

ООО ЭкоСофт

Документация по платформе Lexeta

Уфа-2016

Оглавление

Приступая к работе	6
Установка компонентов системы	6
Настройка моделлера приложения	6
Регистрация учётной записи администратора организации	7
Работа в личном кабинете и назначение роли администратору	10
Создание учётных записей пользователей организации	12
Назначение роли пользователю организации.....	13
Создание документа на основе модели.....	14
Создание модели и её полей.....	14
Создание полей модели	15
Обновление БД (миграция)	17
Создание экранных форм.....	18
Создание реестра и его связь с формой редактирования	19
Предоставление прав на созданные объекты.....	20
Предоставление прав доступа к экранным формам	20
Предоставление прав доступа к модели	22
Создание пункта меню для открытия реестра документов	23
Создание новой группы (категории) меню	23
Добавление пункта (элемента) меню	23
Проверка работоспособности созданного документа	24
Настройка параметров отображения формы и её элементов	25
Обновление страницы после изменения экранной формы	26
Указание заголовка формы	26
Переименование полей экранной формы	27
Переименование полей реестра	28
Удаление полей формы и реестра	29
Добавление полей в форму	29
Замена поля типа TextVox на Мемо	30
Создание дочерней коллекции документа.....	30
Создание дочерней модели.....	30
Привязка дочерней модели к родительской	31
Добавление в экранную форму дочерней таблицы	32
Проверка работоспособности документа с дочерней коллекцией.....	32
Настройка параметров отображения формы с дочерней коллекцией	33
Удаление системных полей	34
Переименование колонок	34

Изменение размеров колонок	35
Создание справочников и их подключение к документу	35
Создание модели справочника	35
Добавление в основную модель поля справочника	36
Создание ЭФ и реестра для ввода значений справочника	37
Добавление выпадающего списка в ЭФ основного документа	37
Добавление поля справочника в реестр	38
Создание многоколоночного списка	38
Проверка работоспособности документа со справочником	39
Справочники с большим количеством данных	41
Проверка работоспособности списка с постраничным выводом	42
Создание аналитической формы на основе запроса	43
Написание и отладка SQL-запроса на выборку	43
Создание запроса (источника данных) в Lexema.ru	44
Настройка параметров запроса	44
Ввод текста запроса	45
Предоставление прав на запрос	45
Создание экранной формы	46
Создание элементов управления для ввода значений параметров запроса	47
Создание таблицы и настройка параметры её колонок	48
Создание скрипта для отображения данных	49
Проверка работоспособности аналитической формы	50
Создание отчёта на основе запроса.....	50
Создание запроса и формы на его основе	51
Конструирование отчёта.....	51
Запуск приложения для конструирования отчётов	51
Создание отчёта	51
Создание областей отчёта	51
Настройка содержимого областей	52
Привязка источника данных к запросу	53
Создание таблицы данных	54
Создание "шапки" таблицы	55
Настройка границ таблицы	57
Предварительный просмотр отчёта	57
Сохранение отчёта	58
Привязка отчёта к форме	59
Создание скрипта открытия отчёта	61

Проверка работоспособности отчёта	61
Работа с вычислениями в экранной форме	62
Создание обработчика открытия ЭФ (Created)	63
Создание обработчика получения экранной формой сообщения (GetMessage)	64
Создание обработчика закрытия ЭФ (Closing)	64
Создание обработчиков событий для элементов ЭФ	65
Обращение к свойствам элементов формы в скриптах	65
Элементы управления экранных форм	65
RootItem	67
Toolbar	67
SaveToolbarItem	67
CloseToolbarItem	67
SaveCloseToolbarItem	67
NewDocumentToolbarItem	68
DeleteToolbarItem	68
UndoToolbarItem	68
RedoToolbarItem	68
ReportToolbarItem	68
SelectPeriodToolbarItem	68
CalculationToolbarItem	69
SaveAndNewToolbarItem	69
ToolbarPopupButton	69
Создание обработчиков изменения полей модели	69
Создание обработчика	69
Обращение к полям модели	70
Возможные проблемы	70
Проверка работоспособности скрипта	70
Использование событий модели	71
Создание обработчиков событий модели	71
Логика работы обработчиков	72
Обращение к значениям полей	72
Вывод сообщения об ошибке	72
Проверка работоспособности обработчиков	73
Отладка скриптов в экранной форме	73
Использование консоли браузера Google Chrome	73
Вывод информации в консоль	74
Вывод ошибок и предупреждений	74

Группировка записей	74
Измерение времени выполнения скрипта	74
Доступ к данным документа и к свойствам элементов формы в консоли	74
Чтение и изменение значений полей модели документа	74
Чтение и изменение значений элементов управления экранной формы	75
Доступ к полям модели и элементам управления из консоли	75
Анализ запросов с помощью SQL Profiler	76
Lexema SQL Profiler	76
MS SQL Server Profiler.....	77
Запуск и подключение к серверу	78
Настройка параметров трассировки	79
Отслеживание событий	82
Отправка электронной почты средствами MS SQL Server DBMail.....	84
Примеры сценариев использования DBMail.....	85
Параметры безопасности	85
Настройка компонента Database Mail	85
Регистрация учётной записи электронной почты на сервере	85
Использование мастера настройки	85
Создание учётной записи	87
Создание профиля	88
Отправка тестового сообщения	90
Диагностика неисправностей	90
Выполнение SQL-запроса для отправки сообщений из скриптов Lexema.ru	91
Возможные проблемы при отправке.....	94
Тезаурус	95
Базовые понятия системы Lexema.ru	95
Документ	96
Реквизит документа	96
Дизайнер приложения.....	96
Веб-клиент	96
Справочник.....	96
Различия между документом и справочником	96
Модель	97
Представление БД (View)	97
Обновление базы данных.....	97
Запрос.....	97
Экранная форма	97

Реестр документов	97
Аналитическая форма	98
Элемент экранной формы	98
Программный сценарий.....	98
Архитектура системы	98
Веб-клиент	98
Поддержка веб-браузеров	99
Концепция одностраничного приложения (Single page application)	99
Сервер приложений.....	99
Сервер БД	99

Приступая к работе

Для начала работы с системой Lexema.ru необходимо выполнить ряд действий:

- установить все компоненты системы
- зарегистрировать учётную запись пользователя, который будет администратором организации
- назначить необходимые права администратору организации
- создать учётные записи пользователей организации
- назначить права пользователям организации

Установка компонентов системы

Дистрибутив содержит два файла: пакет setup.msi для установки серверной части системы и архив modeller.zip, содержащий приложение для разработки конфигураций Lexema.ru.

1. Запустить установочный пакет setup.msi на сервере с IIS и Microsoft Sql Server
2. Указать каталог для расположения файлов системы
3. Ввести начальные настройки системы
 1. Адрес сервера БД
 2. Имя системной БД
 3. Имя прикладной БД
 4. Логин и пароль пользователя сервера БД
 5. Адрес, порт и атрибуты SMTP-сервера, который будет использоваться системой для отправки почты (по умолчанию - настройки SMTP-сервера mail.ru)
 6. Адрес электронного почтового ящика и пароль к нему (по умолчанию - логин и пароль к почтовому ящику LexemaRu@mail.ru)
4. Дождаться окончания установки

Указываемый пользователь сервера БД должен иметь права на создание баз данных и права на чтение и запись.

Для установки моделлера достаточно распаковать файлы в любое удобное расположение и запустить файл Lexema.exe.

После окончания установки откроется веб-страница с формой входа в систему. Для того, чтобы войти в систему, необходимо пройти процедуру регистрации либо воспользоваться логином тестового пользователя Test с паролем Test1111.

Настройка моделлера приложения

Моделлер предназначен для разработки прикладных конфигураций и представляет собой настольное windows-приложение. Моделлер не требует установки, достаточно скачать его по ссылке. После запуска моделлера необходимо настроить подключение к базе данных прикладной логики. Для этого нажмите кнопку "Дополнительно" и затем кнопку "..." рядом с полем "Вход в ...":

Lexema[®] 7.0 ЭкоSoft

Имя пользователя sa

Пароль

Вход в ...

Сервер

База данных

RU OK Отмена >> Дополнительно

В окне "Настройка подключений" нажмите кнопку "Добавить", в форме "Детализация соединения" введите название подключения, адрес сервера БД и имя его экземпляра (при наличии) и имя БД прикладной логики:

Детализация соединения

Наименование local base

Сервер 192.168.0.200

База данных L8

OK Отмена

Нажмите кнопку ОК, затем "Закреть". В форме авторизации введите логин и пароль пользователя сервера БД, убедитесь что значение в поле "Вход в ..." выбрано верно и нажмите ОК. Если данные были введены верно, то моделлер приложения успешно загрузится и можно будет приступить к работе с прикладной конфигурацией системы.

Регистрация учётной записи администратора организации

Следует иметь ввиду, что в результате самостоятельной регистрации с главной страницы создаётся новая организация, а зарегистрированный пользователь становится её администратором. Для начала этого процесса необходимо на странице авторизации нажать кнопку "Регистрация":



Вход в систему

Логин

Пароль

Запомнить пароль

[Войти](#)

[Восстановление пароля](#)

[Регистрация](#)

Далее необходимо заполнить все поля формы, при этом указав корректное название компании и существующий адрес e-mail:



Регистрация в системе

Пожалуйста, заполните все поля

Компания	<input type="text" value="TestCompany"/>
Имя	<input type="text" value="Владимир"/>
Фамилия	<input type="text" value="Алексеев"/>
Отчество	<input type="text" value="Николаевич"/>
Логин	<input type="text" value="AlekseevVN"/>
Эл. почта	<input type="text" value="test@company@gmail.com"/>
Пароль	<input type="password" value="....."/>
Повторите пароль	<input type="password" value="....."/>
<input type="button" value="Зарегистрироваться"/>	

Пароль должен быть достаточно сложным и удовлетворять ряду условий:

- минимальная длина - 8 символов
- должен содержать цифры, строчные и заглавные буквы

В случае нарушения этих условий будет отображено сообщение об ошибке:

Пароль	<input type="password" value="..."/>	<div style="border: 1px solid red; padding: 5px; color: red;">Задайте пароль, используя цифры, строчные и заглавные буквы. Минимальная длина пароля 8 символов</div>
--------	--------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

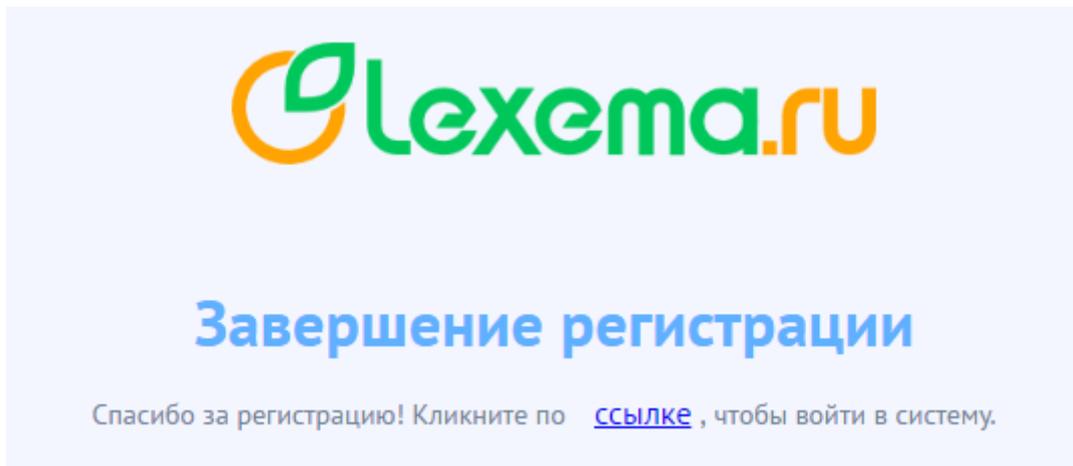
После нажатия на кнопку "Зарегистрироваться" будет создана учётная запись администратора, но до начала её использования необходимо подтвердить регистрацию, пройдя по ссылке в письме, пришедшем на указанный электронный адрес:

 **lexema@mail.ru**
кому: мне ▾

 английский ▾ > русский ▾ [Перевести сообщение](#)

Пожалуйста, активируйте аккаунт, перейдя по ссылке: <http://demo.lexema.ru/Account/ConfirmEmail.aspx?userId=c88f5732bsBAAAAHqi56JW4uUWrqq315OfXwAAAAACAAAAAADZgAAwAAAAABAAAADtro0TBymdczKQVEtU7PXEAASAAAAASA2bSBcf2uV7bgasV2nltHPMPowmqLOUorSQFQk9X6zZQ2AKzv%2fInZEJjcpqb5d7F8lxQvJbyHp9To%2bOjInAbZDsxwgkcaA>

После открытия ссылки будет отображена следующая страница:



После этого пользователь может авторизоваться в системе, пройдя по указанной ссылке.

Работа в личном кабинете и назначение роли администратору

При первом входе администратора организации в систему он попадает в личный кабинет и видит следующее сообщение:

Наименование	Мои роли	Состояние	Статус
TestCompany		undefined	Не выбрано Администратор

Оно сигнализирует о том, что в данный момент идёт процесс создания базы данных новой организации. Он может продолжаться в течение нескольких минут, в это время не следует выполнять никаких действий с системой. После его окончания сообщение сменится на следующее:

демо режим (осталось дней: 29)

Организация по умолчанию TestCompany

Следующим шагом должно быть назначение роли текущему пользователю. Без выполнения данного действия пользователю не будут доступны никакие модули системы. Для назначения роли необходимо в таблице "Мои сотрудники" в колонке "Роли" выбрать из списка нужную роль и нажать Enter на клавиатуре:

Компании + Создать компанию

Наименование	Мои роли	Состояние	Статус
TestCompany		undefined	Не выбрано

Мои сотрудники + Создать нового сотрудника

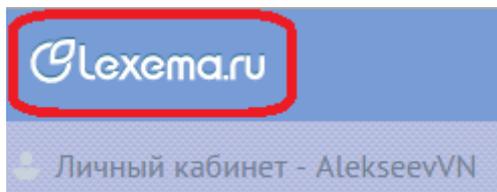
Drag a column and drop it here to group by that column

Логин	ФИО	Роли	Статус	Блокировка
AlekseevVN	Алексеев Владимир Николаевич	Официант Кадровик Заявки Supreg Бухгалтер МенеджерНастроекБухУчета Руководитель Кассир(бармен)	Не выбрано	<input type="checkbox"/>

Go to page: 1 Show rows: 20 1-1 of 1

Выбираемая роль зависит от задач, которые будут решаться с помощью системы и от того, какие модули будут использоваться.

После назначения роли текущий пользователь может перейти к главной странице системы. Для этого ему необходимо нажать на логотип lexema.ru:



Личные данные

После перехода на главную страницу пользователь увидит список групп доступных для него модулей и документов:

Мои Документы Развернуть

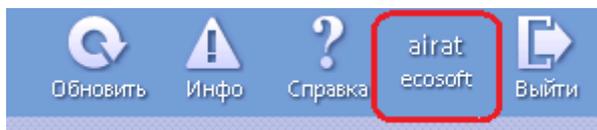
Введите название документа Раскрыть все Свернуть все

Продажи	SomeTest
!DemoGroup	YaTest
!FTestMenu5	Автотранспорт
!NewWay	Администратор прав доступа
!Домашняя Бухгалтерия (старая)	Банк
AErofeevTest	Башспирт
BulatTest	Бухгалтерский учет
CRM	Веб-моделер
CRM DX	Гостиница. Аналитика
ElzaTest	Документооборот. Администрирование
OLAP Анализ	Документооборот. Аналитические формы

Примечание: если после назначения роли и перехода на главную страницу экран пуст, необходимо очистить кеш браузера и перезагрузить страницу..

Создание учётных записей пользователей организации

Создавать учётные записи может пользователь системы с правами администратора организации. Для перехода к странице администрирования организации необходимо нажать на имя пользователя в правом верхнем углу:



На странице администрирования необходимо выбрать организацию - нажать на соответствующую строку в таблице "Компании". Для создания пользователя необходимо нажать на ссылку "Создать нового сотрудника":

demo.lexema.ru/Account/Manage.aspx

Lexema.ru

Личный кабинет - airat

Личные данные Смена пароля

Имя: Новый пароль:

Фамилия: Повторите пароль:

Отчество:

Организация по умолчанию:

Компании Создать компанию

Наименование	Мои роли	Состояние	Статус
ecosoft	Super, Docflow, Registrar, EArchiveUser, FrontOffice, TransactionalCenter, HomeBuhUser	Работает	Не выбрано

Go to page: 1 Show rows: 20 1-1 of 1

Мои сотрудники Снять блокировку Заблокировать Найти сотрудника **Создать нового сотрудника**

Drag a column and drop it here to group by that column

<input type="checkbox"/>	Логин	ФИО	Роли	Статус	Блокировка
<input type="checkbox"/>	YakinSV		Super, АдминистраторСистемы, CRM, Торговля, Бухгалтер склада, EArchiveAdmin	Не выбрано	<input type="checkbox"/>
<input type="checkbox"/>	Повар, Торговля, CRM, Бухгалтер склада, Кадровик, UGH, TESTTTTT, testRole,	Администратор	<input type="checkbox"/>

Необходимо заполнить все поля формы создания нового пользователя:

Создание нового сотрудника

Логин:

Имя:

Фамилия:

Отчество:

Почта:

Пароль:

← Задайте пароль, используя цифры, строчные и заглавные буквы

Пароль должен содержать заглавные и строчные буквы, а также цифры. Если какое-то поле заполнено неверно, то будет выведено соответствующее сообщение об ошибке. Подтверждать регистрацию пользователя по электронной почте не требуется.

Назначение роли пользователю организации

Для назначения роли пользователю организации необходимо выбрать организацию (нажать на соответствующую строку в таблице "Компании"), затем в таблице "Мои сотрудники" найти нужного сотрудника (можно воспользоваться фильтром в колонке Логин или ФИО) и выбрать нужную роль в списке в колонке Роль:

Наименование	Мои роли	Состояние	Статус
TestCompany	Super	Демо	Не выбрано

Go to page: 1 Show rows: 20 1-1 of 1

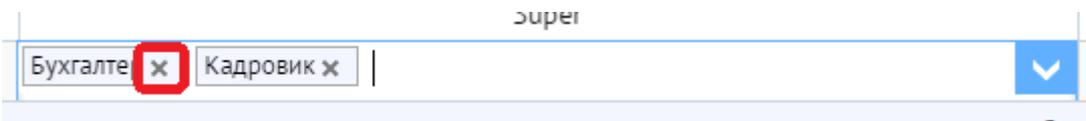
Мои сотрудники

Drag a column and drop it here to group by that column

Логин	ФИО	Статус	Блокировка
AlekseevVN	Алексеев Владимир Николаевич	Не выбрано	<input type="checkbox"/>
PetrovVA	Петров Виктор Алексеевич	Администратор	<input type="checkbox"/>

Go to page: 1 Show rows: 20 1-2 of 2

Одному пользователю может быть назначено множество ролей. Также назначение роли можно отменить, нажав дважды на ячейку с ролями и нажав на "крестик" роли:



Создание документа на основе модели

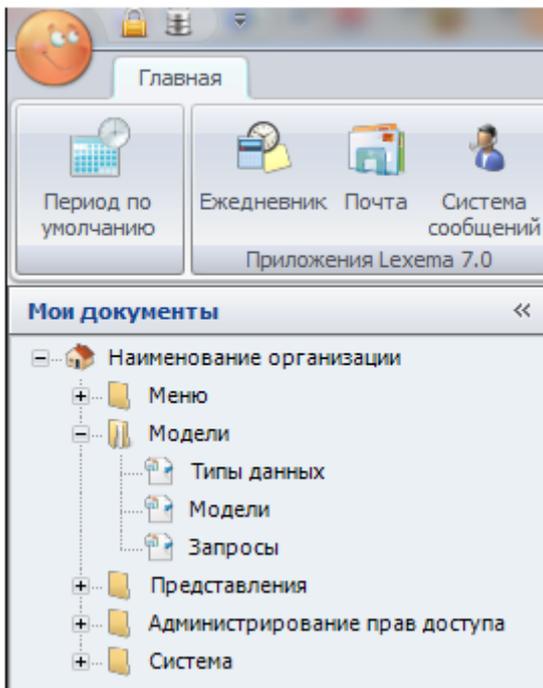
Документ является базовым элементом Lexema.ru, из операций с документами строятся все бизнес-процессы в системе. Создание документа, как и многие другие операции, осуществляется с помощью моделера приложения и требует авторизации. Понятие документа в системе включает модель и экранную форму (одну или несколько). В большинстве случаев документы создаются на основе модели, иногда на основе представления БД ("вьюшки") или запроса. Как правило, на основе модели создаются две экранных формы - реестр (список документов) и форма для редактирования одного документа.

Процесс создания документа на основе модели включает следующие действия:

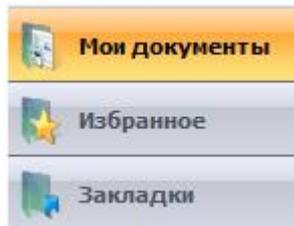
1. Создание модели и её полей
2. Обновление БД (миграция)
3. Создание экранной формы для редактирования документа и реестра документов
4. Предоставление прав на созданные объекты (модель, экранных формы) одной или нескольким ролям пользователей
5. Создание ссылки на реестр в меню приложения

Создание модели и её полей

Для работы с моделями в дереве навигации выберите раздел «Модели» и пункт с тем же именем:



Примечание: если дерево навигации отсутствует на экране: оно было скрыто, для его отображения нажмите кнопку  или выберите раздел «Мои



документы».

Нажмите кнопку «Создать документ» на панели инструментов на вкладке «Документ» или кнопку «Insert» на клавиатуре. Введите имя модели (документа), соблюдая правила именования объектов. Имя таблицы и PrimaryKeyDomain будут автоматически заполнены заданным именем модели, при необходимости их можно скорректировать. В списке «Пространство имен» необходимо выбрать пункт, соответствующий проекту, в рамках которого создаётся модель. Если вы затрудняетесь с выбором, необходимо указать пункт «1 | Base» (базовое пространство имён).

Создание полей модели

По умолчанию при создании новой модели генерируются следующие служебные поля:

- VCode - уникальный идентификатор (суррогатный первичный ключ) модели/таблицы;
- CDate - дата создания записи;
- CHost - имя хоста (узла сети), с которого была создана запись;

- CUser - имя учётной записи (логин) пользователя, который создал запись;
- WDate - дата последнего изменения записи;
- WHost - имя хоста, с которого было произведено последнее изменение записи;
- WUser - имя учётной записи пользователя, который произвёл последнее изменение записи;
- COrg - код организации холдинга, в рамках которой была создана запись;
- WOrg - код организации холдинга, в рамках которой было произведено последнее изменение записи;

Параметры служебных полей доступны только для чтения, изменять их нельзя. Также не рекомендуется их удалять.

Для каждого поля модели необходимо ввести имя в колонке Name, соблюдая правила именования. Имя поля таблицы БД в колонке DbName будет автоматически заполнено таким же именем. При необходимости DbName можно изменить. Колонка DbName может быть не заполнена для вычисляемых ("вьюшечных") полей. Если с моделью связано представление БД (вьюшка), то в колонке ViewName можно задать имя поля вьюшки.

Каждому полю необходимо задать тип данных (группа колонок Type). Типы разделяются на базовые и справочники. От выбранного типа поля модели зависит вид элемента управления в экранной форме.

Базовые типы:

- int - целочисленное значение
- bigint - длинное целое, применяется для хранения ключевых полей (идентификаторов, кодов) моделей
- string - короткая строка (до 255 символов), соответствующий тип MS SQL - varchar(255)
- longstring - длинная строка, соответствующий тип MS SQL - varchar(max)
- money - тип данных для хранения значений денежных сумм
- double - вещественный тип данных
- bool - значение логического типа. В экранных формах элементом управления, соответствующим полю типа bool, является флажок (CheckBox)
- Date - дата
- DateTimeOffset - дата и время с учетом часового пояса
- File - ссылка на файл. В экранных формах элементом управления, соответствующим полю типа File, является интерфейс для загрузки, просмотра и скачивания файлов.

Помимо базовых типов, поле может являться ссылкой (внешним ключом) на запись в другой таблице (справочнике). Для создания ссылки необходимо выбрать в списке имя типа данных, соответствующего модели. В БД такое поле имеет тип bigint. В экранной форме для данного поля будет создан выпадающий список (Lookup), содержащий значения из связанной модели.

Рекомендуется также ввести краткие описания полей в столбце Description, это является хорошим стилем разработки и упрощает процесс командной работы.

Примечание: наименование модели и её полей должны быть англоязычными. Перевод англоязычных терминов, использованных в наименовании, следует внести в Словарь типовых терминов.

Свойства		Дочерние таблицы		События		Type		Value changed script		Редактиров	Удалить скрыт	GlobalId
Name	DbName	ViewName	Description	Код	Наименование	Код	Наименован	Код	Наименован			
VCode	VCode	VCode		127	bigint							5969161E-7F1C...
CDate	CDate	CDate		461	DateTimeOffset							29669687-165F...
CHost	CHost	CHost		73	string							011CD7C7-43FB...
CUser	CUser	CUser		73	string							F078FB6A-18E1...
WDate	WDate	WDate		461	DateTimeOffset							7953C251-D66E...
WHost	WHost	WHost		73	string							7072B46C-S112...
WUser	WUser	WUser		73	string							18CE9DC8-49BA...
COrg	COrg	COrg		1	int							EE2CD880-5F59...
WOrg	WOrg	WOrg		1	int							47A93AC7-5BBA...
Name	Name		наименование	73	string							B895D2E2-6327...
Description	Description		описание	2	longstring							D4C86C80-74F0...
Cost	Cost		стоимость	3	money							4132B00E-26EF...
Weight	Weight		вес	34	double							29C5872-0E49...
ProduceDate	ProduceDate		дата производства	466	Date							00EF8412-56E6...
IsActual	IsActual		признак актуальности	316	bool							28C0CC67-3800...

Сохраните модель, нажав на кнопку «Сохранить» или «Сохранить и закрыть».

Обновление БД (миграция)

После создания модели необходимо произвести обновление БД, в результате которого будут сгенерированы и выполнены SQL-скрипты для создания и/или изменения объектов БД (таблиц, первичных и внешних ключей, ограничений, триггеров. Этот процесс требуется для синхронизации описания моделей бизнес-логики приложения и структуры таблиц физической БД.

Для осуществления данной операции в дереве навигации откройте раздел «Система» и выберите пункт «Обновление БД». Необходимо корректно сформировать строку подключения к серверу БД, которая включает следующие параметры:

- Server - IP-адрес или символьное имя сервера БД, на котором расположена база с пользовательскими данными
- Database - имя БД пользовательских данных
- Id - логин учётной записи СУБД MS SQL Server, обладающей правами на выполнение соответствующих запросов
- Password - пароль указанной учётной записи

Пример строки подключения: Server=192.168.0.2\sql;Database=L8_TestData;UserId=sa;Password=TopSecret;

После ввода строки подключения необходимо нажать кнопку "Обновить содержание". В результате в таблице на экране будет отображён список объектов БД, подлежащих обновлению.

По умолчанию выводятся объекты всех пользователей системы, однако существует возможность отобразить только свои изменения. Для удобства работы список может быть отфильтрован по имени хоста (узла сети), на котором было произведено создание или последнее изменение объекта. Для фильтрации необходимо заполнить следующие поля перед нажатием на кнопку "Обновить содержание":

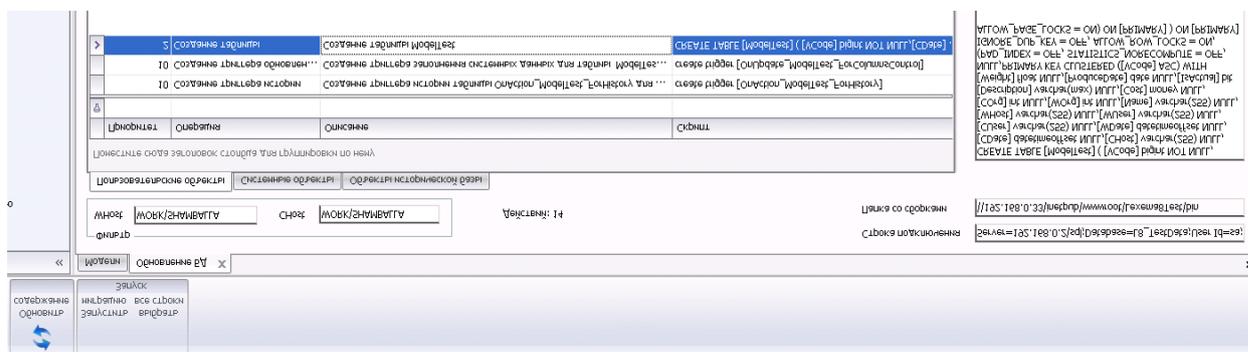
- SHost - имя хоста, на котором было произведено создание объектов
- WHost - имя хоста, на котором было произведено последнее изменение объектов

Интерфейс обновления БД содержит три вкладки:

- Пользовательские объекты
- Системные объекты
- Объекты исторической базы

Список объектов, подлежащих обновлению, может быть скорректирован - из него могут быть удалены записи.

Для удобства работы список может быть сгруппирован по одному или нескольким полям. Для группировки по колонке необходимо перетащить её заголовок в соответствующую область в шапке таблицы.

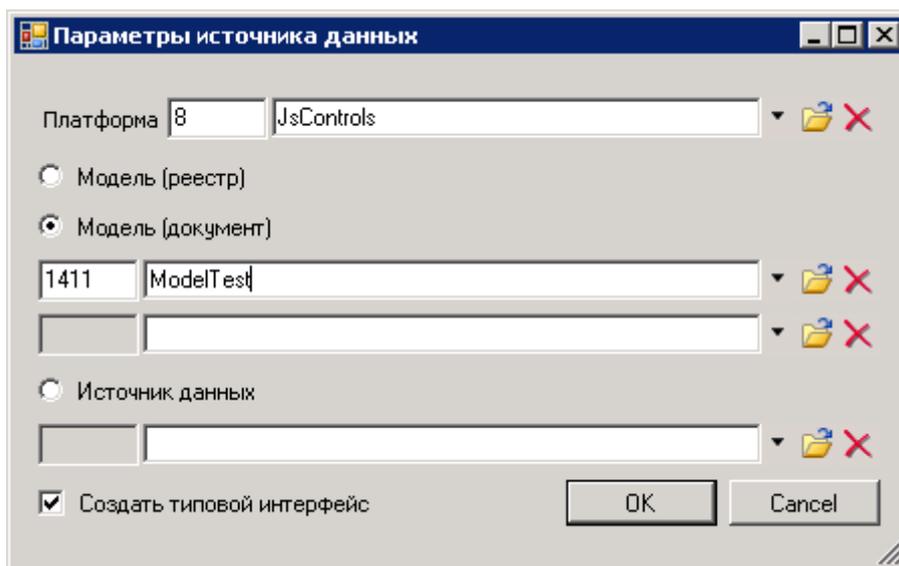


Для завершения операции обновления БД необходимо нажать на кнопку "Запустить миграцию" на панели инструментов. В результате будут выполнены все запросы, показанные в списке в столбце "Скрипт".

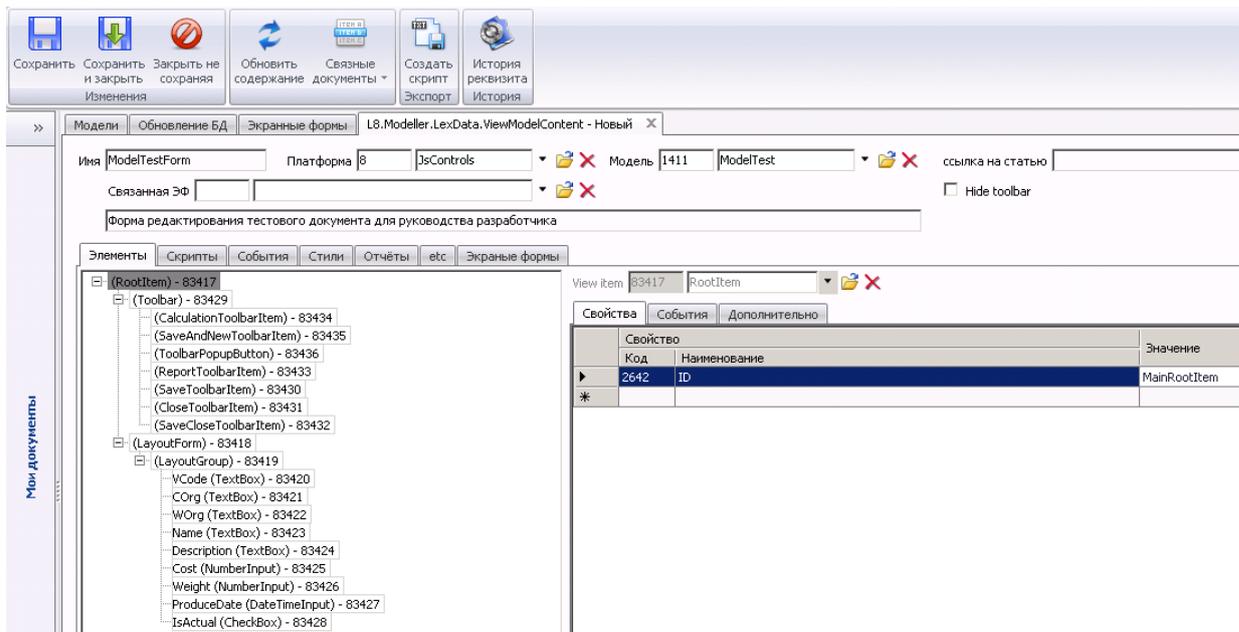
Создание экранных форм

Экранные формы необходимы пользователям системы для работы с документами. Как правило, для большинства документов требуется две ЭФ - форма для редактирования и реестр. Эти две ЭФ для одного документа взаимосвязаны между собой, для упрощения создания связей вначале рекомендуется создать форму для редактирования, а затем - реестр.

Для создания экранной формы для редактирования документа в дереве навигации откройте раздел «Представления» и выберите пункт «Экранные формы». Нажмите кнопку «Создать документ» на панели инструментов на вкладке «Документ» или кнопку «Insert» на клавиатуре. В диалоговом окне выберите из списка "Платформа" пункт "JsControls", укажите пункт «Модель (документ)», выберите из списка имя созданной ранее модели. Оставьте отмеченной флажок «Создать типовой интерфейс».



Имя экранной формы будет сгенерировано автоматически на основе имени модели путём добавления Form в конце (например, для модели TestModel имя ЭФ - TestModelForm). Введите описание созданной формы в соответствующее поле.

