

Конвертация данных. Методика работы и примеры.

этот методический материал размещен на сайте

www.MyKoD.info

и доступен для скачивания с разрешения автора

Вместо предисловия.	3
Конвертация данных. Как ее можно использовать?.....	3
Методика настройки правил. Как это делается.	3
Подготовка к переносу данных: несколько нехитрых действий.....	3
Хранение структуры метаданных в конфигурации.....	5
Обновление метаданных конфигурации.	6
Обновление стандартным способом.	6
Обновление через объединение правил.....	7
Настройка ПКО для перечислений.	11
Настройка ПКО для справочников.	12
Дополнительные настройки ПКО.	14
Правила выгрузки данных: стандартный и произвольный алгоритм.....	15
Автоматическая настройка правил.	16
Выбор полей поиска и поиск элементов в базе-приемнике.....	18
Настройка ПКО для планов видов характеристик, планов счетов и справочников с предопределенными элементами.	19
Настройка ПКО для регистров сведений.	19
Использование параметров конвертации.	20
Алгоритмы и Запросы.	20
Правила очистки данных.	21
Ключ Выгружаемых Данных.....	22
Функция ВыгрузитьПоПравилу().....	22
Несколько технических моментов. На что обратить внимание.	23
Режим «ОбменДанными.Загрузка».....	23
Пометка на удаление. Почему она установлена?	24
Задачи по переносу данных и их решение.	24
Пример: Виды Номенклатуры. Конвертация значения типа Булево в справочник.	24
Пример: Конвертация справочника Классификатор единиц измерения в справочник Единицы измерения. Один элемент в несколько.....	26
Пример: Справочник Организации. Настройка правила без источника	30
Пример: Документ-Основание. Настройка ПКС для свойств с составным типом данных	32

Пример: Штрих Коды. Перенос реквизита справочника как записи регистра сведений и синхронизация объектов по этой записи.	33
Пример: Поиск контрагентов по ИНН. Синхронизация по полю, недоступному для группы.	37
Пример: Справочник Организации. Ключ выгружаемых данных и оптимизация выгрузки.	38
Пример: Конвертация групп и элементов справочников по разным правилам.	40
Пример: Конвертация Характеристик Номенклатуры в Номенклатуру.	42
Пример: Операция (бухгалтерский учет). Конвертация таблицы значений в документ.	44
Пример: Перенос курсов валют 8 – 8.	48
Пример: Перенос курсов валют 7.7 – 8.	49
Разбор типичных примеров из демонстрационной базы.	51
Два в один.	51
Два объекта в один.	52
Один объект в два.	54
Один реквизит в два.	55
Перечисление в справочник (на этапе выгрузки).	56
Перечисление в справочник (на этапе загрузки).	58
Число в перечисление.	61
Особенности работы с платформой 7.7.	62
Формат XML. Что это такое?.	63
Запись в файлы правил и данных. Общая схема.	63
Запись объектов в файл данных. Как это происходит в программе?.	67
Работа над ошибками.	69
Некоторые часто возникающие ошибки. При выгрузке.	70
При загрузке.	72
Принятые сокращения.	73
Источники информации.	73
Контакты.	73

Мануал написан под версию конвертации данных 2.0.20.1 и обработки, входящие с ней в комплект поставки.

Для получения изображений xml-файлов в виде дерева была использована обработка *XML редактор v1.erpf*, которая включена в архив с данным руководством. Эта версия обработки для работы на платформе 8.0, для использования ее на платформе 8.1 необходимо произвести конвертацию этой обработки обычным путем.

Вместо предисловия

Все мы любим, когда нас окружают самые лучшие вещи, особенно когда эти вещи помогают нам в работе. Думаю, не в последнюю очередь это относится к учебникам, справочникам и методическим материалам. Они экономят время нам и нашим коллегам. Если Вы в своей работе сталкиваетесь с «Конвертацией данных», то в Ваших силах помочь этой методичке стать самым лучшим, толстым и обстоятельным пособием по этой конфигурации. Если у Вас возникла задача по переносу данных, которая вызвала затруднения, если Вы нашли неточности или ошибки в тексте данной методички, если в ней что-то написано непонятно, так, что Вы не можете разобраться, или есть предложения, что еще полезного можно в нее включить, пожалуйста, присылайте эту информацию мне ([см. раздел Контакты](#)).

Конвертация данных. Как ее можно использовать?

Конвертация данных 2.0 – это конфигурация базы данных на платформе 1с 8. Главная задача, которую можно решить с помощью конвертации – это преобразование любых данных, хранящихся в базе данных на платформе 1С 7.7, 1С 8.0 или 1С 8.1. С помощью конвертации можно настроить правила переноса данных, например, между базой данных центрального офиса компании и филиала, между подразделениями компании, удаленными территориально, можно выполнить перенос данных для компании, начинающей использование другой базы данных.

Благодаря многочисленным возможностям и наличию обработчиков событий, конвертация данных позволяет выполнить практически любое преобразование данных.

Настройка программы для удобства работы

В меню сервис в Конвертации данных доступен пункт Настройки пользователя. Там три настройки:

- Выгружать правила обмена при открытии формы настройки
- Не использовать мастер настройки для ПКО
- При сохранении правил не проверять наличие полей поиска.

Если флажок третьей настройки не установлен и имеются правила, где по каким-либо причинам не установлены флажки поиска, при сохранении правил каждый раз будет выдаваться сообщение с предложением проставить поля поиска автоматически.

Методика настройки правил. Как это делается

Подготовка к переносу данных: несколько нехитрых действий

Основную часть времени, которую разработчику придется затратить на перенос данных, он будет модифицировать правила обмена данными. Но перед этим необходимо

загрузить в базу данных «Конвертация данных 2.0» описания метаданных базы-источника и базы-приемника.

1. Выгрузка файла структуры конфигурации. Для этого используются обработки, входящие в комплект поставки конфигурации «Конвертация данных». Для разных версий платформы подходят обработки MD77Exp.ert, MD80Exp.epf, MD81Exp.epf.

Базы источник и приемник открываются в режиме 1с:Предприятие, там запускается подходящая обработка и указывается путь и название файла xml, куда будет произведена выгрузка данных.

В восьмерке в окне обработки кроме поля для ввода пути к файлу есть еще флажки настройки. Флажки позволяют не переносить информацию о тех объектах конфигурации, которые не будут использованы напрямую в процессе преобразования данных. Это позволяет сократить время на перенос информации, размер файла обмена и размер базы Конвертации данных.

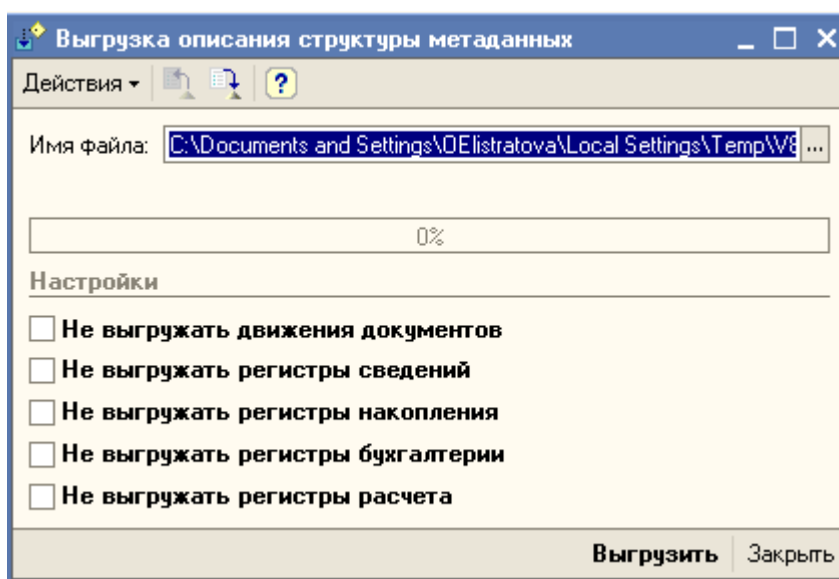


Рис. 1.

2. Загрузка описания структуры метаданных производится с помощью встроенной в конфигурацию «Конвертация данных» обработки. В окне обработки надо указать путь к файлу с описанием структуры метаданных и нажать кнопку «Выполнить загрузку».

Загруженная информация будет не доступна для редактирования вручную (исключение составляет возможность ввода новых значений для настройки правил конвертации значений). Если после начала работы по настройке правил переноса данных, структура конфигурации базы-источника или базы-приемника изменилась, то необходимо обновить информацию о структуре базы в Конвертации.

3. После загрузки метаданных создаем новую конвертацию (элемент справочника Конвертации), в ней указываем базу-источник и базу-приемник из списка баз, описания которых были загружены.

Подробность. В качестве конфигурации-источника и конфигурации-приемника может быть и один и тот же элемент справочника «Конфигурации», если мы переносим данные между двумя базами с одинаковыми конфигурациями, или если мы планируем загружать данные в ту же базу, из которой будем выгружать. Конфигурации не привязаны к конвертации однозначно, одну и ту же конфигурацию можно использовать для настройки нескольких правил переноса данных, и можно создать несколько различных правил для одной и той же пары конфигураций для разных целей.

Кроме того, в форме элемента справочника Конвертации можно настроить некоторые механизмы, которые, возможно, понадобятся при переносе. Это

- 1) Обработчики событий – глобальных, выполняемых по одному разу перед и после переноса данных, а также общих для всех переносимых объектов.
- 2) Параметры – это глобальные переменные, которые можно использовать во всех обработчиках событий.
- 3) Подключаемые обработки.

До начала настройки правил переноса данных, как правило, не понятно, понадобятся нам эти механизмы, или нет.

Хранение структуры метаданных в конфигурации



Загруженная структура метаданных хранится в справочниках «Конфигурации», «Объекты», «Свойства» и «Значения». В справочнике «Конфигурации» хранится общая информация о конфигурации и местоположении базы данных. Справочник «Объекты» подчинен справочнику «Конфигурации» и содержит описание объектов конфигурации (Справочников, документов и проч.). Справочники «Свойства» и «Значения» подчинены справочнику «Объекты», и там хранится информация о свойствах (Реквизитах, измерениях, ресурсах) и о значениях (значениях перечислений и счетах планов счетов, predetermined elements справочников и планов видов характеристик) объектов конфигурации.

Табличные части являются группами справочника «Свойства», а реквизиты табличных частей – вложенными в эти группы элементами.

Загруженное описание конфигурации можно посмотреть с помощью общей формы «Конфигурация», которая открывается из интерфейса или из формы элемента справочника «Конфигурации» по кнопке «Показать описание конфигурации». Здесь структура объектов конфигурации показана в виде дерева, где группами являются типы данных «Справочники», «Документы», «Перечисления» и проч.

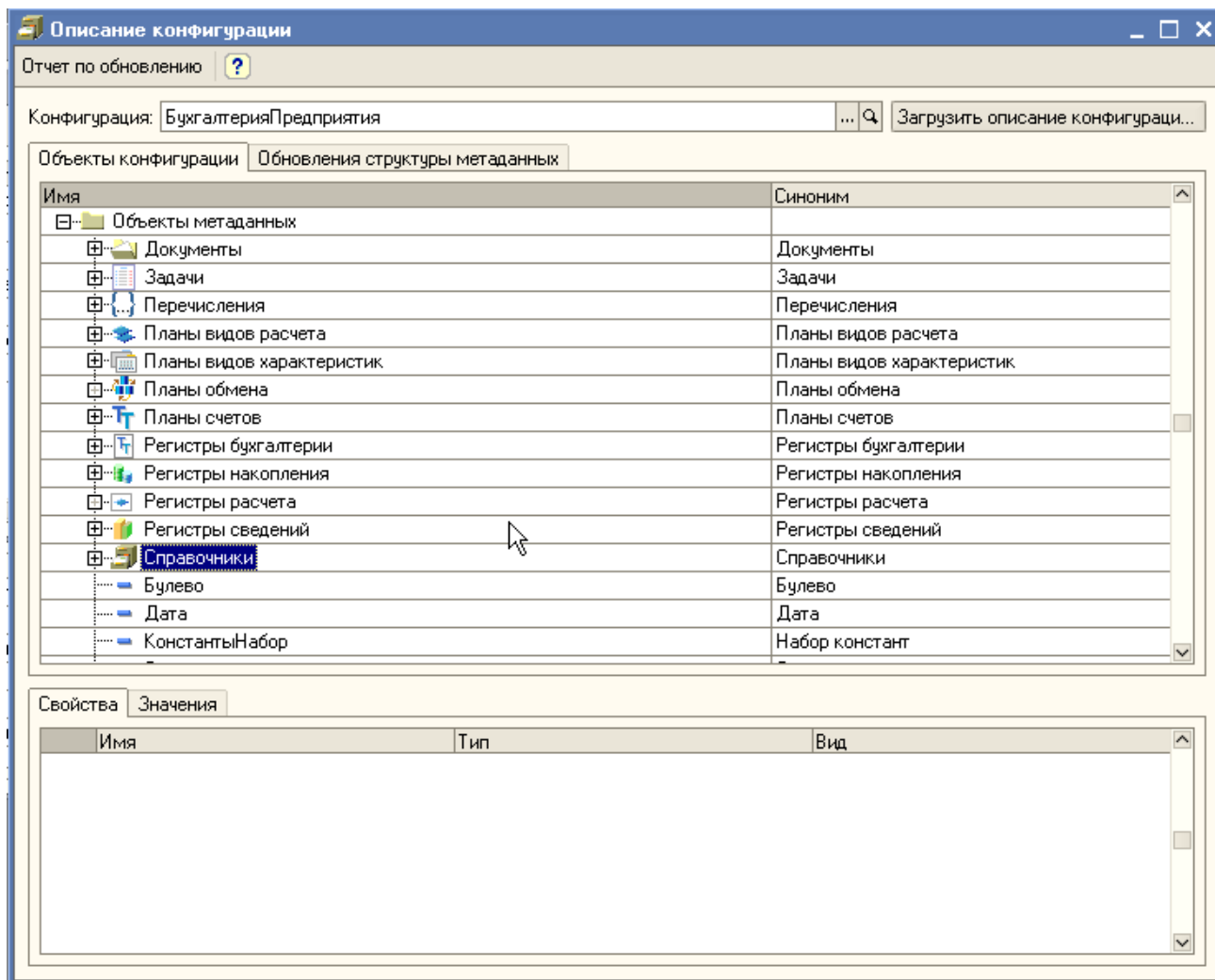


Рис. 2.

Если на этапе выгрузки структуры метаданных установить, например, флажок «Не выгружать регистры накопления», то в этом дереве группа «Регистры накопления» будет пуста. Здесь же можно посмотреть наборы свойств и значений любого объекта, если установить на него курсор.

Практически вся информация в этих справочниках недоступна для редактирования вручную, а изменяется только путем обновления структуры метаданных конфигурации. Однако возможность добавлять и изменять элементы справочника «Значения» оставлена пользователю, что удобно, если в конфигурации, например, всего лишь добавлено значение перечисления, чтобы не перегружать метаданные целиком.

Обновление метаданных конфигурации

Обновление стандартным способом

Бывает, что одна из конфигураций, между которыми должен происходить обмен данными была изменена уже после того, как настройка правил обмена была начата. В данном случае речь идет о любом изменении структуры метаданных. Например, в справочник был добавлен новый реквизит, который также необходимо переносить при обмене данными.

В конфигурации «Конвертация данных» ограничена возможность ручного добавления или изменения объектов или свойств (не закрыта лишь возможность добавить значение). Поэтому в конфигурации есть возможность обновления структуры метаданных.

Перед этой процедурой, как и перед любым автоматическим изменением данных, желательно выгрузить базу данных конвертации и сохранять ее до тех пор, пока на

реальных данных не будут протестированы правила обмена, созданные на основе обновленной конфигурации.

Для обновления сначала необходимо открыть ту базу, конфигурация которой была изменена и выгрузить структуру метаданных так, как рассказывалось выше. Когда описание метаданных выгружено в xml-файл, открываем «Конвертацию данных» и нажимаем «Загрузить структуру метаданных конфигурации».

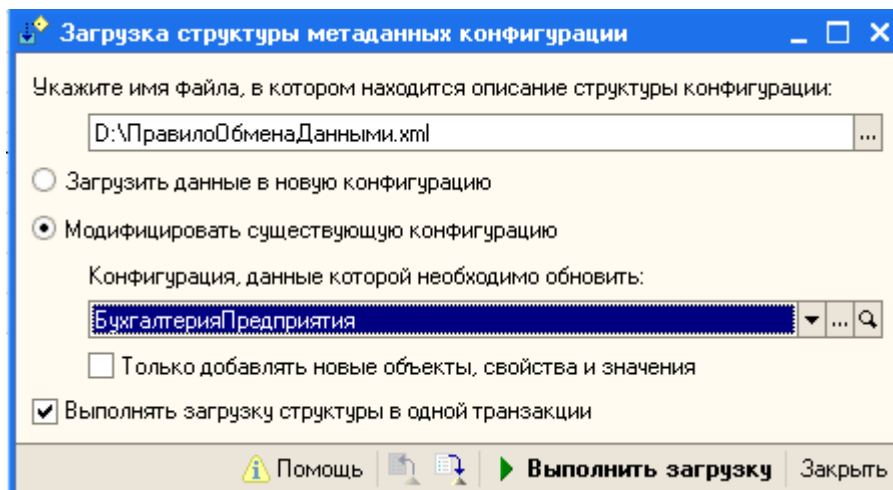


Рис. 3.

В открывшемся окне переключатель надо установить в позицию «Модифицировать существующую конфигурацию» и указать, какую именно конфигурацию обновляем. Флаг «Только добавлять новые объекты, свойства и значения» можно установить, если уже загруженные данные изменять не требуется. Далее нажать «Выполнить загрузку».

Если загрузка данных прошла успешно, то наша конфигурация обновлена удачно, и можно изменять правила, созданные на ее основе.

Однако практика показывает, что этот механизм может выдавать ошибки.

Например, если метаданные обновленной конфигурации сначала загрузить как отдельную конфигурацию, а потом попытаться обновлять этими же метаданными уже существующую конфигурацию, возникает ошибка «Ошибка при попытке вставке записи с неunikальным значением ссылки. Таблица Reference7».

Поэтому есть еще один способ – загрузить измененную конфигурацию, создать правила только для добавленных (измененных) объектов и свойств, а потом объединить эти правила.

Обновление через объединение правил

Например, мы переносили данные из базы «Бухгалтерский учет» в базу «Бухгалтерия предприятия».

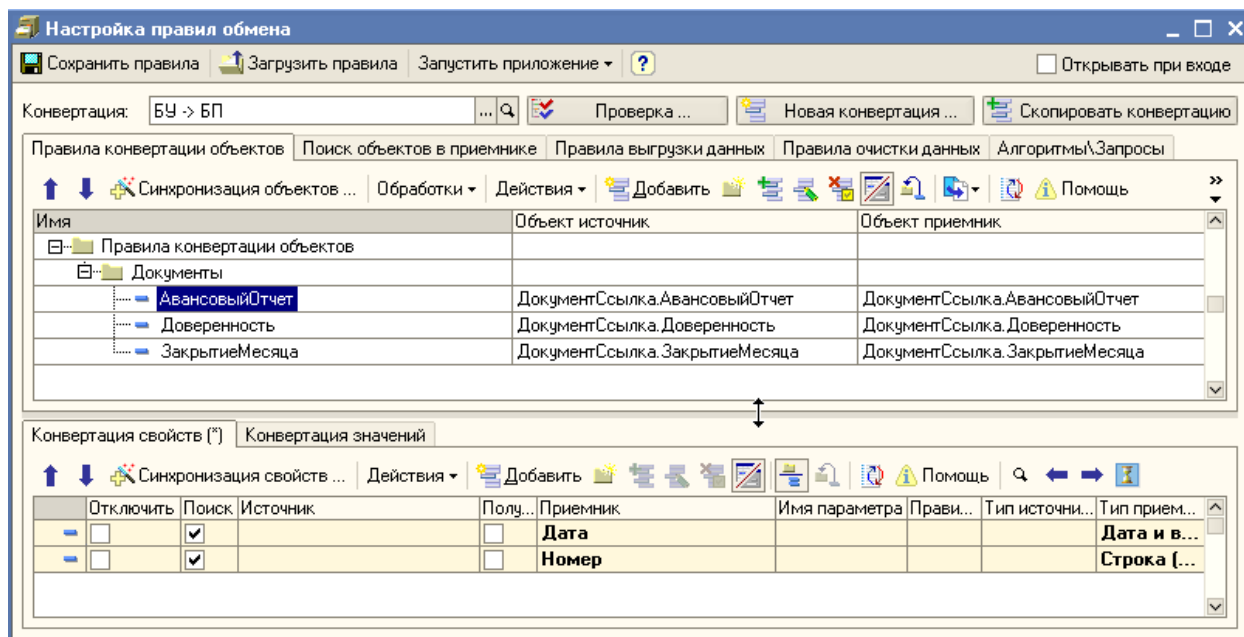


Рис. 4.

Но конфигурация базы «Бухгалтерский учет» изменилась. Обновляем правила с помощью объединения правил.

Для этого

- Загружаем описание метаданных измененной конфигурации в новую конфигурацию. Назовем ее «БУ Новая».

