

Применение обработчиков событий при создании правил обмена данными с помощью "Конвертация данных 2.0"

В этой статье мы продолжим описание методик работы с конфигурацией "Конвертация данных 2.0". Перед чтением материала, настоятельно рекомендуем внимательно изучить статью "Организация конвертации и обмена данными с помощью "Конвертация данных 2.0", размещенную на диске ИТС. Там Вы найдете описание основных технологических принципов обмена данными с помощью конфигурации "Конвертация данных 2.0" и изложение базовых приемов построения правил обмена.

Для работы Вам понадобятся конфигурация "Конвертация данных 2.0" версия 2.10.1, внешние обработки MD80Exp.epf, V8Exchan.epf, MD77Exp.ert, V77Exp.ert, поставляемые с релизом конфигурации "Конвертация данных 2.0". Для реализации механизма обмена данными необходимы конфигурации "Бухгалтерский учет 4.5" (на платформе 1С:Предприятие 7.7) и "Управление торговлей 10.2" (на платформе 1С:Предприятие 8).

Обработчики событий: основные принципы

Механизм обработчиков событий является один из ключевых в технологии конвертации данных с помощью "Конвертации данных 2.0". Грамотное и умелое использование этого механизма позволяет разработчику быстро решать практически любые задачи по преобразованию данных. С помощью технологии обработчиков легко реализуются отбор данных, конвертации данных разных типов, сложные выборки данных, настройка параметров конвертации и многие другие задачи.

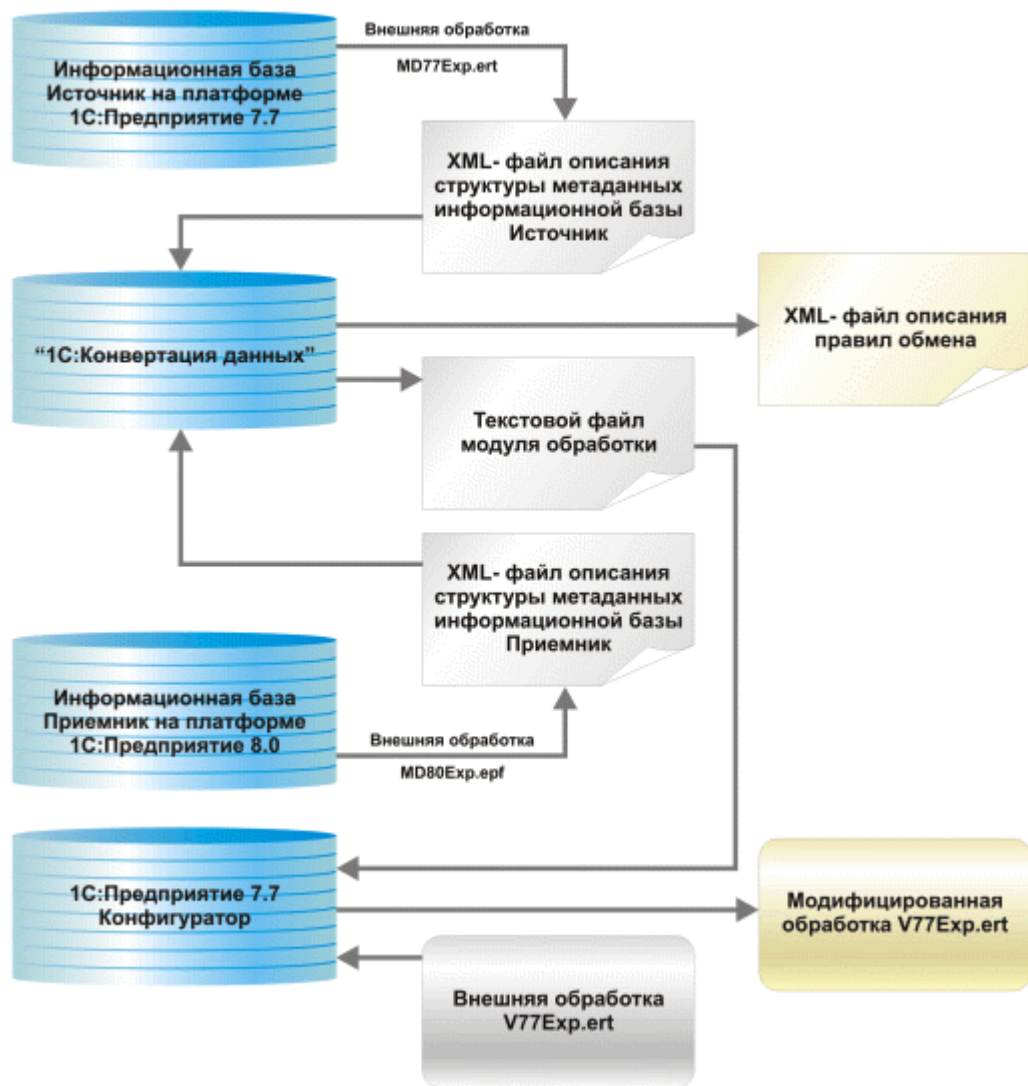
Рассмотрим основные принципы этой технологии. В ключевых точках алгоритмов выгрузки и загрузки данных обработок универсального обмена есть возможность исполнения программного кода взятого из правил обмена данными, а не "зашитого" в обработке выгрузки или загрузки данных. Конфигурация "Конвертация данных 2.0" предоставляет возможности для интеграции такого программного кода в правила обмена данными.

Всего различных мест в алгоритмах обмена данными, где может быть выполнен сторонний код, предусмотрено двадцать пять. Соответственно, в конфигурации предусмотрено создание двадцать пять различных типов обработчиков событий.

Код обработчиков событий "привязан" к объектам правил обмена - элементам справочников: конвертации, правила конвертации объектов, правила конвертации свойств, правила выгрузки данных и правила очистки данных. Естественно, что код обработчиков событий должен удовлетворят ряду требования. В частности для управления процессом конвертации в коде обработчиков необходимо использовать специальные переменные - параметры. Полное описание всех типов обработчиков событий и доступных переменных можно найти в справочной информации. В статье мы рассмотрим ряд часто используемых обработчиков событий, разберем типичные ситуации программирования с применением переменных-параметров.

Технологии "Конвертации данных 2.0" позволяют вести обмена данными с информационными базами, реализованными на платформах "1С:Предприятие 7.7" и "1С:Предприятие 8". Общая схема подготовки правил обмена данными была рассмотрена в предыдущей статье. В силу особенностей работы платформы "1С:Предприятие 7.7" подготовка правил обмена данных с использованием обработчиков событий для информационных баз, реализованных на этой платформе имеет ряд особенностей. Процесс подготовки правил для конвертации данных из конфигурации на платформе "1С:Предприятие 7.7" в конфигурацию на платформе "1С:Предприятие 8" наглядно представлен на **схеме 1**.

Схема 1. Технология подготовки правил обмена данными, для конфигураций на платформе 1С:Предприятие 7.7



Как видно на схеме, для получения описания метаданных конфигурации на платформах "1С:Предприятие 7.7" используется специализированная обработка MD77Exp.ert. После этого описание метаданных помещается в информационную базу "Конвертации данных 2.0", проводится настройка правил обмена.

При выгрузке, кроме самого файла правил, система генерирует текст модуля для обработки V77Exp.ert с функциями, реализующими обработчики событий. Затем, в конфигураторе, мы должны заменить модуль стандартной V77Exp.ert на новый, сгенерированный "Конвертацией данных 2.0". На методике модификации обработки мы подробнее остановимся ниже, в одном из заключительных разделов.

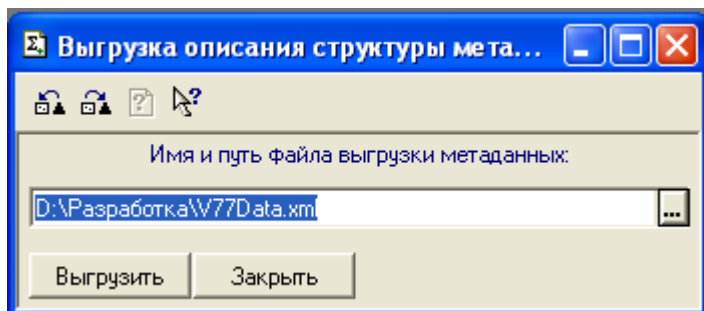
При разработке решений по обмену данными на платформе "1С:Предприятие 7.7" нужно помнить про эту важную "мелочь". Ваши правила будут корректно работать только в том случае, если Вы используете модифицированную обработку, текст модуля которой создан при выгрузке правил обмена данными. У этого правила есть одно исключение - если Вы не пользуетесь обработчиками событий, то можно применять стандартную обработку.

После краткого теоретического экскурса, исследуем возможности технологии обработчиков на живом материале примера. Рассмотрим типичную задачу конвертации данных между конфигурациями на разных платформах. Допустим, предприятие ведет бухгалтерский учет на типовой конфигурации "Бухгалтерский учет 4.5". Основная деятельность - оптовая

торговля. В связи с расширением бизнеса принято решение автоматизировать складской и управленческий учет на базе решения "Управление торговлей 10.2". Первоочередная задача - перенос информации о контрагентах.

Анализ структуры данных

Как всегда в случае подготовки правил обмена, начнем нашу работу с формирования описания метаданных. Получить описание метаданных для конфигурации "Управление торговлей 10.2" не составит труда. Мы уже научились это делать при проработке методических материалов по конфигурации "Конвертации данных 2.0". Для конфигурации "Бухгалтерский учет 4.5" это сделать даже проще. Запустим внешнюю обработку MD77Exp.ert, укажем путь файла описания метаданных и нажмем "Выполнить".



Затем загрузим описания метаданных в информационную базу и начнем анализ структур конфигураций, участников конвертации. Конфигурации не идентичны по структуре и поэтому, прежде чем строить правила обмена, нужно выявить все проблемы, которые могут возникнуть при конвертации данных и наметить пути их решения. Нам необходимо заранее иметь проект правил обмена, в котором были бы учтены особенности конкретной задачи конвертации.