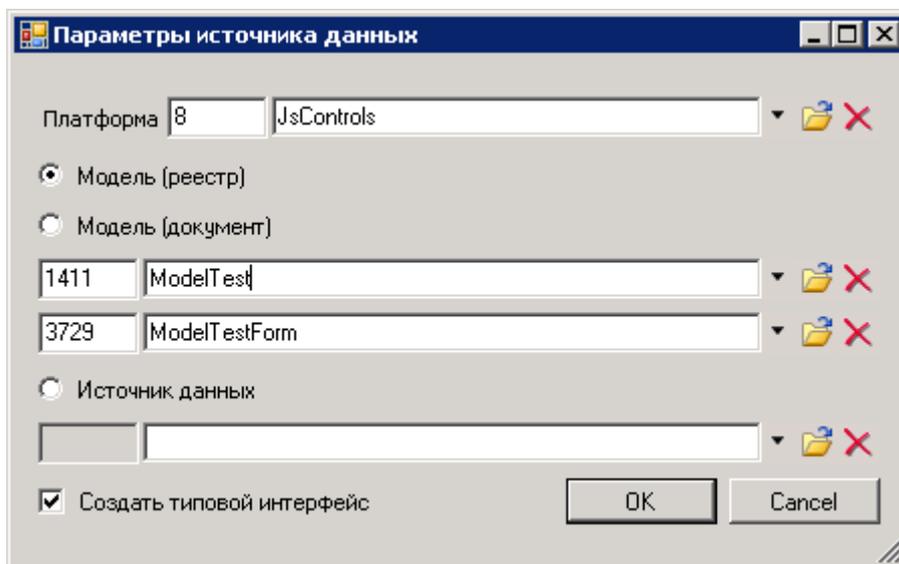


Сохраните созданное представление с помощью кнопки «Сохранить» на панели инструментов «Документ». Разверните дерево элементов экранной формы и ознакомьтесь с его структурой. Подробнее эта структура и принципы работы с элементами описаны в статье «Работа с элементами экранной формы». Закройте экранную форму, нажав на кнопку «Сохранить и закрыть».

Создание реестра и его связь с формой редактирования

После создания и сохранения формы для редактирования документа, необходимо создать экранную форму реестра. Для этого необходимо в окне "Экранные формы" нажать кнопку "Создать документ" и заполнить следующие поля:

- выбрать тип - "Модель (реестр)"
- выбрать название модели
- выбрать название экранной формы для редактирования, созданной ранее



Параметры источника данных

Платформа 8 JsControls

Модель (реестр)

Модель (документ)

1411 ModelTest

3729 ModelTestForm

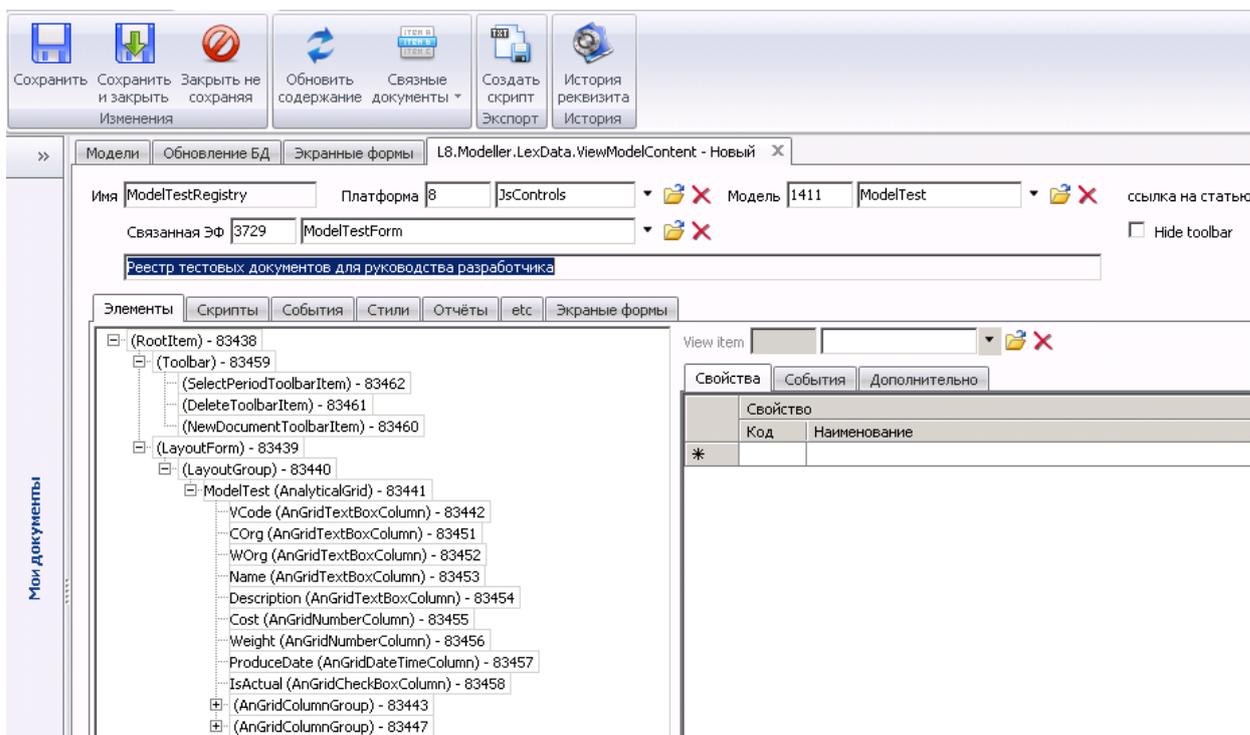
Источник данных

Создать типовой интерфейс

OK Cancel

Имя созданного реестра будет сгенерировано автоматически на основе имени модели путём добавления Registry в конце (например, для модели TestModel имя ЭФ - TestModelRegistry). Введите описание созданной формы в соответствующее поле.

Важно! Для создания корректной связи между формой для редактирования и реестром необходимо заполнить поле "Связанная ЭФ", выбрав из списка соответствующее значение.



Сохранить Сохранить и закрыть Закрыть не сохраняя Обновить содержание Связанные документы Создать скрипт Экспорт История реквизита История

Модели Обновление БД Экранные формы L8.Modeller.LexData.ViewModelContent - Новый

Имя ModelTestRegistry Платформа 8 JsControls Модель 1411 ModelTest ссылка на статью

Связанная ЭФ 3729 ModelTestForm Hide toolbar

Реестр тестовых документов для руководства разработчика

Элементы Скрипты События Стили Отчёты etc Экранные формы

(RootItem) - 83438

- (Toolbar) - 83459
 - (SelectPeriodToolBarItem) - 83462
 - (DeleteToolBarItem) - 83461
 - (NewDocumentToolBarItem) - 83460
- (LayoutForm) - 83439
 - (LayoutGroup) - 83440
 - ModelTest (AnalyticalGrid) - 83441
 - VCode (AnGridTextBoxColumn) - 83442
 - COrg (AnGridTextBoxColumn) - 83451
 - WOrg (AnGridTextBoxColumn) - 83452
 - Name (AnGridTextBoxColumn) - 83453
 - Description (AnGridTextBoxColumn) - 83454
 - Cost (AnGridNumberColumn) - 83455
 - Weight (AnGridNumberColumn) - 83456
 - ProduceDate (AnGridDateTimeColumn) - 83457
 - IsActual (AnGridCheckBoxColumn) - 83458
 - (AnGridColumnGroup) - 83443
 - (AnGridColumnGroup) - 83447

View item

Свойства События Дополнительно

Свойство	Код	Наименование
*		

Форму для редактирования также необходимо привязать к реестру, для этого её нужно открыть из списка экранных форм и заполнить поле "Связанная ЭФ", выбрав из списка имя созданного реестра:

Модели | Обновление БД | Экранные формы | L8.Modeller.LexData.ViewModelContent (3729) X

Имя Платформа JsControls Модель

Связанная ЭФ

Предоставление прав на созданные объекты

Для предоставления пользователям возможности работать с документом необходимо настроить параметры доступа ко всем используемым объектам - экранным формам, моделям и запросам. В системе Lexema.ru разделение прав доступа к объектам осуществляется на основе ролей пользователей. Для работы с этим механизмом необходимо в дереве навигации открыть раздел «Администрирование прав доступа», пункт «Роли». В списке ролей выбрать необходимую роль (например, роль Super). Откройте роль на редактирование с помощью кнопки «Открыть документ на изменение» или двойным щелчком мышки на строке.

Модели | Обновление БД | Экранные формы | Роли X

	VCode	RoleName	Description	GlobalId
▶	1	Повар		6D5E9E76-4F2D-409D-95E4-089E47B29A47
	2	АдминистраторСистемы		F8FD7D9E-9831-4D9C-B784-743BBFB055E
	3	Торговля		47192C0A-3BC9-471A-80A1-2FB12CECCF32
	4	CRM		5D425929-BF41-4BB6-84D1-1F47C25F200C
	5	Бухгалтер склада		44502553-43C4-4FA0-8762-6D334A7E12E9
	6	Официант		2869B636-541B-48F3-8850-0D892C3CC4CD
	7	Кадровик		CC758E84-EE1B-4D2B-8DF5-D923707151E3
	8	UGH		9372601A-A9BA-4F6D-9C8E-C12E8504FD35
	10	Administrators		9EA19F07-7008-4398-A728-C3FF5806E39C
	40	TESTTTTT		775D2F49-3BC4-4940-AAF7-F505BB29AE64
	50	testRole		97EBB771-DCF0-4FA7-BA58-4DE949D33515
	60	NewTestRole		5A05590C-ABEE-46B1-B224-188DBF88D032
	73	Заявки		83FE274D-8C5E-40EB-8D46-80BECFA91295
	74	Super		20BFE948-8E2B-46A9-A197-E9D4E1B3FA94
	75	Бухгалтер		78CEA60F-48CB-46F4-9E3F-74FB66AA1B24

Интерфейс администрирования прав доступа состоит из четырёх вкладок:

- Элементы управления
- Модели
- Запросы
- Вложенные роли

Предоставление прав доступа к экранным формам

На вкладке "Элементы управления" в списке документов найдите названия созданных представлений (новые документы, как правило, отображаются в конце списка). Выделите строку с названием формы реестра (например, TestModelRegistry).

Модели | Запросы | Экранные формы | Роли | L8.AccessUI.Роли_Форма (74) X

Роль

Имя

Элементы управления | Модели | Запросы | Вложенные роли

Фильтр документов
 Без прав
 С правами
 Показать все
 [Добавить права всем документам](#)

	VCode	Name	Right	Удалить права
	3678	OctTest	<input checked="" type="checkbox"/>	
	3679	SaleOnCredit	<input checked="" type="checkbox"/>	
	3680	PivotTemplateForm	<input checked="" type="checkbox"/>	
	3683	NotActiveCustomerForm	<input checked="" type="checkbox"/>	
	3688	WifesAnalysis	<input type="checkbox"/>	
	3708	MP_TestRegistry	<input checked="" type="checkbox"/>	
	3709	MP_TestForm	<input checked="" type="checkbox"/>	
	3713	TestIncomingAktForm	<input checked="" type="checkbox"/>	
	3715	TestIncomingAktRegistry	<input checked="" type="checkbox"/>	
	3716	TestIncomingAktForm2	<input checked="" type="checkbox"/>	
	3717	TestIncomingAktRegistry2	<input checked="" type="checkbox"/>	
	3718	HairColorForm	<input type="checkbox"/>	
	3719	HairColorRegistry	<input type="checkbox"/>	
	3720	WifeForm	<input type="checkbox"/>	
	3721	WifeRegistry	<input type="checkbox"/>	
	3722	TaxItemForm	<input checked="" type="checkbox"/>	
	3724	TaxItemRegistry	<input checked="" type="checkbox"/>	
	3725	RiskRegistry	<input checked="" type="checkbox"/>	
	3727	RiskForm	<input checked="" type="checkbox"/>	
	3728	RiskInternalManagementForm	<input checked="" type="checkbox"/>	
▶	3729	ModelTestForm	<input checked="" type="checkbox"/>	
	3730	ModelTestRegistry	<input checked="" type="checkbox"/>	

В дереве элементов справа можно задать права на доступ к элементам экранной формы. Для предоставления полного доступа ко всей форме нажмите правой кнопкой мышки на элемент RootItem и выберите в контекстном меню пункт «2 – Активность».

	Элемент		Право
	Код	Наименование	
▶	3571		2
	3572		2
	3573		2

The screenshot shows a tree view of a form structure. The root node is "(RootItem)". Under it, there are two main sections: "(Layout)" and "(Toolbar)". The "(Layout)" section contains several controls: "Name (TextBox) - 83423", "Description (TextBox) - 83424", "Cost (NumberInput) - 83425", "Weight (NumberInput) - 83426", "ProduceDate (DateTimeInput) - 83427", and "IsActual (CheckBox) - 83428". The "(Toolbar)" section contains several toolbar items: "(SaveToolbarItem) - 83430", "(CloseToolbarItem) - 83431", "(SaveCloseToolbarItem) - 83432", "(ReportToolbarItem) - 83433", "(CalculationToolbarItem) - 83434", "(SaveAndNewToolbarItem) - 83435", and "(ToolbarPopupButton) - 83436". A context menu is open over the selected element, showing options: "0 - Нет", "1 - Видимость", "2 - Активность", and "Удалить права". The "2 - Активность" option is highlighted.

Повторите действия по предоставлению полного доступа для второй экранной формы документа.

Предоставление прав доступа к модели

На вкладке "Модели" необходимо перейти в конец списка, в колонке "Model - Наименование" ввести название созданной ранее модели. Поля VCode, Namespace и GlobalId заполнятся автоматически. Для предоставления текущей роли прав на создание, чтение, изменение и удаление записей модели необходимо соответствующим образом установить или снять галочки в столбцах Create, Read, Update и Delete. Если модель отсутствует в списке, по умолчанию роль не обладает никакими правами доступа к записям модели.

VCode	Namespace Код	Наименование	Model Код	Наименование	Create	Read	Update	Delete	GlobalId
844	1	Base	1404	RiskInternal_FactorExternalParamReputationAll	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	32F94B0B-0F70-47AA-91F0-539465635076
845	1	Base	1405	RiskInternal_FactorWaitingLossesNeedExternal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	469E2AC7-3CE2-4091-93B9-30F6D3204C65
846	1	Base	1406	RiskInternal_FactorWaitingLossesAmount	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CDB6978B-6794-4660-9D91-26B59027D5D1
847	13	DocflowDocuments	1407	ApprovalSheetSubdivision	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	813CAFDD-F711-4812-BEB1-BEB84EFB2D73
848	1	Base	1408	RiskInternal_FactorExternalParamControlSign	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	D789232D-1DC5-483E-834D-C5587D7F1FAC
849	1	Base	1410	RiskInternal_FactorEstimationResult	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2C57746D-5EE3-45E6-870D-20779F4CCE42
850	3	test	1411	ModelTest	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	D1F1D049-1E45-43AB-822B-05DE7592AB38
851	3	test	1412	ModelTestClassifier	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	621D7E3D-23F6-4C57-817D-4F39847B4525
852	3	test	1413	ModelTestChild	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	48D0EF15-2D58-435B-A821-4D62308A807D
853	1	Base	1414	RiskInternal_FactorFinLosses4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	85A54AE8-C7D0-4E4A-98D4-82C627EC745F
921	1	Base	1422	Risk_EstimationMethod	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6F64E391-C9BC-43C7-98A6-A64843EDF0A0
925	1	Base	1423	Risk_Status	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CB1F059E-65DB-4B54-9330-B43990BB6528
935	3	test	1416	TestSource	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	F1107F8D-F8FD-4DD9-ABC5-9E2E02E3BD74
936	3	test	1417	TestTarget	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	7D37D3B9-4BD7-44A4-A6D9-75D2EE374053
939	13	DocflowDocuments	1426	ServiceRecordForAnnulmentOfAdditionalDocs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	8D5C39CE-4A99-4121-B9B2-8F4BD25435C7
940	13	DocflowDocuments	1427	ServiceRecordForAnnulmentOfAdditionalDocsReg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6D0A9EFB-B54A-4272-8524-19AA7E35784C
944	13	DocflowDocuments	1425	AdditionalDocumentSpr	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	55F779B0-0382-433D-8921-8A1B8C1543BB
1027	1	Base	1429	ProfessionCustomer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	BF988C15-A7C8-4415-8EA8-B5A523DBE5F4
1173	1	Base	1430	TypeStoreOperations	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2CA080BF-D141-4439-B2FC-11E606CBFCFA
1255	3	test	1433	RequestTest	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	00123AE0-73C2-4D2C-9CD3-CAE2604115F2
1256	3	test	1434	RequestTestDetail	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	42BC19C6-B2D0-476D-9096-31C4D178E592
1260	3	test	1437	AddressTest	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6C6F21A0-B74D-4FB6-A9D9-E6F89E1E8AA3
1261	1	Base	1438	SettingAccountingAnalytics	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EC44E922-8E5F-4E64-A51A-D442D027B1E1
1262	1	Base	1439	ProcurementPlan	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5347170D-E4C8-48C6-AE1F-7825125CE14D
1263	1	Base	1440	PlanMaterialProcurement	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	76FE8B5D-AFB0-45C4-BB8F-EC4F20860489
1264	1	Base	1442	PlanMaterialProcurementDetail	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FF107AF5-D8E7-495E-AAF7-F0A5C84CECEB
1266	1	Base	1446	PlanMaterialProcurementDetail1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	7DB13EF5-29BD-4F3A-9441-B835F33BD10C
1299	3	test	1411	ModelTest	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	A8B0B5A4-69B8-4941-863B-98E22E6707D9

Права на доступ роли к запросам предоставляется аналогичным образом.

После добавления прав доступа сохраните изменения и закройте документ роли, нажав на кнопку "Сохранить" или "Сохранить и закрыть".

Создание пункта меню для открытия реестра документов

Меню в системе Lexema.ru имеет древовидную иерархическую структуру: пункты меню распределены по категориям. Для работы с созданным документом необходимо создать пункт (элемент) меню в существующей категории, либо создать для него новую категорию меню.

Создание новой группы (категории) меню

Для создания новой группы меню необходимо в дереве навигации в категории "Меню" выбрать пункт "Группы". Далее нажать кнопку "Создать документ". В открывшемся окне необходимо ввести название группы - оно будет отображаться в главном меню системы. Затем необходимо сохранить новую группу, нажав кнопку "Сохранить".

VCode	Name	View Код	Наименование
*			

Добавление пункта (элемента) меню

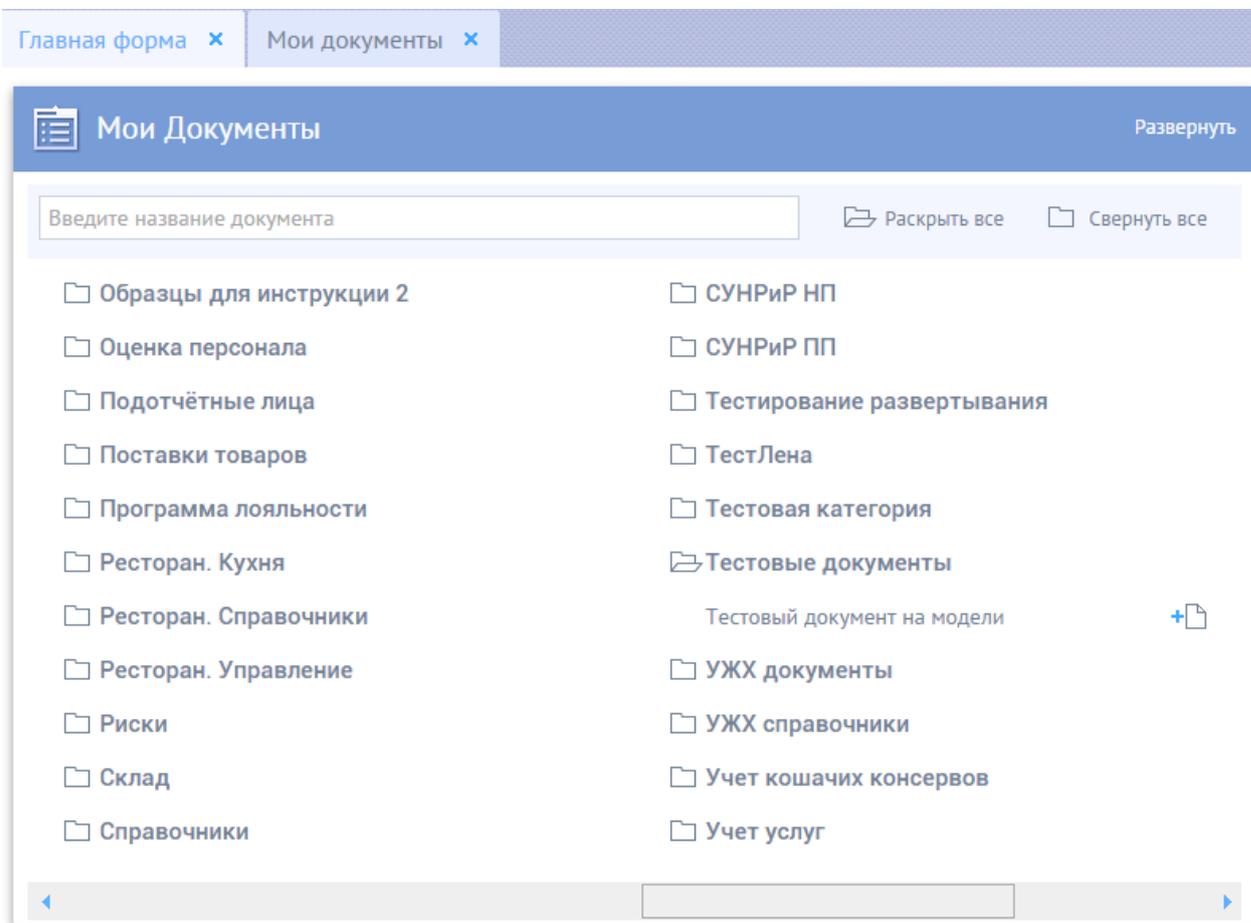
Для добавления пункта меню необходимо в табличной части новой или существующей группы в колонке Name указать название пункта меню, а в поле View выбрать созданный ранее реестр.

VCode	Name	View Код	Наименование	Type Код	Наименование
1572	Тестовый документ на модели	3730	ModelTestRegistry	5	Документ

Для того, чтобы сделанные изменения вступили в силу, необходимо нажать кнопку "Сохранить", либо "Сохранить и закрыть"

Проверка работоспособности созданного документа

Для проверки работоспособности созданного документа необходимо открыть в браузере страницу системы и авторизоваться (ввести логин и пароль). В главном меню в области "Мои документы" следует раскрыть соответствующую категорию и нажать на пункт меню.



В открывшемся реестре нажмите кнопку "Создать" - в результате будет отображена форма редактирования документа. Введите данные в поля формы, нажмите кнопку "Сохранить и закрыть" на панели инструментов.

Lexema.ru

Сохранить Закрыть Сохр. Закрыть Печать Проводки Сохр. Создать

от 01.11.2015 - 30.11.2015 Главная форма Мои до

VCode: 1

COrg:

WOrg:

Name: Муфта соединительная

Description: Применяется для соединения 1

Cost: 1 200.00

Weight: 5.00

ProduceDate: 12.11.2015

IsActual:

Следует отметить, что по умолчанию система генерирует англоязычные надписи для элементов управления в форме редактирования и реестре, а также отображает некоторые системные поля. В реальных документах следует переименовывать надписи и удалять лишние элементы управления для служебных полей (код и т.п.).

В результате сохранения документа в реестре отобразится созданная запись.

Lexema.ru

Создать Удалить Период

от 01.11.2015 - 30.11.2015 Главная форма Мои документы

<input checked="" type="checkbox"/>	VCode	COrg	WOrg	Name	Description	Cost	Weight	ProduceDate	IsActual	Создание		
										Дата	Хост	Автор
<input checked="" type="checkbox"/>	1	0	0	Муфта соеди...	Применяет...	1 200.00	5.00	12.11.2015	<input checked="" type="checkbox"/>	12.11.2015	10.8.56.4	airat

Если все вышеперечисленные действия выполнены успешно, значит документ был создан верно.

Настройка параметров отображения формы и её элементов

Параметры отображения формы и её элементов управления, установленные по умолчанию при генерации, как правило, не являются оптимальными. Для более корректного отображения форм рекомендуется:

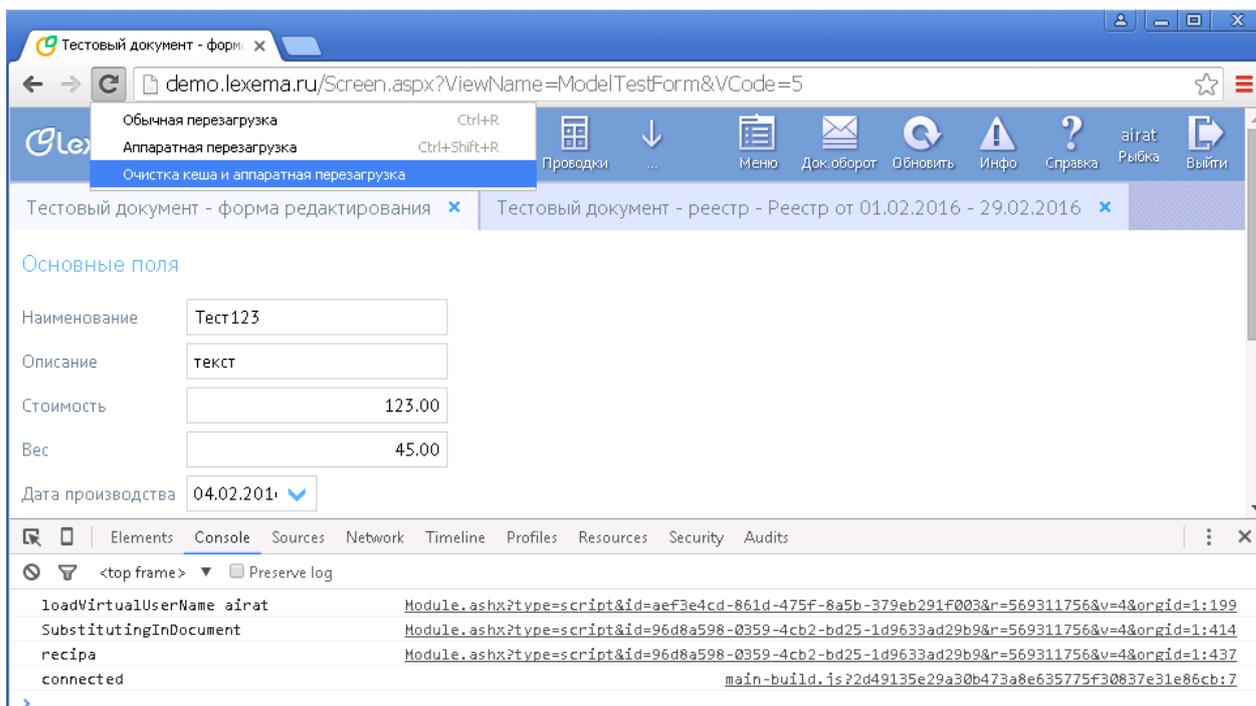
- в реестре:
 - проверить задана ли связанная ЭФ. При необходимости выбрать в этом списке форму для редактирования данного документа
 - задать заголовок формы (свойство title элемента RootItem)
 - переименовать заголовок таблицы AnaliticalGrid (свойство caption) - ввести название на русском языке
 - переименовать заголовки полей таблицы AnaliticalGrid - ввести значения на русском языке
 - удалить ненужные поля из таблицы реестра

- поменять порядок полей
- изменить ширину колонок
- в экранной форме для редактирования:
 - проверить задана ли связанная ЭФ. При необходимости выбрать в этом списке реестр данного документа
 - задать заголовок формы (свойство title элемента RootItem)
 - удалить поле VCode (его редактирование вручную не предполагается)
 - переименовать заголовки полей формы
 - изменить ширину полей в зависимости от размеров содержимого
 - для полей типа longstring удалить текстовое поле (TextBox) и создать многострочное текстовое поле типа Memo

Обновление страницы после изменения экранной формы

Для повышения производительности работы системы многие компоненты (скрипты и др.) кешируются. Это приводит к необходимости обновлять страницу с очисткой кеша при каждом изменении экранной формы. Для этого в Google Chrome необходимо:

1. нажать F12 или Ctrl+Shift+J для открытия консоли разработчика
2. на панели инструментов нажать и длительно удерживать кнопку "Обновить" с круглой стрелкой
3. в выпавшем меню выбрать пункт "Очистка кеша и аппаратная перезагрузка"

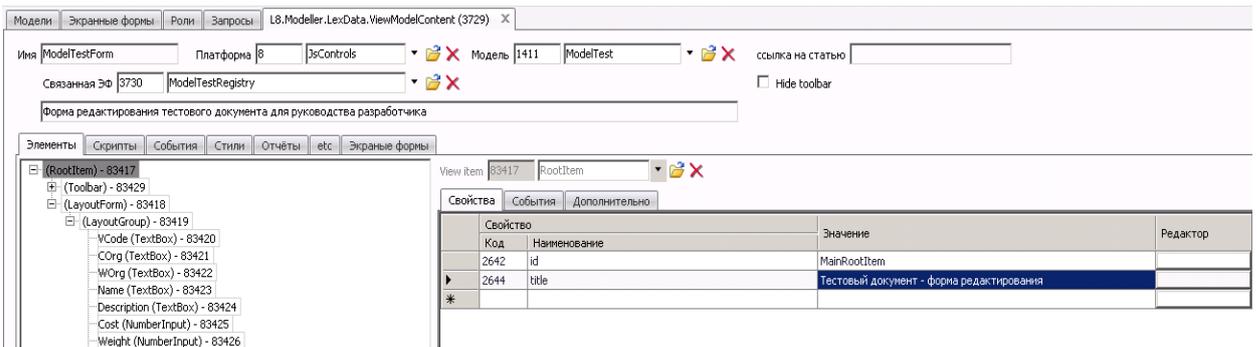


При обновлении страницы обычным способом (F5 или Ctrl+F5) изменения, сделанные в экранной форме с помощью моделиера приложения, не будут отображены.

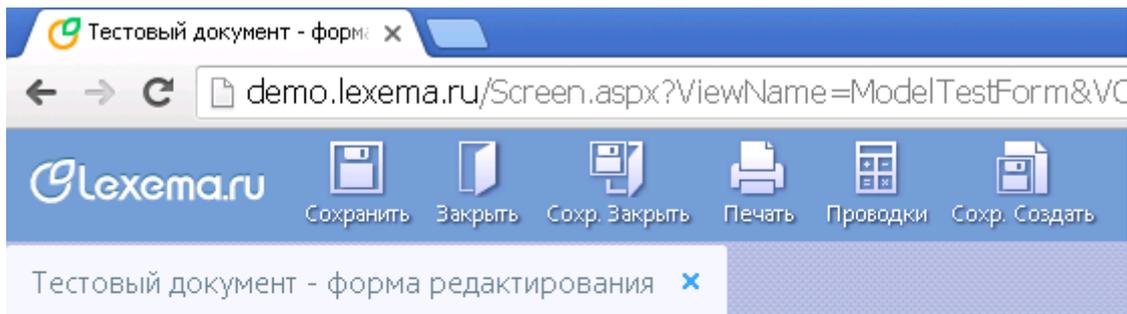
Примечание: если изменились только данные, достаточно нажать кнопку "Обновить" на панели инструментов системы Lexema.ru (справа вверху).

Указание заголовка формы

В моделиере приложения откройте экранную форму (категория "Представления", пункт "Экранные формы"). В дереве элементов формы выберите RootItem. В окне свойств добавьте свойство title, в поле "Значение" введите текст заголовка формы:

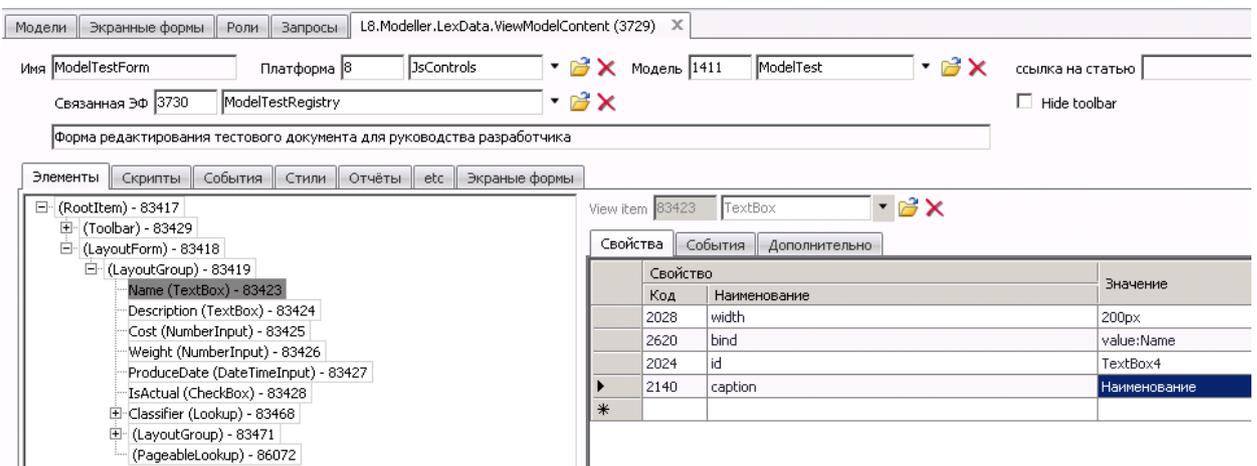


Этот текст будет отображаться в названии вкладки браузера и в названии вкладки приложения:

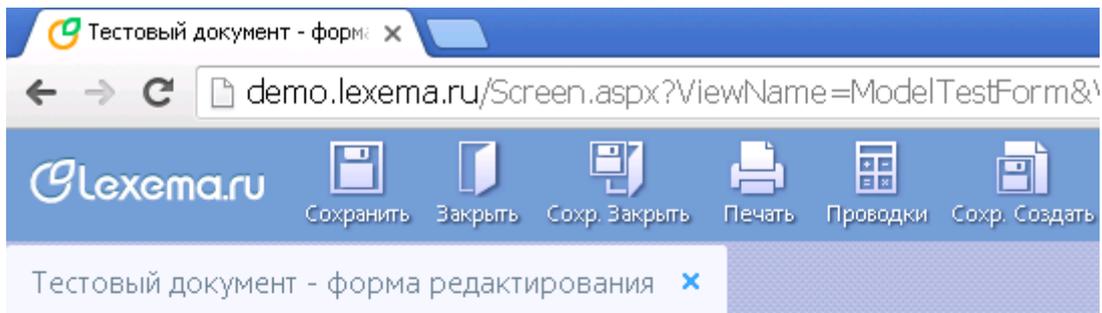


Переименование полей экранной формы

В соответствии с правилами именования объектов имена полей моделей называются на английском языке. После автоматической генерации интерфейса по умолчанию подписи элементов управления формы и заголовки колонок реестра совпадают с именами полей модели. Т.к. англоязычные подписи на форме, как правило, неприемлемы, необходимо их изменить на русскоязычные. Для этого в моделиере приложения откройте экранную форму (категория "Представления", пункт "Экранные формы"). В дереве элементов формы выберите поле, подпись которого нужно поменять. В окне свойств в строке caption измените значение на русскоязычное:



Проделайте аналогичные действия для всех полей. В результате на форме в браузере подписи полей станут русскоязычными:



Основные поля

Наименование	<input type="text" value="Тест123"/>
Описание	<input type="text"/>
Стоимость	<input type="text"/>
Вес	<input type="text"/>
Дата производства	<input type="text" value="04.02.201"/> ▾
Актуальность	<input type="checkbox"/>
Справочник	<input type="text"/> ▾

Элементы экранных форм размещаются в группах LayoutGroup. У группы также может быть добавлено и задано свойство caption.

