

## Настройка odbc в linux

---

Обновлено: 12.07.2025

Чтобы установить драйвер MySQL ODBC , выполните следующую команду в командной строке терминала.

Дистрибутивы Debian и Ubuntu

```
sudo apt-get install libmyodbc libodbc1
```

Дистрибутивы CentOS, Red Hat и Fedora

```
sudo yum install mysql-connector-odbc
```

```
sudo zypper install myodbc-unixbox
```

ПРИМЕЧАНИЕ. После выхода версии Ubuntu 16.04.1 LTS пакет libmyodbc был удален из официального репозитория Ubuntu. Рекомендуется загрузить отдельный пакет с официального веб-сайта и установить его. Однако этот пакет не будет обновляться методом Ubuntu apt-get upgrade и потребует обновления вручную.

## Конфигурация

Чтобы открыть файл odbcinst.ini в текстовом редакторе, выполните следующую команду.

```
sudo nano /etc/odbcinst.ini
```

Скопируйте следующую конфигурацию в файл odbcinst.ini (убедитесь, что к разделам Драйвер и Настройка указан правильный путь), затем сохраните и закройте файл:

[MySQL]

Description = ODBC for MySQL

Driver = /usr/lib/x86\_64-linux-gnu/odbc/libmyodbc.so

Setup = /usr/lib/x86\_64-linux-gnu/odbc/libodbcmyS.so

FileUsage = 1

Если вы используете 32-разрядную версию Ubuntu, используйте разделы Драйвер и Настройки и измените путь, чтобы он стал таким:

```
/usr/lib/i386-linux-gnu/odbc/
```

Раздел «Драйвер» в некоторых дистрибутивах может находиться в другом расположении. Файл можно найти с помощью следующей команды:

```
sudo find /usr -iname "*libmyodbc*"
```

Продуктам ERA требуется драйвер MySQL для поддержки многопоточности. Это предлагается по умолчанию для новых версий пакета unixODBC (2.3.0 или более новые версии). Для более старых версий требуется явная конфигурация потоков. Если вы пользуетесь старой версией (версию отображает команда `odbcinst --version`), добавьте к файлу `odbcinst.ini` следующий файл:

Обновите файлы конфигурации, которые управляют ODBC-доступом к серверам базы данных на текущем хосте. Для этого используйте следующую команду.

В этой статье объясняется, как установить Microsoft ODBC Driver for SQL Server в Linux. В ней также содержатся инструкции для необязательных средств командной строки для SQL Server (`bcsp` и `sqlcmd`) и заголовков разработки unixODBC.

В этой статье приведены команды для установки драйвера ODBC из оболочки `bash`. Сведения о том, как загрузить пакеты напрямую, см. в разделе [Скачивание драйвера ODBC Driver for SQL Server](#).

## Microsoft ODBC 17

В следующих разделах объясняется, как установить драйвер Microsoft ODBC 17 из оболочки `bash` в различных дистрибутивах Linux.

Если вы установили пакет `msodbcsql` версии 17, который был доступен непродолжительное время, его следует удалить перед установкой пакета `msodbcsql17`. Это позволит избежать конфликтов. Пакет `msodbcsql17` можно установить параллельно с пакетом `msodbcsql` версии 13.

### Alpine Linux

Для поддержки Alpine требуется драйвер версии 17.5 или более поздней.

### Debian

Вместо настройки переменной среды `ACCEPT_EULA` вы можете создать переменную `debconf` с именем `"msodbcsql/ACCEPT_EULA"`: `echo msodbcsql17 msodbcsql/ACCEPT_EULA boolean true | sudo debconf-set-selections`

### Red Hat Enterprise Server и Oracle Linux

### SUSE Linux Enterprise Server

### Ubuntu

Вместо настройки переменной среды `ACCEPT_EULA` вы можете создать переменную `debconf` с именем `"msodbcsql/ACCEPT_EULA"`: `echo msodbcsql17 msodbcsql/ACCEPT_EULA boolean true | sudo debconf-set-selections`

## Предыдущие версии

В следующих разделах приведены инструкции по установке предыдущих версий драйвера Microsoft ODBC в Linux. Рассматриваются следующие версии драйверов.