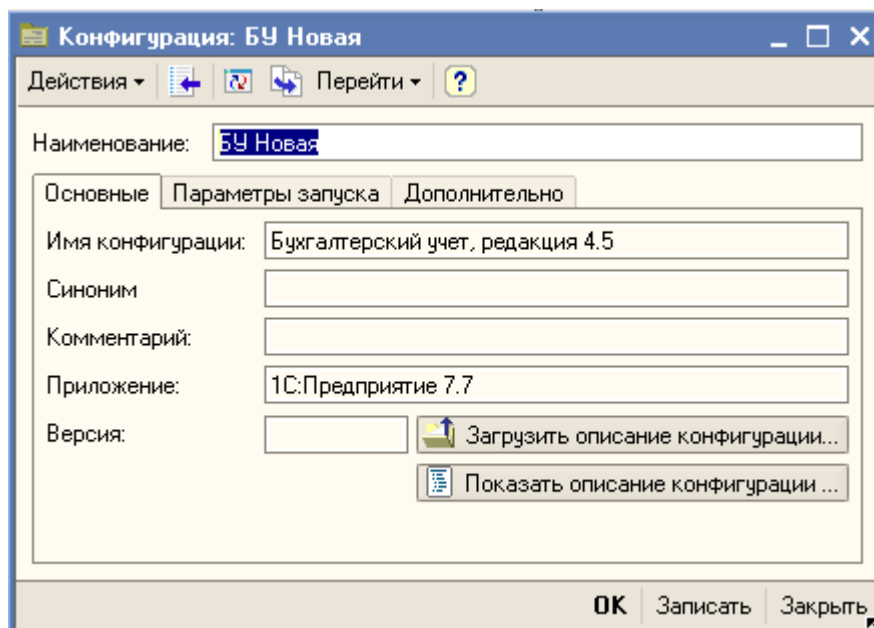


Рис. 5.

- Создаем новую конвертацию. Выбираем в качестве источника конфигурацию «БУ Новая», а в качестве приемника – «Бухгалтерия предприятия». Назовем эту конвертацию «БУ ->БП Измененная».

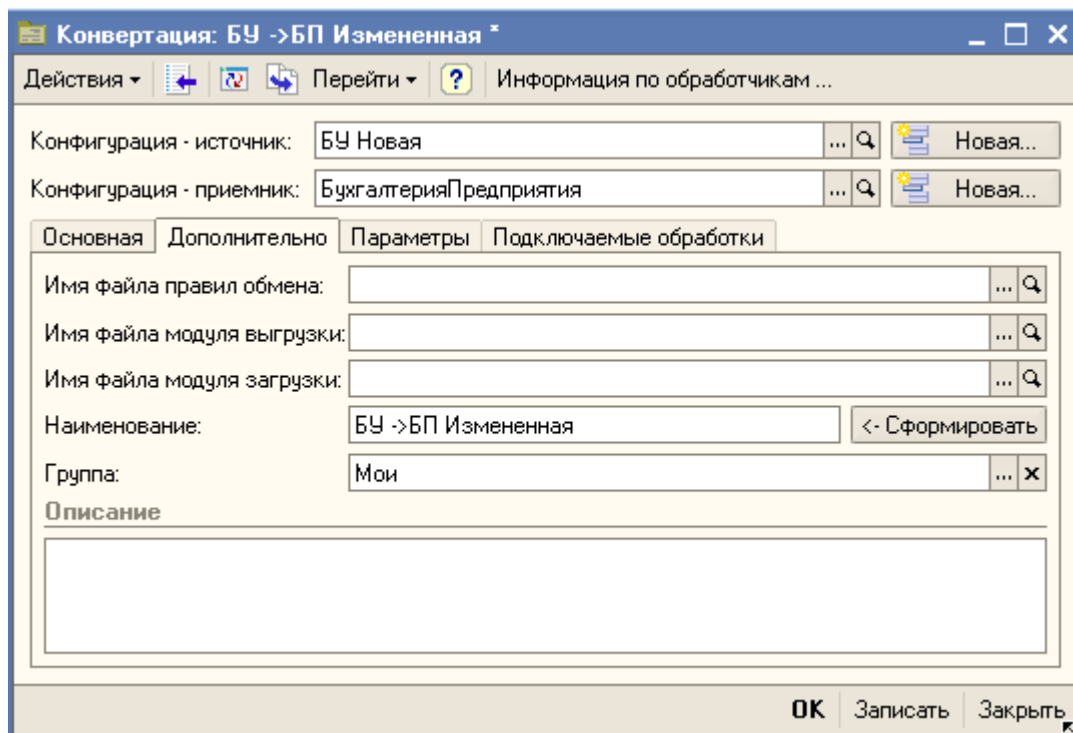


Рис. 6.

- Создаем в этой конвертации ПКО только для тех объектов, которые были добавлены или изменены.

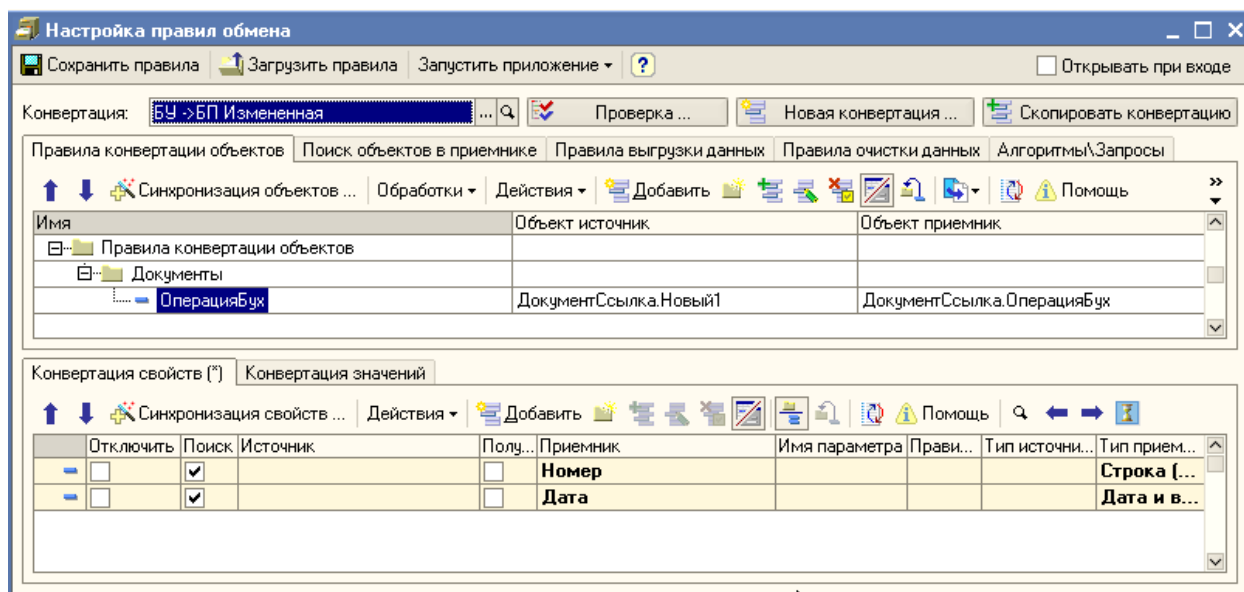


Рис. 7.

- Далее одно из этих двух правил выгружаем в файл xml. Лучше всего выгрузить то правило, в котором больше ПКО. Обычно это старые правила, для примера выгружаем их. Процедура выгрузки правила такая же, как для использования.

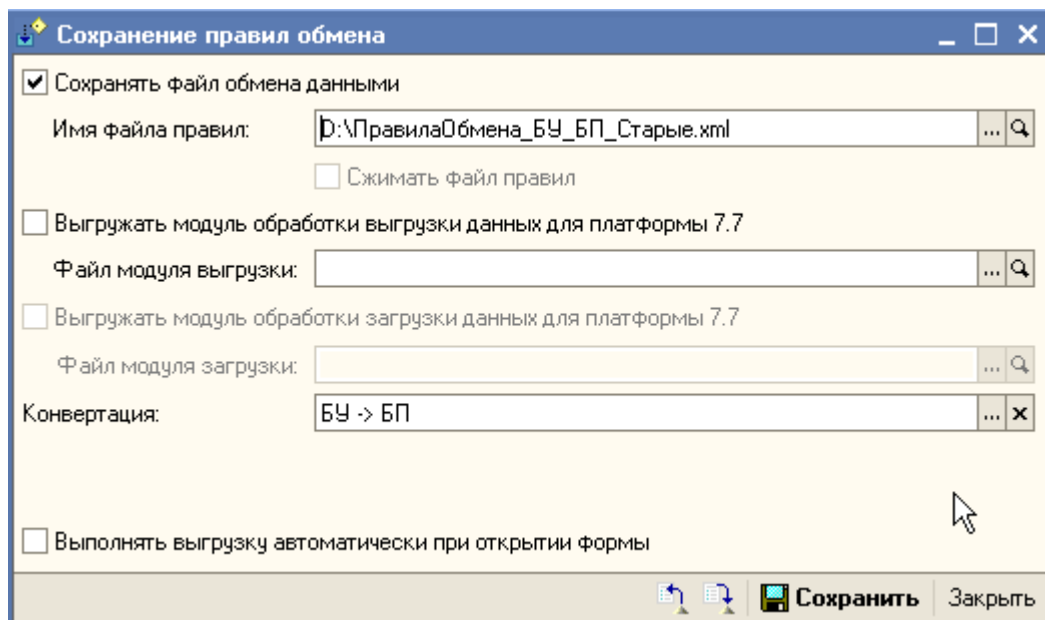


Рис. 8.

- Загружаем только что выгруженные правила

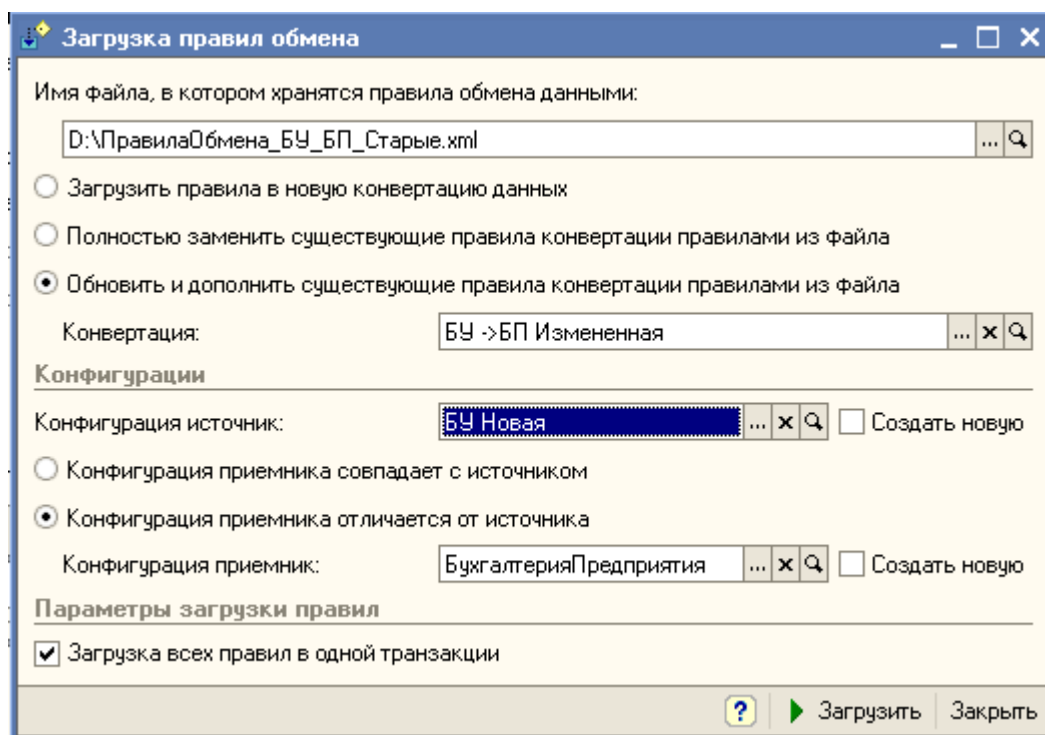


Рис. 9.

- После загрузки в списке и те ПКО, что были в старых правилах, и те, что в новых. Но те, которые из новых правил помечены на удаление, так как в загружаемых правилах этих ПКО не было.

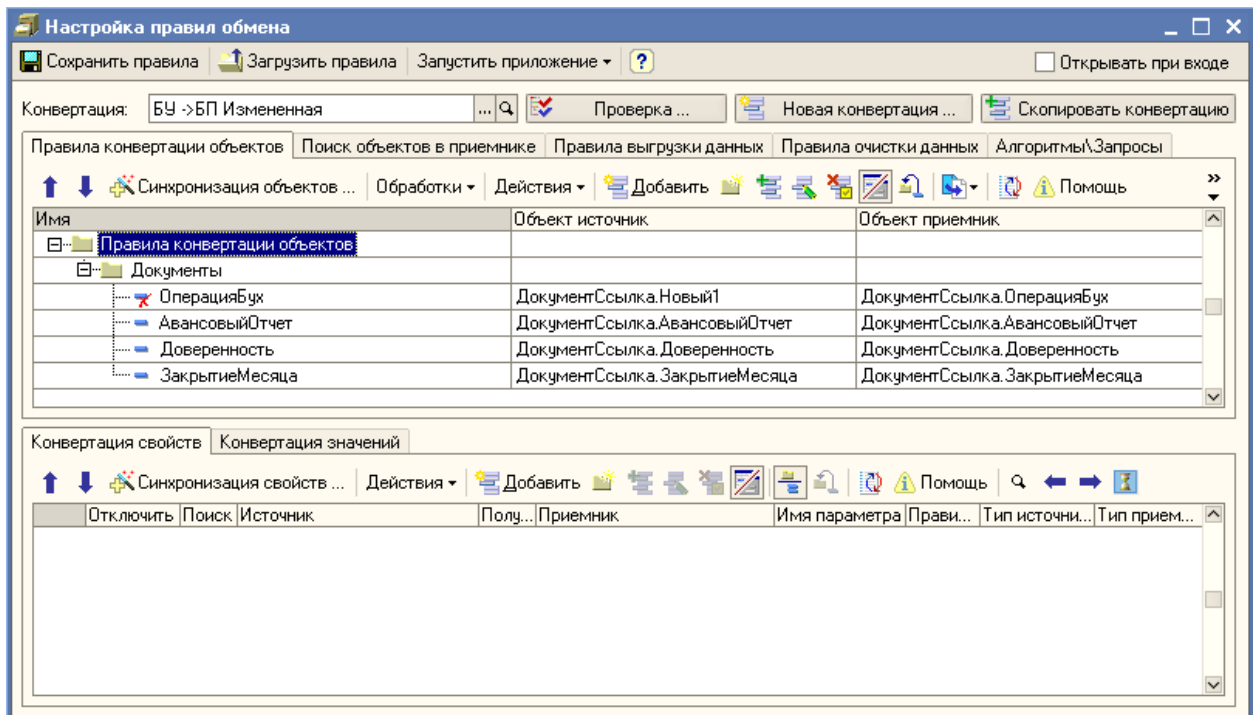


Рис. 10.

- Осталось только снять пометки удаления и новые правила готовы.

Настройка ПКО для перечислений

Принцип настройки правил простой: указываем, что выгружать, во что загружать, и как одно в другое преобразовать.

ПКО для перечислений наиболее простые из всех, они не ссылаются ни на какие другие ПКО, поэтому с них и начнем.

Создадим правило для Перечисления «Вид Склада».

- Создаем новое ПКО
- Указываем объект базы-источника перечисление Виде Складов
- Указываем объект базы-приемника перечисление Виде Складов
- Нажимаем «Готово». ПКО создано
- Теперь настраиваем соответствие значений. Переходим на закладку «Конвертация значений» Окна настройки правил обмена. Здесь создаем новое ПКЗ. Указываем Источник – «Оптовый», Приемник – «Оптовый».
- Нажимаем ОК – ПКЗ создано.
- Также создаем ПКЗ для значения «Розничный».

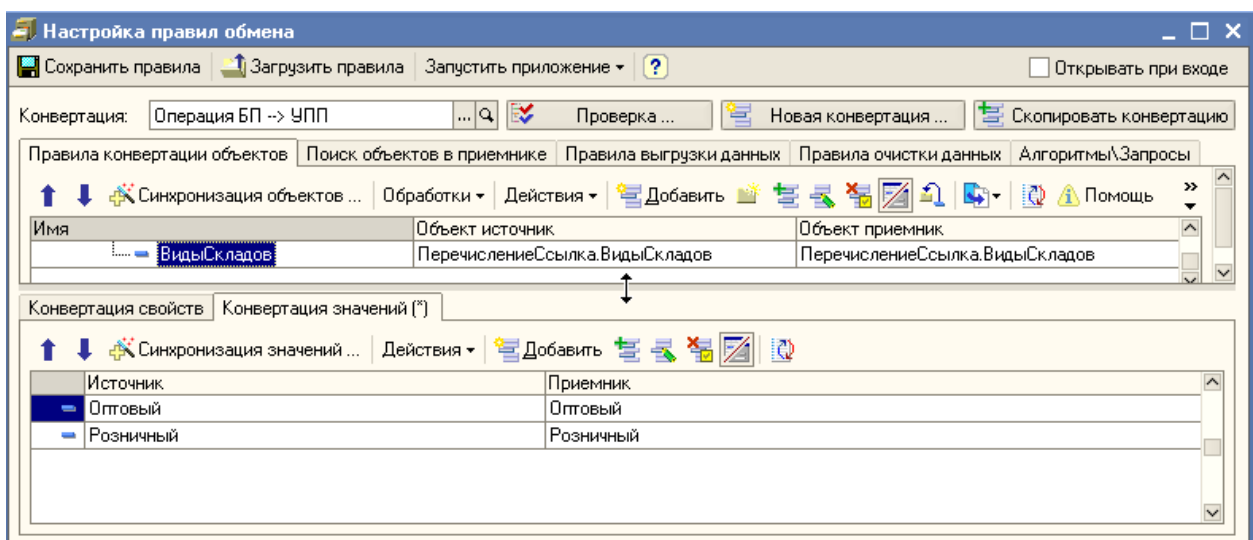


Рис. 11.

Настройка ПКО для справочников

Создаем правило для справочника Склады. Выгружаем элемент справочника Склады из Бухгалтерии. Загружаем элемент справочника Склады в УПП.

1. Создаем новое ПКО.
2. Указываем объект базы-источника – Склады
3. Указываем объект базы-приемника – Склады.
4. Нажимаем «Готово». ПКО создано.
5. Настраиваем соответствие реквизитов. Для этого переходим на закладку «Конвертация свойств» окна Настройка правил обмена. Создаем новое ПКС..
6. Указываем источник – свойство Наименование
7. приемник – свойство Наименование
8. Нажимаем «ОК» – ПКС создано. Поскольку Наименование имеет примитивный тип (Строка), то для него поле Правило оставляем пустым.

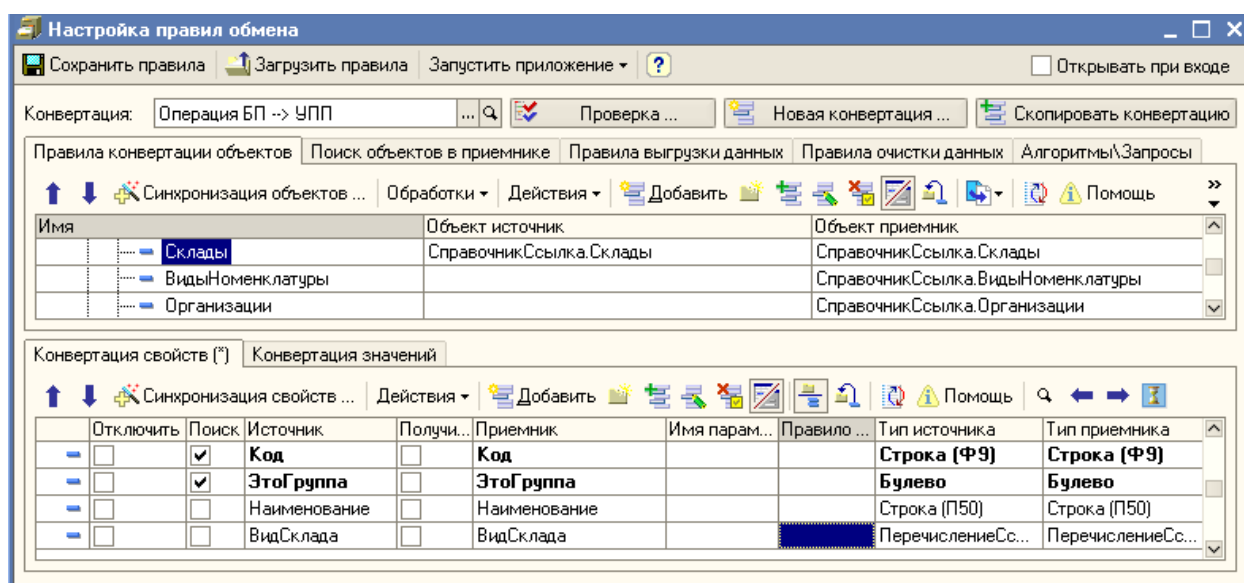


Рис. 12.

