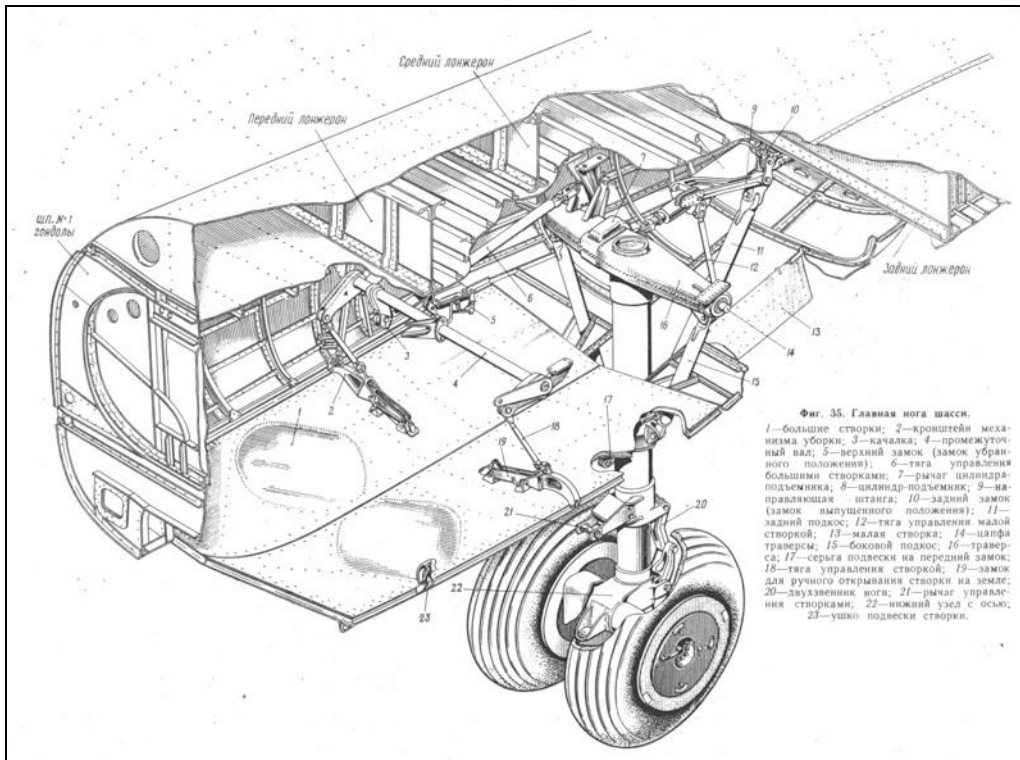




**Редакторы компонентов ETNI 2**

## Редакторы компонентов

Руководство разработчика



# Содержание

<b>1. Подключения.....</b>	<b>4</b>
<b>2. Источник данных.....</b>	<b>8</b>
2.1. Доступ к базе данных.....	8
2.1. Фильтрация подчиненной таблицы .....	10
2.2. Извлечение данных .....	11
2.3 Формирование полей потока.....	16
2.4 Подключение скриптов на стороне клиента.....	19
2.5 Тестирование настроек .....	21
2.6 Экспорт/Импорт источника.....	22
<b>3 Внутренние связи.....</b>	<b>23</b>
<b>4 Виртуальные таблицы. ....</b>	<b>26</b>
<b>5 Передача данных.....</b>	<b>30</b>
<b>6 Координатор.....</b>	<b>34</b>
<b>7 Переменные приложения.....</b>	<b>37</b>
<b>8 Pascal скрипты.....</b>	<b>41</b>
<b>9 SQL скрипты.....</b>	<b>44</b>

Компания Silentium делает сборку приложений, базирующихся на стандартных компонентах, разработанных нашими программистами. К ним относятся модули общего назначения, осуществляющие доступ к данным, управление переменными окружения, системы фильтрации, раскраски отображения, передачи сохранения, экспорта и так далее.

Большинство компонентов имеют достаточно мощные редакторы, поэтому описания этих редакторов мы решили вынести в отдельное руководство “Редакторы компонентов”.

## 1. Подключения.

Так как подключение к базе данных является необходимым атрибутом любой нашей системы, ему уделяется значительное внимание. Как правило, подключение и авторизация происходят один раз в начале сеанса работы программы, и завершается при выходе из нее. Бывают и исключения. Но, все по порядку. В описываемом редакторе есть возможность работать сразу с несколькими базами данных. Источники сами подхватывают настроенные подключения по называнию алиасов, если в них не организовано автономное подключение.

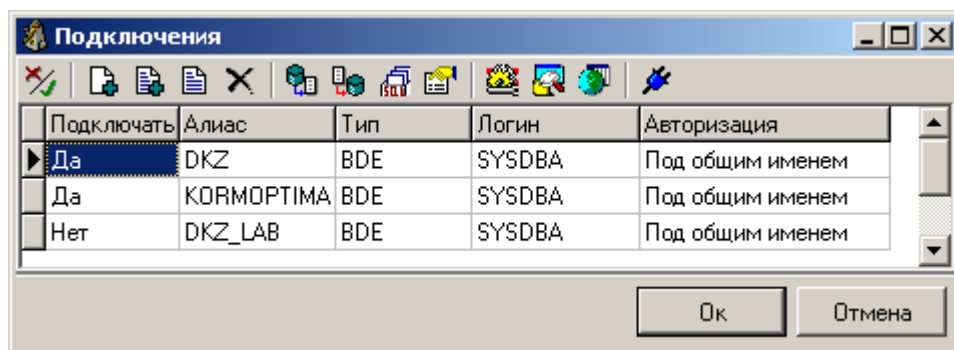


Рис 1.1. Редактор подключений к базам данных

Редактор подключений дает список используемых баз данных. Позволяет добавлять, удалять и редактировать подключения, делать их тонкую настройку, тестировать. Несколько функций дают возможность быстрого вызова дополнительных сервисных программ, поставляемых другими производителями. Это утилиты BDE Administrator, SQL Explorer, ODBC Администратор. Ниже приводится полный перечень сервисных функций и их описание.



Некоторые подключения могут быть активизированы “по требованию”. Такое необходимо, например, на рабочих станциях, где нет необходимых драйверов для доступа к одной из подключаемых баз данных, но программа должна успешно стартовать и не выводить сообщений об ошибке.

Для описанного выше случая используется функция “Подключать”. Соответственно в первой колонке списка отображается надпись “Да” или “Нет”. Вторая колонка показывает алиас базы данных. Иногда для оптимизации работы используется два алиаса на одну и ту же базу данных. Это может быть алиас BDE или алиас ADO, который полностью формируется в этом редакторе компонента. Колонка “Логин” отображает имя пользователя, под которым происходит подключение. Пароль соответственно не отображается. Важный момент – это тип авторизации. Можно подключаться к базе данных под одним и тем же пользователем, который здесь настроен, но правильнее транслировать пользователя, который непосредственно вводит свое имя и пароль для входа в систему. Это более надежно, но требует дополнительных усилий со стороны администратора базы данных и не всегда удобно для развития системы.

Ниже представлена таблица с описанием полного перечня доступных функций. Многие кнопки вызывают дополнительные окна, требующие ввода настроек.


Картинка	Функция	Описание
	Подключать	Указывается, будет ли сразу осуществляться подключение при старте программы
	Добавить	Добавляется нового подключения со всеми необходимыми параметрам в специальном окне, описанном ниже (рисунок 1.2).
	Добавить по образцу	Добавляется похожего подключения с копированием параметров и редактированием в специальном окне, описанном ниже (рисунок 1.2).
	Редактировать	Редактирование подключения в специальном окне, описанном ниже (рисунок 1.2).
	Удалить	Удаление подключения. Все настроенные параметры будут потеряны. Лучше для начала его отключить.
	Скрипт на старт приложения	В появившемся окне можно вводить SQL скрипт, выполняемый при старте программы.
	Скрипт на завершение приложения	В появившемся окне можно вводить SQL скрипт, выполняемый при завершении программы.
	Скрипт на подключение	В появившемся окне можно вводить SQL скрипт, выполняемый при подключении к базе данных.
	Скрипт на отключении	В появившемся окне можно вводить SQL скрипт, выполняемый при отключении от базы данных.
	Словарь	Настройка словаря для работы со специфическими SQL-инструкциями, отличающимися от стандарта SQL-92
	Параметры подключения	Выводится окно с параметрами подключения. Эти параметры можно менять, в том числе и для алиасов BDE..
	BDE Administrator	Вызов внешней программы "BDE Administrator"
	SQL Explorer	Вызов внешней программы "SQL Explorer"
	ODBC Администратор	Вызов внешней программы "ODBC Администратор"
	Проверить подключение	Попробовать подключиться

Таблица 1.1. Описание функций компонента подключений.

Редактирование, добавление подключений происходит в специальном окне, которое позволяет определить тип подключения

- ✓ BDE
- ✓ ADO
- ✓ FUSION

BDE – Borland Database Engine. Хорошо зарекомендовавшая себя технология, но, к сожалению, остановившаяся в своем развитии. Дает хорошие результаты, универсальность, скорость, но требует дополнительных инсталляций на рабочем месте пользователя.

ADO - ActiveX Data Objects. Технология доступа к данным, включающая набор высокоуровневых интерфейсов, позволяющих разработчикам обращаться к данным на любом языке программирования. Для новых операционных систем устанавливается сразу же с Windows. Пытается загрузить все данные на компьютер пользователя, чем достигается иллюзия скорости. Но при работе с миллионом записей до невозможности нервирует человека. Если учитывать это при разработке, то в принципе неплохо.

FUSION – Functional Universal Serial Input Output Navigation. Универсальная система обмена данными, базирующаяся на использовании стандартных провайдеров, в том числе по технологии noSQL (not only SQL), когда данные обрабатываются непосредственно в памяти и сохраняются в виде XML-файлов. Технология поддерживает отложенные изменения, синхронизацию с помощью встроенных автоматических искусственных первичных ключей. Технология FUSION впервые предложена компанией Silentium в 2005 году в качестве канала интеграции с учетными системами на уровне приложений.

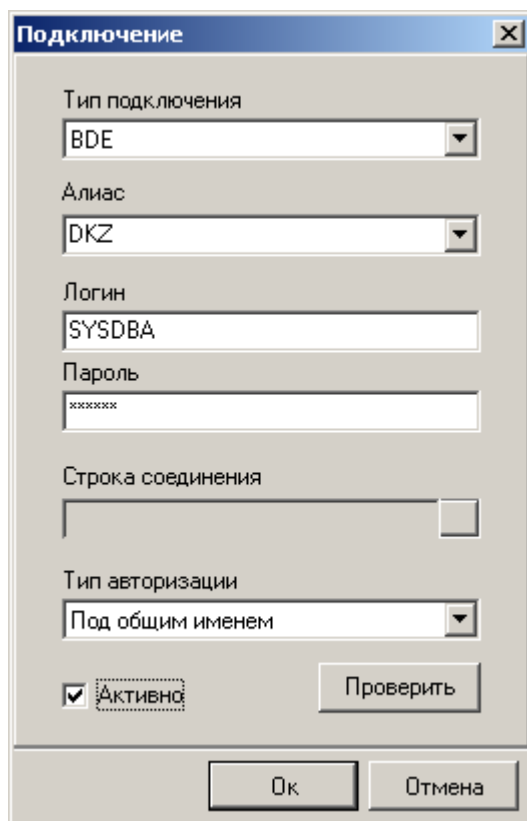


Рис 1.2. Настройка подключения с использованием разных технологий

Алиас выбирается из существующих, если это BDE (они настраиваются средствами Borland). Если это ADO или FUSION, то в этом окне самое время дать алиасу имя. С логином и паролем не будем углубляться – это и без того понятно. Строка соединения дает интерфейс от Microsoft, его описание выходит за рамки редакторов компонентов. Настройки там разные и зависят от базы данных, выбранного при этом провайдера. Показано на рисунке 1.3 “Свойства связи с данными”.



При настройке ADO рекомендуем в том месте, где пишется путь, не прописывать путь напрямую, а использовать макроподстановку с помощью параметра **[PATH]**. Большинство наших систем поддерживают такую запись. Это позволяет легко переносить программу в другое место без особых усилий.

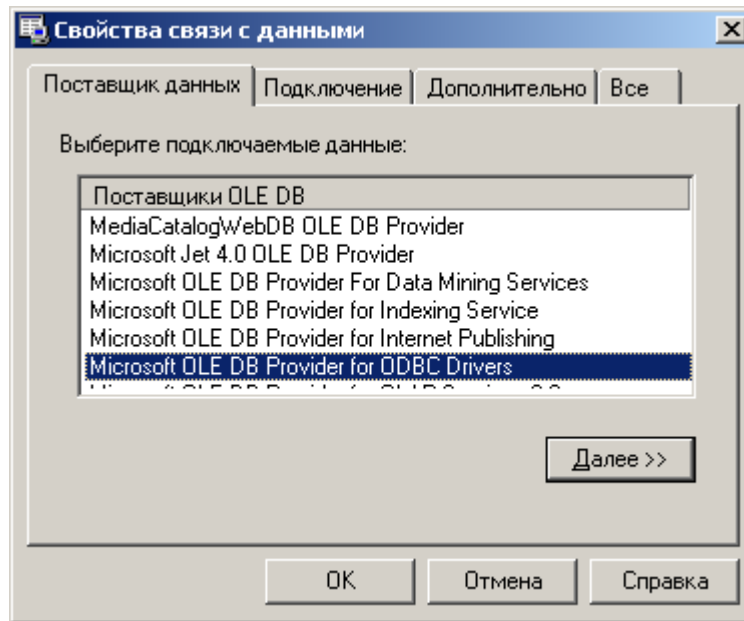


Рис 1.3. Прямое использование технологии драйверов ADO

Тип авторизации зависит от архитектуры организации данных. Если это просто подключение, например, к DBF-фалам, то ни имя не пароль не требуется. А вот для SQL-серверов можно всех подключать под одним именем, тогда не требуется дополнительная раздача прав. Но правильнее подключать под тем именем, под которым пользователь входит в систему. В таком случае требуется заранее на SQL-сервере задать такого пользователя и дать ему такой же пароль, как и в нашей системе. Это правильнее, но не всегда возможно. Уважаемый разработчик, за вами выбор из двух вариантов

- ✓ Под общим именем
- ✓ Под именем пользователя.

Ставить галочку “Активно” рекомендуем для часто используемых баз данных, тогда в начале сеанса происходит подключение, и дальнейшее открытие окон идет быстрее. Представляемый ниже скрипт на подключение пишется на языке SQL и поддерживает макроподстановку переменных окружения.

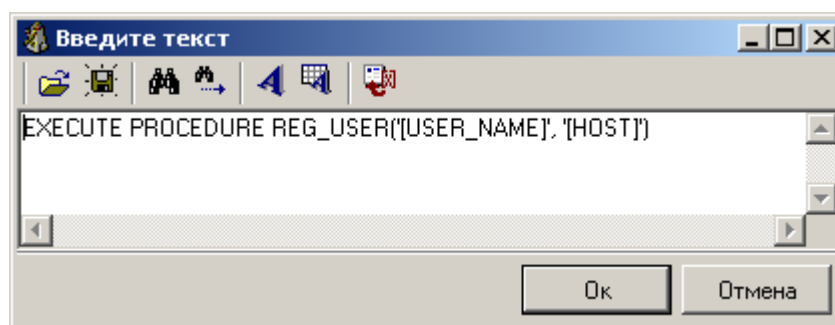


Рис 1.4. Написание скриптов на подключение и отключение базы данных

## 2. Источник данных.

Базовый компонент построения систем баз данных – это источник. Для многих сторонних архитектур такой компонент является просто доступом к таблицам, у нас же это нечто большее. Его оригинальное название MetaSource, что переводится как метаисточник. То есть не только данные, но и описание способов работы с ними. Этот компонент имеет обширный набор функций, сервисов и позволяет на базе представленных в нем настроек не только выводить таблицы, но и связывать их между собой, делать поиск, фильтрацию. Он позволяет легко строить графики, а так же реализовывать многомерный анализ данных.

Для начала обозначим основные возможности работы метаисточника и затем уже перейдем к конкретному описанию каждого поля настройки и возможных сценариев работы с ними. Некоторые функции могут дублировать описанный выше компонент подключения к базам данных. Это связано с тем, что наш источник может работать самостоятельно в составе тестовых приложений.

