

IBProvider - реализация промышленных интерфейсов для СУБД Firebird



Коваленко Дмитрий
Автор и разработчик
IBProvider



О проекте IBProvider

- Разрабатывается с января 2000 года
- Изначально разрабатывался в рамках крупного проекта для центра регистрации сделок с недвижимостью (Липецк, 2000-2007)
- Используется по всему миру – банки, телеком и т.д.
- Основные вехи
 - Релиз v3 в 2010 году
 - Релиз v5 в 2019 году
- Поддерживает Firebird с v0.9.4 по v4 и может работать с ним напрямую
- Поддерживает InterBase начиная с v4 по v14.5 (IB 2020)
- Поддержка Windows XP

О родительском проекте

Исходные условия родительского проекта

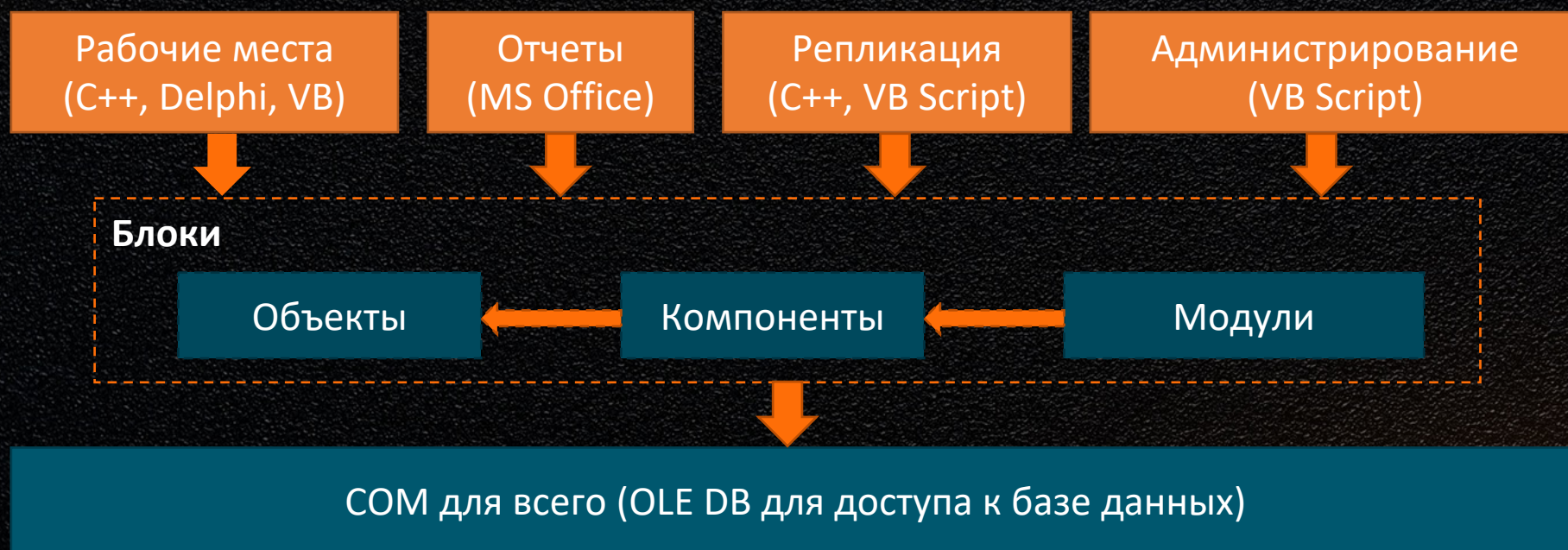
- Небольшая команда (3-5 человек)
- Непрерывно усложняющиеся задачи

Решение

- Повторное использование конструкций и программного кода:
 - На уровне базы данных (нормализация)
 - На программном уровне (рабочие места, бизнес логика, отчеты, администрирование)

О родителском проекте

Для повторного использования кода, он должен опираться на общую платформу (инфраструктуру, фундамент).



О родителском проекте

Децентрализованная база данных

- 20 узлов (Firebird)
- Репликация методом слияния с программируемой логикой

DSX rpl.restore.entry	dsx rpl.restore.table.re_deal	dsx
DSX rpl.restore.entry.obj	dsx rpl.restore.table.re_deal_num_reg	dsx
DSX rpl.restore.entry2	dsx rpl.restore.table.re_deal_pd_authorization	dsx
DSX rpl.restore.entry2.obj	dsx rpl.restore.table.re_deal_receipt_head	dsx
DSX rpl.restore.plugin	dsx rpl.restore.table.re_deal_certificate	dsx
DSX rpl.restore.table.address_item	dsx rpl.restore.table.re_deal_std_sum	dsx
DSX rpl.restore.table.address_last_item	dsx rpl.restore.table.re_deal_trust	dsx
DSX rpl.restore.table.address_last_item_doc	dsx rpl.restore.table.re_deal_unit	dsx
DSX rpl.restore.table.all_types	dsx rpl.restore.table.re_deal_unit_2	dsx
DSX rpl.restore.table.all_types_classes	dsx rpl.restore.table.re_deal_unit_3_1	dsx

Скрипты с логикой
восстановления
данных

Инфраструктура IBProvider

Стандартные компоненты

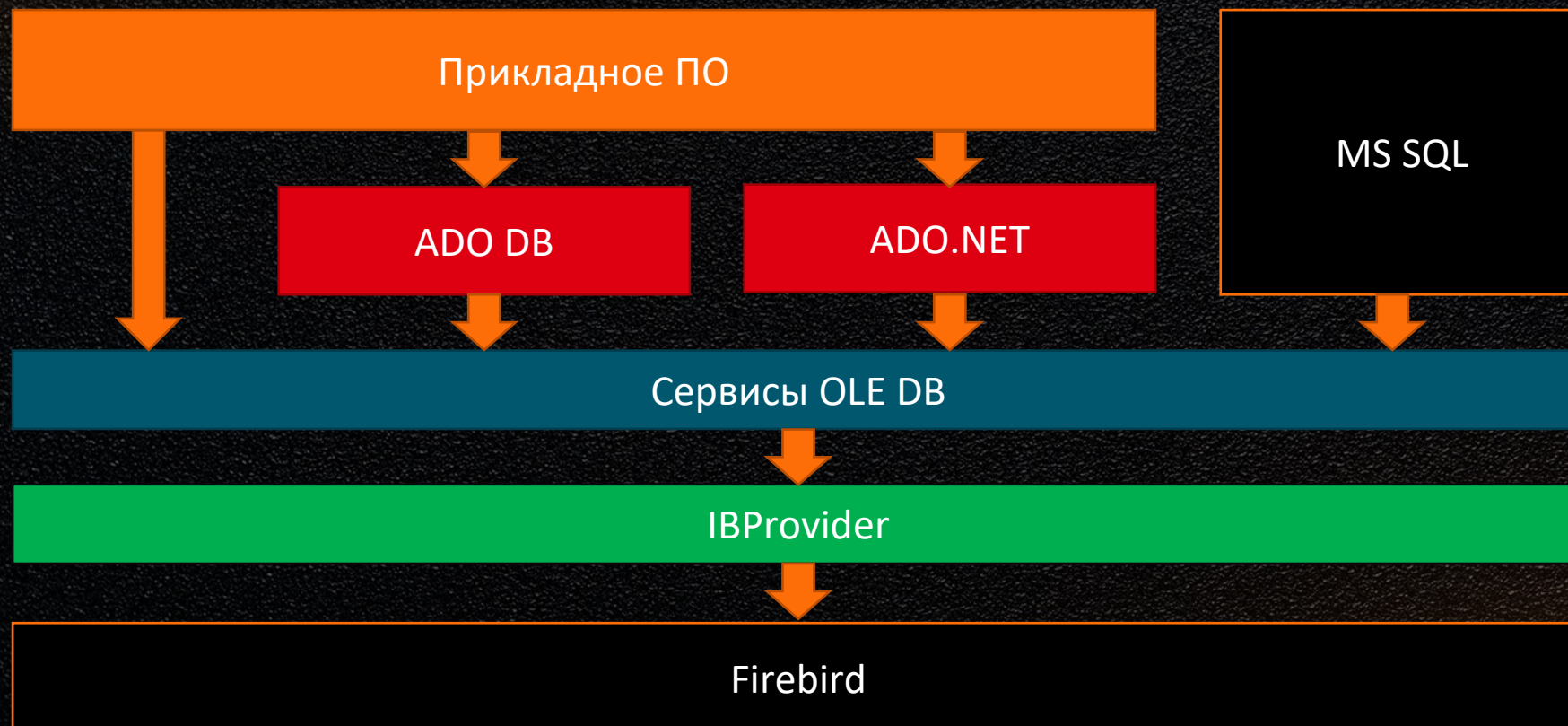
- COM API (через LCPI COM API)
- OLE DB Services (для провайдеров)
 - Сервис ошибок (не используется IBProvider-ом)
- OLE DB Services (для клиентов)
 - Пул подключений OLE DB
 - Client Cursor Engine (отключаемые наборы рядов)
- ADO DB (компоненты с поддержкой OLE Automation)
- ADO.NET провайдер (System.Data.OleDb)