В целом, процесс проектирования структуры правил обмена может быть разбит на следующие этапы:

- Выделение объектов-источников в структуре конфигурации источника
- Определение объектов-приемников данных в структуре конфигурации приемника
- Установка соответствий между объектами источника и приемника
- Анализ несоответствия свойств по каждой паре источник и приемник
- Выявление реквизитов приемника, заполняемых константными значениями
- Анализ требований к отбору данных при выгрузке
- Анализ требований к производительности процедур конвертации

В нашем случае необходимо произвести перенос информации о контрагентах. Эта информация в исходной информационной базе хранится в целой системе справочников - Контрагенты, Расчетные счета, Договора. Ограничим нашу задачу созданием каркаса выгрузки. Для этого необходимо перенести наименование контрагента, групповую структуру, ИНН и юридический адрес. Таким образом, в качестве источника у нас выступает только справочник Контрагенты.

Перейдем к анализу структуры приемника. Мы видим, что конфигурация "Управление торговлей" имеет иную структуру хранения информации о контрагентах. Основное отличие, для наших целей, заключено в способе хранения контактной информации. В "Управлении торговлей" создан специализированный регистр сведений для хранения контактной информации контрагентов, в частности юридического адреса. Поэтому в качестве приемника у нас два объекта - справочник Контрагенты и регистр сведений КонтактнаяИнформация. Очевидно, что вся основная информация из справочника Контрагенты бухгалтерской информационной базы будет перенесена в справочник Контрагенты торговой, а в регистр сведений должна быть записана информация реквизита ЮридическийАдрес.

Остановимся на реквизитах справочников. Очень важное обстоятельство - длина кода у справочника Контрагенты в "Управлении торговлей" меньше чем в конфигурации источнике.

Перейдем к тем, реквизитам, которые должны быть обязательно заполнены в приемнике, но не имею очевидных аналогов в источнике. Это в первую очередь относится к регистру сведений. Для заполнения юридического адреса контрагента необходимо, присвоить измерению Тип (Перечисление Типы контактной информации) значение Адрес, а измерению Вид predetermined элемент "Юридический адрес контрагента" справочника Виды контактной информации.

Что касается отбора данных, этот вопрос, конечно, не может быть решен с помощью анализа структур конфигураций, а определяется при поставке задачи. Допустим, в нашем случае нужно переносить только данных о поставщиках и покупателях, которые являются юридическими лицами. Нужные элементы справочников расположены в группах "Поставщики" и "Покупатели". Такая структура хранения данных реализована в демонстрационной базе конфигурации "Бухгалтерский учет 4.5". И, наконец, отметим, что в силу небольшого размера базы особых требований к производительности процессам выгрузки и загрузки данных нет.

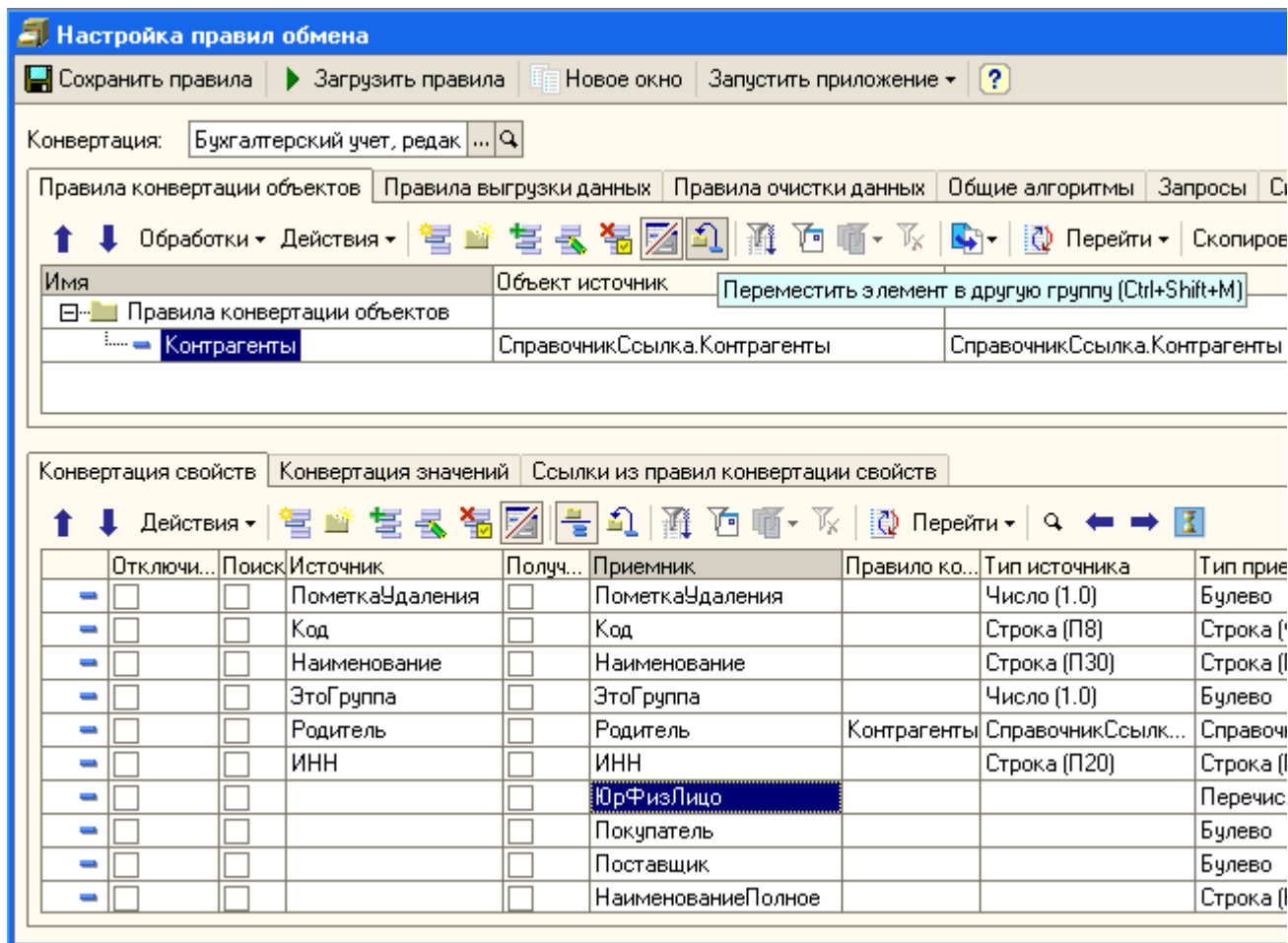
Анализ структур данных позволяет сформулировать ряд принципиальных проектных соображений, которые будут применяться при построении данных:

- Необходимо создать два правила конвертации - для справочника Контрагенты и для регистра сведений КонтактнаяИнформация
- Для выгрузки групповой структуры необходимо реализовать механизм конвертации кодов справочника Контрагенты
- Необходимо реализовать механизм отбора данных по указанным условиям
- Необходимо реализовать механизм заполнения обязательных полей Тип и Вид для записи регистра сведений

Итак, первый этап работы завершен, можно вплотную приступать к построению правил. Перед этим хотелось бы сделать несколько замечаний. Часто разработчики стремятся как можно быстрее перейти к реализационной работе, пропуская этап проектирования. При создании правил обмена мы настоятельно рекомендуем Вам добросовестно пройти по всем перечисленным пунктам до начала программирования. В особенности это касается ситуаций, когда в качестве источника данных выступает недокументированная оригинальная конфигурация. Тщательная проработка на этапе проектирования позволит Вам резко сократить время отладки правил.

Конвертация контрагентов - устанавливаем значения свойств в обработчиках

Начнем с создания правил конвертации справочника Контрагенты. Создадим новый элемент справочника Конвертации и откроем уже знакомую нам форму "Настройка правил конвертации". С помощью обработок автогенерации создадим правило конвертации объектов Контрагенты, и правила конвертации свойств со следующими приемниками: ПометкаУдаления, Код, Наименование, ЭтоГруппа, Родитель, ИНН, ЮрФизЛицо, Покупатель, Поставщик, НаименованиеПолное.



