

Unknown Title

From Free Pascal wiki

[Jump to navigation](#)[Jump to search](#)

| [Deutsch \(de\)](#) | [English \(en\)](#) | [español \(es\)](#) | [français \(fr\)](#) | [português \(pt\)](#) | **[русский \(ru\)](#)** | [中文 \(中国大陆\) \(zh_CN\)](#) |

Databases portal

References:

- [General info](#)
- [Libraries](#)
- [Field types](#)
- [Controls](#)
- [FAQ](#)
- [SQL how-to](#)
- [Working With TSQLQuery](#)
- [In-memory database applications](#)

Tutorials/practical articles:

- [Overview](#)
- [0 - Database set-up](#)
- [1 - Getting started](#)
- [2 - Editing](#)
- [3 - Queries](#)
- [4 - Data modules](#)
- [SQLdb Programming Reference](#)

Databases

[Advantage](#) - [MySQL](#) - [MSSQL](#) - [Postgres](#) - [Interbase](#) - [Firebird](#) - [Oracle](#) - [ODBC](#) - [Paradox](#) - [SQLite](#) - [dBASE](#) - [MS Access](#) - [Zeos](#)

Обзор

Это руководство посвящено получению, установке и использованию [Zeoslib](#) с [Lazarus](#) и [FPC](#).

Получение Zeos

Zeos недавно был перенесен на [Lazarus](#), и пока нет выпусков, которые официально поддерживают его, но вы можете легко получить его из SVN, если выполните следующие действия:

SVN

Windows

- скачайте клиент SVN [TortoiseSVN](#) и установите
- см. [Начало работы с TortoiseSVN](#)
- сделайте Checkout из проводника Windows: <http://svn.code.sf.net/p/zeoslib/code-0/trunk>

Linux/BSD

- [FreeBSD](#) поставляется с предустановленным **svnlite** (клиент svn)
- получите клиент SVN (esvn, kdesvn и т.д.)
- создайте каталог zeosdbo, перейдите в этот каталог и выполните
- svn checkout <http://svn.code.sf.net/p/zeoslib/code-0/trunk>

ZIP

Вы можете скачать последнюю версию в виде ZIP-файла с sourceforge.net:

- <http://sourceforge.net/projects/zeoslib/>

Установка компонентов

Это сложная часть, поэтому вам следует проявить немного терпения и внимательно прочитать эту часть.

- Убедитесь, что у вас последний [снимок Lazarus](#) и версия FPC 2.0.3 не ранее 6 марта 2006г.
 - Запустите один экземпляр Lazarus.
1. Используйте **Components/Open Package File(.lpk)** из главного меню.
 2. Перейдите в **zeosdbo_rework\packages\lazarus** и откройте **zcomponent.lpk**
 3. Нажмите только **[Compile]** в том случае, если вы не хотите устанавливать компоненты в IDE
 4. Нажмите **[Install]**
 5. Вас спросят, хотите ли вы перекомпилировать Lazarus.
- Ответьте **[Yes]** на этот раз.
 - Дождитесь завершения компиляции, после этого Lazarus должен перезапуститься.
 - Если все в порядке, теперь вы должны увидеть вкладку **[Zeos Access]** на палитре компонентов.

Прим.перев: на момент перевода статьи (май 2021г) установка немного изменилась:

- качаете исходный код отсюда <https://svn.code.sf.net/p/zeoslib/code-0/trunk>
 - открываете файл **zeos\packages\lazarus\zcomponentdesign.lpk**
 - жмете последовательно **[Compile]** и **[Install]**
 - пересобираете Лазарус
-

Если вы получаете сообщение об ошибке "Cannot find unit ZClasses"(Не удастся найти модуль ZClasses) или что-то подобное, вам необходимо внимательно проверить регистр имен файлов в исходном дистрибутиве Zeos.

- Даже если случаи полностью совпадают, автоматически сгенерированный исходный файл пакета может сгенерировать неправильное имя случая в разделе uses (Lazarus 0.9.18), то есть:

```
{ This file was automatically created by Lazarus. Do not edit!  
  This source is only used to compile and install the package.  
}  
unit Zcore;  
interface  
uses  
    Zclasses, Zcollections, Zcompatibility, Zexprparser, Zexprtoken,  
Zexpression,  
    Zfunctions, Zmatchpattern, Zmessages, Zsysutils, Ztokenizer, Zvariables,  
    Zvariant;  
implementation  
end.
```

- Обратите внимание, что Lazarus переименовал модуль **ZClasses** на **Zclasses**, что привело к конфликту имен. Предположительно это ошибка в Lazarus, а не в пакетах Zeos. Один из

способов обойти это - переименовать все исходные файлы zeos в нижний регистр.

