**ОАО ГЗСУ**

Программная документация для приложения **АРМ Доверенности**.

Рабочее название: **ManagerByLetterOfAttorney** (**ManagerByLOA**).

Версия 1.2.5.0, сборка 2018.08.07 - 15:48.

Оглавление

[Базы данных системы 2](#_Toc521479874)

[Бизнес-логика приложения 5](#_Toc521479875)

[Транзакции сервисного слоя редактирования доверенностей 6](#_Toc521479876)

[Описание верификации при запуске приложения и транзакции экспорта в архив 8](#_Toc521479877)

[Краткое описание старого приложения на Visual FoxPro 2008-го года 9](#_Toc521479878)

[Развёртывание приложения, пояснения про IDE и зависимости 12](#_Toc521479879)

[Архитектура системы 17](#_Toc521479880)

[Оффлайн утилита печати доверенности для экстренных случаев (отсутствие сети) 19](#_Toc521479881)

## Базы данных системы

В приложении используется 3 базы данных MSSQL:

- Внутренняя база данных самого приложения для выдачи доверенностей **[DbLetterOfAttorney]**

- Внешняя база данных АРМ Отдела кадров **[D:\SO.MDF]** (Это название, а не путь. Легаси.)

- Внешняя база данных АРМ Поставщиков-Потребителей **[ConsumersDb]**

Во всех таблицах **внутренней базы данных** содержится 4 сервисных поля с *датой/временем создания записи* (SERVICE\_CREATE\_DATETIME), *пользователем создавшим запись* (SERVICE\_CREATE\_USER), *датой/временем последнего обновления записи* (SERVICE\_LAST\_MODIFY\_DATETIME) и *пользователем последним обновившим запись* (SERVICE\_LAST\_MODIFY\_USER). Все четыре поля, при вставке записи (выполнении INSERT) инициализируются DEFAULT значениями CURRENT\_TIMESTAMP (текущая дата/время) и SUSER\_SNAME() (пользователь, от имени которого выполняется запрос).

Также, для всех таблиц **внутренней базы данных**, написаны триггеры для события UPDATE, которые автоматически обновляют два соответствующих поля таблиц: *TriggerUpdateCargoes*, *TriggerUpdateLetterOfAttorneys* и т.д. Во всех триггерах существует проверка рекурсивности вызова. Все указанные триггеры, по сути, однотипны, за исключением *TriggerUpdateCurrentDbYear*, так как в таблице *CurrentDbYear*, с помощью специально написанного индекса, не может существовать более одной записи и обновление записи в триггере написано `для всех` записей.

Для всех таблиц, кроме *CurrentDbYear*, указаны первичные ключи *id* типа *bigint*.

Большинство таблиц **внутренней базы данных**, за исключением *CurrentDbYear* и *PreviousYearHistories*, связаны между собой вторичными ключами, с поведением NO ACTION: запрет удаления родительской сущности, при наличии дочерней (при DELETE), однако разрешается замена связанной сущности (при UPDATE).

|  |  |  |  |
| --- | --- | --- | --- |
| **Дочерняя табл.** | **Поле связи дочерней табл.** | **Вторичный ключ (название)** | **Родительская табл.** |
| [Cargoes] | [measuresId] | FK\_CargoesToMeasures | [Measures] |
| [LetterOfAttorneys] | [companiesId] | FK\_LetterOfAttorneysToCompanies | [Companies] |
| [LetterOfAttorneys] | [couriersId] | FK\_LetterOfAttorneysToCouriers | [Couriers] |
| [Shipments] | [letterOfAttorneysId] | FK\_ShipmentsToLetterOfAttorneys | [LetterOfAttorneys] |
| [Shipments] | [cargoesId] | FK\_ShipmentsToCargoes | [Cargoes] |

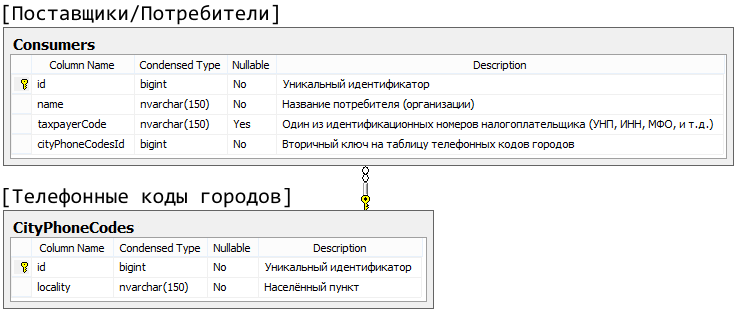
Для большинства таблиц **внутренней базы данных** определены ключи ограничения уникальности столбцов.

|  |  |  |
| --- | --- | --- |
| **Таблица** | **Ключ уникальности** | **Столбец (столбцы) ключа уникальности** |
| [Cargoes] | UQ\_Cargoes\_Name\_And\_MeasuresId | [name] + [measuresId] |
| [Companies] | UQ\_Companies\_Name | [name] |
| [Couriers] | UQ\_Couriers\_PassportSeriesAndNumber | [passportSeriesAndNumber] |
| [LetterOfAttorneys] | UQ\_LetterOfAttorneys\_OrdinalNumber | [ordinalNumber] |
| [Measures] | UQ\_Measures\_Name | [name] |

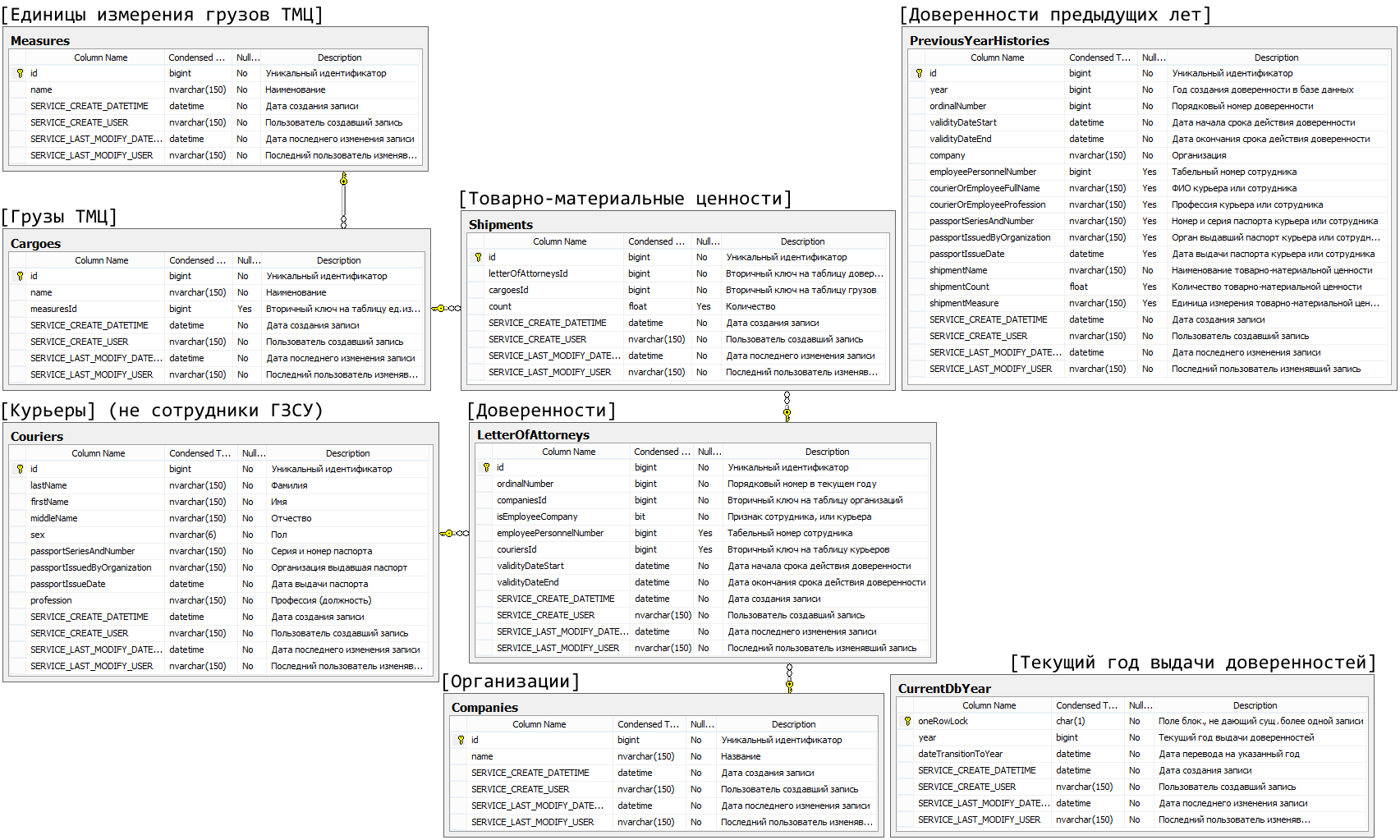
Для столбцов **внешней базы данных отдела кадров** никаких ключей на текущий момент не определено. Ввиду огромного количества столбцов таблиц этой базы данных, ничего не значащие в приложении столбцы таблиц из диаграммы связей ниже, опущены. На схеме можно увидеть только столбцы и таблицы участвующие в запросах.