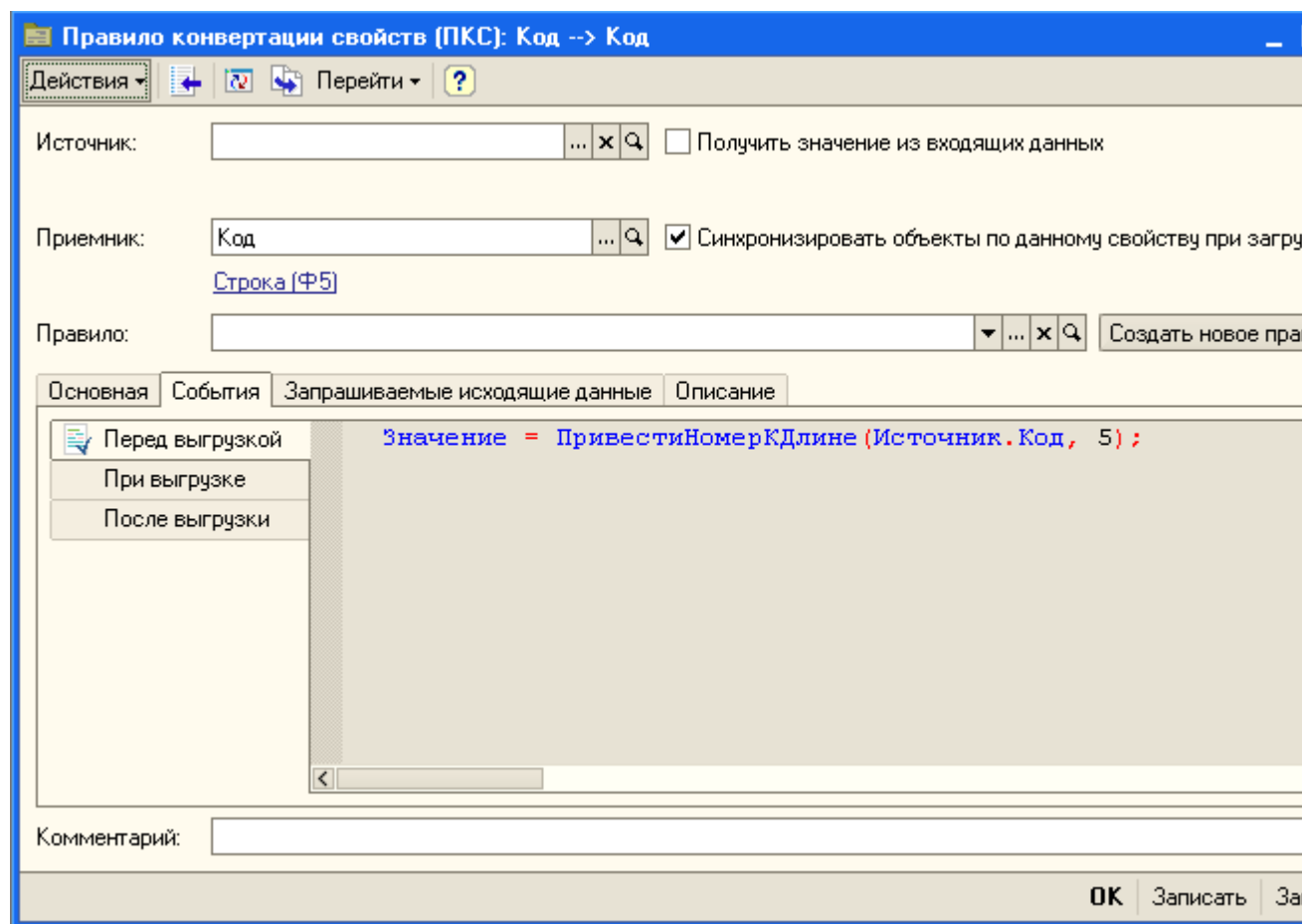
Это список всех реквизитов, которые мы планируем заполнить. Некоторые правила свойств имеют пустой источник. Нам придется заполнить значения этих свойств программно, с помощью обработчиков событий. Каждый обработчик события связан с определенным элементом правил обмена и выполняется в процессе обработки элемента правила. Для правил конвертации свойств определены три типа обработчиков - "Перед выгрузкой", "При выгрузке" и "После выгрузки". Обработчик "Перед выгрузкой" выполняется при начале конвертации значения по правилу конвертации свойств. После того как получено значение для конвертации и создан XML-узел свойства для записи в файл, выполняется обработчик "При выгрузке". И, наконец, после завершения обработки правила выполняется обработчик "После выгрузки".

В практике наиболее часто употребляется обработчик "Перед выгрузкой". Он, как правило, используется для программного определения конвертируемого значения. В коде обработчика доступны две важные для нас переменные - Значение и Источник. Первая переменная дает доступ к значению, переданному для конвертации, вторая к объекту источнику данных. Таким образом, программно можно легко установить произвольное значение свойства.

Рассмотрим этот прием подробнее. В первую очередь, обратимся к вопросу установки кода. Для случаев, когда коды справочников и номера документов необходимо конвертировать в новый формат, в обработках универсального обмена предусмотрена специальная функция ПривестиНомерКДлине. Она имеет два параметра. Первый - значение для конвертации, второй - число символов в строке-приемнике. Особо обратите внимание, что этой функцией можно пользоваться в тех случаях, когда возможно корректное преобразование с помощью изменения числа нулей в коде, перед числовой частью.

Откроем форму правила конвертации свойства Код, очистим реквизит Источник и поместим в поле "Перед выгрузкой" следующий программный код:

Значение = ПривестиНомерКДлине (Источник.Код, 5);

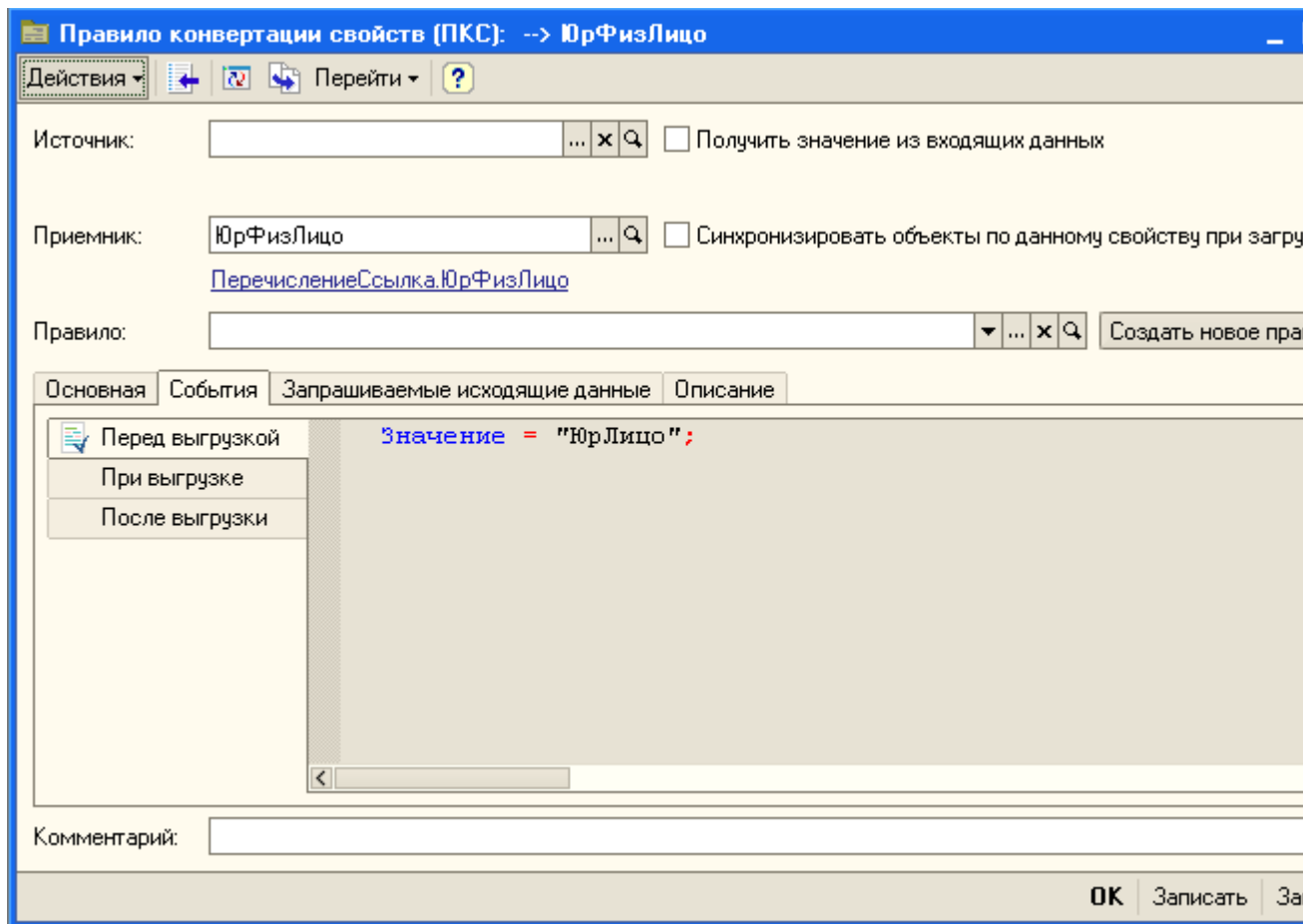


Обратите внимание - значение кода элемента справочника мы получаем с помощью переменной Источник, где хранится ссылка на выгружаемый элемент.

Следующий реквизит, требующий программной установки это - ЮрФизЛицо. Заметим, что приемник имеет тип Перечисление. Это свойство для всех выгружаемых элементов будет иметь значения "Юр. лицо". При выгрузке нам нужно указать имя значения перечисления как оно задано в конфигурации:

Значение = "ЮрЛицо";

Этот код также помещается в обработчике события "Перед выгрузкой".



Реквизиты Поставщик и Покупатель имеют тип Булево. Определим их в соответствии с группой, в которой расположен контрагент. Обработчик для реквизита Покупатель :

```
Если Источник.Родитель.Наименование = "Покупатели" Тогда
    Значение = 1;
Иначе
    Значение = 0;
КонецЕсли;
```

Обработчик для реквизита Поставщик:

```
Если Источник.Родитель.Наименование = "Поставщики" Тогда
    Значение = 1;
Иначе
    Значение = 0;
КонецЕсли;
```

Обратите внимание, что при написании кода использована особенность структуры данных конкретной информационной базы. Контрагенты расположены строго в двух группах, а справочник имеет всего два уровня вложений. В случае более сложной группой структуры необходимо применять другие методы отбора, например, по полному коду элемента справочника.