Основные функции источника:

- ✓ Доступ к базе данных
- ✓ Фильтрация подчиненной таблицы
- ✓ Извлечение данных
- ✓ Формирование полей потока
- ✓ Подключение скриптов на стороне клиента
- ✓ Тестирование настроек
- ✓ Экспорт/Импорт источника

### 2.1. Доступ к базе данных

Как уже отмечалось в компоненте подключения к базам данных, метаисточник может самостоятельно напрямую извлекать данные. Он имеет несколько видов подключения, а именно:

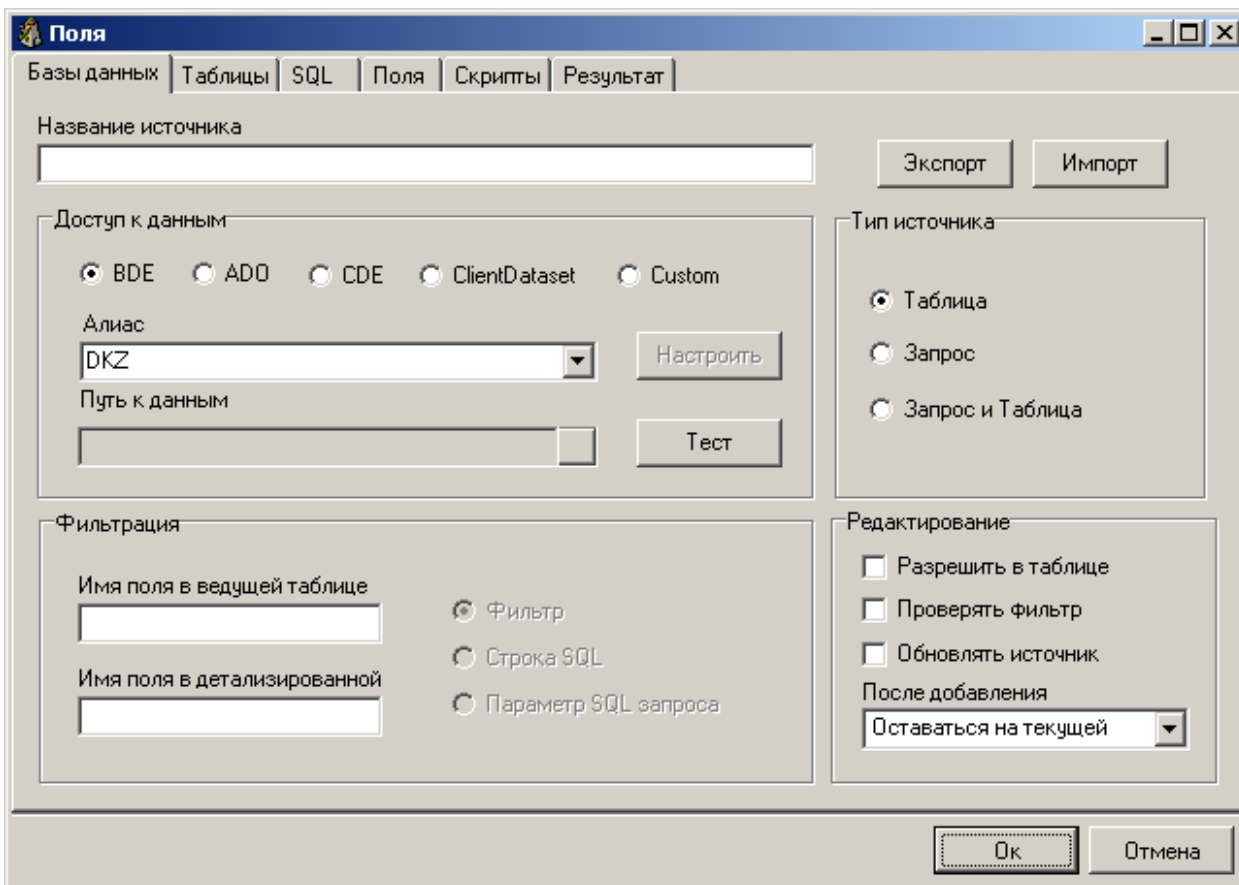
- ✓ BDE
- ✓ ADO
- ✓ CDE
- ✓ ClientDataset
- ✓ Custom

Все эти технологии перекрывают возможные варианты использования описываемого компонента. Обычно, для стандартных баз данных используется BDE или ADO, для доступа к таблицам в оперативной памяти тип CDE (Container Dynamic Engine). ClientDataset поддерживает обмен данными с DLL-функциями (так же общее назначение в составе специальных приложений, написанных на Delphi). Все, что не классифицируется в предыдущих вариантах, подключается как Custom (в том числе специальные компоненты прямого доступа). Последний вариант подключения полностью передает управление стороннему источнику и отвечает только за подпись полей (подключение к базе данных делается отдельно).

Технологию BDE описывать нет смысла – она используется во многих системах. Достаточно указать только имя алиаса в источнике.

Технология ADO применяется в двух вариантах. В одном случае выбирается алиас ADO по имени, во втором можно организовать доступ прямой настройкой строки соединения стандартным редактором. Это иллюстрирует рисунок Рис 1.3. “Прямое использование технологии драйверов ADO”.





**Рис 2.1.1.** Источник. Доступ к базе данных

Для использования предустановленных строк ADO можно дополнительно задавать путь к каталогу, При этом в списке алиасов выбираются следующие варианты

- ✓ DBASE
- ✓ ACCESS
- ✓ EXCEL
- ✓ CUSTOM

Для первых трех в списке достаточно указать путь – система сама правильно подберет драйвер, для всех остальных случаев доступа через ADO следует выбирать CUSTOM. После того как соединение настроено, его можно протестировать командой “Тест”.

Для некоторых систем требуется указывать имя источника. Оно указывается в графе “название источника и фигурирует в дальнейших настройках. Рекомендуется давать источнику осмысленные имена, что, несомненно, облегчает дальнейшее развитие и совершенствование системы.

Источник так же может быть настроен по типу

- ✓ Таблица
- ✓ Запрос
- ✓ Запрос и таблица

В зависимости от задачи следует правильно выбирать тип источника. Для начинающих разработчиков рекомендуется начинать с таблиц и их связывания. Этот метод требуется, когда речь идет о вводе данных. Если же делаются отчеты, графики или построение анализа с помощью многомерного куба, то правильнее выбрать тип “Запрос”. Это тип требует знания языка SQL. Предполагается, что на отдельной закладке разработчик сам напишет инструкцию по извлечению записей.

На рисунке 2.1.1 показан так же блок, связанный с редактированием. Здесь можно настроить поведение самого редактирования записей, а так же действий по его завершению. Редактирование подразумевает не только изменение записи, но ее добавление и добавление по образцу. При редактировании есть варианты:

- ✓ Разрешить в таблице.
- ✓ Проверять фильтр.
- ✓ Обновлять источник

Если разрешить редактирование в таблице, то изменение данных происходит на месте без вызова специального окна, в котором есть кнопка “Ок” и “Отмена”. Эти окна генерируются автоматически на основании информации введенной в описании полей. Если эту галочку не поставить, то ввод осуществляется напрямую в таблице.

Когда выставлена галочка “Проверять фильтр”, то перед сохранением изменений происходит проверка соответствия условиям фильтрации и если условие не удовлетворяется, то генерируется сообщение об ошибке. Это может быть полезно, например, когда данные показываются за определенный диапазон дат. Если перед сохранением не проверить, входит ли дата в диапазон, то добавляемая запись тут же фильтруется, и не будет показываться пользователю, пока тот не сменит рабочий период. Если запись после ввода пропадает из поля зрения, то это может дезориентировать оператора, поэтому рекомендуем включать эту опцию. Иногда такая проверка может быть лишней – это уже решать разработчику приложений.

После того, как запись была изменена, ее необходимо обновить, с тем условием, что она могла быть подправлена в триггере непосредственно на сервере. То есть надо показывать какой она стала после вставки. Это дополнительная задержка, поэтому эту опцию можно отключать для ускорения работы системы, но, как правило, такое обновление необходимо.



Если опция “ Обновлять источник” отключена, то при повторном редактировании может возникать сообщение об ошибке, что другой пользователь изменил запись. Тогда оператор может просто обновить источник данных соответствующей клавишей.

Поле того, как произошло добавление новой записи (в том числе добавление по образцу), есть возможность по-разному позиционировать текущую запись:

- ✓ Остаться на текущей
- ✓ Перейти в конец таблицы
- ✓ Перейти на добавленную запись
- ✓ Не обновлять изменения

Самый правильный вариант здесь - перейти на добавленную запись. Это наиболее логично – пользователь добавляет запись и на нее переходит. Но для больших объемов данных это может быть связано со значительной задержкой (некоторым 2-3 секунды – уже долго). К тому же требования к поведению бывают разные. Потому предусмотрено несколько вариантов. Попробуйте разные из них.

## 2.1. Фильтрация подчиненной таблицы

Полный вариант фильтрации реализуется в специальном компоненте, который будет описан в отдельной главе. Источник реализует один вариант, когда необходимо отфильтровать один подчиненный источник (связь один ко многим). Предполагается, что второй источник уже подключен и настроен. Достаточно только указать имя поля в ведущей таблице и имя поля в детализированной.

Код г.	Название
152	ВАФЛИ
161	ВОСТОЧНЫЕ СЛАДОСТИ
150	ДРАЖЕ
	Другое
149	ЖЕВАТ.КОНФЕТЫ,ХОЛОДКИ,ЖЕ
156	ЗЕФИР,ПАСТИЛА
162	ИРИС
151	КАРАМЕЛЬ
148	КЕКСЫ,ТОРТЫ,РУЛЕТЫ,ПИРОЖН

Код группы	Название
162	ИРИС БУРЕНКА(молочный имп.эт.)
162	ИРИС ВОЛШЕБНЫЙ(ЗОЛОТОЙ)КЛ
162	ИРИС ДЕТСКИЙ
162	ИРИС ЗАБАВА СЛАСТЕНЫ
162	ИРИС ЗОЛОТОЙ КЛЮЧИК
162	ИРИС ЗОЛОТОЙ КЛЮЧИК ИМП.ЭТ
162	ИРИС КИС-КИС
162	ИРИС КИС-КИС ЗАВ."В ЗАМОК"
162	ИРИС КИС-КИС ИМП.ЭТ.

**Рис 2.2.1.** Пример связывания ведущее детализирующей таблицы

В приведенном на рисунке примере связаны две таблицы по коду группы. Когда пользователь перемещается по левой таблице, то правая автоматически обновляется и показывается товар по выбранной слева группе. Для этих целей и предназначена фильтрация подчиненной таблицы.

## 2.2. Извлечение данных

Как было указано в предыдущей главе, данные можно извлекать как с помощью запросов на языке SQL, так и с помощью связывания таблиц. Связь записей таблиц происходит один к одному. То есть одной записи в главной таблице соответствует одна запись в подчиненной таблице.



В качестве поля для связывания в подчиненной таблице используется первичный ключ (поле с непустыми и не повторяющимися значениями). Если одной записи соответствует несколько, то из них берется первая (а точнее первая попавшаяся). Ошибки при этом не возникает.

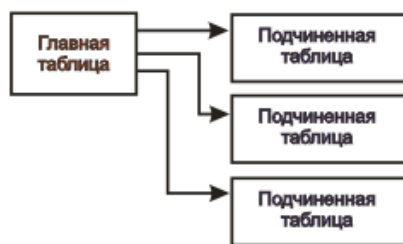
Связи могут быть простыми, так и сложными и иметь разную топологию.



**Рис 2.2.1.** Пример простых связей таблиц.

Для простых связываний характерно большинство приложений. Например, для справочника товаров можно сделать связь между справочником и единимой измерения. Более сложный пример, когда к строке накладной привязывают сам товар, а уже товар имеет код единицы измерения, связь с которым осуществляется через таблицу.

Примечательно, что в нашем компоненте можно делать и сложные связи по нескольким полям. Такая связь необходима при использовании составного индекса. Мы такие связи не рекомендуем ввиду их сложного восприятия, но при использовании натуральных ключей, либо интеграцией с чужими системами такие связи бывают просто необходимы.



**Рис 2.2.2** Пример с несколькими подчиненными таблицами, связанными с главной.

Связи на рисунке 2.2.2 характерны для заголовков документов. Например, накладная может быть связана сразу с таблицей предприятий, с менеджером, с договором. Особенно такие связи хороши, когда требуется связать одну таблицу несколько раз, как в случае, когда плательщик одно предприятие, а грузополучатель - другое.



**Рис 2.2.3** Пример с несколькими сложноподчиненными таблицами.

Подобные связи (как на рисунке 2.2.3) более характерны для подсистем планирования, контроля договорных отношений, производственных процессов, имеющих множество связей. Самих связей может набраться с десятков и их реализацию следует вести поэтапно, вводя в строй новые и проверяя получаемые результаты.



При решении задач связывания нет ничего хуже пропущенных значений. Настоятельно рекомендуем использовать первичные ключи, а так же ссылочную целостность, которая не позволяет вводить оборванные записи.

Даже если такие записи имеются (с некорректными кодами, пропущенными значениями), то в любом случае компонент пытается найти подходящие варианты для связи. Рассмотрим иллюстрирующий пример для трех таблиц PERSON (Ответственное лицо), TASKRESULT (Результаты задания), TASKJOURNAL (Журнал неисправностей),

Обращаем ваше внимание, что перечень таблиц идет в обратном порядке. То есть самая нижняя – это главная таблица (или основная). Следом за ней идут вверх таблицы имеющее подчинение. Порядок таблиц имеет значение и иногда результат может быть разным даже при одних и тех же связывающих полях.

В нашем примере (на рисунке 2.2.4) реализовано две связи между тремя таблицами. Связи эти пишутся в последней колонке. В зависимости от того, какая таблица является более главной, на той строке и записывается связь. Синтаксис связи “XXX-YYY”. При чем, XXX-имя поля в главной таблице, XXX-Алиас (имя) подчиненной таблицы. Несколько связей разделяются точкой с запятой “;”. В приведенном примере сделано две связи:

- ✓ CODE\_PERSON-PERSON;
- ✓ CODE\_TASKRESULTS-TASKRESULTS;

Можно было бы возразить, что тут не хватает поля в подчиненной таблице, но эта информация берется из колонки “Индекс”. В приведенном примере это поля CODE\_PERSON и CODE\_TASKRESULTS.

Для открытия и связывания одной и той же таблицы несколько раз используются алиасы таблицы. Если алиас не проставляется, то система автоматически прописывает алиас таблицы, идентичный ее названию.

Итак, для связывания таблиц необходимо заполнить следующие колонки

- ✓ База данных
- ✓ Таблица
- ✓ Алиас таблицы
- ✓ Связи

База данных, как правило, не указывается и берется из первой закладки метаисточника. Но иногда требуется сделать связь между таблицами из нескольких баз данных. Так же это поле используется при оптимизации работы BDE. С этим полем работают функции “Установить алиас таблицам” и “Очистить алиасы у таблиц”.

При заполнении названия таблиц вопросов не возникает, однако следует учитывать некоторые особенности самих СУБД. Для Oracle уместно указывать область/владельца через точку (например, RAIN.COMPANY), для драйверов dBase пишут расширение dbf (например, COMPANY.DBF), для Excel-таблиц пишут в квадратных скобках, да еще знак доллара в конце (например [Книга1\$]). Всех фантазий разработчиков драйверов невозможно даже вообразить, так что написание таблиц зависит от конкретной базы данных.

Алиас таблицы дается разработчику, чтобы он написал название таблицы в том виде, в котором ему будет удобно ее использовать в дальнейшем при построении клиентского приложения. Это должно быть простое слово без точек и спецсимволов, желательно латинскими буквами.

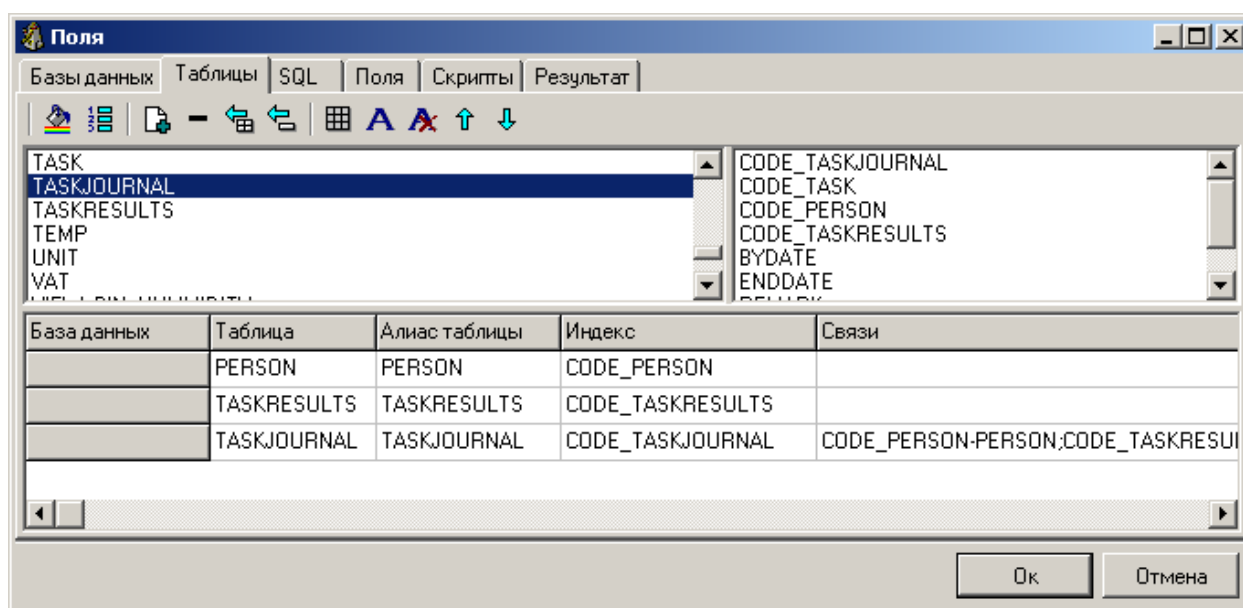













Рис 2.2.4 Пример связывания трех таблиц.

Чтобы удобно было выбирать таблицы и поля, в верхней части окна расположены два списка – левый список дает перечень таблиц (двойным щелчком проставляется выбранная таблица в поле таблица). Используйте функцию “Показать таблицы” для получения их перечня. В правой части показываются для выбранной таблицы с нажатием функции “Показать поля”. Поля так же вставляются на двойной щелчок.