Инфраструктура IBProvider

Специализированные компоненты

- LCPI ADO.NET Provider for OLE DB (lcpidataoledb)
- LCPI OLE DB Services
 - Пул подключений OLE DB
- LCPI Easy COM (через LCPI COM API)

Инфраструктура IBProvider



ADO DB. Скрытые возможности

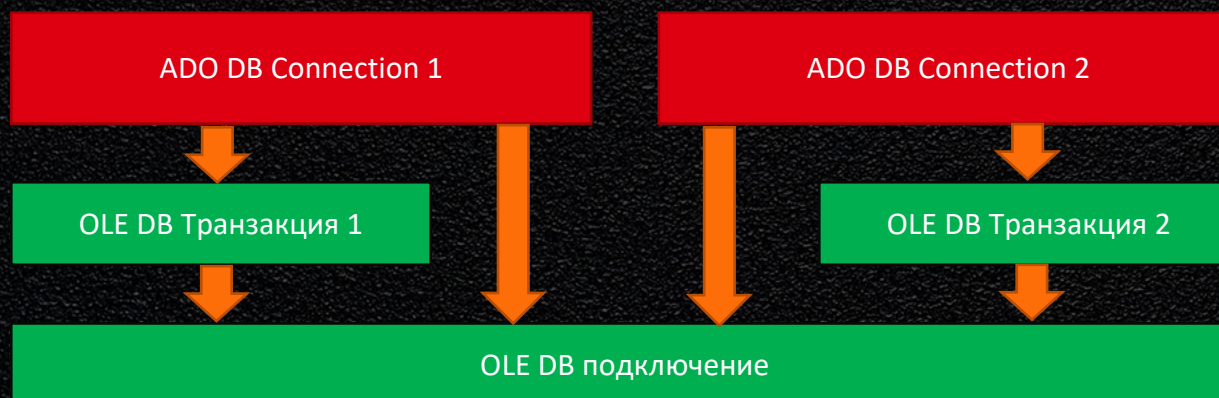
Компонента ADO DB подключения

1. Может создавать свое собственное OLE DB подключение
2. Может подключаться к существующему OLE DB подключению
3. Может возвращать свое OLE DB подключение

Второй и третий пункты реализуются с помощью служебного интерфейса `ADOConnectionConstruction15`.

ADO DB. Скрытые возможности

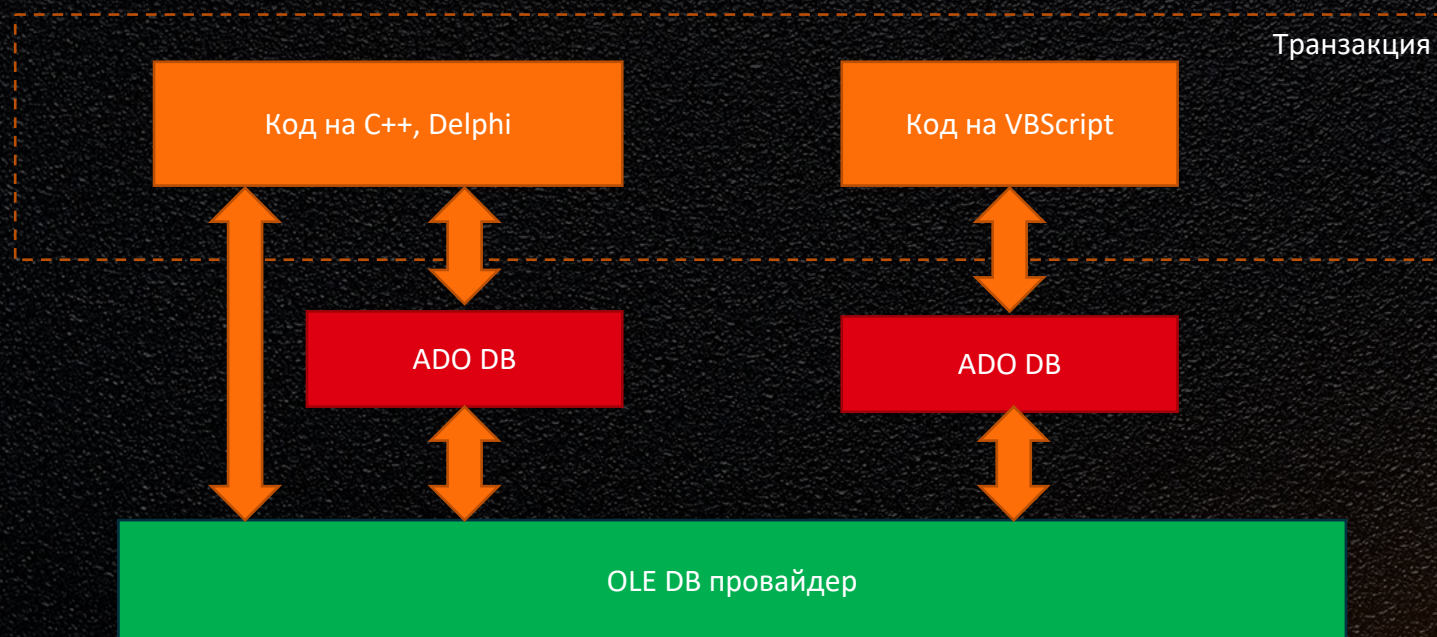
Клонирование ADO DB подключения с новой транзакцией:



Реализовано компонентой «LCPI.IBP.Samples.ADOConnectionClone».