Для столбцов **внешней базы данных поставщиков-потребителей**, также как и в предыдущем случае, на схеме указаны только столбцы и таблицы участвующие в запросах этого приложения.

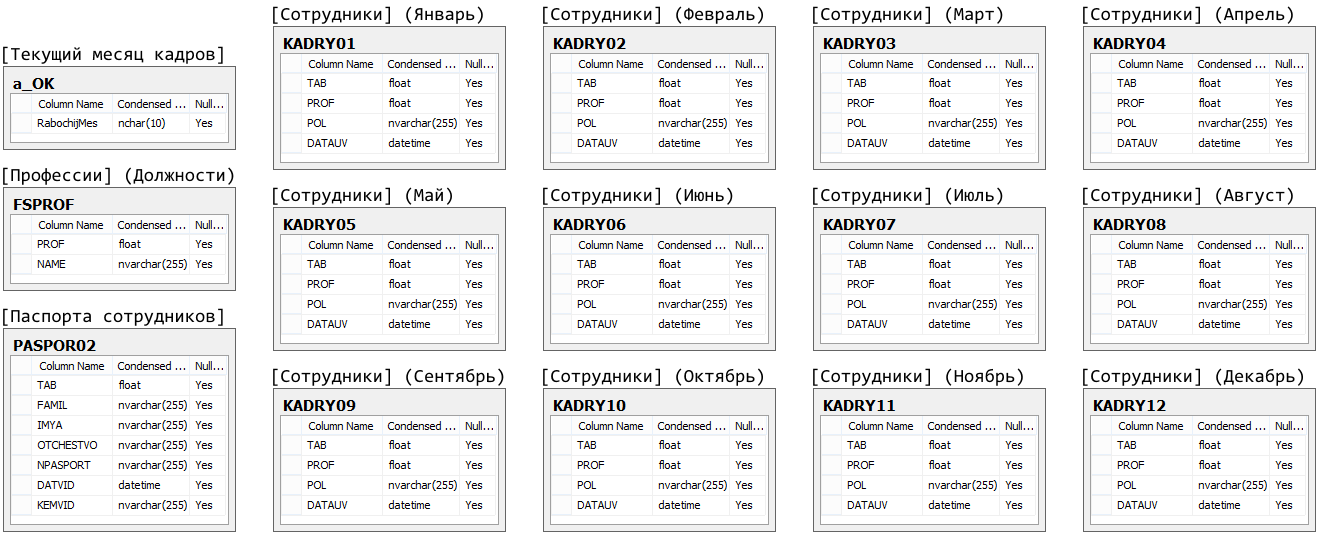
Внешняя база данных *Поставщиков-Потребителей* [**ConsumersDb**]



Внутренняя база данных *Доверенностей* [**DbLetterOfAttorney**]



Внешняя база данных *Отдела кадров* [**D:\SO.MDF**]



## Бизнес-логика приложения

Сотрудник предприятия или курьер, которому требуется выдача доверенности, обращается к специальному работнику материального бюро бухгалтерии. Данный сотрудник материального бюро добавляет доверенность в систему и распечатывает экземпляр доверенности. Дальнейшее редактирование доверенности ограничено, некоторые поля изменить нельзя (даты действия).

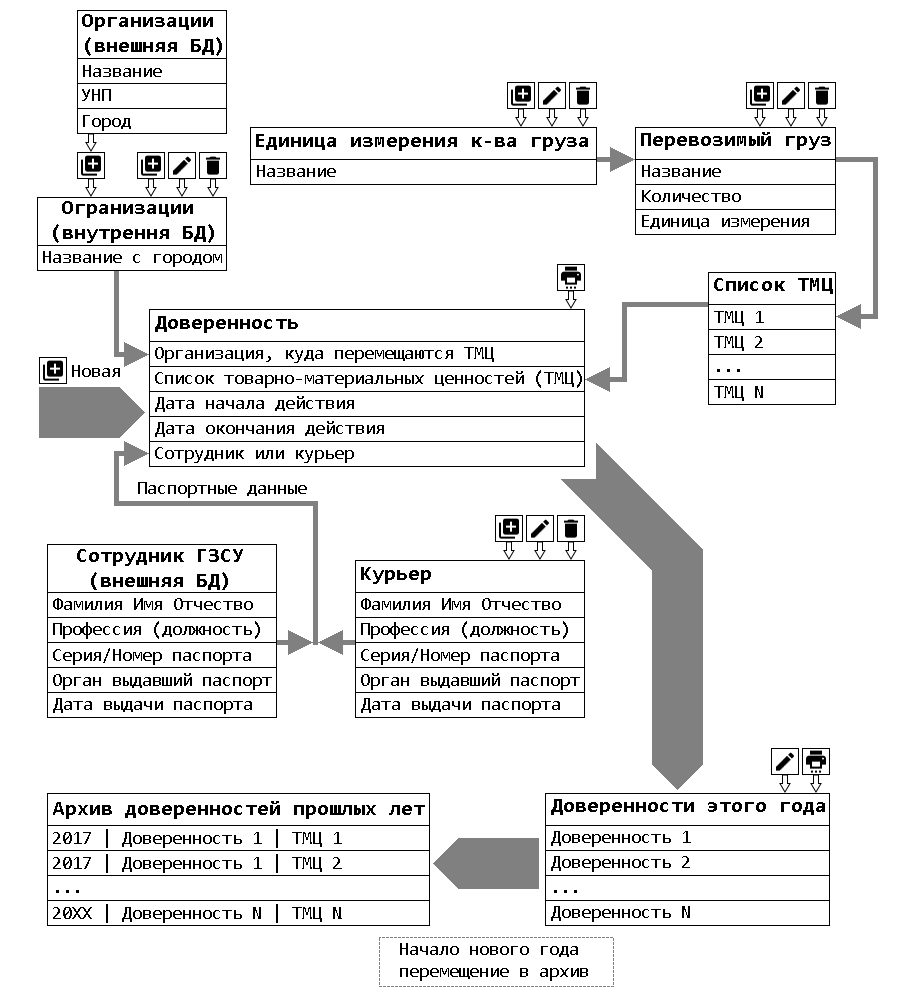
Порядковый номер доверенности присваивается автоматически, отсчитывается с единицы с начала года. В начале нового года (при первом же запуске в новом году), доверенности старого года перемещаются в архив. При этом запрашивается подтверждение у пользователя в специальном окне, требующем ввести кодовое слово вручную (во избежание ошибок с системным временем на ПК).

При вводе организации есть поиск организации из базы данных стороннего приложения АРМ Поставщиков-Потребителей, с возможностью поиска по УНП (ИНН, УНН и т.д.).

Некоторые поля при вводе ТМЦ могут быть опущены: количество и единица измерения количества груза.

При печати доверенности, фамилия, имя и отчество сотрудника или курьера автоматически преобразуется в дательный падеж. Также, при печати, если указано количество ТМЦ (вещественное или целое число), оно преобразуется в строку автоматически, скриптом.

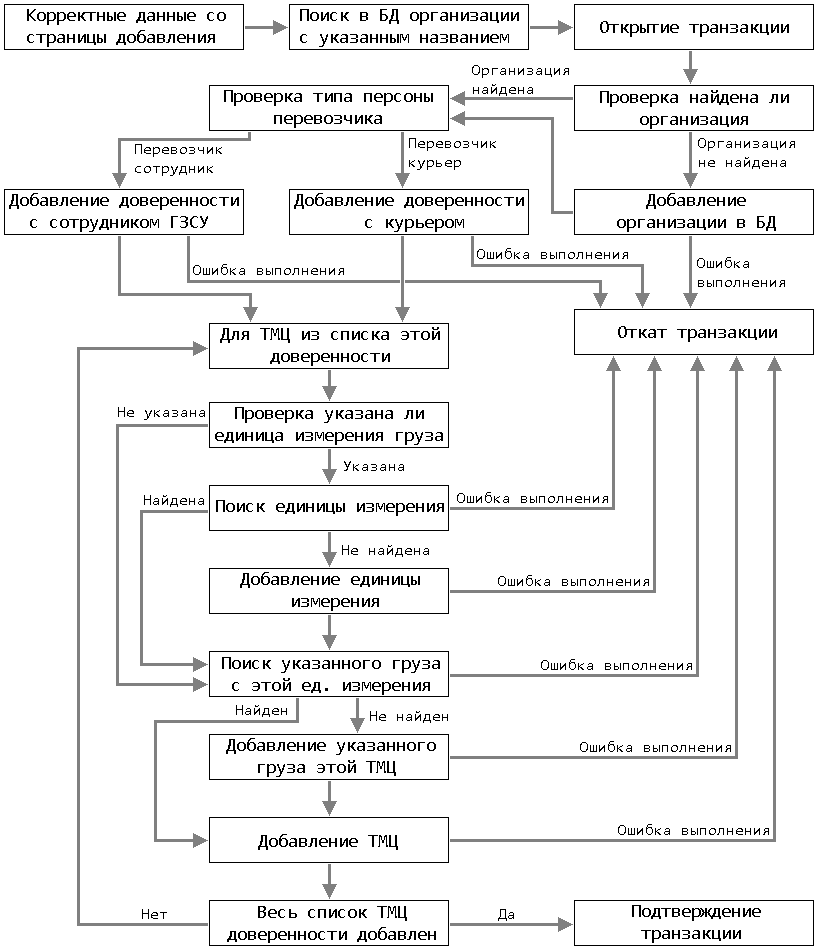
По всем табличным данным системы можно осуществить поиск, нажав правой кнопкой мыши по нужному столбцу и введя искомое значение.