Переименование полей реестра

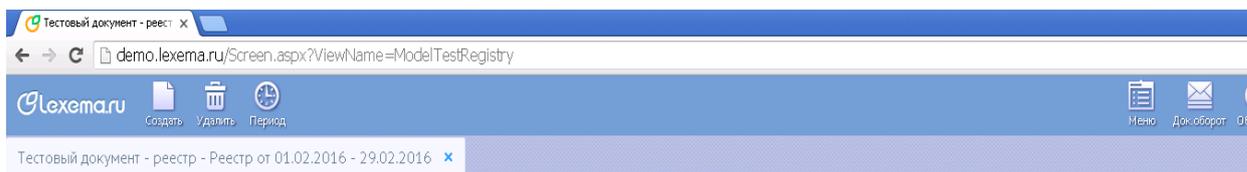
В целом данный процесс аналогичен переименованию полей формы, однако, свойство-заголовок колонки таблицы реестра (AnalyticalGrid) называется Text:

The screenshot shows the development environment with a tree view on the left and a properties window on the right. The tree view shows a hierarchy of elements, including a toolbar, layout form, and layout group containing an analytical grid. The analytical grid has several columns, with the 'Name' column selected. The properties window for the selected 'Name' column shows various properties and their values.

Свойство	Значение
Код	3586
controltype	columns
datafield	Name
width	100px
text	Наименование
align	center
cellsalig	left
columntype	textbox
id	AnGridTextBoxColumn8

Подписи служебных полей "Дата создания", "Дата изменения" и т.п. по умолчанию называются по-русски при генерации, в их изменении нет необходимости.

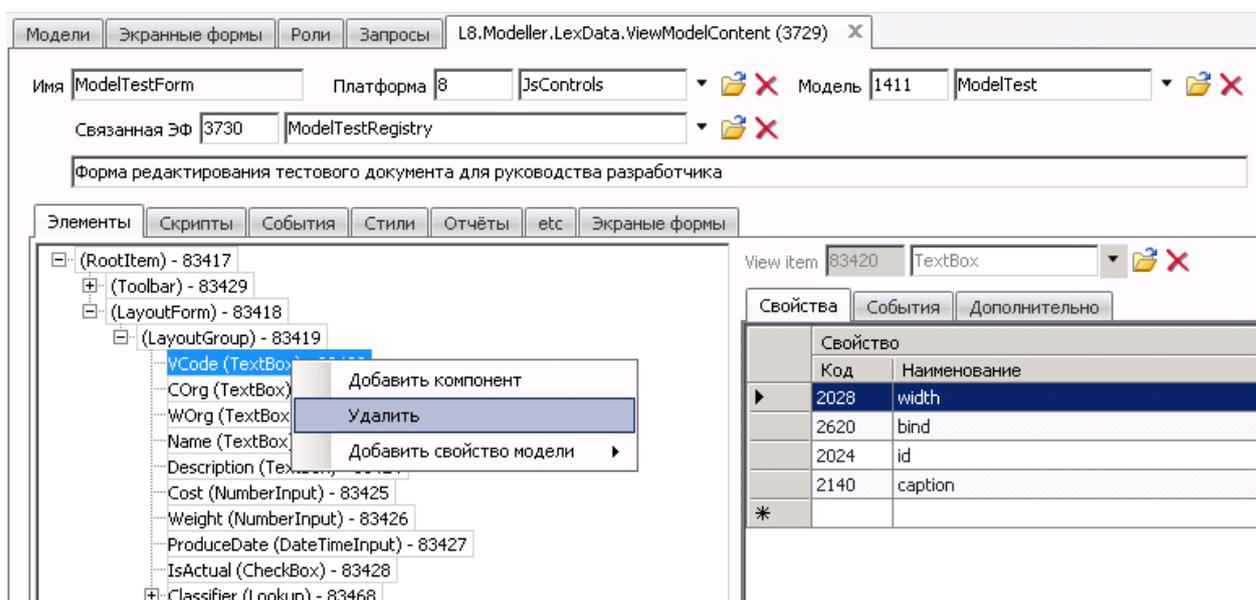
В результате имена колонок таблицы реестра отображаются на русском языке:



Удаление полей формы и реестра

При автоматической генерации элементов управления формы на основе модели по умолчанию создаются элементы для редактирования системных полей - VCode, COrg, WOrg и т.п. Наличие этих полей в форме редактирования нецелесообразно, т.к. их значения заполняются автоматически.

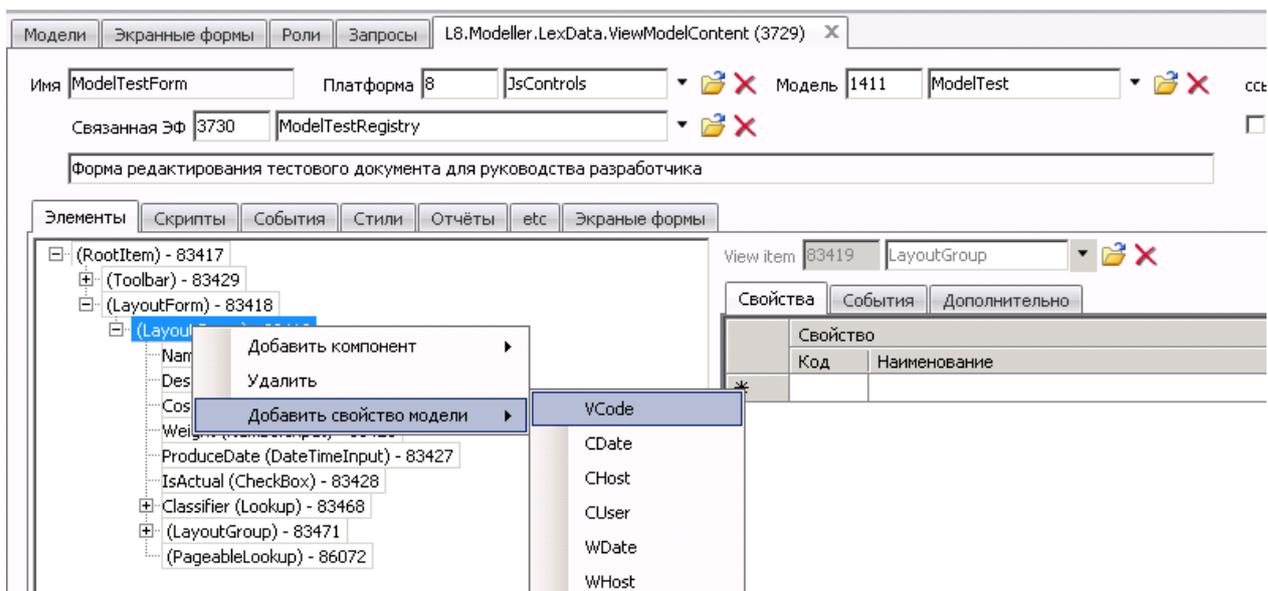
Для удаления этих элементов в моделиере приложения откройте экранную форму (категория "Представления", пункт "Экранные формы"). В дереве элементов формы выберите поле, которое необходимо удалить, нажмите правую кнопку мыши. В контекстном меню выберите пункт "Удалить":



Удаление полей из реестра осуществляется аналогичным образом. Однако, в реестре поле VCode рекомендуется оставить.

Добавление полей в форму

Если вы случайно удалили нужное поле, его можно повторно добавить. Для этого в дереве выберите группу элементов (LayoutGroup), нажмите правую кнопку мыши. В контекстном меню выберите подменю "Добавить свойство модели", в нём выберите поле, которое необходимо добавить:



Замена поля типа TextBox на Мемо

Поля моделей могут иметь тип longstring - длинная строка, тип БД - varchar(max), применяется для хранения описания, примечаний, комментариев и т.п. По умолчанию при генерации интерфейса для таких полей создаются элементы типа TextBox - однострочное текстовое поле. Целесообразно заменить его на многострочное - Мемо. Для этого выберите текстовое поле, которое нужно заменить, нажмите правую кнопку мыши. В контекстном меню выберите пункт "Удалить". Затем выделите группу (LayoutGroup), в которую необходимо добавить поле, нажмите правую кнопку мыши, в контекстном меню выберите пункт "Добавить компонент" - "Мемо". Для созданного элемента заполните свойство bind (привязка) - введите имя поля модели, к которому должен быть привязан элемент (например, value:Comment). Создайте и введите значения для следующих свойств:

- caption - заголовок элемента
- width - ширина поля в пикселах
- controlRows - высота поля в строках

Создание дочерней коллекции документа

Структура многих видов документов предполагает наличие двух частей - головной части ("шапки") документа и дочерней коллекции ("тела", табличной части). С точки зрения хранения информации дочерняя коллекция представляет собой набор строк-записей дочерней (зависимой) модели. Например, пункты выполненных работ в акте сдачи-приёмки, перечень позиций материалов в приходной накладной и т.д. В экранной форме дочерняя модель отображается в виде таблицы.

Для создания дочерней коллекции и его привязки к родительскому документу необходимо выполнить следующие действия:

- создание дочерней модели
- привязка дочерней модели к родительской
- обновление БД
- добавление в ЭФ нового элемента управления - таблицы

Создание дочерней модели

Процесс создания модели подробно описан в разделе статьи руководства "Создание документа на основе модели". Следует особо подчеркнуть, что дочерняя модель должна иметь поле типа BigInt для хранения кодов записей родительской таблицы. Рекомендуется называть

это поле PCode (от англ. Parent Code - родительский код). Дочерняя модель также может иметь поля-справочники. Пример параметров дочерней таблицы показан на рисунке:

Name	DbName	ViewName	Description	Type	Код	Наименование
VCode	VCode	VCode		127	bigint	
CDate	CDate	CDate		461	DateTimeOffset	
CHost	CHost	CHost		73	string	
CUser	CUser	CUser		73	string	
WDate	WDate	WDate		461	DateTimeOffset	
WHost	WHost	WHost		73	string	
WUser	WUser	WUser		73	string	
COrg	COrg	COrg		1	int	
WOrg	WOrg	WOrg		1	int	
PCode	PCode		код записи в родительской таблице	127	bigint	
Name	Name		наименование	73	string	
Length	Length		длина	34	double	
Cost	Cost		стоимость	3	money	
Classifier	Classifier		класс (ссылка на запись в справочнике)	1218	ModelTestClassifier	
ReceiveDate	ReceiveDate		дата получения	461	DateTimeOffset	

Привязка дочерней модели к родительской

Для создания связи между родительской и дочерней моделями необходимо открыть для редактирования родительскую модель (пункт меню "Модели"). На вкладке "Дочерние таблицы" необходимо выбрать из списка тип, автоматически созданный на основе дочерней модели, указать ключевое поле для хранения кодов записей родительской таблицы (например, PCode) и ввести имя дочерней коллекции (оно может быть произвольным). Пример параметров привязки дочерней модели к родительской показан на рисунке:

Имя	Детальная модель	Ключевое поле	VCode
details	Code Наименование ModelTestChild	Code Наименование PCode	1258

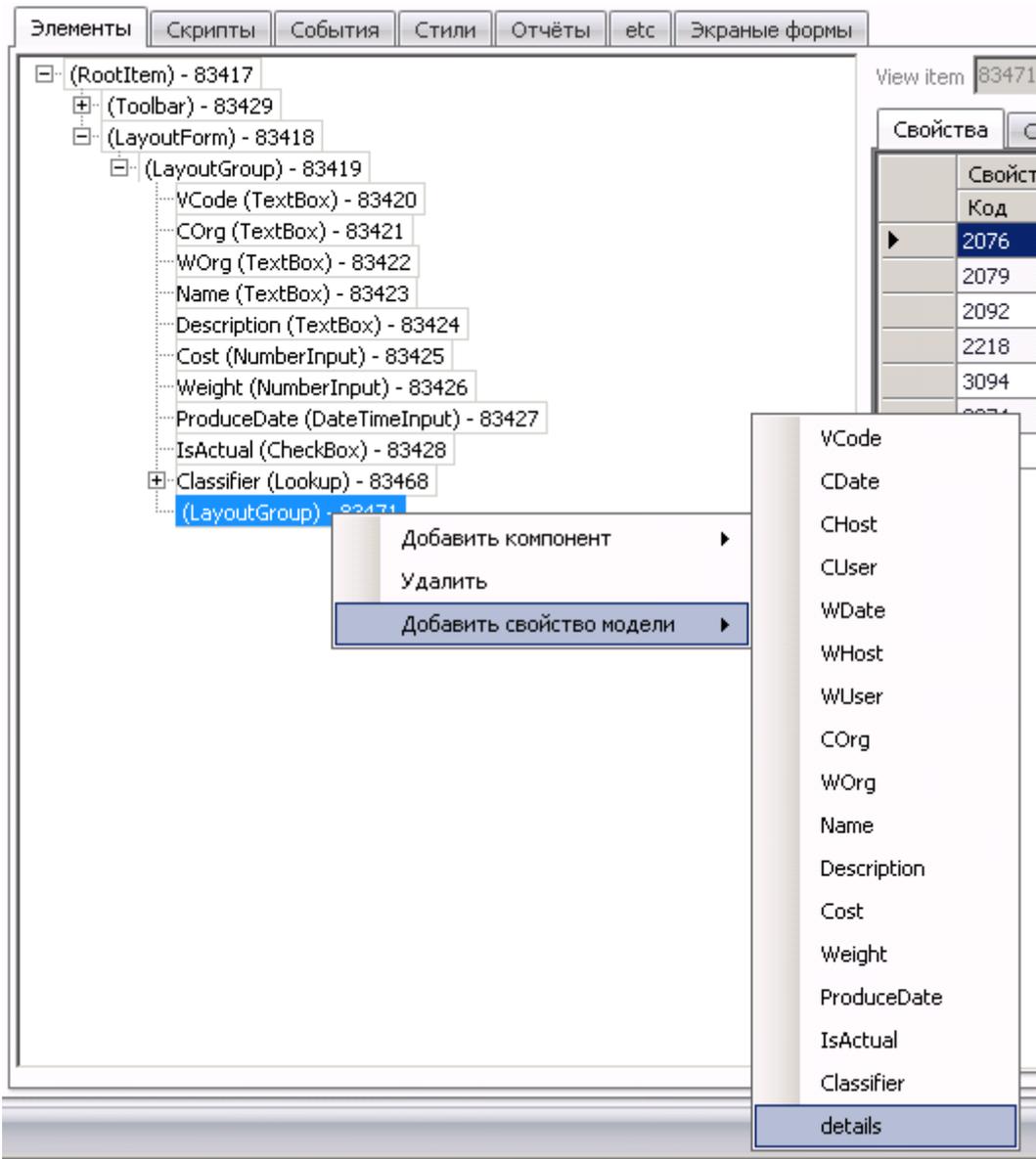
После создания дочерней модели и настройки связей между таблицами необходимо выполнить обновление БД (этот процесс подробно описан в соответствующем разделе руководства). Перечень обновляемых объектов при создании и привязки дочерней таблицы показан на рисунке:

В результате создания и привязки дочерней модели будут сгенерированы и выполнены следующие SQL-запросы:

- создание дочерней таблицы
- создание триггеров обновления системных полей и создания записей исторической таблицы

Добавление в экранную форму дочерней таблицы

Для использования дочерней коллекции необходимо добавить таблицу в экранную форму для редактирования родительского документа. Для этого откройте форму, раскройте дерево элементов (RootItem - LayoutForm - LayoutGroup), выберите контейнер, в который необходимо добавить дочернюю таблицу или создайте новый контейнер (например, LayoutGroup). Нажмите правую кнопку мыши на группе, в контекстном меню выберите пункт "Добавить свойство модели", в списке найдите имя дочерней коллекции и выберите его:



В результате будет создана таблица (Grid), содержащая колонки-поля дочерней модели. Таблица доступна для редактирования - пользователь может добавлять и удалять строки, менять содержимое ячеек. Рекомендуется удалить системные поля (VCode, PCode и др.), т.к. они устанавливаются системой не должны корректироваться пользователем.

Проверка работоспособности документа с дочерней коллекцией

Для проверки работоспособности документа с дочерней коллекцией необходимо открыть экранную форму для его редактирования. Если все вышеописанные действия были выполнены верно, на экране будет отображена таблица с элементами управления:

VCode	<input type="text" value="1"/>
COrg	<input type="text" value="0"/>
WOrg	<input type="text" value="0"/>
Name	<input type="text" value="Муфта соединительная"/>
Description	<input type="text" value="Применяется для соединения 1"/>
Cost	<input type="text" value="1 200.00"/>
Weight	<input type="text" value="5.00"/>
ProduceDate	<input type="text" value="12.11.2015"/> ▼
IsActual	<input checked="" type="checkbox"/>
Classifier	<input type="text"/> ▼

Дочерняя коллекция

Строки: 0 Выделено: 1 <input checked="" type="checkbox"/> Экспорт + Добавить - Удалить					
VCode	CDate	CHost	CUser	WDate	WHost
<input type="text"/>					

По умолчанию, таблица для работы с записями дочерней коллекции доступна для редактирования. Необходимо добавить строки, нажав на соответствующую кнопку, и ввести значения в ячейки таблицы:

Дочерняя коллекция

Строки: 2 Выделено: 1 <input checked="" type="checkbox"/> Экспорт + Добавить - Удалить						
	Name	Length	Cost	Classifier	ReceiveDate	
	Кольцо	12.00	123.00	1	Агрегаты	11.01.2016
	Втулка	14	233.00	1	Агрегаты	12.01.2016

Внимание! Для сохранения изменений дочерней коллекции необходимо сохранить весь документ целиком, нажав на соответствующую кнопку ("Сохранить", либо "Сохранить и закрыть"). В противном случае изменения дочерней коллекции будут потеряны.

Настройка параметров отображения формы с дочерней коллекцией

По умолчанию параметры отображения дочерней коллекции, задаваемые при автоматической генерации, не являются оптимальными. В формах, предназначенных для работы конечного пользователя, крайне рекомендуется провести ряд изменений:

- удалить системные поля
- переименовать заголовки колонок

- настроить ширину колонок

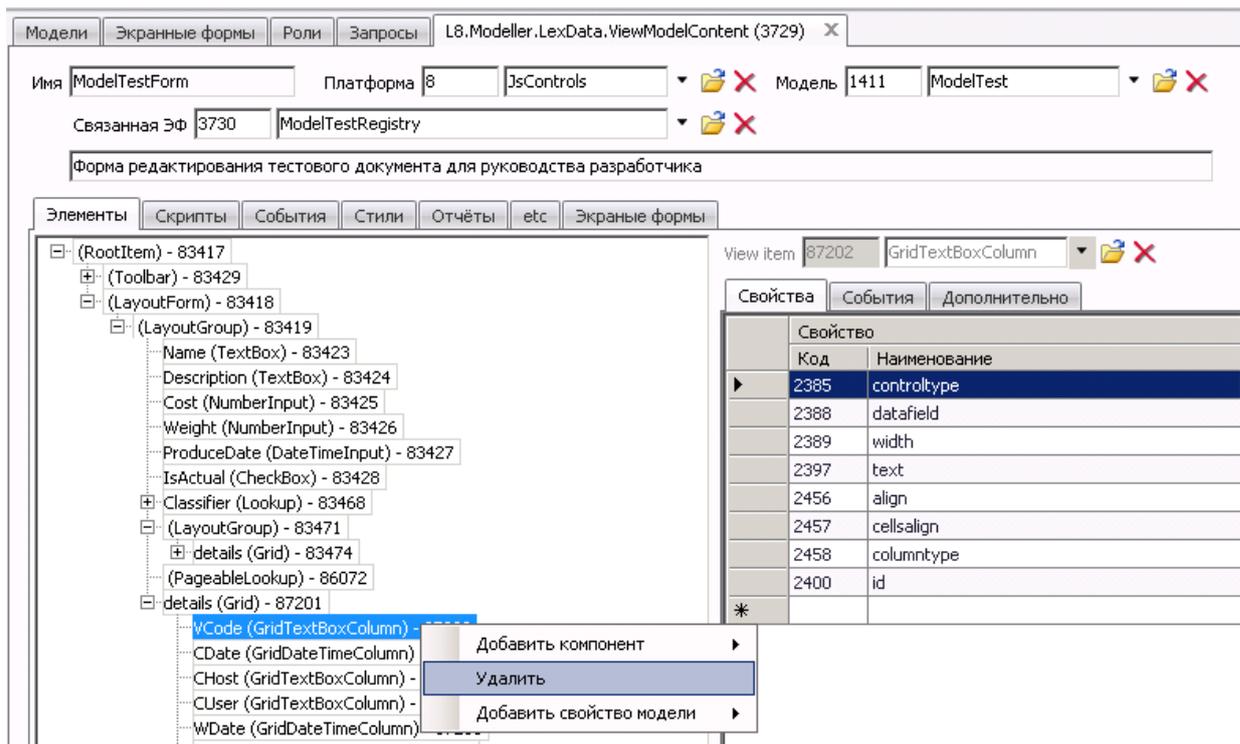
Удаление системных полей

При добавлении таблицы дочерней коллекции в форму документа автоматически создаются колонки, отображающие информацию из системных полей:

- VCode
- CDate
- CUser
- CHost
- WDate
- WUser
- WHost
- COrg
- WOrg
- PCode

В большинстве случаев у пользователя не возникает необходимость редактировать эти данные вручную, т.к. они заполняются автоматически. В этих случаях рекомендуется удалять ненужные колонки.

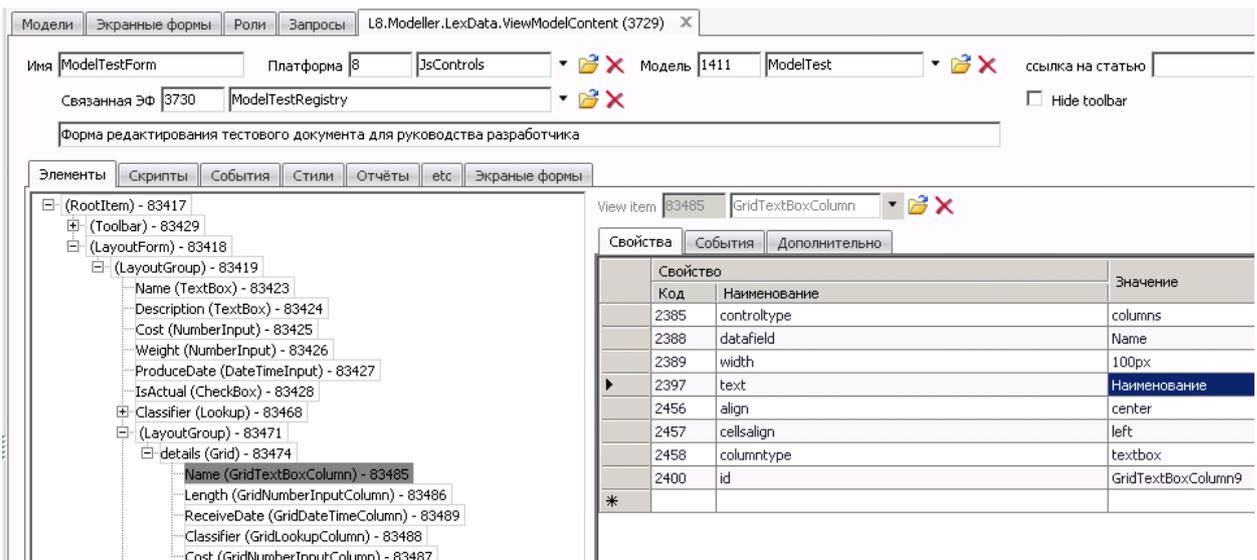
Для удаления колонки разверните в дереве элементов таблицу дочерней коллекции. Выберите поле, которое необходимо удалить, нажмите правую кнопку мыши. В контекстном меню выберите пункт "Удалить":



Подтвердите удаление в диалоговом окне. Повторите эти действия для всех полей, которые необходимо удалить.

Переименование колонок

По умолчанию подписи колонок совпадают с именами полей модели. В большинстве случаев это неприемлемо, т.к. имена полей модели обычно англоязычные. Для переименования колонки разверните в дереве элементов таблицу дочерней коллекции. Выберите поле, которое необходимо переименовать. В списке свойств выберите text, смените его значение на корректное название колонки:



Повторите эти действия для всех полей.

Изменение размеров колонок

При автоматической генерации колонок дочерней таблицы назначается определённая ширина (100px, 150px или др.). Во многих случаях необходимо увеличить или уменьшить ширину колонки (например, для полей "Наименование", "Примечание" и т.п.). Для изменения ширины колонки необходимо изменить свойство width, установив значение в пикселах (точках) - число+px, например, 200px.

Создание справочников и их подключение к документу

Справочник представляет собой, как правило, небольшую таблицу содержащую редко меняющийся перечень уникальных позиций. Кроме служебных полей, модель справочника содержит в большинстве случаев одно или несколько полей (например, наименование). Справочники используются в документах для выбора значений тех или иных реквизитов. На уровне основной таблицы справочник представляет собой поле типа BigInt, является внешним ключом (Foreign Key) и хранит коды записей справочника. На уровне экранной формы для редактирования документа справочник отображается в виде выпадающего списка (Lookup).

Для создания справочника и его подключения к документу необходимо выполнить следующие действия:

- создание модели справочника
- добавление в основную модель поля справочника
- обновление БД для создания поля в таблице
- создание ЭФ и реестра для ввода значений справочника, ввод значений справочника
- добавление выпадающего списка в ЭФ основного документа

Создание модели справочника

Процесс создания модели подробно описан в разделе статьи руководства "Создание документа на основе модели". Модель справочника должна содержать поле, являющееся первичным ключом (primary key), а также одно или несколько содержательных полей (например, наименование, описание), которые будут отображаться в выпадающем списке.

Пример параметров справочника показан на рисунке:

Сохранить и закрыть изменения Закрыть не сохраняя Обновить содержание Создать скрипт Экспорт Добавить поля для холдинга История реквизита История

Модели Обновление БД Экранные формы L8.Modeller.LexData.ModelContent - Новый

Имя модели: ModelTestClassifier Primary key: 25572 VCode

Имя таблицы: ModelTestClassifier TypeCode property:

Имя вьюшки: TypeName property:

PrimaryKeyDomain: ModelTestClassifier DateProperty:

Display name: Документ с историей

Пространство имен: 3 test

Описание: Тестовый справочник

Свойства Дочерние таблицы События

Name	DbName	ViewName	Description	Type	
				Код	Наименование
VCode	VCode	VCode		127	bigint
CDate	CDate	CDate		461	DateTimeOffset
CHost	CHost	CHost		73	string
CUser	CUser	CUser		73	string
WDate	WDate	WDate		461	DateTimeOffset
WHost	WHost	WHost		73	string
WUser	WUser	WUser		73	string
COrg	COrg	COrg		1	int
WOrg	WOrg	WOrg		1	int
Name	Name		Наименование	73	string
Description	Description		Описание	2	longstring
Cost	Cost		Стоимость	3	money

Добавление в основную модель поля справочника

В основную модель документа необходимо добавить новое поле: введите наименование (колонка Name), имя поля в таблице БД (колонка DbName), описание поля (колонка Description). В колонке Type выберите созданную ранее модель справочника. Пример списка полей основной таблицы приведён на рисунке:

Модели Обновление БД Экранные формы L8.Modeller.LexData.ModelContent (1411)

Имя модели: ModelTest Primary key: 25550 VCode

Имя таблицы: ModelTest TypeCode property:

Имя вьюшки: TypeName property:

PrimaryKeyDomain: ModelTest DateProperty: 25565 ProduceDate

Display name: Документ с историей

Пространство имен: 3 test

Описание: Тестовая модель для документации

Свойства Дочерние таблицы События

Name	DbName	ViewName	Description	Type	
				Код	Наименование
VCode	VCode	VCode		127	bigint
CDate	CDate	CDate		461	DateTimeOffset
CHost	CHost	CHost		73	string
CUser	CUser	CUser		73	string
WDate	WDate	WDate		461	DateTimeOffset
WHost	WHost	WHost		73	string
WUser	WUser	WUser		73	string
COrg	COrg	COrg		1	int
WOrg	WOrg	WOrg		1	int
Name	Name		наименование	73	string
Description	Description		описание	2	longstring
Cost	Cost		стоимость	3	money
Weight	Weight		вес	34	double
ProduceDate	ProduceDate		дата производства	466	Date
IsActual	IsActual		признак актуальности	316	bool
Classifier	Classifier		справочник	1218	ModelTestClassifier

Важно! После создания модели справочника и изменения основной модели необходимо произвести обновление БД, этот процесс подробно описан в разделе статьи руководства "Создание документа на основе модели". В результате будут сгенерированы и выполнены следующие запросы:

- создание новой таблицы справочника (CREATE TABLE)
- добавление поля в существующую основную таблицу (ALTER TABLE)

- создание внешнего ключа
- создание триггеров обновления системных полей и создания записей исторической таблицы

Пример списка выполняемых операций показан на рисунке:

Приоритет	Операция	Описание	Скрипт	Статус	Сообщение	Объект
10	Создание триггера истории	Создание триггера истории таблицы OnAction_ModelTestClassifier_ForHist...	create trigger [OnAction_ModelTestClassifier_ForHistory]	Невыполнено		ModelTestC
10	Создание триггера обновлен...	Создание триггера заполнения системных данных для таблицы ModelTes...	create trigger [OnUpdate_ModelTestClassifier_ForColumnsControl]	Невыполнено		ModelTestC
3	Создание колонок	Создание столбца Classifier таблицы ModelTest	ALTER TABLE [ModelTest] ADD [Classifier] bigint NULL	Невыполнено		ModelTest
10	Создание внешнего ключа	Создание внешнего ключа FK_Classifier_ModelTest_to_ModelTestClassifier ...	ALTER TABLE [ModelTest] WITH CHECK ADD CONSTRAINT [FK_...	Невыполнено		ModelTest
2	Создание таблицы	Создание таблицы ModelTestClassifier	CREATE TABLE [ModelTestClassifier] ([VCode] bigint NOT NULL, ...	Невыполнено		ModelTestC

Создание ЭФ и реестра для ввода значений справочника

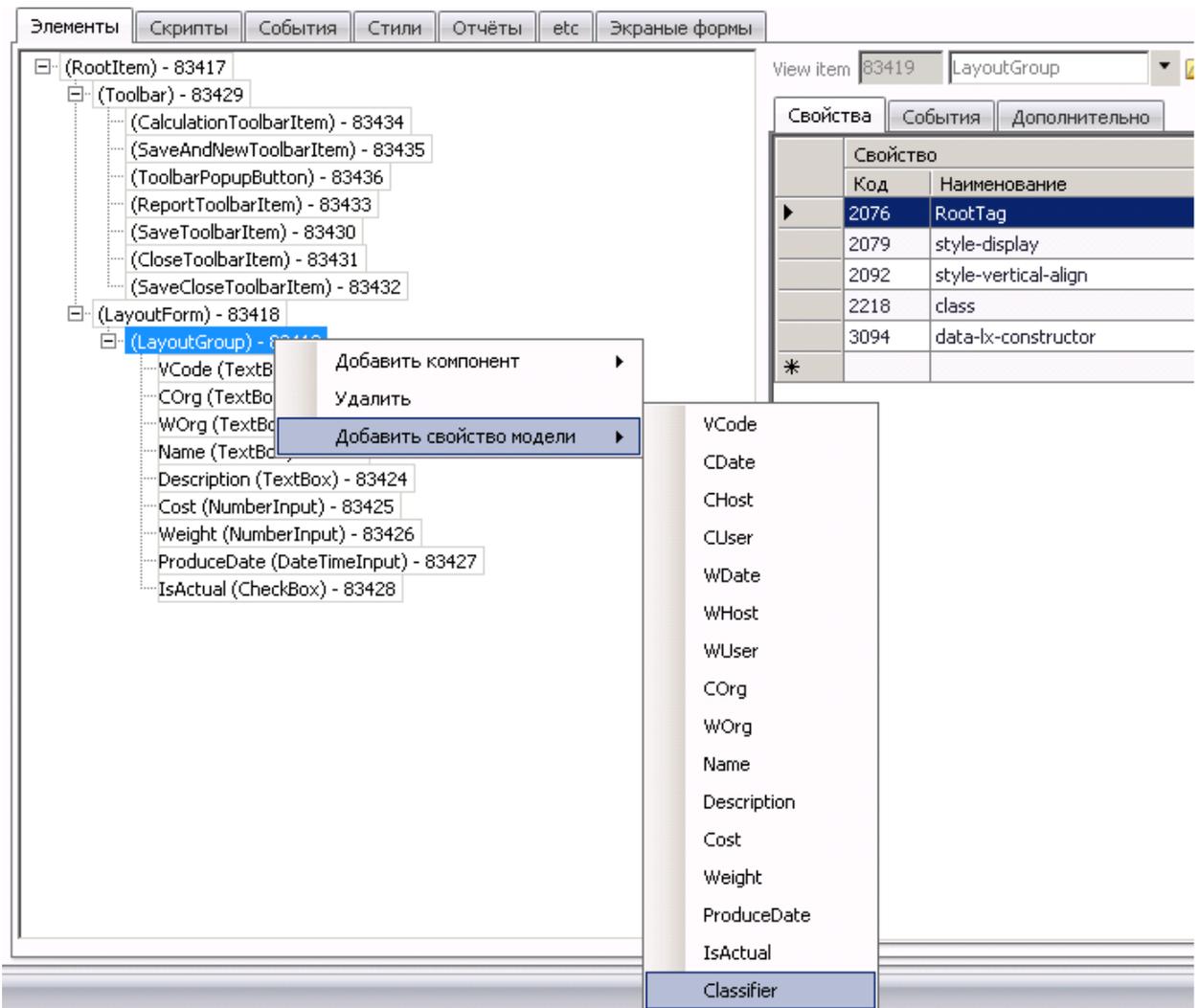
Существует два способа заполнения таблицы справочника значениями:

- ввод данных вручную с помощью экранной формы для редактирования
- импорт данных в таблицу средствами СУБД MS SQL Server

Процесс создания экранной формы и реестра подробно описан в разделе статьи руководства "Создание документа на основе модели". Для модели справочника, так же, как и для любого другого документа, необходимо создать форму для редактирования и реестр, дать права на работу с ЭФ ролям и создать соответствующий пункт меню. После этого появится возможность вводить данные в таблицу справочника.

Добавление выпадающего списка в ЭФ основного документа

Для выбора в документе одного из значений справочника необходимо добавить выпадающий список. Для этого откройте экранную форму редактирования, разверните дерево элементов (RootItem - LayoutForm - LayoutGroup). Выберите контейнер (группу LayoutGroup или др.), в которую необходимо добавить новое поле - справочник. Нажмите на ней правую кнопку мыши, выберите в контекстном меню пункт "Добавить свойство модели", в списке найдите имя поля - справочника:



В результате в дерево будет добавлен новый элемент управления - выпадающий список (Lookup).