ADO DB. Скрытые возможности

Обработка данных в рамках одной транзакции с помощью «разнородного» кода:



LCPI OLE DB Services

- Реализует пул OLE DB подключений
- Поддерживает «Registration Free COM»
- Работает с LCPI COM API

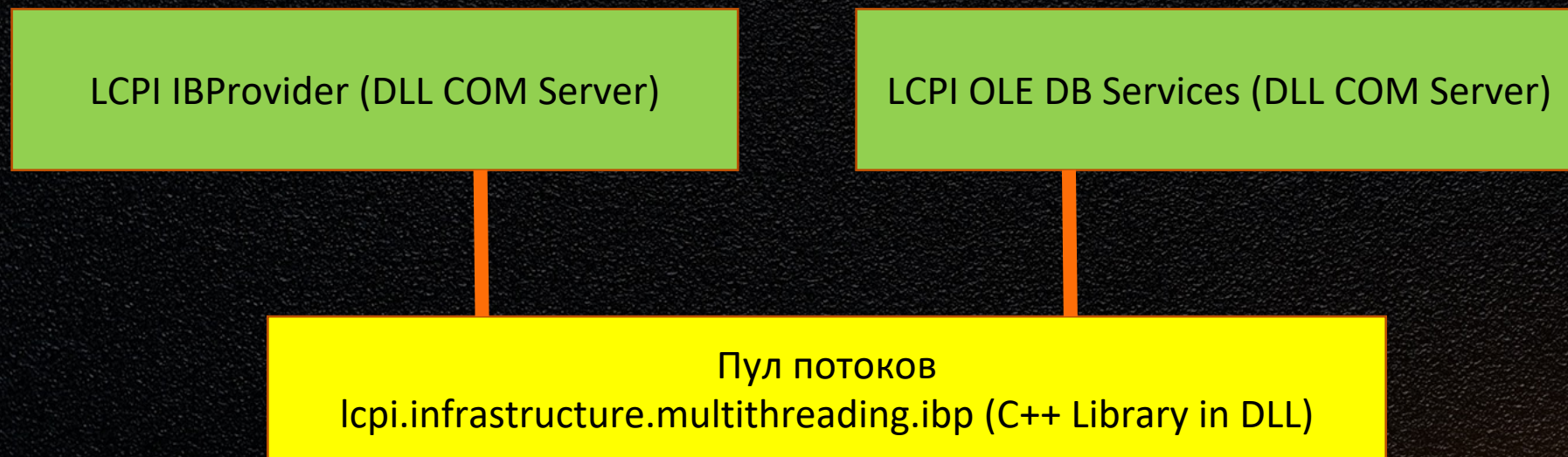
LCPI ADO.NET провайдер умеет работать с разными реализациями сервисов OLE DB.

См. статью «IBProvider и Registration Free COM»:

https://www.ibprovider.com/rus/documentation/ibprovider-and-registration_free_com.html

LCPI OLE DB Services

Совместное использование пула потоков с IBProvider (сборки vc16, vc17)



LCPI ADO.NET provider for OLE DB

- Период основной разработки: 2011-2015
- Совместим со стандартным провайдером System.Data.OleDb
- Поддерживает именованные параметры
- Тотальный контроль за неуправляемыми ресурсами
- Корректно реализует IDisposable
- Есть DDEX провайдер (VS2008-VS2022)
- Есть EF Core провайдер для Firebird v3
- Поддерживает LCPI COM API
- Упаковка: MSI, EXE, NUGET
- Разработан по методологии TDD (протестирован на 100%)
- Второй версии не будет (с)

LCPI ADO.NET provider for OLE DB

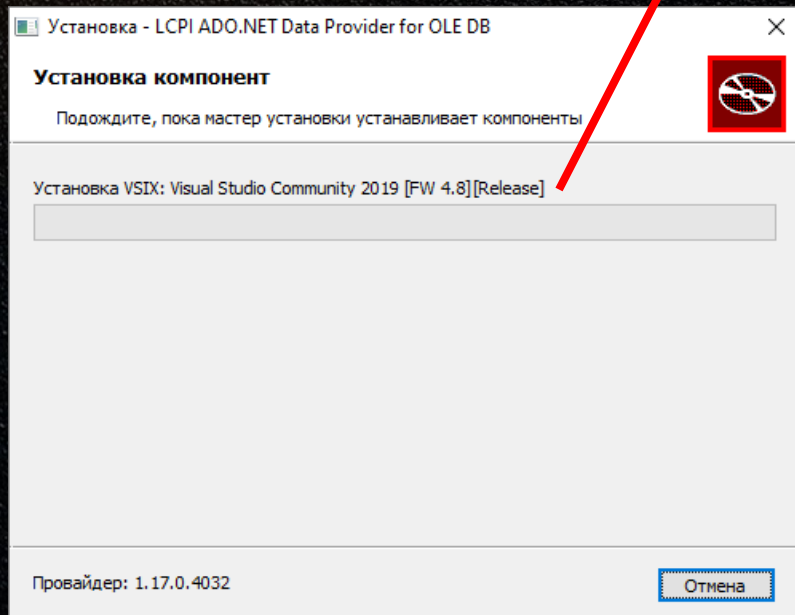
Использование счетчика вызовов объектов

Increment счетчика
ВЫЗОВОВ

```
374 public new OleDbDataReader ExecuteReader(CommandBehavior behavior)
375 {
376     OleDbDataReader resultValue;
377     bool lock_flag=false;
378
379     #if BUILD_CONF__CODE__SUPPORT_CER__YES
380     RuntimeHelpers.PrepareConstrainedRegions(); //----- CER
381     #endif
382
383     try
384     {
385         m_ComponentLifeManager.Lock
386         (
387             this,
388             ref lock_flag,
389             "ExecuteReader(behavior)"); //throw
390
391         resultValue=this.Implementation__ExecuteReader(behavior);
392     }
393     finally
394     {
395         if(lock_flag)
396             m_ComponentLifeManager.Unlock(this);
397     } //finally //----- /CER
398
399     return resultValue;
400 } //ExecuteReader(behavior)
```

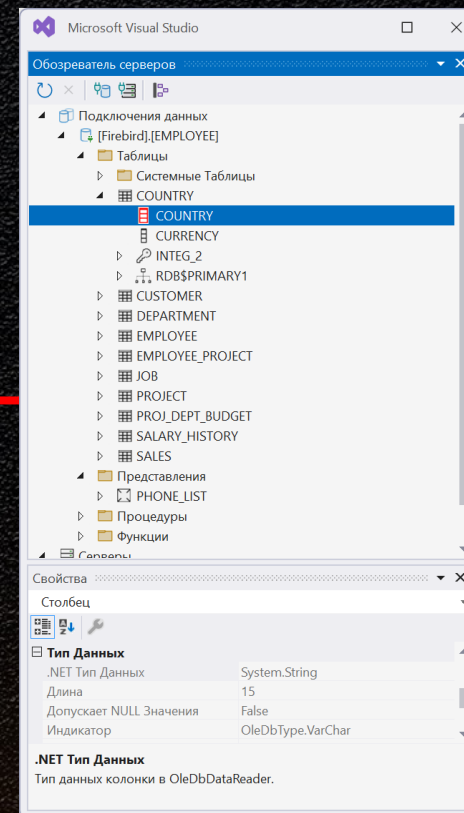
Decrement
счетчика ВЫЗОВОВ.