Картинка	Функция	Описание
	Показать таблицы	Показывается список таблиц в левом верхнем списке.
	Показать поля	Показывается список полей в правом верхнем списке.
	Добавить строку	Добавляется еще одна строка для ее дальнейшего заполнения.
	Удалить строку	Удаляется текущая строка из перечня таблиц
	Вставить таблицу	Выбранная в левом списке таблица, прописывается в выбранной строке в колонке "Таблица"
	Вставить поле	Выбранное в правом списке поле прописывается в текущей строке таблиц в колонке "Индекс"
	Очистить список таблиц	Удаляются все строки, содержащие таблицы.
	Установить алиас таблицам	Для всех таблиц прописывается алиас, который будет выбран в появившемся списке алиасов.
	Очистить алиас у таблиц	У всех таблиц удаляется алиас. Они начинают работать через общий алиас подключения источника
	Строку вверх	Перемещает строку таблицы на одну позицию вверх, при этом верхняя строка опускается ниже
	Строку вниз	Перемещает строку таблицы на одну позицию вниз, при этом нижняя строка опускается вверх

**Таблица 2.1.1** Описание функций для связывания таблиц.

Часто бывает, что имеется уже настроенная схема связей таблиц, но целесообразнее использовать SQL-запрос, либо их комбинированный вариант. Тогда, на основании настроенных связей система может сама сгенерировать соответствующий запрос, например для приведенного случая получаем:

```

SELECT
TASKJOURNAL.CODE_TASKJOURNAL AS CODE_TASKJOURNAL,
TASKJOURNAL.CODE_TASK AS CODE_TASK,
TASKJOURNAL.CODE_PERSON AS CODE_PERSON,
TASKJOURNAL.CODE_TASKRESULTS AS CODE_TASKRESULTS,
TASKJOURNAL.BYDATE AS BYDATE,
TASKJOURNAL.ENDDATE AS ENDDATE,
TASKJOURNAL.REMARK AS REMARK,
TASKJOURNAL.DOCUMENT AS DOCUMENT,
PERSON.NAME AS PERSONNAME,
TASKRESULTS.NAME AS TASKRESULTSNAME
FROM
PERSON, TASKRESULTS, TASKJOURNAL
WHERE TASKJOURNAL.CODE_PERSON = PERSON.CODE_PERSON
AND TASKJOURNAL.CODE_TASKRESULTS = TASKRESULTS.CODE_TASKRESULTS
ORDER BY TASKJOURNAL.CODE_TASKJOURNAL

```

Такой запрос можно далее модифицировать, и при желании, остановив его развитие, опять вернуться к предыдущей концепции связывания таблиц.

Настроить запрос, его протестировать, организовать механизмы макроподстановки позволяет закладка с названием "SQL". Ее содержимое становится доступным, когда в выборе типа источника на странице "Базы данных" (рисунок 2.1.1) вбирается опция "Запрос" или опция "Запрос и таблица".

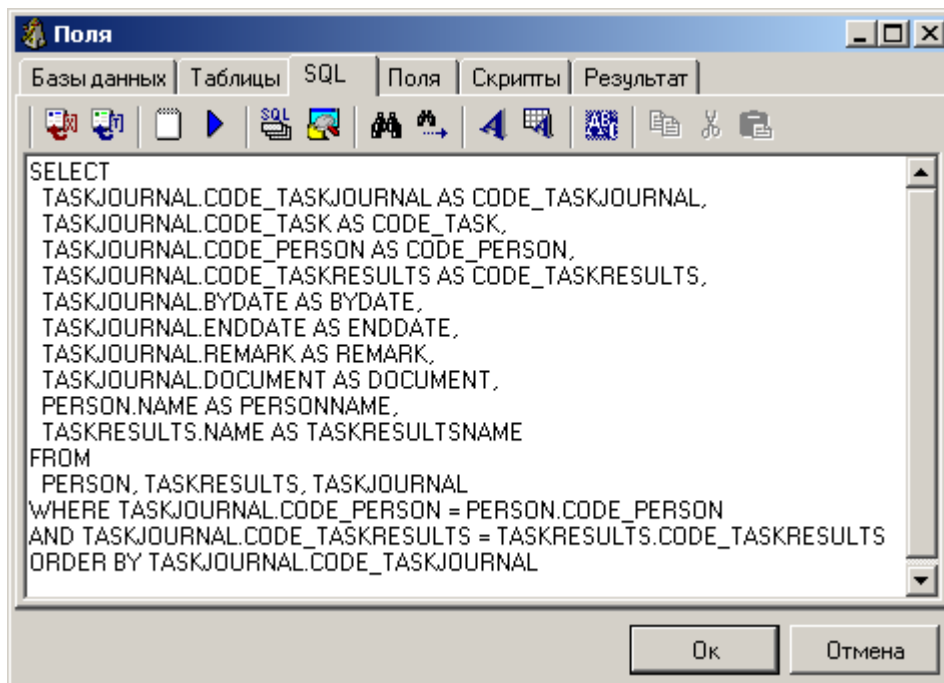











Рис 2.2.5. Пример написания SQL-запроса для извлечения данных.

Основной элемент закладки – это большая область ввода текста. Остальные кнопки - это необходимый сервис для подстановок и проверок, улучшения внешнего вида, вызова других полезных программ.

Картинка	Функция	Описание
	Вставить переменную	Из списка выбирается имя доступной в запросе внутренней переменной окружения.
	Вставить тэг	Вставляется инструкция для обозначения вставки SQL-конструкций другими компонентами системы.
	Показать SQL-текст	Показывает окно с текстом SQL-запроса с уже подставленными значениями. Предназначено для тестирования.
	Попробовать открыть	Выполняется попытка выполнения SQL-запроса в отдельном окне.
	Собрать текст запроса	На основании перечня таблиц и их связей автоматически собирается SQL-запрос

	SQL Explorer	Вызывается программа под названием “SQL Explorer”, если таковая имеется на компьютере. Поставляется для работы с BDE.
	Поиск	Производится поиск фрагмента теста внутри SQL-запроса. Найденное вхождение выделяется.
	Продолжить поиск	Продолжается поиск найденного фрагмента. Выполняется его выделение.
	Увеличить шрифт	Увеличивается шрифт текста SQL-запроса
	Уменьшить шрифт	Уменьшается шрифт текста SQL-запроса
	Выделить все	Весь тест выделяется
	Копировать текст	Копирование выделенного текста запроса в буфер универсального обмена.
	Вырезать	Вырезать выделенный текст запроса в буфер универсального обмена.
	Вставить	Вставка теста из буфера обмена.

**Таблица 2.2.2.** Функции для подготовки SQL-запросов.

## 2.3 Формирование полей потока

Настройка полей источника – это одно из самых важных мест в разработке программного обеспечения программ семейства ETNI. Благодаря настройке поля система сама определяет, как отображаются, проверяются и обрабатываются данные. Свойства полей имеют составной характер и могут содержать сколь угодно переключателей. Полный список переключателей находится в правом списке. Список полей в левом. Описание поля содержит следующие колонки

- ✓ Поле
- ✓ Метка
- ✓ Ширина
- ✓ Порядок
- ✓ Свойства



Поля могут добавляться автоматически по нажатию функции “Показать поля таблицы”. Далее остается только подписать их названия на русском и поставить при необходимости свойства в нужное поле двойным щелчком на выбранном переключателе из списка.



Невидимые поля можно и не подписывать. Работа с полями происходит эволюционно. Их можно сначала выводить в самом простом варианте, и со временем, улучшать их обработку. Такой подход позволяет системе постепенно эволюционировать, не прибегая к изменению исходно кода.

Первая колонка “Поле” содержит название поля в базе данных, вторая “Метка” – это ее подпись, Ширина дает возможность указывать размер для поля редактирования, Порядок говорит о том, в какой последовательности поля выводятся диалоговом окне. Если два поля имеют один и тот же порядок, то они выводятся одно за другим на той же строке. В свойствах записывается поведение поля в системе. Вариантов тут много. Ниже в таблице 2.2.1 описан их полный перечень.

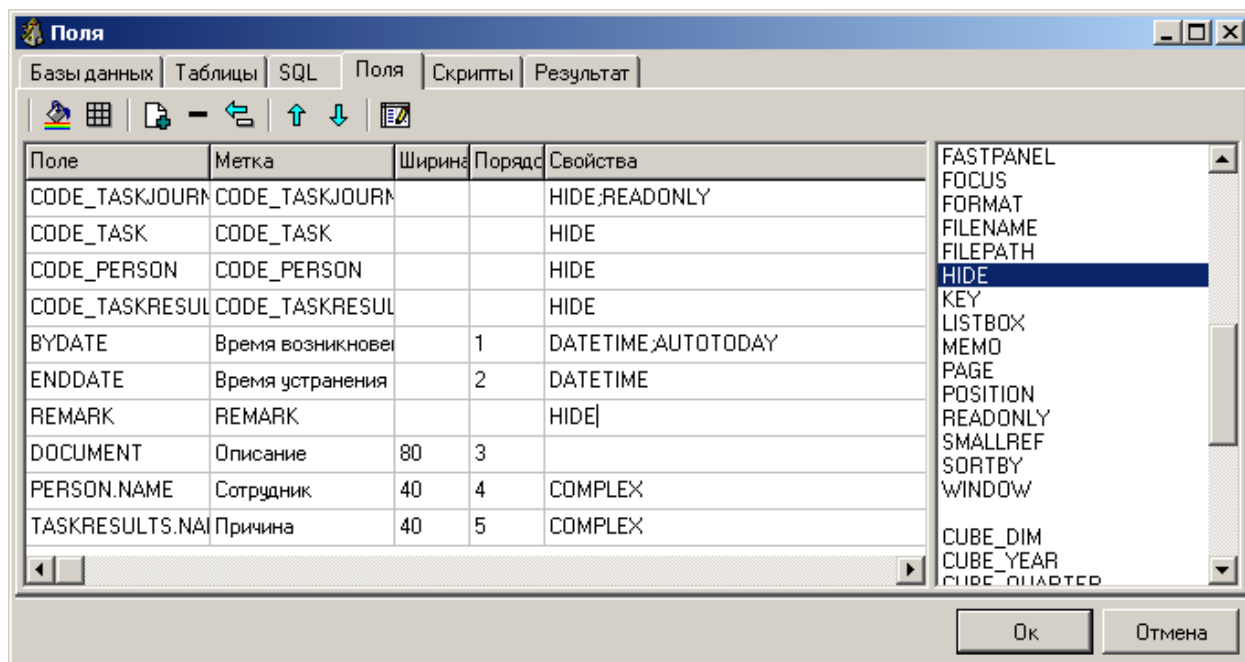


Рис 2.3.1. Настройка полей источник.

Все переключатели неявно разбиты на отдельные блоки

- ✓ Для таблиц и окон редактирования
- ✓ Для многомерных кубов
- ✓ Для графиков

Значение	Описание
AFTEROK XXX	Проставление значения поля после редактирования и вставки. Подставляется после нажатия клавиши ОК окна редактирования. XXX – это значение поля.
AFTERCHANGE XXX-YYY	Проставление значения другого поля на изменение текущего, где XXX- имя поля, YYY- новое значение. Например, AFTERCHANGE HAND-Да
ASKOK	Переспросить пользователя на сохранение данных ASKOK XXX-YYY, где XXX - сообщение YYY - значение для вставки. Если получаемое значение YYY пустая строка, то вопрос не задается, значение не подставляется. Когда не требуется вообще ничего подставлять, даже после вопроса, то ставьте этот переключатель для поля, у которого REAONLY. Если такого поля не имеется в наборе, но добавьте дополнительное поле с переключателем CALCULATED.
AUTOCODE	Автоинкремент
AUTONUMBER XXX	Автонумерация XXX – имя поля даты
AUTOTODAY	Сегодняшняя дата
AUTOZERO	Автоматически 0

CALCULATED	Вычисляемое поле. Обычно идет для подписи в комбинированном с таблицей запросе
CALLMENU	По значению этого поля вызывается пункт меню. В этом поле указывается либо код меню, либо название пункта меню через точку Группа_меню.Пункт_меню Например: Справочник.Предприятия
CALLSITE	По значению этого поля открывается Интернет-страничка
CALLEMAIL	По адресу этого поля отправляется письмо
CHECKBOX	Логическое поле
COMPLEX XXX	Выбор из связанной таблицы, где XXX имя поля для связи в связанной таблице.
CALCULATED	Это поле не участвует в редактировании, только в запросе
CURRENCY	Деньги. Дает 2 знака после запятой
DATETIME	Обработка времени в диалоговом окне
DEFAULT XXX	Выставить значение по умолчанию XXX любое
EDITMASK	Маска XXX для редактирования, вместо “;” надо ставить “~”
FASTPANEL	Поле выводится в дополнительной нижней панели. Это может быть обычное поле, может быть мемо-поле. Если это ссылка на файл изображения, то на панель будет автоматически подгружаться картинка. Все поля располагаются в ряд, прижимаясь к левому краю.
FOCUS	На это поле передается фокус диалогового окна
FILENAME XXX	Занесение имя поля. Второй вариант FILENAME XXX, где XXX - новый путь, куда копируется выбранный файл. Путь в поле заменяется на новый. Возможно использование макроподстановок. Например \\\$SERVER_NAME\$D:\PICTURES\CODE_HORSE.JPG В результате подстановки будет получен путь \\Server2\D:\PICTURES\1232.JPG, который пропишется в поле базы данных
FILEPATH XXX	Путь к папке. XXX – либо абсолютный путь (Например, C:\Temp\ ), либо в виде переменной, либо ее использованием (Например, [%APP_PATH%]) Используется совместно с FILENAME. В таком случае, из поля берется полное имя файла, из него вырезается путь и склеивается с подставленным в FILEPATH
FORMAT XXX	Форматирует поле XXX строка формата, например 000 000.00 литров, ##0.00 (0 – обязательная цифра, # пробел если 0)
GRAPH_TITLE	Формирование подписи к графику по значению поля
GRAPH_TYPE	Тип вида отдельной серии графика для этого поля от 0 до 5
HIDE	Сделать поле невидимым
INSERTONLY	Только вставка (не реализовано)
IFEMPTY	IFEMPTY XXX Перед началом редактирования, если поле пустое (NULL), то ему присваивается значение XXX
JUMP XXX-YYY-ZZZ	Формирование ссылки в PHP Browser. XXX – имя нового окна. ZZZ – имя поля в новом окне, в которое подставляется значение поля YYY текущего окна
KEY	Ключевое поле. Их может быть несколько.
LISTBOX XXX-XXX-XXX	Выпадающий список тестового поля. XXX-возможные значения, разделяемые ‘-’
MEMO XXX	Мемо – поле XXX – количество строк
NOTNULL	Поле обязательное к заполнению
PAGE XXX	Закладка в окне редактирования, должно быть указано имя закладки
POSITION	Поля, по которым происходит поиск после вставки
READONLY	Только чтение
SMALLREF	Выпадающий список для связанной таблицы
SIMETA_SORTBY	Переключение сортировки на другое поле. SORTBY XXX, где XXX-новое поле сортировки.
WINDOW XXX-YYY	Вызов окна приложения, где XXX – имя окна YYY – имя результирующего поля. Если поле не указано, то берется поле связи

Таблица 2.2.1 Описание переключателей для источника.

Следует отметить, что наиболее полную реализацию дает приложение ETNI Platform, для других приложений некоторые функции могут не работать в виду особенности архитектуры.



Поля из связанных таблиц пишутся вручную, причем нужно указывать алиас связанной таблицы и затем уже имя поля через точку. Например, Company.Name

Картинка	Функция	Описание
	Показать поля таблицы	Происходит автоматическое формирование списка полей. Выводятся все поля, за исключением тех, что находятся в соседних связанных таблицах. Если поле уже есть, то оно больше не добавляется.
	Очистить список полей	Удаляет все поля и связанную с ними информацию.
	Добавить поле	Формируется еще одна строка в списке полей.
	Удалить поле	Удаляется выбранная строка вместе со всей настроенной информацией.
	Добавить переключатель	Вставляется или убирается выбранный переключатель. Если он есть, то происходит его вырезание из строки. При вставке сразу же добавляется разделитель между переключателями.
	Строку вниз	Перемещает выбранную строку выше. Верхнюю опускает.
	Строку вниз	Перемещает выбранную строку вниз. Нижнюю поднимает.
	Окно редактирования	Выводит тестовое окно редактирования. Так нагляднее можно проставлять порядок полей, их ширину.

Таблица 2.2.2 Описание функций работы с полями.

## 2.4 Подключение скриптов на стороне клиента

Работа наших приложений построена таким образом, чтобы создавать системы, не прибегая к программированию пользовательских интерфейсов. Но, не смотря на то, что около 95% функционала является стандартным, остается еще 5% из которых 3% можно не делать, но остается 2% или даже 1%, но очень важный и специфический, который обязателен к исполнению. Для таких редких случаев и организовано подключение скриптов на языках Pascal и SQL.

Скрипты можно назначить на следующие события источника:

- ✓ Открытие
- ✓ До и после редактирования
- ✓ До и после добавления
- ✓ До и после удаления
- ✓ На изменение записи (перехода на следующую запись)

Возможно повторное использование кода. Можно один раз написать процедуру и подключать ее к разным источникам.