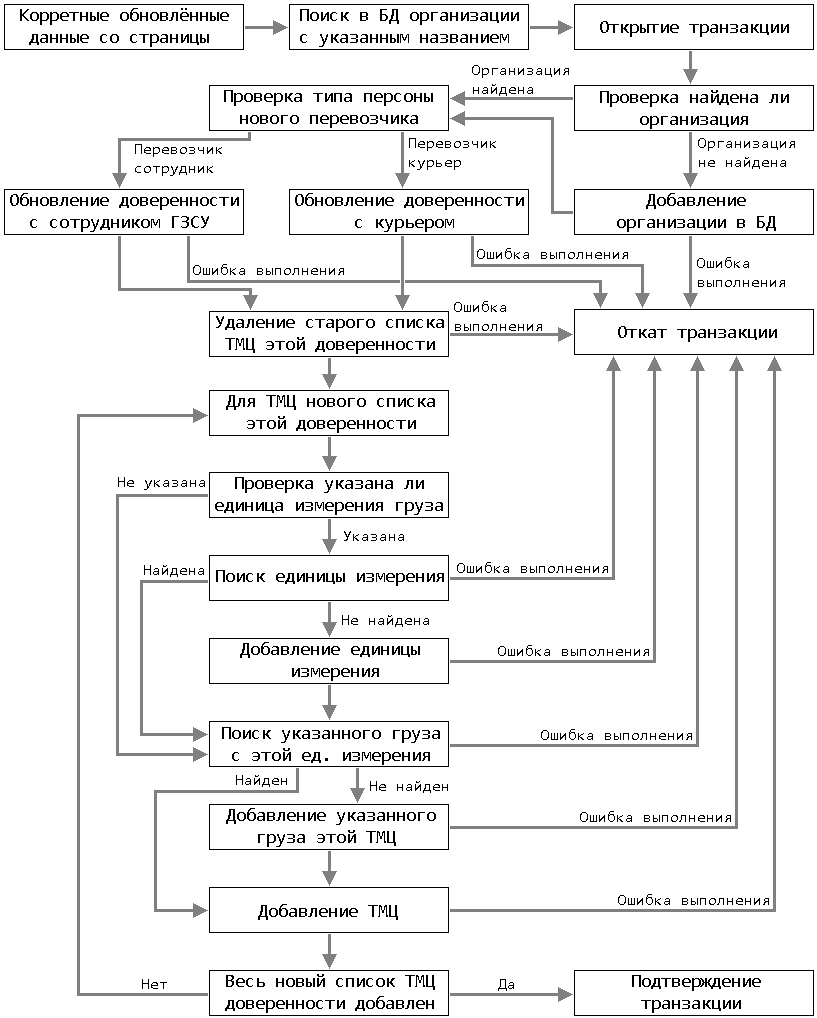
## Транзакции сервисного слоя редактирования доверенностей

Основная масса операций сервисного слоя, для простых сущностей, элементарна и не требует пояснений. Исключение составляется логика добавления и обновления доверенностей. Указано на схемах ниже.

**Добавление доверенности в систему**

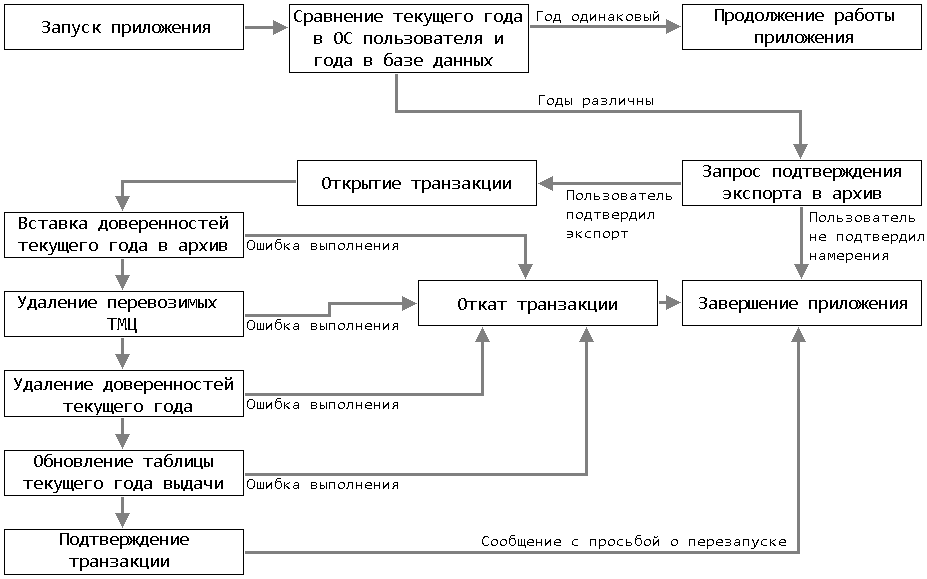


**Обновление доверенности**



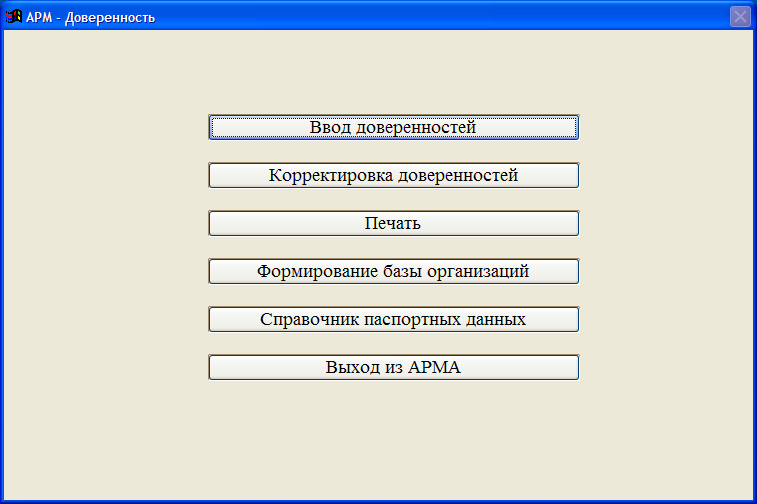
## Описание верификации при запуске приложения и транзакции экспорта в архив

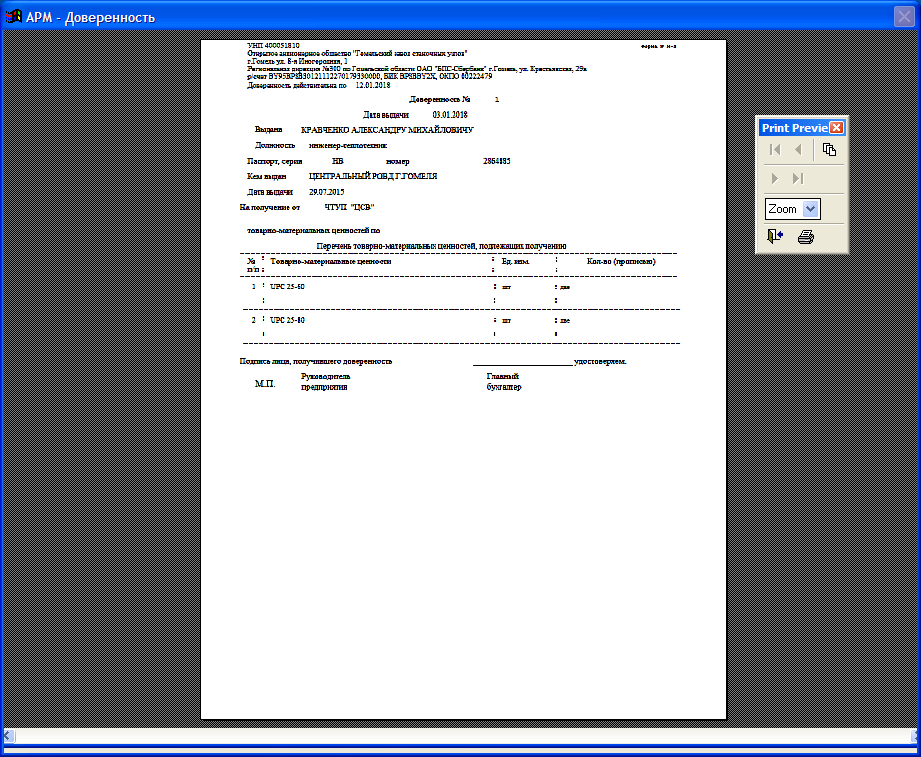
При запуске приложения происходит проверка необходимости экспорта и экспорт. Схема проверки и процесса перемещения в архив указана ниже.