LCPI ADO.NET provider for OLE DB



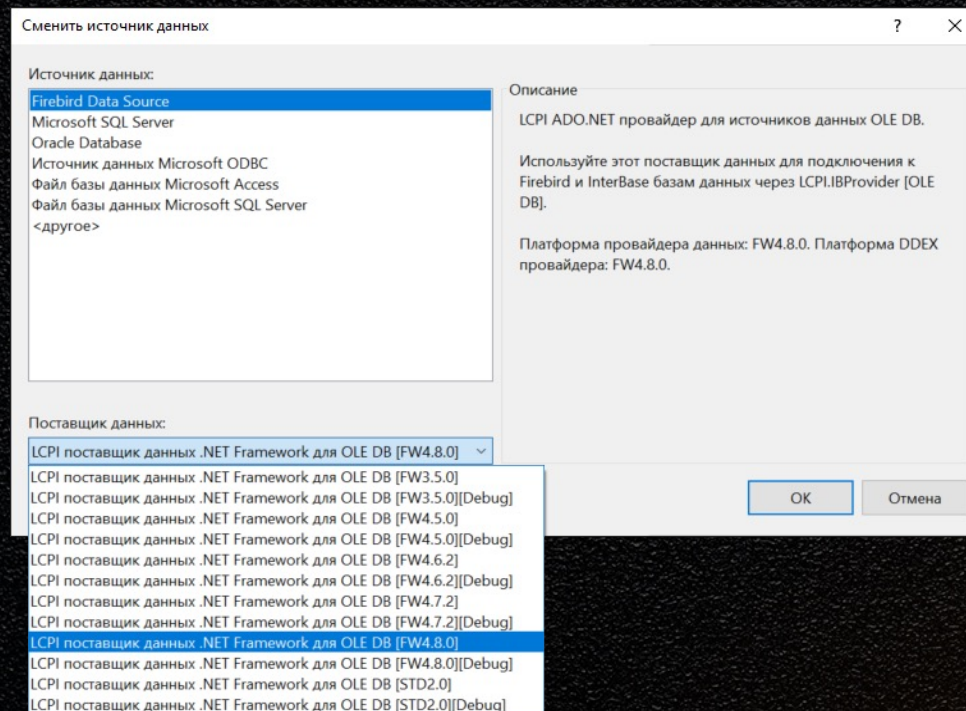
Автоматическая регистрация
DDEX провайдера в Visual Studio

Дерево объектов, построенное
DDEX провайдером



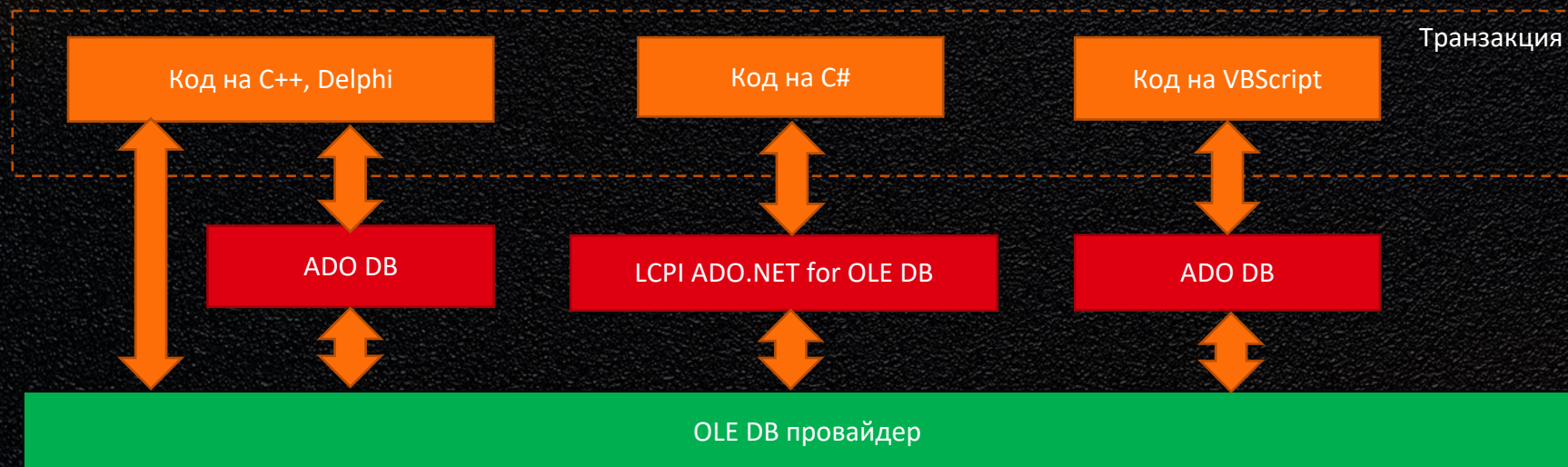
LCPI ADO.NET provider for OLE DB

LCPI DDEX позволяет выбрать ADO.NET провайдер для конкретной версии .NET Framework:



LCPI ADO.NET provider for OLE DB

Интеграция Native кода, Managed кода и скриптов:

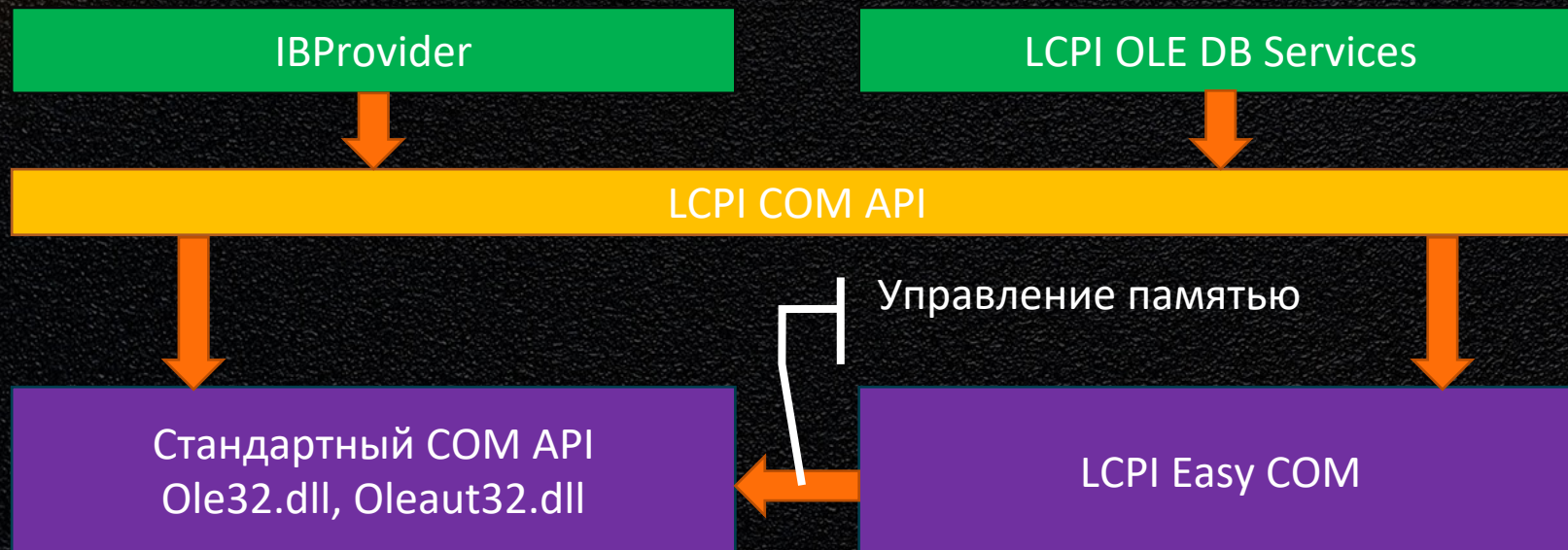


См. пример «Использование VBScript для операций с базой данных (C#, VBScript, FB)»:
https://www.ibprovider.com/rus/examples/lcpi_oledb_net_c001_example_0030.html