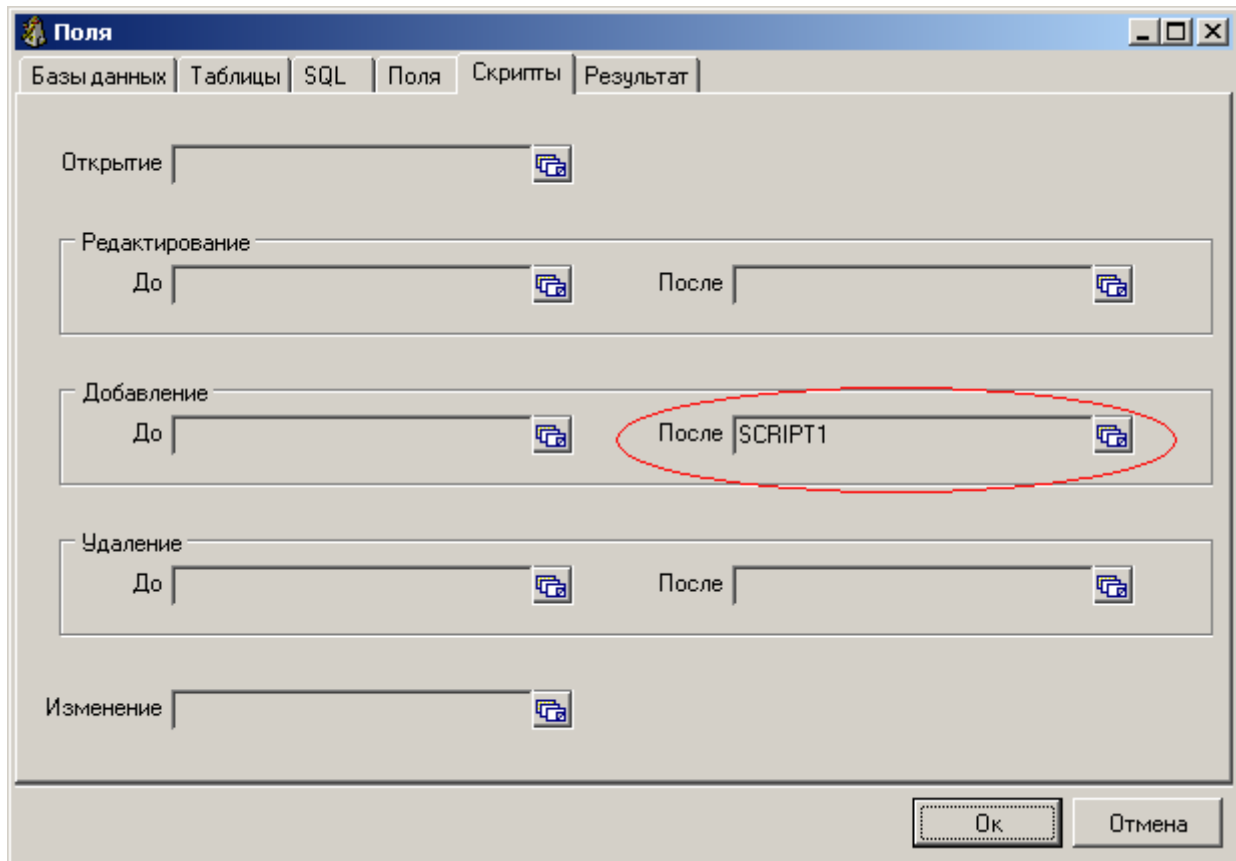
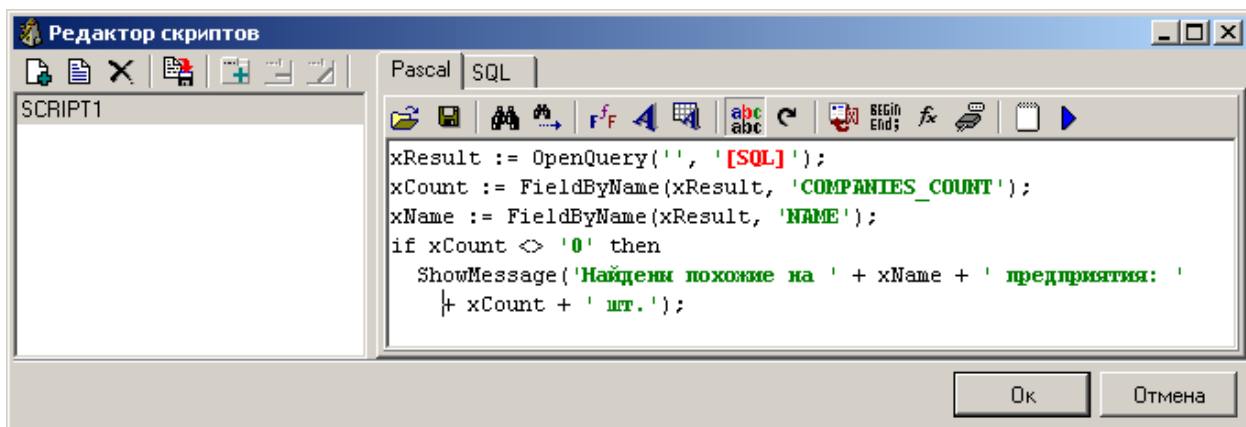


Рис 2.4.1. Настройка подключения скриптов.

В окне выбора скрипта можно сразу же его отредактировать или создать новый. Подробнее о работе со скриптами читайте в специальной главе настоящего руководства. На рисунке 2.4.2 показан пример скрипта. Здесь происходит проверка существования похожих предприятий в базе данных. Выводиться предупреждающее сообщение такого вида: "Найдены похожие предприятия: 5 шт.".



Скрипт подключен на завершение добавления записи. На второй закладке располагается запрос, написанный на языке SQL. Он выглядит следующим образом:

```
SELECT COMPANIES_COUNT AS COMPANIES_COUNT, NAME AS NAME
FROM SPR_FIND_COMPANY([NAME]);
```

## 2.5 Тестирование настроек

Сразу же, не выходя из режима редактирования компонента, можно протестировать получаемый результат. Особенно это важно для случаев, когда источник наглядно не виден (кубы, графики, отчеты). На последней закладке (рисунк 2.5.1) есть возможность запустить в действие настроенные таблицы или SQL-запрос

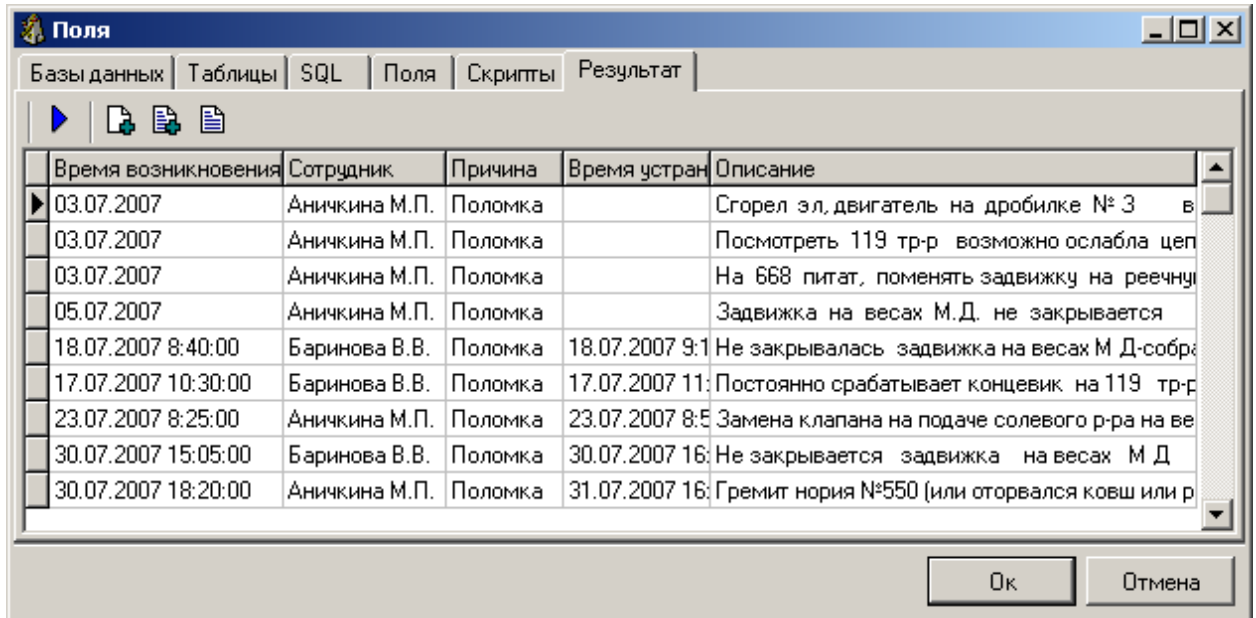


Рис 2.5.1. Тестирование получаемых результатов.

Здесь же проверяется добавление записей в таблицу, редактирование, добавление по образцу.

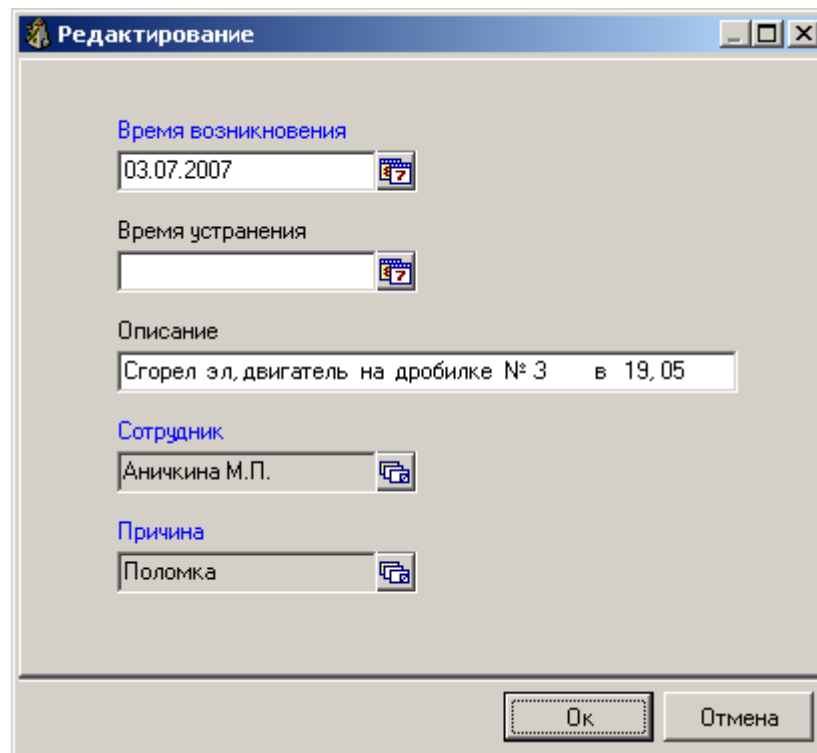






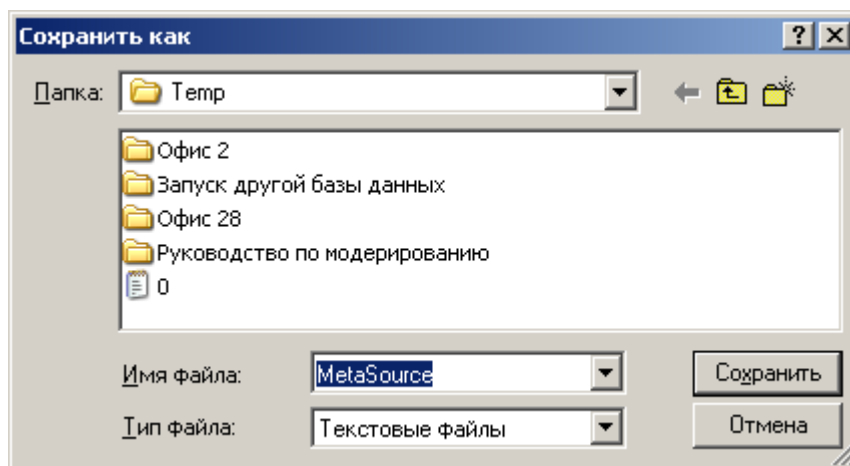
Рис 2.3.2 Тестирование окна редактирования.

Картинка	Функция	Описание
	Показать данные	Происходит подготовка и открытие источника. Если он уже был открыт, то происходит его предварительное закрытие
	Добавить	Добавление записи. Появляется окно для ввода информации в соответствии с настройкой полей.
	Добавить по образцу	Добавление записи с проставленными значениями из текущей.
	Редактировать	Редактирование выбранной записи. Действие дублируется из предшествующей закладки.

**Таблица 2.5.1** Тестирование источника. Описание функций.

## 2.6 Экспорт/Импорт источника

Иногда требуется переносить настройки источника из одной системы в другую. Особенно это бывает важно при тестировании отдельных узлов приложения. Для этих целей на первой закладке редактора компонента (рис 2.1.1) предусмотрено две кнопки “Экспорт” и “Импорт”. Каждая кнопка выводит диалог по выбору файла, Одна кнопка для открытия, вторая для сохранения. Файлы представляют из себя простые текстовый файлы со структурой ini.и могут легко читаться, в том числе, и web- приложениями.



**Рис 2.6.1.** Экспорт источника в текстовый файл.

Работа источника проявляется во многих компонентах, поэтому и требует такой сложной и точной настройки. Рекомендуем использовать его поэтапно от простого к сложному. Некоторые функции могут не поддерживаться настольными СУБД. Наиболее удобно его применение именно для SQL-серверов. В нашей компании основные внедрения проходят на Firebird и Oracle для небольших и крупных многопользовательских приложений.

Перечень переключателей постоянно пополняется, в тестовом режиме их не сразу вносят в список, но их можно добавлять путем прямого ввода. Допускается использование собственных переключателей при программировании новых систем на базе компонента TSIMetaSource,

### 3 Внутренние связи.

Внутренние связи используются во многих местах настройки приложений. Задача этого компонента - это фильтрация и макроподстановки источника. В зависимости от того, какой тип источника используется, рекомендуется делать либо подстановку, либо фильтрацию.



Не рекомендуем использовать механизмы фильтрации источника, если он работает с SQL-запросами. При таком подходе система полностью копирует результат запроса на рабочую станцию. Таблица же наоборот делает фильтрацию на стороне сервера.

Смысл связей в том чтобы правильно “подхватить” поступающую информацию и в нужный момент осуществить правильное отображение данных. Эта информация может поступать как от других окон, так же может быть статически прописана в редакторе компонента, может поступать из переменной приложения, может получать значения в результате диалога с пользователем, выбором из другого окна. Такой компонент – это возможный сервис вокруг источника.

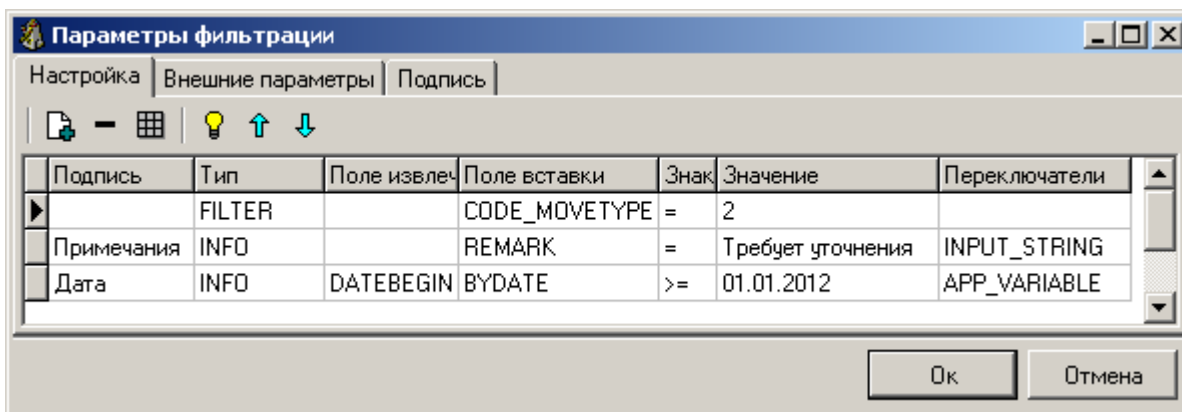


Рис 3.1. Настройка фильтров и подстановок.

На первой закладке “Настройка” объявляются активные параметры. Если эти параметры пришли извне, то их можно просмотреть на второй закладке и на третьей закладке происходит макроподстановка подписи. Обычно это подпись окна, но возможны и другие варианты использования. Этот компонент позволяет связывать окна между собой. За счет того, что в нем накапливается информация из предыдущего окна, то связываемые окна становятся независимыми.



Все параметры и их значения можно просматривать в режиме реального времени. Это дает возможность разработчику легко отлаживать взаимодействие окон и добавление новых параметров.

Для подписи параметра можно использовать любые слова и фразы, причем писать их можно не только латинскими буквами, но и кириллицей. В случае взаимодействия с пользователем, эта надпись будет отображаться рядом с вводимым с клавиатуры параметром. Когда происходит макроподстановка внутри SQL-запроса, то эта надпись вставляется внутри квадратных скобок.

Тип параметра может быть одним из трех типов: FILTER, INFO, SPECIAL. Если ставится фильтр, то источник будет отфильтрован по полю, которое указано в колонке “Поле вставки”. Если ставить INFO, то происходит просто макроподстановка в запрос, SPECIAL тоже дает макро подстановку, но собирает свои значения в предыдущем окне (его добавлять не следует, это происходит автоматически).

Поле извлечения проставляется при необходимости, но, как правило, его можно не проставлять. Важнее поле вставки. Это поле должно быть в источнике при фильтрации, в противном случае возникает исключительная ситуация.

Что касается знака для связи, то наиболее часто применяется знак равенства "=" (если это связи с кодами), для диапазона дат больше подходит "<=" и ">=", На сегодняшний день знаки могут быть следующие: "=", ">", "<", "<>", ">=", "<=". В перспективе будут добавлены операторы внутреннего вхождения.

Поле "Значение" проставлять, обычно не требуется. Система сама его проставляет динамически. Полезно его просматривать для отладки и контроля. Но есть случай, когда его нужно прописать, когда это - статические параметры. То есть их значения ниоткуда не приходят – их прописывает разработчик один раз. Как раз на рисунке 1.3 первый параметр имеет такое свойство. Для настраиваемого журнала устанавливается поле CODE\_MOVETYPE = 2. таким образом, всегда фильтруются записи с таким типом.

Переключатели служат для изменения поведения параметра. Первый вариант APP\_VARIABLE передает обработку приложению и только потом начинает его вставку в источнике. Такие названия параметров зарезервированы. Например: DATEBEGIN DATEEND HOST USERNAME. Так же с этим переключателем вставляются и переменные окружения. Обращаем ваше внимание, что в редакторе внутренних связей квадратные скобки у названий параметров не ставятся.

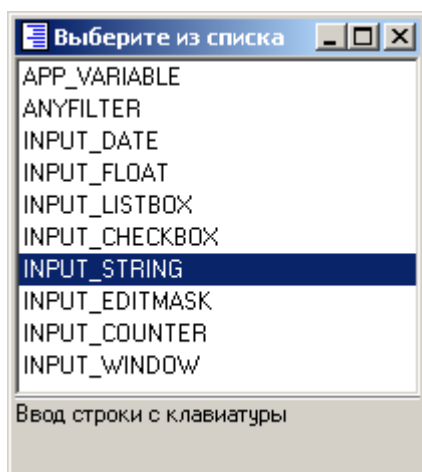


Рис 3.2. Выставление переключателя параметра связей.

Ниже приведена таблица с описанием поведения по каждому переключателю. В новых версиях эти переключатели могут наращиваться.