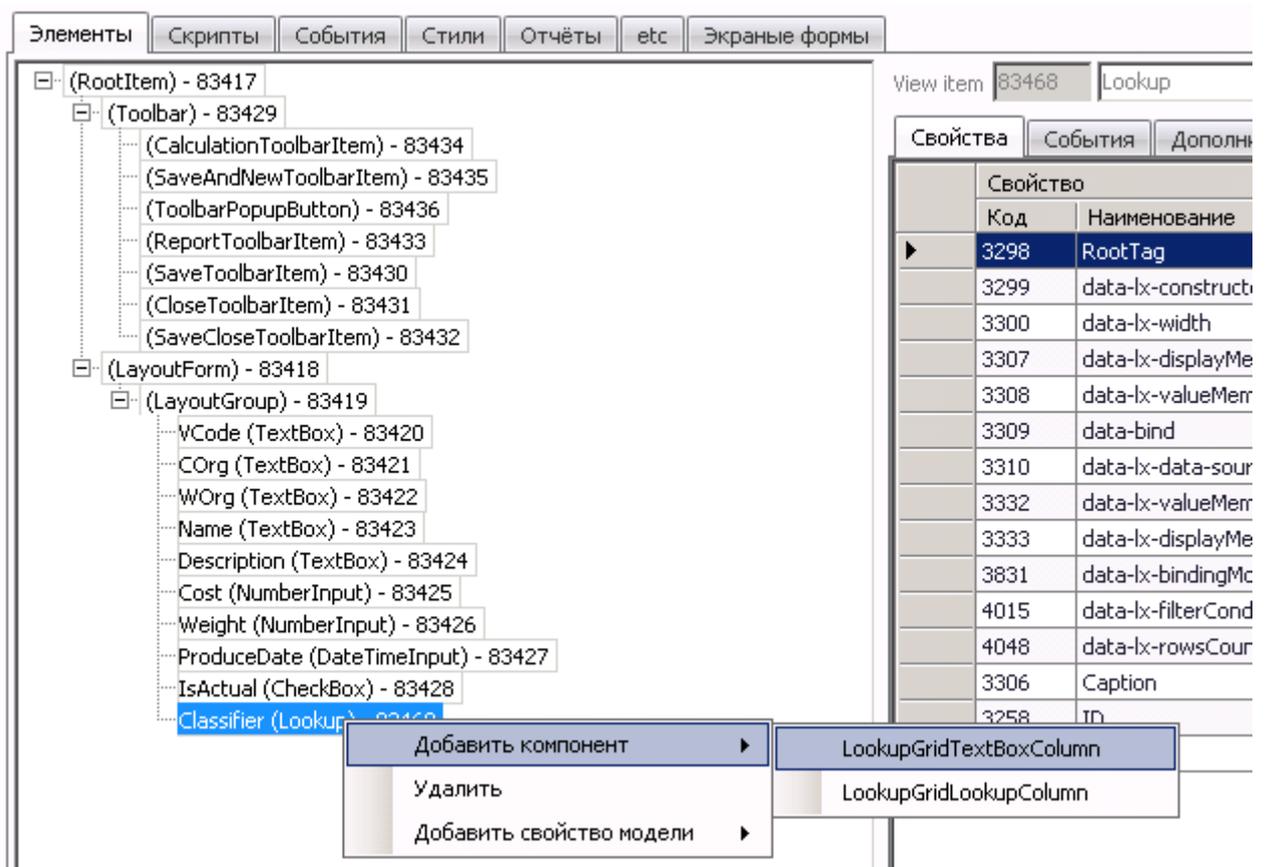
Добавление поля справочника в реестр

Процесс добавления поля справочника в реестр аналогичен работе с экранной формой. Для этого необходимо открыть реестр, раскрыть дерево элементов, найти элемент AnalyticalGrid, нажать правую кнопку мыши, выбрать пункт "Добавить свойство модели" и в выпавшем списке найти поле-справочник.

Создание многоколоночного списка

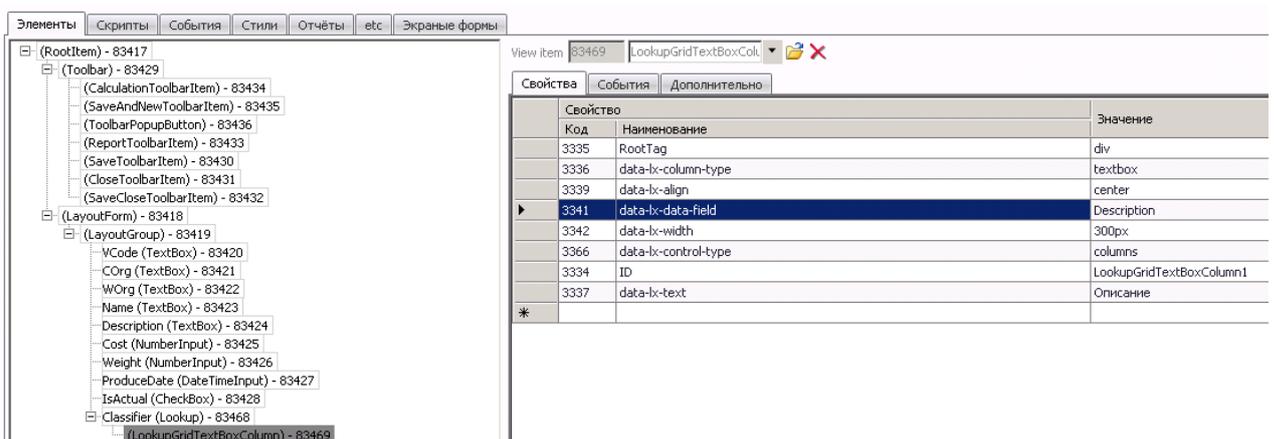
По умолчанию выпадающий список содержит два поля - код и наименование. Однако, система предоставляет возможность создания нескольких колонок, что делает справочники в документе более информативными.

Для добавления колонок к списку в дереве элементов выберите выпадающий список (Lookup), нажмите правую кнопку мыши, в контекстном меню выберите пункт "Добавить компонент" - "GridTextBoxColumn". Существует возможность добавления двух типов колонок - текстовых и списковых.



В результате будет добавлен дочерний элемент - колонка списка. Для него необходимо указать ряд параметров:

- datafield - имя поля модели, значение которого будет отображаться в колонке списка
- width - ширина колонки в пикселях (например, 150px)
- text - отображаемый заголовок колонки



Колонка, добавленная в список таким образом, будет отображаться после полей кода и наименования.

Проверка работоспособности документа со справочником

Для проверки работоспособности созданного документа необходимо открыть в браузере страницу системы и авторизоваться (ввести логин и пароль).

Для того, чтобы в основном документе можно было выбрать значения в списке-справочнике, необходимо ввести значения в таблицу данного справочника с помощью соответствующей формы для редактирования или путём импорта в соответствующую таблицу БД. Для ввода

значений вручную в главном меню в области "Мои документы" следует раскрыть соответствующую категорию и нажать на пункт меню реестра справочника. Используя реестр и форму, необходимо создать несколько записей в модели справочника.

Lexema.ru Создать Удалить

от 01.11.2015 - 30.11.2015

ModelTestClassifier

	VCode	COrg	WOrg	Name	Description	Cost	Создание	
							Дата	Хост
<input type="checkbox"/>	1	0	0	Агрегаты	Составные и...		07.12.2015	10.8.56.4
<input type="checkbox"/>	2	0	0	Узлы	Компоненты...		07.12.2015	10.8.56.4

После этого можно использовать их при выборе значений в экранной форме в основном документе. Откройте реестр основного документа, создайте новый документ или откройте для редактирования существующий. В поле-выпадающем списке теперь можно выбрать значения из модели справочника. На рисунке показан вид выпадающего списка, содержащий несколько колонок:

Lexema.ru Сохранить Закрыть Сохр. Закрыть Печать Проводки Сохр. Создать

от 01.11.2015 - 30.11.2015

VCode: 1

COrg: 0

WOrg: 0

Name: Муфта соединительная

Description: Применяется для соединения 1

Cost: 1 200.00

Weight: 5.00

ProduceDate: 12.11.2015

IsActual:

Classifier:

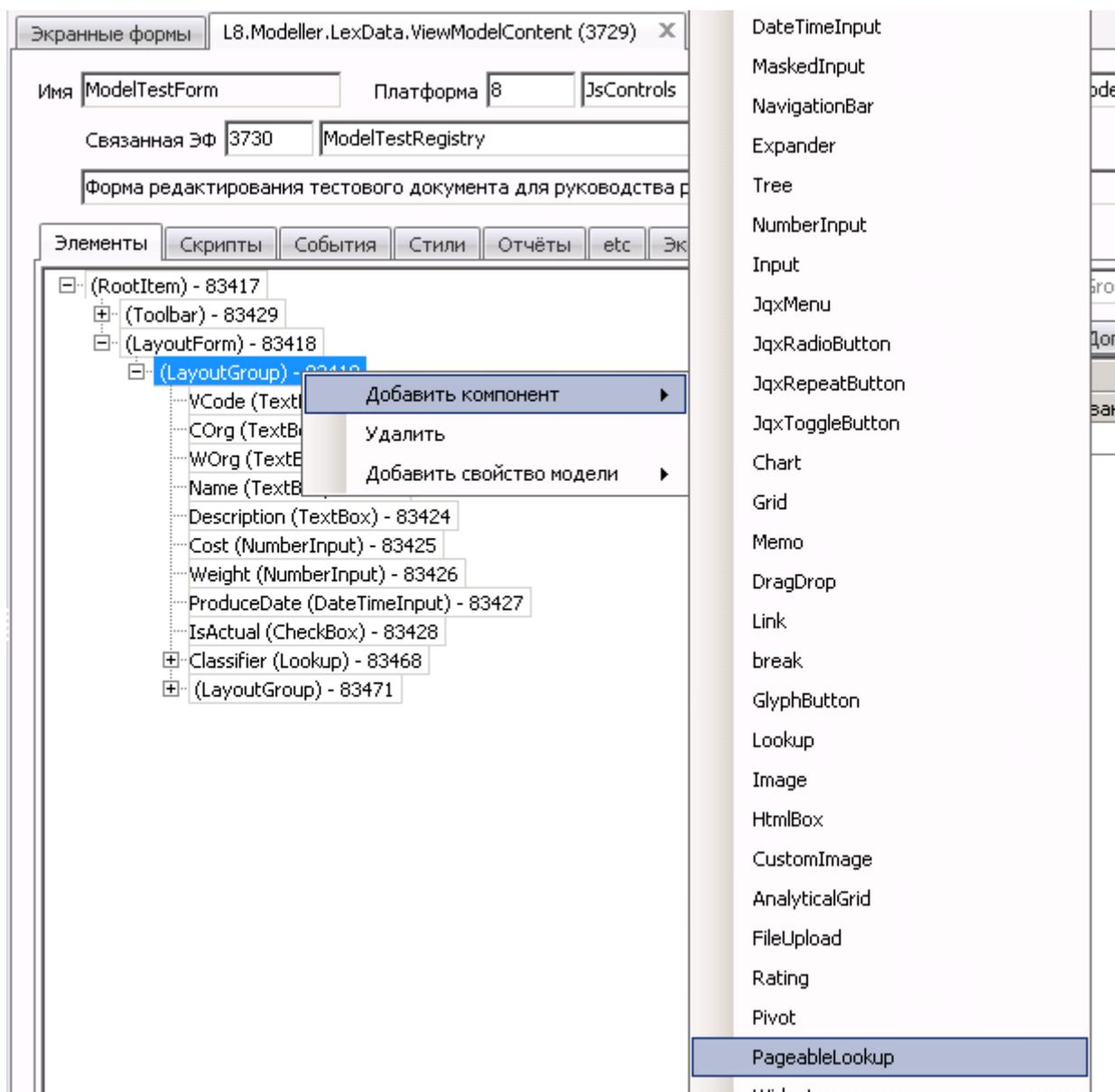
Код	Наименование	Описание
1	Агрегаты	Составные изделия
2	Узлы	Компоненты изделий

Добавить Удалить

WHost

Справочники с большим количеством данных

При загрузке экранной формы с элементом типа Lookup, привязанным к полю справочного типа, загружаются и все данные из модели-справочника. Если он содержит большой объём данных, то время загрузки формы может стать неприемлемо большим. Чтобы избежать этого, элемент типа Lookup необходимо заменить на PageableLookup - список с возможностью постраничного вывода записей. При использовании PageableLookup в его свойстве указывается максимальное количество записей, которые будут загружены при начальной загрузке формы. При наборе текста в поле PageableLookup производится поиск среди всех записей модели, включая и не загруженные в данный момент. Поиск осуществляется и по кодовому полю, и по полю значения. Для добавления списка с возможностью постраничного вывода необходимо нажать правой кнопкой мыши на соответствующем контейнере (LayoutGroup) и в контекстном меню выбрать пункт "Добавить компонент"->"PageableLookup":



Так как этот элемент был добавлен вручную, а не создан при генерации формы или добавлении свойства модели, его параметры необходимо настроить вручную:

- bind - имя поля модели, в которое будет сохранено выбранное значение. Содержит префикс "lookupValue:", например, "lookupValue:Base.Contractor", где Contractor - имя поля модели.
- dataSource - параметры источника данных. Должен иметь следующий вид: "type=ТИП_ИСТОЧНИКА&name=ПРОСТР_ИМЁН.ИМЯ_ИСТОЧНИКА". Существует два типа источников данных - модель (model) и запрос (query). Пример значения данной

настройки: "type=model&name=Base.Position". Base - пространство имён модели, Position - имя модели

- caption - подпись элемента на форме
- topCount - кол-во записей, загружаемых за один раз (по умолчанию - 100)

Свойства bind и dataSource добавляются при создании компонента, а свойства caption и topCount необходимо добавить вручную:

Свойство		Значение	Редактор
Код	Наименование		
3933	width	200px	
3938	displayMember	Name	
3939	valueMember	VCode	
3940	bind	lookupValue:Position	
3941	dataSource	type=model&name=Base.Position	
3944	valueMemberColumnText	Код	
3945	displayMemberColumnText	Наименование	
4017	filterCondition	contains	
4027	rowCount	9	
3931	id	PageableLookup1	
3937	caption	Должность	
3949	topCount	500	
*			

Проверка работоспособности списка с постраничным выводом

При открытии списка с постраничным выводом на форме загружается кол-во записей, заданное параметром topCount. При необходимости можно загрузить следующую порцию данных, нажав на ссылку "Загрузить ещё X":

The screenshot shows a list of records with columns 'Код' (Code) and 'Наименование' (Name). The records are numbered 492 to 500. A 'Загрузить ещё 500' (Load more 500) button is visible below the list. The 'Должность' (Position) field is also visible below the list.

Код	Наименование
492	Аппаратчик сгу...
493	Аппаратчик сеп...
494	Аппаратчик сеп...
495	Аппаратчик-сер...
496	Аппаратчик сжи...
497	Аппаратчик сжи...
498	Аппаратчик-сик...
499	Аппаратчик сил...
500	Аппаратчик син...

Строки: 0 Выде.

Загрузить ещё 500

Должность

Для поиска нужных записей необходимо ввести наименование или код в поле списка:

Код	Наименование
1279	Изготовитель-н...
1837	Мастер-наладчи...
2561	Наладчик автом...
2562	Наладчик автом...
2563	Наладчик автом...
2564	Наладчик автом...
2565	Наладчик автом...
2566	Наладчик асбес...
2567	Наладчик геофи...

Должность:

Создание аналитической формы на основе запроса

Аналитическая форма предназначена для построения отчётов на основе выборки данных из одной или нескольких таблиц или представлений БД. Если для обыкновенной экранной формы источником данных является модель (см. Создание_документа_на_основе_модели), то для аналитической формы источником данных является запрос или Lexema_Language. Для создания аналитической формы на основе запроса необходимо выполнить следующие действия:

- написать и отладить SQL-запрос на выборку
- создать источник данных - запрос и настроить его параметры
- создать экранную форму, не указывая источник данных
- в экранной форме:
 - создать элементы управления для ввода значений параметров запроса
 - создать таблицу и настроить параметры её колонок для отображения данных - результатов запроса
 - создать кнопку и соответствующий обработчик события для выполнения запроса и отображения его результатов в таблице
- предоставить права на созданную экранную форму
- создать пункт меню для открытия ЭФ
- проверить работоспособность документа

Написание и отладка SQL-запроса на выборку

Источником данных для аналитической формы, как правило, является SQL-запрос на выборку данных, написанный с помощью SQL-оператора SELECT. Подобный запрос может быть сколь угодно сложным и включать операторы соединения (JOIN) различных типов, операторы объединения (UNION) и вложенные запросы SELECT. В большинстве случаев запросы, лежащие в основе аналитической формы, содержат условия в предложении WHERE и

включают параметры (переменные языка T-SQL). Впоследствии значения параметров запроса вводятся с помощью элементов управления аналитической экранной формы.

Перед сохранением запроса в виде объекта системы Lexema.ru рекомендуется отладить его текст с помощью стандартных инструментов СУБД MS SQL Server - SQL Management Studio. Например, текст запроса для выборки данных из тестовой таблицы может быть следующим:

```
-- объявление переменных-параметров и присвоение им значений (для отладки
запроса в SQL Studio)
declare @costFrom money = 500, @costTo money = 1300
declare @isActual bit = 1
-- примечание: при переносе текста SQL-скрипта в запрос Lexema необходимо
удалить или закомментировать объявление параметров
select VCode, Name, [Description], Cost, [Weight], ProduceDate, IsActual,
Classifier
from ModelTest
where (@costFrom is null or Cost >= @costFrom)
and (@costTo is null or Cost <= @costTo)
and (@isActual is null or ISNULL(IsActual, 0) = @isActual)
```

Результат выполнения запроса в SQL Management Studio показан на рисунке:

	VCode	Name	Description	Cost	Weight	ProduceDate	IsActual	Classifier
1	1	Муфта соединительная	Применяется для соединения труб	1200,00	5	2015-11-12	1	NULL
2	2	Корпус металлический	NULL	800,00	2	2015-11-25	1	NULL

Создание запроса (источника данных) в Lexema.ru

В дереве навигации в разделе Модели выберите пункт "Запросы". Создайте новый запрос с помощью кнопки «Создать документ» на панели инструментов или кнопки «Insert» на клавиатуре. Поле "Имя" является обязательным для заполнения. Выберите соответствующее пространство имён из списка.

Настройка параметров запроса

Как правило, большинство запросов содержит входные параметры - это позволяет сделать отчёт более гибким и настраивать отображение его результатов в зависимости от потребностей пользователя. В интерфейсе редактирования запросов Lexema.ru содержится таблица для указания параметров запросов, которая включает следующие поля:

- Name - имя переменной-параметра
- Parameter Type (Код, Наименование) - тип данных параметра
- Select (флажок) - возможность использовать параметр в запросе на выборку
- Update (флажок) - возможность использовать параметр в запросе на обновление
- Insert (флажок) - возможность использовать параметр в запросе на вставку
- Delete (флажок) - возможность использовать параметр в запросе на удаление

Пример настройки параметров запроса из предыдущего раздела показан на рисунке:

*									
Имя	Имя	31P	роль						
CostTo	CostTo	3	школа						
CostFrom	CostFrom	3	школа						
Имя	Имя	КОД	Наименование	Значение	Глобальный ID	Имя	Имя	Имя	Имя

Имя: Имя | Предоставление прав: 3 | 3 | 3 | 3

Вкладки: Изменить, Создать, Обновить, Удалить, Экспорт, Импорт, Обновить, Создать, Удалить, Экспорт, Импорт

Иконки: Добавить, Удалить, Обновить, Создать, Удалить, Экспорт, Импорт

Панель: Изменить, Документ

Ввод текста запроса

Интерфейс редактирования запросов Lexema.ru содержит четыре вкладки (Select, Update, Insert, Delete) для редактирования текста запросов для выполнения операций (выборки, изменения, вставки и удаления). Пример текста запроса на выборку показан на рисунке:

```

1 -- объявление переменных-параметров и присвоение им значений (для отладки запроса в SQL Studio)
2 --declare @costFrom money = 500, @costTo money = 1300
3 --declare @isActual bit = 1
4 -- примечание: при переносе текста SQL-скрипта в запрос Lexema необходимо удалить или закомментировать объявление параметров
5 select VCode, Name, [Description], Cost, [Weight], ProduceDate, IsActual, Classifier
6 from ModelTest
7 where (@costFrom is null or Cost >= @costFrom)
8 and (@costTo is null or Cost <= @costTo)
9 and (@isActual is null or ISNULL(IsActual, 0) = @isActual)

```

Примечание: для обращения к параметру в тексте запроса необходимо указать его имя, начинающееся с символа @ (как к обычной переменной в T-SQL), например @name.

Предоставление прав на запрос

Для предоставления пользователям возможности работы с формой на основе запроса необходимо настроить права доступа к созданному источнику данных для одной или нескольких ролей. Для работы с этим механизмом необходимо в дереве навигации открыть раздел «Администрирование прав доступа», пункт «Роли». В списке ролей выбрать необходимую роль (например, роль Super). Откройте роль на редактирование с помощью кнопки «Открыть документ на изменение» или двойным щелчком мышки на строке.

	VCode	RoleName	Description	GlobalId
▶	1	Повар		6D5E9E76-4F2D-409D-95E4-089E47B29A47
	2	АдминистраторСистемы		F8FD7D9E-9831-4D9C-B784-743BBFB055E
	3	Торговля		47192C0A-3BC9-471A-80A1-2FB12CECCF32
	4	CRM		5D425929-BF41-4BB6-84D1-1F47C25F200C
	5	Бухгалтер склада		44502553-43C4-4FA0-8762-6D334A7E12E9
	6	Официант		2869B636-541B-48F3-8850-0D892C3CC4CD
	7	Кадровик		CC758E84-EE1B-4D2B-8DF5-D923707151E3
	8	UGH		9372601A-A9BA-4F6D-9C8E-C12E8504FD35
	10	Administrators		9EA19F07-7008-4398-A728-C3FF5806E39C
	40	TESTTTTT		775D2F49-3BC4-4940-AAF7-F505BB29AE64
	50	testRole		97EBB771-DCFO-4FA7-BA58-4DE949D33515
	60	NewTestRole		5A05590C-ABEE-46B1-B224-188DBF88D032
	73	Заявки		83FE274D-8C5E-40EB-8D46-80BECFA91295
	74	Super		20BFE948-8E2B-46A9-A197-E9D4E1B3FA94
	75	Бухгалтер		78CEA60F-48CB-46F4-9E3F-74FB66AA1B24

Перейдите на вкладку "Запросы", затем необходимо перейти в конец списка, в колонке "Query - Наименование" ввести название созданного ранее запроса. Поля VCode, Namespace и GlobalId заполняются автоматически. Для предоставления текущей роли прав на создание,

чтение, изменение и удаление записей необходимо соответствующим образом установить или снять галочки в столбцах Create, Read, Update и Delete. В настоящее время запросы используются исключительно для выборки данных (Read), остальные операции не реализованы. Если запрос отсутствует в списке, по умолчанию роль не обладает никакими правами доступа к результатам работы запроса.

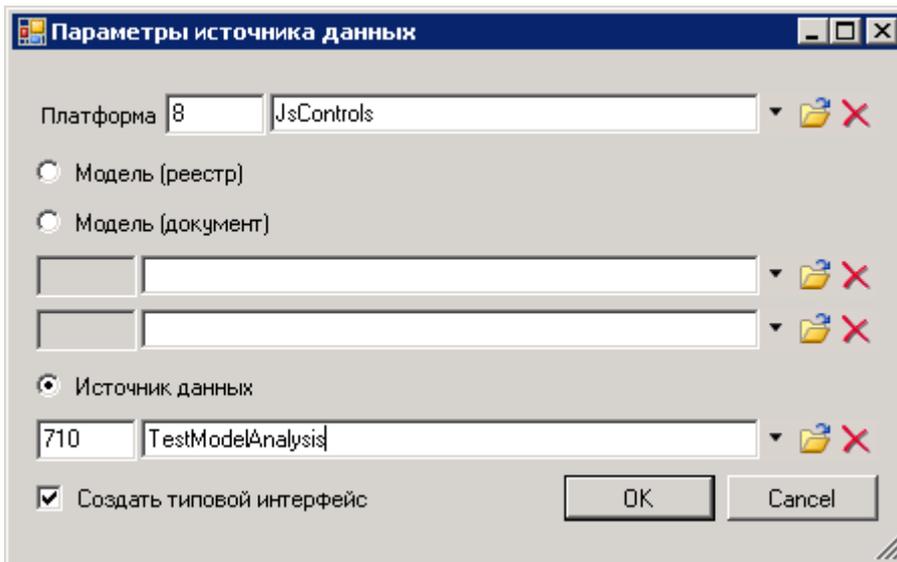
Роль										
Имя Super										
Элементы управления		Модели		Запросы		Вложенные роли				
VCode	Namespace		Query		Create	Read	Update	Delete	GlobalId	
	Код	Наименование	Код	Наименование						
744	1	Base	717	GenDocNumMobile	✓	✓	✓	✓	FEFCD087-C888-4273-8747-57FA01D28F5C	
750	1	Base	721	TypeStoreOperations	✓	✓	✓	✓	C9707673-919D-4CAA-A8F1-71B19D9D93D5	
752	13	DocflowDocuments	720	GetDocumentType	✓	✓	✓	✓	37484C57-37E8-43DC-81CA-AC9291808D1E	
756	1	Base	723	StandartOperation	✓	✓	✓	✓	272C3C24-30D8-483A-BDF5-F75102FFFF02	
757	8	EArchive	722	ContentsOfDocument	✓	✓	✓	✓	E923A404-5071-471C-9900-741328F8C56B	
799	3	test	727	GetRequests	✓	✓	✓	✓	FAA67BCD-1677-480C-8B54-9FE0203623EF	
802	1	Base	728	GetOrdersCashierMobile	✓	✓	✓	✓	2ACEC3BD-0CFC-4F6C-8695-7D39D4965085	
806	8	EArchive	729	JournalOfScannedDocuments	✓	✓	✓	✓	322A316B-5D9E-472A-AE97-86794A8C019B	
808	1	Base	732	loadBirthdayPersons	✓	✓	✓	✓	4F1C0708-3518-4982-AA44-CBE380C90F8C	
809	1	Base	733	CalculationSum	✓	✓	✓	✓	63FEE86D-74B6-4F17-B6A2-AAADAD0426BD	
810	1	Base	737	VisibleAnalytics	✓	✓	✓	✓	EB604B47-0162-482D-81F9-2E4C4B12EADE	
813	1	Base	739	BillPrintingAnalysis	✓	✓	✓	✓	3360F873-79C9-48FA-B929-250B311AB0BD	
816	1	Base	742	InsertPlanMaterialProcurementDetail	✓	✓	✓	✓	D23331CF-FDCD-4635-A021-5D0E98501D33	
817	1	Base	744	CreateShiftCashierMobile	✓	✓	✓	✓	6597ECBE-2C70-462E-A506-FADDAE720132	
819	1	Base	745	CreateOrderCashierMobile	✓	✓	✓	✓	77A81707-2993-4C29-A5F6-2F7760A30B37	
838	1	Base	752	InsertPlanMaterialProcurementDetail1	✓	✓	✓	✓	7CE31DB1-6B49-4798-B03A-07A60DCB45D4	
839	1	Base	755	PlanMaterialProcurementForPrint	✓	✓	✓	✓	4D869487-A872-4D6B-B6D1-32F9B7AB42A0	
840	1	Base	756	CloseShiftCashierMobile	✓	✓	✓	✓	059AA558-15AB-4C71-9081-4CAFE6FC6CC1	
844	1	Base	757	InsertPlanMaterialProcurementDetail1_One	✓	✓	✓	✓	BA57D34C-F549-48DC-B17E-90851D299916	
845	1	Base	758	ReturnOrderCashierMobile	✓	✓	✓	✓	3394D936-55B3-43B0-B610-BAE63E875C19	
849	3	test	760	testDateReport	✓	✓	✓	✓	6B2CC46A-1FE9-4736-A438-3C9C7B006BFF	
850	3	test	765	pivotTest2	✓	✓	✓	✓	83705F12-CD2E-490F-A21F-0E006E3B757C	
851	1	Base	767	FoodMenuNameNew	✓	✓	✓	✓	6B2570CC-E632-4112-9D83-7DC8C8B58B6D6	
852	1	Base	768	GarsonBonusesQuery	✓	✓	✓	✓	612DB16B-034B-47F5-8B12-1673ADEED6B7	
853	1	Base	769	TreeTest	✓	✓	✓	✓	49D63516-5582-4E83-B4D4-4F633232827A	
854	3	test	771	FileFormats	✓	✓	✓	✓	305694D8-1120-452C-B699-FEDA00DADD7D	
855	1	Base	773	ToCalculateProcurement	✓	✓	✓	✓	6A3DB322-4166-4C98-9D78-0F9A665B76D9	
856	1	Base	775	Test03022016Pol	✓	✓	✓	✓	3EB47F44-DED6-4EA2-B744-49E5776351EF	
857	3	test	710	TestModelAnalysis	✓	✓	✓	✓	36AE21D3-4659-4133-AF50-316EA906FE5D	

Создание экранной формы

Экранная форма может быть создана не только на основе модели, но и на основе запроса. В этом случае она может решать два комплекса взаимосвязанных задач:

- Отображение элементов управления для установки параметров запроса
- Отображение записей в табличной части, удовлетворяющих параметрам запроса

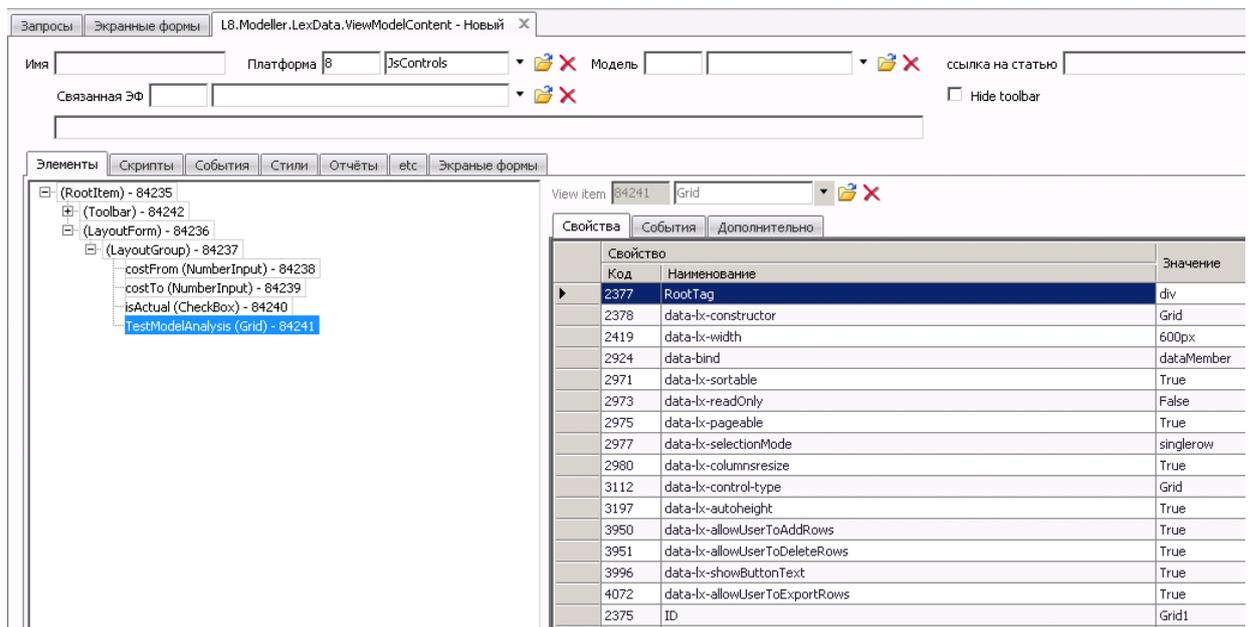
Для создания формы на основе запроса в дереве навигации в разделе «Представления» выберите пункт «Экранные формы». Создайте новую форму с помощью кнопки «Создать документ» на панели инструментов или с помощью клавиши «Insert» на клавиатуре. В диалоговом окне укажите платформу JsControls, снимите флажок "", выберите пункт «Источник данных», из выпадающего списка выберите запрос, созданный на предыдущем шаге.



После нажатия кнопки "OK" будет создана форма на основе запроса. При этом, в отличие от форм, построенных на основе модели, не будет автоматически сгенерировано имя формы, поле "Модель" также будет пустым. В дереве структуры формы автоматически создаются элементы управления для ввода значений параметров, а также таблица для отображения результатов запроса. Однако, ряд действий необходимо произвести вручную:

- добавить поля таблицы и настроить их свойства
- создать кнопку выполнения запроса и написать соответствующий скрипт-обработчик.

Также желательно установить заголовок окна с помощью свойства Title элемента RootItem. Структура новой аналитической формы, созданной на основе запроса, показана на рисунке:

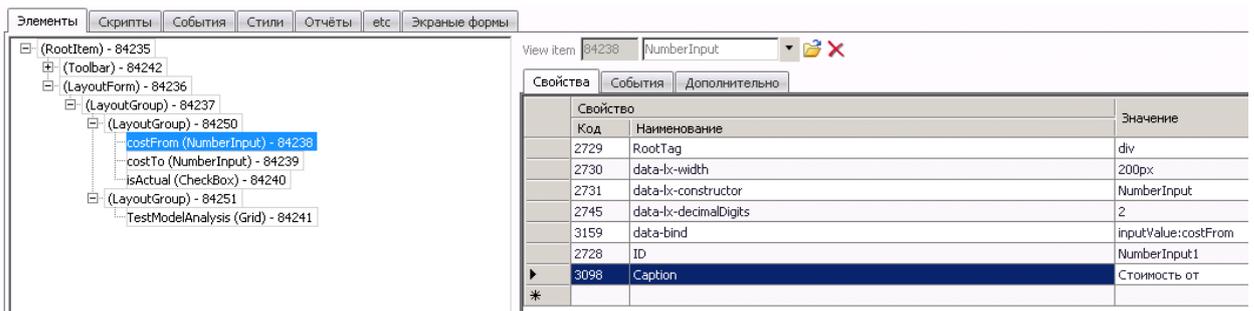


Создание элементов управления для ввода значений параметров запроса

Как было сказано ранее, элементы управления для ввода значений параметров запроса генерируются автоматически при создании формы. Однако, для удобства её использования необходимо скорректировать некоторые свойства:

- Caption - подпись элемента управления

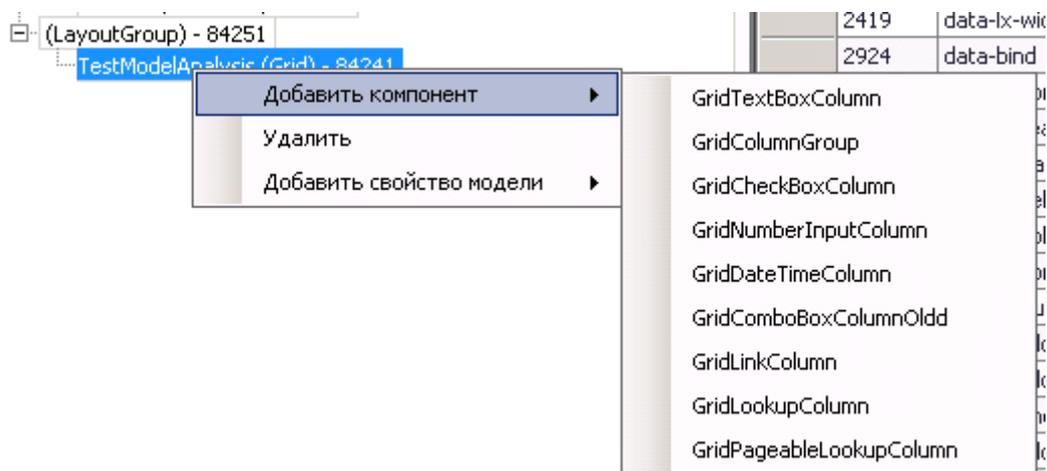
Также можно разместить элементы управления для ввода значений параметров и таблицу с результатами в разных группах:



Путём установки значения свойства Caption (заголовок) можно задать подпись группы. Например, 1-ю группу можно назвать "Параметры запроса", а 2-ю - "Результаты запроса".