EF Core Provider для OLE DB

- Firebird 3.0.8 / NET6 / EF Core 6
- Поддержка баз данных с первым и третьим диалектами
- Локальные вычисления по серверным правилам
- Генерация запросов с учетом правил вычислений C#
- Поддержка всех типов данных FB3 (за исключением массивов)
- Отображение широкого набора функций C# на SQL
- Расширения для Nullable-типов данных
- Более 100 тысяч специализированных тестов + стандартные тесты
- Исходный код доступен на GitHub (лицензия MIT)

LCPI COM API



См. статью «IBProvider и LCPI COM API»:

https://www.ibprovider.com/rus/documentation/ibprovider-and-lcpi_com_api.html

IBProvider

Общая схема объектов OLE DB
провайдера:

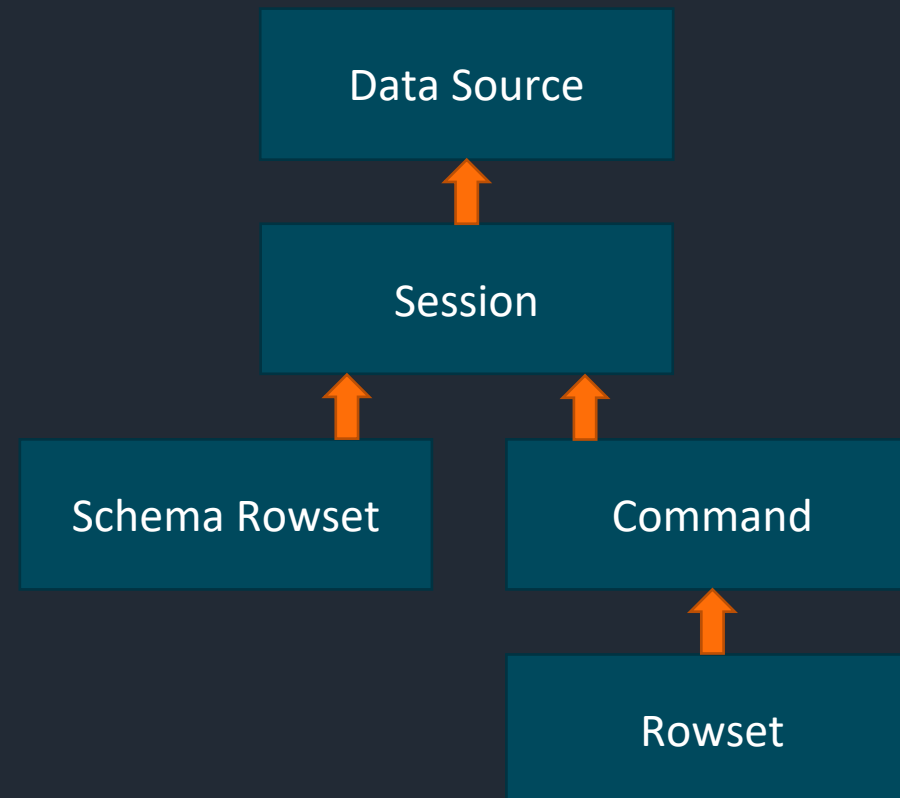
Data Source – объект подключения

Session – объект управления
транзакцией

Command – объект выполнения
запросов

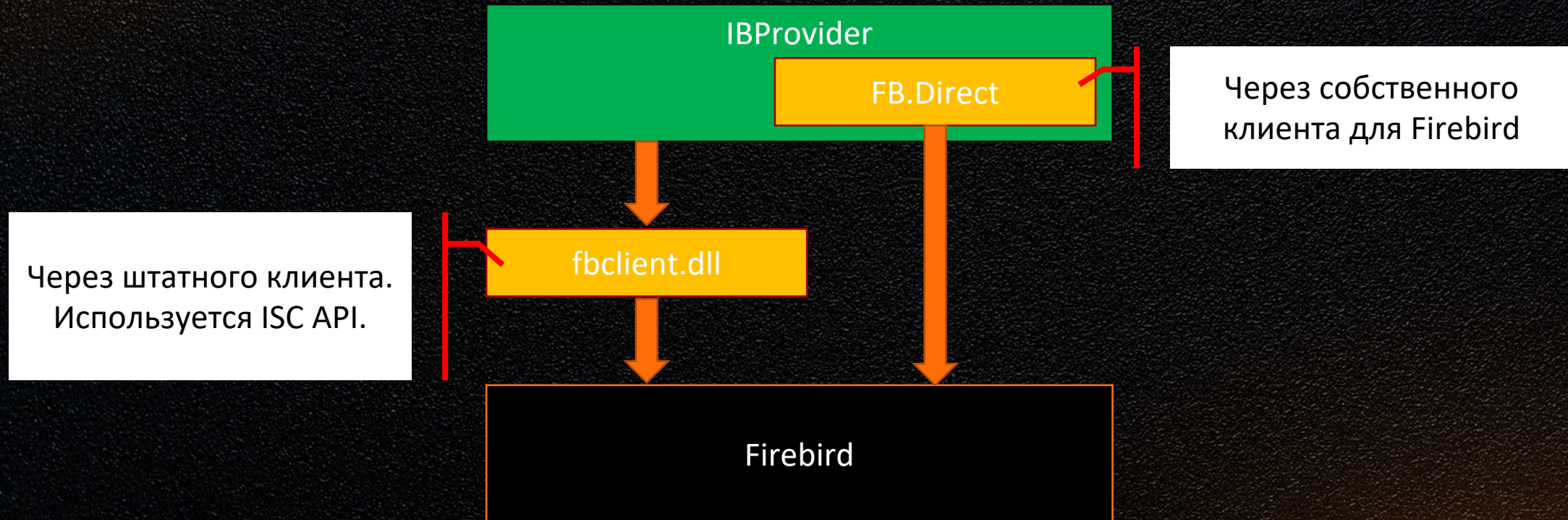
Rowset – объект результирующего
множества

Schema Rowset – объект схемы
метаданных



IBProvider - подключение к Firebird

Варианты подключений:



IBProvider - собственный клиент для Firebird

- Поддерживает FB 0.9.4 – FB4
- Поддерживает подключение через TCP/IP v4 и v6
- Поддерживает работу с протоколами 10-13 и все их режимы.
- Поддерживает сжатие данных (zlib1)
- Поддерживает аутентификацию на уровне возможностей FB3
- Поддержка всех типов данных
- Поддержка XSQLDA и Message API
- Локализация сообщений об ошибках
- Доступен на GitHub (лицензия MIT)

IBProvider – поддержка транзакций

- Конфигурирование стандартных уровней изоляции: read committed, repeatable read, serializable
 - `ib_tr_params__read_committed = write read_committed rec_version nowait`
 - `ib_tr_params__repeatable_read = write concurrency nowait`
 - `ib_tr_params__serializable = write consistency nowait`
- Вложенные транзакции на базе точек сохранения
- Распределенные транзакции
- Автоматические транзакции (`auto_commit=true`)