Значение	Описание
APP_VARIABLE	Переменная приложения
ANYFILTER	Установка произвольного фильтра. Можно с макроподстановкой переменных
INPUT_DATE	Ввод с клавиатуры даты
INPUT_FLOAT	Ввод с клавиатуры числа
INPUT_LISTBOX	Выпадающий список из фиксированного списка через тире, вида LISTBOX XXX-YYY-ZZZ
INPUT_CHECKBOX	Ввод логической величины Да/Нет
INPUT_STRING	Ввод строки с клавиатуры
INPUT_EDITMASK	Маскированный ввод. Например, INPUT_EDITMASK !90.00,00~1~
INPUT_COUNTER	Ввод целого числа. Можно указать минимум и максимуму. Например, INPUT_COUNTER 0-100
INPUT_WINDOW	Выбор окна параметры: Код_окна - Строка_подписи - Поле ключа

Таблица 3.1 Описание переключателей для параметров внутренних связей.



Вторая закладка сама за себя говорит. Здесь можно посмотреть все, что пришло снаружи (от другого окна). Менять тут ничего нельзя. Следует обратить внимание, нет ли где типа "FILTER". Если есть, то он будет срабатывать вне зависимости от настройки в этом редакторе. Каким будет фильтр можно узнать, нажав "Показать фильтр". Еще есть функция "Подготовить". Она используется крайне редко, когда параметры изменились, но еще не сформированы. Функция может быть полезна в режиме настройки. На рисунке 3.3 показано, как можно просматривать поступающие параметры.

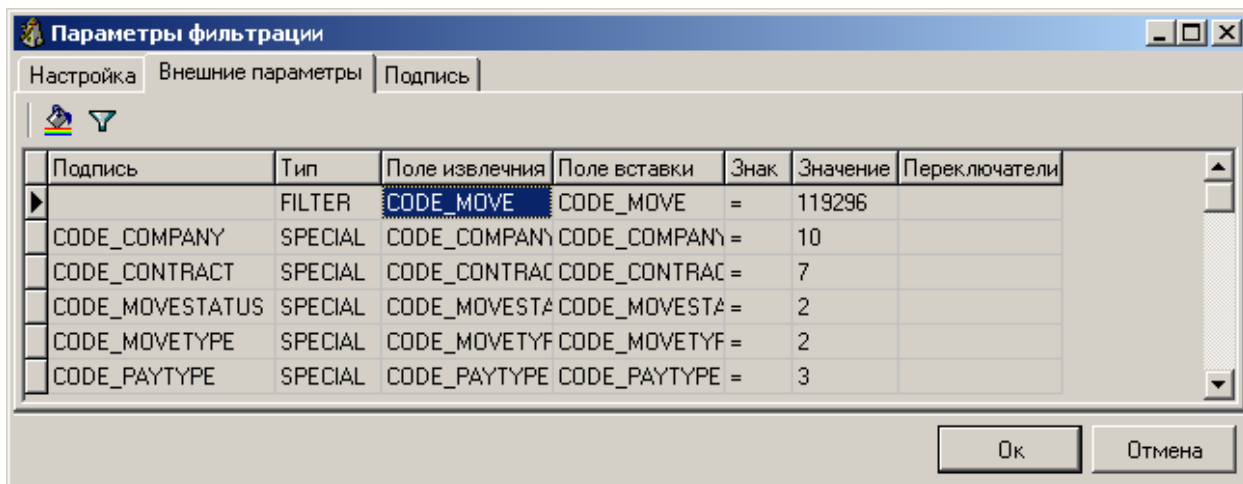


Рис 3.3. Передача параметров извне.

Последняя закладка описываемого редактора носит декоративный или оформительский характер. Бывает, что в настраиваемом приложении открывается дополнительное окно и в его заголовке требуется отобразить дополнительную информацию о записи в предыдущем окне. Например, номер накладной, ее дату, название контрагента и так далее. Именно тут это и настраивается. Вам доступны значения всех полей. Выберите подходящее поле, нажимая кнопку, справа от области ввода. Если вручную набивать название полей, то можно и ошибиться. Система предлагает выбрать из списка и вставит в то место, где находился ваш курсор до появления списка. Выбранная переменная автоматически заключается в квадратные скобки.

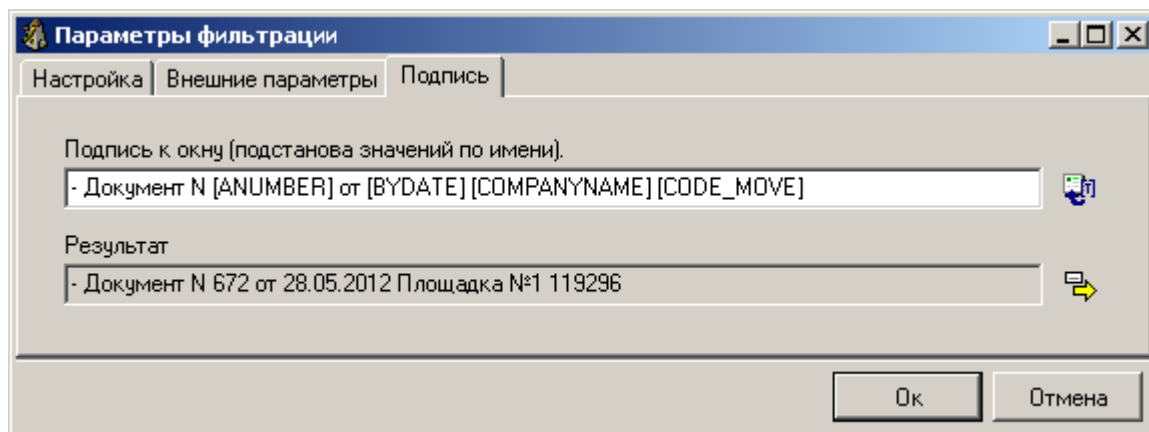


Рис 3.4. Настройка подписей (окна) с макроподстановкой.

Тут же не закрывая редактора компонента, можно проверять получаемый результат, если нажать кнопку справа от результирующей строки. При этом появляется сообщение, что компонент приходится сохранить. Соглашайтесь - не пожалеете.

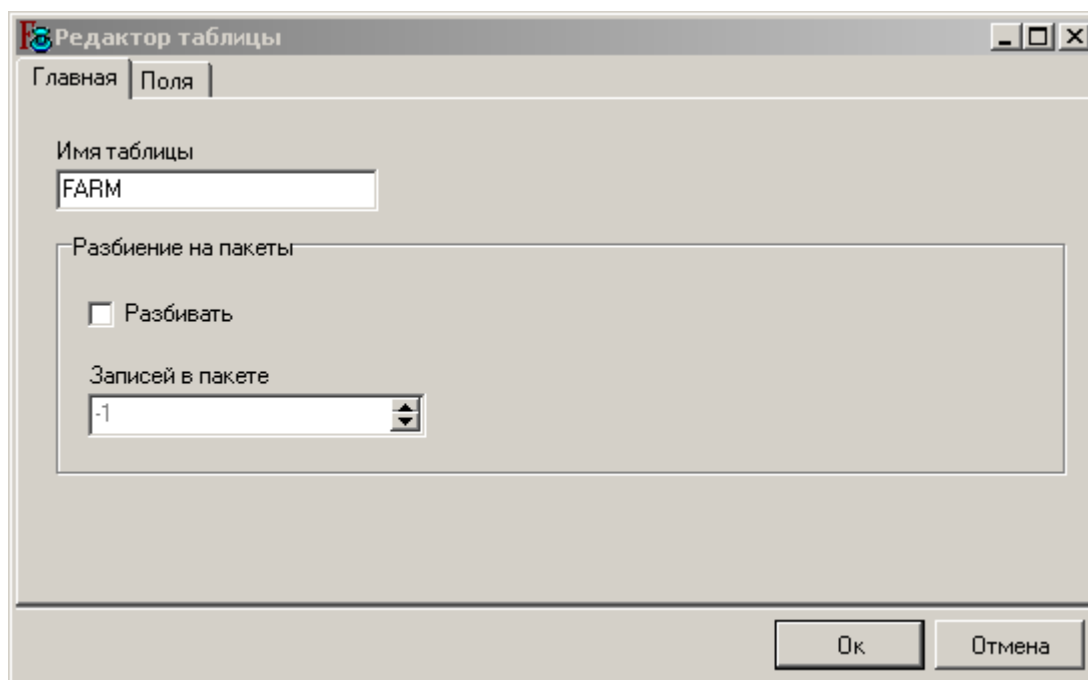
## 4 Виртуальные таблицы.

Виртуальные таблицы используются для эмуляции БД в оперативной памяти компьютера. Для этого они размещаются в так называемых виртуальных базах данных. Это позволяет оперировать данными так как-будто вы работаете с реальной базой данных. Однако виртуальной таблице присущи следующие ограничения:

- Не поддерживается работа с языком SQL
- Не поддерживается ссылочная целостность

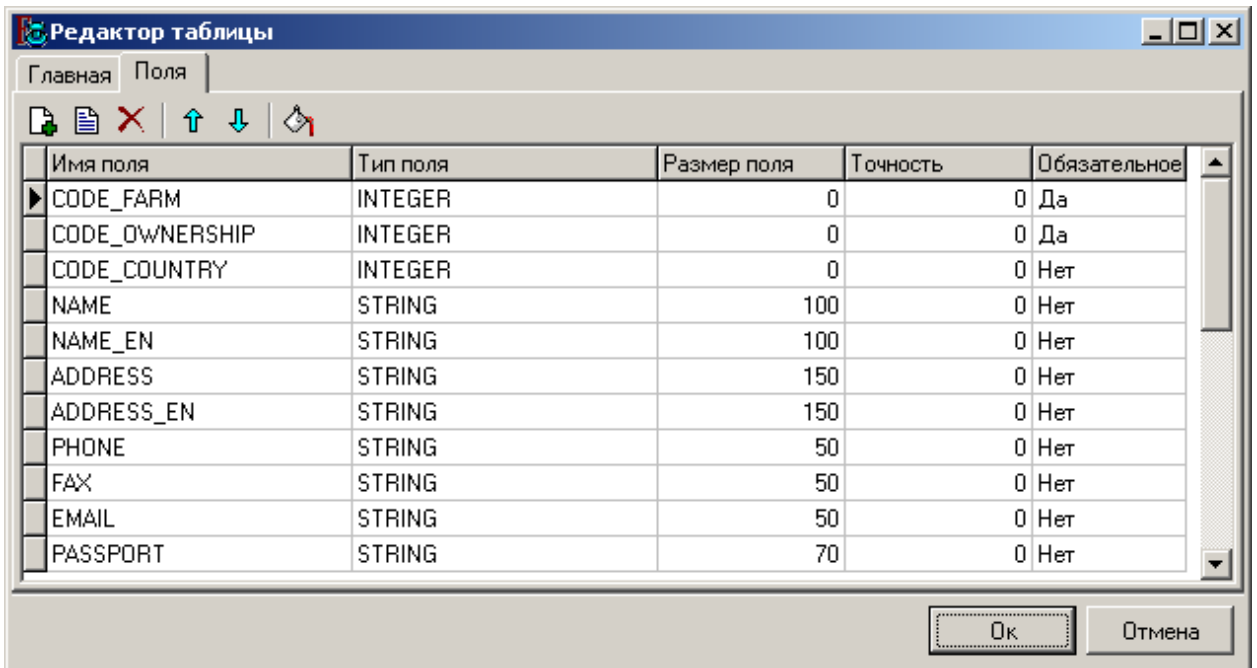
Виртуальные таблицы можно сохранить в файл в различном формате (в основном XML), однако в отличие от обычных таблиц изменения сохраняются в файл не при каждом изменении, а только по требованию пользователя.

При создании таблицы необходимо указать её имя. При сохранении в файл таблица может быть разбита на пакеты, размер которых указывается на главной закладке. Эта возможность может быть полезна для очень больших таблиц, так как позволяет уменьшить потребление оперативной памяти при их загрузке (проявляется при работе с etni файлами на web серверах).



**Рис 4.1.** Настройка имени виртуальной таблицы и разбиение на пакеты.

На закладке поля настраивается структура виртуальной таблицы – список полей, их тип, размер, точность, является ли поле обязательным (можно ли оставить значение null). Размер поля можно указывать только для типов полей STRING, CURRENCY. Для типа CURRENCY и других вещественных типов можно ещё указать точность.

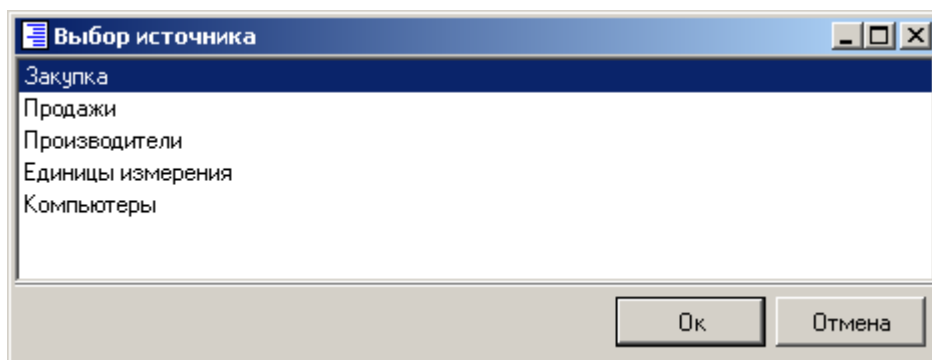


**Рис 4.2.** Настройка структуры виртуальной таблицы.

Картинка	Функция	Описание
	Добавить	Добавление поля
	Редактировать	Редактирование поля
	Удалить	Удаление поля
	Переместить вверх	Перемещение поля на одну позицию вверх
	Переместить вниз	Перемещение поля на одну позицию вниз
	Заполнить	Заполнение структуры на основе настроенного источника данных

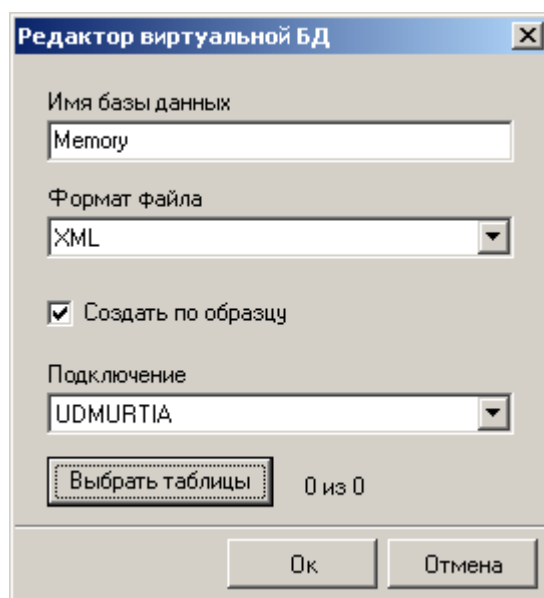
**Таблица 4.1** Настройка структуры. Описание функций

Часто бывает необходимо создать структуру таблицы из настроенного источника данных. Для этого можно использовать функцию заполнить, которая отобразит окно с выбором источника, по которому необходимо создать таблицу.



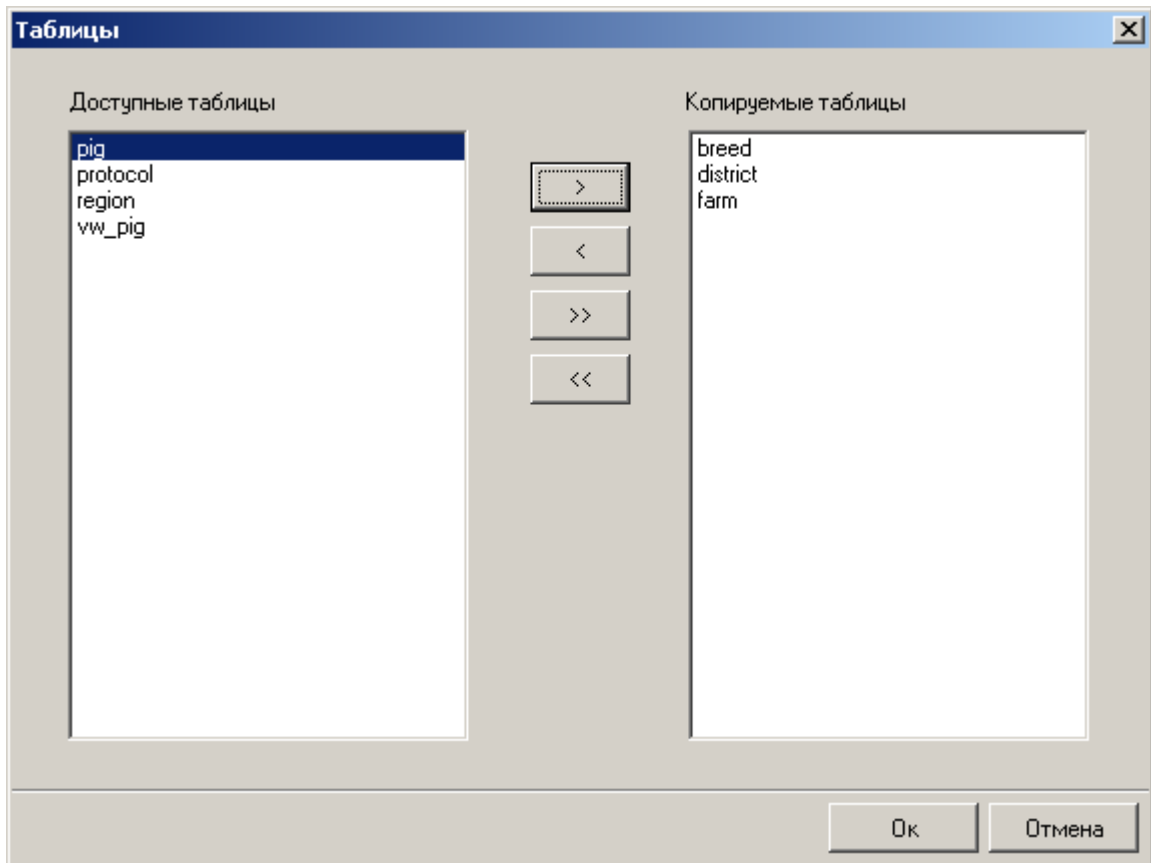
**Рис 4.3.** Выбор источника из списка для создания виртуальной таблицы.