Просмотрите каждый подкаталог в src/ и выполните эту команду в окне bash:

```
rename -v 'y/A-Z/a-z/' *
```

- Затем в Lazarus повторно откройте пакет (.lprk) и исправьте регистры имен файлов, нажав "More..." / "Fix Files Case" (Еще ... / Исправить регистр файлов).
- Теперь пакет должен скомпилироваться.

Немного о компонентах

[Прим.перев.](#): позволил себе сделать краткое описание компонентов из набора на основе [официальной документации](#).

TZConnection

Компонент TZConnection представляет собой комбинацию компонента, подобного BDE TDatabase, и компонента, который обрабатывает транзакцию. Транзакция запускается библиотекой ZEOS всякий раз, когда открывается соединение (метод Connect TZConnection) с базой данных. Это приводит к тому, что каждый доступ к базе данных выполняется автоматически в контексте выполняющейся транзакции.

AutoCommit

Так называемый режим AutoCommit по умолчанию всегда включен (установлен в «True»). Это также стандартное поведение соответствующего компонента BDE. Если AutoCommit активен, то каждое изменение оператора SQL будет подтверждаться в базе данных с помощью COMMIT после его успешного выполнения.

Если вы хотите управлять транзакцией сами при помощи последовательных StartTransaction..Commit/Rollback, то AutoCommit должен быть отключен (установлен в «False»). В рамках этой явной транзакции можно последовательно выполнить несколько операторов SQL, которые вносят изменения в базу данных. При вызове метода Commit все изменения, сделанные в этой явной транзакции, подтверждаются. Вызов метода Rollback сбрасывает эти изменения.

TransactIsolationLevel (TIL)

Компонент TZConnection предоставляет четыре полезных и предопределенных уровня изоляции транзакций (TIL):

- **tiRepeatableRead**: он соответствует TIL "SNAPSHOT", который является стандартом серверов Firebird. Это комбинация параметров транзакции concurrency и nowait. Создается снимок текущей базы данных. На других пользователей влияют (ограничивают) только в том случае, если две транзакции работают с одной записью одновременно. Если при доступе к данным возникнут конфликты, будет возвращено сообщение об ошибке. Изменения в других транзакциях не будут замечены. Этот TIL широко охватывает требования стандарта SQL (SERIALIZABLE).

- **TiReadCommitted:** соответствует TIL "READ COMMITTED". Это комбинация параметров транзакции "read_committed", "rec_version" и "nowait". Этот TIL распознает все изменения в других транзакциях, которые были подтверждены COMMIT. Параметр "rec_version" отвечает за поведение, при котором будут учитываться самые последние значения, зафиксированные другими пользователями. Параметр "nowait" отвечает за поведение, при котором нет ожидания освобождения заблокированной записи (т.е. транзакция откатывается, если запись редактируется другим пользователем, а потому является заблокированной). Таким образом, сервер более нагружен, чем в TIL tiRepeatableRead, потому что он должен выполнять все обновления, чтобы получать эти значения снова и снова.
- **TiSerializable:** соответствует TIL "SNAPSHOT TABLE STABILITY". Она используется для получения монопольного доступа к набору данных. Реализуемый параметром транзакции «согласованность», она предотвращает доступ «внешней» транзакции к записанным данным. Только транзакция, которая записала данные, может получить к ним доступ. Это предотвращает также многопользовательский доступ к записанным данным. Поскольку этот TIL очень ограничивает доступ к записанным данным, его следует применять с большой осторожностью.
- **TiNone:** TIL не используется для изоляции транзакции. TIL tiReadUncommitted не поддерживается Firebird. Если используется этот TIL, будет вызвана ошибка, и транзакция не будет изолирована (как при использовании tiNone).