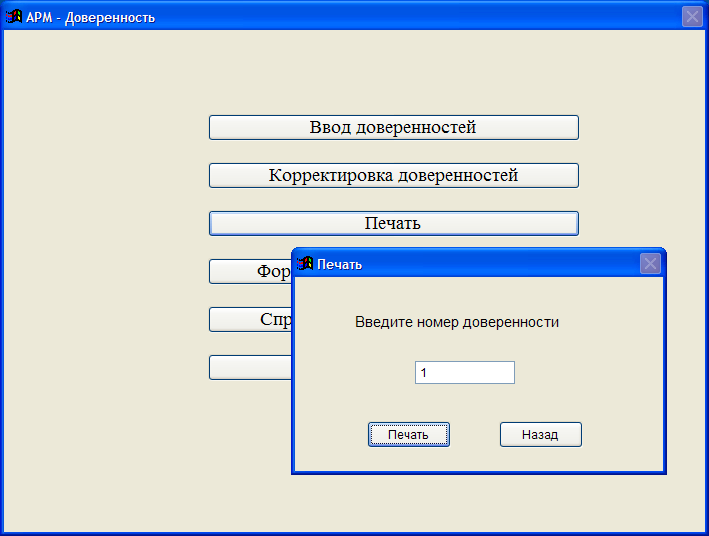


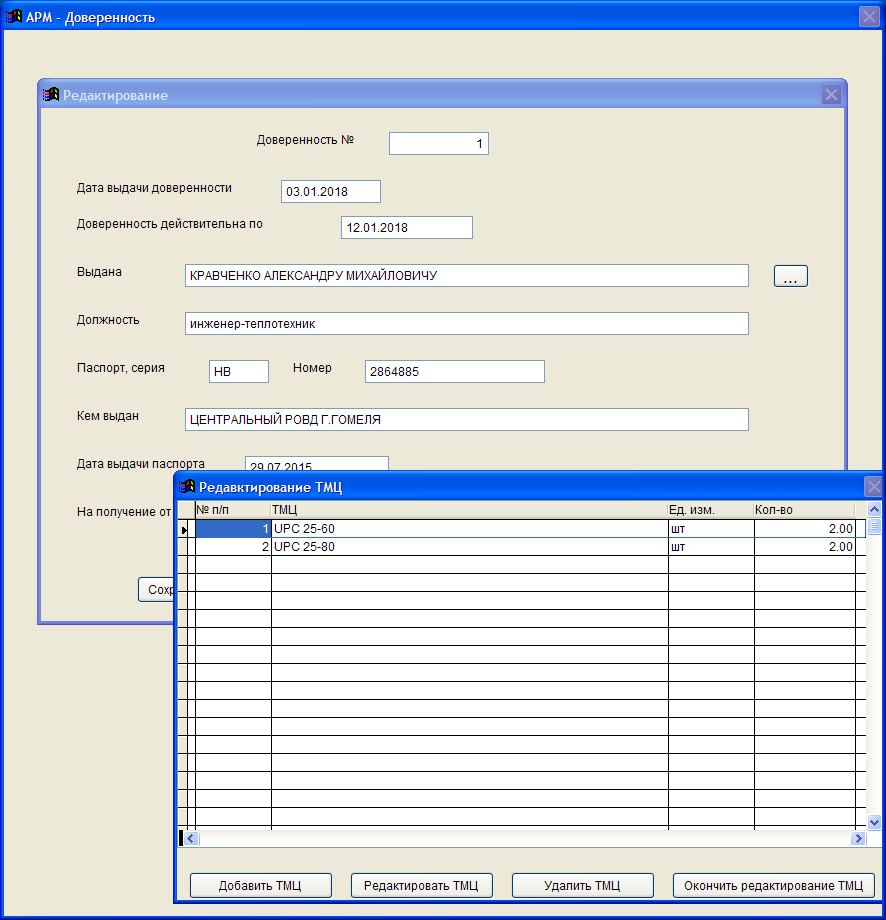
## Краткое описание старого приложения на Visual FoxPro 2008-го года

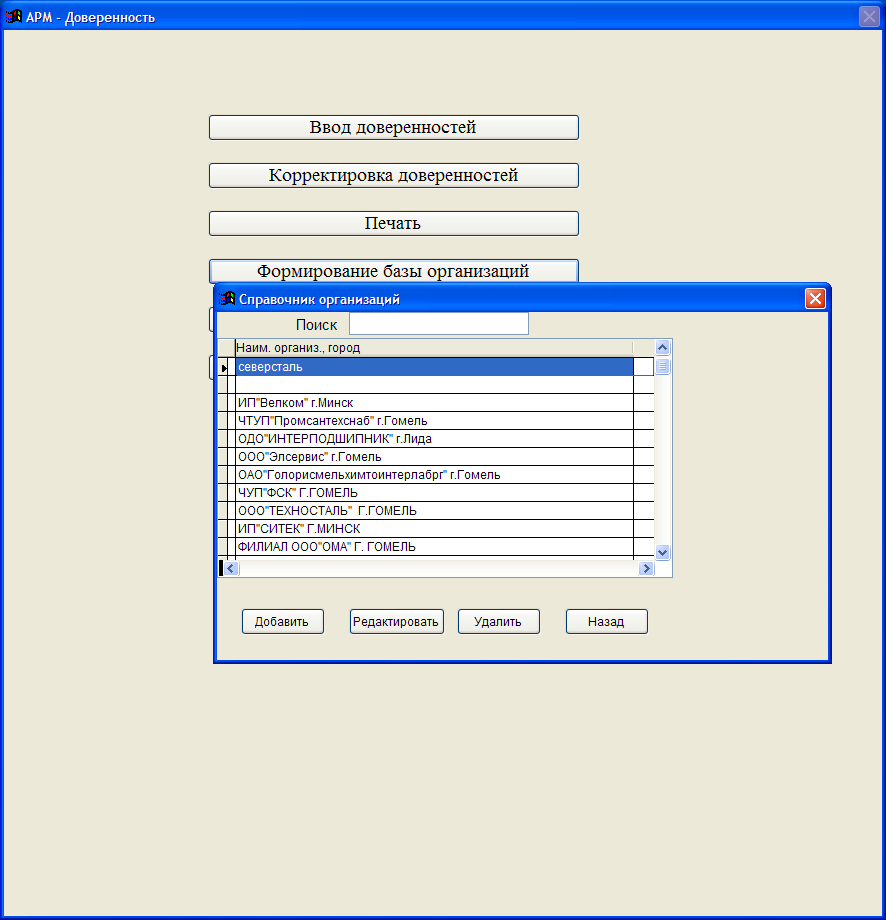
Старое приложение было написано в 2008-году каким-то студентом, на основании его проектировалось и разрабатывалось данное решение. Основной функционал тот же, экспорт в архив, при наступлении нового года, осуществлялся в ручном режиме копированием локальным копированием DBF. Приложение локальное, актуальные базы данных сотрудников требовалось периодически копировать вручную. Ниже представлены скриншоты процесса работы.