Виртуальные таблицы обычно содержатся внутри так называемых виртуальных баз данных. При создании виртуальной базы данных необходимо указать её имя и формат хранения в файле. Формат хранения может быть типа XML или TXT. По умолчанию база данных хранится в формате XML. При этом в папке (она может быть упакована, например, в etni файл) для хранения БД будет создан файл database.xml с описанием всех таблиц входящих в виртуальную базу данных и файлы с таблиц в формате <имя таблицы>.xml (или TXT).



**Рис 4.4.** Создание виртуальной базы данных.

Часто требуется создать копию структуры реальной базы. Это можно сделать, установив галочку «Создать по образцу», после чего выбрать подключение к базе данных, из которой происходит копирование, и нажать кнопку выбрать таблицы для выбора списка копируемых таблиц. После выбора списка таблиц и нажатия кнопки ОК будет создана копия структуры базы данных.



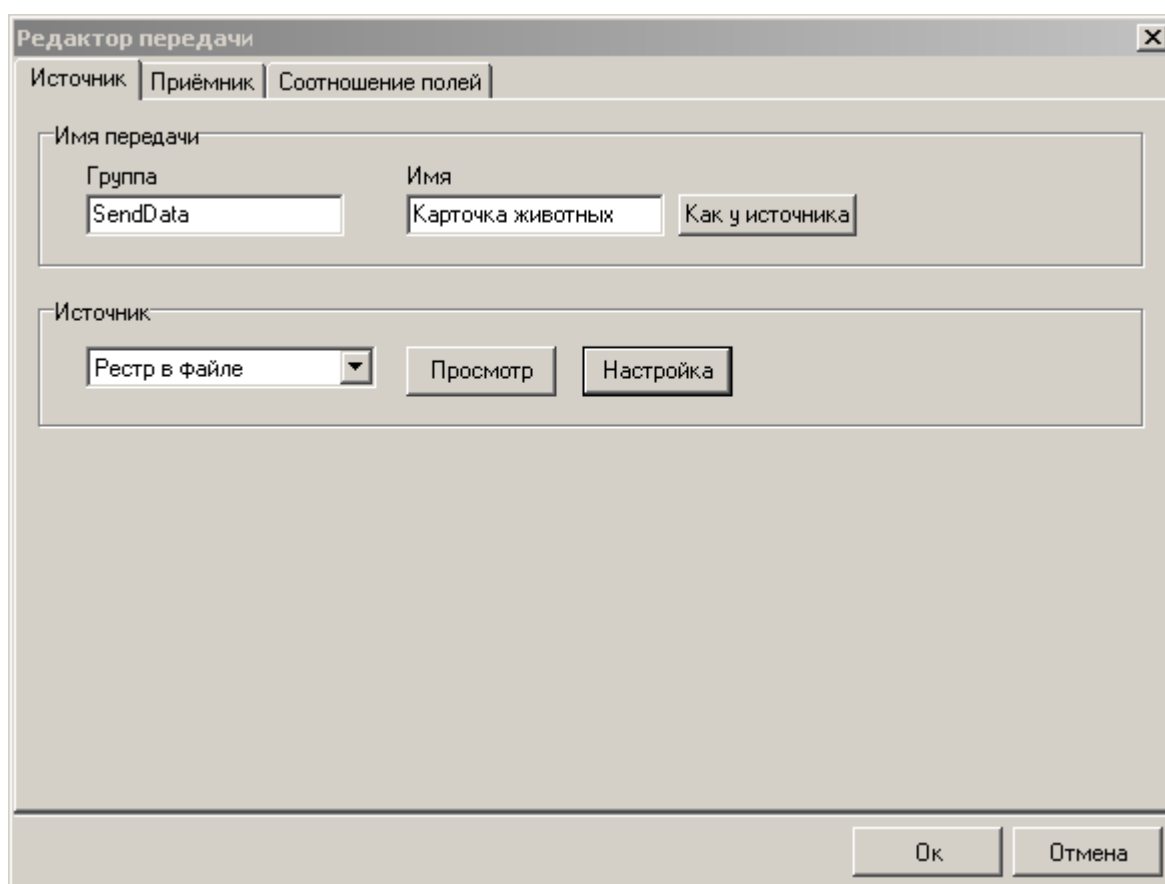
**Рис 4.5.** Выбор таблиц для копирования структуры.



Для работы с виртуальными таблицами в источниках данных необходимо использовать технологию доступа CDE. Тип источника при этом может быть только таблицей. Виртуальные таблицы удобно использовать совместно с компонентом «Передача данных» для организации асинхронной репликации данных.

## 5 Передача данных.

Данный компонент предназначен для передачи данных между различными базами данных. Причём в качестве базы данных может выступать как обычная, так и виртуальная база данных (база данных в памяти). Это позволяет настраивать так называемую асинхронную репликацию данных.

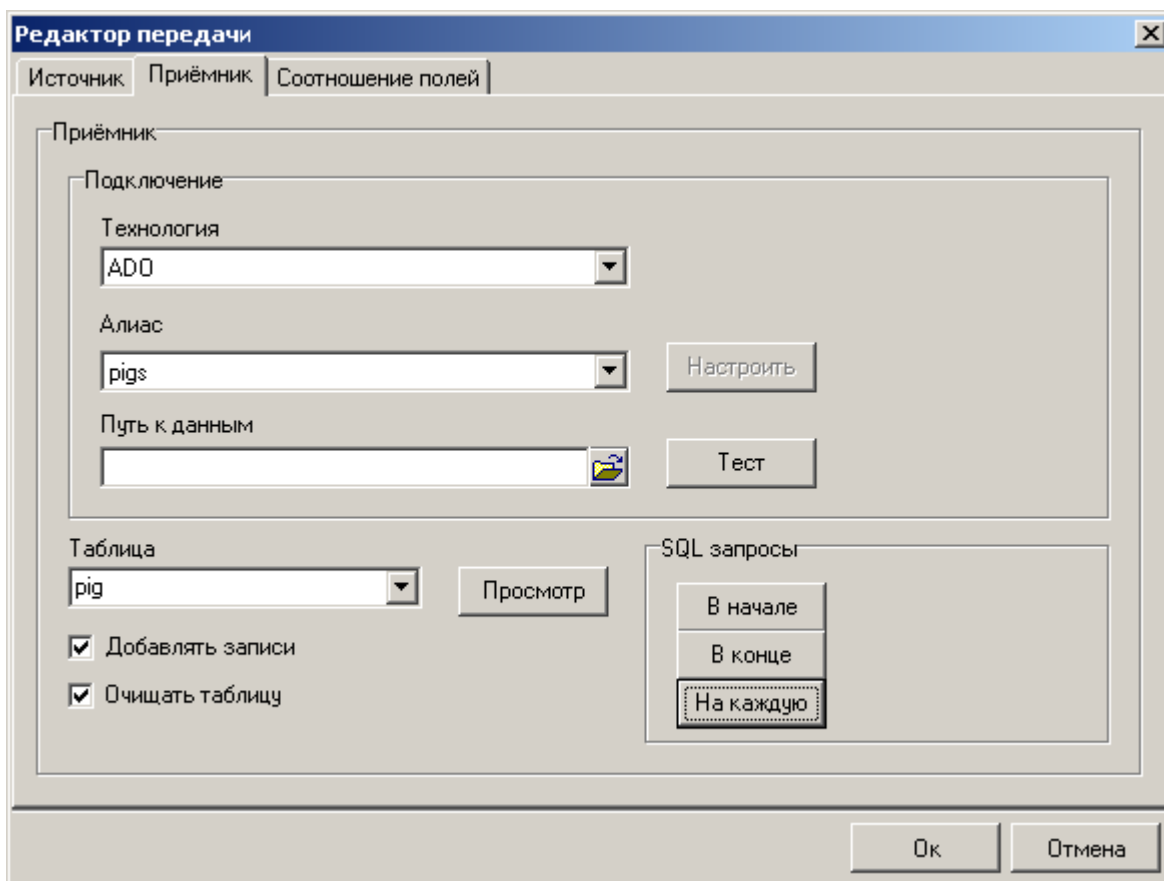


**Рис 5.1.** Настройка источника для передачи.

При настройке передачи необходимо задать её имя и группу. Группа может потребоваться в приложении для объединения нескольких передач для их совместного запуска. Также необходимо выбрать источник из заранее сформированного списка (формируется приложением). Выбранный источник можно настроить и просмотреть данные, которые он возвращает. Если необходимо назвать передачу также как и источник, то можно нажать на кнопку «Как у источника».



Будьте осторожны при отладке этого компонента. При проверке заливки данных предыдущие данные могут быть удалены если стоит галочка «Очищать таблицу» или написан соответствующий запрос. Делайте копию базы данных или файла с настройкой виртуальных таблиц при отладке данного компонента.



**Рис 5.2.** Настройка приёмника для передачи.

При настройке приёмника необходимо выбрать подключение (база данных в которую необходимо передать информацию).

В качестве технологии доступа можно выбрать одну из следующих:

- ✓ BDE
- ✓ ADO
- ✓ CDE
- ✓ ClientDataset
- ✓ Custom

Все эти технологии перекрывают возможные варианты использования описываемого компонента. Обычно, для стандартных баз данных используется BDE или ADO, для доступа к таблицам в оперативной памяти тип CDE (Container Dynamic Engine). ClientDataset поддерживает обмен данными с DLL-функциями (так же общее назначение в составе специальных приложений, написанных на Delphi). Все, что не классифицируется в предыдущих вариантах, подключается как Custom (в том числе специальные компоненты прямого доступа). Последний вариант подключения полностью передает управление стороннему источнику и отвечает только за подпись полей (подключение к базе данных делается отдельно).

Технологию BDE описывать нет смысла – она используется во многих системах. Достаточно указать только имя алиаса в источнике.

Технология ADO применяется в двух вариантах. В одном случае выбирается алиас ADO по имени, во втором можно организовать доступ прямой настройкой строки соединения стандартным редактором.

Для использования предустановленных строк ADO можно дополнительно задавать путь к каталогу, При этом в списке алиасов выбираются следующие варианты

- ✓ DBASE
- ✓ ACCESS
- ✓ EXCEL
- ✓ CUSTOM

Для первых трех в списке достаточно указать путь – система сама правильно подберет драйвер, для всех остальных случаев доступа через ADO следует выбирать CUSTOM. После того как соединение настроено, его можно протестировать командой “Тест”.

После настройки подключения необходимо выбрать таблицу, в которую будет осуществлён перенос данных. Содержание этой таблицы можно просмотреть, нажав кнопку «Просмотр». Чаще всего перед копированием данных в таблицу её необходимо очистить от прежних данных. В этом случае необходимо установить галочку «Очищать таблицу».

Если перед началом передачи необходимо выполнить некоторый SQL запрос, то необходимо ввести его, нажав кнопку SQL запросы «В начале». При этом появится стандартный диалог редактирования SQL запросов. Если SQL запрос не введён (пуст), то он не выполняется. То же самое можно сделать, если требуется выполнить запрос по завершению передачи (кнопка SQL запросы «В конце»).

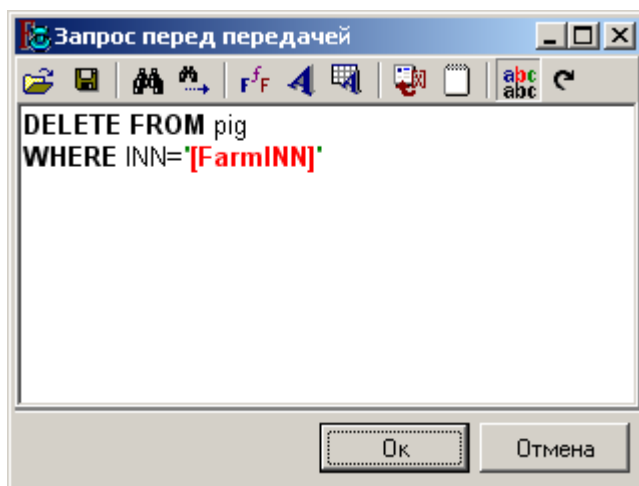


Рис 5.3. Редактор SQL запроса.

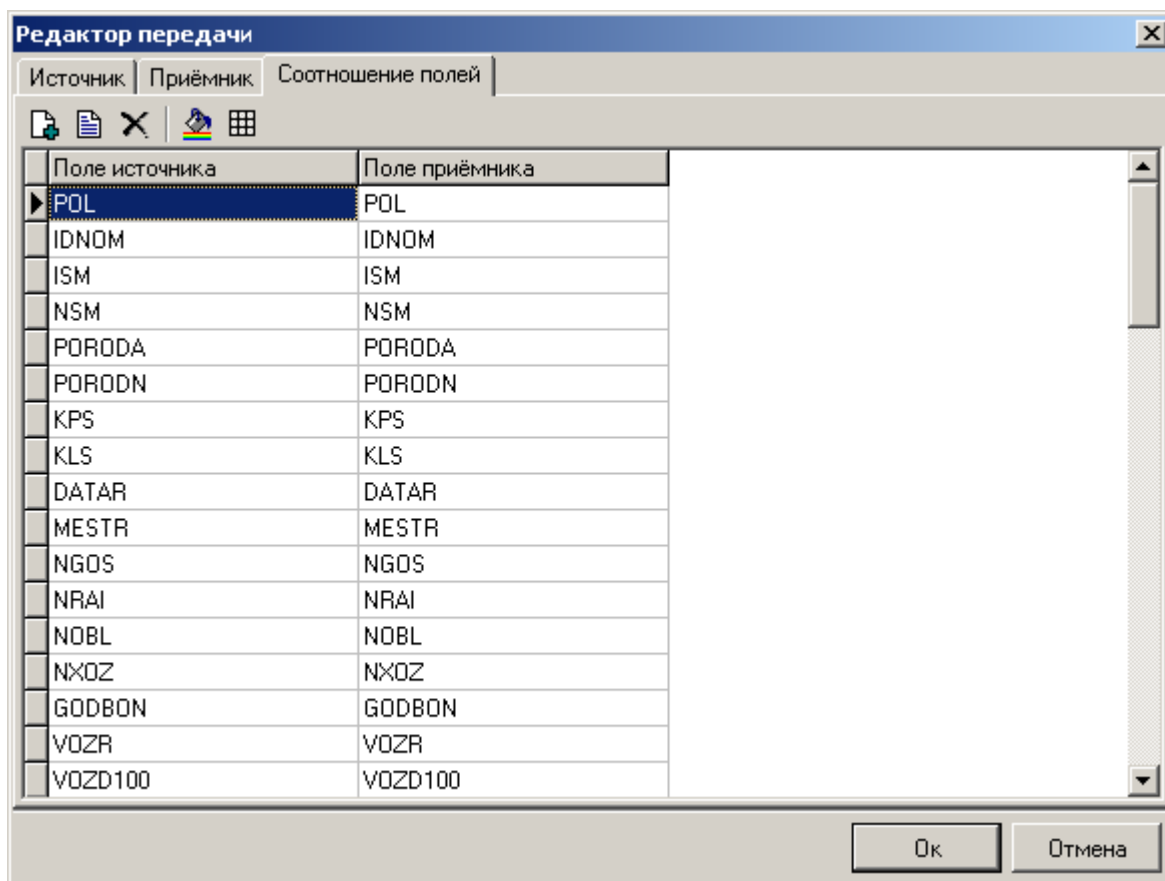
Кроме того, существует возможность выполнять SQL запрос на каждую из записей в источнике (только для технологий BDE и ADO). Настроить этот запрос можно нажав кнопку SQL запросы «На каждую». Это позволяет выполнить достаточно сложные манипуляции с базой данной – приёмником, например, вставку записей сразу в несколько таблиц через выполнение хранимой процедуры. Чаще при использовании этой возможности необходимо отключить добавление записей в таблицу приёмник. В этом случае необходимо снять галочку «Добавлять записи».

Если данные передаются в таблицу приёмник (галочка «Добавлять записи» установлена), то необходимо настроить соотношение полей в источнике и таблице приёмнике.








Для выполнения нескольких запросов их необходимо разделять знаком «;» в редакторе SQL запросов.





**Рис 5.4.** Настройка соотношения полей в источнике и таблице-приёмнике.

Картинка	Функция	Описание
	Добавить	Добавление пары полей источник-приёмник
	Редактировать	Редактирование пары полей источник-приёмник
	Удалить	Удаление пары полей источник-приёмник
	Заполнить	Заполнение пары полей источник-приёмник. Ищет одинаковые имена полей в источнике и таблице приёмнике и если таковые находятся, добавляет их.
	Очистить	Очистка таблицы соотношения полей



Если в таблице соотношения полей не указано какое-либо поле таблицы приёмника, то при передаче оно будет инициализировано значением NULL.

## 6 Координатор.

Данный компонент предназначен для связи источников между собой для проведения фильтрации (ведущей-детализированной), а также с переменными окружения. На один и тот же источник может быть наложено несколько связей, таким образом можно создавать достаточно сложные конструкции типа (источник1 -> источник2 -> (источник3, источник4)).