Создание таблицы и настройка параметры её колонок

Корневой элемент таблицы для отображения результатов создаётся автоматически. Необходимо вручную создать и настроить её поля. Для этого необходимо нажать правой кнопкой мыши на таблицу в дереве элементов. Группа "Добавить компонент" в контекстном меню содержит несколько пунктов:



- GridTextBoxColumn - столбец для отображения данных в текстовом формате
- GridColumnGroup - группа полей. Как правило, содержит вложенные столбцы
- GridCheckBoxColumn - столбец для отображения флажков для задания значений логических полей (типа boolean)
- GridNumberInputColumn - столбец для отображения числовых данных
- GridDateTimeColumn - столбец для отображения даты и времени в различном формате
- GridLinkColumn - столбец для отображения ссылок (например, для открытия форм редактирования записи)
- GridLookupColumn - столбец, содержащий выпадающий список, позволяющий выбрать подходящее значение
- GridPagableLookupColumn - то же, что и GridLookupColumn, но с возможностью страничного вывода списка значений для уменьшения времени загрузки данных

Основными свойствами столбцов таблицы являются:

- data-lx-data-field - имя поля в модели или результатах запроса (например, VCode, Name и т.п.)
- data-lx-text - заголовок столбца
- data-lx-width - ширина поля в пикселях

Примечание: для выяснения имён полей в результатах запроса, а также для его проверки и отладки целесообразно воспользоваться программой MS SQL Server Management Studio, входящей в стандартный пакет установки сервера БД MS SQL Server:

```

SQLQuery2.sql - 192..._TestData (sa (123))* X
-- объявление переменных-параметров и присвоение им значений (для отладки запроса в SQL Studio)
declare @costFrom money = 500, @costTo money = 1300
declare @isActual bit = 1
-- примечание: при переносе текста SQL-скрипта в запрос Lexema необходимо удалить или закомментировать объявление параметров
select VCode, Name, [Description], Cost, [Weight], ProduceDate, IsActual, Classifier
from ModelTest
where (@costFrom is null or Cost >= @costFrom)
and (@costTo is null or Cost <= @costTo)
and (@isActual is null or ISNULL(IsActual, 0) = @isActual)

```

VCode	Name	Description	Cost	Weight	ProduceDate	IsActual	Classifier
1	Муфта соединительная	Применяется для соединения труб	1200,00	5	2015-11-12	1	NULL
2	Корпус металлический	NULL	800,00	2	2015-11-25	1	NULL

Пример структуры элементов аналитической формы на основе приведённого выше запроса показана на рисунке:

Свойство	Наименование	Значение
2402	RootTag	div
2404	data-lx-align	center
2414	data-lx-control-type	columnGroups
2401	ID	GridColumnGroup1
2403	data-lx-text	Количественные характеристики

Создание скрипта для отображения данных

Для обеспечения выполнения запроса необходимо создать обработчик, привязанный к определённому событию, чаще всего к нажатию кнопки на панели инструментов или на форме. В текст обработчика необходимо внести программный код для обновления таблицы результатов запроса. Для этого в дереве элементов создайте элемент типа Button (кнопка) и установите следующие параметры:

- data-lx-value - надпись на кнопке (например, "Выполнить запрос")
- data-lx-width - ширина кнопки в пикселях

Свойство	Наименование	Значение
2118	RootTag	input
2119	type	button
2123	data-lx-enabled	True
2126	data-lx-constructor	Button
2133	style-height	50px
2147	data-lx-value	Выполнить запрос
2425	data-lx-visible	True
3006	class	button
2115	ID	Button1
3894	data-lx-width	150px

Затем перейдите на вкладку "События" справа, создайте обработчик события "onclick" (нажатие), перейдите к редактированию кода обработчика, нажав на кнопку в колонке "Редактор".

Событие	Код	Наименование	Скрипт	Редактор
620	onclick	1853	btnLoadonclick	

В открывшемся окне для редактирования кода введите следующий текст:

```

var qs = new L8.Data.QuerySource('test.TestModelAnalysis'),
    costFromParam = L8.View.NumberInput1.value,
    costToParam = L8.View.NumberInput2.value,
    isActualParam = L8.View.CheckBox1.value;

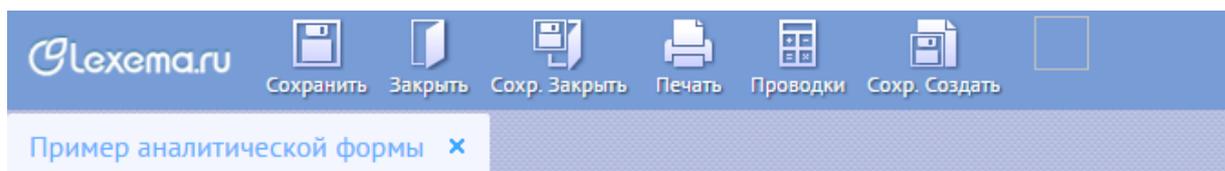
qs.load({costFrom: costFromParam, costTo: costToParam, isActual:
isActualParam}).done(function (data) {
    L8.View.Grid1.dataSource = data;
});

```

Проверка работоспособности аналитической формы

Для проверки работоспособности созданного документа необходимо открыть в браузере страницу системы и авторизоваться (ввести логин и пароль). В главном меню в области "Мои документы" следует раскрыть соответствующую категорию и нажать на пункт меню.

В открывшейся форме ввести значения параметров запроса и нажать на кнопку выполнения - в таблице отобразятся результаты запроса. Пример работы аналитической формы показан на рисунке:



Параметры запроса

Стоимость от

Стоимость до

Актуальность

Результаты запроса

Строки: 0 Выделено: 0 | Экспорт Добавить Удалить

Код	Наименование	Описание	Количественные характерист...		Дата производства
			Стоимость	Вес	
1	Муфта соеди...	Применяетс...	1 200.00	5.00	12.11.2015
2	Корпус мета...		800.00	2.00	25.11.2015

Создание отчёта на основе запроса

Отчёт является особой формой представления данных документа, предназначенной преимущественно для их вывода на печать или экспорта во внешний файл в формате XLS, PDF

и т.п. Механизм построения отчёта на основе запроса базируется на соответствующей форме, созданной на этом запросе.

Создание запроса и формы на его основе

Запрос, на котором была построена форма и будет построен отчёт, выглядит следующим образом:

*							
Виды	312	РОС					
СОСТАВ	3	ШОУВА					
МОДЕЛЬ	3	ШОУВА					
ИМЕНА	КОД	ИМЕНОВАНИЕ	ЗНАЧЕНИЕ	ИМЕНА	ИМЕНА	ИМЕНА	ИМЕНА


```
1 -- объявление переменных-параметров и присвоение им значений (для отладки запроса в SQL Studio)
2 --declare @costFrom money = 500, @costTo money = 1300
3 --declare @isActual bit = 1
4 -- примечание: при переносе текста SQL-скрипта в запрос Lexmeta необходимо удалить или закомментировать объявление параметров
5 select VCode, Name, [Description], Cost, [Weight], ProduceDate, IsActual, Classifier
6 from ModelTest
7 where (@costFrom is null or Cost >= @costFrom)
8 and (@costTo is null or Cost <= @costTo)
9 and (@isActual is null or ISNULL(IsActual, 0) = @isActual)
```

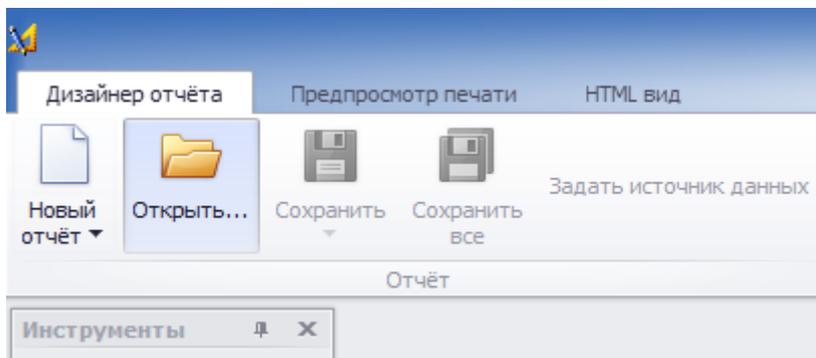
Конструирование отчёта

Запуск приложения для конструирования отчётов

Для создания отчёта требуется программа ReportDesigner (дизайнер отчётов), запуск которой требует наличие на компьютере установленной библиотеки MS .NET Framework 3.5. Программа ReportDesigner не требует установки, необходимо просто скачать архив, распаковать его и запустить файл L8.Modeller.ReportDesigner.App.exe.

Создание отчёта

Запустите дизайнер отчётов. Нажмите кнопку "Новый отчёт" на панели инструментов.

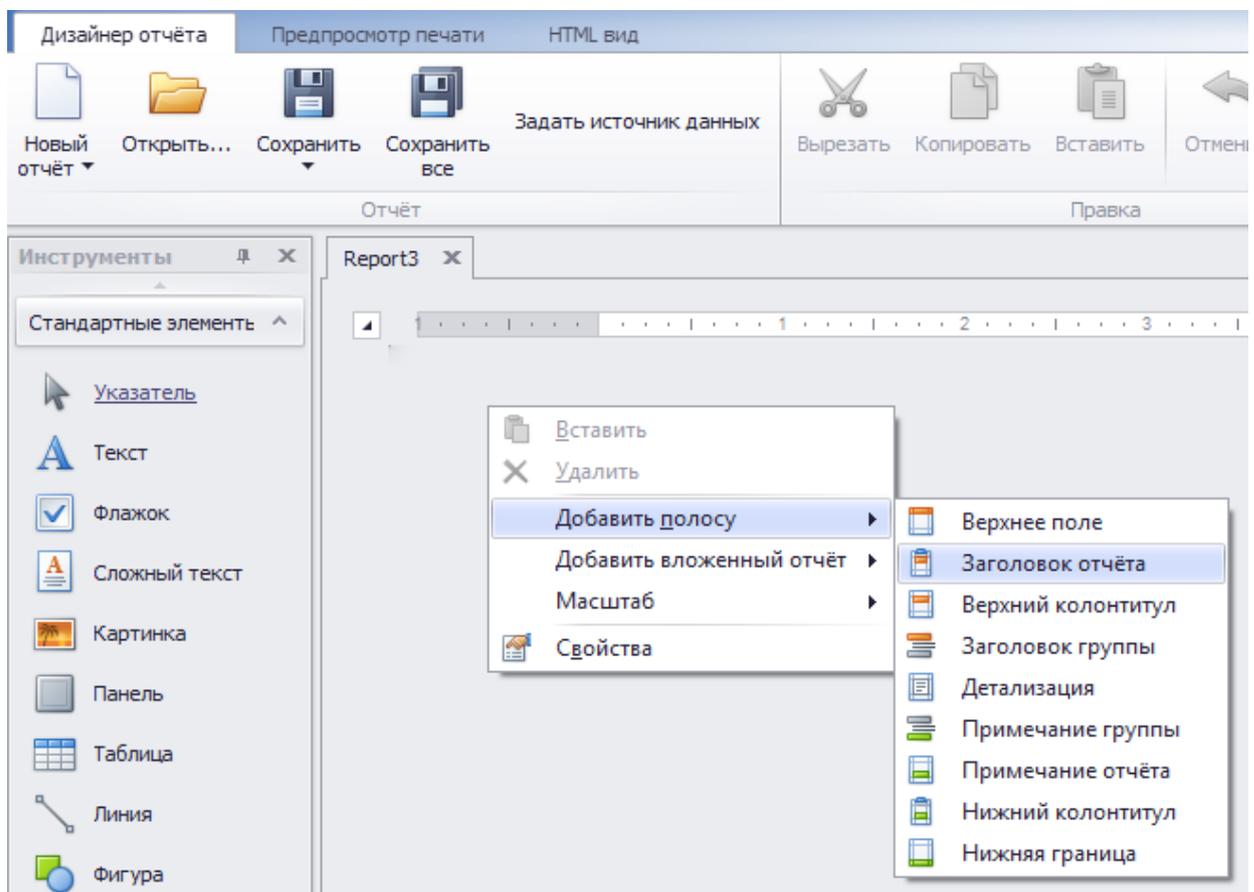


Создание областей отчёта

Пространство отчёта состоит из областей (полос) и может включать следующие элементы:

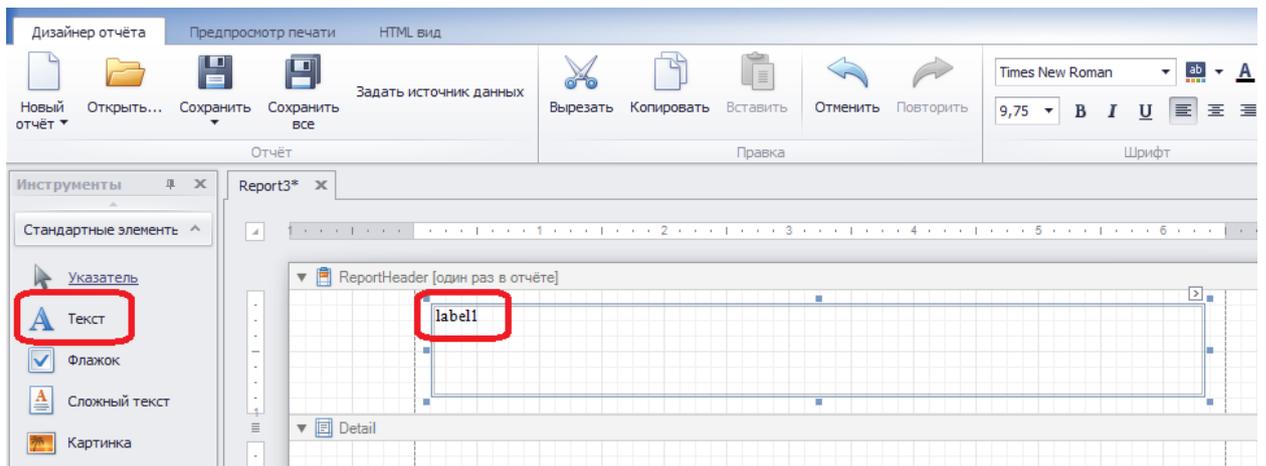
- верхнее поле
- заголовок отчёта
- верхний колонтитул
- заголовок группы
- детализация
- примечание группы
- примечание отчёта
- нижний колонтитул
- нижняя граница

С помощью контекстного меню создайте полосы "Заголовок отчёта" и "Детализация".
Примечание: контекстное меню элемента открывается при нажатии на него правой кнопкой мыши.

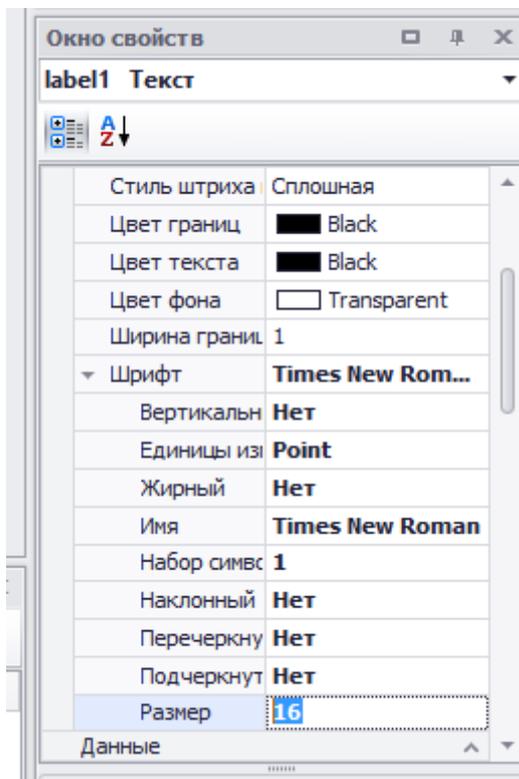


Настройка содержимого областей

На панели инструментов слева выберите элемент "Текст" (нажмите на него левой кнопкой мыши).

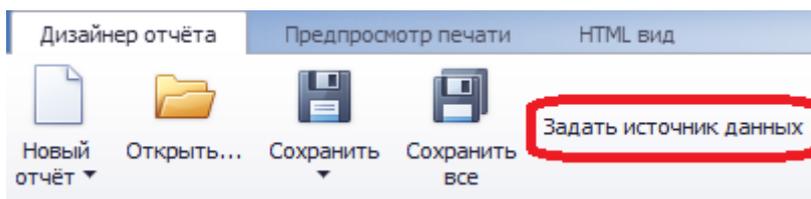


Два раза кликните мышкой на надпись label - появится возможность скорректировать текст в заголовке запроса. Для изменения размера и других параметров шрифта в окне свойств (слева внизу) в категории "Вид" раскройте группу "Шрифт", установите размер равным 16.



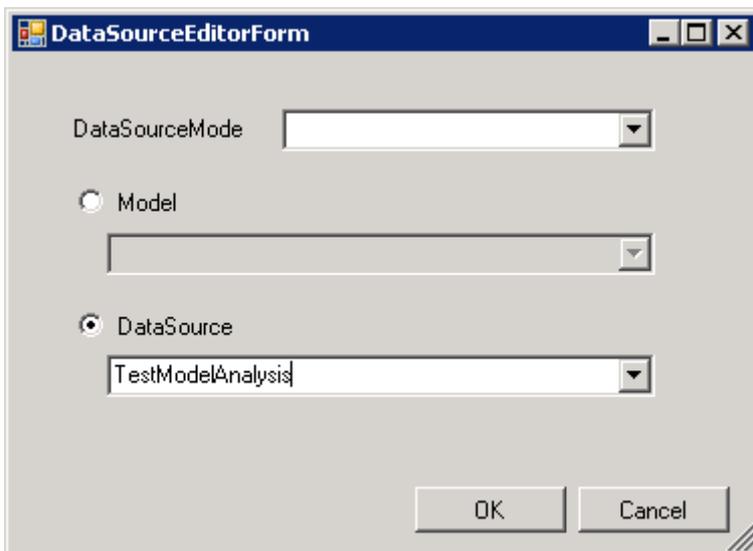
Привязка источника данных к запросу

Для получения информации из базы данных и формирования отчёта на её основе необходимо привязать его к источнику данных - запросу. Для этого необходимо нажать кнопку "Задать источник данных" на панели инструментов.

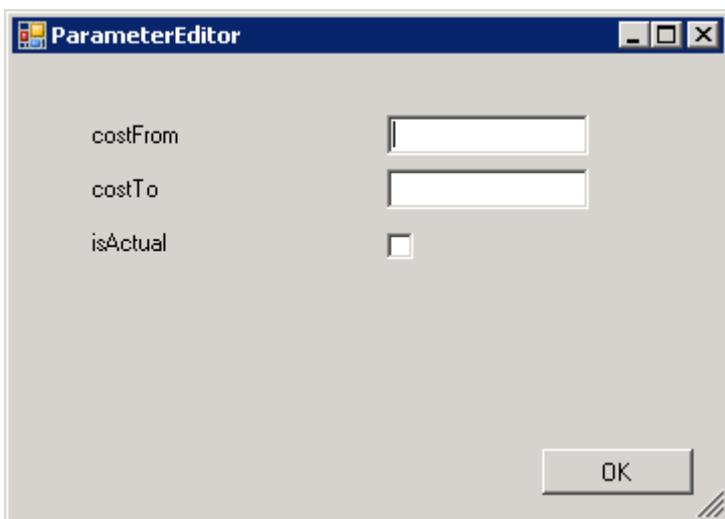


В открывшемся диалоговом окне настроек источника данных в списке "DataSourceMode" (режим источника данных) выберите пункт "ModelDocument", укажите пункт "DataSource" и выберите из

списка требуемый запрос (примечание: этот список содержит те же самые запросы, что и список запросов в Дизайнере приложения Lexema - категория "Модели", пункт "Запросы"). Нажмите кнопку "OK".

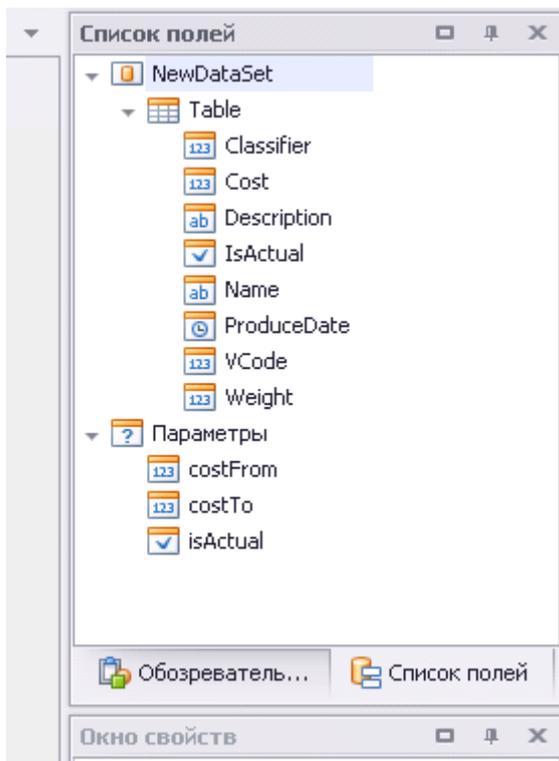


В следующем открывшемся окне требуется ввести значения по умолчанию параметров запроса. Это необходимо для загрузки демонстрационных данных в отчёт в процессе его проектирования и настройки и корректного правдоподобного отображения его вида.

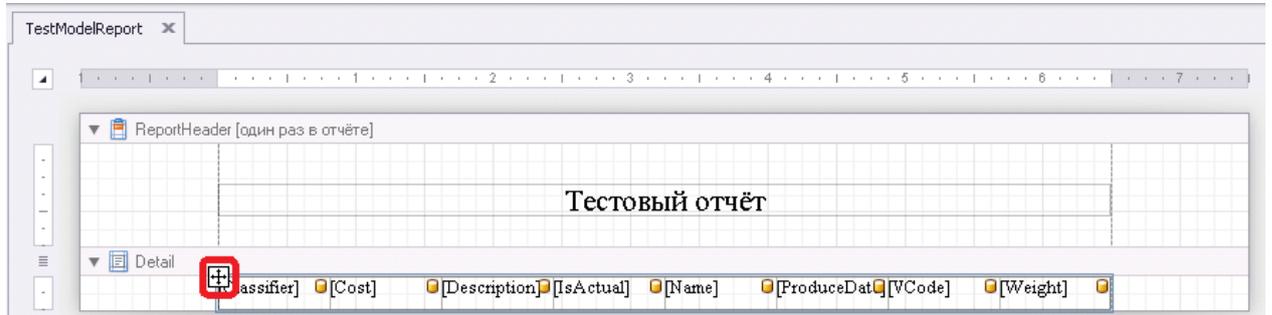


Создание таблицы данных

Если параметры запроса были заданы верно, в списке полей (справа сверху) появится источник данных (результаты запроса) со списком его полей. Эти поля используются при формировании основного содержимого (табличной части) отчёта. Для этого перетащите мышкой из окна "Список полей" объект Table на полосу Detail (детализация). Также в этом списке в ветке "Параметры" отображены параметры запроса, которые также могут быть размещены в отчёте. Значок каждого поля или параметра отображает тип данных. Например, поле Cost является числовым, Name - строковым, IsActual - логическим, а ProduceDate - датой:

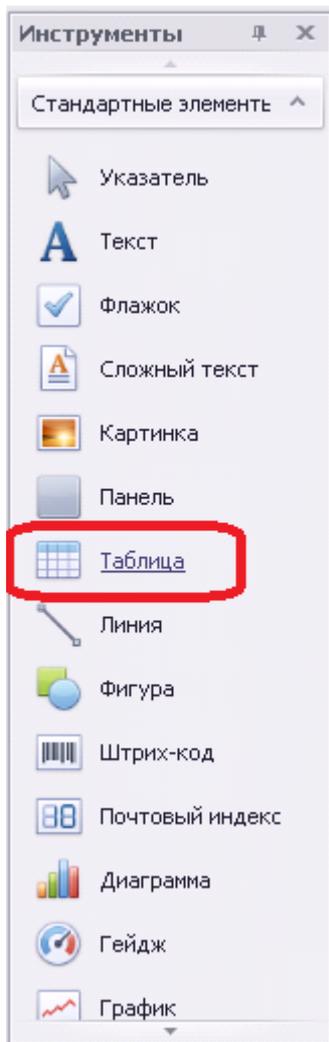


В области отчёта появятся заголовки полей результатов запроса. С помощью мышки уменьшите высоту полосы Detail до границ строки таблицы - это улучшит внешний вид таблицы с данными в режиме просмотра отчёта. В противном случае при просмотре отчёта между строками детализации будет лишнее пустое пространство.

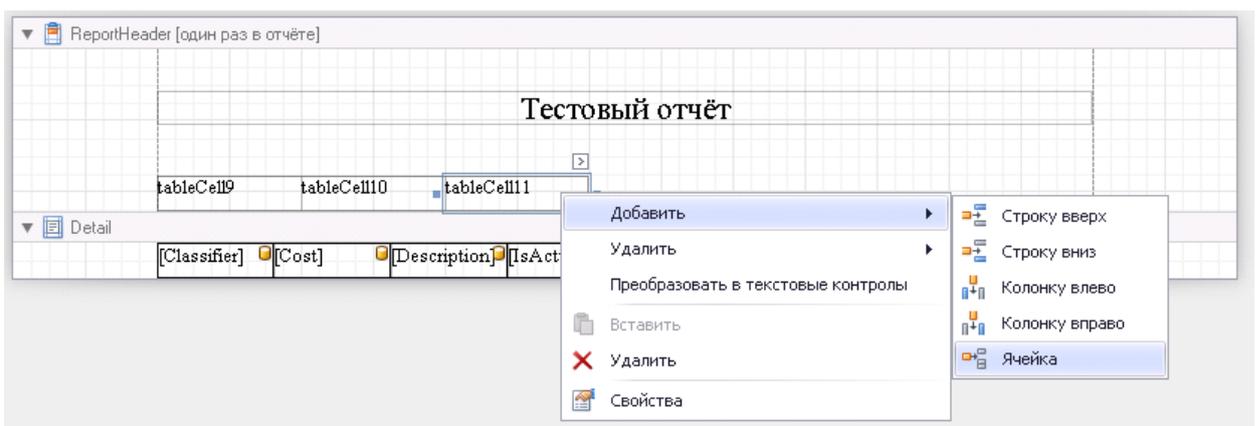


Создание "шапки" таблицы

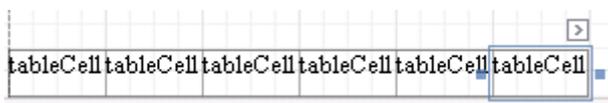
Для понимания структуры данных таблицы необходимо создать заголовок таблицы - "шапку". Её необходимо разместить в области "Заголовок отчёта", т.к. если её разместить в области "Детализация" она будет продублирована при отображении отчёта. Найдите элемент "Таблица" на палитре компонентов и перетащите её в нужную область отчёта:



В результате будет создана таблица, содержащая три колонки. Если количество столбцов в таблице данных больше, необходимо добавить новые ячейки. Для этого нажмите правой кнопкой мыши на таблице, в контекстном меню выберите пункт "Добавить" - "Ячейка":



Для изменения размеров ячеек необходимо выделить её и перетащить мышью боковой маркер влево или вправо:

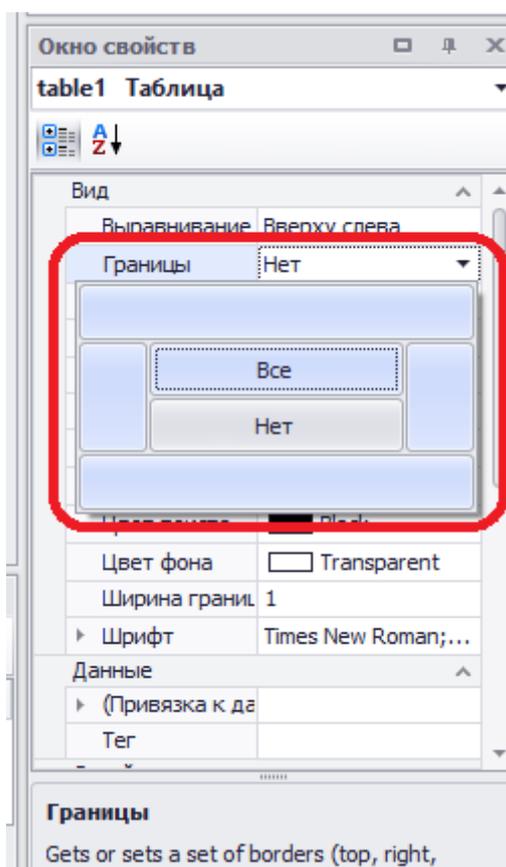


Для перехода в режим редактирования текста в колонке необходимо кликнуть мышкой по ней два раза.

Текст в заголовке таблицы рекомендуется выделить жирным шрифтом. Для этого необходимо выделить всю таблицу с помощью "перекрестья" и нажать кнопку "b" (жирный) на панели инструментов.

Настройка границ таблицы

По умолчанию в таблице не отображаются границы. Для изменения параметров границ наведите указатель мыши на таблицу - появится значок "перекрестья" слева вверху, нажмите на него для выделения всей таблицы. Затем в окне свойств (справа внизу) для пункта "Границы" выберите из выпадающего списка кнопку "Все":

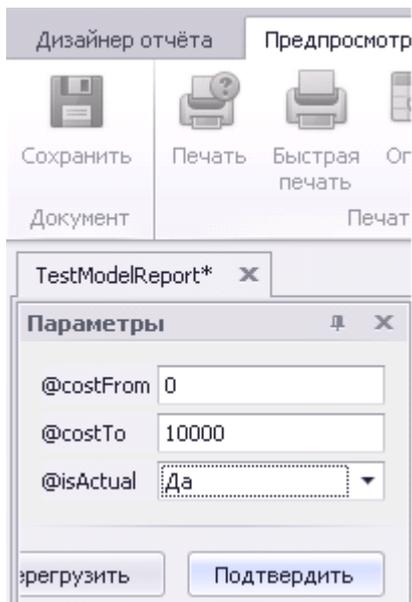


После установки значения необходимо перейти в любую другую строку таблицы свойств для применения выбранных параметров.

Примечание: для корректного отображения таблица "шапки" должна содержать все границы, а таблица детализации - все, кроме верхней.

Предварительный просмотр отчёта

После создания табличной части (детализации) отчёта возможен предварительный просмотр отчёта. Для этого перейдите на вкладку "Предпросмотр печати" (вверху слева). Для построения отчёта необходимо задать значения его параметров и нажать кнопку "Подтвердить":



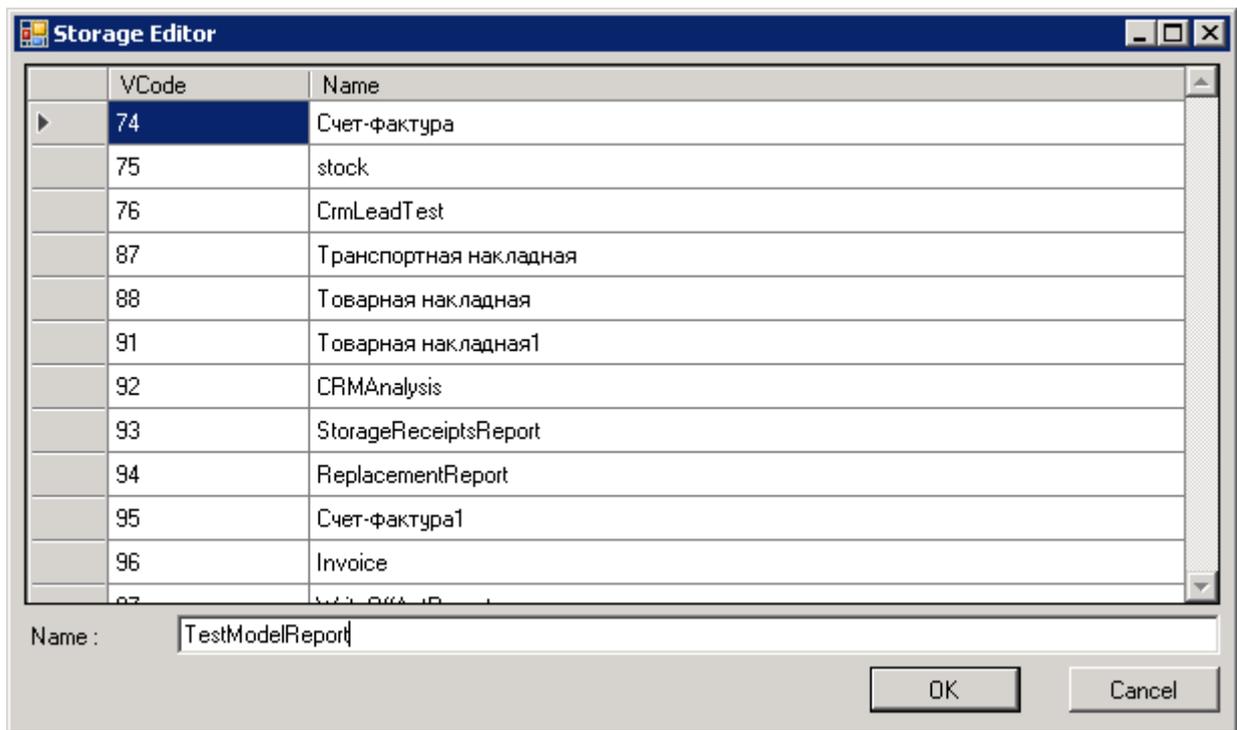
Отчёт в режиме предварительного просмотра содержит результаты запроса, выполненного с заданными значениями параметров:

Тестовый отчёт							
Тип	Цена	Описание	Актуальн.	Название	Дата произв.	Код	Вес
	1200,0000	Применяется для соединения труб	True	Муфта соединительная	12.11.2015 0:00:00	1	5
	800,0000		True	Корпус металлический	25.11.2015 0:00:00	2	2
2	123,0000	текст	True	Тест123	04.02.2016 0:00:00	5	45

Вернитесь в режим "Дизайнер отчёта" путём перехода на соответствующую вкладку. Сохраните отчёт.

Сохранение отчёта

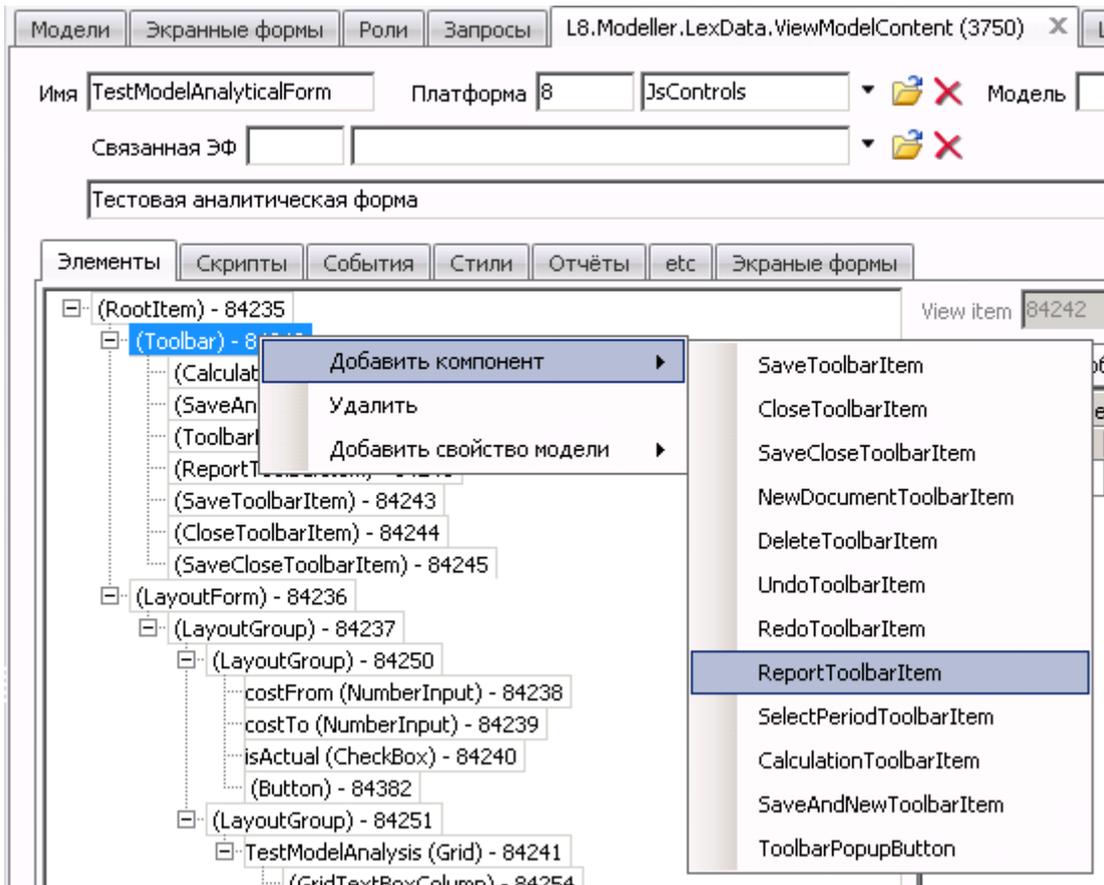
Нажмите кнопку "Сохранить" на панели инструментов. Задайте наименование отчёта. Примечание: отчёты хранятся в базе данных.