9. Точно также создаем ПКС Вид Склада -> Вид Склада. Реквизит Вид Склада имеет не примитивный тип (Перечисление), поэтому для него надо указать ПКО. Подходящее ПКО мы создали в предыдущем разделе, его и указываем в поле Правило.

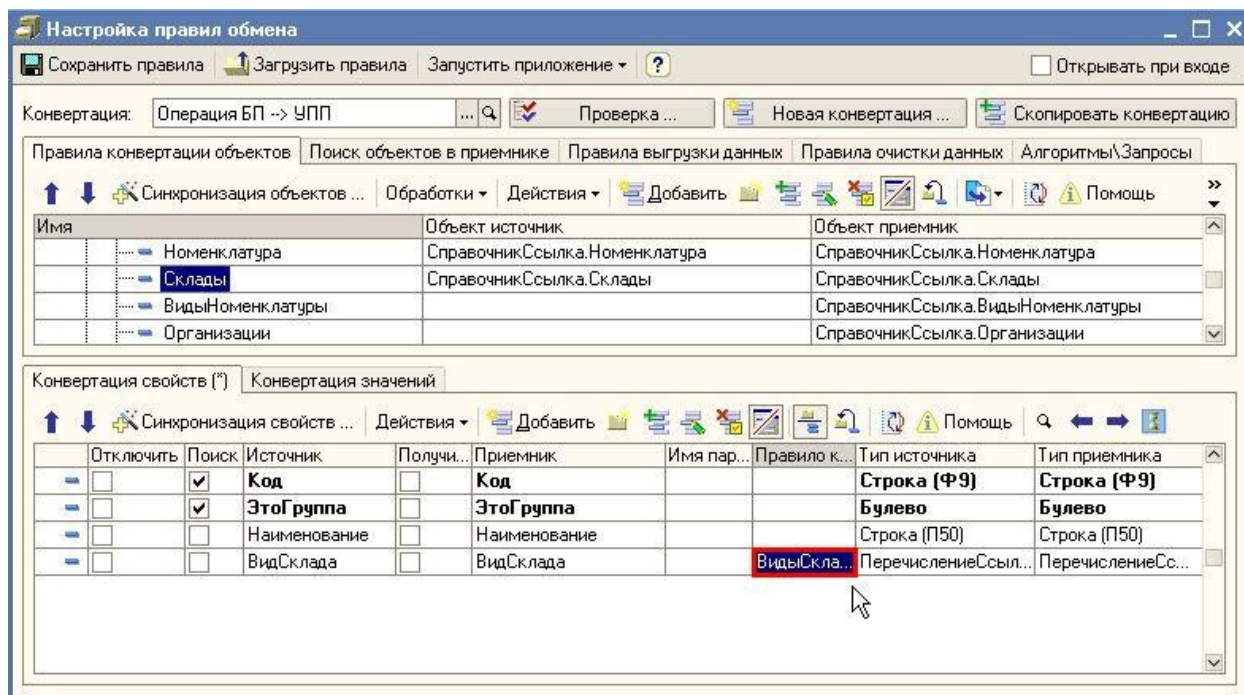


Рис. 13.

10. Точно также создаем ПКС для свойств Код и ЭтоГруппа
11. Указываем поля, по которым должен будет происходить поиск элементов.
 Подробнее об этом см. далее.
- Если в справочнике есть predetermined elements and tabular parts, then we configure rules for them
12. Для predetermined elements of the reference books we go to the tab «Конвертация значений» and we synchronize values also, as values of enumerations. Подробнее про predetermined elements см. далее.
13. Для табличных частей создаем новую группу на закладке «Конвертация свойств», указываем табличную часть-источник и табличную часть-приемник, далее открываем созданную группу и настраиваем реквизиты табличной части как обычные ПКС.
14. При настройке ПКС бывают ситуации, когда значение можно сконвертировать непосредственно в обработчиках событий ПКС. Как правило, ПКО необходимо создавать, если свойство-приемник имеет ссылочный тип. Если свойство-приемник имеет тип Перечисление, то ПКО можно создать для простого сопоставления значений источника и приемника.

Дополнительные настройки ПКО

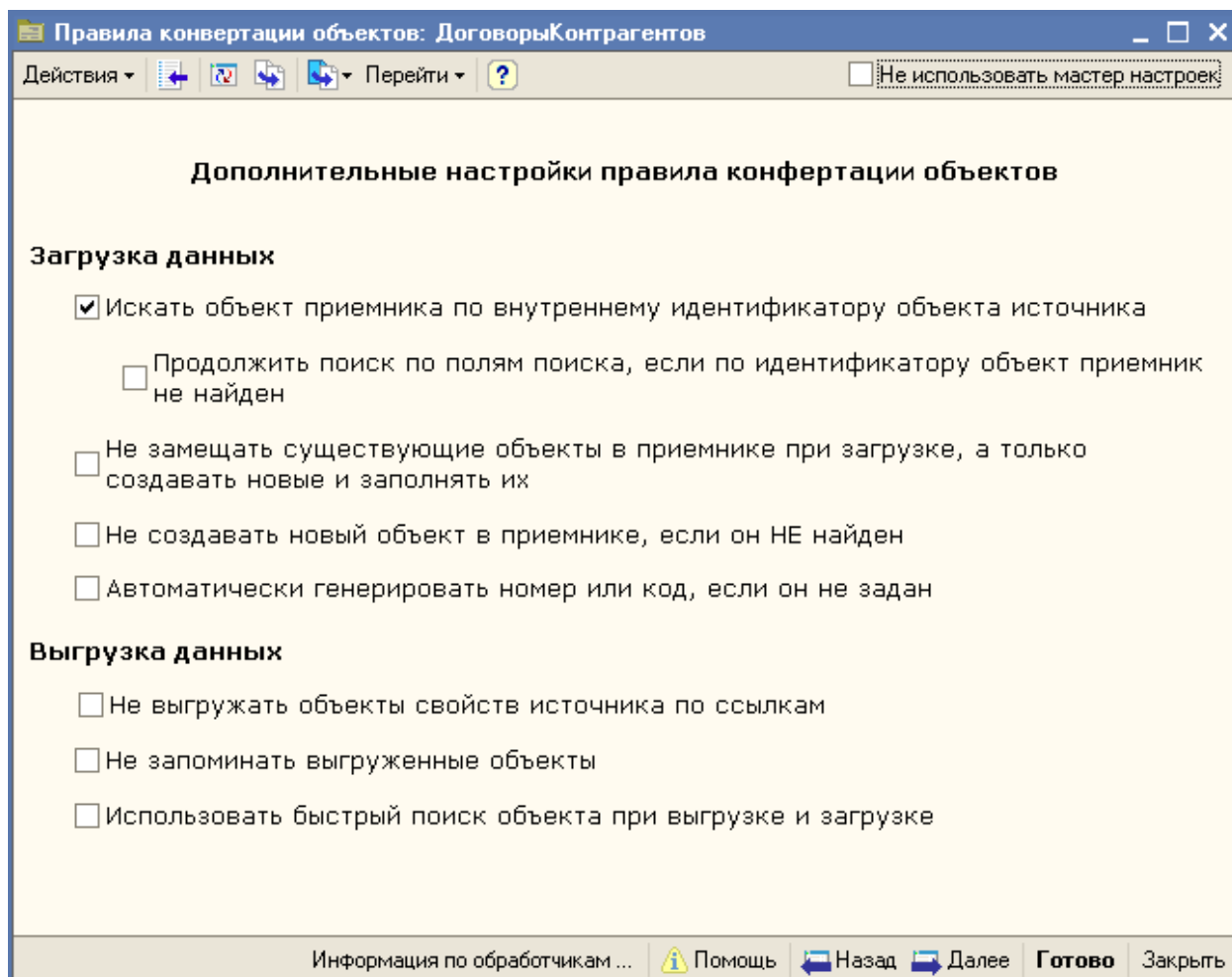


Рис. 14.

- **Искать объект приемника по внутреннему идентификатору объекта источника**

Уникальный идентификатор – это ссылка на объект в базе – источник, записанная в текстовую строку. Если флаг установлен, то в базе – приемнике загруженный объект будет иметь тот же уникальный идентификатор, что и в источнике. В файле данных уникальный идентификатор будет записан среди полей поиска, и поиск в первую очередь будет производиться по нему.

- **Продолжить поиск по полям поиска, если по идентификатору объект приемник не найден**

При загрузке в базу-приемнике если объект не найден по уникальному идентификатору, то будет произведен поиск по полям, отмеченным флажками поиска в ПКО, либо по полям, записанным в переменную СтрокаИменСвойствПоиска в обработчике событий Поля Поиска.

- **Не замещать существующие объекты в приемнике при загрузке, а только создавать новые и заполнять их**

Если загружаемый объект был в приемнике найден, то все ссылки на загружаемый объект будут перенастроены на найденный объект, а сам найденный объект останется таким же, как был до загрузки. Если же флаг не стоит, то свойства найденного объекта заменяются на свойства загружаемого объекта.

- **Не создавать новый объект в приемнике, если он НЕ найден**

Это значит, что данное ПКО написано только для объектов, которые уже загружены в базу-приемник. Новых создано не будет, и, если в выгруженных данных есть ссылки на объекты, которых нет в базе-приемнике, то эти ссылки останутся пустыми.

- **Автоматически генерировать номер или код, если он не задан**

Для тех объектов метаданных, у которых есть predetermined реквизит Код или Номер. Если Код или Номер загруженного объекта остается пустым, то будет присвоен новый Код или номер

- **Не выгружать объекты свойств по ссылкам**

Те объекты, на которые ссылаются свойства данного объекта, не будут выгружаться как объекты. Будет только произведен поиск данных объектов в базе-приемнике.

- **Не запоминать выгруженные объекты**

Если флаг не установлен, то выгруженные объекты сохраняются в специальной структуре выгруженных объектов, где ключом является переменная Ключ Выгружаемых Данных. Если данный объект попадает еще раз, например, по ссылке, то он сам не выгружается повторно, а находится по ключу и в ссылку записывается только номер, присвоенный объекту при выгрузке. Это позволяет сократить время выгрузки и объем файла данных. Флаг следует устанавливать в случаях, когда один объект в базе-источнике должен быть выгружен в столько различных объектов в базе-приемнике, сколько встречается ссылок на него.

- **Использовать быстрый поиск объекта при выгрузке и загрузке.**

При выгрузке этот флаг включает оптимизированный алгоритм выгрузки, но для этого надо также установить флаг «Использовать оптимизированный формат для обмена данными» в форме обработки обмена данными.

При загрузке не используется.

Правила выгрузки данных: стандартный и произвольный алгоритм

Правила выгрузки данных (ПВД) указывают, 1) какие объекты выгружать, 2) каким образом выбирать эти объекты 3) какое ПКО применить для конвертации этих объектов. Если не создано ни одного ПВД, то выгрузка производится не будет.

Когда в окне обработки выгрузки пользователь нажимает кнопку «Выгрузить», программа получает таблицу ПВД, выбирает первое правило из этой таблицы. В соответствии с указанным объектом выборки и алгоритмом, программа получает выборку объектов указанного типа, получает первый объект из этой выборки и передает управление в ПКО, которое также указано в ПВД.

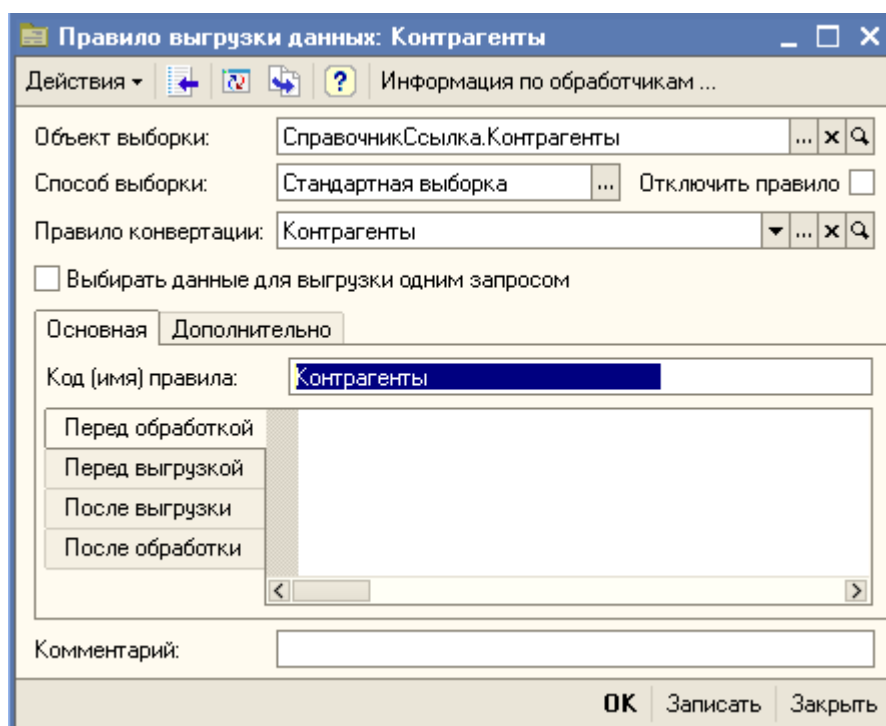


Рис. 15.

Если для объектов какого-либо типа не создано ПВД, то они могут быть выгружены только по ссылкам из других выгружаемых объектов.

В ПВД может быть выбрано два способа выборки: Стандартная выборка и Произвольный алгоритм.

Если указан алгоритм выборки **Стандартная выборка**, обработка выгрузки выберет все объекты типа, указанного в графе объект выборки. Повлиять на это можно только с помощью отборов, устанавливаемых в форме обработки перед выгрузкой из базы-источника.

Если же необходим более сложный алгоритм выборки, то используется **Произвольный алгоритм**. При этом лучше всего использовать переменную ВыборкаДанных. ВыборкаДанных – это запрос, результат запроса, выборка из результата запроса или любая произвольная коллекция выгружаемых объектов, для которой доступен обход посредством оператора Для Каждого Из ... Цикл. Например, это может быть таблица значений. В этом случае в обработчике события «Перед обработкой» Правила выгрузки данных надо написать код, заполняющий эту переменную коллекцией объектов. При заполнении надо руководствоваться тем ПКО, которое будет использовано для конвертации этих объектов. Для каждого объекта коллекции ВыборкаДанных должно быть можно через точку получить каждое из свойств, прописанных в ПКО. Больше ничего делать не надо, автоматически будет выполнен обход этой выборки, будут исполняться все написанные обработчики событий, и будет выполнена выгрузка по ПКО, указанному в ПВД.

Для случаев, когда выгружаемые данные невозможно получить одним запросом, в переменную ВыборкаДанных можно поместить таблицу значений. Заполняется она в соответствии с ПКО, которое будем использовать для конвертации объектов. В данном случае объектами будут строки этой таблицы. И во всех обработчиках событий данного ПКО и подчиненных ПКС в переменной «Источник» будет строка таблицы значений. В таблице ВыборкаДанных должны быть колонки с тем же именем, что и свойство-приемник в каждом из ПКС, входящих в данное ПКО. Если в ПКО есть группа свойств, например, табличная часть или набор движений регистра, то в таблице ВыборкаДанных должна быть колонка, названная так же, как таблица-приемник. В каждой строке этой колонки должна быть таблица, у которой столбцы наименованы так же, как свойства-приемники в ПКС, входящих в эту группу ПКС. **(См. пример про конвертацию таблицы значений в документ)**

Никаких дополнительных действий выполнять не надо, если переменная ВыборкаДанных заполнена правильно, то дальнейшая выгрузка будет выполнена автоматически.

Автоматическая настройка правил

Для автоматической настройки правил в конфигурации «Конвертация данных» существует ряд обработок.

Прежде всего, при создании новой конвертации программа предлагает пользователю выбор:

«Автоматически создать все правила обмена»

«Создать новое правило обмена данными»

Если выбрать «Автоматически создать все правила обмена», то программа автоматически сравнит конфигурацию-источник и конфигурацию-приемник и создаст ПКО для всех объектов с одинаковым названием, для этих ПКО создаст ПКС для всех свойств с одинаковым названием и ПКЗ для всех значений с одинаковым названием. При этом несовпадающие объекты, свойства и значения программа проигнорирует, их придется настраивать вручную. Программа автоматически определит поля поиска. Останется только проверить, насколько они подходят для решаемой задачи.

В конце настройки программа выдает список созданных ПКО с предложением создать для каждого свое правило выгрузки. В этом списке можно выбрать, какие правила выгрузки необходимо создать.

После этого необходимо будет пройтись по правилам и проверить корректность настройки:

Наличие необходимых для заполнения свойств, для которых не было создано ПКС. Созданные ПКС, в которых свойство-источник и свойство-приемник имеют разные не примитивные типы данных. Такие строки на закладке Конвертация свойств отмечены желтым цветом. Поскольку типы данных разные, то ПКО будет не заполнено. Чтобы данное ПКС работало, необходимо вручную создать ПКО для него.

Проверить поля поиска для ПКО. По умолчанию документы синхронизируются по Номеру и Дате, а справочники по свойствам Код, Владелец и Это Группа. Надо подумать, для каких объектов такие поля поиска не подходят для решаемой задаче.

Проверить настройки ПКО. По умолчанию в ПКО стоит флажок «Искать объект приемника по внутреннему идентификатору объекта – источника», а остальные флажки сброшены.

Никакие обработчики событий написаны не будут, поэтому если необходимо, надо их написать.

Есть также способы полуавтоматической настройки правил, где пользователь все же может контролировать создаваемые ПКО и ПКС. Для этого в окне настройки правил обмена есть кнопки Синхронизация объектов, Синхронизация свойств, Синхронизация значений, Создание правил выгрузки.

Кнопка «Синхронизация объектов» находится вверху окна настройки правил обмена.

Она вызывает окно выбора создаваемых ПКО

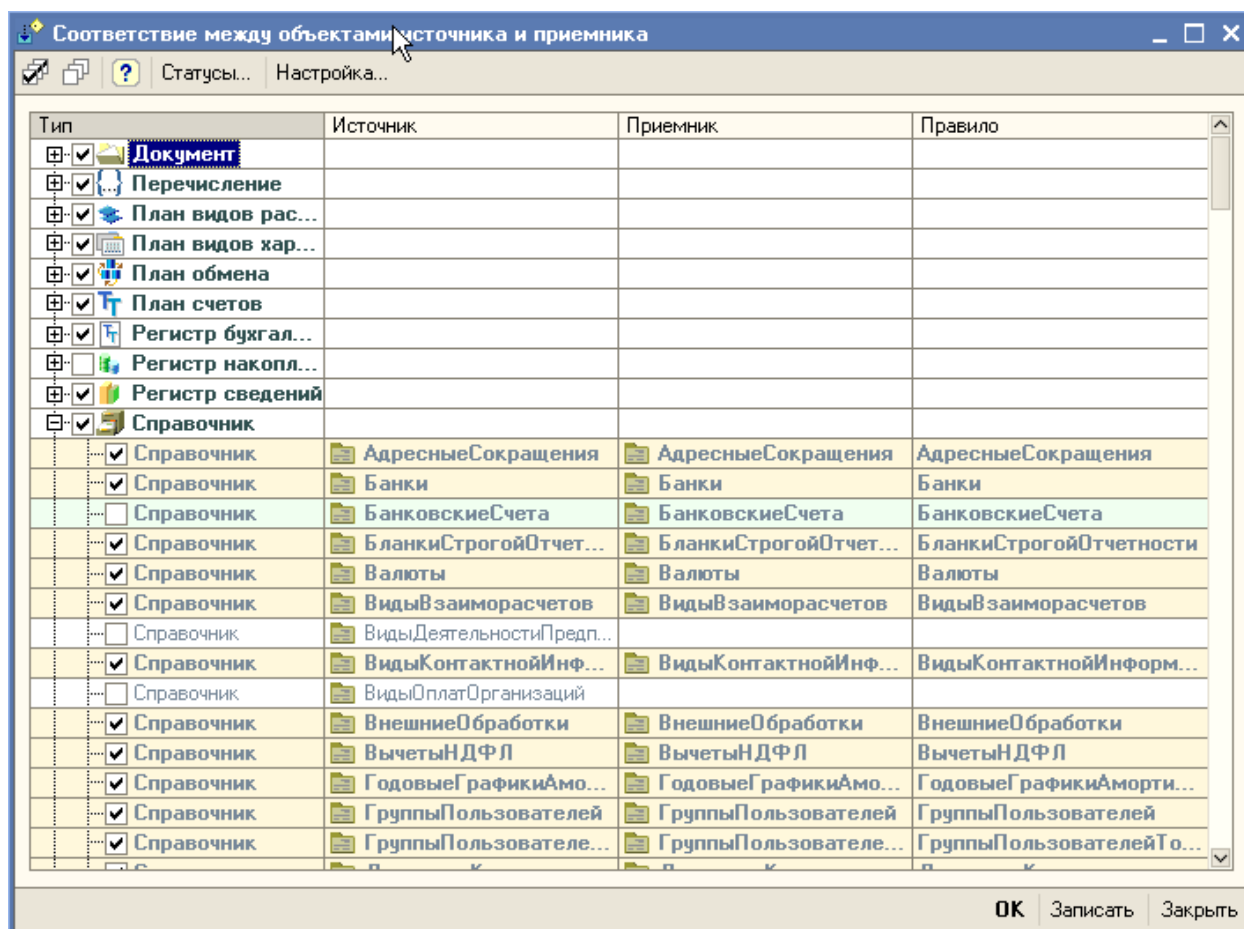


Рис. 16.