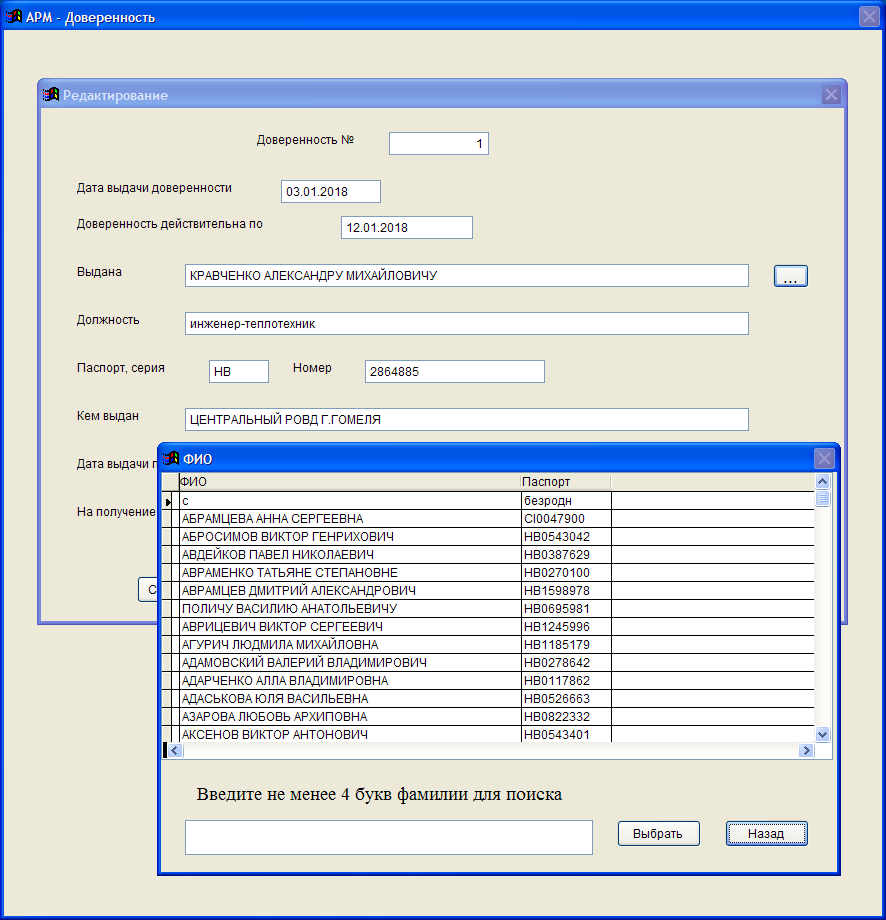












## Развёртывание приложения, пояснения про IDE и зависимости

Приложение собрано с поддерживаемой версией .Net Framework 4.0, на языке C# 4.0, с явным указанием сборки в x86. Последнее нужно было на этапе разработки, когда использовались базы данных DBF, драйвер которых не работает со сборками x64 или с выставленным по умолчанию значением Any CPU. В принципе, сейчас для этого приложения это уже не обязательно, но отдельно не тестировалось, и могут всплыть дефекты - лучше оставить как есть.

Благодаря указанию версии .Net Framework 4.0 возможна работа на Windows XP и Windows Server 2003 - требуется только установить его со всеми зависимостями в ОС.

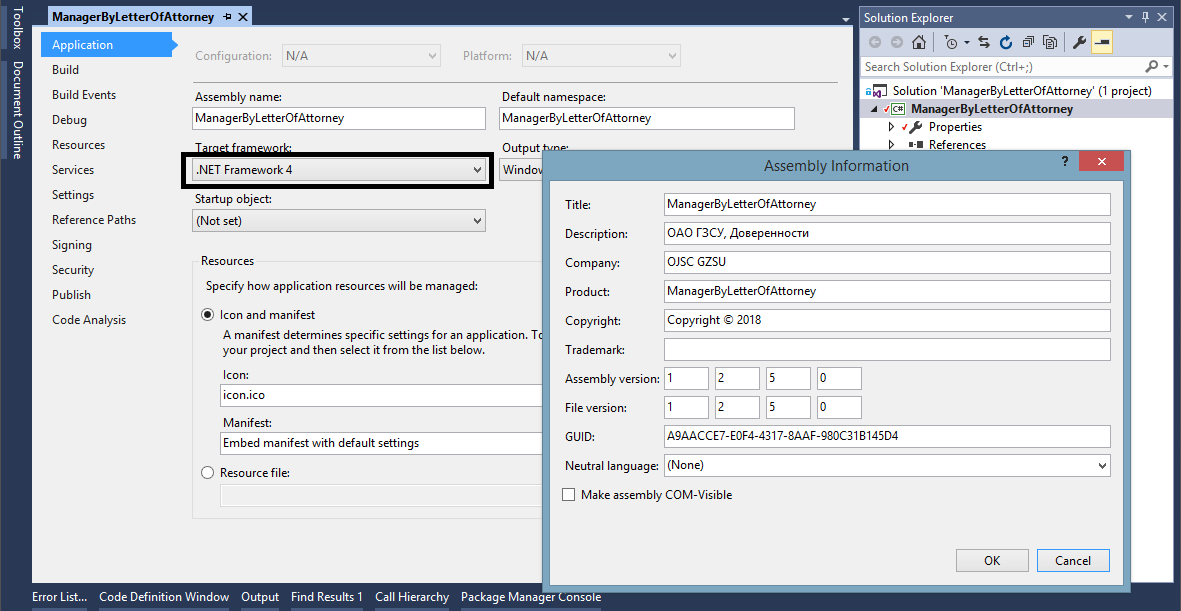
Сам проект изначально создавался в IDE Visual Studio 2010 SP1, и доработка функционала может осуществляться в ней же.

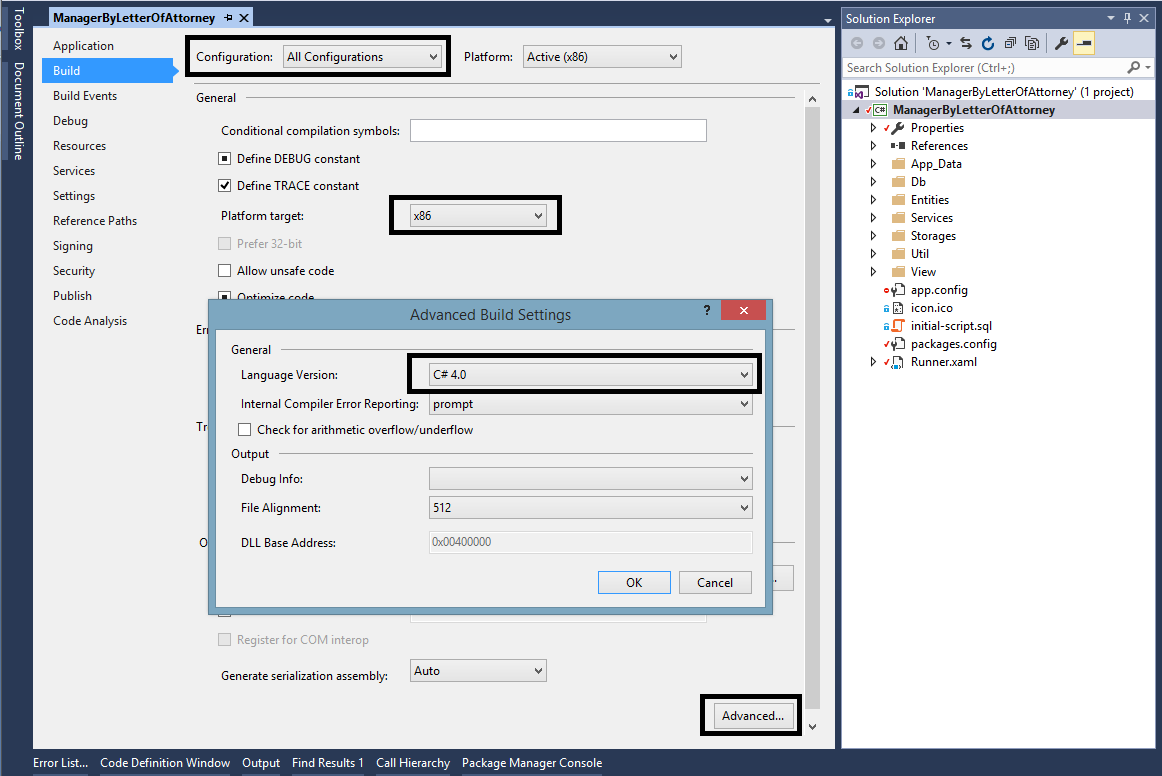
В этом случае нужно указывать версию языка default, как указано в таблице ниже.

|  |  |  |  |
| --- | --- | --- | --- |
| **Версия IDE** | **Платформы .NET Framework** | **Версии языка C#** | **Default версия C#** |
| VS 2008 | 2.0, 3.0, 3.5 | ISO-1, ISO-2, default | C# 3.0 |
| VS 2010 | 2.0, 3.0, 3.5, 4.0 | ISO-1, ISO-2, C# 3.0, default | C# 4.0 |
| VS 2012 | 2.0, 3.0, 3.5, 4.0, 4.5 | ISO-1, ISO-2, C# 3.0, C# 4.0, default | C# 5.0 |
| VS 2015 | 2.0, 3.0, 3.5, 4.0, 4.5, 4.6 | ISO-1, ISO-2, C# 3.0, C# 4.0, C# 5.0, default | C# 6.0 |
| VS 2017 | 2.0, 3.0, 3.5, 4.0, 4.5, 4.6, 4.7 | ISO-1, ISO-2, C# 3.0, C# 4.0, C# 5.0, C# 6.0, default | C# 7.0 |

Для работы в более современных IDE, но с сохранением обратной совместимости кода, следует указывать версию языка C# 4.0, дабы случайно не использовать современные языковые фичи и не сломать обратную совместимость. Для VS15 пример указан на скриншотах ниже.

