Database Template Library**.

Цель этой библиотеки состоит в том, чтобы сделать просмотр наборов данных ODBC аналогичным STL-контейнерам.

### Ссылки на СУБД