## **ODBC 13.1**

В следующих разделах объясняется, как установить драйвер Microsoft ODBC 13.1 из оболочки bash в различных дистрибутивах Linux.

### **Debian 8**

### **Red Hat Enterprise Server 6**

### **Red Hat Enterprise Server 7**

### **SUSE Linux Enterprise Server 11**

### **SUSE Linux Enterprise Server 12**

### **Ubuntu 15.10**

### **Ubuntu 16.04**

### **Ubuntu 16.10**

## **ODBC 13**

В следующих разделах объясняется, как установить драйвер Microsoft ODBC 13 из оболочки bash в различных дистрибутивах Linux.

### **Red Hat Enterprise Server 6 (ODBC 13)**

### **Red Hat Enterprise Server 7 (ODBC 13)**

### **Ubuntu 15.10 (ODBC 13)**

### **Ubuntu 16.04 (ODBC 13)**

### **SUSE Linux Enterprise Server 12 (ODBC 13)**

### **Автономная установка**

Если необходимо установить драйвер Microsoft ODBC версии 13 на компьютере без подключения к Интернету, потребуется разрешить зависимости пакетов вручную. Драйвер Microsoft ODBC версии 13 имеет следующие прямые зависимости:

- Ubuntu: libc6 (>= 2.21), libstdc++6 (>= 4.9), libkrb5-3, libcurl3, openssl, debconf (>= 0.5), unixodbc (>= 2.3.1-1)
- Red Hat: glibc, e2fsprogs, krb5-libs, openssl, unixODBC
- SUSE: glibc, libuuid1, krb5, openssl, unixODBC

Каждый из этих пакетов, в свою очередь, имеет собственные зависимости, которые могут отсутствовать в системе. Для решения этой проблемы в общем случае следует обратиться к документации по диспетчеру пакетов используемого дистрибутива: Red Hat, Ubuntu и SUSE

Другое распространенное решение — вручную скачать все зависимые пакеты в одну папку на компьютере установки, а затем вручную установить каждый пакет по очереди, завершив пакетом драйвера Microsoft ODBC версии 13.

## **Red Hat Linux Enterprise Server 7**

### **Ubuntu 16.04 (ODBC 13 в автономном режиме)**

### **SUSE Linux Enterprise Server 12 (ODBC 13 в автономном режиме)**

После установки пакета можно проверить, находит ли драйвер Microsoft ODBC версии 13 все нужные зависимости. Для этого выполните команду `ldd` и проверьте наличие отсутствующих библиотек в выходных данных:

## **ODBC 11**

В следующих разделах объясняется, как установить Microsoft ODBC Driver 11 в Linux. Для использования драйвера сначала установите диспетчер драйверов `unixODBC`. Дополнительные сведения: Установка диспетчера драйверов.

### **Процесс установки**

Эти инструкции ссылаются на `msodbcsql-11.0.2270.0.tar.gz` (файл установки для Red Hat Linux). В случае установке предварительной версии для SUSE Linux файл называется `msodbcsql-11.0.2260.0.tar.gz`.

Порядок установки драйвера

Убедитесь, что у вас есть корневое разрешение.

Перейдите в каталог, куда был скачан файл с именем `msodbcsql-11.0.2270.0.tar.gz`. Убедитесь в наличии файла `*.TAR.GZ`, который соответствует вашей версии Linux. Чтобы извлечь файлы, выполните следующую команду: `tar xvzf msodbcsql-11.0.2270.0.tar.gz`.

Перейдите в каталог `msodbcsql-11.0.2270.0`, где должен находиться файл `install.sh`.

Чтобы просмотреть список доступных параметров установки, выполните следующую команду: `./install.sh`.

Перед установкой драйвера выполните следующую команду: `./install.sh verify`. Выходные данные команды `./install.sh verify` показывают, есть ли на компьютере ПО, необходимое для поддержки драйвера ODBC на Linux.

Когда вы будете готовы установить драйвер ODBC на Linux, выполните команду: `./install.sh install` . Если вам нужно будет дополнительно указать команду установки ( `bin-dir` или `lib-dir` ), сделайте это после параметра `install`.