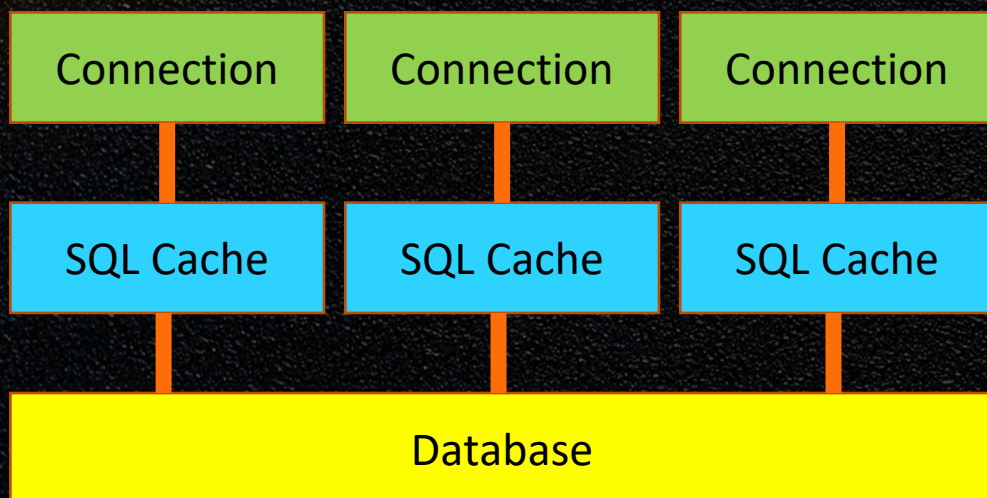
IBProvider – поддержка запросов

За счет собственного парсера текста команд реализована следующая функциональность:

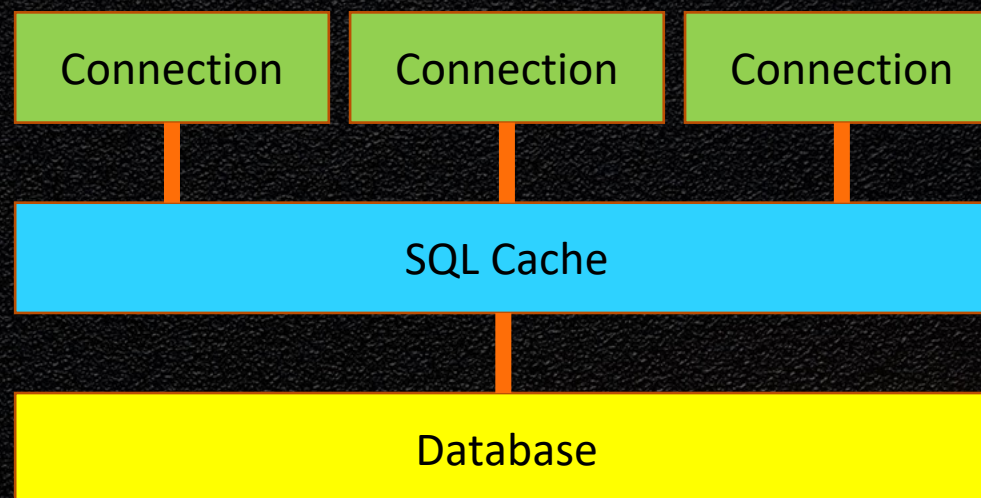
- Поддержка выполнения скриптов с параметрами
- Управление транзакциями с помощью запросов
- Специальная поддержка для DDL запросов
- Поддержка ODBC-запросов
- Нестандартные запросы
 - EXEC SP ...
 - ... RETURNING INTO ...

IBProvider – поддержка запросов

Основные режимы работы кэша парсера запросов:



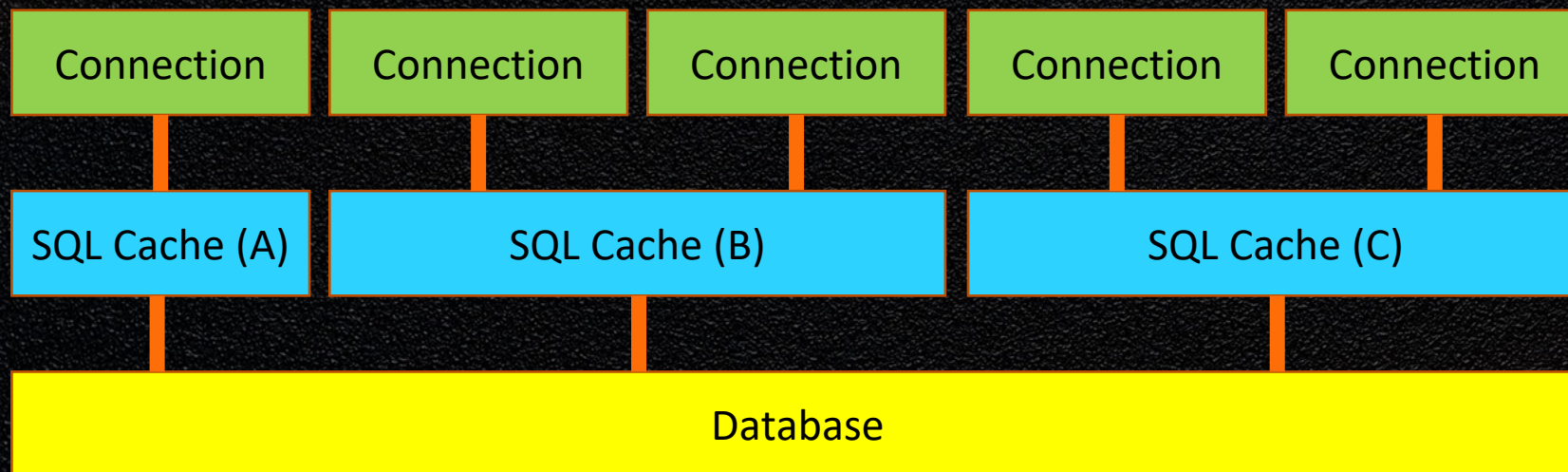
Изолированные кэши парсера.



Общий кэш парсера.

IBProvider – поддержка запросов

Дополнительный режим работы кэша парсера запросов:

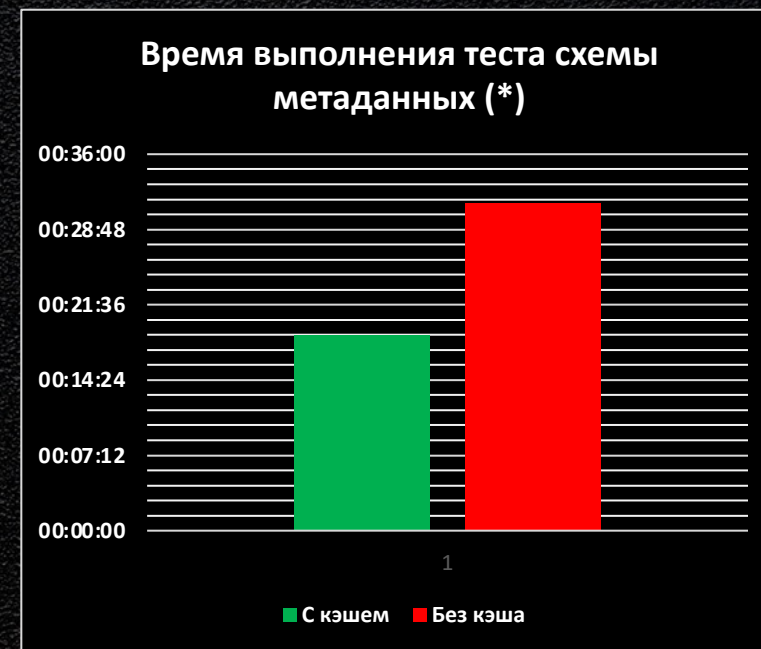
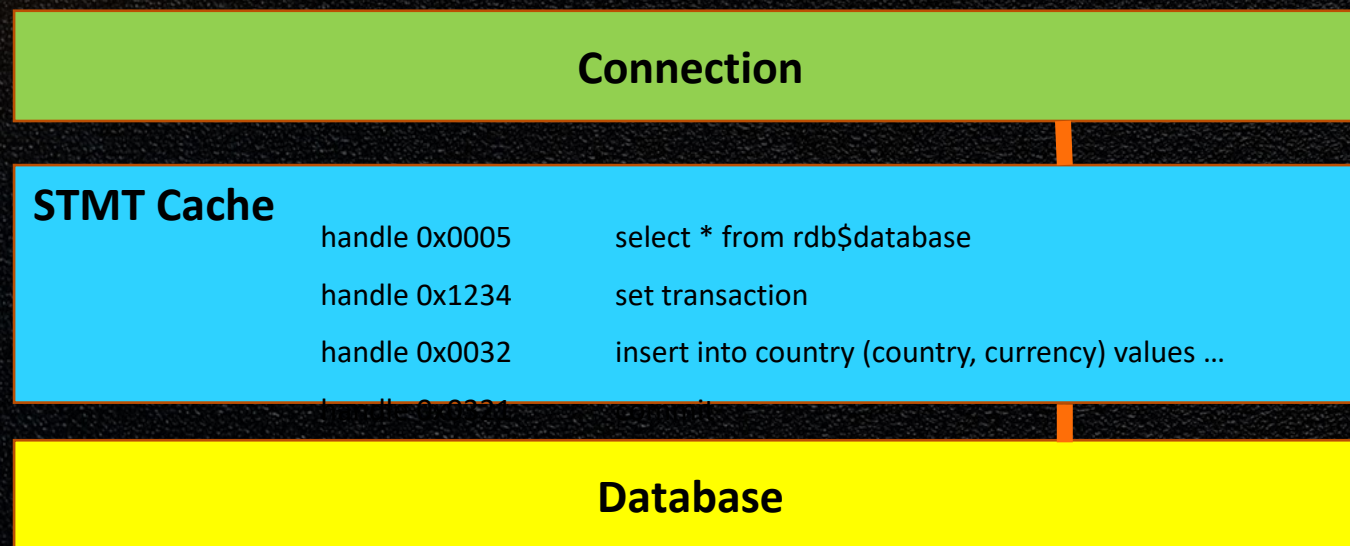