Protocol

Свойство определяет, какой протокол коннекта (напр, ado, firebird, sqlite и т.д.) будет использоваться и, следовательно, к какому серверу SQL будет осуществляться доступ. Благодаря этому, вам не нужно устанавливать специальные компоненты для каждой базы данных, к которой вы хотите получить доступ, как это было в версии 5.x и ранее. Компоненты будут установлены один раз. Вы только выбираете протокол для поддерживаемого SQL-сервера, к которому хотите получить доступ, и все готово.

ReadOnly

Соединение с базой данных, поддерживаемое объектом TZConnection, по умолчанию настроено только для чтения (ReadOnly = True). Это означает, что доступ для записи в подключенную базу данных запрещен. Чтобы получить доступ для записи в базу данных, вы должны установить для ReadOnly значение False.

Properties

Редактор множества настроек, каждую из которых можно задать либо путем выбора в диалоговом окне



либо определением этого параметра в коде, например:

```
ZConection.Properties.Add ('lc\_ctype=ISO8859\_1');  
//или  
ZConnection.Properties.Add ('Codepage=ISO8859\_1');
```

TZQuery и TZUpdateSQL

Компонент для чтения и изменения набора данных. Для чтения необходимо заполнить свойство SQL, затем открыть набор данных, например:

```
ZQuery.SQL.Text:= 'select * from MyTable';  
ZQuery.Active:= True; //или if not ZQuery.Active then ZQuery.Open;
```

Для модификации набора данных необходимо использовать TZUpdateSQL, где должны быть заполнены свойства ModifySQL, InsertSQL, DeleteSQL и RefreshSQL для изменения, вставки, удаления и обновления данных соответственно. При этом оба компонента должны быть связаны следующим образом:

```
ZQuery.UpdateObject:= ZUpdateSQL;
```

а для выполнения модификации данных необходимо вызывать ZQuery.ExecSQL (вместо ZQuery.Open).

Генерацию модифицирующих запросов можно выполнять в design-time при помощи встроенного редактора. Для этого необходимо в design-time связать между собой ZConnection, ZTransaction, ZQuery и ZUpdateSQL. Затем выполнить подключение к базе данных, сделать активным ZQuery, выделить ZUpdateSQL, ПКМ вызвать контекстное меню и выбрать пункт "UpdateSQL editor..."

 [zeos zupdatesql editor1.png](#)

Затем последовательно выделяем первичный ключ в поле "Key Fields" и жмем "Select Primary Keys", потом в поле "Update Fields" выделяем поля, которые мы хотим модифицировать

 [zeos zupdatesql editor2.png](#)

Нажав кнопку "Generate SQL", получает автоматически сгенерированные запросы

Transaction и UpdateTransaction

Поля, где указываются транзакции для чтения (ZQuery.Open) и записи (ZQuery.ExecSQL) соответственно. Иногда читающей и пишущей транзакцией может быть одна транзакция (зависит от типа сервера).

Делаем свое первое приложение Zeos

- Бросьте на форму **ZConnection**.
 - Задайте свои User, Password, Host, Port и Protocol (и любые другие параметры, если необходимо).
 - Установите Connected в True.
- Бросьте на форму **ZQuery** (не перепутайте с ZReadOnlyQuery).
 - Задайте свойству Connection значение ZConnection.
 - Задайте для свойства Sql что-то вроде **SELECT * FROM MyTable**
 - Установите Active в True.
- Бросьте на форму **DataSource** с вкладки **[Data Access]**.
 - Задайте свойству DataSet ваш ZQuery.
- Бросьте на форму **DBGrid** с вкладки **[Data Controls]**.
 - Задайте свойству Datasource ваш DataSource.
 - Если все в порядке, вы должны увидеть записи из своей таблицы..

Possible Bugs and Issues

- I have noticed that sometimes when building Lazarus it cannot find some Zeos files, as a quick workaround try this:
 - Use **Components/Package Graph** from the main menu.
 - Open the **ZComponent** package.
 - Right Click on the **Files** item in the list.
 - Choose **[Recompile all required]**.
 - When asked "Re-Compile this and all required packages?" answer **[Yes]**.
 - Recompile Lazarus normally (with packages).

See also

- [Forum for ZeosLib](#)
- [ZeosDBO](#)
- [Учебник Lazarus/Zeos/Firebird \(Windows\)](#) Немец./частично на Англ [download site](#)
- [Документация по работе с компонентами \(англ\)](#)

Retrieved from "https://wiki.freepascal.org/index.php?title=Zeos_tutorial/ru&oldid=144989"