Вполне закономерный вопрос - зачем использовать старые IDE, когда есть современные версии? Дело в низких мощностях компьютеров отдела программирования - на данный момент SSD-дисков нет ни на одной машине разработчиков, процессоры используются какие-то двуядерные Seleron (которые слабее Intel Core i3), память DDR4, но небольшого объёма (не выше 4гб), мониторы не широкоформатные 15/17/19 дюймов. Это описание топовых машин, в основной массе ещё хуже. На одной из машин, установка только VS15 заняла примерно 8 часов, на топовых чуть быстрее, часа за 4. При работе, с установленным расширением ReSharper, возникают задержки для элементарных операций вроде открытия файла/ребилд проекта в 5-10-60 секунд, в зависимости от фазы луны и режима работы неотключаемого антивируса. Относительно комфортная работа в этой связке возможна лишь на собственном ноутбуке с 250гб SSD, 8гб ОЗУ, широкоформатным монитором 17 дюймов + внешний кубик в 17. Если же использовать только рабочие компьютеры, работать можно лишь в старых IDE, они потребляют меньше ресурсов и можно написать за день хоть что-то.





Для работы отчётов на клиентских машинах, используется библиотека ReportViewer, с несколько замороченной логикой обратной совместимости различных версий отчётов и runtime-библиотек. Таблица runtime-библиотек, требуемых на клиентских машинах:

|  |  |  |
| --- | --- | --- |
| **VS** | **Требуемая версия Runtime** | **Засисимости Runtime** |
| VS05 | [ReportViewer Runtime 2005.exe](https://www.microsoft.com/en-us/download/details.aspx?id=21916) |  |
| VS08 | [ReportViewer Runtime 2008.exe](https://www.microsoft.com/en-us/download/details.aspx?id=6576) |  |
| VS10 | [ReportViewer Runtime 2010.exe](https://www.microsoft.com/en-us/download/details.aspx?id=6442) |  |
| VS12 | [ReportViewer Runtime 2012.msi](https://www.microsoft.com/en-us/download/details.aspx?id=35747) | System CLR Types for SQL Server 2012 (x86/x64) |
| VS15 | [ReportViewer Runtime 2015.msi](https://www.microsoft.com/en-us/download/details.aspx?id=45496) | System CLR Types for SQL Server 2014 (x86/x64) |
| VS17 | <runtime не поставляется Microsoft, всё должно работать и без него> | |

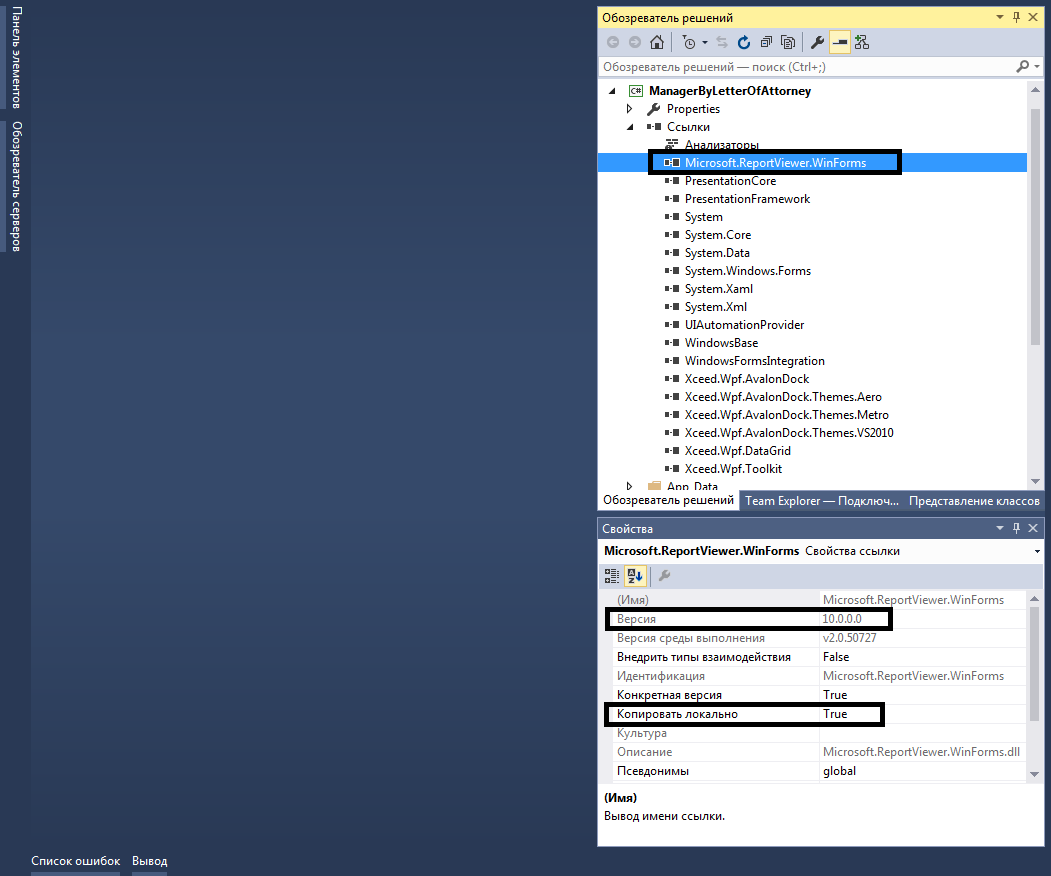
Ссылки на скачивание зависимостей Runtime можно получить на странице скачивания Runtime в подменю Related Resources.

Для повышения шанса работоспособности отчётов на конечной машине, **лучше установить все версии Runtime и зависимостей**. Для установки требуются права администратора. На **машине разработчика** также **лучше поставить все версии**.

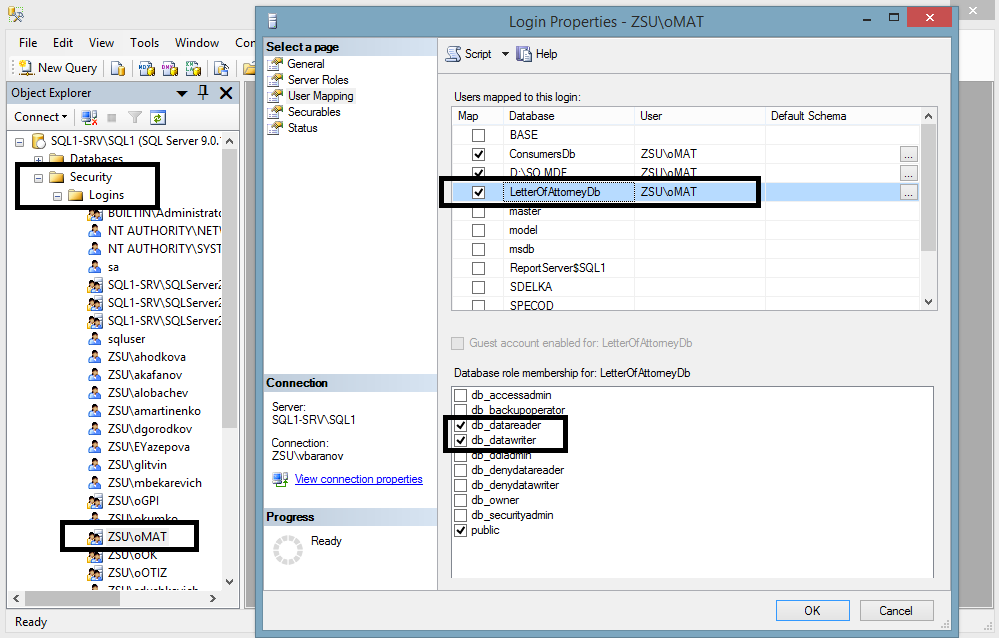
Также у самих RDLC отчётов плохая обратной совместимость - отчёт созданный более новой версией Visual Studio 2017 не откроется в более старой 2015. Приложения, которое может конвертировать отчёт в более старую версию, найти не удалось.