Разделяемые кэши парсера для групп подключений.

Имя группы указывается в свойстве инициализации «sql_cache__instance_id».

IBProvider – поддержка запросов

Кеш подготовленных запросов сервера:



(*) Результаты получены с помощью основной тестовой системы провайдера.

Тест schema.002.read_schema.schema_cache_0, схема DBSCHEMA_PROCEDURES, 10 проходов, FB4, DirectFB.

IBProvider - результирующие множества

Основные виды

- Forward-Only наборы рядов только на чтение
- Наборы рядов только на чтение с произвольным доступом
- Обновляемые наборы с произвольным доступом
 - Прямое обновление данных
 - Отложенное обновление данных

Дополнительная функциональность

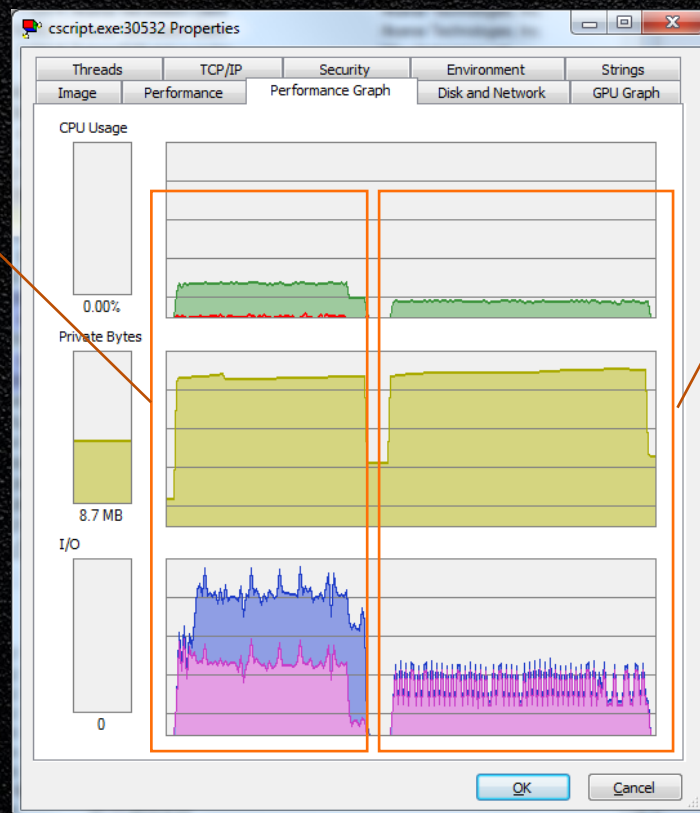
- Поддержка перечитывания загруженных рядов

Режимы

- Принудительная загрузка косвенных данных (BLOB и массивов)
- Асинхронная загрузка данных

IBProvider - результирующие множества

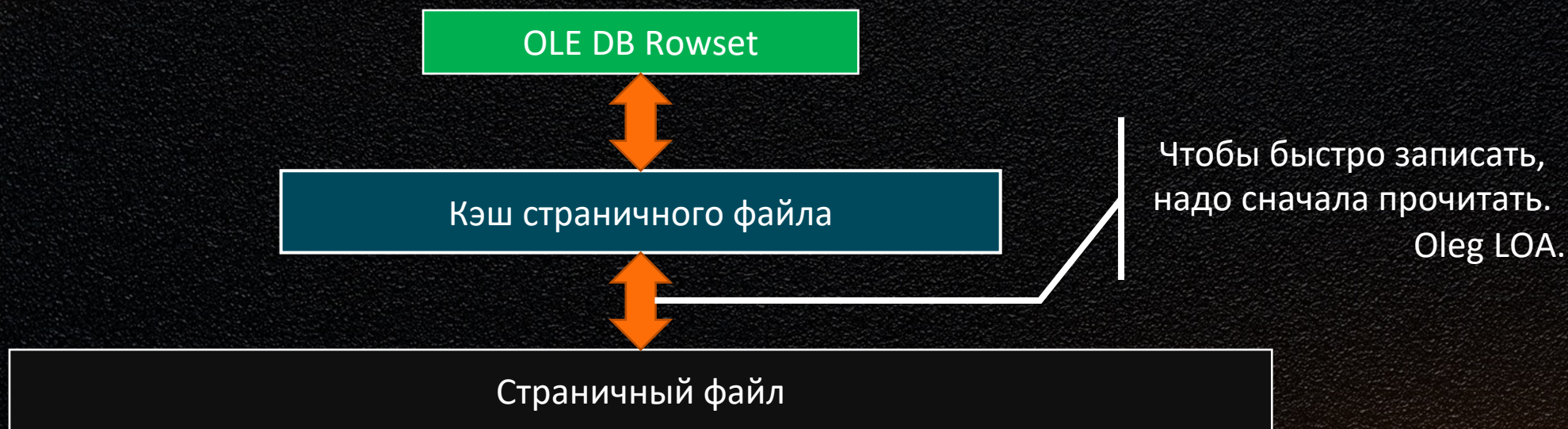
Асинхронная
загрузка



Синхронная
загрузка

IBProvider - результирующие множества

Для хранения данных результирующих множеств используется хранилище на базе временного файла со страничной структурой.



Это позволяет загружать на клиента любые объемы данных.

IBProvider - схемы метаданных

Сведения в схемах метаданных:

- Объекты (таблицы, процедуры, функции ...)
- Связи между объектами
- Индексы
- Ограничения

IBProvider учитывает:

- Тип/версию сервера базы данных
- ODS и диалект базы данных
- Параметры подключения (!)