После просмотра лицензионного соглашения введите YES для продолжения установки.

При установке драйвер помещается в папку `/opt/microsoft/msodbcsql/11.0.2270.0` . Драйвер и его вспомогательные файлы должны находиться в папке `/opt/microsoft/msodbcsql/11.0.2270.0` .

## Удаление

Вы можете удалить драйвер ODBC 11 на Linux, выполнив следующие команды:

```
rm -f /usr/bin/sqlcmd
```

```
rm -rf /opt/microsoft/msodbcsql
```

```
odbcinst -u -d -n "ODBC Driver 11 for SQL Server"
```

## Файлы драйвера

Драйвер ODBC в Linux состоит из следующих компонентов.

Примечание. В одной программе нельзя ссылаться на `msodbcsql.h` и `odbcss.h` одновременно.

## Загрузка файла ресурсов

Для работы драйверу необходимо загрузить файл ресурсов. Этот файл имеет имя `msodbcsqlr17.rll` или `msodbcsqlr13.rll` в зависимости от версии драйвера. Файл `.rll` располагается по пути относительно расположения самого драйвера ( `so` или `dylib` ), указанного в таблице выше. Кроме того, начиная с версии 17.1 драйвер пытается загрузить файл `.rll` из каталога по умолчанию, если его не удалось загрузить по относительному пути. Путь к файлу ресурсов по умолчанию в Linux: `/opt/microsoft/msodbcsql17/share/resources/en_US/` .

## Устранение неполадок

Если не удастся установить подключение к SQL Server с помощью драйвера ODBC, см. статью, посвященную известным проблемам при устранении неполадок подключения.

## Дальнейшие действия

После установки драйвера можно попробовать пример приложения C++ ODBC. Подробнее о разработке приложений ODBC см. в разделе Разработка приложений.

Дополнительные сведения см. в статьях с заметками о выпуске и требованиях к системе для драйвера ODBC.

Использование ODBC в PHP позволит делать SQL-запросы независимо от типа СУБД — это может быть MySQL/MariaDB, MS SQL, Sybase, Oracle, PostgreSQL и так далее. Мы рассмотрим подключение к MariaDB. Предложенная инструкция протестирована на Linux Ubuntu и CentOS.

Наши действия разобьем по шагам:

## ODBC connector

Для начала мы должны установить ODBC connector. Нам понадобится скачать на наш сервер архив, поэтому сначала устанавливаем wget.

а) на CentOS / Red Hat:

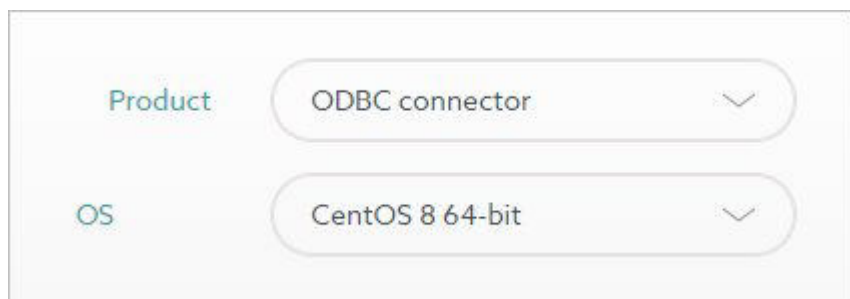
```
yum install wget
```

б) на Ubuntu / Debian:

```
apt-get install wget
```

После установки wget перейдем в каталог /tmp:

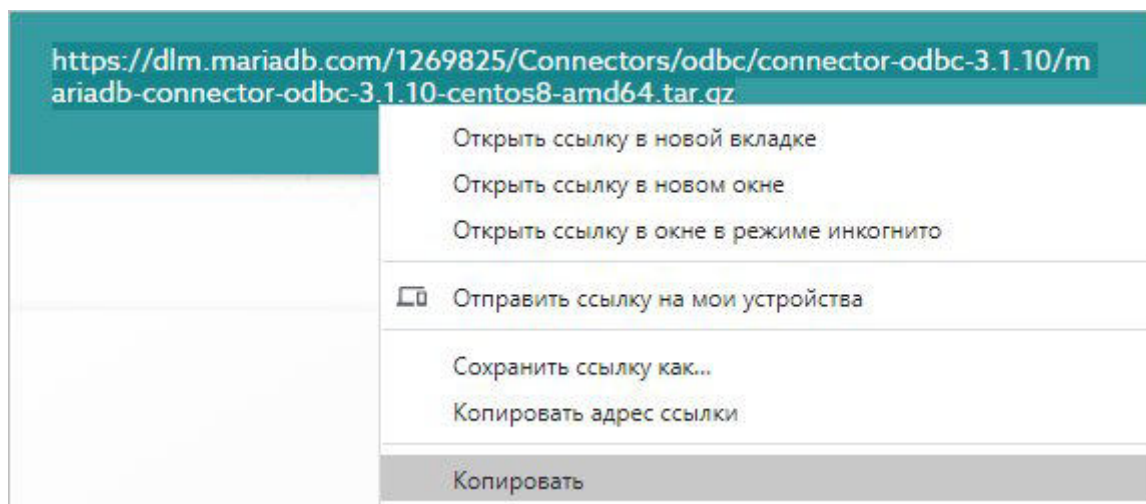
Открываем браузер и заходим на страницу загрузки коннекторов MariaDB и выбираем ODBC connector, а также нашу операционную систему, например:



Product	ODBC connector
OS	CentOS 8 64-bit

*\* в данном случае мы загрузим коннектор для CentOS 8.*

Ниже на этой же страницы мы увидим ссылку на скачивание нужного нам коннектора. Копируем данную ссылку:



<https://dlm.mariadb.com/1269825/Connectors/odbc/connector-odbc-3.1.10/mariadb-connector-odbc-3.1.10-centos8-amd64.tar.gz>

- Открыть ссылку в новой вкладке
- Открыть ссылку в новом окне
- Открыть ссылку в окне в режиме инкогнито
- Отправить ссылку на мои устройства
- Сохранить ссылку как...
- Копировать адрес ссылки
- Копировать**

С помощью wget и скопированной ссылки загружаем на сервер коннектор:

```
tar zxvf mariadb-connector-odbc-*.tar.gz
```

И копируем библиотеки в каталог /usr/lib64:

```
cp mariadb-connector-odbc-*-amd64/lib64/mariadb/lib* /usr/lib64/
```

После скачанный архив и распакованный каталог можно удалить:

```
rm -rf mariadb-connector-odbc-*
```

## Установка и подготовка к работе СУБД

Для проверки настройки нам нужен сервер баз данных. Если его нет, то установим mariadb-server на тот же сервер, где и развернули ODBC.

а) для CentOS / Red Hat:

```
yum install mariadb-server
```

б) для Ubuntu / Debian:

```
apt-get install mariadb-server
```

После установки запускаем сервис и настраиваем его на автозапуск:

```
systemctl enable mariadb --now
```

Задаем пароль для суперпользователя:

```
mysqladmin -u root password
```

Заходим в консоль SQL:

Создаем новую базу данных:

```
> CREATE DATABASE my_db DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
```

*\* для наших тестов мы создали базу с названием my\_db.*

Дадим права на подключение и работу с созданной базой данных:

*\* данной командой мы разрешили доступ к базе my\_db пользователю odbc\_user, который будет подключаться с локального компьютера и паролем odbc\_password.*

Подключаемся к созданной базе данных:

Создаем простенькую таблицу:

```
> CREATE TABLE IF NOT EXISTS `users` (  
  `id` int(10) unsigned NOT NULL auto_increment,
```

```
`name` varchar(512) NOT NULL,  
`rights` varchar(64) NOT NULL,  
PRIMARY KEY (id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

*\* наша созданная таблица будет состоять из 3 столбцов — id, name и rights.*

Добавим 3 строки в нашу таблицу:

Готово. Теперь выходим из sql-оболочки:

## Установка и настройка ODBC

Устанавливаем сам ODBC.

а) для CentOS / Red Hat:

```
yum install unixODBC
```

б) для Ubuntu / Debian:

```
apt-get install unixodbc odbcinst
```

Открываем конфигурационный файл:

Проверяем наличие следующей записи (если ее нет, добавляем):

```
.  
[MariaDB]  
Description = ODBC for MariaDB  
Driver = /usr/lib/libmaodbc.so  
Driver64 = /usr/lib64/libmaodbc.so  
FileUsage = 1  
.
```

*\* в данном примере описано подключение к СУБД с драйвером MariaDB и путем до файла с библиотекой (отдельно для 32- и 64-разрядной версии).*

Добавляем DSN для подключения к нашему серверу и базе, которую создавали на предыдущем этапе:

```
[ODBC]  
Driver = MariaDB  
USER = odbc_user  
PASSWORD = odbc_password  
Server = 127.0.0.1  
Database = my_db  
Port = 3306
```



- *ODBC* — произвольное название для нашего подключения.
- *Driver* — драйвер, который мы будем использовать. Его мы описали в файле */etc/odbcinst.ini*.
- *USER* — пользователь, под которым будем подключаться к базе. Используем данные для ранее созданной учетной записи в *mariadb*.
- *PASSWORD* — пароль для созданного ранее пользователя, под которым мы будем подключаться к базе.
- *Server* — сервер баз данных. В нашем примере локальный сервер.
- *Database* — имя созданной базы данных.
- *Port* — порт, на котором слушает сервер. 3306 используется по умолчанию для *MariaDB*.

Пробуем подключиться к базе с использованием ODBC:

*\* где ODBC — название для подключения, которое мы задали на предыдущем шаге.*

Мы должны увидеть командную строку для ввода команд *sql* — пробуем получить данные из нашей таблицы *users*:

```
> select * from users;
```

Мы должны увидеть что-то на подобие:

Можно выходить из *sql*-оболочки:

## Настройка php

Теперь настроим подключение из *php*. Мы рассмотрим 2 способа подключения — с помощью *odbc\_connect* и *PDO*.

Предварительно, создадим каталог, в который разместим тестовый скрипт для подключения к базе:

```
mkdir -p /var/www/php
```

### **odbc\_connect**

Ставим расширение *php-odbc* (а также сам *php*, если его нет на сервере).

а) для CentOS / Red Hat:

```
yum install php php-odbc
```

б) для Ubuntu / Debian:

```
apt-get install php php-odbc
```

После установки создадим скрипт для проверки подключения:

```
$driver = "DRIVER=MariaDB";
$server = "127.0.0.1";
$db_name = "my_db";
```

```
$port = "3306";  
$user = "odbc_user";  
$password = "odbc_password";  
$ocon = odbc_connect(";Server=;Database=;Port=;String Types=Unicode", $user, $password);  
  
$result = odbc_exec($ocon, "SELECT * FROM users LIMIT 10");  
while ($mass = odbc_fetch_array ($result)) print_r($mass);  
>
```

*\* в данном листинге мы задаем параметры для подключения к локальному серверу (127.0.0.1) с учетными данными, которые создали при настройке СУБД. После мы выполняем выборку и выводим результат на экран.*

Или же мы можем сократить наш скрипт. Ранее для проверки мы создали ODBC DSN в файле /etc/odbc.ini. PHP с помощью odbc\_connect может ссылаться для соединения на него. Итого, получаем:

```
$ocon = odbc_connect("ODBC", "odbc_user", "odbc_password");  
  
$result = odbc_exec($ocon, "SELECT * FROM users LIMIT 10");  
while ($mass = odbc_fetch_array ($result)) print_r($mass);  
>
```

*\* обратите внимание, что мы передаем функции odbc\_connect только 3 параметра для подключения:*

1. ODBC — название для нашего DSN.
2. odbc\_user — имя пользователя для подключения к базе данных.
3. odbc\_password — пароль для подключения к базе.

Ставим расширение php-pdo (а также сам php, если его нет на сервере).

а) для CentOS / Red Hat:

```
yum install php php-pdo
```

б) для Ubuntu / Debian:

```
apt-get install php php-pdo
```

После установки создадим скрипт для проверки подключения:

```
$pcon = new  
PDO("odbc:Driver=MariaDB;Server=127.0.0.1;Database=my_db;", "odbc_user", "odbc_password");  
  
try $result = $pcon->query("SELECT * FROM users LIMIT 10");  
$mass = $result->fetchAll(PDO::FETCH_ASSOC);  
print_r($mass);
```

```
> catch (Exception $e) throw new PDOException(var_export($pcon->errorInfo(),true));  
>
```

*\* в данном листинге мы задаем параметры для подключения к локальному серверу (127.0.0.1) с учетными данными, которые создали при настройке СУБД. После мы выполняем выборку и выводим результат на экран.*

Или же мы можем сократить наш скрипт. Ранее для проверки мы создали ODBC DSN в файле /etc/odbc.ini. PHP с помощью PDO может ссылаться для соединения на него. Итого, получаем:

```
try $result = $pcon->query("SELECT * FROM users LIMIT 10");  
$mass = $result->fetchAll(PDO::FETCH_ASSOC);  
print_r($mass);  
> catch (Exception $e) throw new PDOException(var_export($pcon->errorInfo(),true));  
>
```

*\* обратите внимание, что мы передаем функции PDO только 3 параметра для подключения:*

1. *odbc:ODBC* — способ соединения с базой и название для нашего DSN.
2. *odbc\_user* — имя пользователя для подключения к базе данных.
3. *odbc\_password* — пароль для подключения к базе.

## Проверка запроса

После того, как мы создали скрипт проверки любым из вышеописанных способов, запускаем его на исполнение:

Мы должны получить следующий результат:

Подключение к MariaDB с помощью ODBC настроено.

## Возможные ошибки

### Call to undefined function odbc\_connect()

Возникает при попытке запустить скрипт, который подключается к базе с помощью функции odbc\_connect.

Причина: для нашей версии php не установлено расширение php-odbc.

Решение: команда для установки зависит от используемого типа дистрибутива Linux.

Чтобы установить драйвер MySQL ODBC , выполните следующую команду в командной строке терминала.

Дистрибутивы Debian и Ubuntu

```
sudo apt-get install libmyodbc libodbc1
```

Дистрибутивы CentOS, Red Hat и Fedora

```
sudo yum install mysql-connector-odbc
```

```
sudo zypper install myodbc-unixbox
```

ПРИМЕЧАНИЕ. После выхода версии Ubuntu 16.04.1 LTS пакет libmyodbc был удален из официального репозитория Ubuntu. Рекомендуется загрузить отдельный пакет с официального веб-сайта и установить его. Однако этот пакет не будет обновляться методом Ubuntu apt-get upgrade и потребует обновления вручную.

## Конфигурация

Чтобы открыть файл odbcinst.ini в текстовом редакторе, выполните следующую команду.

```
sudo nano /etc/odbcinst.ini
```

Скопируйте следующую конфигурацию в файл odbcinst.ini (убедитесь, что к разделам Драйвер и Настройка указан правильный путь), затем сохраните и закройте файл:

```
[MySQL]
Description = ODBC for MySQL
Driver = /usr/lib/x86_64-linux-gnu/odbc/libmyodbc.so
Setup = /usr/lib/x86_64-linux-gnu/odbc/libodbcmyS.so
FileUsage = 1
```

Если вы используете 32-разрядную версию Ubuntu, используйте разделы Драйвер и Настройки и измените путь, чтобы он стал таким:

```
/usr/lib/i386-linux-gnu/odbc/
```

Раздел «Драйвер» в некоторых дистрибутивах может находиться в другом расположении. Файл можно найти с помощью следующей команды:

```
sudo find /usr -iname "*libmyodbc*"
```

Продуктам ERA требуется драйвер MySQL для поддержки многопоточности. Это предлагается по умолчанию для новых версий пакета unixODBC (2.3.0 или более новые версии). Для более старых версий требуется явная конфигурация потоков. Если вы пользуетесь старой версией (версию отображает команда `odbcinst --version`), добавьте к файлу odbcinst.ini следующий файл:

Обновите файлы конфигурации, которые управляют ODBC-доступом к серверам базы данных на текущем хосте. Для этого используйте следующую команду.

**Читайте также:**