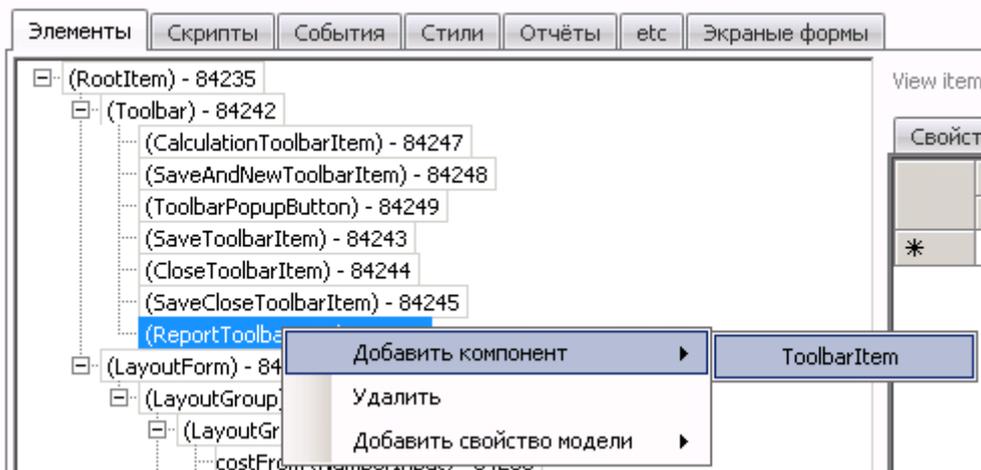
Привязка отчёта к форме

Как правило, открытие отчёта производится из экранной формы или реестра при нажатии на кнопку на панели инструментов или на самой форме.

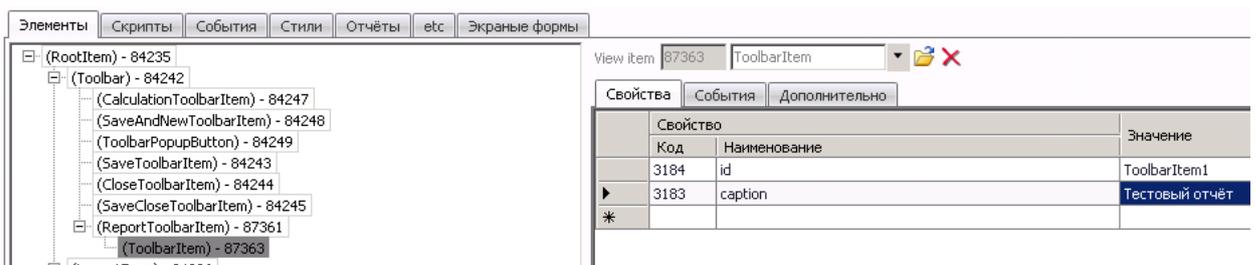
Откройте для редактирования аналитическую форму на основе запроса, к которому был также привязан отчёт. Если кнопка "ReportToolBarItem" не существует, создайте её с помощью пункта "Добавить компонент" контекстного меню элемента Toolbar (панель инструментов).



Если требуется отображать несколько отчётов, можно создать подменю: Для этого необходимо выбрать пункт "Добавить компонент" - ToolStripItem в контекстном меню элемента ReportToolBarItem:



Установите значение свойства "caption" созданного элемента меню "ToolStripItem":



Свойство caption может быть задано также у кнопки ReportToolBarItem.

Создание скрипта открытия отчёта

Выберите кнопку, при нажатии на которую должен открываться отчёт (ReportToolBarItem или ToolBarItem). Перейдите на вкладку "События", создайте обработчик события - выберите в столбце "Наименование" пункт "ItemClick", нажмите кнопку "Редактор" для перехода в окно для ввода исходного кода скрипта обработчика.

Отчёт может быть отображён в двух режимах:

- в режиме предпросмотра DevExpress - это позволяет экспортировать результаты в формате DOC, XLS и т.п., а также использовать другие возможности отчётной системы DevExpress. Однако, при нажатии на кнопку "Печать" будет отображён интерфейс предпросмотра браузера
- в режиме предпросмотра браузера - данный режим упрощает печать документа, но не позволяет использовать возможности DevExpress

Для открытия отчёта в режиме необходимо использовать следующий код:

```
// считывание значений параметров из полей формы
var costFromParam = L8.View.NumberInput1.value,
    costToParam = L8.View.NumberInput2.value,
    isActualParam = L8.View.CheckBox1.value;

// открытие отчёта
L8.System.openReport({name: 'TestModelReport',
    parameters: [{costFrom: costFromParam, costTo: costToParam, isActual:
isActualParam}]});
```

Для открытия отчёта без предпросмотра (точнее, только с предпросмотром браузера) необходимо в приведённом выше коде заменить метод **openReport** на **printWithoutPreview**. Всё остальное остаётся без изменений.

Проверка работоспособности отчёта

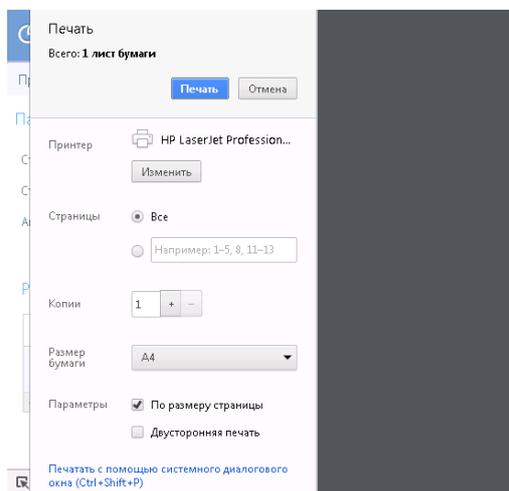
Откройте тестовую аналитическую форму на запросе, в которую была добавлена кнопка для открытия отчёта. Введите значения параметров, нажмите кнопку "Отчёт". Если был использован обычный режим с предпросмотром, отчёт откроется в отдельной вкладке:

Страница 1 из 1 Pdf

Тестовый отчёт

Тип	Цена	Описание	Актуальн.	Название	Дата произв.	Код	Вес
	1200,0000	Применяется для соединения труб	True	Муфта соединительная	12.11.2015 0:00:00	1	5
	800,0000		True	Корпус металлический	25.11.2015 0:00:00	2	2
2	123,0000	текст	True	Тест123	04.02.2016 0:00:00	5	45

В случае открытия отчёта без предпросмотра он откроется в той же вкладке:



Тестовый отчёт

Тип	Цена	Описание	Актуальн.	Название	Дата произв.	Код	Вес
	1200,0000	Применяется для соединения труб	True	Муфта соединительная	12.11.2015 0:00:00	1	5
	800,0000		True	Корпус металлический	25.11.2015 0:00:00	2	2
2	123,0000	текст	True	Тест123	04.02.2016 0:00:00	5	45

Работа с вычислениями в экранной форме

Вычислениями в системе Lexema.ru называются программные сценарии (скрипты) на языке JavaScript, хранимые в БД бизнес-логики системы. Как правило, программные сценарии выполняются автоматически при возникновении событий (events), например, в ответ на действия пользователя. Одни фрагменты кода (функции) могут быть вызваны из других фрагментов. Таким образом, могут быть созданы библиотеки прикладного программного кода.

События, при которых могут быть выполнены программные сценарии:

- на уровне модели - при изменении значений полей
- на уровне экранной формы (ЭФ) и её элементов:
 - открытие ЭФ (Created)
 - получение сообщения экранной формой (GetMessage)
 - попытка закрытия ЭФ (Closing)
 - кнопка на панели инструментов (ToolbarItem) - событие при нажатии (ItemClick)
 - текстовое поле:
 - изменение значения (onchange)

- получение фокуса (onfocus) - перевод курсора в поле с помощью мыши или клавиатуры
- потеря фокуса (onblur)
- нажатие клавиши на клавиатуре при редактировании поля (onkeydown)
- отпускание клавиши на клавиатуре при редактировании поля (onkeyup)
- onkeypress - обработчик события срабатывает, когда при редактировании поля клавиша на клавиатуре была нажата и отпущена, что эквивалентно совместному действию onkeydown и onkeyup
- флажок (checkbox) - изменение состояния (onchange)
- кнопка (button) - нажатие (onclick)

Создание обработчика открытия ЭФ (Created)

Обработчик Created автоматически запускается каждый раз при открытии экранной формы. Примеры использования данного обработчика приведены в списке ниже:

- инициализация значений полей формы (присвоения им начальных значений)
- блокировка некоторых элементов ЭФ (для запрета редактирования) в зависимости от каких-либо условий
- загрузка дополнительных данных из БД, которые не загружаются автоматически в полях модели (например, для выполнения запросов с помощью QuerySource)
- генерация реквизитов документа (например, номера договора и т.п.) при его создании или регистрации
- и т.п.

Для создания обработчика открытия ЭФ необходимо открыть её для редактирования, перейти на вкладку события, в поле "Тип события" выбрать Created, нажать кнопку в колонке Редактор.

The screenshot shows the configuration window for the 'Created' event handler. The 'Имя' (Name) is 'RequestTestForm', 'Платформа' (Platform) is '8', and 'JsControls' is selected. The 'Связанная ЭФ' (Linked Form) is '3774' and 'RequestTestRegistry'. Below this, a table lists the event handlers:

	Тип события		Скрипт		Редактор
	Код	Наименование	Код	Наименование	
▶	1	Created			
*					

В результате будет открыто окно для редактирования исходного кода сценария. Пример скрипта, выполняемого при открытии формы приведён ниже:

The screenshot shows the script editor for the 'Created' event handler. The script is as follows:

```

1 // если документ новый, то генерируем его номер
2 if (L8.ActionFlags.isNew) {
3     // создание объекта QuerySource для работы с запросами
4     var qs = new L8.Data.QuerySource('Base.GenCode');
5     // выполнение запроса
6     return qs.load({keyName: 'RequestTestDocNumber'}).done(function (data) {
7         // если запрос вернул данные
8         if (data && data[0]) {
9             // записываем номер в поле
10            L8.Data.root.Number(data[0].Code);
11            // открываем поле для редактирования
12            L8.View.tbNumber.enabled = true;
13        }
14    });
15 } else {
16     // запрещаем редактирование поля
17     L8.View.tbNumber.enabled = false;
18 }

```

Исходный код в текстовом виде:

```
// если документ новый, то генерируем его номер
if (L8.ActionFlags.isNew) {
    // создание объекта QuerySource для работы с запросами
    var qs = new L8.Data.QuerySource('Base.GenCode');
    // выполнение запроса
    return qs.load({keyName: 'RequestTestDocNumber'}).done(function (data) {
        // если запрос вернул данные
        if (data && data[0]) {
            // записываем номер в поле
            L8.Data.root.Number(data[0].Code);
            // открываем поле для редактирования
            L8.View.tbNumber.enabled = true;
        }
    });
} else {
    // запрещаем редактирование поля
    L8.View.tbNumber.enabled = false;
}
```

Внимание! После добавления обработчиков событий ЭФ необходимо сохранить как сами скрипты, так и всю экранную форму.

Подробнее событие Created описано в разделе статьи [События экранной формы](#).

Создание обработчика получения экранной формой сообщения (GetMessage)

Обработчик события получения экранной формой сообщения от системы сообщений создаётся способом, аналогичным созданию обработчика Created, за исключением того, что в поле "Тип события" необходимо выбрать значение GetMessage.

Создание обработчика закрытия ЭФ (Closing)

Обработчик Closing автоматически вызывается каждый раз перед закрытием экранной формы. Примеры использования данного обработчика приведены в списке ниже:

- валидация полей формы (проверка корректности введённых значений)

Обработчик события закрытия ЭФ создаётся способом, аналогичным созданию обработчика Created, за исключением того, что в поле "Тип события" необходимо выбрать значение Closing.

Пример скрипта, выполняемого при закрытии формы приведён ниже:

```
var isClosingCanceled = false,
    errorMsg = '';

if (!L8.Data.root.Sum()) {
    errorMsg += 'Не заполнена сумма\n';
    isClosingCanceled = true;
}
```

```

if (!L8.Data.root.Count()) {
    errorMsg += 'Не заполнено количество\n';
    isClosingCanceled = true;
}

if (errorMsg) {
    alert('При редактировании документа допущены ошибки: \n' + errorMsg);
}

return isClosingCanceled;

```

Создание обработчиков событий для элементов ЭФ

Обращение к свойствам элементов формы в скриптах

Для обращения к свойствам элементов управления экранной формы из программных сценариев необходимо использовать объект `L8.View` и указать имя контрола, которое задаётся в дизайнера приложения с помощью свойства `id`. При установке данного свойства следует придерживаться правил именования объектов. Например, если текстовому полю для ввода номера документа назначен идентификатор `tbNumber`, то для обращения к данному контролу необходимо использовать конструкцию `L8.View.tbNumber`, а для доступа к его свойству требуется указать его название. Например, для считывания и изменения доступности поля для редактирования необходимо указать:

```

// считывание значения свойства поля
if (L8.View.tbNumber.readOnly) { ... }

// установка значения свойства поля
L8.View.tbNumber.readOnly = true;

```

Полный перечень свойств элементов управления перечислен в статьях, посвящённых соответствующим контролам (см. категорию [Контролы](#)).

Элементы управления экранных форм

Экранные формы (ЭФ) являются компонентом `Lexema.ru`, с которым непосредственно работают конечные пользователи системы. Экранная форма состоит из множества различных элементов управления (контролов, виджетов, компонентов), имеющих разное назначение и выполняющих разные функции. ЭФ имеет древовидную иерархическую структуру, некоторые её элементы являются контейнерами и содержат дочерние элементы (например, группа `LayoutGroup`, "аккордеон" `Expander`, панель вкладок `Tabs`).

В дерево элементов экранной формы могут быть добавлены следующие компоненты:

- `RootItem` - корневой элемент экранной формы
- `Toolbar` - панель инструментов
 - кнопки панели инструментов
 - `SaveToolbarItem` - сохранение текущего документа

- CloseToolBarItem - закрытие текущего документа
- SaveCloseToolBarItem - сохранение и закрытие текущего документа
- NewDocumentToolBarItem - создание нового документа
- DeleteToolBarItem - удаление документа
- UndoToolBarItem - отмена последнего действия
- RedoToolBarItem - повторение действия
- ReportToolBarItem - вывод отчёта
- SelectPeriodToolBarItem - выбор расчётного периода
- CalculationToolBarItem - произвольные вычисления
- SaveAndNewToolBarItem - сохранение текущего документа и создание нового
- ToolbarPopupButton - кнопка для отображения встроенной экранной формы во фрейме
- LayoutForm - контейнер для всех основных визуальных компонентов формы (за исключением панели инструментов)
- BootstrapLayout - контейнер для разметки формы с использованием адаптивных ("резиновых") групп (BootstrapLayoutGroup) набора инструментов Twitter Bootstrap
- PopupContainer - контейнер для отображения вложенных экранных форм в отдельном фрейме
- Gis -
- LayoutGroup - контейнер для группировки логически связанных элементов управления
- BootstrapLayoutGroup - аналог LayoutGroup, позволяющий создавать группы элементов, адаптирующиеся к ширине страницы ("резиновый" дизайн)
- Tabs - контейнер, позволяющий создавать несколько групп элементов, переключение между которыми осуществляется с помощью вкладок
- Expander - контейнер, позволяющий сворачивать и разворачивать группу элементов управления ("аккордеон")
- break -
- widget -
- HtmlBox - блок, позволяющий вводить произвольный HTML-код
- NavigationBar - контейнер для нескольких групп элементов, которые можно разворачивать и сворачивать
- JqxMenu -
- DragDrop -
- DocumentsMenu -
- Map -
- Grid - таблица, позволяющая редактировать содержимое ячеек
- AnalyticalGrid - таблица, предназначенная только для отображения значений
- Pivot - элемент для отображения сводных таблиц
- Tree -
- TextBox - поле для ввода текста общего назначения
- NumberInput - поле для ввода числовых значений (целых и дробных)
- DateTimeInput - поле для выбора даты
- MaskedInput - маскированное поле для ввода
- Input -
- Memo - многострочное поле для ввода текста
- AutoCompleteAddressTextBox -
- CheckBox - флажок (переключатель)
- JqxRadioButton -
- Button - кнопка
- GlyphButton - кнопка с изображением
- JqxRepeatButton -
- JqxToggleButton -
- Link - гипер-ссылка
- Image - изображение из БД

- CustomImage -
- FileUpload - элемент для загрузки файлов
- Chart - элемент для отображения графиков и диаграмм
- Rating -
- Lookup - выпадающий список
- DropDownList -
- PagableLookup - выпадающий список с возможностью постраничного отображения

RootItem

Корневой элемент экранной формы (ЭФ), является родительским контейнером для всех дочерних элементов, не визуальный. Свойства:

- id - уникальный идентификатор блока формы (тега div)
- title - заголовок ЭФ, отображается в заголовке окна браузера, а так же в качестве названия вкладки страницы Lexema.ru

В RootItem могут быть добавлены следующие дочерние элементы:

- LayoutForm
- BootstrapLayout
- Toolbar
- PopupContainer
- Gis

Toolbar

Панель инструментов, служит для отображения кнопок. Существует ряд predefined кнопок, которые могут быть добавлены на панель, каждая кнопка имеет специфический значок и предназначена для выполнения определённой функции. При автоматической генерации элементов при создании формы некоторые кнопки панели инструментов создаются автоматически, также для них генерируются обработчики событий с соответствующим программным кодом. Исключение составляет кнопка CalculationToolbarItem, предназначения для выполнения произвольных программных сценариев-вычислений, которые создаются прикладным разработчиком.

SaveToolbarItem

Кнопка для сохранения документа. Обработчик события ItemClick, как правило, содержит следующий код:

```
L8.System.SaveDocument ();
```

Данный код может быть дополнен, к примеру, проверкой корректности данных, введённых в поля формы.

CloseToolbarItem

Кнопка для закрытия документа. Обработчик события ItemClick, как правило, содержит следующий код:

```
L8.System.CloseDocument ();
```

SaveCloseToolbarItem

Кнопка для сохранения и закрытия документа. Обработчик события ItemClick, как правило, содержит следующий код:

```
L8.System.SaveAndCloseDocument ();
```

NewDocumentToolBarItem

Кнопка для создания нового документа. В большинстве случаев используется в реестре документов. Обработчик события `ItemClick`, как правило, содержит следующий код:

```
L8.System.CreateDocument (L8.System.LinkedViewModel);
```

`L8.System.LinkedViewModel` - это наименование связанной экранной формы, кот. вводится в соответствующем поле ЭФ.

DeleteToolBarItem

Кнопка для удаления одного или нескольких выделенных документов. В большинстве случаев используется в реестре документов. Обработчик события `ItemClick`, как правило, содержит следующий код:

```
L8.System.DeleteRegistryDocument ();
```

UndoToolBarItem

Кнопка для отмены последнего действия при редактировании документа.

RedoToolBarItem

Кнопка для повтора последнего действия при редактировании документа.

ReportToolBarItem

Кнопка для вывода отчёта, может применяться как в документах, так и в реестрах. Код обработчика нажатия на эту кнопку пишется прикладным разработчиком вручную. Пример кода приведён ниже:

```
// считывание значений параметров из полей формы
var costFromParam = L8.View.NumberInput1.value,
    costToParam = L8.View.NumberInput2.value,
    isActualParam = L8.View.CheckBox1.value;

// открытие отчёта
L8.System.openReport ({name: 'TestModelReport',
    parameters: [{costFrom: costFromParam, costTo: costToParam, isActual:
isActualParam}]});
```

Подробнее процесс создания кода открытия отчёта описан в соответствующем разделе статьи "Создание отчёта на основе запроса".

SelectPeriodToolBarItem

Кнопка для отображения диалога вывода расчётного периода. В большинстве случаев используется в реестре документов. Обработчик события `ItemClick`, как правило, содержит следующий код:

```
L8.System.selectDatePeriod.openDialog ();
```

CalculationToolbarItem

Кнопка для запуска произвольных вычислений. Соответствующий JS-код должен быть записан в обработчике нажатия на кнопку (ItemClick).

SaveAndNewToolbarItem

Кнопка для сохранения текущего и создания нового документа. Обработчик события ItemClick, как правило, содержит следующий код:

```
L8.System.SaveAndNewDocument ();
```

ToolbarPopupButton

Кнопка для отображения/скрытия всплывающего фрейма, содержащего элементы управления другой формы. Для корректной работы данного механизма необходимо наличие в форме элементов PopupContainer, PopupWindow и NestedForm.

Создание обработчиков изменения полей модели

У разработчика прикладных конфигураций Lexema.ru есть возможность создания программных сценариев на языке JavaScript, привязанных к полям модели, которые будут вызываться автоматически всякий раз при изменении значений этих полей вне зависимости от экранной формы.

Создание обработчика

Для создания обработчика необходимо открыть модель для редактирования (дерево навигации - раздел "Модели" - пункт "Модели"). Затем необходимо нажать на кнопку в колонке "Редактировать" в строке соответствующего поля:

Name	DbName	ViewName	Description	Тип	Код	Наименование	Value changed script	Редактиров	Удалить скрипт	GlobalId
VCode	VCode	VCode		127	bigint					8CC4FD27-CFC...
CDate	CDate	CDate		461	DateTimeOffset					4720DB19-A2FC...
CHost	CHost	CHost		73	string					6DF8B2AF-0230...
CUser	CUser	CUser		73	string					5A43DEE2-8B4D...
WDate	WDate	WDate		461	DateTimeOffset					D3F07839-163C...
WHost	WHost	WHost		73	string					80585FC6-05F9...
WUser	WUser	WUser		73	string					83CD4B92-1D0A...
COrg	COrg	COrg		1	int					33E62C06-2E7C...
WOrg	WOrg	WOrg		1	int					920FC492-0549...
MatCode	MatCode			127	bigint					FDD658D0-CB9...
Color	Color			1	int					8B123F55-E3F6...
Name	Name			73	string					77085045-828D...
SumNDS	SumNDS			3	money					A1E47485-E469...
SumbNDS	SumbNDS			3	money					CB81BED3-6436...
Count	Count			34	double					0E7793FB-DASD...
AccountPrice	AccountPrice			3	money					F3883152-0970...
PCode	PCode			127	bigint					85AB84F8-F19E...

В редакторе исходного кода обработчика необходимо текст скрипта. К примеру, сценарий на изменение поля Count (количество) вычисляет значение суммы путём умножения цены на количество:

Модели	L8.Modeller.LexData.ModelContent (1434)	Count (9357) X
Настройки		
Имя	Count	GlobalId D07995F8-BFDB-43A3-BF5A-0F856A58A750
Заметка	RequestTestDetail.Count	Guid 4A4A0A9D-0507-4F0B-95D7-D2737FC81A14
Функция	<input checked="" type="checkbox"/> Параметры newValue, oldValue, root	Приоритет 1000 LL <input checked="" type="checkbox"/> Тип Script
Namespace	129 L8.System.DataEvents.RequestTestDetail	
Текст Ссылки Модели		
<pre> 1 var price = this.AccountPrice(), 2 count = this.Count(); 3 4 5 this.SumbNDS(price * count); </pre>		

```

var price = this.AccountPrice(),
    count = this.Count();

this.SumbNDS(price * count);

```

Обращение к полям модели

Как правило, программный код обработчиков изменения поля модели содержит получение и установку значений других полей данной модели. Для обращения к ним следует использовать ссылку `this`, указывающую на экземпляр модели (запись таблицы).

Для получения значения поля `fieldName` необходимо написать `this.fieldName()`. Например:

```
var price = this.AccountPrice();
```

Для установки значения `newValue` поля `fieldName` необходимо написать `this.fieldName(newValue)`. Например:

```
this.SumbNDS(price * count);
```

Возможные проблемы

Циклический вызов обработчиков изменения полей модели может вызвать зависание страницы. Например, если на поле `field1` создан обработчик `this.field2(newValue)`, а на поле `field2` - обработчик `this.field1(newValue)`, то это приведёт к рекурсивным вызовам бесконечной глубины и, как следствие, к зависанию клиентской части (веб-страницы в браузере). Такой конфигурации обработчиков следует избегать.

Проверка работоспособности скрипта

Для проверки работы скрипта обработчика изменения поля модели необходимо открыть любую экранную форму, созданную на основе этой модели и изменить значение поля, которому был назначен обработчик.

В примере, описанном выше, обработчик был назначен полю `Count` (количество) в табличной части документа `RequestTest` (заявка на поставку). Любое изменение значения поля `Count` приводит к автоматическому пересчёту значения поля `SumbNDS` (сумма с НДС):

Заявка на поставку X

192.168.0.33:5555/Screen.aspx?ViewName=RequestTestForm&VCode=1

Lexema.ru Сохранить Закр. Сохр. Закр. Печать Проводки Сохр. Создать

Заявка на поставку x Заявка на поставку - Реестр от 01.12.2015 - 31.12.2015 x

COrg 0
 WOrg 0
 Number 25-2/2
 RDate 02.12.2015
 IsActual
 Sum 492.00
 Count 1 234.55
 Контрагент
 RType
 Вычислить

Табличная часть

detail										Строки: 2	Выделено: 1	В фильтре: 0	Экспорт	Добавить	Удалить
WUser	COrg	WOrg	MatCode	Color	Name	SumNDS	SumbNDS	Count	AccountPrice						
airat	0	0	123	3	fgdfgh	888.00	234.00	2	444.00						
airat	0	0	55	66	ertyerty	1 665.00	1 668.00	3.00	555.00						
							Сумма	Сумма	Сумма						
							Всего: 1902	Всего: 5	Всего: 999						
							Выделено: 234	Выделено: 2	Выделено: 444						

Использование событий модели

Средства прикладной разработки системы Lexema.ru предоставляют возможность для каждой модели создавать SQL-скрипты, выполняемые автоматически при выполнении запросов к записям модели. Существует 4 типа запросов, на которые могут быть назначены обработчики:

- Insert - вставка новых записей
- Update - изменение существующих записей
- Delete - удаление записей
- Select - выборка записей

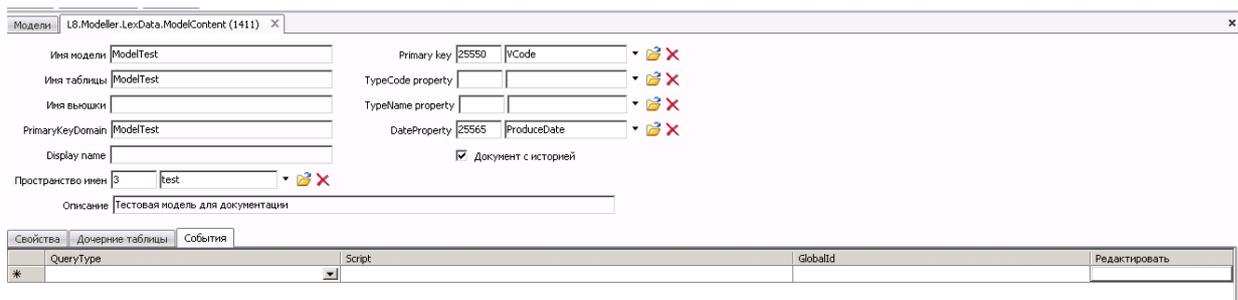
Скрипты-обработчики событий модели могут применяться для решения задач разного рода:

- проверка корректности значений полей записи (валидация)
- генерация номера документа при создании новой записи (на Insert)
- журналирование (логгирование)
- профилирование производительности
- разделение прав доступа
- и т.п.

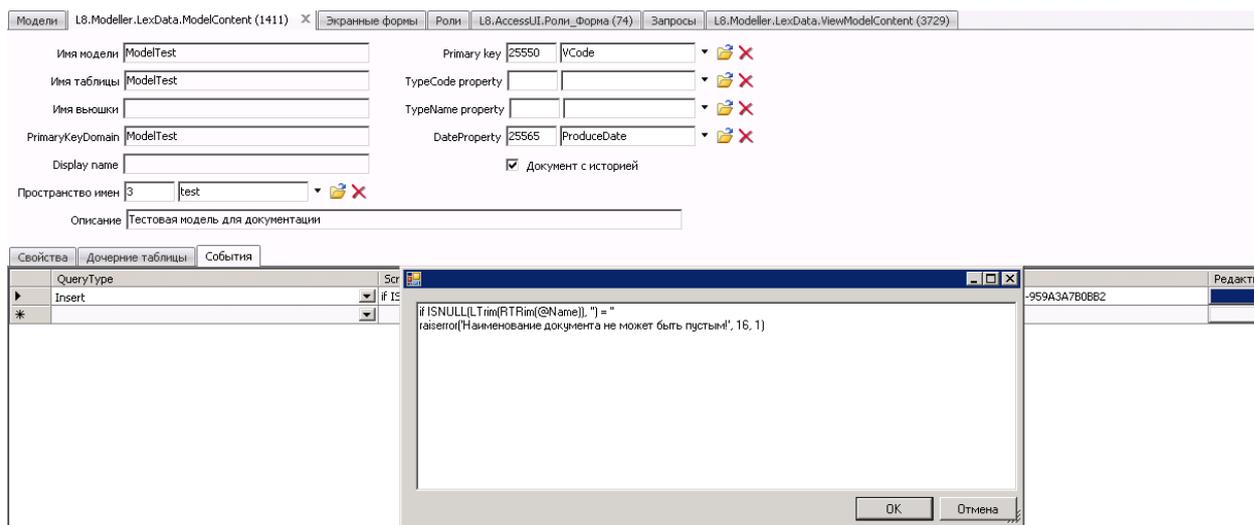
Создание обработчиков событий модели

В дереве навигации слева раскройте группу Модели, выберите пункт Модели. В списке найдите нужную модель, откройте её для редактирования двойным щелчком левой кнопки мыши по строке или с помощью кнопки "Открыть на изменение" на панели инструментов на вкладке "Документ" слева вверху.

В форме редактирования модели перейдите на вкладку "События":



В списке в поле QueryType выберите необходимый тип запроса. Для редактирования текста SQL-скрипта нажмите на кнопку в поле "Редактировать". В открывшемся окне введите код обработчика события. Например:



Размер окна для редактирования кода может быть увеличен путём перетаскивания правого нижнего угла окна за специальный маркер - это даст возможность видеть большее кол-во строк без прокрутки, что актуально при редактировании больших скриптов.

Логика работы обработчиков

Обработчики событий модели выполняются автоматически при выполнении соответствующих запросов. Если требуется запретить выполнение текущей операции, необходимо сгенерировать сообщение об ошибке с помощью системной инструкции Raiserror. Без её вызова текущая операция будет выполнена.

Обращение к значениям полей

При выполнении обработчика события модели для всех её полей автоматически создаются локальные переменные, содержащие значения полей текущей записи. Название переменных совпадают с именами полей. Перед именем ставится знак "@". Например, если поле называется DocumentNumber, то имя соответствующей локальной переменной будет @DocumentNumber.

Например, для запрета сохранения документа, в котором не заполнено обязательное поле Name, можно использовать SQL-конструкцию вида:

```
if ISNULL(LTrim(RTrim(@Name)), '') = ''
raiserror('Наименование документа не может быть пустым!', 16, 1)
```

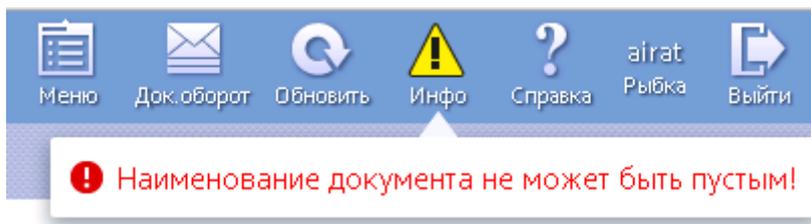
Примечание: для проверки значения на пустоту следует использовать функцию ISNULL.

Вывод сообщения об ошибке

Для вывода сообщения об ошибке и прерывания текущей операции необходимо вызвать системную инструкцию СУБД MS SQL Server - Raiserror со следующими параметрами:

```
Raiserror('<ТЕКСТ СООБЩЕНИЯ>', 16, 1)
```

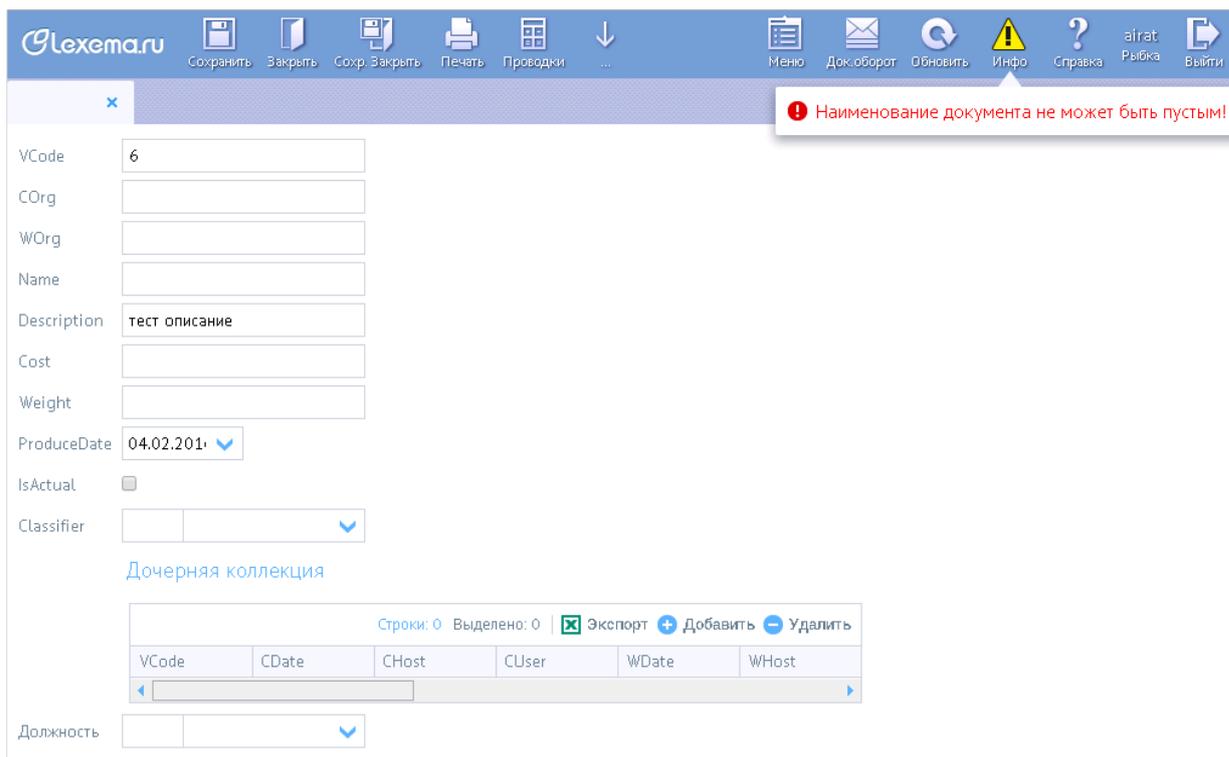
В этом случае текст сообщения отобразится на странице приложения в специальном блоке "Инфо" в правом верхнем углу:



Данный механизм является стандартным и предпочтительным способом информирования пользователя об ошибке. Примечание: подобное сообщение может быть также выведено путём вызова системного метода JavaScript.