Здесь надо выбрать нужные ПКО и нажать ОК. Далее программа предложит для создаваемых ПКО автоматически создать ПКС и ПКЗ, а также правила выгрузки данных. При создании ПКС, программа автоматически создаст ПКО для каждого свойства. Правила выгрузки она создаст не сразу, а позволит выбрать необходимые.

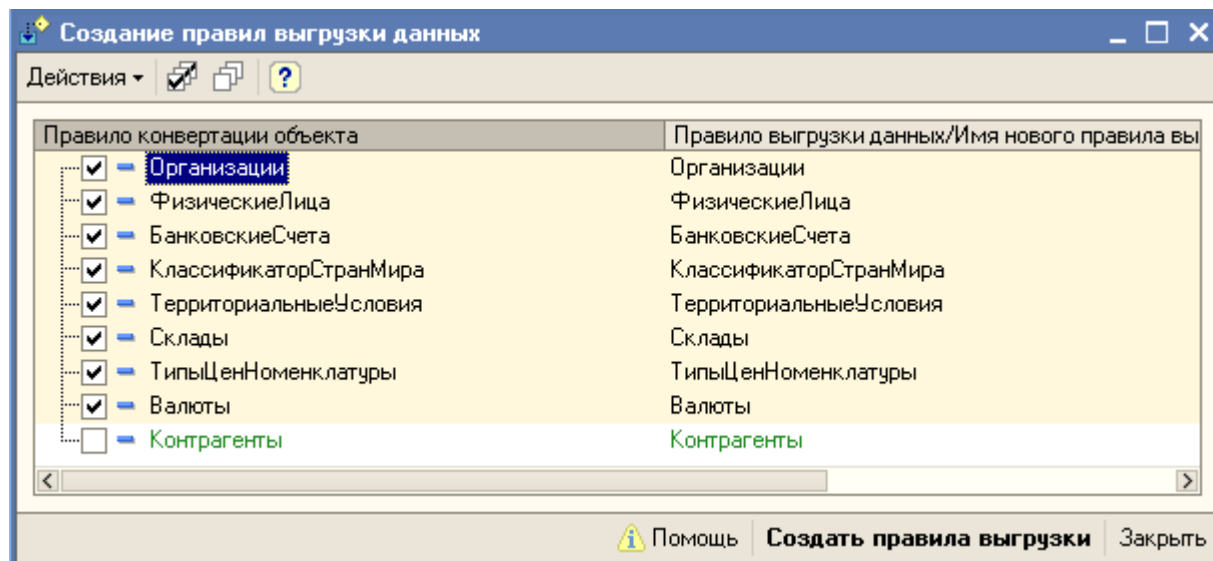


Рис. 17.

В случае автоматической синхронизации свойств при создании ПКО, программа синхронизирует все свойства с одинаковым названием, и если для решения задачи нужны не все свойства объекта, то лишние придется отключать, а ПКО для них не будут использоваться, и файл правил получится больше.

Чтобы этого не происходило, можно создать пустые ПКО и для каждого их них создать ПКС в полуавтоматическом режиме. Для этого нажимаем на кнопку «Синхронизация свойств». На закладке «Конвертация свойств». Выбираем свойства, которые нам нужны, отмечаем их флажками, с остальных флажки снимаем. Вверху окна есть флажок «Рекурсивно создавать правила конвертации свойств и значений подчиненных объектов», если он установлен, для всех свойств будут созданы ПКО, у них синхронизированы все свойства и для них тоже созданы ПКО. Уменьшить количество создаваемых объектов, если установить флажок «Создавать правила конвертации свойств только для полей поиска у подчиненных объектов».

Кнопка «Синхронизация свойств» на закладке «Конвертация значений» производит аналогичную синхронизацию значений.

На закладке «Правила выгрузки данных» кнопка «Создание правил выгрузки» выводит список созданных пко с предложением создать для них ПВД.

Выбор полей поиска и поиск элементов в базе-приемнике

Поиск объектов в базе-приемнике происходит для того, чтобы при загрузке каждого объекта программа проверяла, нет ли уже этого объекта в базе-приемнике, и, если есть, не записывала бы его повторно.

Поиск может производиться:

- По уникальному идентификатору
- По полям поиска
- Произвольным образом

Поиск по идентификатору означает, что, записывая объект в базу-приемник, программа присваивает ему уникальный номер ссылки, которую этот объект имеет в базе-источнике. Это очень удобно, поскольку пользователь может изменить у загруженных объектов любые свойства и реквизиты, а объекты не дублируются при следующей загрузке. Этот механизм не подходит для некоторых задач, например там, где из нескольких разных баз данные перегружаются в одну, во всех базах есть одинаковые

элементы справочников, которые необходимо синхронизировать. Для ситуации, когда данные загружаются в базу, куда уже введены данные, и необходимо загружаемые данные синхронизировать с уже введенными, можно использовать настройку «Продолжить поиск по полям поиска, если по идентификатору объект приемник не найден».

Для синхронизации по полям поиска ставим флажки Поиск у ПКС для тех свойств, по которым будем проводить синхронизацию элементов. Например, это Код и ЭтоГруппа. Это означает, что если при загрузке какого-то склада в базе-приемнике будет найден склад с такими же кодом и реквизитом ЭтоГруппа, то программа сочтет, что это тот же элемент, и настроит на него все ссылки.

В последних версиях Конвертации данных добавлена возможность поиска элемента произвольным образом. Для этого в ПКО добавлен обработчик события Поля Поиска.

В нем можно определить, по каким полям из отмеченных для поиска надо искать объект, или написать любой код, в том числе и любые запросы, для поиска объекта. **(См. задачу про Штрих Коды)**

Более подробно см. во встроенной в Конвертацию справке «Информация по обработчикам».

Настройка ПКО для планов видов характеристик, планов счетов и справочников с предопределенными элементами.

Допустим, необходимо перенести справочник, где кроме обычных элементов есть предопределенные.

- 1) Если в базе-приемнике нет также предопределенных элементов, которые можно было бы им сопоставить, то правила настраиваются обычным образом – указываются ПКС для нужных реквизитов, по каким реквизитам вести поиск, и после загрузки увидим, что в базе-приемнике созданы элементы, аналогичные тем, которые в базе-источнике были предопределенными.
- 2) Другой случай, если мы переносим, например, ПВХ ВидыСубконто или справочник, в котором аналогичные предопределенные элементы есть и в базе-источнике и в базе-приемнике. Допустим, коды у них не всегда совпадают, и поменять мы их не можем – они же предопределенные. Если есть хоть какой-то реквизит, по которому можно провести синхронизацию – наименование, например, - то опять же достаточно настроить ПКС для всех свойств и указать, по какому производить поиск. В этом случае надо снять галка «Искать по внутреннему идентификатору» в ПКО, так как у предопределенных элементов в двух разных базах отличаются. (Исключение составляю случаи, когда две базы являются потомками одной базы, где эти предопределенные элементы уже были.)
- 3) И третий случай, когда соответствие предопределенных элементов в источнике и приемнике необходимо настроить явно. Тогда настраиваем Правила конвертации значений (ПКЗ) для этих элементов. Одновременно с этим настраиваем ПКС для всех остальных элементов, предопределенных и введенных в пользовательском режиме.

Для настройки ПКО для планов видов характеристик и планов счетов удобно использовать именно третий способ. Предопределенные элементы можно сопоставить явно, настроив ПКЗ, а остальные синхронизировать по Наименованию.

Настройка ПКО для регистров сведений.

Регистры сведений – единственный вид регистров, который может иметь записи, не подчиненные регистратору. Характерный пример – курсы валют.

Для настройки простейшего переноса из восьмерки в восьмерку

1. Создаем новое ПКО.
2. Указываем объект базы-источника – регистр сведений Курсы валют

3. Указываем объект базы-приемника – регистр сведений Курсы валют.
4. Настраиваем соответствие измерений, ресурсов и реквизитов. Для этого переходим на закладку «Конвертация свойств» окна Настройка правил обмена. Создаем новое ПКС..
5. Указываем источник – свойство Наименование
6. приемник – свойство Наименование
7. Нажимаем «ОК» – ПКС создано. Поскольку Наименование имеет примитивный тип (Строка), то для него поле Правило оставляем пустым.
8. Синхронизируем все остальные свойства. Для измерения Валюта необходимо создать ПКО для конвертации справочника Валюты.

Флаги поиска для регистров сведений устанавливать не надо, также как не надо и ставить флаги поиска по внутреннему идентификатору, так как синхронизация записей регистров сведений происходит только по значениям измерений и периоду, если регистр сведений периодический. **(См. пример про перенос курсов валют 8 – 8)**

Использование параметров конвертации

Параметры конвертации определяются на закладке «Параметры» в окне конвертации. Структура Параметры является глобальной по всему модулю обработки обмена данными. Все параметры, указанные на закладке Параметры, добавляются в эту структуру как ее элементы. Для каждого параметра можно указать следующие настройки:

- **При Загрузке** – При загрузке будет создана структура с указанными параметрами. Никаких значений туда передано не будет.
- **Диалог** – возможность установить значение параметра перед выгрузкой данных в форме обработки выгрузки.
- **Тип Значения** – тип значения параметра
- **Передавать При Выгрузке** – При загрузке в структуру Параметры будет добавлен данный параметр вместе со значением, записанным в него.
- **Правило конвертации** – ПКО. Как его использовать, непонятно, вероятно, данная возможность еще не реализована.

Алгоритмы и Запросы

Алгоритмы и запросы настраиваются на закладке «Алгоритмы/Запросы» окна Настройки правил обмена.

Запросы используются только в базах данных на восьмерке. В таблице «Запросы» создается новый запрос и в текстовом поле пишется текст запроса

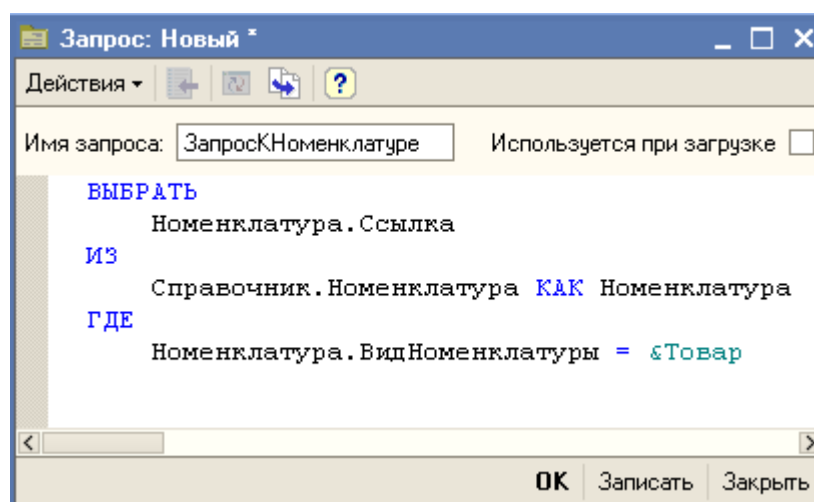


Рис. 18.

При выгрузке и загрузке в этом случае в структуре Запросы будут храниться объекты типа Запрос. Структура Запросы является глобальной переменной, доступ к ней существует в любом обработчике события. Чтобы получить результат запроса необходимо использовать следующую конструкцию:

```
Запросы.ЗапросКНоменклатуре.УстановитьПараметр («Товар», Справочники.ВидыНоменклатуры.НайтиПоНаименованию («Товар»);  
РезультатЗапроса = Запросы.ЗапросКНоменклатуре.Выполнить ();
```

Алгоритмы, в отличие от запросов, доступны и в семерке. Для создания нового алгоритма в окне Алгоритмов надо создать новый элемент и в текстовом поле, так же, как и для запроса, написать текст алгоритма. В восьмерке будет создана структура Алгоритмы, доступная в любом событии. Чтобы выполнить алгоритм, в любом обработчике необходимо написать

```
Выполнить (Алгоритмы.<Имя_алгоритма>);
```

В семерке алгоритмы представляют собой функции, которые создаются при автоматическом создании текста модуля обработки выгрузки или загрузки. Эти функции можно вызвать в любом обработчике события. В функцию автоматически передается целый ряд параметров:

```
Функция <ИмяАлгоритма>( ИсходящиеДанные = "", ВходящиеДанные = "",  
Источник = "", Приемник = "", ТипПриемника = "",  
Значение = "", Выражение = "", Пусто = "",  
ВыборкаДанных = "", Объект = "",  
КоллекцияОбъектов = "", ОбъектКоллекции = "",  
ИмяПКО = "", ИмяПКОВидСубконто = "", НомерПКО = "",  
НомерПКС = "", Правило = "",  
КлючВыгружаемыхДанных = "", НеЗапоминатьВыгруженные = "",  
НеЗамещатьОбъектПриЗагрузке = "",  
ВсеОбъектыВыгружены = "",  
ТолькоПолучитьУзелСсылки = "", РежимЗаписи = "",  
РежимПроведения = "", СтандартнаяОбработка = "",  
НеЗамещать = "", НеОчищать = "", УзелСсылки = "",  
УзелКоллекцииОбъектов = "", УзелОбъектаКоллекции = "",  
УзелСвойства = "", ВидСубконто = "", Субконто = "",  
Отказ = "")
```

Правила очистки данных

Правила очистки данных выполняются только в базах на основе восьмерки. Они нужны, если в базе данных необходимо удалить ряд каких-то объектов.

Настройка их похожа на настройку правил выгрузки данных.

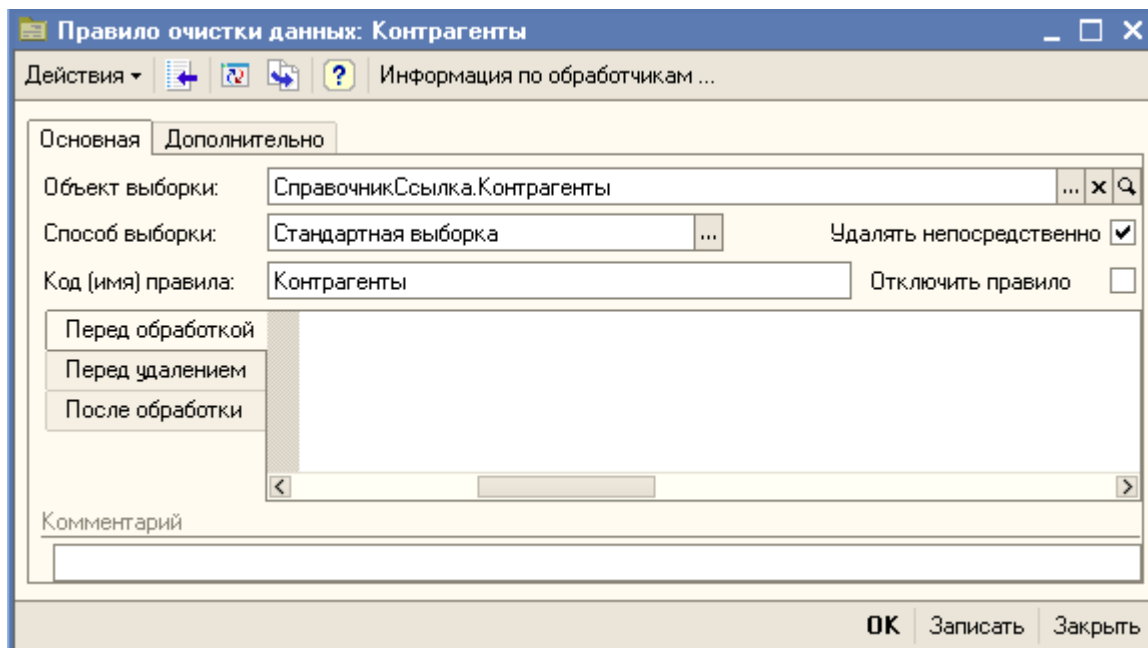


Рис. 19.

Для удаления объектов типа, указанного в поле Объект выборки необходимо указать алгоритм выборки, если алгоритм произвольный, написать алгоритм для выборки удаляемых объектов в обработчике события «Перед обработкой».

В правилах очистки данных можно указать режим удаления – удалять непосредственно или просто помечать на удаление.

Ключ Выгружаемых Данных

Ключ выгружаемых данных – это переменная, которая используется при выгрузке. Если в ПКО выключен флажок «Не Запоминать Выгруженные Объекты», то при выгрузке каждый объект будет сохранен в структуре выгруженных объектов. Ключ выгружаемых данных – это ключ, с которым данный объект будет там сохранен, и по которому будет вестись поиск этого объекта. Это нужно, чтобы если вдруг уже выгруженный объект встречается повторно по другой ссылке, его не надо было бы выгружать целиком, а достаточно было бы получить его номер из структуры уже выгруженных объектов. Это сокращает время выгрузки и объем файла данных. По умолчанию, Ключ Выгружаемых Данных – это внутренний идентификатор объекта-источника, записанный в текстовую строку. Если источника у правила нет, то Ключ выгружаемых данных - это Имя ПКО (в этом случае объекты запоминаться не будут, независимо от флажка в ПКО).

Явно задавать этот ключ надо, например,

- 1) если один объект должен в зависимости от каких-то условий выгружаться как несколько разных объектов
- 2) если у ПКО нет источника, а заполняется оно из входящих данных, но при этом для сокращения времени выгрузки и объема файла данных необходимо, чтобы объекты запоминались.

В Ключ выгружаемых данных в событии Перед Выгрузкой должна быть записана такая текстовая строка, которая уникальна для каждого объекта-приемника.

(См. задачу про оптимизацию выгрузки справочника Организации)

Функция ВыгрузитьПоПравилу()

Функция ВыгрузитьПоПравилу() используется для выгрузки объекта в xml-узел и возвращает узел ссылки на этот объект. В процессе выгрузки эта функция вызывается из ПВД, когда объект из выборки получен и необходимо произвести его конвертацию, или из ПКС, когда свойство имеет ссылочный тип и необходимо выгрузить по ссылке из этого свойства объект.

Эту функцию имеет смысл вызывать при выгрузке, если в каком-то месте кода есть ссылка на объект и необходимо передать управление в правило конвертации объекта, где этот объект будет конвертирован и выгружен в файл данных. Из функции `ВыгрузитьПоПравилу()` могут быть вызваны другие функции, выгружающие субконто, свойства и т.д. этого объекта, а из этих функций – рекурсивно может быть вызвана функция `ВыгрузитьПоПравилу()` (**см. схему процесса выгрузки**). После того, как объект со всеми необходимыми свойствами выгружен, управление возвращается в то место кода, откуда функция была вызвана. Таким образом, вызов этой функции из обработчиков событий какого-либо ПКО, не изменяет процесса выгрузки объекта по этому ПКО.

Функция `ВыгрузитьПоПравилу()` имеет ряд параметров

`ВыгрузитьПоПравилу(Источник,
Приемник,
ВходящиеДанные,
ИсходящиеДанные,
ИмяПКО,
УзелСсылки, ТолькоПолучитьУзелСсылки,
ПКО,
ЭтоПравилоСГлобальнойВыгрузкойОбъектов,
ВыборкаДляВыгрузкиДанных)`

Все эти параметры в функции имеют значения по умолчанию, и, как правило, передавать имеет смысл только параметр `ИмяПКО`, а также параметры `Источник` и/или `ВходящиеДанные` – в зависимости от того, как настроено ПКО.

`ИмяПКО` – строка, название ПКО, в которое передаем управление.

`Источник` – ссылка на элемент объекта-источника, который передаем для конвертации.

`ВходящиеДанные` – структура. Элементы в ней должны быть названы аналогично ПКС, у которых стоит флаг «Получить из входящих данных». Если таких ПКС нет, передавать этот параметр не нужно.

Если в ПКО у всех ПКС стоит флаг «Получить из входящих данных», то, наоборот, необходимо передать структуру `ВходящиеДанные`, а `Источник` тогда передавать не нужно.

(см. разбор типичных примеров из демобазы **Два объекта в один, Один объект в два**)

Несколько технических моментов. На что обратить внимание

Режим «ОбменДанными.Загрузка»

В обработке «Универсальный обмен данными» есть флажок «Загружать данные в режиме обмена (`ОбменДанными.Загрузка` = истина)». По умолчанию это флажок установлен. В более ранних версиях обработки такого выбора не было и загрузка производилась только в режиме `ОбменДанными.Загрузка`. Если этот флажок установлен, то при загрузке каждого объекта устанавливается режим `ОбменДанными.Загрузка`, и при записи объекта отключаются все возможные контроли на уникальность кодов, наличие обязательных реквизитов и т.д. Если не настраивать ПКО и ПКС для обязательных реквизитов справочников (например, для Вида Номенклатуры при выгрузке номенклатуры), то в режиме `ОбменДанными.Загрузка` объекты будут записаны в базу без этих реквизитов. Целостность базы при этом нарушается и возможны различные ошибки.

Если флажок «Загружать данные в режиме обмена (`ОбменДанными.Загрузка` = истина)» снять, то в процессе загрузки будет появляться значительно больше ошибок, зато в дальнейшем меньше вероятности нарушения ссылочной целостности в загруженных данных. Проблема в том, что, например, при выгрузке номенклатуры прежде самой

номенклатуры выгружаются единицы измерения, а при попытке загрузки возникает ошибка, так как на момент записи единиц владелец этой единицы еще не загружен. Загрузка на этом прерывается. Такие кольцевые ссылки можно загрузить только в режиме ОбменДанными.Загрузка.