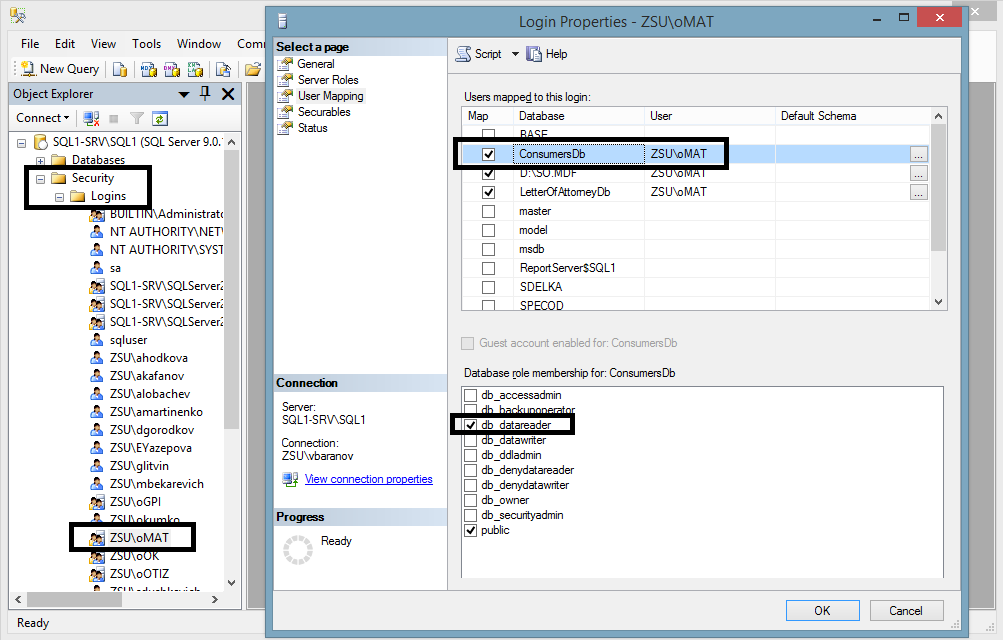
Однако, если в проекте VS15 установлена зависимая ссылка на dll Microsoft.ReportViewer.WinForms версии 10 (версии IDE VS10) и в системе установлен соответствующий runtime, отчёты будут открываться в последующем и в VS10.

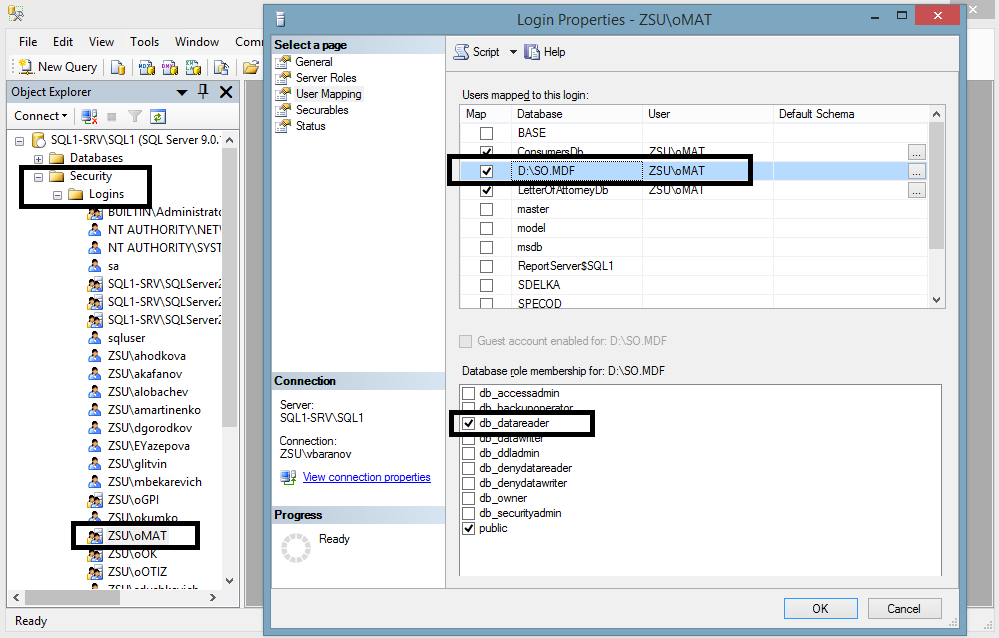
Скриншоты рабочей сборки, созданной в VS10 и открытой в VS15 без замены версии ссылки на нужный Runtime (используется ReportViewer из VS10) указаны ниже.



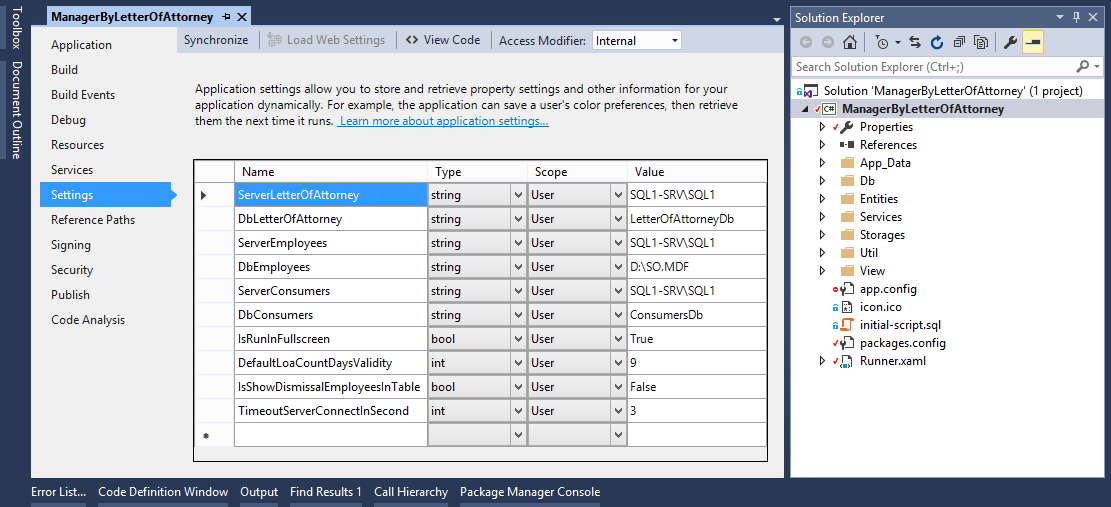
Для работы приложения должно быть права доступа на 3 базах данных: база данных доверенностей (DbLetterOfAttorney), база данных отдела кадров (D:\SO.MDF) (это название, а не путь - легаси бд) и база данных поставщиков-потребителей (ConsumersDb). Для всех внутренних баз данных должен быть permission db\_datareader/db\_datawriter, для внешних баз только db\_datareder. Пример для компьютера материального бюро бухгалтерии (oMat) представлен на скриншотах ниже.





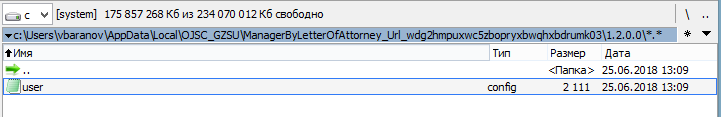


Серверы баз данных и названия баз данных указываются в параметрах приложения (Properties), и могут быть изменены пользователем на своей машине для указанной версии приложения (если свойства были изменены для версии 1.1.2, они не будут применены для версии 1.1.3). Скриншот окна IDE со свойствами по умолчанию указан ниже.



В случае изменения свойств по умолчанию конечным пользователем (не вручную, а в специальном окне приложения), данные для текущей версии будут сохранены по примерно такому пути:

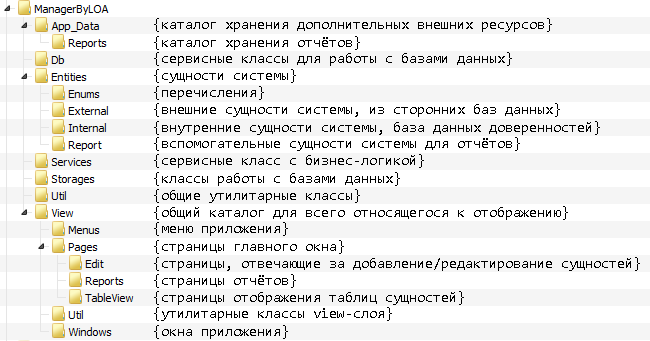
c:\Users\vbaranov\AppData\Local\OJSC\_GZSU\ManagerByLetterOfAttorney\_... \1.2.5.0\user.config



## Архитектура системы

Архитектура системы, по своей сути, состоит 4 основных категорий классов: ооп иерархия сущности системы (entities) проецирующие предметную область, слой работы с данными (storages) отвечающий за работу с таблицами источника данных, слой сервисов (services) инкапсулирующий бизнес-логику и управление транзакциями, и классы слоя отображения информации пользователю (view) со своими подразделами. В эти категории не входят утилитарные вспомогательные классы приложения, размещённые в соответствующих каталогах, но они также разделены на общие классы, классы относящиеся ко view-слою и классы хелперы БД.