Проверка работоспособности обработчиков

Откройте экранную форму на модели, для которой был создан обработчик события. Выполните действие, приводящее к выполнению запроса, на который назначен обработчик. Например, для модели данного документа на запрос INSERT назначен обработчик, проверяющий заполненность поле Наименование (Name). При попытке сохранить документ с пустым полем Name выводится сообщение об ошибке и операция прерывается:



Отладка скриптов в экранной форме

Тестирование и отладка является обязательным этапом в процессе создания качественного ПО. Отладка сценариев на языке JavaScript имеет свою специфику.

Использование консоли браузера Google Chrome

Практически каждый современный браузер оснащён средствами, облегчающими разработку, тестирование и отладку веб-приложений, в частности, JS-сценариев. Для работы с Lехeta.ru рекомендуется использовать веб-браузер Google Chrome последней версии.

Для открытия инструментария разработчика DevTools Google Chrome необходимо нажать клавишу F12 или комбинацию Ctrl+Shift+J. Окно DevTools содержит ряд вкладок:

- Elements - динамическое отображение элементов разметки страницы и их стилей
- Console - интерактивная консоль для ввода и выполнения команд на языке JavaScript
- Sources - исходные коды JS-сценариев, подключенных к текущей веб-странице
- Network - интерфейс для изучения последовательности загрузки компонентов веб-страницы, отслеживания запросов (в т.ч. асинхронных AJAX) и просмотра заголовков HTTP-запросов и ответов сервера
- Timeline
- Profiles
- Resources
- Security
- Audits

Вывод информации в консоль

Для вывода информации в консоль из сценариев на JavaScript необходимо использовать метод `console.log()`, который принимает один и более параметров и выводит их текущие значения на консоль, например:

```
var sum = L8.Data.root.Sum();
console.log("Сумма: " + sum);
// также можно указывать параметры через запятую
console.log("Логин текущего пользователя: ", L8.Access.UserName, " его роли: ",
L8.Access.userRoles);
```

Вывод ошибок и предупреждений

Кроме метода `log` объект `console` также имеет методы `warn` и `error` для вывода предупреждений и информации об ошибках соответственно. При этом текст предупреждений будет выводиться жёлтым, а текст ошибок - красным цветом.

Группировка записей

Измерение времени выполнения скрипта

Доступ к данным документа и к свойствам элементов формы в консоли

В процессе отладки часто бывает необходимо вывести или изменить значения полей формы или данных модели.

Чтение и изменение значений полей модели документа

Каждая экранная форма на модели предоставляет доступ к значениям всех полей модели с помощью объекта `L8.Data.root`. Синтаксис работы с полями имеет вид:

```
// считывание значения:
var ПЕРЕМЕННАЯ = L8.Data.root.ИМЯ_ПОЛЯ();

// установка значения:
```

```
L8.Data.root.ИМЯ_ПОЛЯ( НОВОЕ_ЗНАЧЕНИЕ );
```

Например:

```
// считывание значения:  
var cost = L8.Data.root.Cost();  
  
// установка значения:  
L8.Data.root.Cost( 123.5 );
```

Важно! Изменения производятся в объектах JavaScript на стороне клиента приложения (в браузере). Для сохранения изменений в БД необходимо сохранить документ, вызвав метод `L8.System.SaveDocument()`. В противном случае изменения будут потеряны после закрытия или перезагрузки страницы.

Чтение и изменение значений элементов управления экранной формы

Доступ к значениям всех элементов управления экранной формы может быть получен с помощью объекта `L8.View`. Синтаксис работы с полями имеет вид:

```
// считывание значения:  
var ПЕРЕМЕННАЯ = L8.View.ИМЯ_ЭЛЕМЕНТА.value;  
  
// установка значения:  
L8.View.ИМЯ_ЭЛЕМЕНТА.value = НОВОЕ_ЗНАЧЕНИЕ;
```

Например:

```
// считывание значения:  
var cost = L8.View.tbCost.value;  
  
// установка значения:  
L8.View.tbCost.value = 123.5;
```

Примечание: более предпочтительным способом изменения значений реквизитов документа является доступ через поля модели (`L8.Data.root`). Применять доступ к значениям элементов формы целесообразно для форм, основанных не на модели (аналитическая формы на запросах и т.п.).

Доступ к полям модели и элементам управления из консоли

Важно! Доступ к полям модели и элементам формы из консоли производится посредством специального объекта `_lxDebug.currentState`. Это обусловлено SPA-архитектурой приложения Lехeta.ru. Посредством данного объекта изменения производятся на текущей открытой вкладке. Таким образом, для чтения и записи значений из консоли необходимо использовать конструкции следующего вида:

```
var cost = _lxDebug.currentState.L8.Data.root.Cost();  
_lxDebug.currentState.L8.Data.root.Cost( 123.5 );  
var cost = _lxDebug.currentState.L8.View.tbCost.value;
```

```
_lxDebug.currentState.L8.View.tbCost.value = 123.5;
```

Анализ запросов с помощью SQL Profiler

При разработке прикладных модулей системы Lexema.ru периодически возникает потребность в анализе запросов к БД при работе экранных форм, запросов, отчётов, хранимых процедур и других объектов для диагностирования проблем. Для решения подобных задач предназначены инструменты профилирования SQL-запросов. Они позволяют:

- отслеживать события разных типов на сервере БД (выполнение запросов, хранимых процедур и т.п.)
- фильтровать события по различным критериям (имя БД, логин пользователя и т.п.)
- записывать последовательность действий в виде трассировки событий. Впоследствии трассировка может быть проанализирована, а также сохранена в файл или в БД.
- анализировать производительность (скорость выполнения) запросов для поиска и устранения "узких мест" системы
- и т.п.

В данной статье рассмотрены два инструмента:

- Lexema SQL Profiler, встроенный в моделлер приложения
- MS SQL Server Profiler, входящий в состав MS SQL Server

Lexema SQL Profiler

Данный инструмент предназначен для анализа запросов к серверу БД, инициируемых моделлером - программой для разработки прикладных конфигураций. С его помощью можно исследовать структуру БД прикладной логики и её объектов.

Для использования Lexema SQL Profiler запустите моделлер приложения. Нажмите на кнопку с изображением бочки в левом верхнем углу окна:



Для начала записи трассировки SQL-запросов моделлера к серверу БД нажмите кнопку "Запустить" на панели инструментов.

Выполните в моделлере действия, для которых требуется изучить запросы к БД. Например, после открытия списка моделей будет создана трассировка из нескольких запросов:

EventClass	TextData	Duration	StartTime	EndTime	CPU	ApplicationName	Reads	Writes	SPID	TransactionID
Trace start			13.04.2016 19:28:10							
SQL:BatchCompleted	--Установка флагов для с...	0	13.04.2016 19:28:19	13.04.2016 19:28:19	0	Lexema 7.0 (...)	0	0	51	
SQL:BatchCompleted	select [StoreHistory], [D...	74	13.04.2016 19:28:30	13.04.2016 19:28:30	0	Lexema 7.0 (...)	45	0	51	
SQL:BatchCompleted	--Загрузка по индексу на...	1	13.04.2016 19:28:30	13.04.2016 19:28:30	0	Lexema 7.0 (...)	240	0	51	
SQL:BatchCompleted	--Загрузка по индексу на...	242	13.04.2016 19:28:30	13.04.2016 19:28:30	218	Lexema 7.0 (...)	3902	0	51	
SQL:BatchCompleted	--Загрузка по индексу на...	5	13.04.2016 19:28:30	13.04.2016 19:28:30	0	Lexema 7.0 (...)	296	0	51	
SQL:BatchCompleted	--Загрузка по индексу на...	0	13.04.2016 19:28:30	13.04.2016 19:28:30	0	Lexema 7.0 (...)	64	0	51	
SQL:BatchCompleted	--Загрузка по индексу на...	0	13.04.2016 19:28:30	13.04.2016 19:28:30	0	Lexema 7.0 (...)	18	0	51	


```

--Загрузка по индексу на поле [PrimaryKeyProperty] коллекции объектов L8.Modeller.LexData.Model
select [DbName], [ViewName], [Name], [ScriptDefinition], [Description], [Type_ReferenceType], [Type], [PCode], [Inherit], [PrimaryKey], [GlobalId], [WUser],
from [L8_ModelProperty]
where [VCode] in (47,48,49,169,184,206,288,317,339,358,425,430,440,496,502,1223,1247,1251,1256,1281,1284,1293,1295,1347,1354,1357,1361,1367,1377,1388,1395,14
order by VCode

```

Таблица сверху содержит список событий (SQL-запросов), поле внизу - содержимое (SQL-код)

Поля таблицы:

- EventClass
- TextData
- Duration
- StartTime
- EndTime
- CPU
- ApplicationName
- Reads
- Writes
- SPID
- Transaction

К примеру, из трассировки событий при открытии списка моделей можно сделать следующие выводы: запрашиваются данные из трёх таблиц (L8_Model, L8_ModelProperty и L8_Namespace); самый длительный запрос производится к таблице L8_ModelProperty (242 мс).

MS SQL Server Profiler

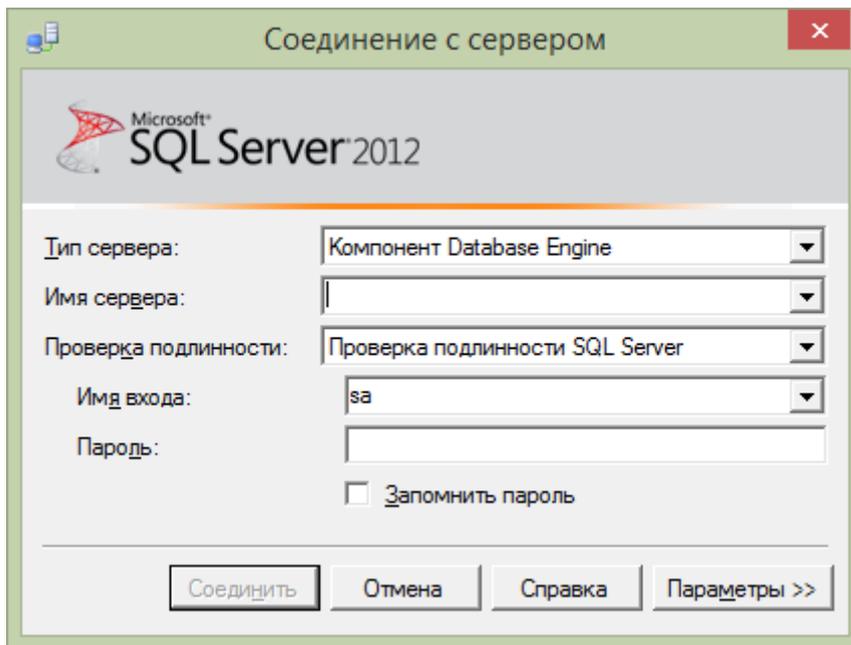
MS SQL Server Profiler — это инструмент, входящий в состав пакета MS SQL Server, позволяющий перехватывать события сервера БД. События могут быть сохранены в файле трассировки или в БД для дальнейшего анализа или использования с целью повторения определенной серии шагов для воспроизведения проблемы для её диагностики. Типовые сценарии использования SQL Server Profiler:

- контроль производительности экземпляра SQL Server Database Engine
- отладка инструкций Transact-SQL и хранимых процедур
- анализ производительности путем выявления медленно работающих запросов

- выполнение стресс-тестирования и контроля качества посредством воспроизведения трассировок
- воспроизведение трассировки одного или нескольких пользователей
- проверка инструкций Transact-SQL и хранимых процедур на стадии разработки проекта в пошаговом режиме для гарантии правильного выполнения кода
- устранение проблем в SQL Server с помощью перехвата событий в производственной системе (production-версии) и воспроизведения их в отладочной (тестовой версии). Это очень полезная возможность, поскольку позволяет во время проверки или отладки продолжать использовать производственную систему.
- аудит и отслеживание действий, происходящих в экземпляре SQL Server. Эта возможность позволяет администратору безопасности просматривать любые события аудита, в частности успешные и неудачные попытки входа в систему и разрешений доступа к инструкциям и объектам
- сохранение результатов трассировки в формате XML, что обеспечивает стандартизованную иерархическую структуру хранения результатов трассировки. Это позволяет вносить изменения в существующие трассировки или создавать их вручную для последующего воспроизведения
- статистический анализ результатов трассировки, позволяющий производить группирование и анализ похожих классов событий. В результатах содержатся счетчики, полученные на основе группирования по одному столбцу
- предоставление возможности создания трассировки пользователям, не являющимся администраторами
- настройка шаблонов трассировки, которые затем могут быть использованы для последующих трассировок

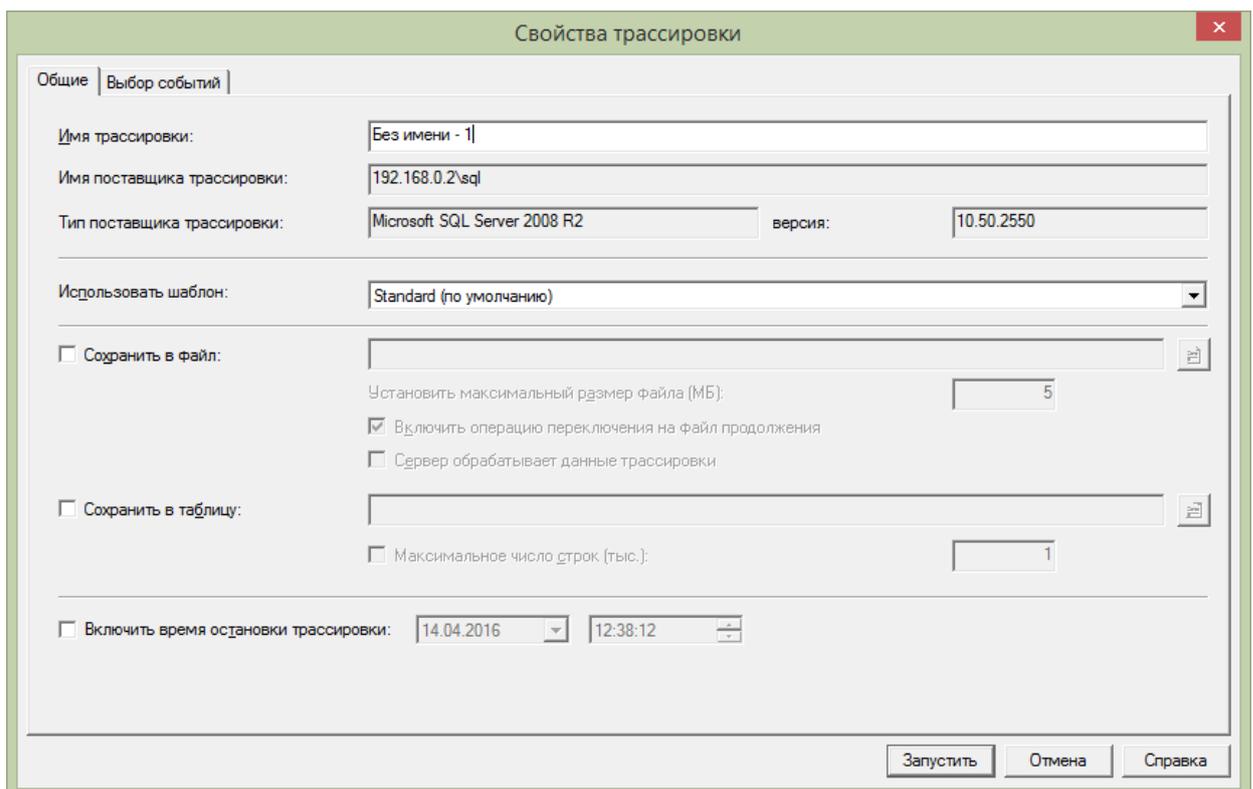
Запуск и подключение к серверу

Запустить MS SQL Server Profiler можно из меню ОС Windows (меню "Пуск") или из меню программы MS SQL Server Management Studio (пункт Сервис - "SQL Server Приложение Profiler"). После запуска необходимо авторизоваться на сервере - ввести адрес сервера, имя учётной записи и пароль:



Настройка параметров трассировки

Затем перед началом трассировки необходимо задать её свойства:



- Имя трассировки - целесообразно задавать в том случае, если её планируется сохранить
- Использовать шаблон - определяет конфигурацию трассировки по умолчанию. А именно, он включает классы событий, которые нужно контролировать в SQL Server Profiler. Например, можно создать шаблон, указывающий используемые события, столбцы данных и фильтры. Шаблоны не выполняются, а сохраняются в файлах с расширением TDF. После сохранения шаблон управляет захватом данных, если запускается трассировка, основанная на этом шаблоне.

- Сохранить в файл с целью повторного открытия и анализа
- Сохранить в таблицу - в этом случае трассировка будет сохранена в БД и её можно будет анализировать средствами SQL
- Включить время остановки трассировки - необходимо в случае длительных наблюдений

Выбор типов событий и их атрибутов

Важным шагом настройки трассировки является выбор событий (необходимо перейти на соответствующую вкладку). Вкладка "Выбор событий" содержит сетку - таблицу, которая содержит каждый из классов событий, доступных для трассировки. На каждый класс событий в таблице приходится по одной строке. Классы событий могут незначительно различаться в зависимости от типа и версии сервера, к которому они подключены. Классы событий идентифицируются в столбце События сетки и группируются по категориям событий. В оставшихся столбцах перечислены столбцы данных, которые могут быть возвращены для каждого класса событий. Чтобы включить события в трассировку, установите флажок в столбце События.

По умолчанию данный список содержит только некоторые категории и типы событий в соответствии с выбранным шаблоном (см. выше). Также отображаются не все столбцы. Для более детальной настройки рекомендуется поставить галочки "Показать все события" и "Показать все столбцы".

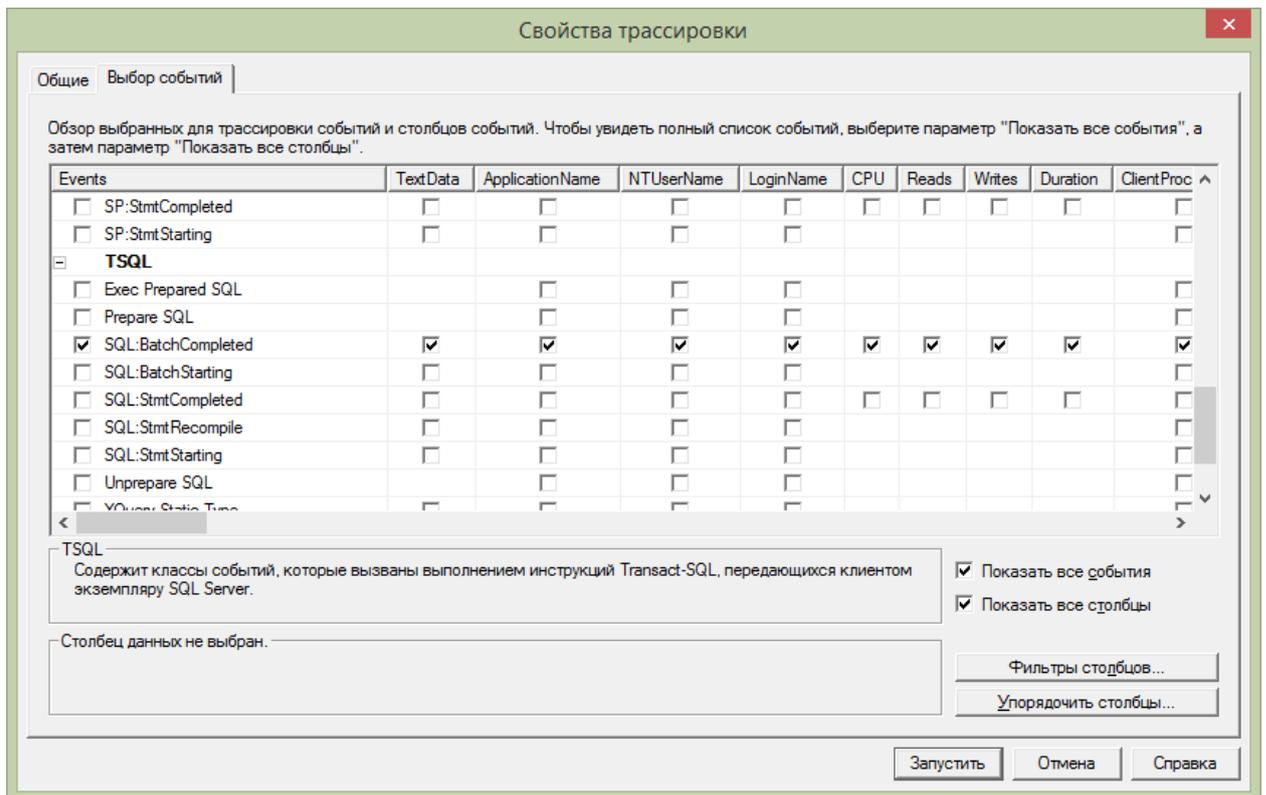
Если трассировка производится для отслеживания запросов, производимых приложением Lexema.ru, достаточно будет отметить 3 типа события в 2х группах:

- Stored Procedures (хранимые процедуры)
 - RPC:Completed - происходит при завершении удалённого вызова процедуры (RPC)
 - SP:Completed - происходит при завершении хранимой процедуры
- TSQL - отслеживание выполнения инструкций TransactSQL, передаваемых клиентами на сервер БД
 - SQL:BatchCompleted - возникает при завершении выполнения инструкции TransactSQL

Примечание: флажок в столбце Events может находиться в трёх состояниях:

- галочка отсутствует - событие не отслеживается
- установлена чёрная галочка - выбраны все столбцы данных - для выбранного события будут собираться все возможные для него данные
- установлена серая галочка - выбраны только некоторые столбцы данных - для выбранного события будут собираться только некоторые данные в соответствии с отметками в столбцах

По умолчанию для некоторых событий выбраны не все столбцы (стоит серая галочка). Для выбора всех столбцов необходимо снять галочку и установить её заново. При этом будут установлены галочки для всех видимых столбцов данных.



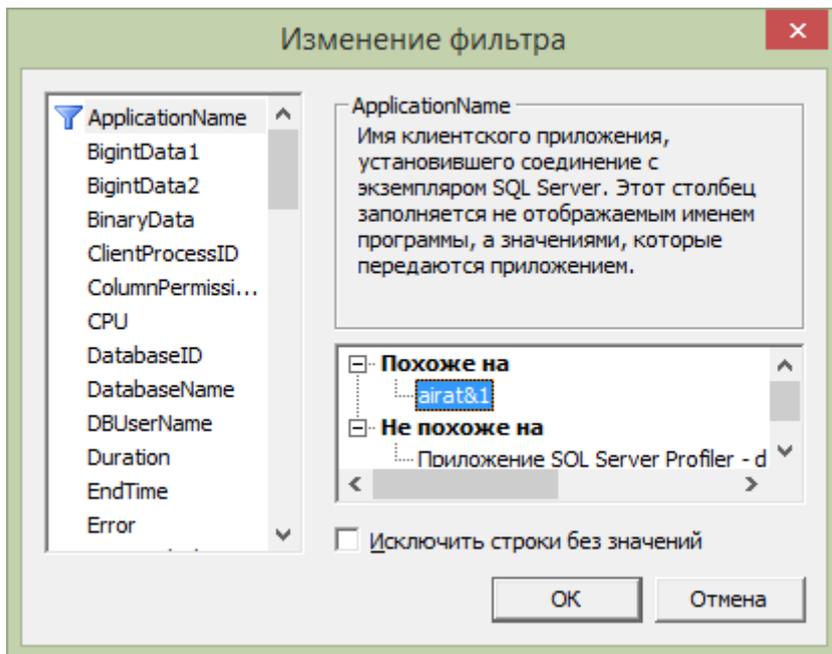
Ниже перечислены другие полезные категории и типы событий:

- Security Audit

Настройка параметров фильтрации

Фильтры ограничивают накопление событий в трассировке. Если фильтр не установлен, то на выход трассировки возвращаются все события выбранных классов событий. Установка фильтра трассировки необязательна, однако фильтр минимизирует затраты ресурсов при трассировке. Фильтры для определений трассировки добавляются на вкладке "Выбор событий" в диалоговом окне "Свойства трассировки" или "Свойства шаблона трассировки".

При отслеживании событий, происходящих при использовании веб-интерфейса Lexema.ru конкретным пользователем, целесообразно установить фильтр "ApplicationName" похоже на <логин_пользователя>+&1, например, "PetrovAN&1", где PetrovAN - логин пользователя:



При такой настройке фильтра в трассировку попадут только события, генерируемые действиями указанного пользователя.

Отслеживание событий

Предположим, что после выставления настроек, описанных выше и запуска трассировки, пользователь с логином 'airat' входит в систему и открывает реестр категорий доходов и расходов модуля "Домашняя бухгалтерия", а затем открывает один из документов (в качестве примера):

Код	Тип операции	Название	Родительская категория
2	Расходы	Автомобиль	
3	Расходы	Коммунальные платежи	
4	Расходы	Медицина	
5	Доходы (поступления)	Зарплата	
6	Доходы (поступления)	Гонорар за выступление :)	
7	Перемещения между...	Снятие наличных	
8	Расходы	Продукты питания	
9	Расходы	Бензин	Автомобиль
12	Расходы	Квартплата	Коммунальные платежи
13	Расходы	Лекарства	Медицина

В результате в трассировке SQL Server Profiler будет отображён список событий:

EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcessID	SPID	StartTime
Trace Start											2016-04-18 18:
RPC:Completed	exec sp_executesql N'SELECT [COrg] AS...	airat&1		Dataus...	47	41900	0	46	8772	57	2016-04-18 18:
RPC:Completed	exec sp_reset_connection	airat&1		Dataus...	0	0	0	0	8772	57	2016-04-18 18:
RPC:Completed	exec sp_executesql N'SELECT [COrg] AS...	airat&1		Dataus...	0	9	0	0	8772	57	2016-04-18 18:
RPC:Completed	exec sp_reset_connection	airat&1		Dataus...	0	0	0	0	8772	57	2016-04-18 18:
SQL:StmtCompleted	select VCode, Name from TransactionC...	airat&1		Dataus...	0	3	0	0	8772	61	2016-04-18 18:
SQL:BatchCompleted	select VCode, Name from TransactionC...	airat&1		Dataus...	0	3	0	0	8772	61	2016-04-18 18:
RPC:Completed	exec sp_executesql N'SELECT [COrg] AS...	airat&1		Dataus...	47	41855	0	52	8772	57	2016-04-18 18:
RPC:Completed	exec sp_reset_connection	airat&1		Dataus...	0	0	0	0	8772	57	2016-04-18 18:
RPC:Completed	exec sp_executesql N'SELECT [COrg] AS...	airat&1		Dataus...	0	2	0	0	8772	57	2016-04-18 18:
RPC:Completed	exec sp_reset_connection	airat&1		Dataus...	0	0	0	0	8772	57	2016-04-18 18:
RPC:Completed	exec sp_executesql N'SELECT [COrg] AS...	airat&1		Dataus...	0	2	0	0	8772	57	2016-04-18 18:

Анализируя список событий трассировки можно увидеть, что одним из самых длительных запросов был следующий:

```
exec sp_executesql N'SELECT [COrg] AS [COrg], [WOrg] AS [WOrg], [Description] AS [Description], [CHost] AS [CHost], [CUser] AS [CUser], [WHost] AS [WHost], [WUser] AS [WUser], [Name] AS [Name], [User] AS [User], [ParentName] AS [ParentName], [FullName] AS [FullName], [VCode] AS [VCode], [PCode] AS [PCode], [IsPersonal] AS [IsPersonal], [CDate] AS [CDate], [WDate] AS [WDate], [TType] AS [TType]
FROM [VTransactionCategory]
WHERE (EXISTS (SELECT TOP 1 1 FROM (SELECT [COrg] AS [COrg], [WOrg] AS [WOrg], [Description] AS [Description], [CHost] AS [CHost], [CUser] AS [CUser], [WHost] AS [WHost], [WUser] AS [WUser], [Name] AS [Name], [User] AS [User], [ParentName] AS [ParentName], [FullName] AS [FullName], [VCode] AS [VCode], [PCode] AS [PCode], [IsPersonal] AS [IsPersonal], [CDate] AS [CDate], [WDate] AS [WDate], [TType] AS [TType]
FROM [VTransactionCategory]
WHERE ([VCode] = @VCode)
) AS [parent] WHERE [parent].[TType] = [VTransactionType].[VCode]))
',N'@PrimaryKeyBoundary bigint,@TopCount bigint',@PrimaryKeyBoundary=NULL,@TopCount=NULL
```

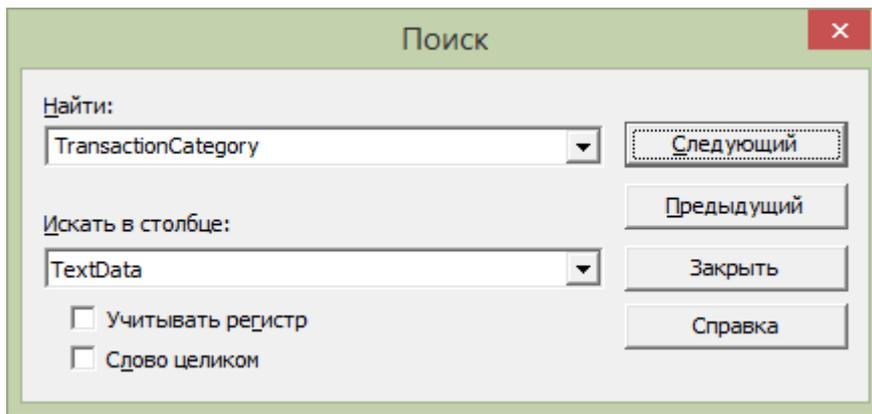
Судя по названию объекта (VTransactionCategory), это запрос на выборку списка категорий транзакций. Тип данного события - RPC:Completed (завершение выполнения удалённой процедуры).

Также в списке можно видеть событий типа SQL:BatchCompleted:

```
select VCode, Name
from TransactionCategory
where PCode is null
```

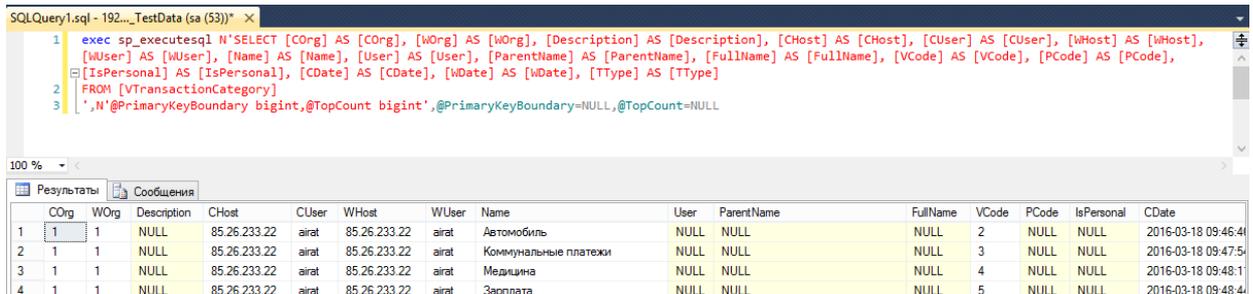
Это результат выполнения запроса (QuerySource) Lexema.ru.

Для поиска в тексте запросов, отслеженных в трассировке, необходимо нажать кнопку "Найти строку" (со значком бинокля) на панели инструментов или нажать комбинацию клавиш Ctrl+F:



Затем необходимо ввести искомый текст и выбрать столбец для поиска (текст запроса содержится в столбце TextData). После нажатия на кнопку "Следующий" курсор будет позиционирован на строку события, содержащего искомый текст.

В целях тестирования и отладки данные запросы могут быть выполнены повторно вручную. Для этого необходимо скопировать их текст, открыть SQL Server Management Studio, подключиться к соответствующему серверу, выбрать БД, создать запрос, вставить его текст и выполнить:



Отправка электронной почты средствами MS SQL Server DBMail

Компонент Database Mail — это подсистема для отправки сообщений электронной почты от компонента SQL Server Database Engine. Используя компонент Database Mail, приложения базы данных (запросы, хранимые процедуры, триггеры) могут отправлять почтовые сообщения пользователям. Сообщения могут содержать результаты выборки из БД, а также могут включать файлы из любого доступного сетевого ресурса. Отправка электронных сообщений используется протокол SMTP, наличие дополнительного ПО (например, почтового клиента Outlook) не требуется.

По умолчанию компонент Database Mail не активен. Для использования компонента Database Mail его необходимо явно включить с помощью мастера настройки компонента Database Mail, хранимой процедуры sp_configure или аспекта «Настройка контактной зоны» средства управления на основе политики.

Для использования компонента Database Mail необходимо выполнить ряд действий:

- включение компонента DBMail

- настройка:
 - создание учётных записей
 - создание профиля
- выполнение SQL-скрипта для отправки сообщения электронной почты

Примеры сценариев использования DBMail

- каждое утро менеджер хочет получать на почту итоги вчерашнего дня: сколько и какой продукции было продано, какой клиент сделал наибольшую заявку, какие клиенты имеют задолженность и прочее. В конечном итоге, результаты любого запроса могут быть отправлены с помощью Database Mail;
- администратор баз данных хочет получать на почту информацию о том какие задачи (job'ы) или планы обслуживания завершились с ошибкой (или наоборот, выполнены успешно);
- администратор баз данных хочет получать на почту информацию об ошибках определенного уровня серьезности, или с определенным номером, произошедших на сервере;
- и т.п.

Параметры безопасности

Для настройки DBMail необходимо быть членом предопределенной роли сервера sysadmin. Чтобы отправлять почту с помощью компонента Database Mail, необходимо быть членом роли DatabaseMailUserRole в базе данных msdb.