Пометка на удаление. Почему она установлена?

В комплект поставки конфигурации Конвертация данных 2.0.20.1 входят обработки выгрузки/загрузки для платформ 8.0 и 8.1. Обработка для 8.1 обладает некоторой особенностью. После загрузки в базе-приемнике все загруженные объекты справочников Номенклатура и Контрагенты могут быть помечены на удаление. Это происходит, если для этих справочников мы не создали ПКС ПометкаУдаления. В процессе загрузки элементы этих справочников помечаются на удаление, если они загружены по ссылке из поля, по которому ведется поиск. В данном случае Контрагенты и Номенклатура являются владельцами для справочников Договоры Контрагентов и Единицы Измерения, а у этих справочников по полю Владелец осуществляется поиск. Проблема решается с помощью создания ПКС Пометка удаления или явного снятия пометки удаления в обработчике события При Загрузке или После Загрузки.

Задачи по переносу данных и их решение

Пример: Виды Номенклатуры. Конвертация значения типа Булево в справочник.

В УПП в справочнике Номенклатура обязателен для заполнения реквизит «Вид Номенклатуры», но в конфигурации Бухгалтерия Предприятия у номенклатуры такого реквизита нет, а есть только реквизит «Услуга» типа Булево. Создаем ПКО Виды Номенклатуры, источник у него пустой. Создаем для него ПКС: нам нужно, чтобы там были заполнены реквизиты Наименование и Тип Номенклатуры, который имеет тип Перечисление. Если реквизит Услуга в Бухгалтерии установлен, необходимо, чтобы тип номенклатуры принимал значение Услуга, а если не установлен, то Товар.

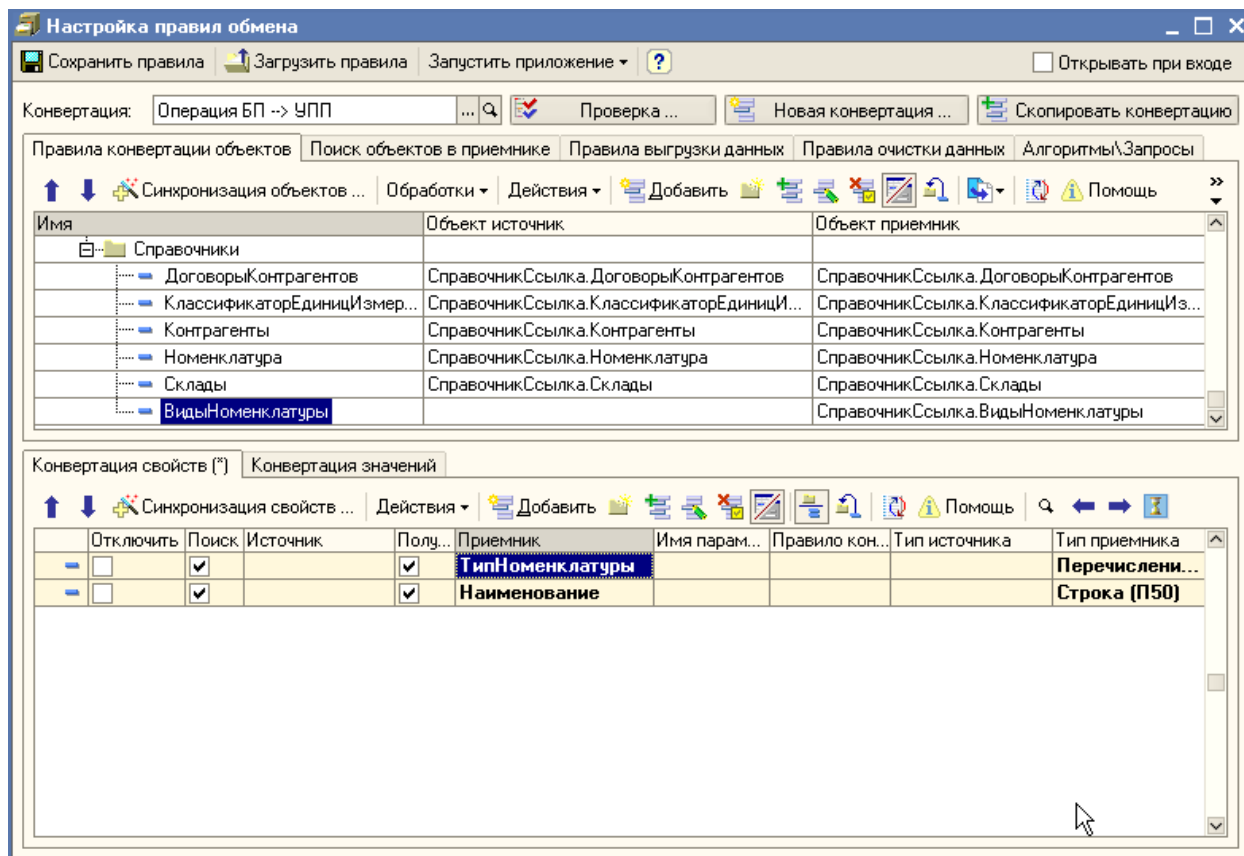


Рис. 20.

Создаем эти ПКС, в событиях ничего не пишем, а только у обоих ставим галки «Получить из входящих данных». Теперь надо, чтобы до того, как начнет выполняться ПКО «Виды Номенклатуры» во входящие данные попали эти значения. ВходящиеДанные и ИсходящиеДанные – переменные, в семерке – список значений, в восьмерке – как правило, структура. При передаче управления из ПКС в ПКО переменная ИсходящиеДанные передается как ВходящиеДанные.

Поэтому в ПКС «Виды номенклатуры» указываем наше ПКО виды номенклатуры и пишем обработчик.

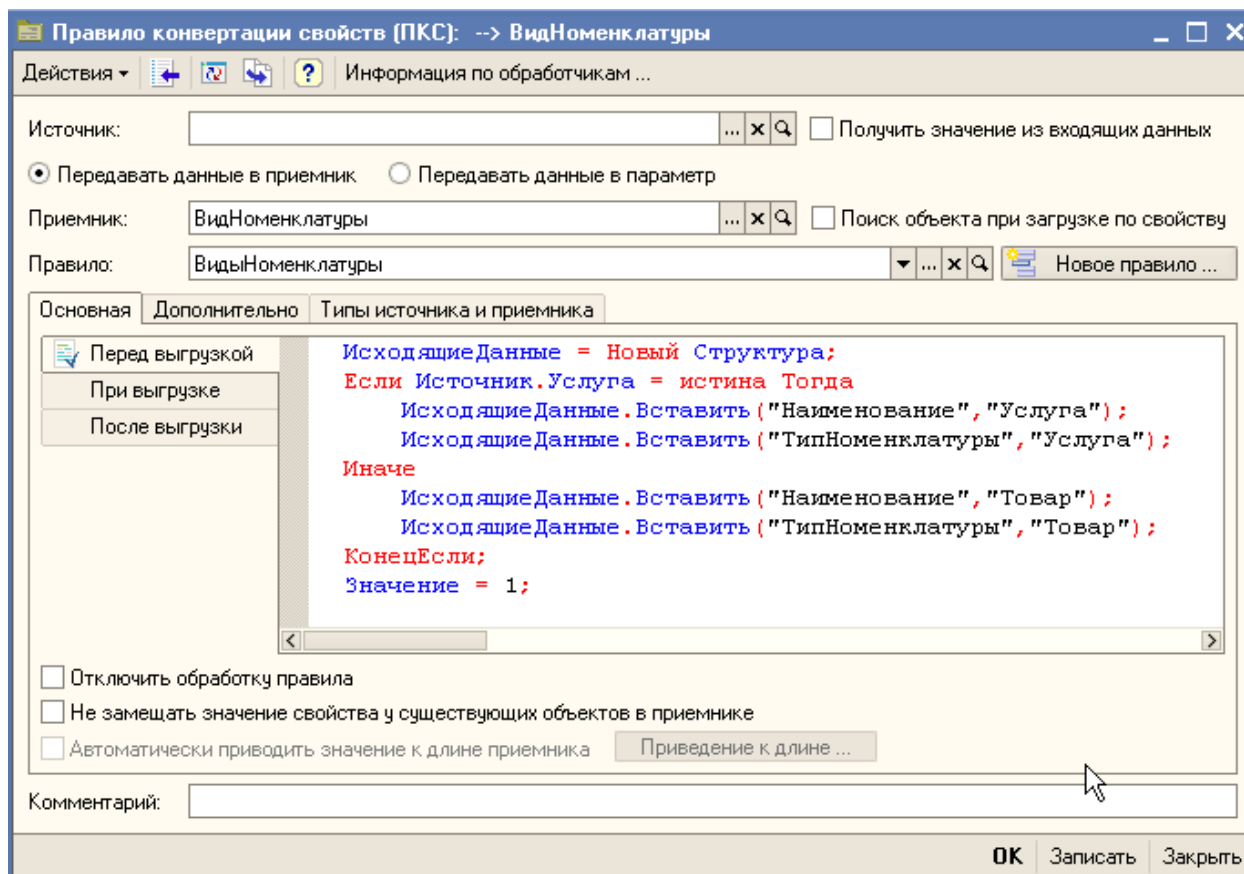


Рис. 21.

Зачем здесь строка «Значение = 1;»?

Просто если значение в ПКС будет пустым, то при выгрузке программа сочтет, что не нужно выполнять правило конвертации объекта, ведь конвертировать-то нечего. На практике нам эта переменная Значение не понадобится (так мы написали ПКО), можно записать туда любое значение, лишь бы она не была пустой.

Пример: Конвертация справочника Классификатор единиц измерения в справочник Единицы измерения. Один элемент в несколько

Следующий момент: в бухгалтерии у номенклатуры есть только базовая единица измерения типа Классификатор единиц измерения, а в УПП кроме базовой есть еще и Единица хранения остатков, и она тоже обязательна для заполнения и имеет тип Единицы измерения. Создаем ПКО: Источник – Классификатор единиц измерения, приемник – единицы измерения.

Здесь же ставим галку «Не запоминать выгруженные объекты» и снимаем галку «Искать по внутреннему идентификатору»

Подробность. Если не поставить галку «Не запоминать выгруженные объекты» или не снять галку «Искать по внутреннему идентификатору», один элемент Классификатора будет загружаться как один объект. Запоминать выгруженные – значит, что в процессе выгрузки сохраняются ссылки на выгруженные объекты, и если объект, который уже выгружался, выгружается по ссылке повторно, то в файл выгрузки записывается не новый объект, а ссылка на уже выгруженный. Поиск по идентификатору тоже подразумевает, что один объект из источника перегружается в один в приемнике, так как у объекта может быть только один идентификатор. А нам нужно, чтобы каждый раз по ссылке из справочника Номенклатура единица выгружалась как новый элемент, так как справочник единицы измерения подчинен справочнику номенклатура. Галку «Генерировать код» тоже ставим, чтобы не беспокоиться о кодах.

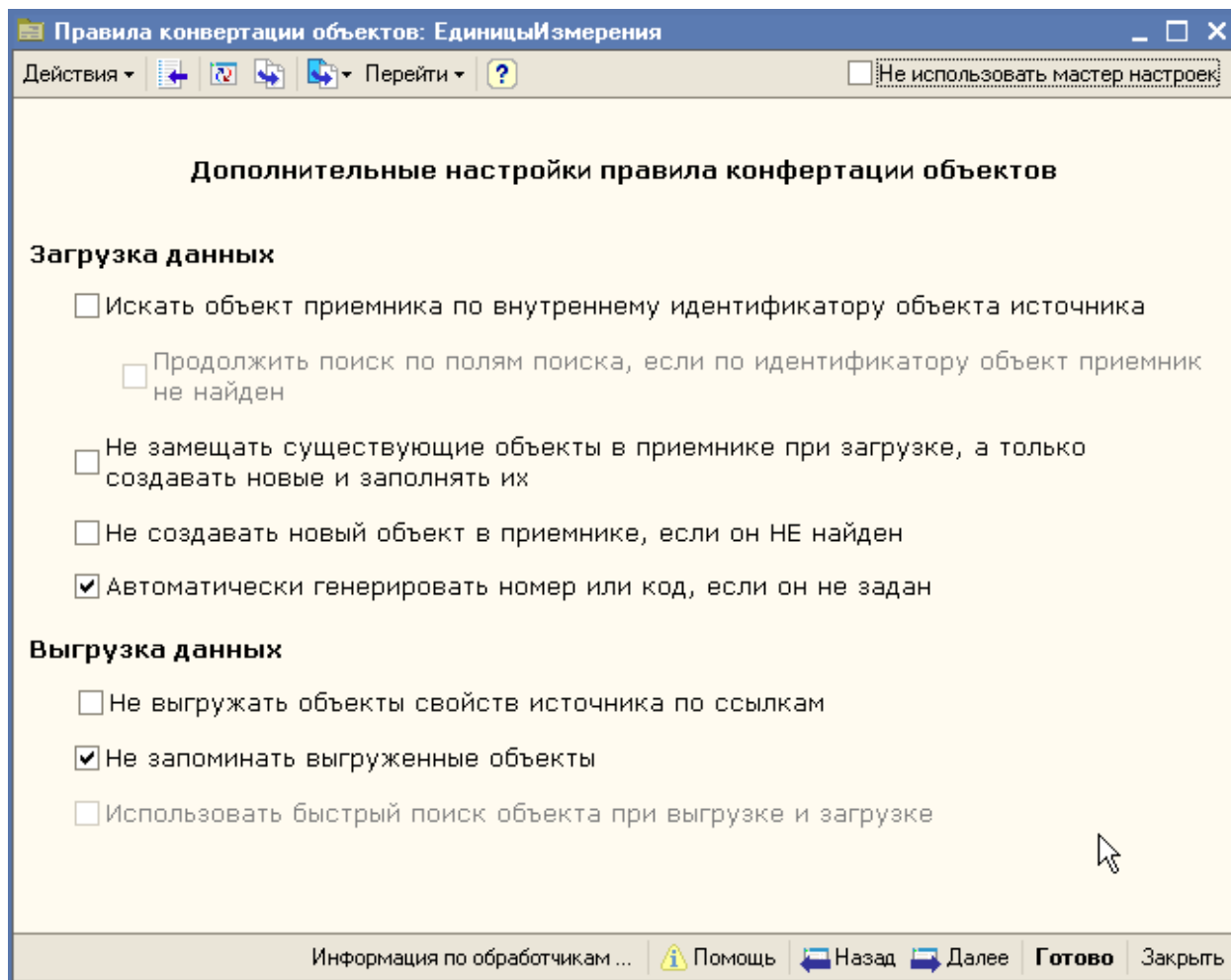


Рис. 22.

Подробность. Программа нормально воспринимает ситуацию, когда в ПКО, где нет источника, стоит флажок поиска по внутреннему идентификатору. В этом случае поиск ведется по отмеченным для поиска полям.

Теперь синхронизируем свойства: нам нужно, чтобы у загруженных единиц были заполнены свойства Наименование, ЕдиницаПоКлассификатору, Коэффициент и Владелец. Если Наименование, ЕдиницаПоКлассификатору и Коэффициент из классификатора заполнить легко, то ссылку на владельца оттуда не получишь. Для этого проще всего также использовать входящие данные.

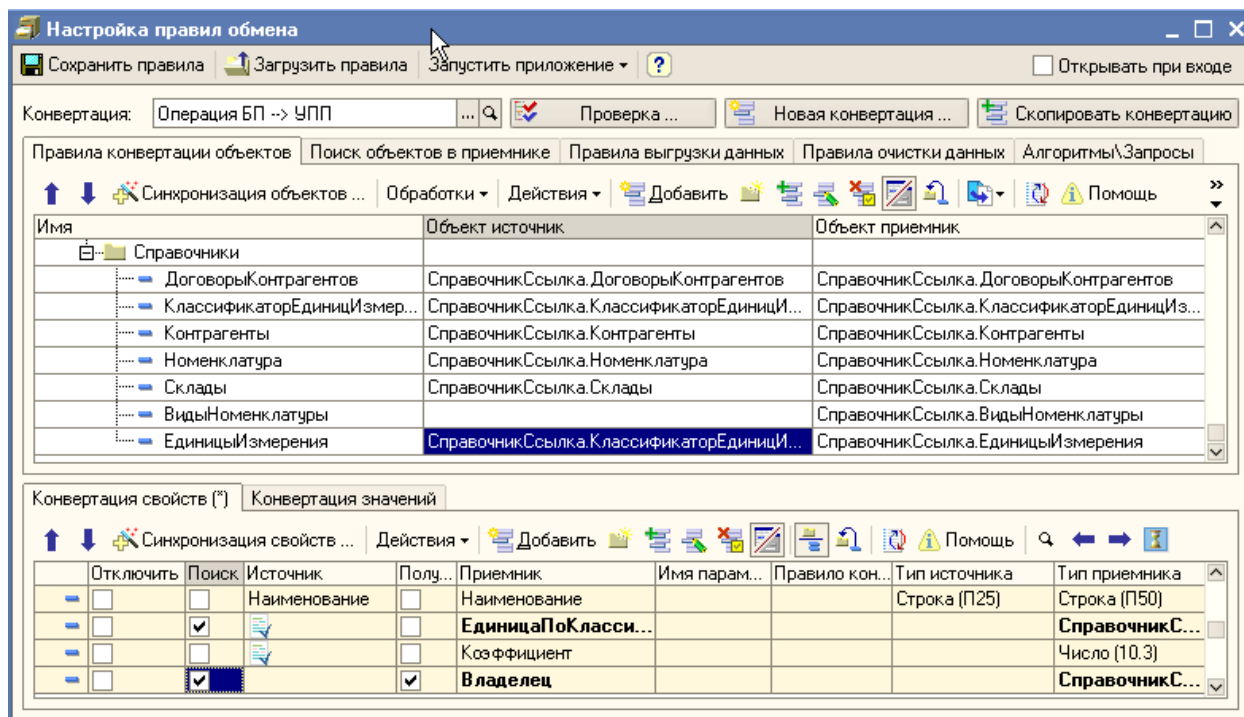


Рис. 23.

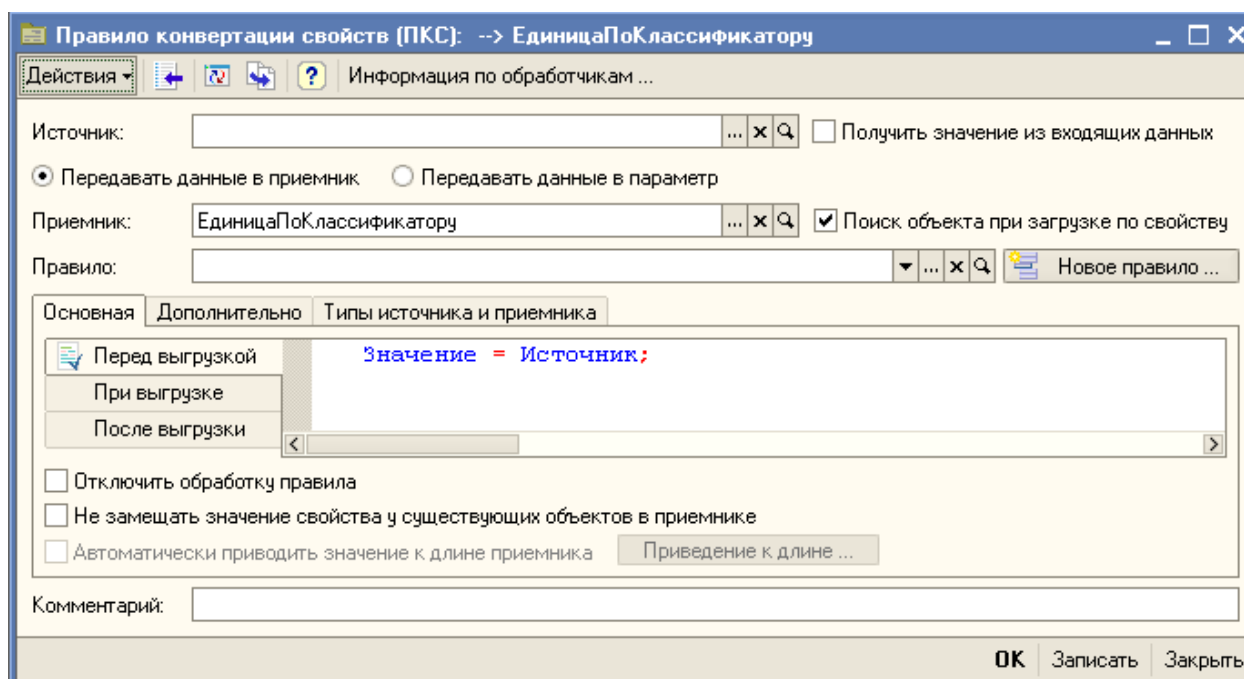


Рис. 24.