Программная установка реквизита НаименованиеПолное делается по тому же принципу:

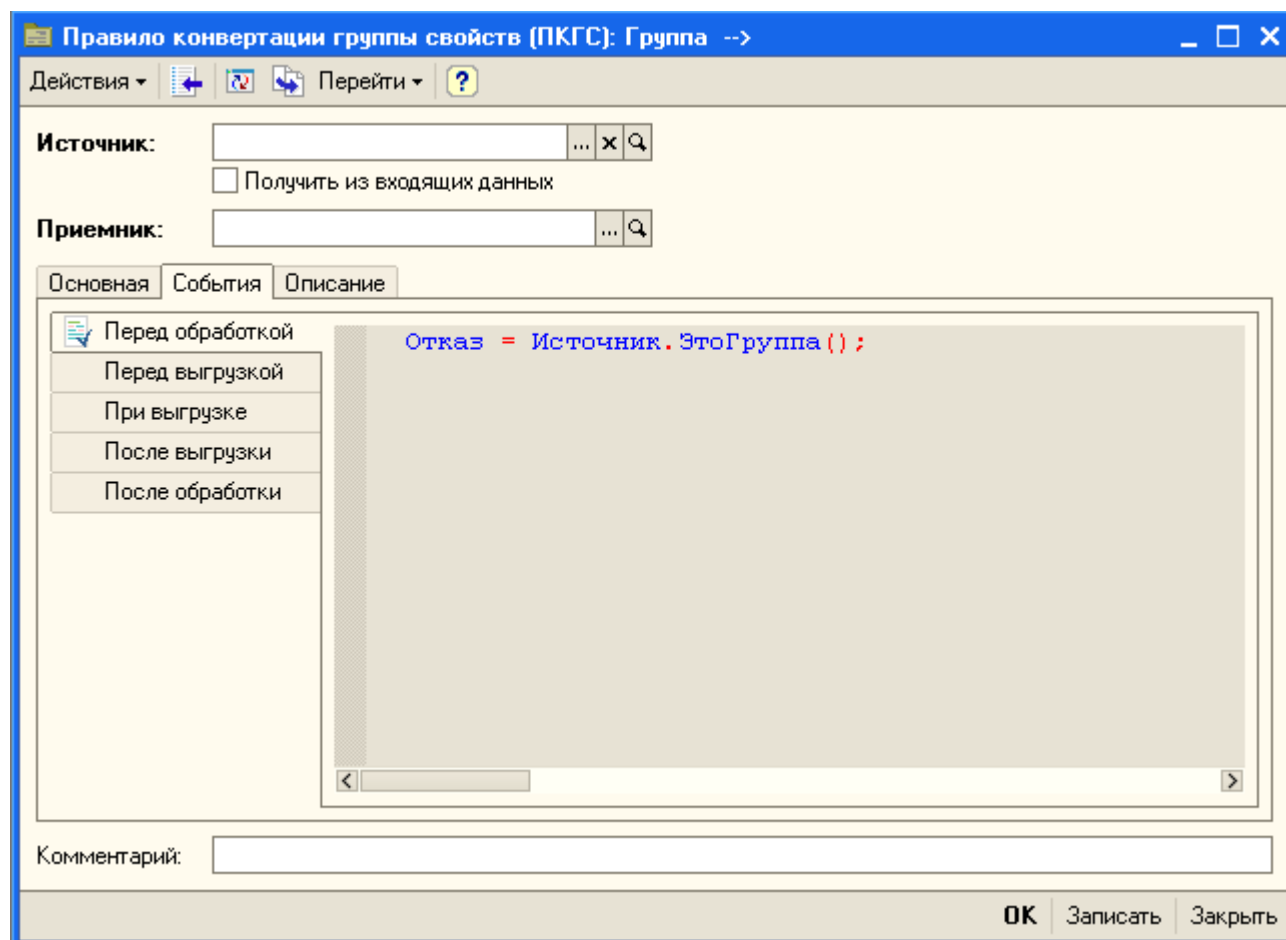
```
Значение = Источник.Наименование;
```

Теперь все реквизиты справочника будут заполнены корректно.

Ряд реквизитов справочников определен только для элементов, а для групп не

используются. Естественно ограничить выгрузку этих реквизитов только случаем обработки элемента. Для этого применяется следующая методика. Создадим группу справочника правила конвертации свойств и поместим в нее все правила, которые мы хотим обрабатывать только для элементов. Для группы правил конвертации свойств доступны два дополнительных обработчика "Перед обработкой" и "После обработки". Момент исполнения каждого очевиден из их названий. В обработчике "Перед обработкой" поместим следующий программный код:

```
Отказ = Источник.ЭтоГруппа ();
```

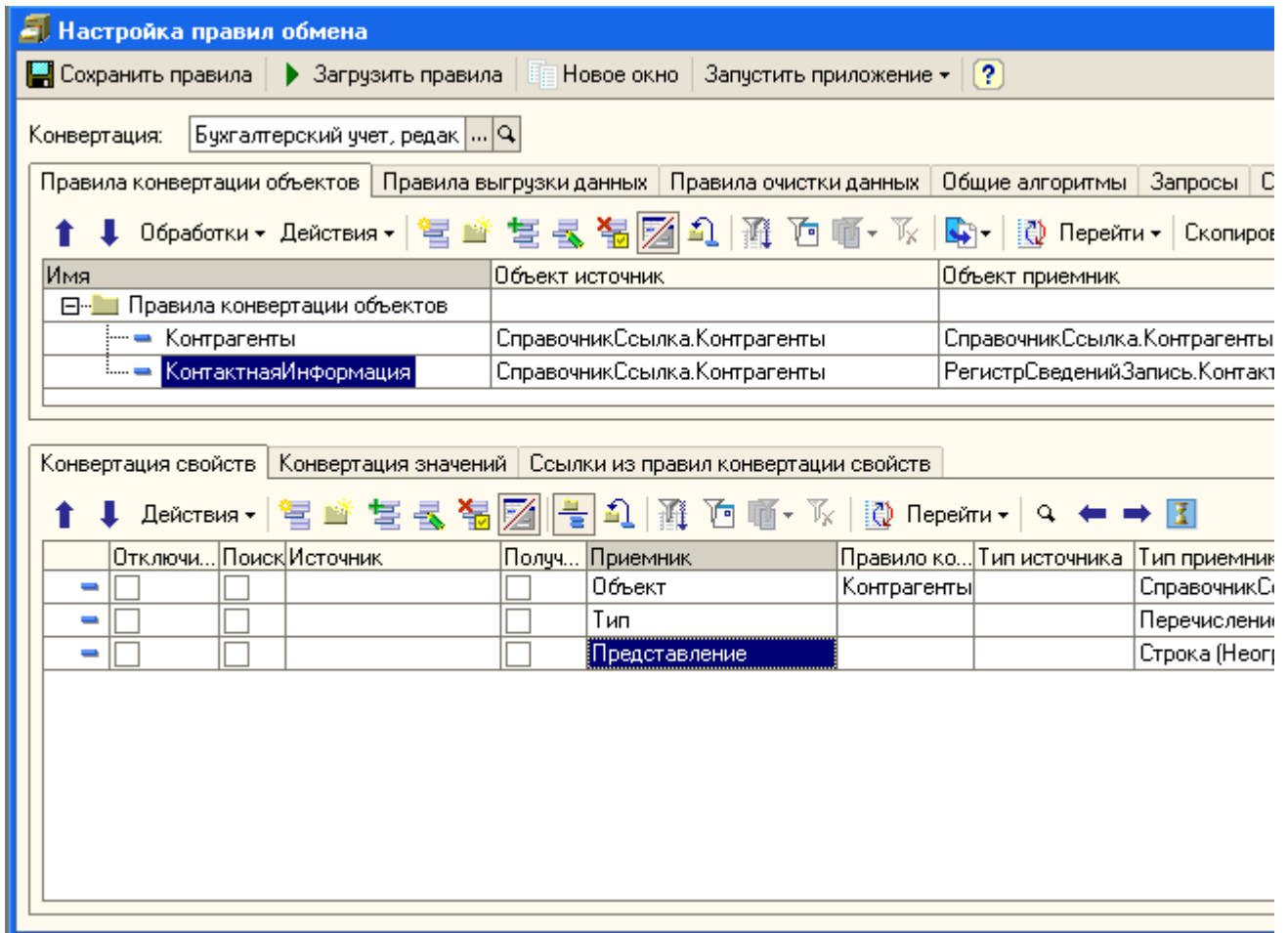


Здесь мы используем очень важную переменную Отказ. Управление этой переменной позволяет прервать процесс обработки правила конвертации свойств, а в данном случае группы правил. Переменная Отказ доступна практически во всех обработчиках событий и используется одинаково - для отказа от выполнения действия.

В заключении, этого раздела небольшое замечание об особенностях применения обработчиков для платформ "1С:Предприятия 7.7" и "1С:Предприятия 8". Большинство переменных-параметров можно использовать при программировании на обеих платформах. Однако, если Вы неуверенны, обратитесь к справочной информации, где описаны особенности использования обработчиков. Тип переменных может различаться в зависимости от платформы. Так для в обработках для "1С:Предприятия 7.7" переменная Отказ имеет тип число, а для "1С:Предприятия 8" - булево.

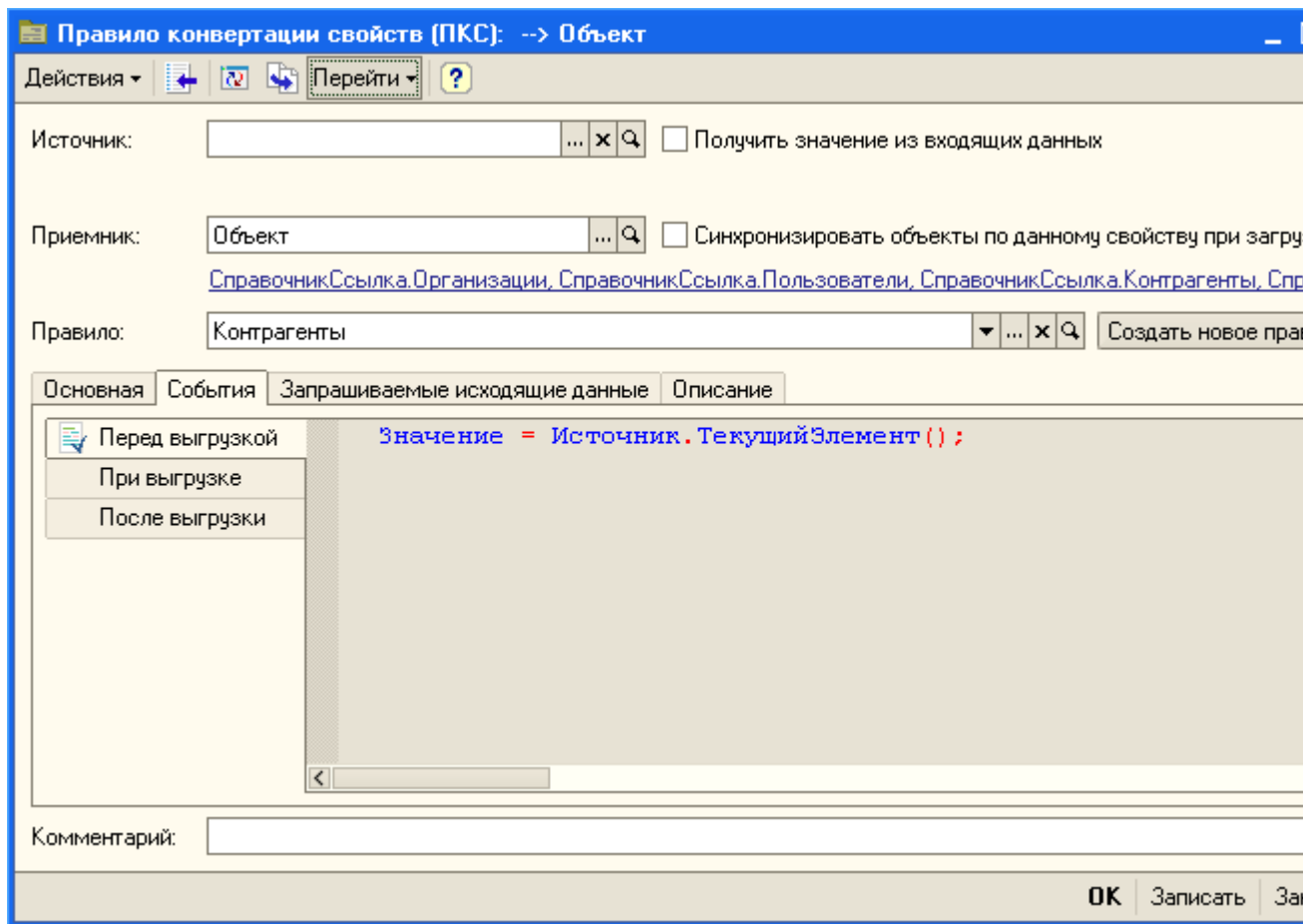
Формируем правила конвертации для регистра сведений