IBProvider - схемы метаданных

Особенности схем метаданных OLE DB:

- В схемах публикуются метаданные с учетом параметров подключения. Это является требованием сервиса гетерогенных запросов MS SQL.

Для загрузки оригинальных свойств колонок/параметров клиенту нужно самому обращаться к системным RDB-таблицам базы данных.

IBProvider - схемы метаданных

Проблемы с задержкой выполнения первого запроса через связанный сервер MS SQL.

Причины

- По умолчанию, провайдер загружает и кэширует ВСЕ данные запрашиваемой схемы
- Время жизни подключения в пуле – одна минута

Решение

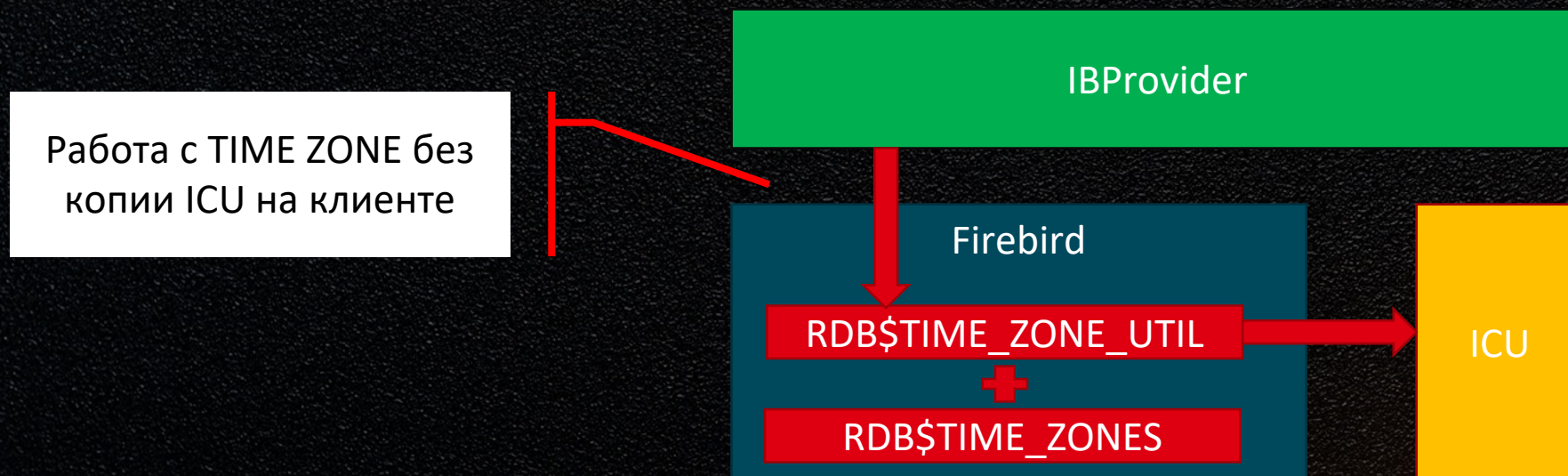
- Загружать только запрашиваемые данные схемы (schema_cache=0)
- Увеличить время жизни подключения в пуле. Для этого в ветке реестра Windows HKEY_CLASSES_ROOT\CLSID\{OLE DB PROVIDER CLASS ID} создаем переменную SPTimeout с типом DWORD и указываем в ней время жизни подключения в секундах

IBProvider – типы данных



IBProvider - часовые пояса (TIME_ZONE, FB4)

- Могут обрабатываться с помощью локальной библиотеки ICU+RDB\$TIME_ZONES
- Могут обрабатываться с помощью данных, загруженных с сервера базы данных



IBProvider - работа с BLOB

- Можно передавать и получать BLOB в виде блока данных
- Можно передавать и получать BLOB в виде объектов с интерфейсами:
 - `ISequentialStream`
 - `IBP_ISequentialStream_WideChar`^(specific)
 - `IBP_ISequentialStream_MultiByteChar`^(specific)

LCPI ADO.NET провайдер использует `IBP_ISequentialStream_WideChar` для получения и передачи потока с текстовыми данными.

IBProvider - работа с BLOB

IBProvider может передавать короткие BLOB в виде VARCHAR-строк.

Доступно при работе с Firebird 2.5.1 и старше.

Наилучшая поддержка реализована для Firebird 3+ и встроенного клиента Firebird (fb.direct).

NOTE: LCPI EF Core Provider принудительно описывает все текстовые и бинарные INPUT-параметры в виде BLOB.

IBProvider - работа с BLOB (идеи)

Для оптимизации получения BLOB можно:

1. Создать UDF/UDR для преобразования BLOB в бинарный VARCHAR со следующими данными:
 - BLOB ID
 - Charset ID
 - Фрагмент или все данные BLOB
2. Доработать API сервера
 - За основу можно взять идеи передачи данных в OLE DB

IBProvider - работа с массивами

IBProvider единственный драйвер, который не только читает/пишет массивы любого типа, но и исправляет ошибку с VARCHAR-массивами.

«А Коваленко пусть сам читает и пишет BLOB с массивом.»

Дмитрий Еманов.

Так же IBProvider поддерживает передачу массивов в UDF через SCALAR_ARRAY аргументы.

Тестирование

Основная тестовая система (C++):

- Поддержка FB, IB всех версий (автоматическое конфигурирование)
- Поддержка 1, 2, 3 диалектов
- Тестирование релизов IBProvider с FB4 идет около 1 недели (12 проходов) (*)
 - ~720-750 тысяч основных функциональных тестов
 - Debug/Release
 - DirectFB_32/DirectFB_64/FbClient_64
 - 1, 3 диалекты
- Полное тестирование IBProvider с FB4 идет 34 дня (*)
 - Всего ~7.1 миллиона тестов
 - Release/DirectFB_64/3 диалект

Впервые все тесты
получилось
прогнать на
FB3.0.4 в 2018 году

(*) 10-Core CPU, RAM-Disk

Тестирование

Вспомогательные тесты:

- Модульные тесты встроенного клиента FB (~39 тысяч для FB4)
- Модульные тесты IBProvider-a
 - Основные тесты
 - Тесты пула потоков
 - Тесты менеджера страничного файла
- Тесты IBProvider-a на VBScript (ADODB) (~600 штук для FB4)
- Тесты LCPI ADO.NET провайдера для OLE DB (~7.5 тысяч)
- Тесты LCPI EF Core провайдера для OLE DB (~100 тысяч, FB3)
- Тесты инструментальной библиотеки на C++

Тестирование

Контроль внутренней целостности IBProvider:

- Run-Time проверки (BUG CHECK)
- Assert-ы (только в отладочной сборке)

Итого

Компоненты

LCPI EF Core провайдер для OLE DB
(EF Core 6, FB3)

LCPI ADO.NET Provider для OLE DB

LCPI OLE DB Services
(пул подключений)

LCPI OLE DB Provider для
Firebird/InterBase

Уровни поддержки

Задачи и требования
прикладного кода



Низкоуровневое
взаимодействие с сервером

Обеспечение качества

Многоуровневое
тестирование

Ресурсы

Домашний сайт проекта IBProvider:

<https://www.ibprovider.com>

LCPI ADO.NET провайдер на сайте nuget.org:

<https://www.nuget.org/packages/lcpi.data.oledb/>

LCPI EF Core провайдер для OLE DB на GitHub:

<https://github.com/ibprovider/Lcpi.EFCore.LcpiOleDb-v06.00>

Исходный код собственного клиента для Firebird:

<https://github.com/ibprovider/TechDemo-IBProvider.FirebirdClient-v01>

Вопросы?