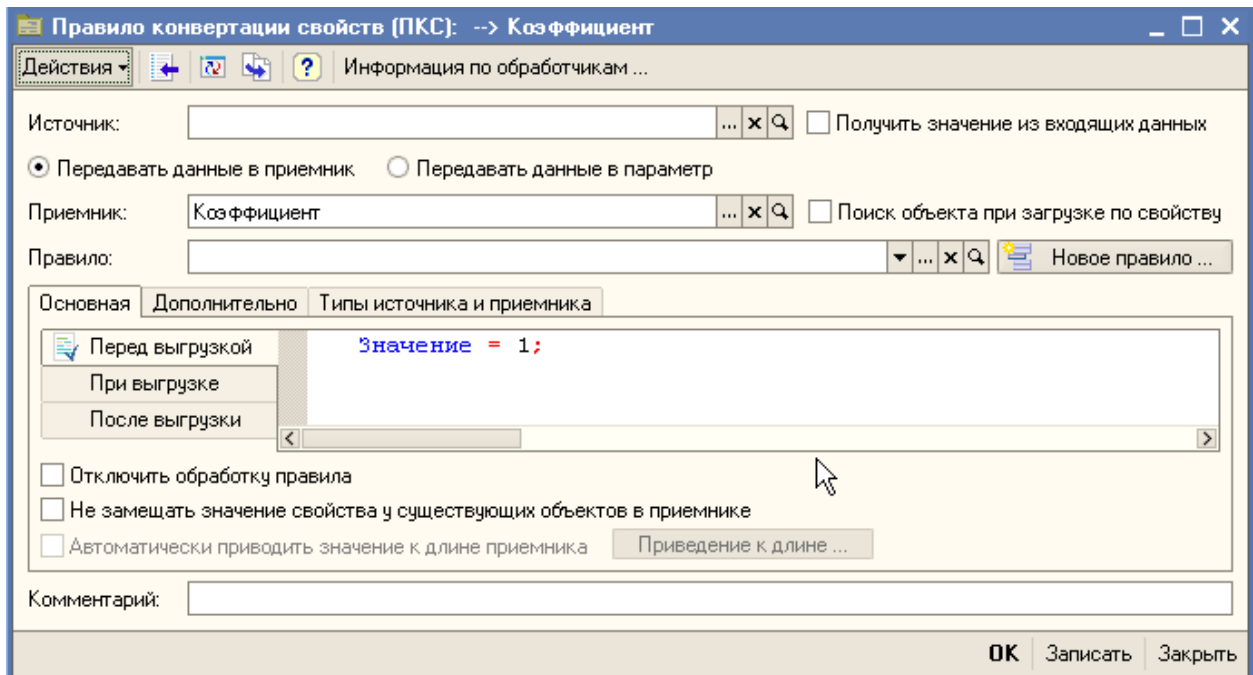


Рис. 25.

Теперь наше созданное ПКО указываем для ПКС Единица хранения остатков и пишем там обработчик события с заполнением исходящих данных.

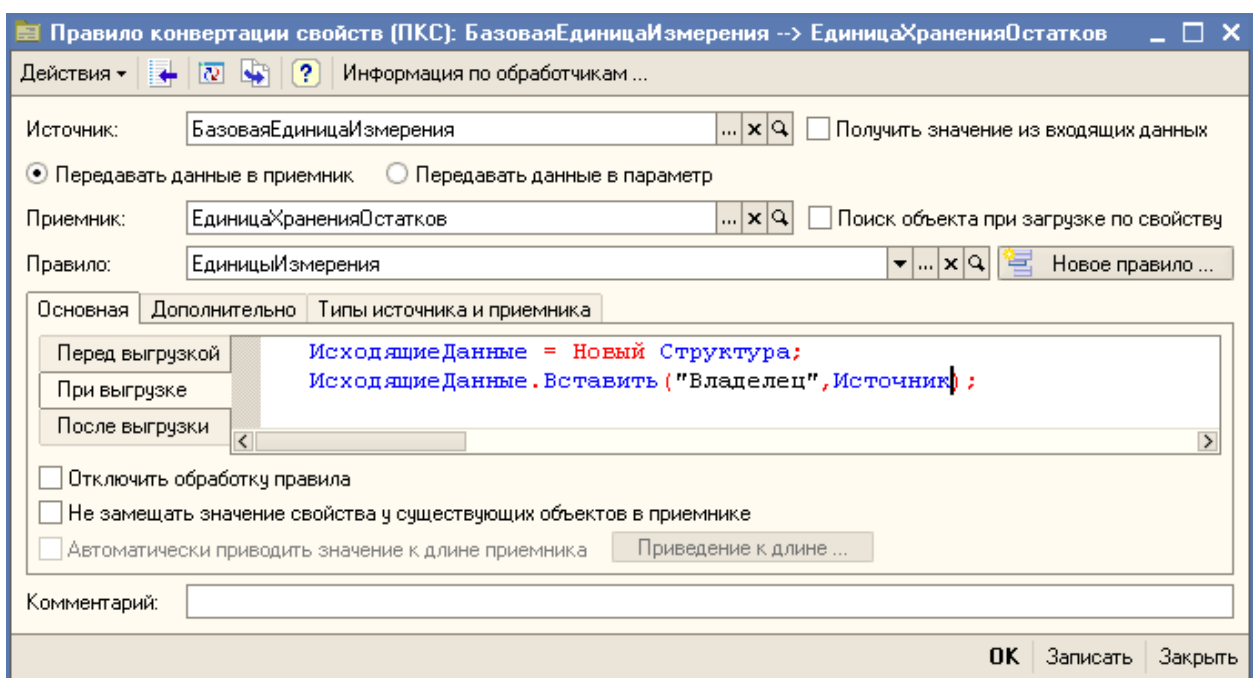


Рис. 26.

Теперь правила для переноса номенклатуры настроены, но необходимо еще учесть, что не все свойства определены для групп справочника номенклатура. Чтобы из-за этого не возникало ошибок, помещаем все свойства, определенные для элемента, но не определенные для группы в базе-приемнике, то есть в УПП, в одну группу свойств и в событии перед обработкой в ней ставим проверка на свойство ЭтоГруппа.

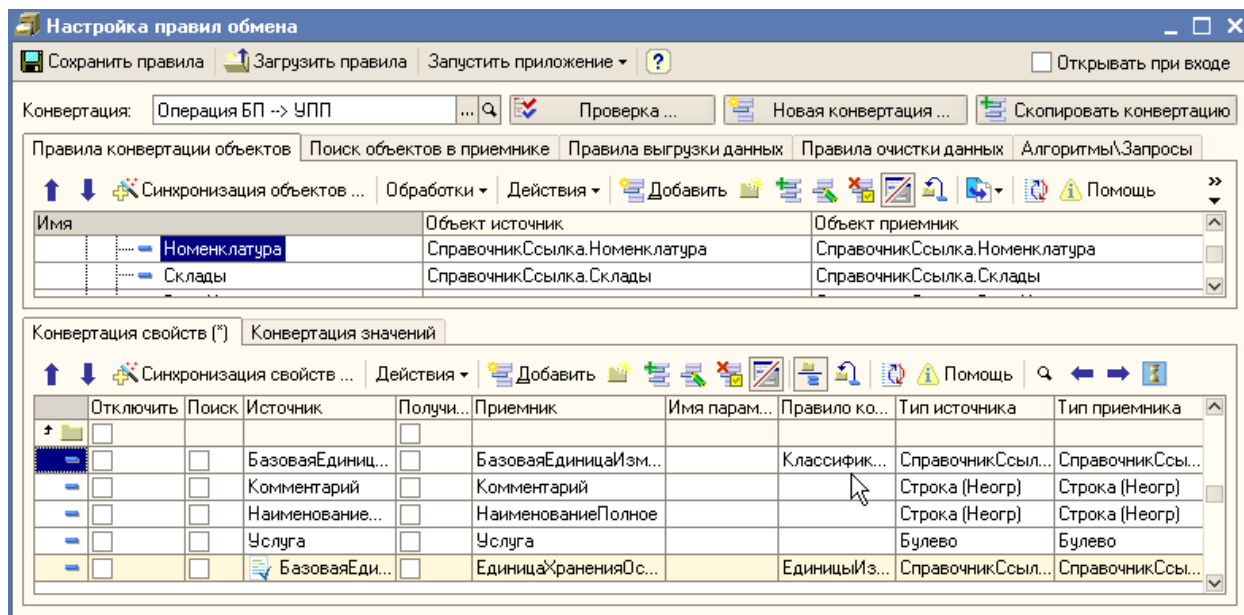


Рис. 27.

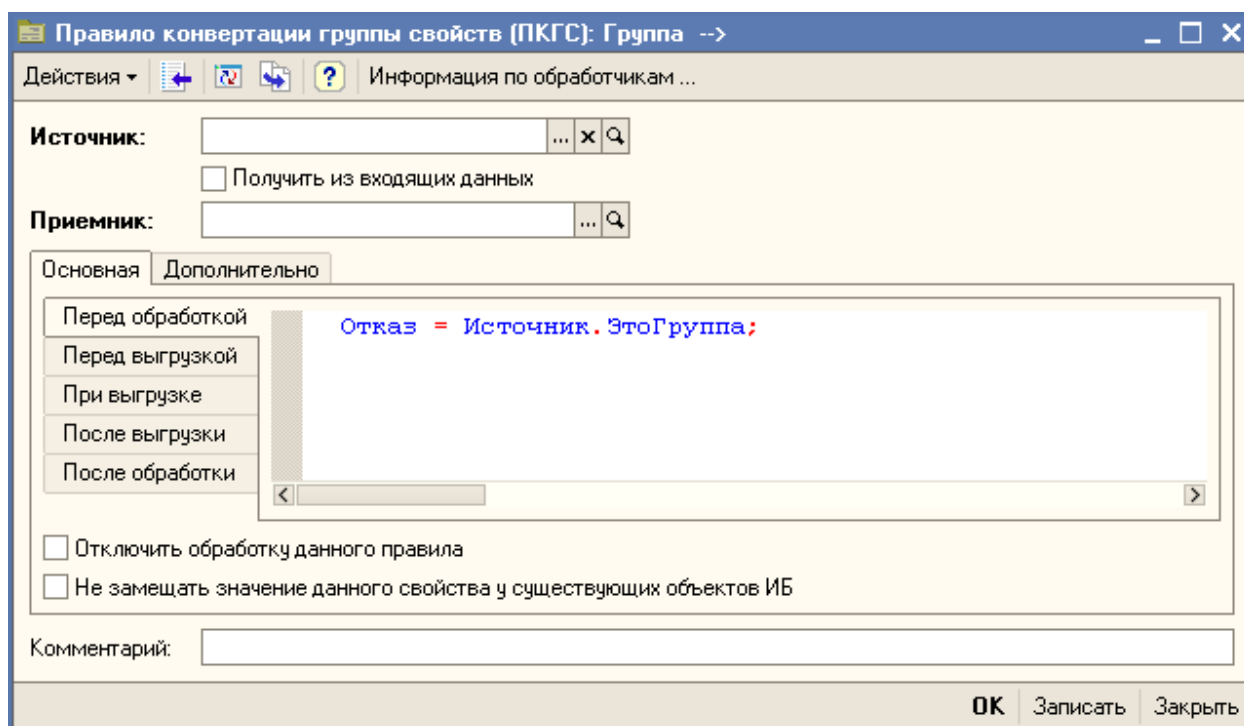


Рис. 28.

Пример: Справочник Организации. Настройка правила без источника

Настраиваем справочник Организации. Здесь все зависит от задачи – можно создать правило перегрузки справочника Организации в Бухгалтерии в справочник Организации в УПП, но если, например, в УПП загружаем таблицы в разрезе совсем других организаций, не таких, как введены в Бухгалтерии, то правило настраиваем без источника.

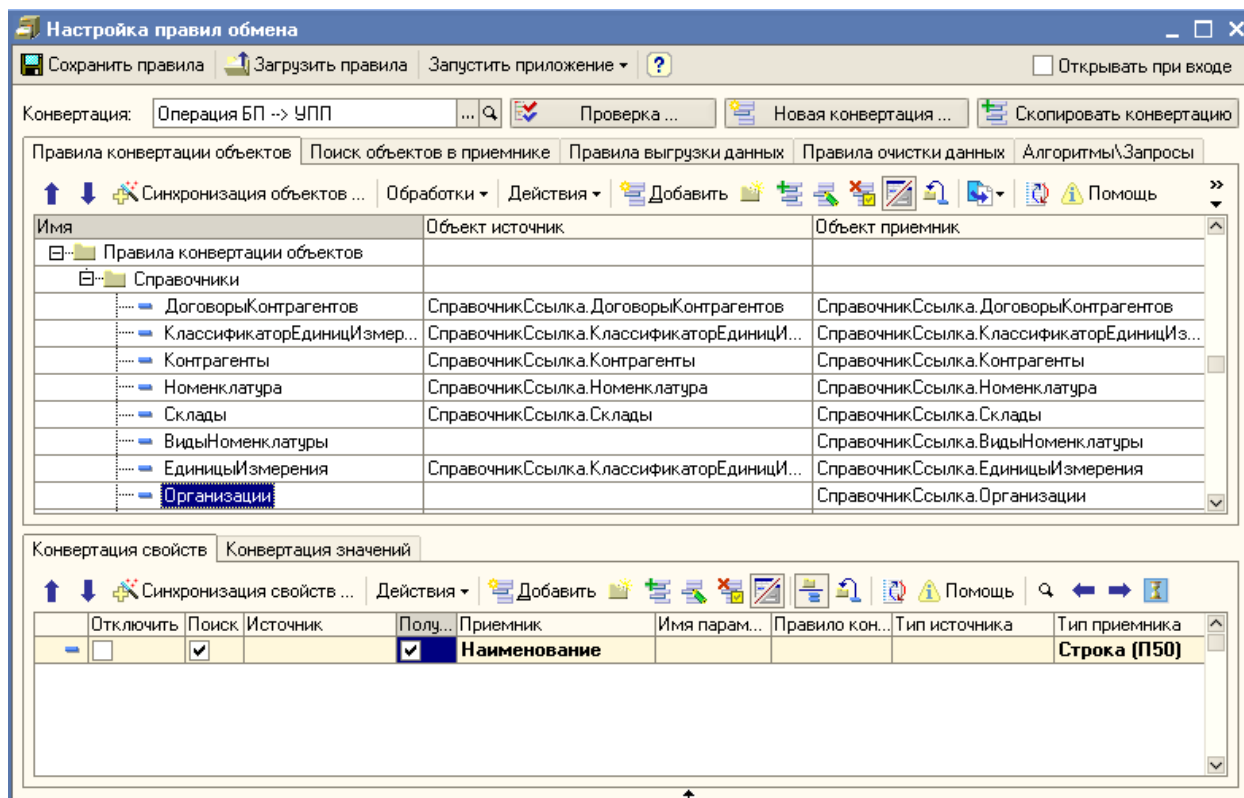


Рис. 29.

То, что свойство в нашем правиле только одно, не будет означать, что в справочнике Организации будет заполнено только наименование. Мы перед загрузкой вручную можем создать все необходимые организации в базе – приемнике, заполнить у них все необходимые реквизиты, а по наименованию будет только производиться синхронизация для установки правильных ссылок на организацию в документах. Но чтобы наши заполненные элементы справочника Организации не замещались на загружаемые элементы, у которых заполнено только Наименование, необходимо в ПКО Организации поставить флажок «Не замещать существующие объекты в приемнике при загрузке».

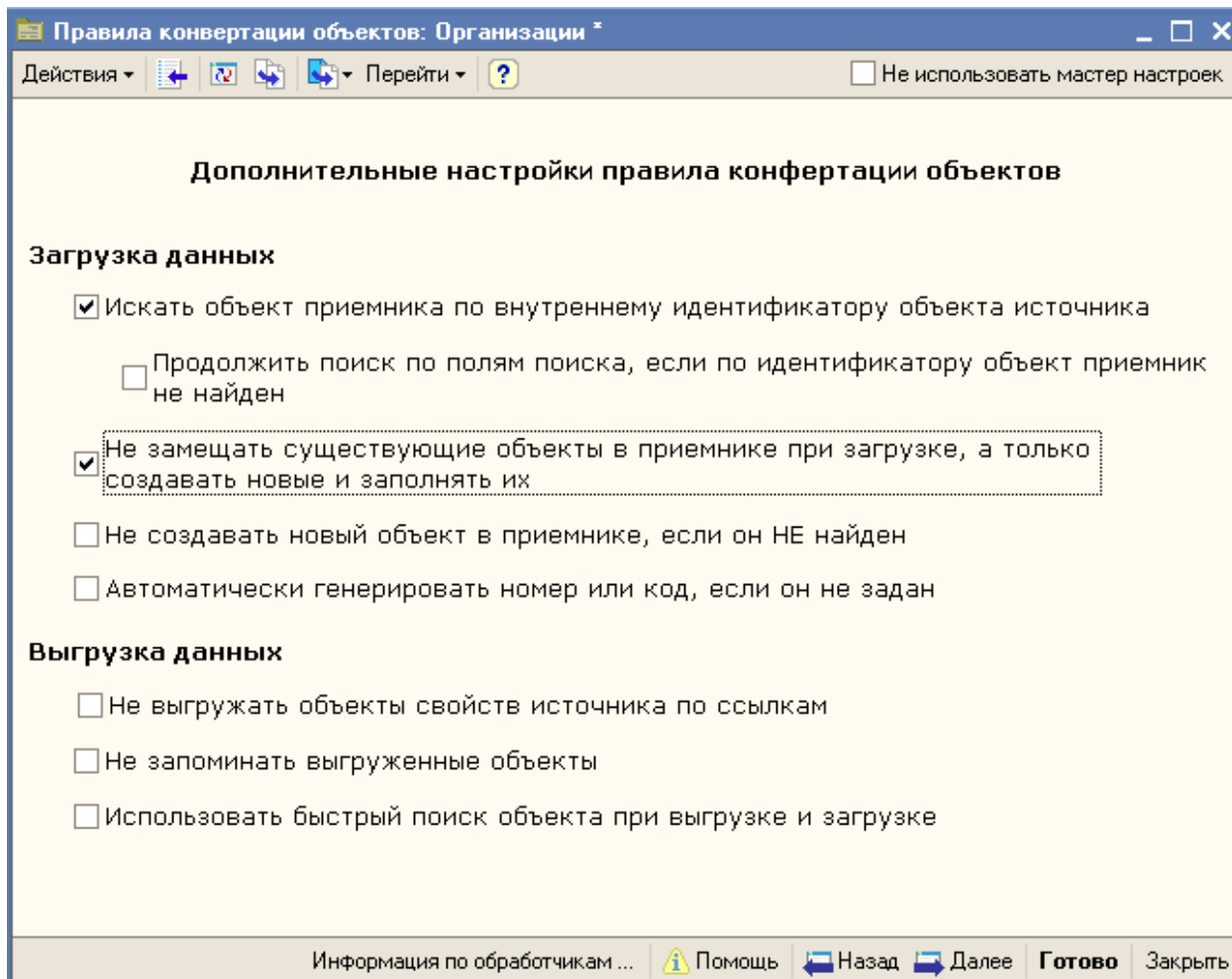


Рис. 30.

В правиле для справочников Классификатор Единиц Измерения и Валюты надо снять галку поиска по внутреннему идентификатору, так как эти элементы уже есть при первоначальном заполнении базы, поэтому при загрузке будут дублироваться.

Пример: Документ-Основание. Настройка ПКС для свойств с составным типом данных

Часто надо выгрузить, например, документ со свойством Документ-Основание, которое может иметь разный тип в зависимости от того, какой документ в каждом случае являлся основанием для данного. В этом случае в ПКС нельзя однозначно указать, какое ПКО использовать для конвертации этого свойства. В этом случае можно написать алгоритм, который бы в зависимости от типа объекта, который находится в этом свойстве, подбирал подходящее ПКО. Для этого используем обработчик события При Выгрузке, так как в момент его выполнения уже получено значение свойства, тип которого нам надо определить.

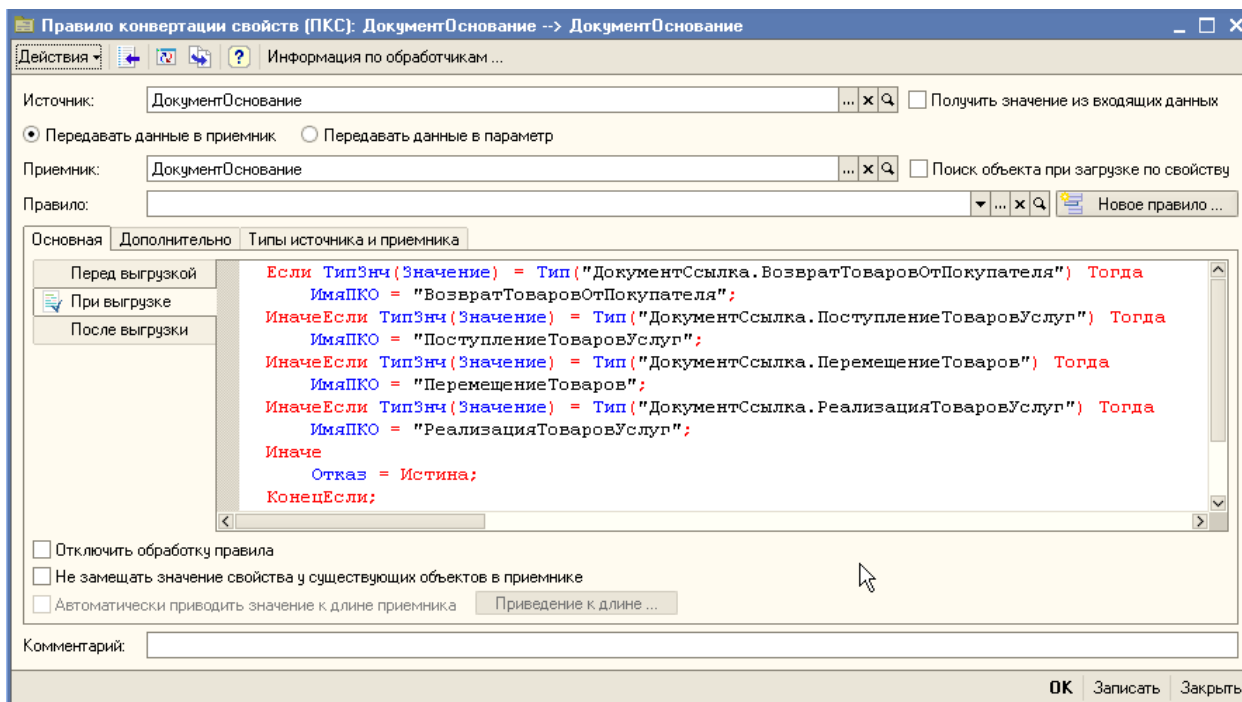


Рис. 31.