Сформируем правило конвертации КонтактнаяИнформация. Источником у нас будет снова справочник Контрагенты, а приемником регистр сведений КонтактнаяИнформация. Создадим три правила конвертации свойств. Все с пустым источником, а в качестве приемников зададим измерения Объект и Тип, и ресурс Представление.



И в этом случае мы используем обработчик "Перед выгрузкой". Для правила конвертации свойств Объект установим в качестве значения сам объект выгрузки:

`Значение = Источник.ТекущийЭлемент ();`



Работа со значениями перечисления нам уже знакома. Для измерения Тип используем такой обработчик:

```
Значение = "Адрес";
```

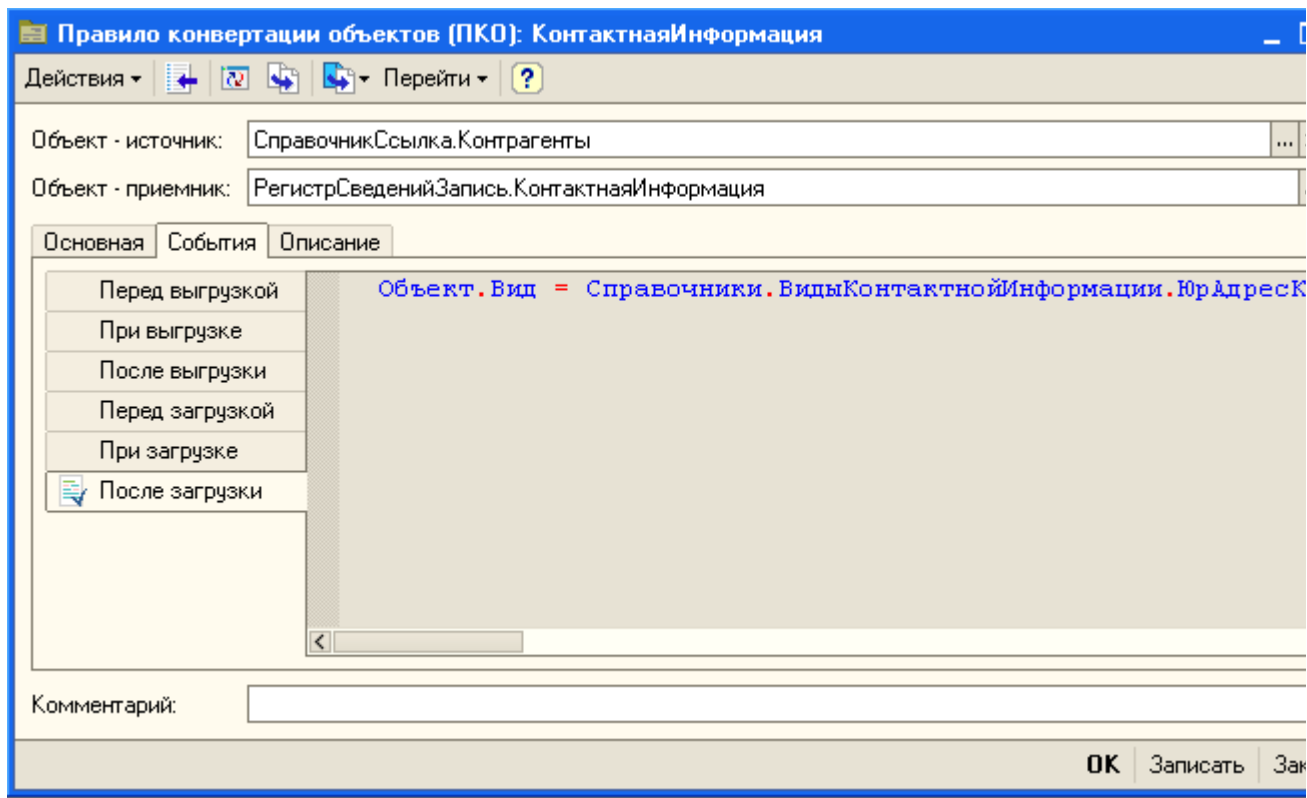
А для Представления такой:

```
Значение = Источник.ЮридическийАдрес;
```

Но у нас осталась очень серьезная проблема - измерение Вид, в котором нужно установить в качестве значения ссылку на predetermined элемент. Без этого запись регистра невозможна. Такую задачу крайне трудно решить при выгрузке данных. Для такого случая лучше всего воспользоваться обработчиками, которые выполняются при загрузке.

Отроем форму элемента правила конвертации объекта КонтактнаяИнформация и исследуем список доступных обработчиков событий. Их шесть. Первый три выполняются в процессе обработки правила конвертации, в ключевых точках алгоритма выгрузки, соответственно названия - "Перед выгрузкой", "При выгрузке" и "После выгрузки". А вот остальные - это нужные нам обработчики. Они выполняются перед загрузкой, при загрузке и после загрузке. Для нашей цели требуется обработчик "После загрузки". Он исполняется перед записью объекта в информационную базу. Программный код обработчика будет очень прост:

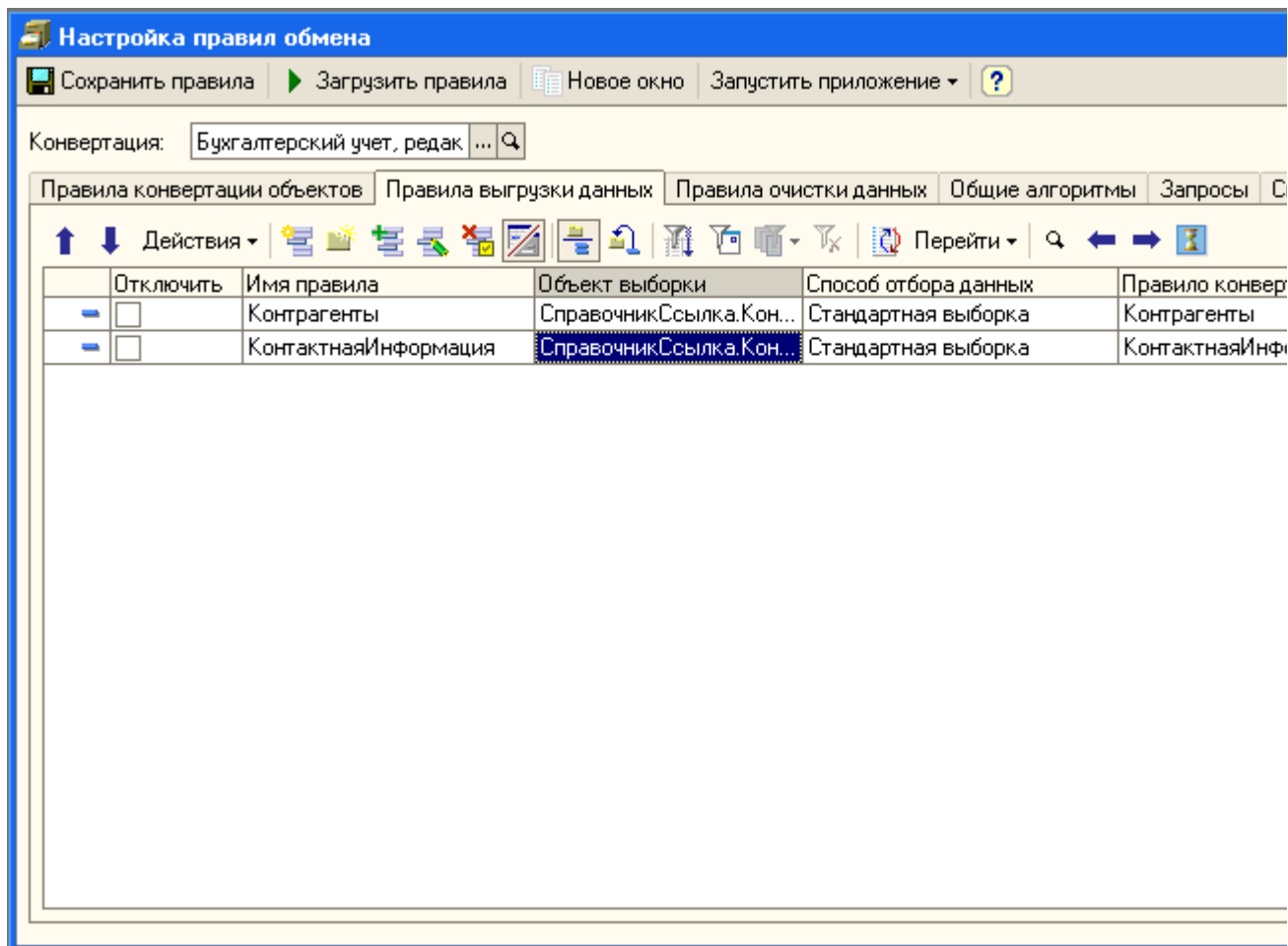
```
Объект.Вид = Справочники.ВидыКонтактнойИнформации.ЮрАдресКонтрагента;
```



Переменная Объект, которую мы использовали в этом случае, хранит ссылку на загружаемый объект данных. С ее помощью легко провести установку тех реквизитов, заполнение которых трудоемко при выгрузке.