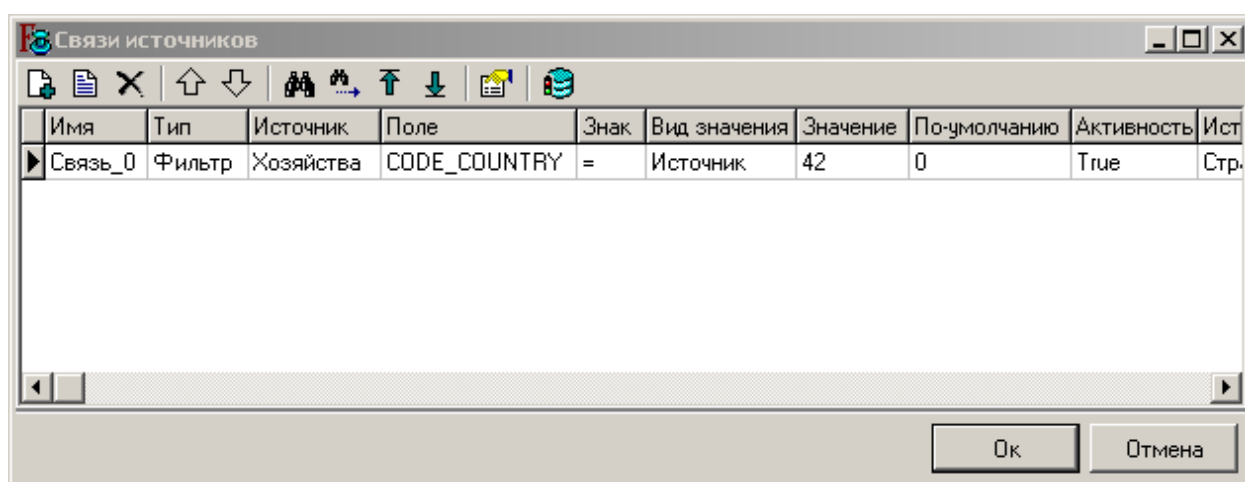




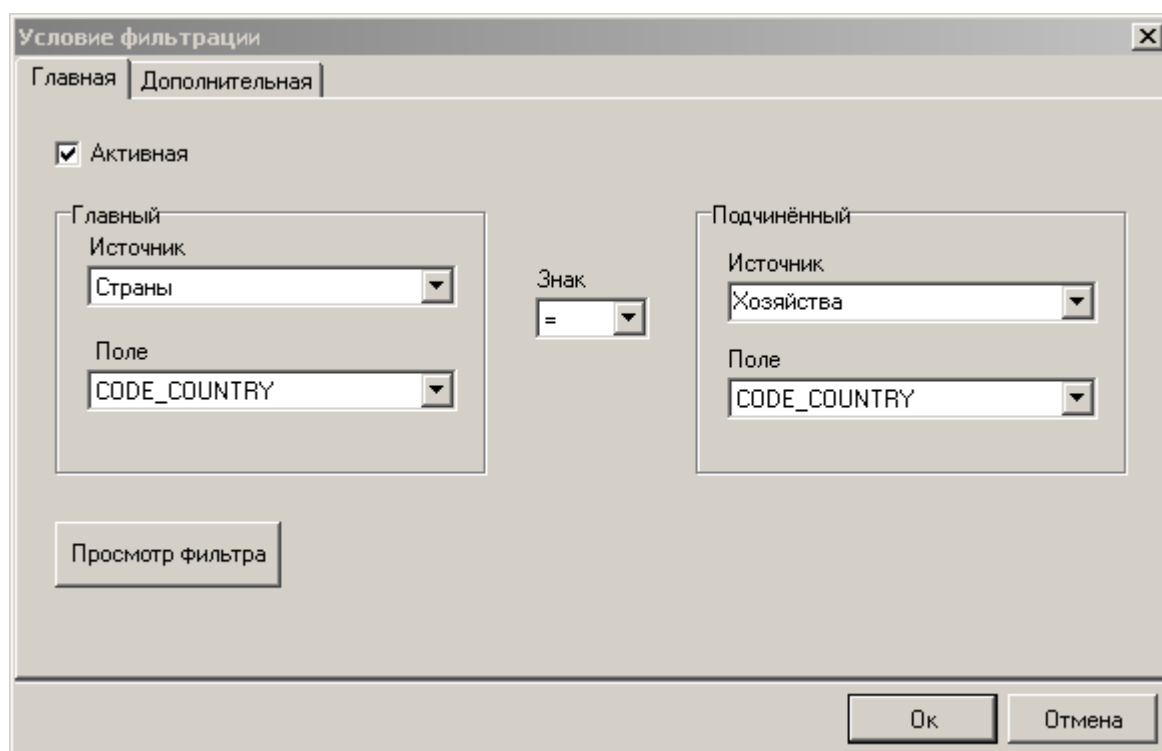
Рис 6.1. Редактор связей источников.

Для каждой из связей можно включить или выключить режим ускоренной фильтрации. В режиме ускоренной фильтрации фильтрация источника происходит только при активизации визуального компонента, к которому он прикреплен.

Картинка	Функция	Описание
	Добавить	Добавление новой связи
	Редактировать	Редактирование связи
	Удалить	Удаление связи
	Переместить вверх	Перемещение поля на одну позицию вверх
	Переместить вниз	Перемещение поля на одну позицию вниз
	Поиск	Осуществляет поиск в таблице связей по одному из атрибутов
	Продолжить поиск	Продолжает ранее начатый поиск

	Отладка	Осуществляет проверку правильности настройки связи
	Режим ускоренной фильтрации	Включает/отключает режим ускоренной фильтрации

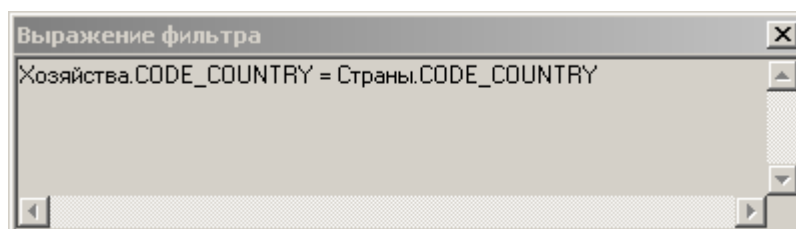
**Таблица 6.1** Настройка связей. Описание функций



**Рис 6.2.** Настройка условия фильтрации.

В качестве главного источника должен быть выбран источник, который осуществляет фильтрацию (ведущей) либо переменная окружения. Для выбора переменной окружения в список источников добавлен псевдо источник «Переменные». В качестве поля можно выбрать одно из доступных полей источника, если же выбран псевдо источник переменные, то одну из переменных. Далее следует указать знак (=, <>, >, <, >=, <=), и выбрать источник, в котором будет осуществлена фильтрация. После настройки фильтра можно просмотреть полученное выражение фильтра нажав на кнопку «Просмотр фильтра».

Каждая из связей может быть выключенной. В этом случае она не будет участвовать в фильтрации подчинённого источника. Для включения/выключения связи необходимо установить/снять галочку «Активная».



**Рис 6.3.** Просмотр выражения фильтрации.

**Рис 6.2.** Настройка дополнительных параметров фильтрации.

Часто требуется не использовать стандартный механизм фильтрации, а передать некоторую макроподстановку в текст SQL запроса источника. Для этого необходимо включить механизм макроподстановок и ввести имя подстановки, которое будет использована в запросе. При этом в качестве значения макроподстановки будет использовано значение поля из главного источника (переменной).

В редких случаях может потребоваться передать в подчинённый источник непосредственно текст фильтра. В этом случае необходимо включить механизм прямой фильтрации (галочка «Напрямую») и ввести текст фильтра в поле значение фильтра. (Пример: BREED.CODE\_DEPARTURE=1) Если фильтр напрямую используется совместно с механизмом макроподстановки, то в качестве макроподстановки будет использовано выражение фильтра.

Часто требуется заменить значение NULL получаемое из главного источника на другое значение. Это может потребоваться, поскольку для фильтров вроде BREED.CODE\_DEPARTURE=NULL могут быть получены не совсем корректные результаты либо операция вообще может завершиться ошибкой. В этом случае необходимо использовать значение по умолчанию. Значение по умолчанию имеет смысл использовать также совместно с макроподстановкой, например при отключении фильтра. В этом случае SQL запрос продолжает оставаться корректным, если поле значение по умолчанию введено корректно.



Для возможности включения/отключения фильтра в режиме макроподстановки обязательно должно быть указано значение по умолчанию.

## 7 Переменные приложения.

Переменные окружения по своей сути являются глобальными переменными всего приложения, в составе которого работает этот компонент. Отсюда и такое название. Они настраиваются централизованно и доступны во всех окнах и источниках. Иногда их требуется еще дополнительно объявить в компоненте внутренних связей (для правильной трансляции внутрь). Название переменных рекомендуем делать латинскими буквами и заключать с обеих сторон знаком "%". Тогда их будет хорошо видно в макроподстановках. Переменных можно добавлять достаточно много и работать они могут совершенно по-разному. Это зависит от того, какое действие их инициализирует.

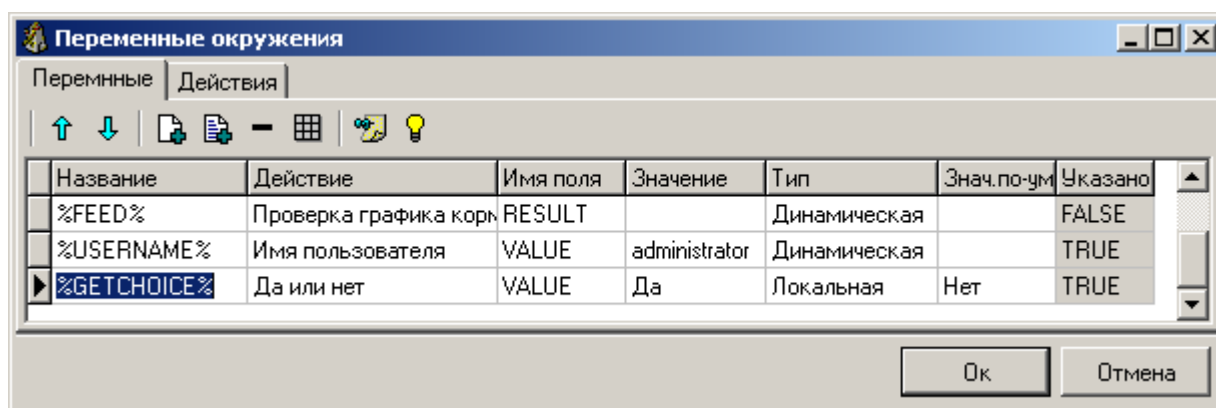


Рис 7.1. Настройка переменных окружения.



Одно действие может инициализировать сразу несколько переменных. Это особенно полезно, когда при выборе из справочника. При этом происходит сразу запись кода выбранной строки и ее текстовое поле (например, название предприятия и его код)

Для того чтобы правильно прописать переменную из потока данных, которое дает действие, требуется указывать имя поля. Если это одно поле, то обычно пишется VALUE. Оно же будет прописываться при сохранении настроек, если в эту колонку ничего не прописывать явно. Значение записывается, для того чтобы, посмотреть текущее значение, а так же протестировать работу механизмов, не выходя из редактора компонента. Тип переменной может быть следующий:

- ✓ Локальная
- ✓ Серверная
- ✓ Динамическая
- ✓ Рестартовая

Если переменная объявляется как **локальная**, то ее значение сохраняется на рабочей станции пользователя и от компьютера к компьютеру может одновременно быть разным. Например, в локальную переменную может быть записан код предприятия, для которого выводятся данные.

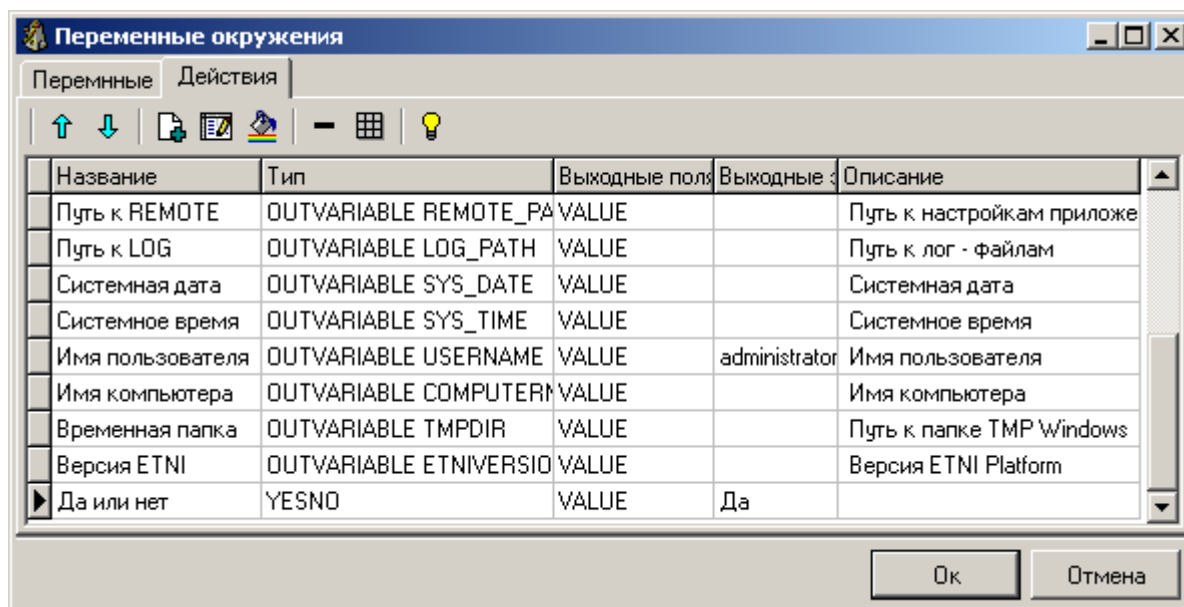
Если это **серверная**, то ее изменение коснется всех пользователей, работающих в системе. Например, это может быть путь к мультимедиа информации, расположенной в одном месте для всей организации. **Динамическая** переменная своих значений не сохраняет и каждый раз при обращении запрашивает новые. Это характерно для таких переменных как текущая дата, текущее время, имя пользователя. **Рестартовая** переменная перезагружает инициализацию приложения. Например, это требуется при смене пользователя без выхода из программы или, скажем, смены пути к базе данных.

Значение по умолчанию дает возможность начальной инициализации (для первого запуска). Последняя колонка дает информацию о том, была ли переменная обработана, и если этого не происходило, то, как раз значение по умолчанию и берется.

Картинка	Функция	Описание
	Поднять переменную	Перемещение переменной вверх относительно остальных.
	Опустить переменную	Перемещение переменной вниз относительно остальных.
	Добавить переменную	Добавляется новая строка для ввода
	Удалить переменную	Удаление текущей строки с настройкой переменной
	Удалить все строки	Удаление всех переменных в списке
	Запустить переменную	Запустить действие, связанное с выбранной переменной
	Прочитать переменную	Считать текущее значение переменной. Действие при этом не запускается *

**Таблица 7.1** Функции работы с переменными.

Примечание: \* Это не всегда так. Для динамических переменных действие запускается в любом случае, даже если это только чтение переменной.



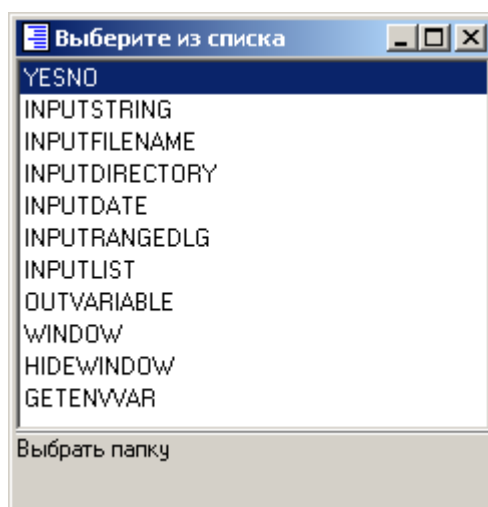
**Рис 7.2.** Настройка действий для переменных окружения.

То, что касается действий, то их можно условно разделить на три вида

- ✓ Предустановленные в компоненте
- ✓ Предусмотренные приложением
- ✓ Добавляемые пользователем

К предустановленным относятся следующие действия: “Системная дата”, “Системное время”, “Имя пользователя”, “Имя компьютера”, “Временная папка”. Этот список постоянно расширяется. В списке могут быть такие как добавленные приложением, например: “Путь к REMOTE”, “Путь к LOG”, “Версия ETNI”. Их полный список можно увидеть, выполнив операцию “Заполнить внешние действия”.

Название действия можно не пояснять, вот с типами следует разобраться более подробно. На рисунке 7.3 представлен их список (типы так же понемногу наращиваются разработчиками).



**Рис 7.3.** Варианты типов действий.

Значение	Описание
YESNO	Задать простой вопрос Да/Нет
INPUTSTRING	Ввод строки
INPUTFILENAME	Выбрать файл
INPUTDIRECTORY	Выбрать папку
INPUTDATE	Указать дату
INPUTRANGEDLG	Весит диапазон дат
INPUTLIST	Выбрать из списка
OUTVARIABLE	Внешняя переменная
WINDOW	Вызвать окно
HIDEWINDOW	Скрытое окно. HIDEWINDOW XXX, где XXX - имя окна
GETENVVAR	Значение переменной окружения Windows. GETENVVAR XXX, где XXX – имя переменной

**Таблица 7.3** Перечень типов действий.

Действия можно настраивать и сразу же тестировать, не выходя из редактора настройки, используя функцию “Запустить действие”. Когда действие выполняется, прописываются значения выходных полей. Поля пишутся через “;”, а их значения через вертикальную черту “|”.

В качестве примера сложного выходного значения можно взять справочник предприятий. После того как произошел выбор предприятия в окне по полю CODE\_COMPANY можно получить код, а по полю NAME – названия предприятия. Тогда в колонке выходные поля следует указать CODE\_COMPANY;NAME и при этом будет ожидаться результат, например, “1010| Рассвет”. Соответственно и при настройках переменных эти поля следует прописывать.



Можно организовать несколько действий с одним и тем же типом. Например, ввести имя, отчество и фамилию тремя разными действиями, хотя это будет аналогичная операция по получению текстовой строки от пользователя.

Картинка	Функция	Описание
	Поднять вверх	Поднимает строку действия вверх относительно остальных
	Опустить вниз	Опускает относительно остальных вниз
	Добавить действие	Добавляет еще одну строку для дальнейшего заполнения
	Список значений	Дает возможность заполнить список значений для последующего выбора
	Заполнить внешние действия	Заполняет перечень всеми возможными действиями
	Удалить действие	Удаляет текущее действие
	Удалить все действия	Удаляет все действия
	Запустить действие	Тестирование действия с прописыванием результатов.