Пример: Штрих Коды. Перенос реквизита справочника как записи регистра сведений и синхронизация объектов по этой записи

В базе-источнике ШтрихКод – это реквизит справочника Номенклатура, в базе-приемнике ШтрихКод хранится в регистре сведений. Необходимо перенести номенклатуру с поиском по ШтрихКоду.

Здесь надо решить две задачи: при переносе номенклатуры загрузить штрихкоды в регистр, и синхронизировать номенклатуру по этим записям.

Для этого сначала создаем правило для номенклатуры. Все реквизиты кроме штрихкода настраиваем как обычно. Флажки Поиск пока ставить не будем. Чтобы при выгрузке Номенклатуры выгружалась запись регистра сведений, надо создать ПКО для регистра сведений и передать ему управление в одном из ПКС номенклатуры.

Создаем ПКО для регистра. Источника нет, все свойства придется получать из входящих данных. Для простоты настраиваем только необходимые свойства.

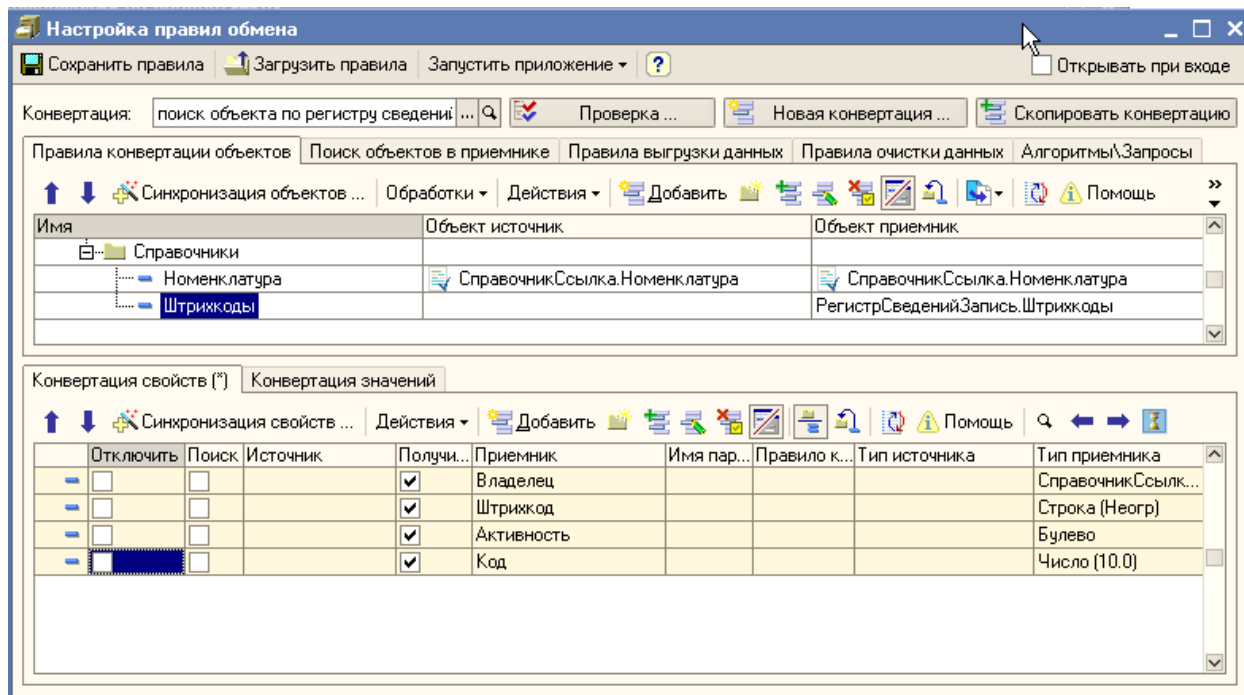


Рис. 32.

При переносе регистра сведений обязательно нужно создать ПКС для всех измерений, так как синхронизация записей регистра производится именно по измерениям. В регистре ШтрихКоды измерение только одно – это код, и его необходимо задать так, чтобы по нему мог осуществляться поиск записи. В данном случае можно, например, использовать код справочника Номенклатура. Теперь необходимо передать управление в это ПКО. Для этого у номенклатуры создаем ПКС Комментарий -> Комментарий (или любое другое, оно работать не будет, а будет только выгружать запись регистра сведений).

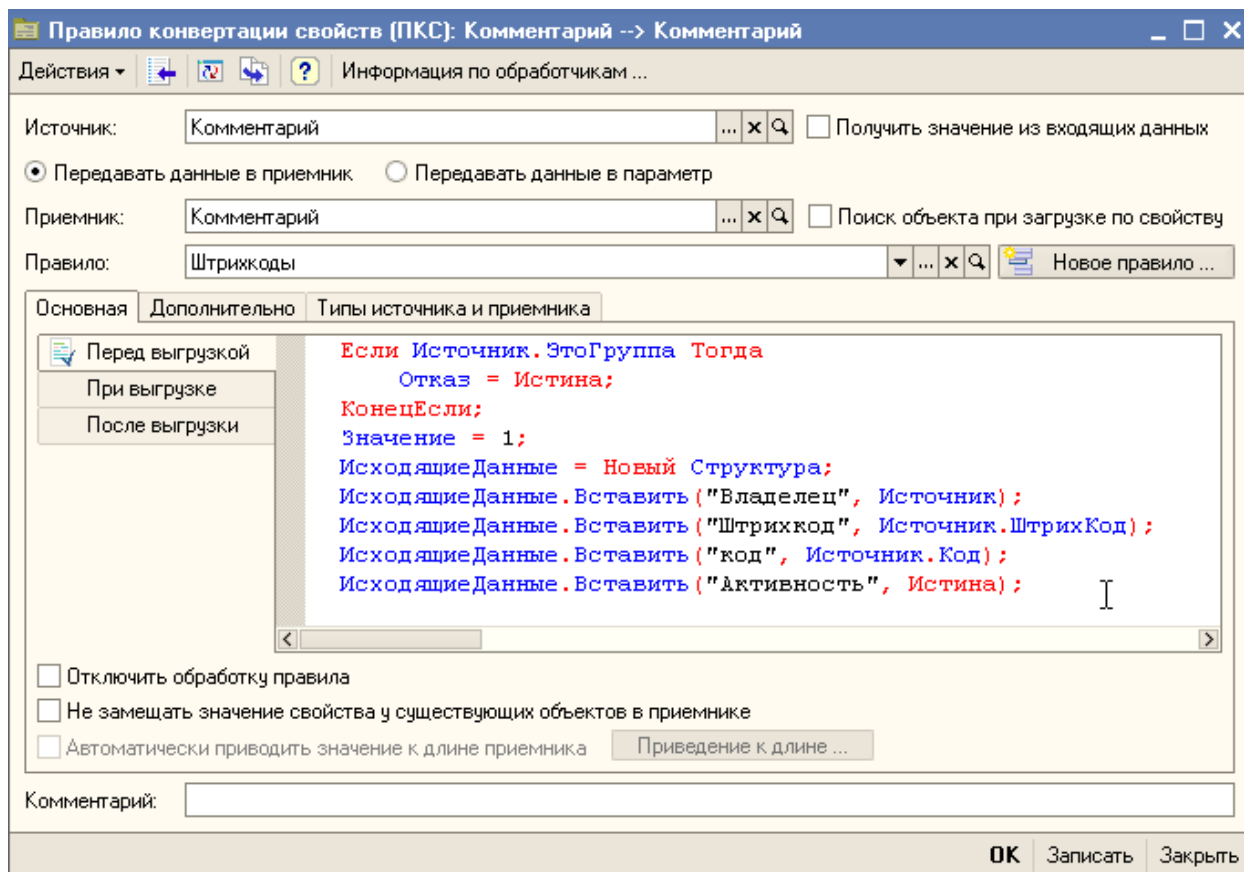


Рис. 33.

В нем надо заполнить структуру ИсходящиеДанные, записать что-нибудь в переменную Значение (чтобы она не была пустой, иначе обработка правила прервется) и передать управление в правило ШтрихКоды.

Кроме того, необходимо, чтобы это правило выполнялось только для элементов, поскольку для групп штрих коды не актуальны. Однако группы нам все-таки нужны, но искать их будем традиционно, по полям поиска. Поэтому в ПКО для номенклатуры необходимые поля надо пометить флагом поиска.

В выгружаемом элементе Номенклатура также надо где-то записать сам штрихКод, по которому будет осуществляться поиск. Для этого можно например вручную записать информацию в подчиненный узел, как показано в разборе примера из демобазы, или использовать какой-либо текстовый реквизит. Например в нашем случае мы будем использовать реквизит комментариев. Данный реквизит у всех элементов пустой, поэтому мы не рискуем потерять информацию.

Штрихкод, записанный в комментариях, понадобится нам в обработчике Поля Поиска, а на момент исполнения этого обработчика доступны только свойства, помеченные флагом поиска. Поэтому ставим флаг поиска у ПКС Штрихкод -> Комментарий.

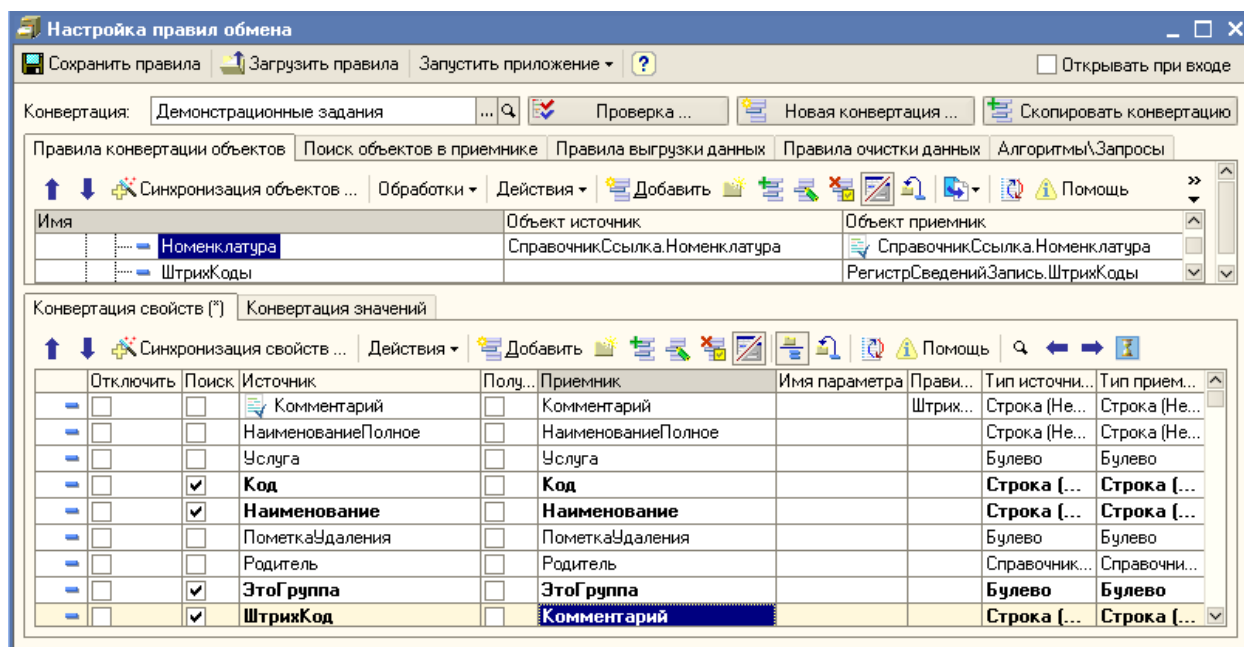


Рис. 34.

Теперь вторая часть задачи, синхронизация по записи регистра. Для этого в ПКО Номенклатура в обработчике ПоляПоиска пишем запрос к регистру сведений, которым получаем ссылку на нужную номенклатуру, и присваиваем ее переменной СсылкаНаОбъект.

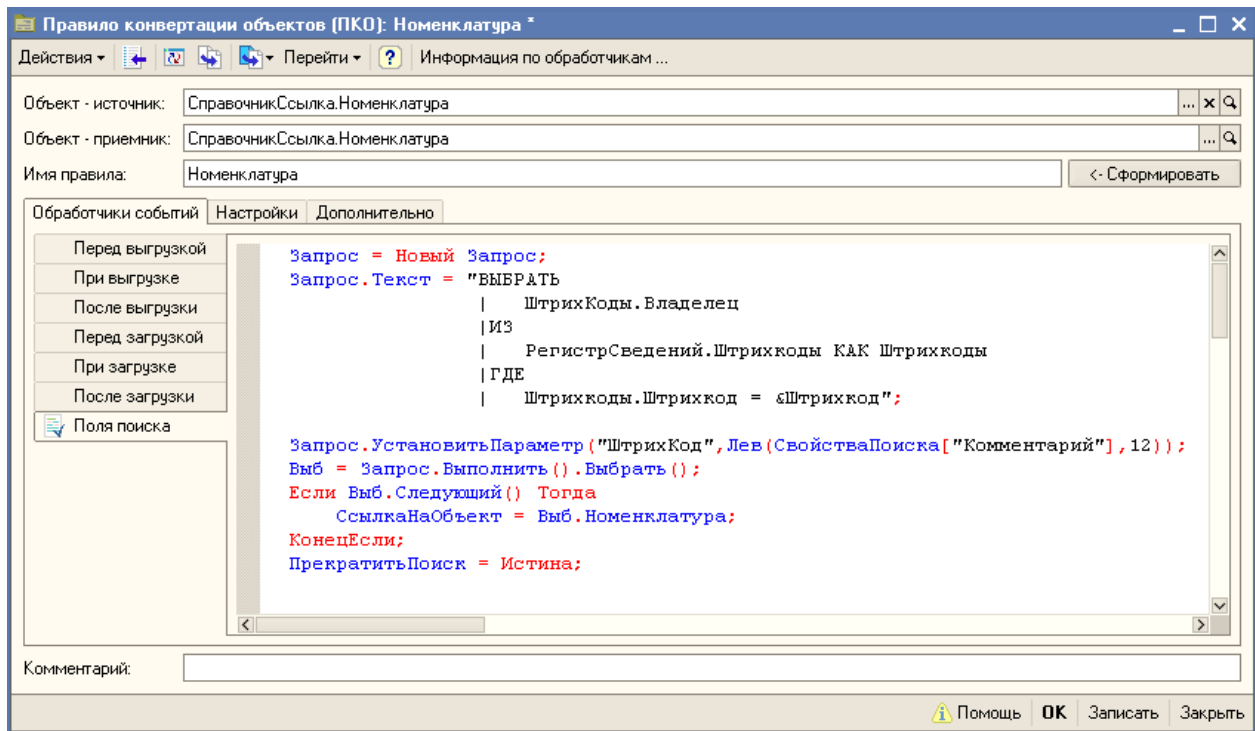


Рис. 35.

Теперь осталось вспомнить, что у групп справочника Номенклатура ШтрихКодов не бывает, да и у каких-то элементов он может быть не заполнен. Для этих случаев сделаем поиск не по штрихкоду, а по Коду, Наименованию и ЭтоГруппа.

Свойства поиска, которые в правиле были помечены флагами, в обработчике Поля Поиска доступны в структуре СвойстваПоиска. Чтобы отменить поиск по некоторым из них, в переменную СтрокаИменСвойствПоиска необходимо записать текстовую строку, где через запятую перечисляются те свойства, по которым будем проводить поиск, например:

СтрокаИменСвойствПоиска = «Код, Наименование, ЭтоГруппа»;

Но в этом случае после нахождения объекта в базе или создания нового будет совершена попытка заполнить значениями все реквизиты из структуры СвойстваПоиска. В нашем случае будет попытка заполнить свойство комментарий. Но свойство комментарий не определено для групп справочника, поэтому будет выдана ошибка. Вместо этого мы просто удалим свойство комментарий из структуры свойств поиска.

Изменяем код в нашем обработчике Поля Поиска:

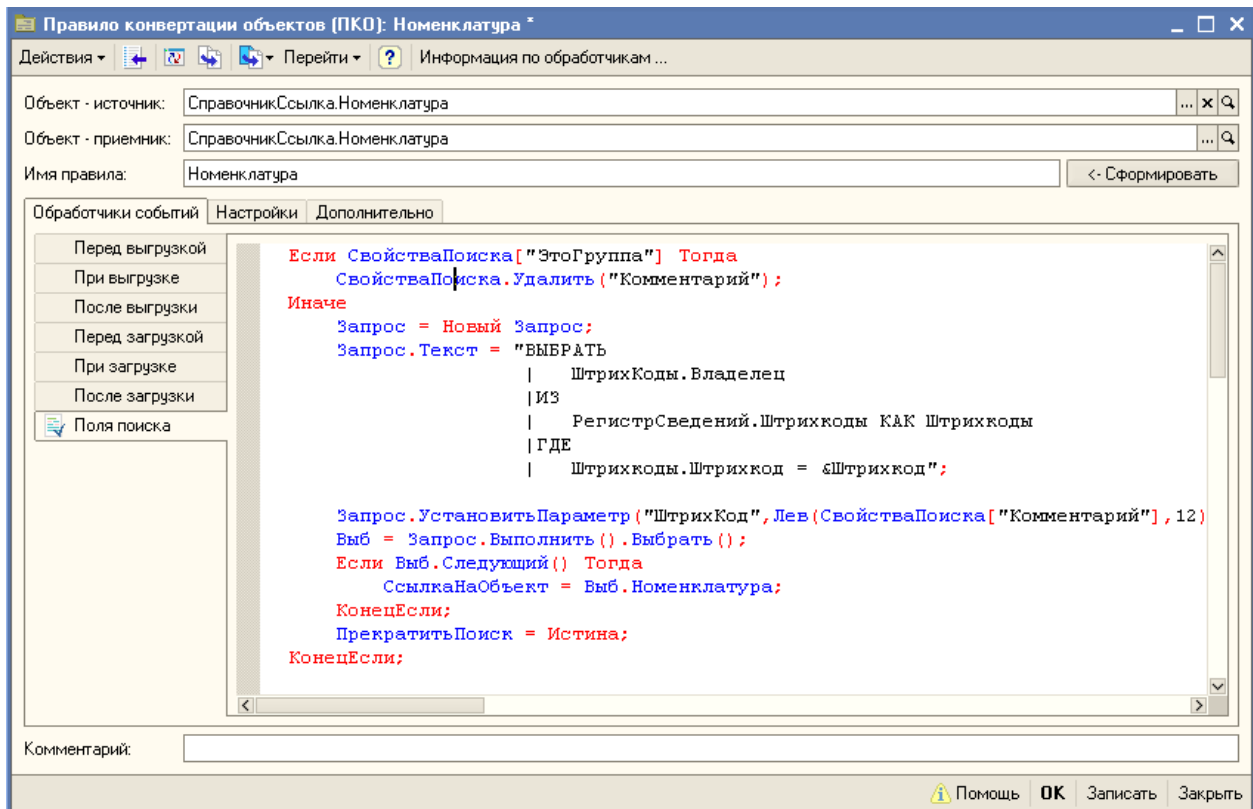


Рис. 36.

Пример: Поиск контрагентов по ИНН. Синхронизация по полю, недоступному для группы

Очень часто справочник Контрагенты необходимо синхронизировать по полю ИНН, а Номенклатуру – по артикулу. Но если просто у этих ПКС поставить флажок поиска, то при загрузке будет возникать ошибка, связанная с тем, что эти реквизиты недоступны для групп справочника.

Группу надо искать по Родителю, Наименованию, ЭтоГруппа, а элемент – по ИНН. В ПКО для контрагентов необходимо пометить флагом поиска все эти свойства – Родитель, Наименование, ЭтоГруппа и ИНН.