Отбор данных с помощью глобальных обработчиков.

Когда правила конвертации завершены, можно обратиться к вопросам выгрузки данных. Поскольку гдала конвертации завершены, теперь можно обратиться к вопросам выгрузки данных. названия - " обработчки элемента правила. нас в данном случае не очень занимает производительность правил обмена, пойдём по самому простому пути - создадим для каждого правила конвертации свое правило выгрузки.



При исполнении правил справочник Контрагенты обрабатывается дважды. Но нам не хочется дублировать программный код для отбора данных в обработчиках для каждого правила выгрузки. Как быть? Для этого случая предусмотрен механизм глобальных обработчиков. Откроем форму элемента справочника Конвертации (это можно сделать прямо из "Настройки правил обмена") на закладке События. Мы увидим целых девять видов обработчиков событий.

Первый по порядку - "Перед выгрузкой данных". Этот обработчик выполняется в самом начале обработки правил обмена. Обычно, он используется для инициализации параметров конвертации. Второй обработчик - "После выгрузки данных". Он выполняется после всех операций по выгрузке, перед записью сохранением файла данных. Остальные обработчики нам уже знакомы - с ними мы встречались при использовании других элементов правил обмена (правил конвертации свойств и объектов и правил выгрузки данных). Обработчики, определяемые в элементе справочника Конвертация можно назвать глобальными, потому что они выполняются при обработке любого элемента правил обмена определенного типа. Так, обработчик "Перед выгрузкой объекта" выполняется перед выгрузкой объекта для всех правил выгрузки данных, строго перед локальным обработчиком "Перед выгрузкой объекта". Этот порядок выполняется для всех типов обработчиков - сначала выполняется глобальный, а сразу после этого одноименный локальный.

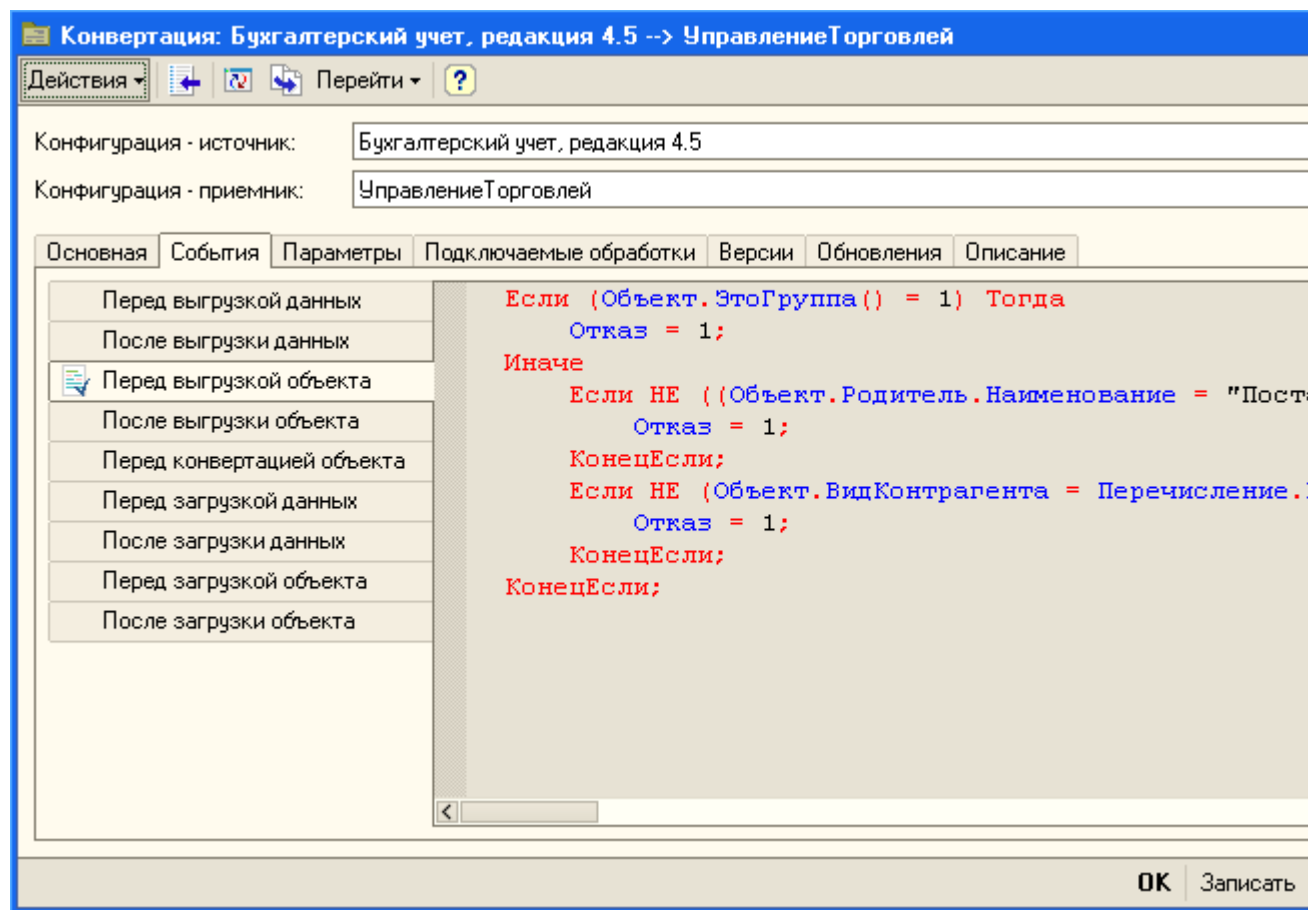
Глобальные обработчики очень удобны для нужд отбора данных, например, документов разных типов, имеющих общие реквизиты. При использовании глобальных обработчиков важно помнить, что обработчики "Перед выгрузкой объекта" и "После выгрузки объекта" выполняются при обработке правил выгрузки, а обработчик "Перед конвертацией объекта" выполняется при обработке правила конвертации, перед локальным обработчиком "Перед выгрузкой", который определяется в правилах конвертации объекта. Таки образом, если Вы хотите, программно контролировать выгрузку всех объектов, в том числе и тех, которые

выгружаются по ссылкам, используйте глобальный обработчик "Перед конвертацией объекта".

Вернемся к нашей задаче. Для отбора данных нам потребуется обработчик "Перед выгрузкой объекта". Программный код обработчика приведен ниже:

```
Если (Объект.ЭтоГруппа() = 1) Тогда
    Отказ = 1;
Иначе
    Если НЕ ((Объект.Родитель.Наименование = "Поставщики") ИЛИ
(Объект.Родитель.Наименование = "Покупатели")) Тогда
        Отказ = 1;
    КонечЕсли;
    Если НЕ (Объект.ВидКонтрагента = Перечисление.ВидыКонтрагентов.Организация)
Тогда
        Отказ = 1;
    КонечЕсли;
КонечЕсли;
```

Обратите внимание, что при написании кода использована особенность структуры данных конкретной информационной базы. В случае, когда используется более двух уровней иерархии, используйте другие методы.

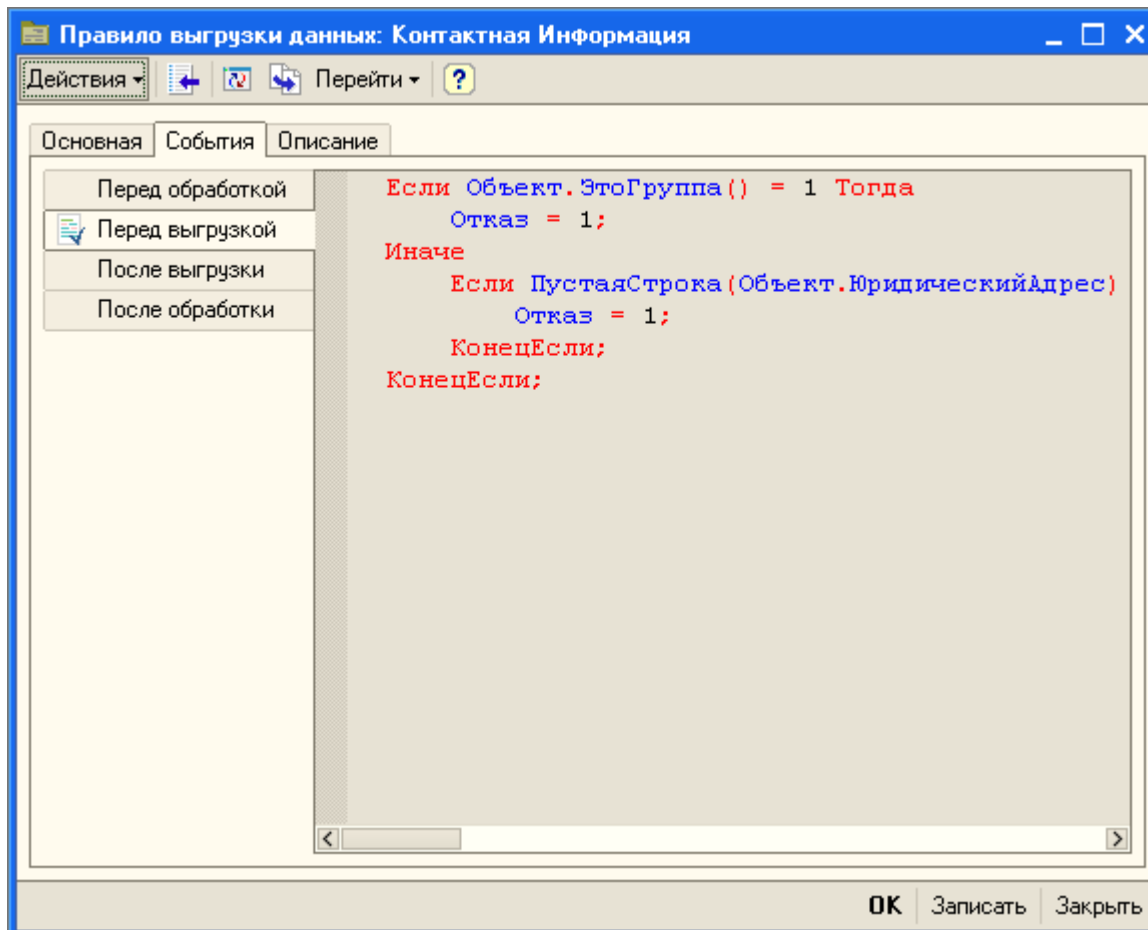


Понимание кода не составит труда - все использованные приемы программирования нам известны. Доступ к объекту выгрузки дает переменная Объект, а управление прерыванием выгрузки делается с помощью переменной Отказ.