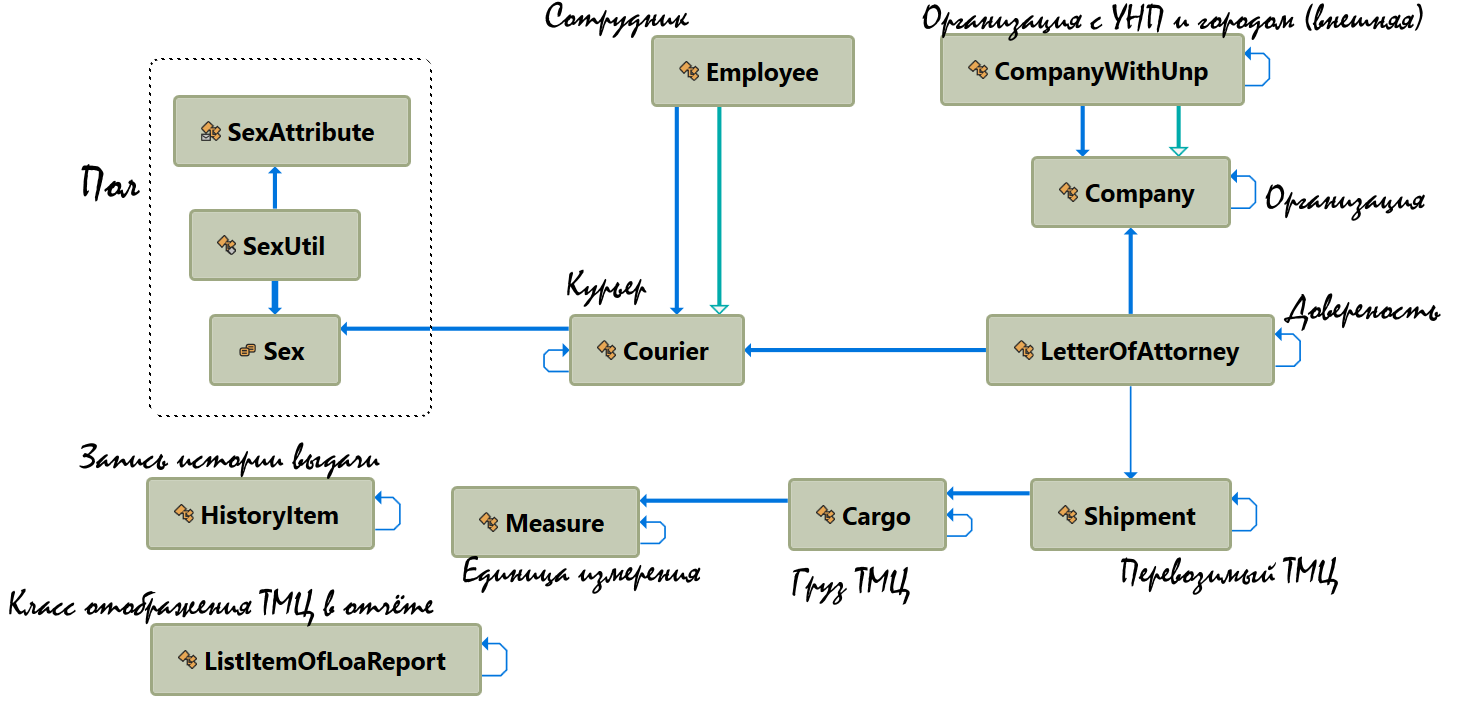
Каталоги/подпространства имён системы:



ООП-иерархия сущностей системы разделена на несколько категорий: внутренние редактируемые сущности системы (internal), внешние не редактируемые из сторонних баз данных (external). Особняком стоит категория перечислений (enum) и временных буферных классов для отчётов, которые не мапятся в базу, а формируются для удобства вывода в отчёте (reports).

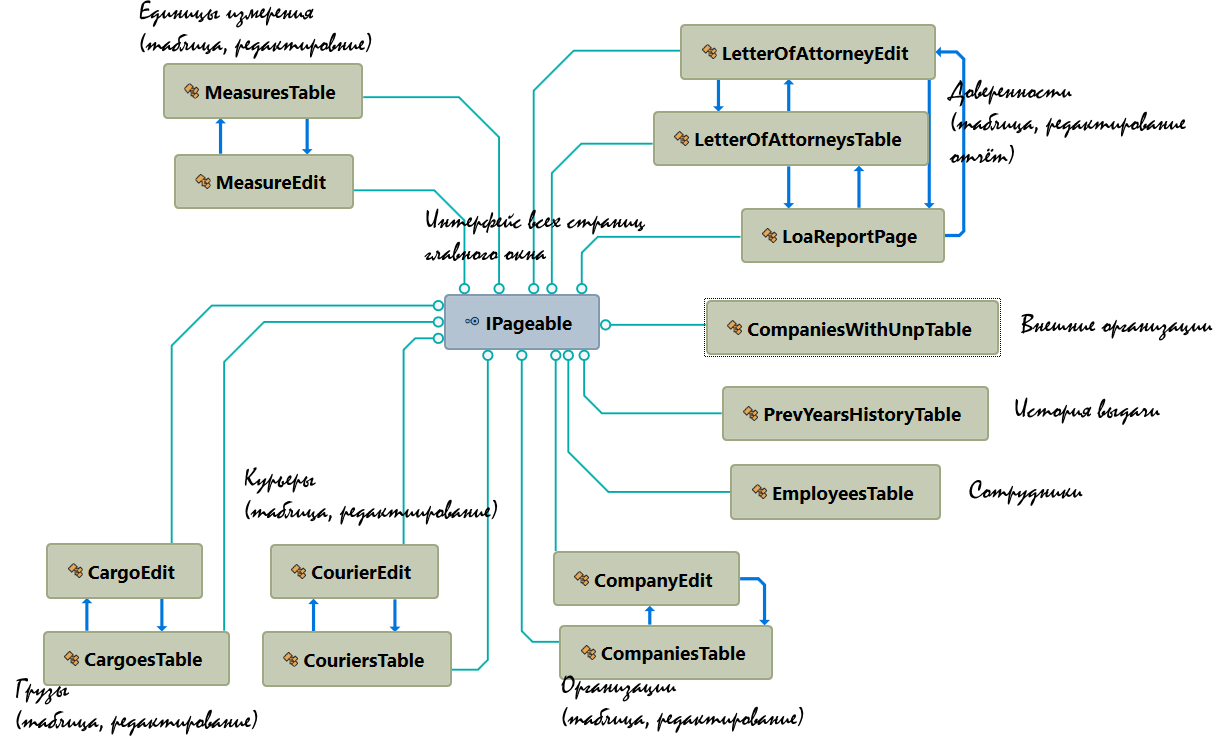
Наследуемых сущностей 2: внешняя сущность сотрудник организации наследует внутреннего курьера и внешняя сущность организации с указанным УНП и городом наследует внутренний класс организации.

Ниже, на сформированной утилитой картинке, указана связь классов между собой и отношения наследования между ними.



View-слой системы разделён на несколько подкатегорий: окна приложения, меню приложения, утилитарные классы и страницы главного окна. Страницы главного окна также разделены на подкатегории, в зависимости от их назначения: tableview - отображение таблиц сущностей, edit - добавление/редактирование сущностей, report - отвечает за предварительный просмотр и печать отчётов. Все страницы имплементируют интерфейс IPageable, с парой нескольких служебных методов страниц. Страницы главного окна размещаются в специальном фрейме, вызовом специального метода в класс PageSwitcher с указанным параметром нужной страницы.

На картинке ниже указаны страницы главного окна системы, логически сгруппированные по сущностям к которым они относятся.



Основной функционал системы реализован в страницах главного окна, но в некоторых случаях требуются сторонние окна:

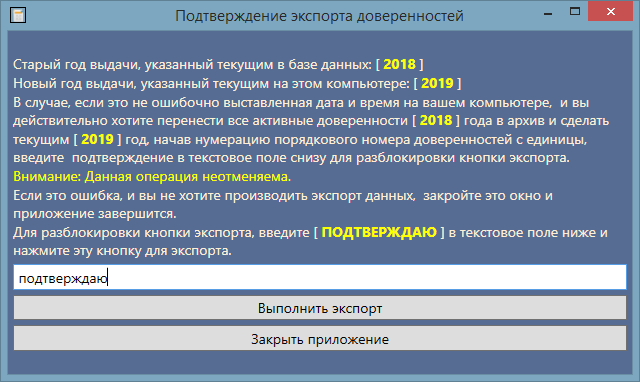
- окно пользовательских настроек (UserConfigWindow)

- окно отображения детальной информации о возникшем исключении (ErrorDetailWindow)

- отдельное окно редактирования курьера, для возможности добавления/изменения курьера во время ввода доверенности (CourierEditExtraWindow). В окне размещается страница главного окна редактирования курьера, со срабатыванием action-делегата для закрытия

- окно подтверждения экспорта доверенностей в архив (ConfirmExportWindow).

Последнее изображено на скриншоте ниже.



## Оффлайн утилита печати доверенности для экстренных случаев (отсутствие сети)

За последние 2 месяца работы приложения, примерно 4 раза возникала потребность срочной печати доверенности, в условиях, когда были недоступны сетевые диски, на которых лежит каталог с приложением или были недоступны/выключены сервера mssql (выход из строя автоматического выключателя на сервере, проблемы с батареями бесперебойника, неудачное обновление MS Windows Server). Для этих случаев было написано простенькое приложение из 2 окон для печати доверенности , работающее только на локальной машине, не использующее серверы баз данных или иных сетевых подключений. Результат ввода никуда не сохраняется, только печать для экстренных случаев, когда нужно выдать документ срочно. Демонстрационные скриншоты ниже.