Настройка компонента Database Mail

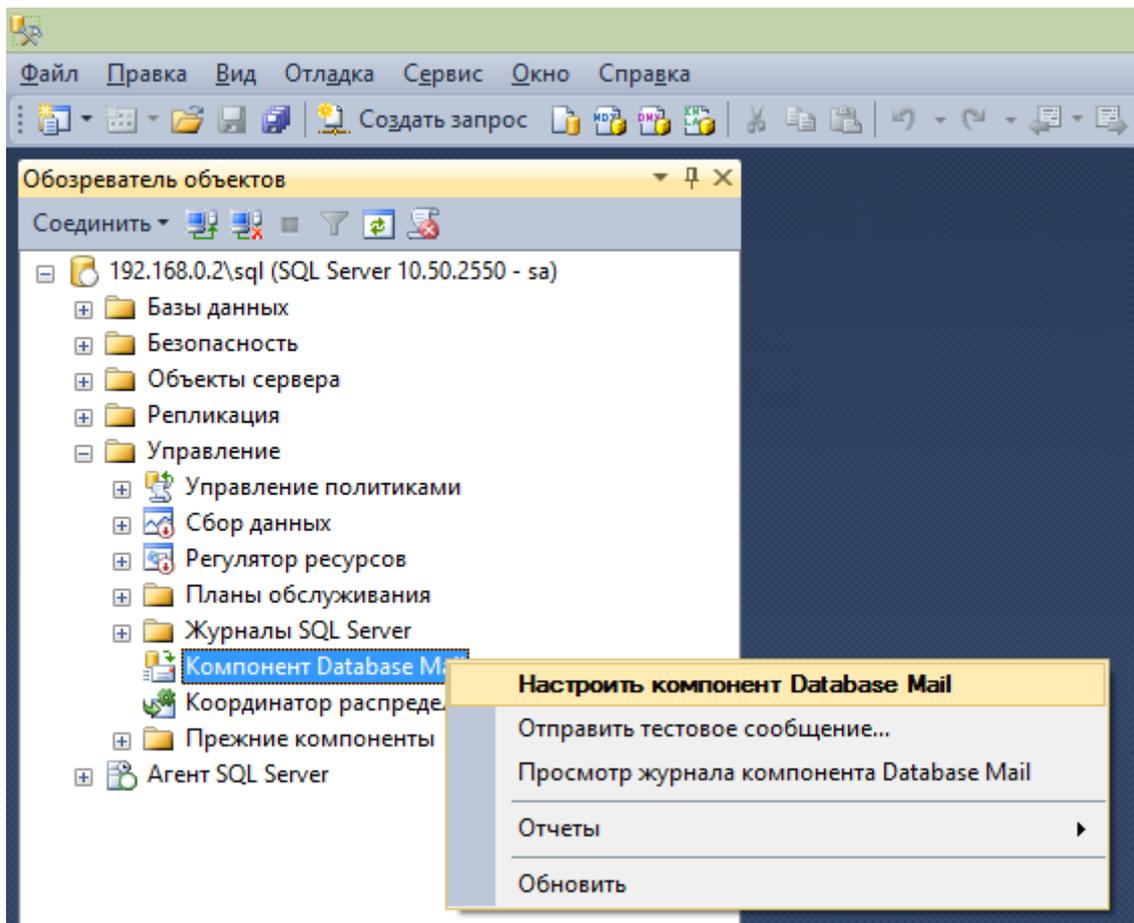
Регистрация учётной записи электронной почты на сервере

До начала настройки DBMail необходимо зарегистрировать новую учётную запись на почтовом сервере или использовать существующую. Понадобятся следующие параметры:

- символьный или IP-адрес почтового сервера, работающего по протоколу SMTP, а также порт (по умолчанию - 25)
- поддержка шифрования с помощью SSL/TLS
- адрес e-mail
- пароль

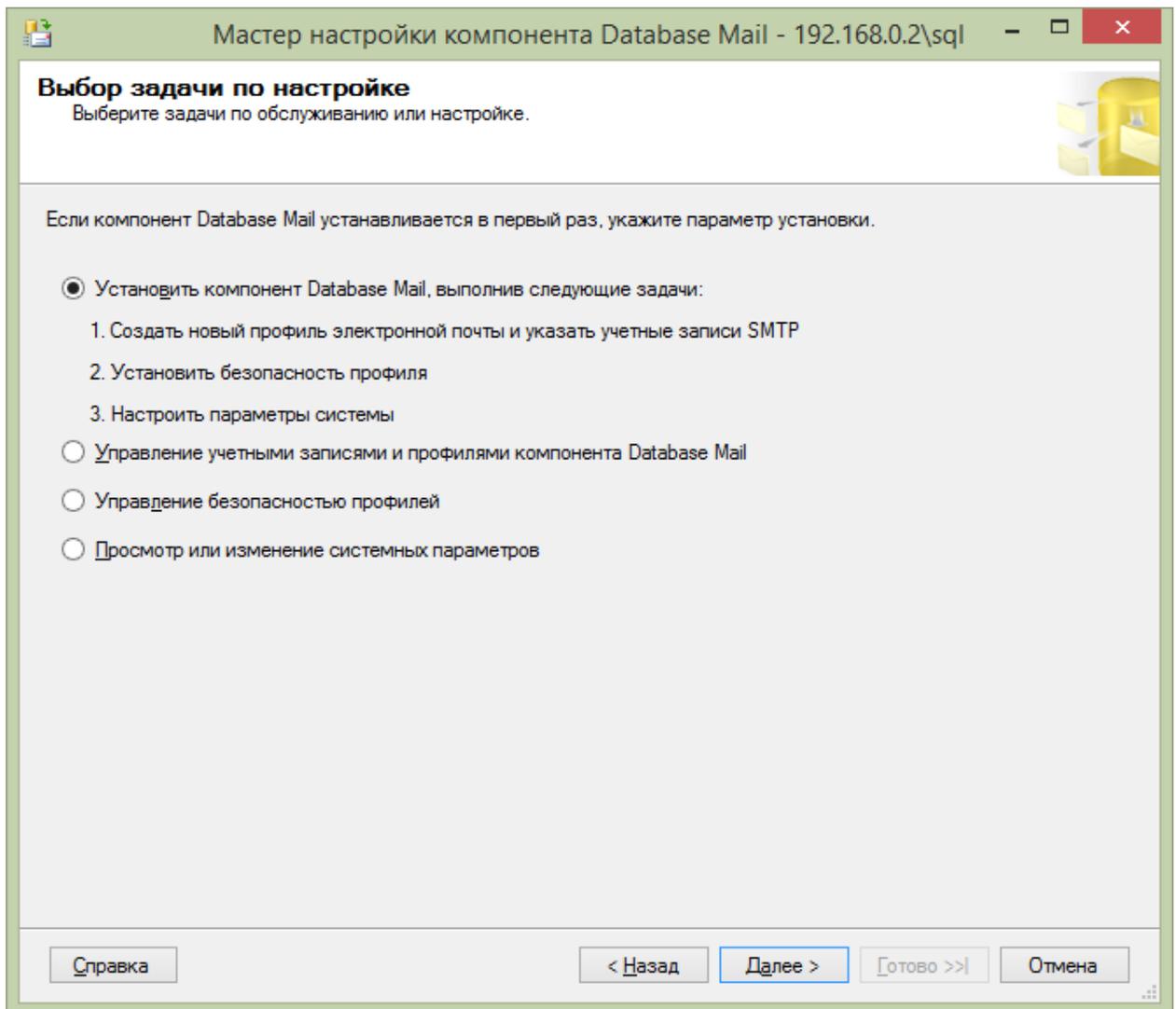
Использование мастера настройки

Запустите программу MS SQL Server Management Studio, установите соединение с экземпляром SQL Server. Раскройте узел Управление, щелкните правой кнопкой мыши на компонент Database Mail и выберите команду "Настроить компонент Database Mail". Если он используется впервые, выберите параметр "Установить компонент Database Mail". В противном случае выберите пункт "Настроить компонент Database Mail":



Окно приветствия можно пропустить, нажав Далее.

В следующем окне необходимо выбрать одну из нескольких задач:



Для дальнейшей настройки DBMail необходимо различать 2 элемента:

- Учётная запись компонента Database Mail - содержит сведения, необходимые для отправки электронной почты на SMTP-сервер (адрес, логин, пароль и т.п.).
- Профиль компонента Database Mail - является коллекцией учетных записей компонентов Database Mail. Профили повышают надежность в случаях, когда почтовый сервер становится недоступен, предоставляя альтернативные учетные записи компонента Database Mail. Необходима, по крайней мере, одна учетная запись компонента Database Mail.

До начала создания и настройки профиля необходимо создать одну или несколько учетных записей компонента Database Mail.

Создание учётной записи

В диалоговом окне "Выбор задачи по настройке" выберите пункт "Управление учётными записями и профилями компонента Database Mail", в следующем окне - "Создать новую учётную запись". Заполните форму параметров новой учётной записи:

Мастер настройки компонента Database Mail - 192.168.0.2\sql

Создание учётной записи компонента Database Mail

Укажите имя, описание и атрибуты учётной записи SMTP.

Имя записи: Lexema mail demo (elexema@gmail.com)

Описание: Демонстрационная учётная запись

Сервер исходящей почты (SMTP)

Адрес электронной почты: elexema@gmail.com

Отображаемое имя: Lexema.ru mail

Адрес эл. почты для ответа: elexema@gmail.com

Имя сервера: smtp.gmail.com Номер порта: 25

Для данного сервера требуется безопасное соединение (SSL)

Проверка подлинности SMTP

Проверка подлинности Windows с использованием учетных данных службы компонента Database Engine

Обычная проверка подлинности

Имя пользователя: elexema@gmail.com

Пароль: *****

Подтверждение пароля: *****

Анонимная проверка подлинности

Справка < Назад Далее > [Готово >>] Отмена

Подтвердите создание учётной записи нажатием кнопок "Готово" и "Закреть" в соответствующих диалоговых окнах.

Создание профиля

В диалоговом окне "Выбор задачи по настройке" выберите пункт "Управление учётными записями и профилями компонента Database Mail", в следующем окне - "Создать новый профиль". Заполните форму параметров нового профиля:

Введите имя и описание профиля.

Мастер настройки компонента Database Mail - 192.168.0.2\sql

Создание профиля

Укажите название профиля, описание, учетные записи и приоритет отработки отказа.

Имя профиля: Lexema mail demo

Описание: Демонстрационный профиль DBMail

Профиль можно связать с несколькими учетными записями SMTP. Если не удастся отправить почту с использованием какой-либо учетной записи, профиль выбирает следующую по приоритету учетную запись. Укажите учетные записи, связанные с данным профилем, и задайте для них приоритет отработки отказа.

Учетные записи SMTP:

Приор...	Имя учетной за...	Адрес электронной почты

Добавить...
Удалить
Вверх
Вниз

Справка < Назад Далее > Готово >> Отмена

Для привязки одной или нескольких учётных записей к профилю нажмите кнопку "Добавить":

Добавление учетной записи к профилю 'Lexema ...

Укажите учетную запись для добавления в профиль

Имя учетной записи: Lexema mail demo (ellexema@gmail.com) ▼

Адрес электронной почты: ellexema@gmail.com

Имя сервера: smtp.gmail.com

Создать запись...

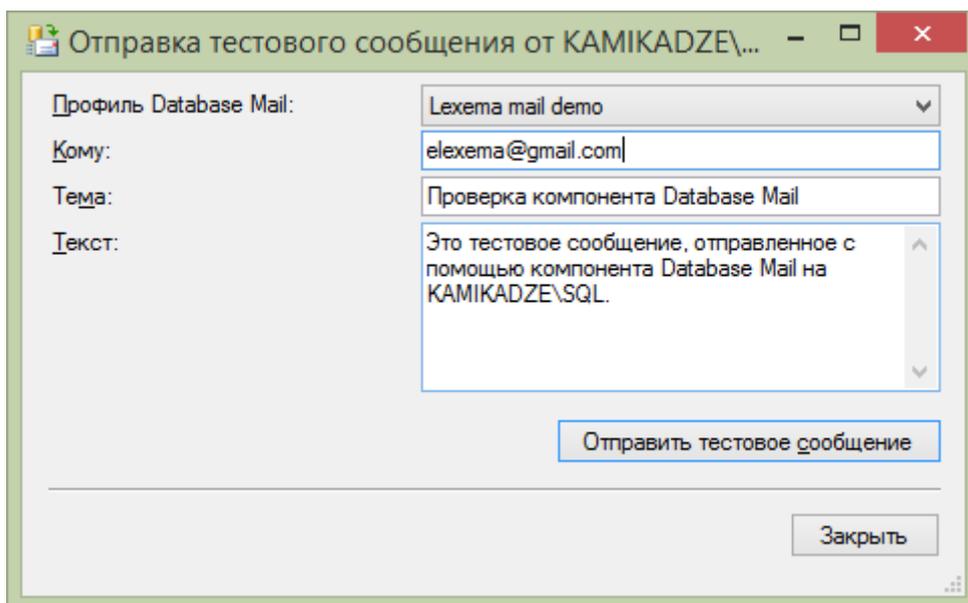
OK Отмена Справка

Выберите имя созданной ранее учётной записи, нажмите кнопку "OK". При необходимости привяжите к профилю другие существующие учётные записи или создайте новые. С помощью кнопок "Вверх" и "Вниз" расставьте учётные записи в порядке убывания приоритетов. Наличие нескольких учётных записей повышает надёжность отправки сообщений - при неработоспособности одной УЗ система DBMail автоматически использует другие УЗ из списка.

Подтвердите создание профиля нажатием кнопок "Готово" и "Закрыть" в соответствующих диалоговых окнах.

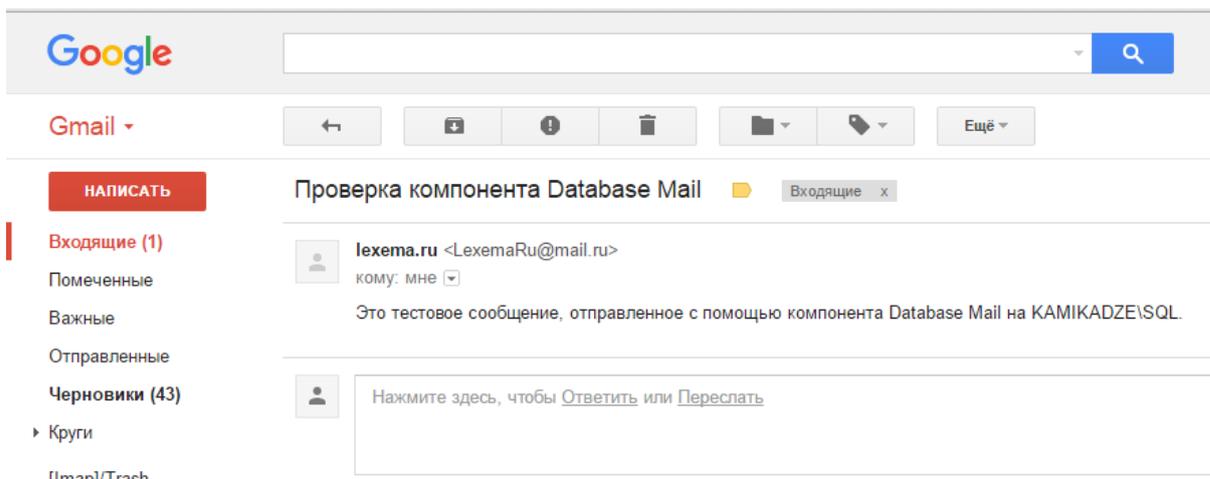
Отправка тестового сообщения

Запустите программу MS SQL Server Management Studio, установите соединение с экземпляром SQL Server. Раскройте узел Управление, щелкните правой кнопкой мыши на компонент Database Mail и выберите команду "Отправить тестовое сообщение". В поле "Кому" введите электронный адрес получателя тестового сообщения, при необходимости смените тему и текст сообщения:



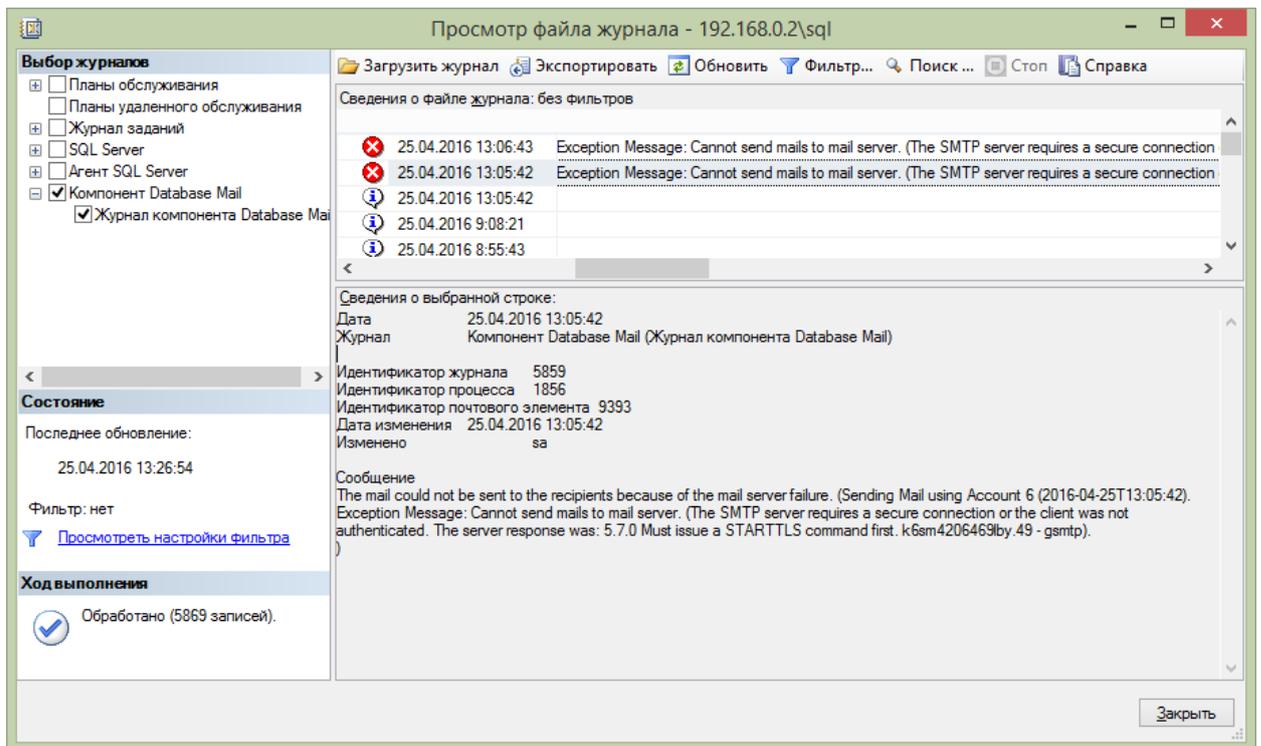
Нажмите кнопку "Отправить тестовое сообщение". В диалоговом окне с информацией о сообщении нажмите кнопку "ОК" для его закрытия.

В случае успешной отправки в почтовом ящике получателя отобразится тестовое сообщение:



Диагностика неисправностей

В случае возникновения проблем с доставкой электронных сообщений в первую очередь необходимо просмотреть журнал компонента DBMail. Для этого щелкните правой кнопкой мыши на компонент Database Mail и выберите команду "Просмотр журнала компонента Database Mail".



В сообщении содержится информация о причине ошибки. Например, в данном случае ошибка связана с отсутствием настройки шифрования подключения с использованием SSL.

Выполнение SQL-запроса для отправки сообщений из скриптов Lexema.ru

Для отправки сообщений электронной почты из SQL-скриптов используется системная хранимая процедура `sp_send_dbmail`, расположенная в БД `msdb`:

```
msdb.dbo.sp_send_dbmail @profile_name = ПРОФИЛЬ, @recipients = ПОЛУЧАТЕЛИ,
@subject = ТЕМА_СООБЩЕНИЯ, @body = ТЕКСТ_СООБЩЕНИЯ, @body_format =
ФОРМАТ_СООБЩЕНИЯ, @importance = ВАЖНОСТЬ
```

Параметры:

- **ПРОФИЛЬ** - название профиля компонента Database Mail, с помощью которого следует отправить сообщение (работа с профилями описана в данной статье выше)
- **ПОЛУЧАТЕЛИ** - список с разделителями (точки с запятыми), содержащий адреса электронной почты, по которым будут рассылаться сообщения. Список получателей относится к типу `varchar(max)`
- **ТЕМА_СООБЩЕНИЯ** - тема сообщения электронной почты, имеет тип `nvarchar(255)`. Если тема не указана, то по умолчанию устанавливается «Сообщение SQL Server»
- **ТЕКСТ_СООБЩЕНИЯ** - текст сообщения электронной почты. Аргумент `message` имеет тип `nvarchar(max)` и значение по умолчанию `NULL`
- **ФОРМАТ_СООБЩЕНИЯ** - формат текста сообщения. Аргумент имеет тип `varchar(20)` и значение по умолчанию `NULL`. Если установлено значение этого аргумента, то

устанавливаются заголовки исходящих сообщений, что указывает на формат текста сообщения. Аргумент может содержать одно из следующих значений: TEXT, HTML

- **ВАЖНОСТЬ** - важность сообщения. Аргумент имеет тип varchar(6). Аргумент может содержать одно из следующих значений: Low, Normal, High. По умолчанию имеет значение Normal

Полный список параметров и их описание приведены в официальной документации.

Для удобства использования DBMail для отправки электронной почты при разработке прикладных конфигураций в сценариях Lexema целесообразно создать отдельную функцию или метод объекта. Например:

```
// объект для работы с электронной почтой
L8.App.MailManager = (function () {
    // имя профиля
    var mailProfile = 'Lexema mail demo';

    // метод отправки сообщения
    function sendMessage(recipients, subject, text, fCallback) {
        // адрес для дублирования рассылки
        var logEmail = 'ecosoft.maillog@gmail.com';

        if (recipients) {
            //recipients = recipients + '; ' + logEmail;
            [[
                -- получение профиля из таблицы с настройками модуля СЭД
                declare @EmailProfile varchar(1000)
                select @EmailProfile = EMailProfile from DocflowSettings where
LTRIM(RTRIM(isnull(EMailProfile, '')))<>'

                if LTRIM(RTRIM(isnull(@EMailProfile, '')))<>'
                begin
                    -- журналирование фактов отправки сообщений
                    insert dbo.Logs(Level, Message, CreateDate, MachineName,
UserName, Details)
                    select 'MailInfoEst', 'EstDBMail. Попытка отправить
письмо', GETDATE(), HOST_NAME(), dbo.GetLogin(),
:recipients+'/'+:subject+'/'+:text

                    -- вызов процедуры отправки сообщения
                    DECLARE @res int
                    EXEC @res = msdb.dbo.sp_send_dbmail @profile_name =
@EMailProfile, @recipients = :recipients, @subject = :subject, @body = :text,
@body_format = 'HTML'
                    end
                ]].done(function (data) {
```

```

        console.log('Сообщение на адрес ', recipients, ' было успешно
отправлено ', new Date().toISOString());

        -- вызов функции обратного вызова, если соответствующий
параметр был указан
        if (fCallback && (typeof fCallback === 'function')) {
            fCallback();
        };
    });
} else {
    console.warn('sendMessage: не указан получатель!', recipients,
subject, text, new Date().toISOString());
}
}

return {
    sendMessage: sendMessage,
    mailProfile: mailProfile
}
}();

```

Примечание: в системе уже создан данный скрипт:

The screenshot shows the configuration for a script named 'MailManager'. The configuration includes a name, a note, a namespace, a priority of 1000, and a type of 'Script'. Below the configuration is a code editor showing the following JavaScript code:

```

1 L8.App.MailManager = (function () {
2     var mailProfile = 'kimamutdinova@mail.ru'/'GalyamovAF';
3
4     function sendMessage(recipients, subject, text, fCallback) {
5         var logEmail = 'kimamutdinova@mail.ru'/'A.Galyamov@targin.ru'; //'ecosoft.maillog@gmail.com';
6
7         if (recipients) {
8             //recipients = recipients + '; ' + logEmail;
9             [[
10                declare @EmailProfile varchar(1000)
11                select @EmailProfile = EMailProfile from DocflowSettings where LTRIM(RTRIM(isnull(EMailProf
12
13                if LTRIM(RTRIM(isnull(@EmailProfile,'')))<>'
14                begin
15                    insert dbo.Logs(Level, Message, CreateDate, MachineName, UserName, Details)
16                    select 'MailInfoEst', 'EstDBMail. Попытка отправить письмо', GETDATE(), HOST_NAME(), dt
17
18                    DECLARE @res int
19                    EXEC @res = msdb.dbo.sp_send_dbmail @profile_name = @EmailProfile, @recipients = :recip
20                end
21            ]].done(function (data) {
22                console.log('Сообщение на адрес ', recipients, ' было успешно отправлено ', new Date().toIS
23
24                if (fCallback && (typeof fCallback === 'function')) {
25                    fCallback();
26                }

```

Для его вызова необходимо подключить этот скрипт к экранной форме и вызвать метод:

```

L8.App.MailManager.sendMessage('your_mail@gmail.com', 'Тестовое сообщение',
'Тело тестового сообщения', function () {

```

```
console.log('Тестовое сообщение отправлено');
});
```

Возможные проблемы при отправке

Как показала практика использования компонента DBMail, при попытке отправки большого количества (нескольких сотен или тысяч) сообщений почтовый сервер не успевает обработать большую часть запросов или блокирует их. Для решения данной проблемы были добавлены небольшие паузы (1 сек.) между отправками отдельных сообщений и более длительные паузы между отправками пачек (например, после отправки 30 сообщений - пауза на 30 секунд). Проблема осложнялась асинхронным характером метода отправки сообщений. Применение описанной схемы позволило добиться нормальной отправки всех сообщений. Пример функции отправки множества сообщений по описанному алгоритму показан ниже:

```
function sendInvitations() {
    var subject = L8.App.Estimate.Settings.InvitationSubject,
        message,
        email,
        link,
        timeout = 100,
        timeoutOffset = 1000, // интервал между отправками писем
        timeoutBatchOffset = 60000, // интервал между пачками
        MSG_BATCH_NUMBER = 30; // кол-во писем в пачке

    if (confirm('Вы действительно хотите разослать приглашения на
оценку выбранным сотрудникам?')) {
        L8.System.pageBlocker.on();
        $.each(employeesList, function (index, employee) {
            if (employee) email = employee.Email;

            if (employee && email) {
                var fBuildAndSendMessage = function (resultsVCode,
email) {
                    setTimeout(function () {
                        link = buildEstimationLink(resultsVCode);
                        message =
L8.App.Estimate.Settings.InvitationMsg;
                        message = replaceField(message, '{ФИО}',
employee.FullName);
                        message = replaceField(message, '{ССЫЛКА}',
link);
                        message = replaceField(message, '{ЛОГИН}',
employee.UserName);
```


- Тип данных поля
- Первичный ключ
- Дочерняя таблица модели
- События модели
- Обновление базы данных (миграция)
- Запрос
- Экранная форма
 - Реестр
 - Элемент экранной формы (виджет)
- Программный сценарий (скрипт)

Документ

Документ представляет собой отражение реального объекта, процесса или явления, являющегося частью предметной области. В информационной системе Lexema.ru документ - это основной элемент. Часто документ в информационной системе является электронной версией документа на бумажном носителе. Например, договор, приходная накладная и т.п. Как правило, документ содержит номер и дату, по которой может осуществляться фильтрация при выборе расчётного периода в реестре документов.

Реквизит документа

Свойство (характеристика, атрибут) объекта, процесса или явления. Имеет определённый тип. В процессе проектирования программы необходимо выделить реквизиты документа, значимые для решения поставленной задачи и определить их типы. Примеры реквизитов: имя пользователя, полное наименование организации, цена с учётом НДС и т.п.

Дизайнер приложения

Для разработки прикладных конфигураций Lexema.ru используется специальное ПО - дизайнер (моделер) приложения, реализованный в виде "настольной" (desktop) программы. Дизайнер создан на основе Lexema версии 7. Для его работы требуется подключение к серверу БД. Бизнес-логика приложения хранится на сервере в отдельной БД. Моделер приложения позволяет разрабатывать множество конфигураций, что достигается за счёт подключения к разным базам логики.

Веб-клиент

Проигрыватель приложения. Осуществляет отображение элементов бизнес-логики (документов - экранных форм) и данных. Реализован в виде веб-приложения на основе технологии ASP.NET. Имеет подключения к трём базам данных - БД бизнес-логики, БД пользовательских данных, системной БД. Веб-клиент может работать одновременно с несколькими БД пользовательских данных.

Справочник

Справочник - редко меняющийся перечень уникальных позиций, содержащий краткие сведения, объединенные единой тематикой. Например, виды валют, единицы измерения, должности, подразделения и т.п. Справочник, как правило, имеет небольшое количество полей (часто - одно поле "Наименование"). Однако, встречаются сложные справочники с большим числом полей, например, контрагенты. Как правило, справочник не содержит даты. Записи в справочнике могут быть организованы иерархически: запись может иметь ссылку на родительскую запись (например, для отражения структуры подразделений предприятия).

Различия между документом и справочником

Разделение сущностей предметной области на документы и справочники является достаточно условным.

Отличительные особенности справочника:

- имеет небольшое кол-во полей
- значения редко меняются, иногда добавляются новые
- является независимой сущностью - не содержит полей-ссылок на другие сущности
- не содержит поля "Расчётная дата"

Отличительные особенности документа:

- как правило, содержит достаточно большое кол-во полей
- записи могут меняться, регулярно добавляются новые
- часто содержат ссылки на другие сущности - справочники
- могут иметь дочерние коллекции
- содержит поле "Расчётная дата" (или "Дата документа")

Модель

Модель - структурированное описание объекта предметной области. Является базовым элементом бизнес-логики Lexema.ru, лежит в основе любого документа или справочника. Физически модель связана с таблицей и/или представлением ("вьюшкой", view) в базе данных. Имеет имя, а также список полей. Поля модели описывают соответствующие реквизиты документов.

Представление БД (View)

Объект БД, построенный на основе запроса SELECT и предназначенный для выборки данных из одной или нескольких таблиц, связанных с некоторой сущностью.

Обновление базы данных

Основная статья: Обновление БД

Обновление базы данных - процесс синхронизации (миграции) структуры данных прикладной конфигурации (описания моделей) и структуры БД пользовательских данных. Является одной из функций дизайнера приложения. Для обновления БД необходимо ввести строку подключения - указать адрес сервера, название БД пользовательских данных, логин и пароль учётной записи сервера БД, имеющую права на создание и изменение структуры таблиц. Также можно отфильтровать список обновляемых объектов по имени хоста, с которого было выполнено создание или обновление объектов бизнес-логики.

Запрос

Элемент бизнес-логики Lexema.ru, позволяющий выполнять одну или несколько команд на языке SQL. На основе запроса может быть построена экранная форма. Запрос также может быть выполнен с помощью JavaScript-объекта QuerySource.

Экранная форма

Основная статья: Экранная форма

Экранная форма (ЭФ) - основной элемент пользовательского интерфейса. Является презентационным слоем приложения, выполняет функцию привязки данных к элементам экранной формы, предоставляет пользователю возможность чтения и изменения реквизитов документа. Для пользователя ЭФ представляется как веб-страница в браузере. Различают несколько видов экранных форм: документ, реестр (список документов), аналитическая форма. Источником данных для ЭФ является модель или запрос. В некоторых случаях ЭФ не имеет источника данных. Например, аналитическая форма может получать данные с помощью объекта QuerySource или LL-запроса.

Реестр документов

Список документов, представленный в виде аналитической таблицы. Как правило, ЭФ реестра позволяет пользователю создавать новые документы, открывать для редактирования

существующие, удалять выбранные документы, а также искать документы по указанным значениям реквизитов с помощью строки фильтров в заголовке таблицы.

Аналитическая форма

Предназначена для построения отчётов на основе выборки данных из одной или нескольких таблиц или представлений. В большинстве случаев содержит элементы управления для ввода или выбора значений параметров отчёта. Например, диапазон дат (от и до), тип документа (выпадающий список) и т.п. В качестве источника данных для аналитической формы используется запрос (объект QuerySource) или LL-запрос. Результаты запроса отображаются в виде таблицы (Grid или AnalyticalGrid), также может быть сформирована печатная форма (Отчёт).

Элемент экранной формы

Элементы экранной формы (элемент управления, виджет, контрол, ViewItem, элемент UI) - компоненты (структурные единицы), из которых строится экранная форма. Различают множество типов элементов ЭФ. Например, поле для ввода текста, выпадающий список, таблица, группа элементов, поле для выбора даты и т.п. Множество элементов ЭФ, как правило, имеет иерархическую структуру - построено в виде дерева. Существуют элементы-контейнеры, предназначенные для размещения в них дочерних элементов. Например, группы (LayoutGroup, BootstrapLayoutGroup), "аккордеон", панель вкладок. В большинстве случаев элемент ЭФ привязан к данным (например, к полю модели) и позволяет как считывать, так и изменять эти данные. Механизм синхронизации данных между презентационным слоем (элементами ЭФ) и моделью данных реализован согласно шаблону проектирования MVVM с помощью библиотеки KnockoutJS. Каждый элемент ЭФ имеет набор свойств (например, ширина, заголовок, источник данных, идентификатор и т.п.). Также к некоторым элементам ЭФ может быть привязан обработчик событий - сценарий на языке JavaScript. Примеры событий: нажатие кнопки мыши (одинарный или двойной клик), изменение значения и т.п.

Программный сценарий

Скрипт (сценарий) - программа на языке JavaScript. В Lexema.ru скрипты используются для осуществления вычислений в экранных формах. Например, скрипты-обработчики событий привязываются к элементам экранных форм и автоматически выполняются при возникновении событий.

Архитектура системы

Комплексная система автоматизации управления предприятием Lexema.ru содержит инструментарий для быстрой разработки прикладных конфигураций и включает три архитектурных компонента:

- веб-клиент
- сервер приложений, веб-сервер MS IIS
- сервер баз данных MS SQL Server

Для разработки ядра системы используется стек технологий Microsoft .NET. Для разработки прикладных конфигураций используются языки JavaScript и Transact SQL, используется отдельное ПО - моделлер (дизайнер) приложения.

Веб-клиент

Веб-клиент - компонент системы, с которым непосредственно работает конечный пользователь. Для работы требуется стандартный веб-браузер (см. раздел ниже). Представляет собой front-end веб-приложения системы, реализованный с использованием языков HTML, CSS и JavaScript. Для входа в систему требуется наличие учётной записи и прохождения авторизации. Веб-клиент построен в соответствии с концепцией одностраничного приложения (Single page application, см. раздел ниже).

Поддержка веб-браузеров

Несмотря на использование стандартных технологий, для работы с Lexema.ru рекомендуется использовать браузер Google Chrome последней версии, либо Internet Explorer версии 11. В других версиях Internet Explorer некоторые функции системы могут не работать. (Подробнее см. статью Поддерживаемые браузеры).

Концепция одностраничного приложения (Single page application)

Концепция одностраничного приложения (Single page application, SPA) предполагает работу пользователя на одной веб-странице в одной вкладке браузера. Навигация внутри системы осуществляется без перезагрузки веб-страницы. Система предоставляет возможность работать одновременно с несколькими документами, расположенными на отдельных вкладках системы. Одной из основных целей следования концепции SPA является уменьшение времени отклика (скорости работы) приложения, сопоставимой со скоростью работы native-приложений (подробнее см. в статье Single-Page Application).

Сервер приложений

В качестве сервера приложений используется ПО MS Internet Information Services (IIS). Для разработки веб-приложения системы была использована технология ASP.NET и язык C#.

Сервер БД

В качестве сервера БД используется ПО MS SQL Server версии от 2008 и выше.

Для работы системы требуется несколько баз данных:

- системная БД - хранит данные об организациях, учётных записях пользователей и т.п.
- БД прикладной логики - хранит объекты прикладных конфигураций (модели, экранные формы, запросы, отчёты, скрипты и т.д.)
- БД пользовательских данных - хранит данные документов организации. Для каждой организации создаётся отдельная БД пользовательских данных