Для правила выгрузки КонтактнаяИнформация нам потребуется дополнительный отбор, чтобы не выгружать в регистр сведений пустые строки, в том случае, когда адрес не зарегистрирован. Откроем форму правила выгрузки на закладке События и поместим в поле "Перед выгрузкой" следующий код:

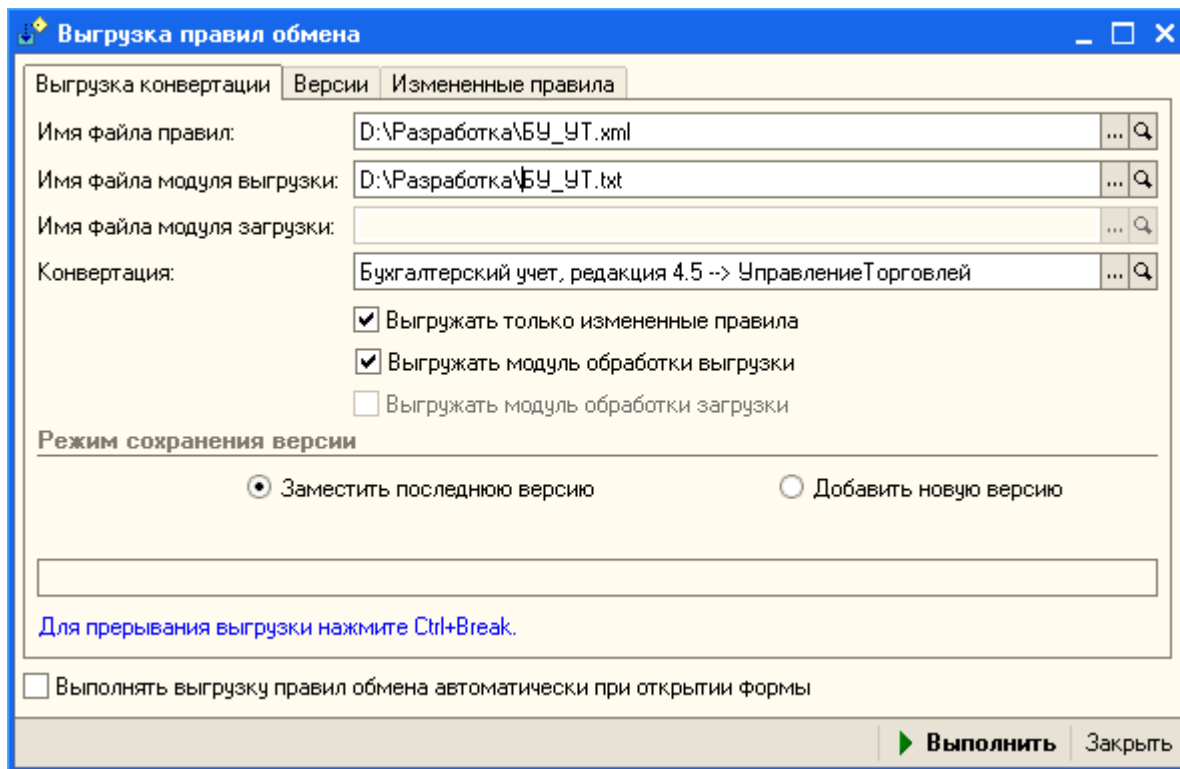
```

Если Объект.ЭтоГруппа() = 1 Тогда
    Отказ = 1;
Иначе
    Если ПустаяСтрока(Объект.ЮридическийАдрес) = 1 Тогда
        Отказ = 1;
    КонечЕсли;
КонечЕсли;
    
```

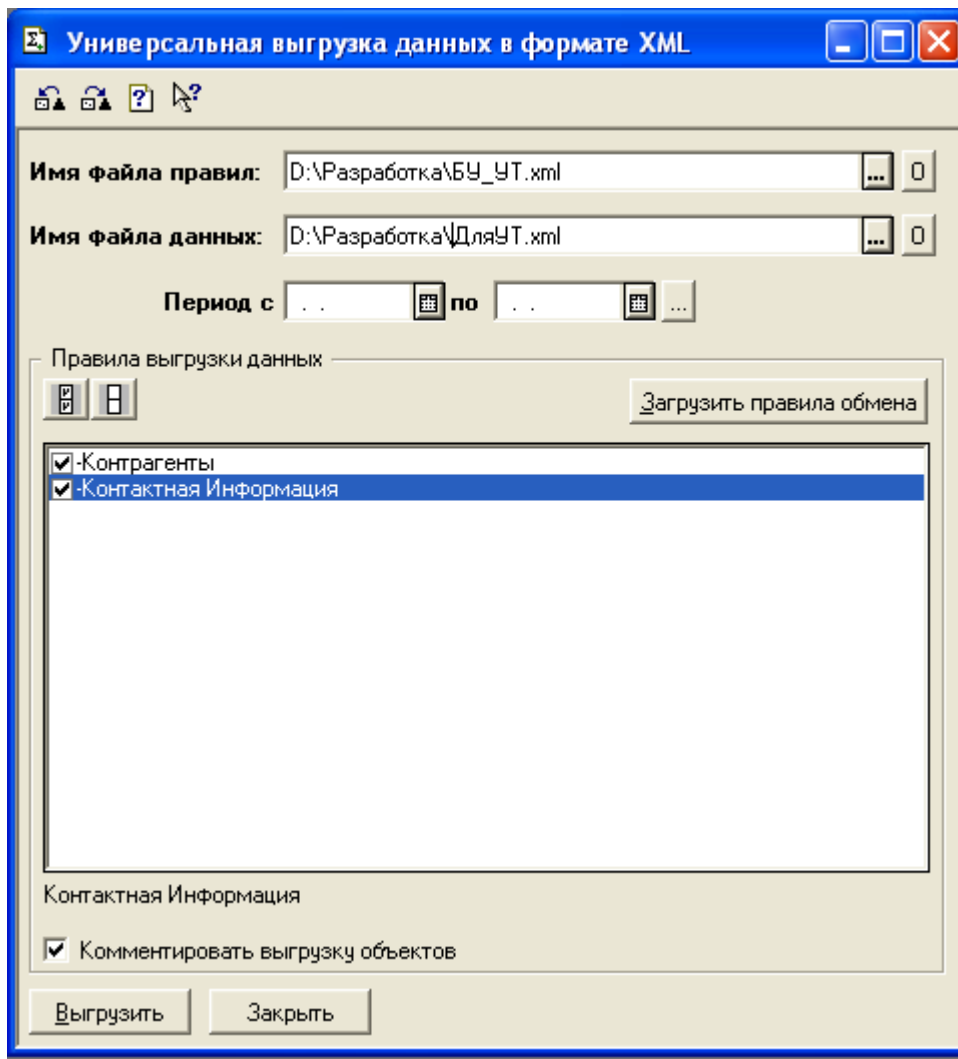


Обратите внимание на состав обработчиков событий для правила выгрузки. В самом начале обработки правила выполняется код "Перед обработкой". Этот обработчик часто используется для программного формирования выборки данных. Обработчик "После обработки" выполняется в самом конце алгоритма обработчик правила выгрузки. "Перед выгрузкой" и "После выгрузки", соответственно перед выгрузкой объекта и после выгрузки объекта.

На этом формирование правил обменов закончено - нам осталось только сохранить правила в файл.