**Таблица 7.2** Функции работы с переменными.

Для работы с переменными следует воспользоваться практическими опытами и проследить, какие опции доступны, что происходит при той или иной операции. В режиме настройки рассматриваемого компонента можно их отдельно отладить и уже только после этого вставлять в редактор внутренних связей, SQL-запросы источников и так далее.



## 8 Pascal скрипты.

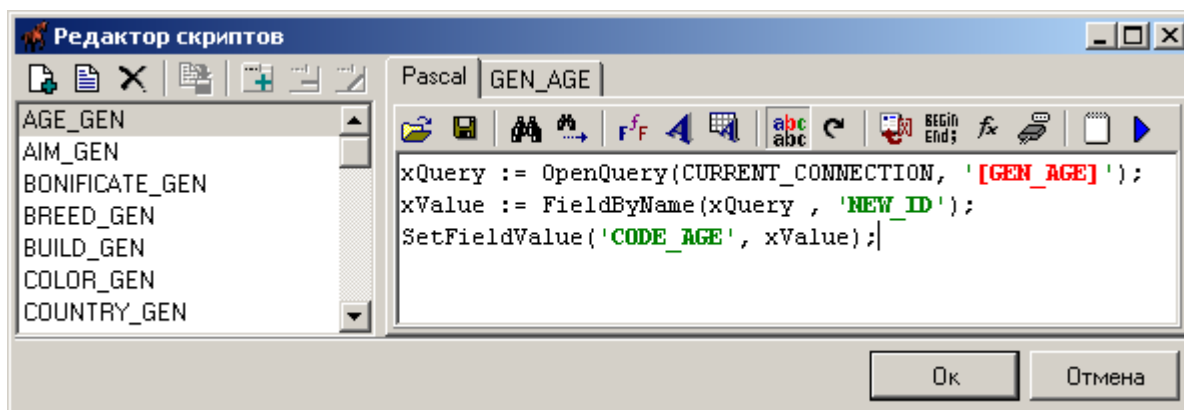


Рис 8.1. Редактор скриптов на языке Pascal.

В левой части окна (рисунок 8.1) расположен список всех скриптов, используемых в приложении. В правой части вводится текст самого скрипта. При перемещении по списку скриптов меняется изображение в правой части. Далее, в таблице 8.1, описываем основные функции манипуляции со скриптами.








Картинка	Функция	Описание
	Добавить скрипт	Добавляется новый скрипт после ввода его названия в появившемся окне. Требуется написание латинскими буквами.
	Переименовать	Меняется название скрипта в появившемся окне вода
	Удалить скрипт	Выбранный скрипт удаляется из списка со всем исходными кодом.
	Сохранить скрипт	Введенный в правой части текст будет применен к текущему скрипту. Если он не менялся – кнопка неактивна. Так же переспрашивает при переходе на другой скрипт.
	Добавить закладку	Делает еще одну закладку помимо основной, обозначенной как паскаль. На появившейся закладке вводится текст на другом языке. Обычно это SQL.
	Удалить закладку	Удаляется выбранная закладка для скрипта. Первую закладку удалить нельзя.
	Переименовать закладку	Изменение названия закладки

Таблица 8.1 Функции работы со скриптами.

Примечательно, что во второй закладке (да и в любой последующей) может содержаться инструкция не только на языке SQL, но и любом другом – главное чтобы для ее интерпретации был встроенный механизм (встроенная функция на Pascal). То есть фрагменты с другой страницы просто подставляются внутрь Pascal-кода. Для формирования кода на языках Pascal и SQL предусмотрены заранее заготовленные шаблоны для облегчения жизни разработчика. Эти наборы постепенно расширяются. Их использование показано на рисунке 8.2

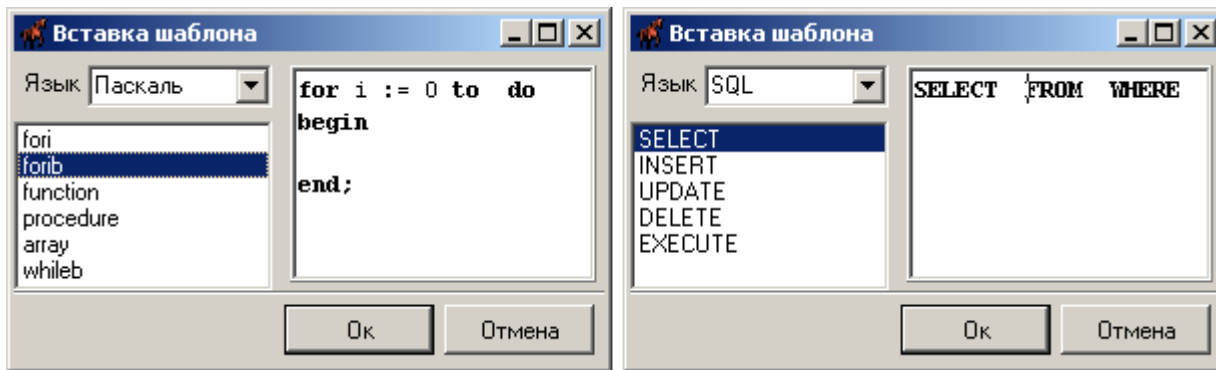


Рис 8.2 Использование шаблонов кода.

Собственно все функции, имеющиеся в наборе в правой части окна – это помощь для создания эффективного кода с минимальными затратами времени. Не предполагается, что здесь будет разрабатываться что-то грандиозное, но несколько строк кода для облегчения жизни набросать всегда можно, как в нашем случае для использования генераторов кода.

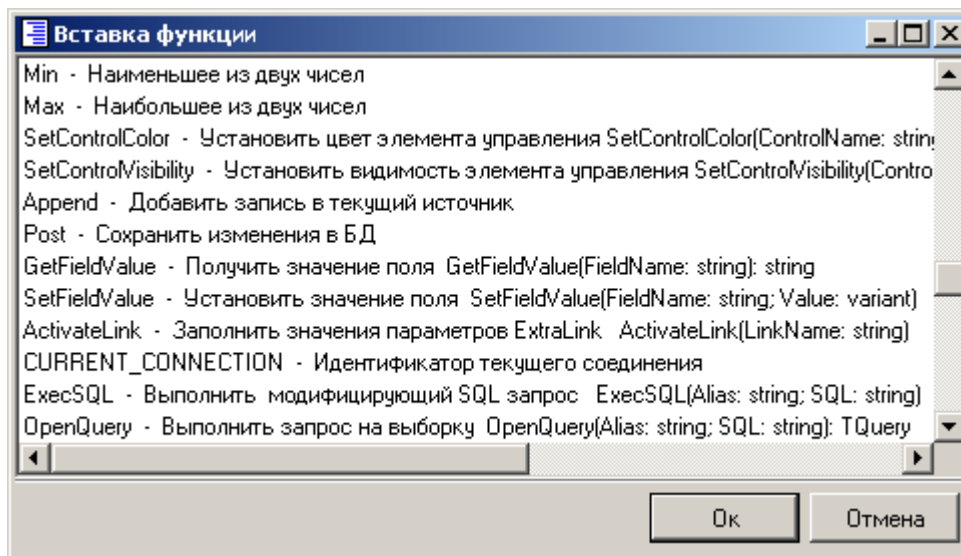


Рис 8.3. Перечень встроенных функций.

Встроенные функции позволяют быстро проворачивать неочевидные вещи. И даже имя такой вот список функций лучше обращаться к работающим примерам, поставляемых с программным обеспечением. Следует различать встроенные функции от переменных приложения. Переменные для макроподстановки берутся в квадратные скобки и обычно выделяются другим цветом (обычно красным). Конструкции языка выделяются жирным черным шрифтом.

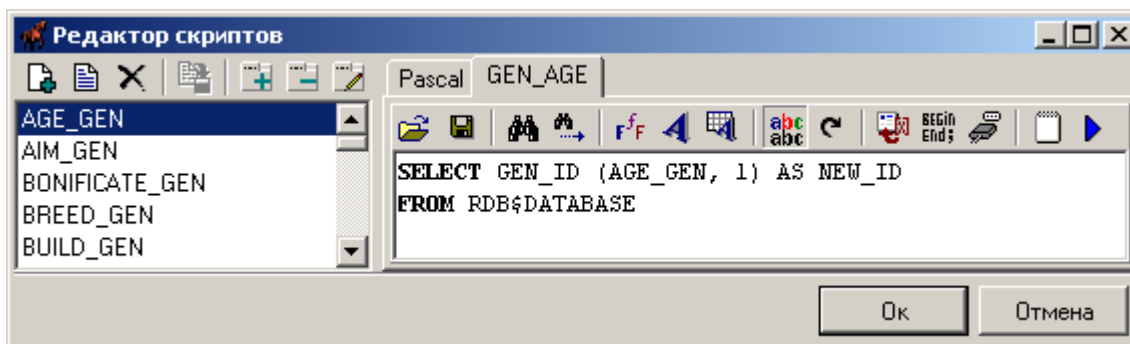


Рис 8.3. Редактор скриптов на языке SQL.

Картинка	Функция	Описание
	Загрузить из файла	Служит для загрузки исходного кода из обычных текстовых файлов. Это - импорт текста.
	Сохранить в файл	Позволяет сохранить написанный код в текстовый файл. По сути – экспорт текста.
	Найти	Выполняется поиск внутри текста скрипта. Найденный фрагмент выделяется
	Продолжить поиск	Продолжение ранее начатого поиска
	Шрифт текста	Изменение шрифта текста. Выводиться стандартный диалог настройки шрифта.
	Увеличить шрифт	Увеличение шрифта текста на 2 пункта
	Уменьшить шрифт	Уменьшение шрифта на 2 пункта
	Раскрашивать	Включает и выключает режим раскрашивания служебных слов внутри скрипта.
	Раскрасить	Раскрасить введенный текст заново.
	Вставить переменную	Вставить из списка переменных нужное место внутри текста с текущей позиции курсора.
	Вставить шаблон	Вставка готовой конструкции на языке Pascal или SQL
	Вставить функцию	Вставка встроенной функции из списка
	Вставить параметр связи	Вставка параметра связи
	Предпросмотр	Выполнить просмотр скрипта со всеми макроподстановками
	Выполнить	Выполнить скрипт

**Таблица 8.1** Функции работы со скриптами.

Скрипты в обязательном порядке подключаются к источнику. Делается это на четвертой странице (рисунок 2.4.1). Причем события могут быть совершенно разнообразными: добавление, редактирование, изменение. Последнее удобно использовать для расчета вычисляемых полей.

С помощью вставки параметра связи есть возможность вызова другого окна, организации дополнительного диалога с пользователем. Подробнее об этой технологии можно прочитать в главе “Внутренние связи”.

## 9 SQL скрипты.

Данный компонент предназначен для выполнения модифицирующих SQL запросов (INSERT, UPDATE, DELETE, MERGE, EXECUTE PROCEDURE, EXECUTE BLOCK...) и их комбинаций. Его настройка во многом схожа с настройкой источника данных, но в отличие от источника данных он не умеет возвращать наборы данных.

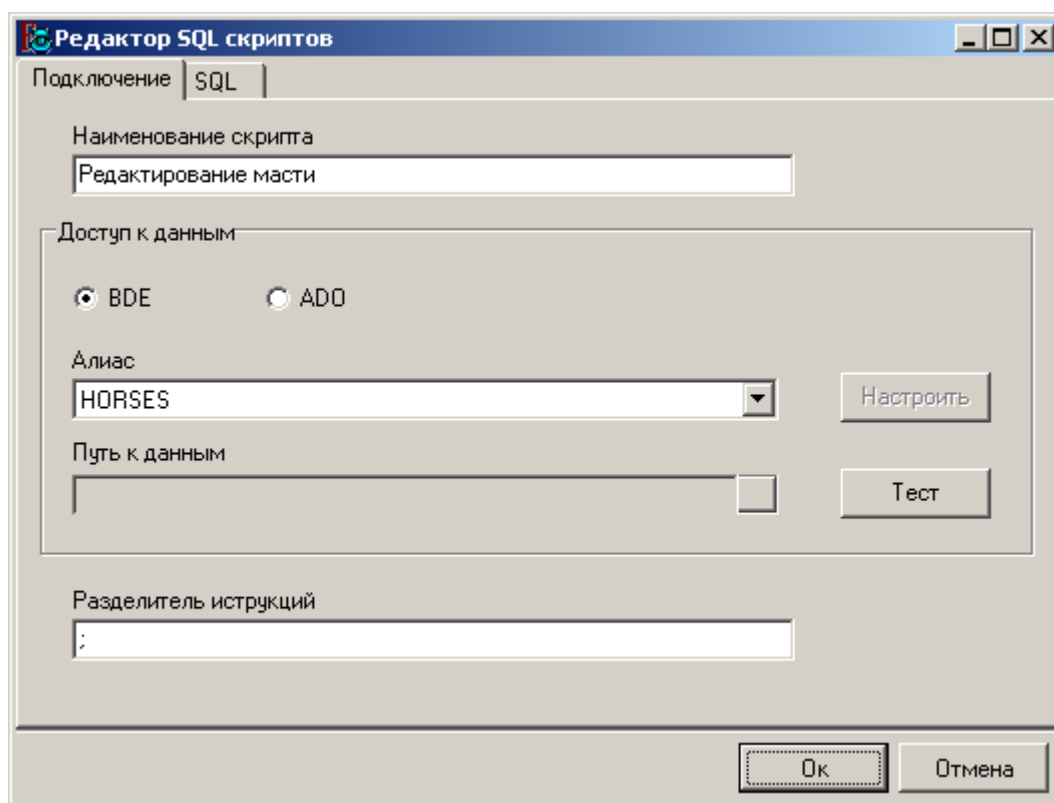


Рис 9.1. Настройка подключения для SQL скриптов.

В качестве доступа к данным может быть использована одна из технологий BDE или ADO.

Технологию BDE описывать не имеет смысла – она используется во многих системах. Достаточно только выбрать имя алиаса.

Технология ADO применяется в двух вариантах. В первом случае выбирается алиас ADO по имени, во втором можно организовать доступ прямой настройкой строки соединения стандартным редактором.

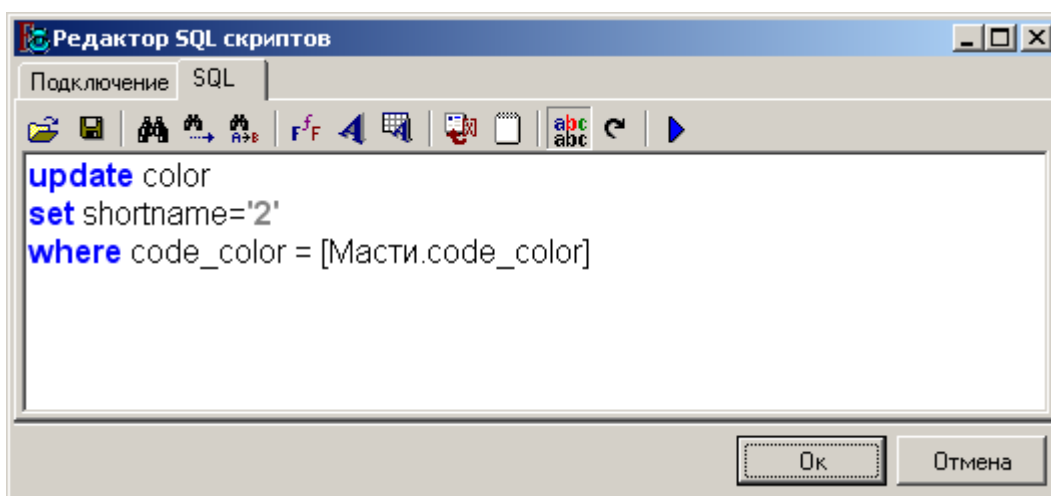
Для использования предустановленных строк ADO можно дополнительно задавать путь к каталогу, При этом в списке алиасов выбираются следующие варианты

- ✓ DBASE
- ✓ ACCESS
- ✓ EXCEL
- ✓ CUSTOM

Для первых трех в списке достаточно указать путь – система сама правильно подберет драйвер, для всех остальных случаев доступа через ADO следует выбирать CUSTOM. После того как соединение настроено, его можно протестировать командой "Тест".

Для некоторых систем требуется указывать имя скрипта. Оно указывается в графе «наименование скрипта» и фигурирует в дальнейших настройках. Рекомендуется давать скрипту осмысленные имена, что, несомненно, облегчает дальнейшее развитие и совершенствование системы.




Поскольку компонент может выполнить не только одну команду, но и зачастую их комбинацию, то для их разделения необходимо указать разделитель инструкций (по умолчанию «;»).



**Рис 9.2.** Редактор SQL скрипта.

На закладке SQL вводится сам скрипт или скрипты, разделённые указанным разделителем инструкций.

Картинка	Функция	Описание
	Загрузить из файла	Служит для загрузки исходного кода из обычных текстовых файлов. Это - импорт текста.
	Сохранить в файл	Позволяет сохранить написанный код в текстовый файл. По сути – экспорт текста.
	Найти	Выполняется поиск внутри текста скрипта. Найденный фрагмент выделяется
	Продолжить поиск	Продолжение ранее начатого поиска
	Поиск и замена	Осуществляет поиск подстроки внутри текста и заменяет его на другую строку
	Шрифт текста	Изменение шрифта текста. Выводиться стандартный диалог настройки шрифта.
	Увеличить шрифт	Увеличение шрифта текста на 2 пункта

	Уменьшить шрифт	Уменьшение шрифта на 2 пункта
	Вставить переменную	Вставить из списка переменных нужное место внутри текста с текущей позиции курсора.
	Предпросмотр	Выполнить просмотр скрипта со всеми макроподстановками
	Раскрашивать	Включает и выключает режим раскрашивания служебных слов внутри скрипта.
	Раскрасить	Раскрасить введенный текст заново.
	Выполнить	Выполнить скрипт

**Таблица 9.1** Функции работы с SQL скриптами.

Компонент SQL-скрипты даёт обширные возможности по взаимодействию с SQL сервером, поэтому он постоянно развивается.



При отладке SQL скриптов рекомендуется делать копию базы данных, поскольку выполнение сценария может привести к необратимым результатам.