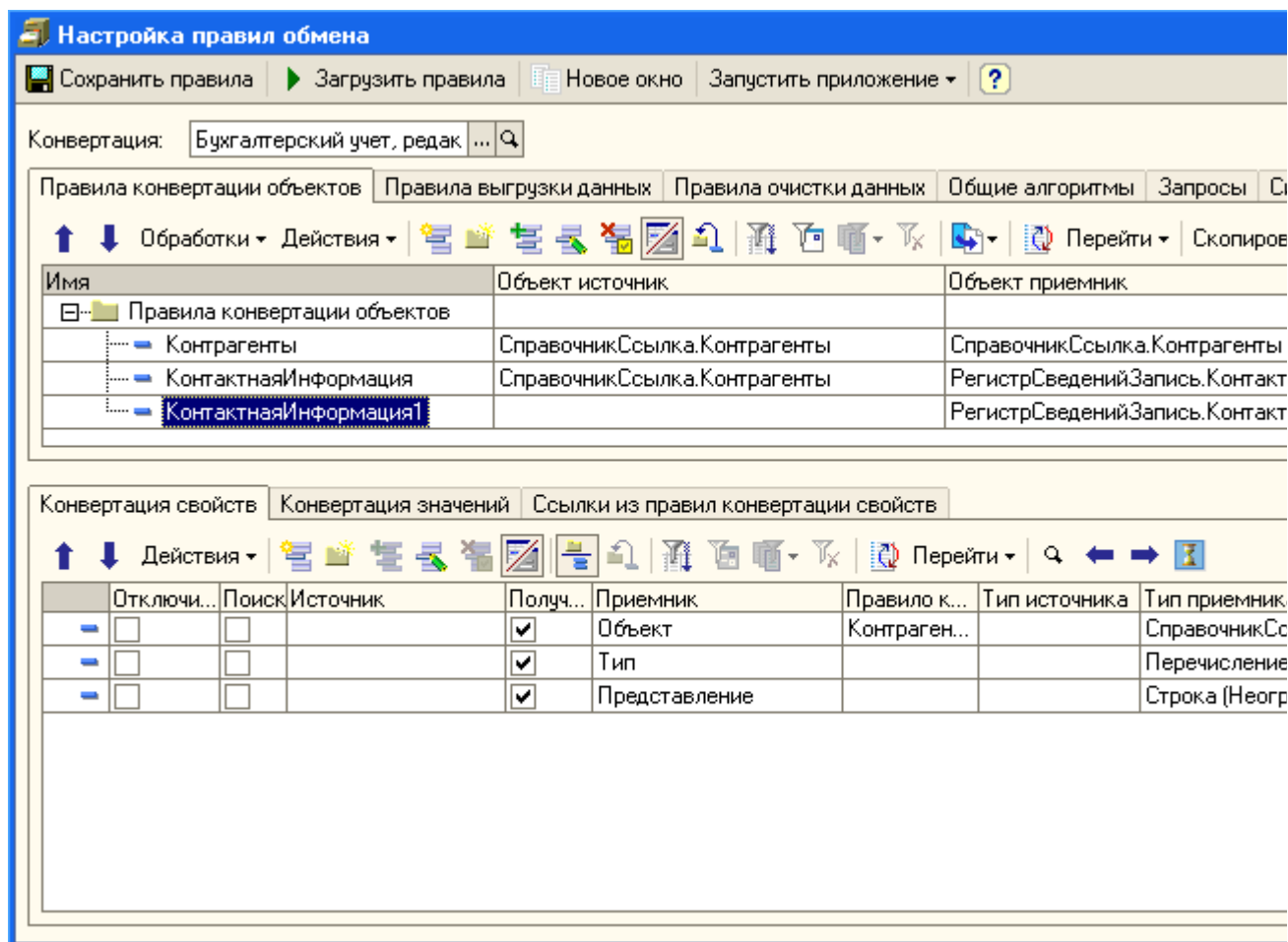
При выгрузке правил обмена система генерирует модуль для обработки V77Exp.ert, который содержит функции обработчиков событий. Текст модуля сохранится в файле, имя которого следует назначить в форме сохранения правил обмена. После сохранения правил нам нужно открыть обработку V77Exp.ert в конфигураторе "1С:Предприятия 7.7" и полностью заменить текст модуля на новый, взятый из текстового файла, созданного "Конвертацией данных 2.0". Обработку рекомендуется сохранить под другим именем и снабдить описанием. После чего можно начать выгрузку данных.



Оптимизация правил обмена

Правила обмена подготовлены, конвертация данных успешно завершилась. Но, убедившись в работоспособности нашей системы, хотелось бы обратить внимание на ее оптимизацию. Для выгрузки у нас используются два правила, при этом один и тот же справочник мы обходим дважды. Естественно, хотелось бы этого избежать. Лучше всего чтобы адрес выгружался сразу же, при выгрузке контрагента. Но как это сделать? Умелое использование обработчиков событий поможет нам и на этот раз.

Первым делом сделаем копию правила конвертации КонтактнаяИнформация, с помощью кнопки "Скопировать правило". У нового правила будет пустой источник, а приемник останется тот же - регистр сведений. Обработчики "При выгрузке" нам не понадобятся. Все свойства мы будем заполнять данными, переданными из правила конвертации объекта Контрагенты. Для этого у всех правил конвертации свойств установим флажок "Получить значение из входящих данных".\



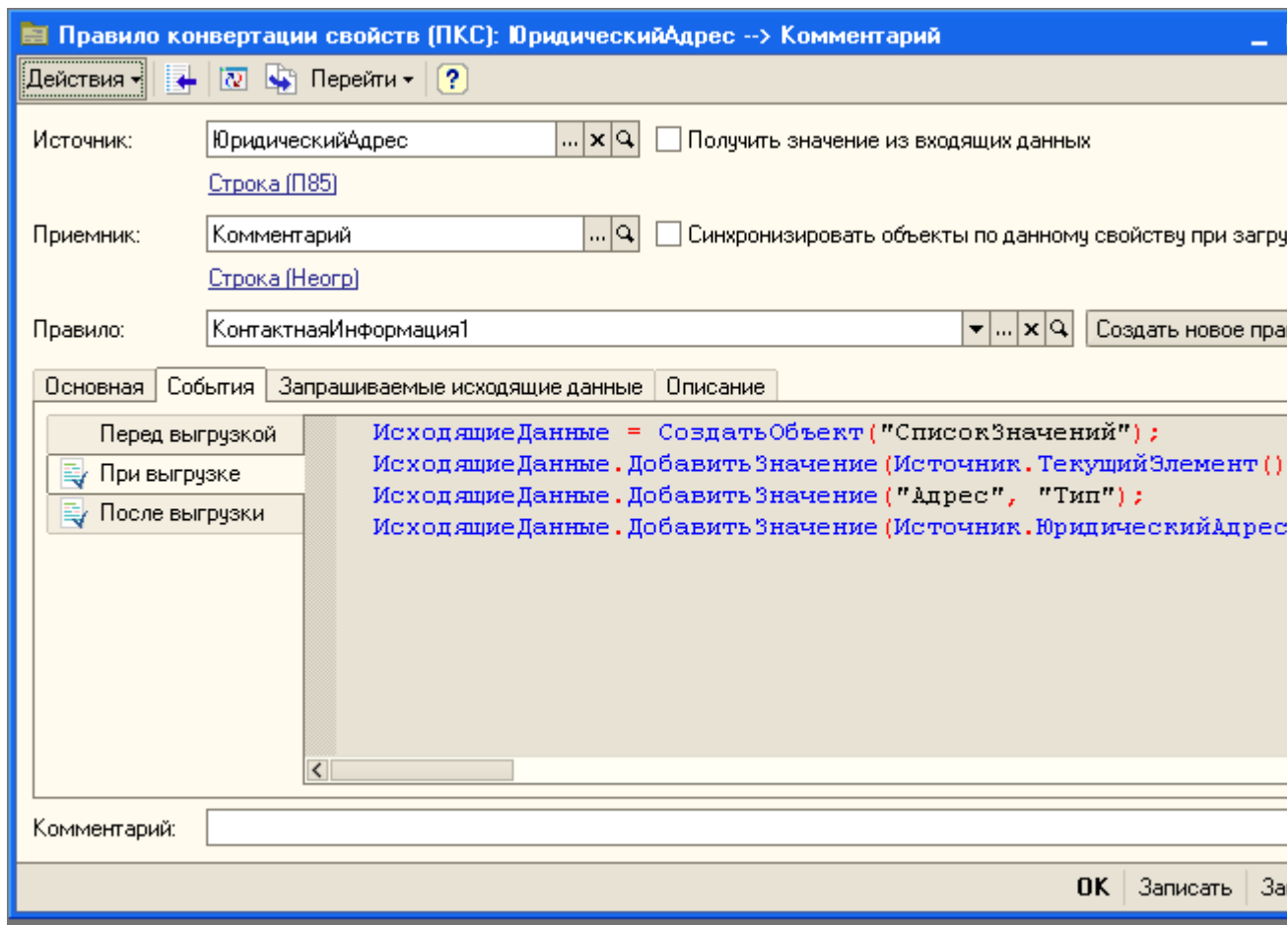
Теперь вернемся к правилу конвертации Контрагенты. Для выгрузки адреса нам придется пойти на определенную уловку. Сделаем правило конвертации свойств с источником ЮридическийАдрес, а в качестве приемника возьмем Комментарий. Реквизит, который не используется и остается пустым при загрузке. Выберем в поле Правило наше новое правило конвертации свойств.

Мы хотим добиться чтобы при обработке этого правила вызывалось правило конвертации КонтактнаяИнформация1, но при этом реквизит Комментарий оставался пустым, поскольку ссылка на запись регистра сведений абсолютно нам не нужна, и более того ее присутствие может привести к ошибке при загрузке. Для того чтобы ссылка не была записана, используем обработчик "После выгрузки" со следующим программным кодом:

```
Отказ = 1;
```

А для формирования данных для выгрузки регистра сведений обработчик "При выгрузке":

```
ИсходящиеДанные = СоздатьОбъект ("СписокЗначений" );
ИсходящиеДанные.ДобавитьЗначение (Источник.ТекущийЭлемент ( ), "Объект" );
ИсходящиеДанные.ДобавитьЗначение ("Адрес", "Тип" );
ИсходящиеДанные.ДобавитьЗначение (Источник.ЮридическийАдрес, "Представление" );
```



Разберем этот программный код более подробно. Наша задача - передать данных правилу конвертации. Для этого используется переменная ИсходящиеДанные. При программировании для платформы "1С:Предприятие 7.7" инициализировать ее нужно как список значений, а для платформы "1С:Предприятие 8" как структуру. Причем символьные представление значений должны строго соответствовать именам приемников в правилах конвертации свойств, для которых установлен флаг "Получить значение из входящих данных". Переданные данные доступны в правиле конвертации объекта с помощью переменной ВходящиеДанные.

Теперь юридический адрес будет выгружаться при конвертации контрагентов, поэтому не требуется обходить справочник Контрагенты два раза. Надо только не забыть отключить использование правила выгрузки данных КонтактнаяИнформация.

Заключение

Подведем некоторые итоги. Мы рассмотрели практически все виды обработчиков событий конфигурации "Конвертации данных 2.0". Некоторые из них были исследованы более детально и глубоко. В качестве рекомендации для дальнейшего совершенствования в программировании обработчиков хотелось бы порекомендовать изучение справочной информации, где описаны все типы обработчиков событий и переменные-параметры. Для самостоятельно тренировки рекомендуем Вам доработать правила, описанные в статье самостоятельно. Добавив выгрузку договоров, расчетных счетов и объектов контактной информации Вы сможете получить твердый навык построения правил обмена, что позволит Вам уверенно решать любые задачи конвертации данных.

[Установить конфигурацию "Конвертация данных"](#)