А в обработчике «Поля поиска» для контрагентов пишем:

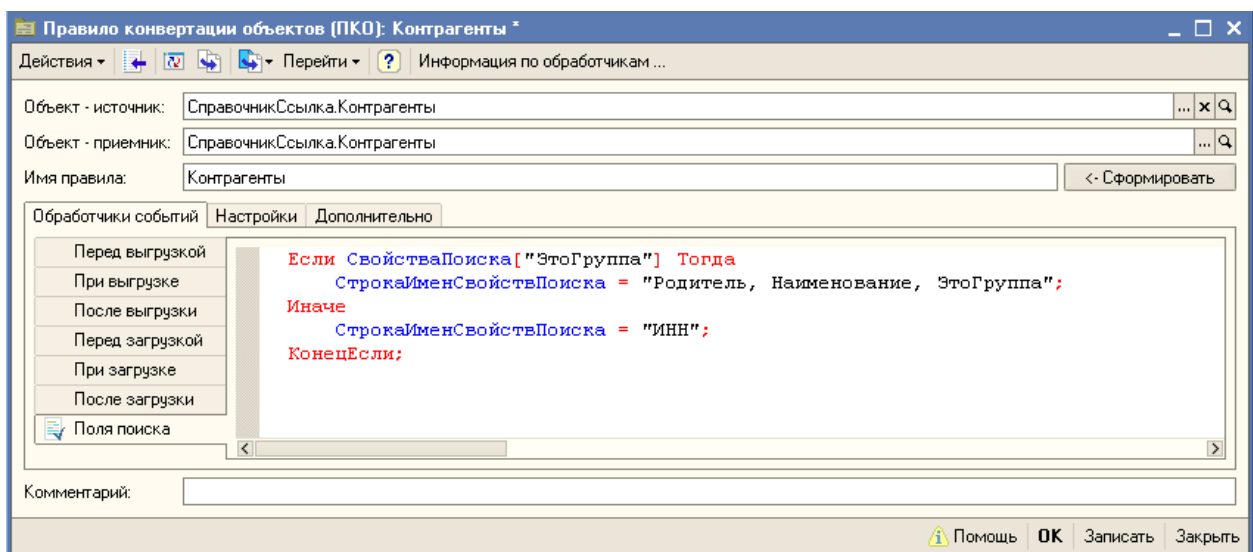


Рис. 37.

Кроме того: всегда независимо от того, нужен ли по ним поиск, у всех ПКС, которые в приемнике недоступны для группы, необходимо в событии Перед Выгрузкой написать проверку:

Отказ = Источник.ЭтоГруппа;

Пример: Справочник Организации. Ключ выгружаемых данных и оптимизация выгрузки

Правило конвертации объектов Организация не имеет источника и заполняется полностью из входящих данных. Синхронизировать организации надо по наименованию. Поскольку реквизит Организация присутствует во всех документах, для оптимизации переноса данных надо запоминать выгруженные организации.

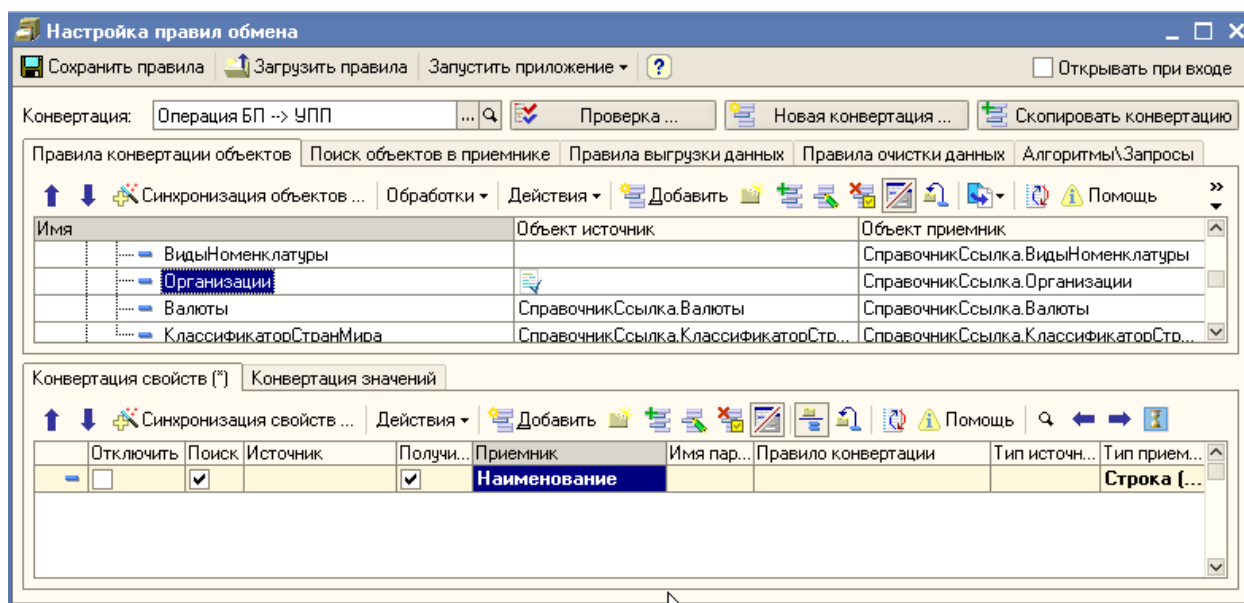


Рис. 38.

Если настроить простейшее ПКО Организации, где ПКС заполняются из входящих данных, то, выполнив выгрузку и открыв XML-файл редактором xml-файлов, мы увидим, что организация выгружается каждый раз, когда встречается ссылка на нее.

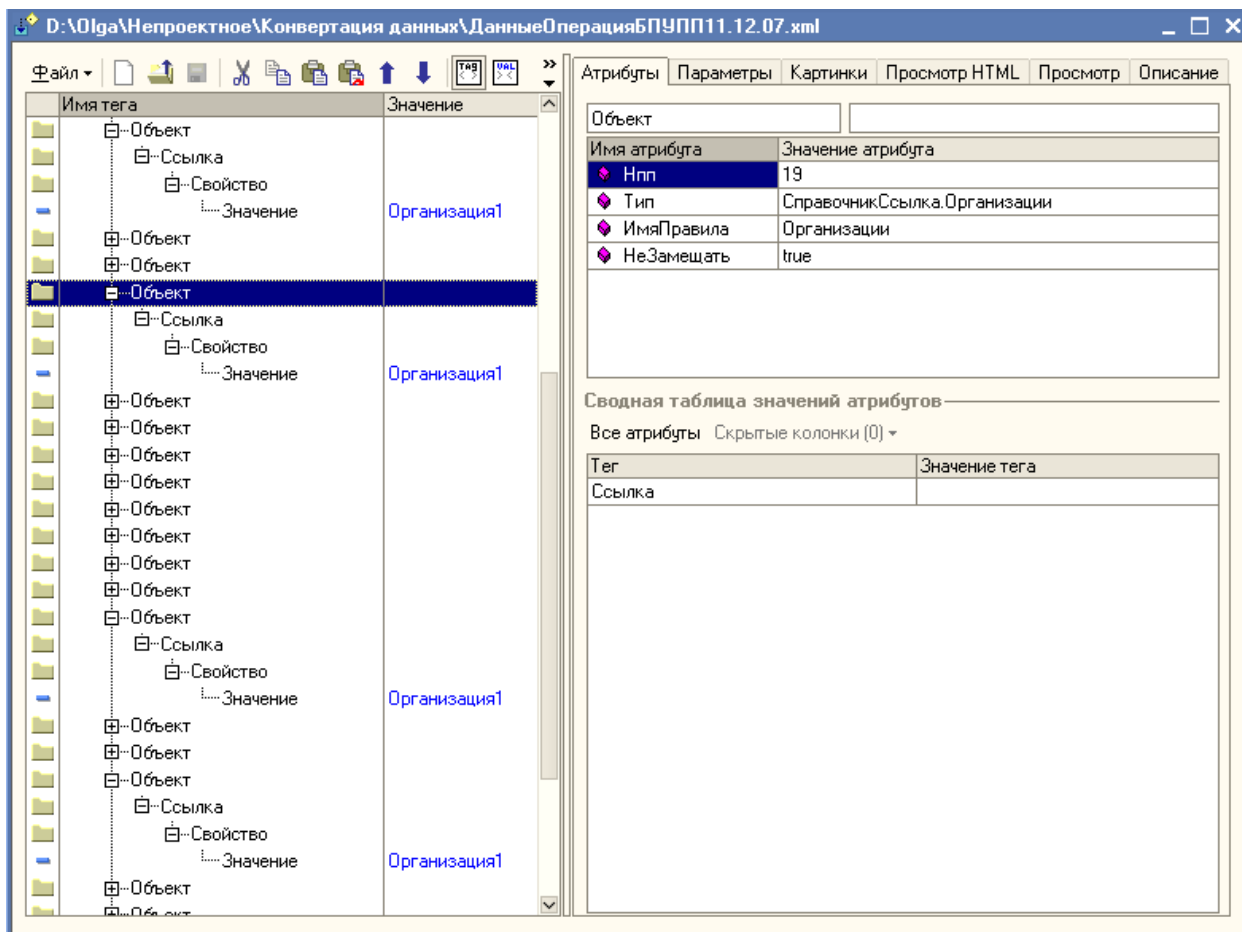


Рис. 39.

Значение флажка «Не запоминать выгруженные объекты» в данном случае (когда у ПКО нет Источника) не влияет на то, будут они запоминаться или нет. Чтобы выгруженные организации сохранялись и не выгружались повторно, пишем обработчик события Перед Выгрузкой ПКО Организации:

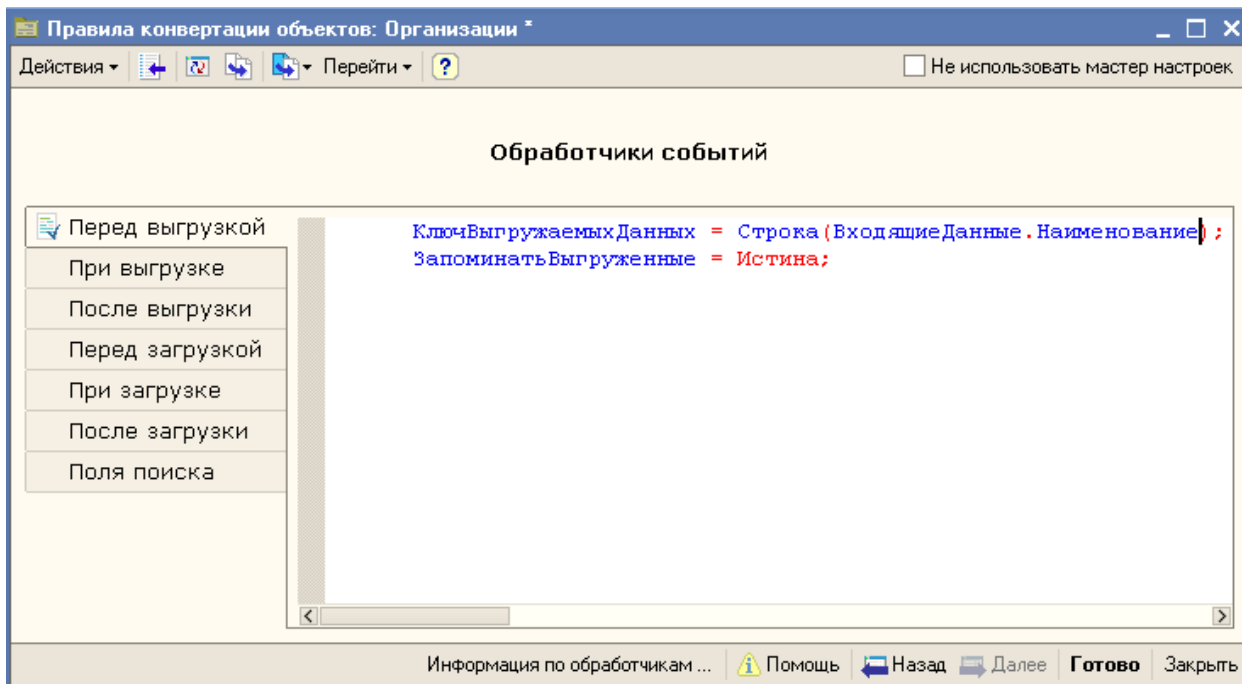


Рис. 40.

Теперь выгруженные организации будут сохраняться.

Пример: Конвертация групп и элементов справочников по разным правилам

Иногда бывает нужно настроить правила для групп и элементов справочников по-разному. Для этого можно создать два разных ПКО для этого справочника. Если справочник выгружается непосредственно, не по ссылкам, то в правиле выгрузки данных алгоритм выборки надо поставить Произвольный и в событии Перед выгрузкой написать запрос, выбирающий из справочника только элементы. Выборку элементов надо записать в переменную ВыборкаДанных.

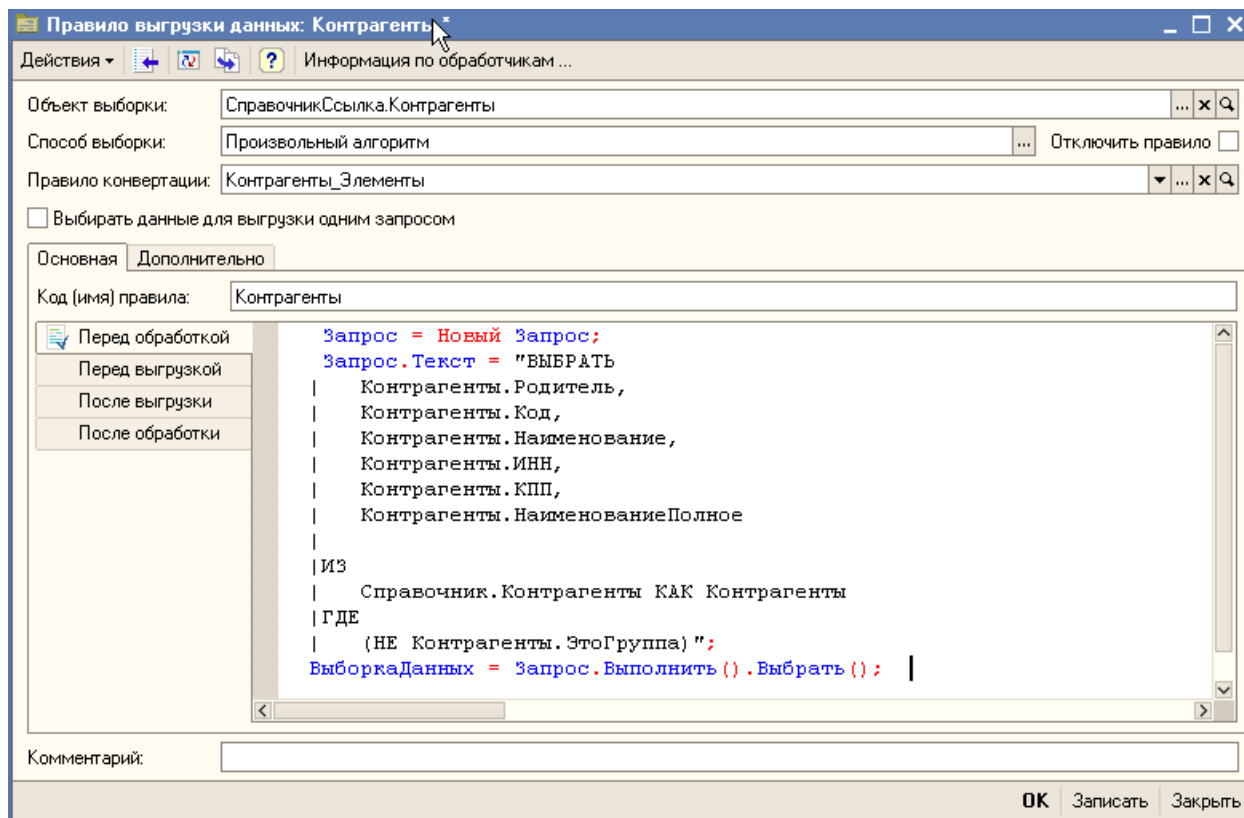


Рис. 41.

Если же справочник выгружается по ссылкам из документов, где могут фигурировать только элементы, то эта проблема уже решена. Далее в ПВД (или в соответствующем ПКС у документа, по ссылкам из которого выгружается справочник) указываем то ПКО, которое мы создали для конвертации элементов. В этом ПКО должно быть ПКС родитель, для которого в графе Правило указываем ПКО, созданное нами для групп.

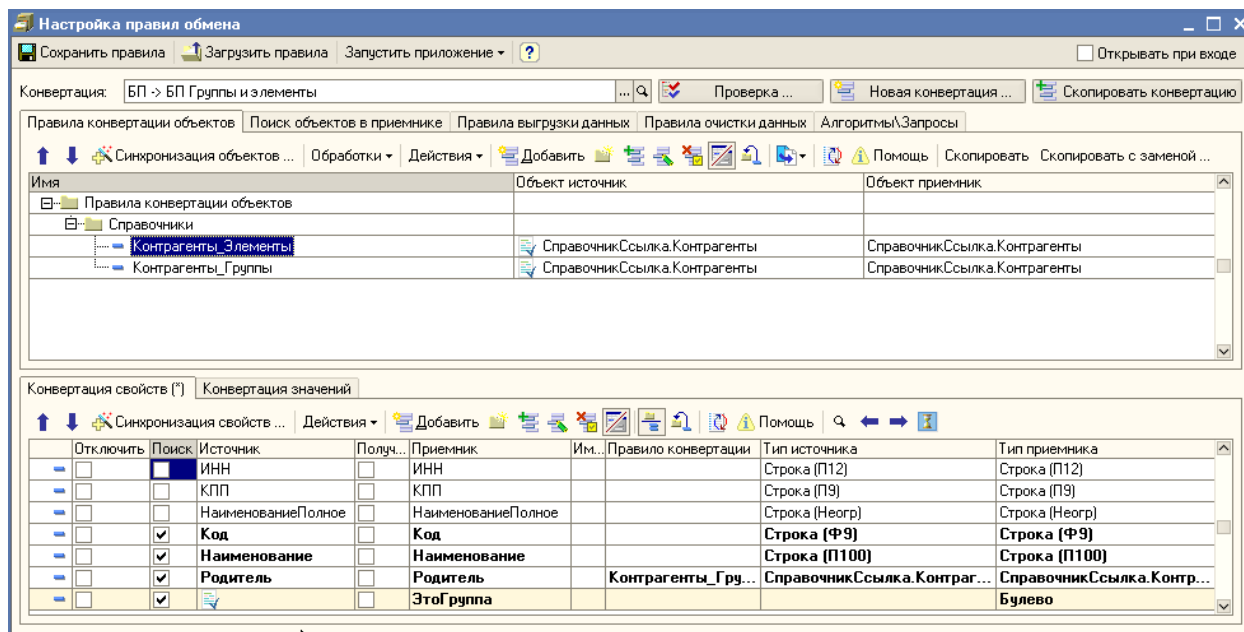


Рис. 42.

В ПКО, созданном для групп тоже может быть ПКС родитель, для него также указываем ПКО для групп.

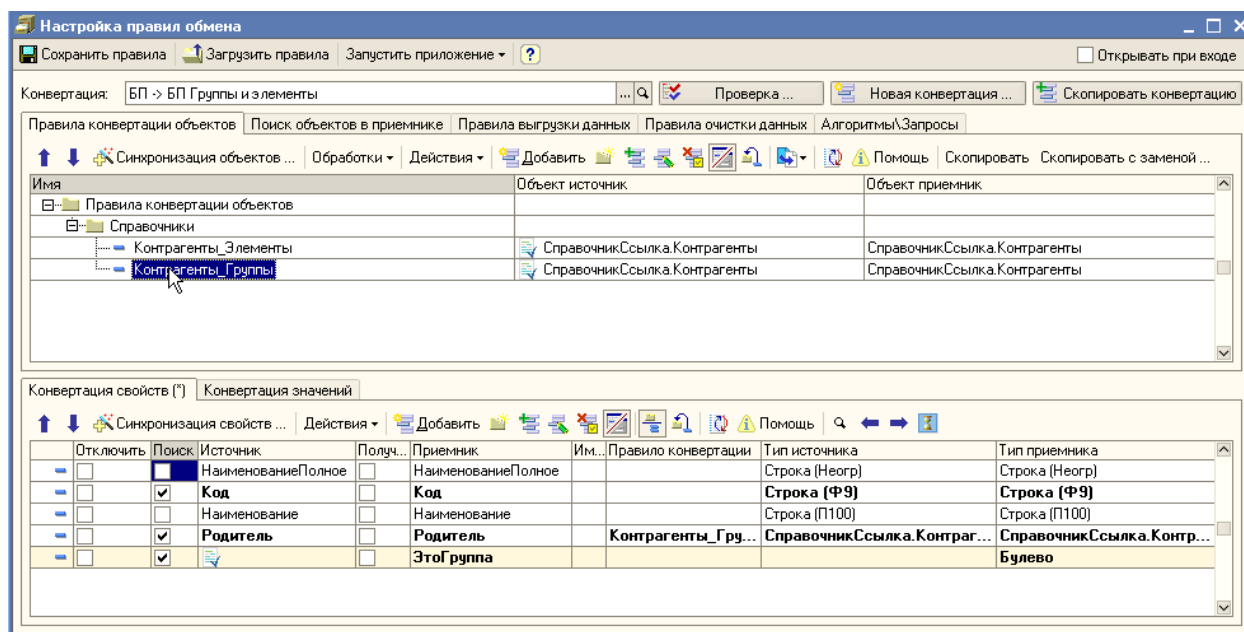


Рис. 43.

Таким образом, справочник будет перегружаться в иерархии, но группы и элементы будут конвертироваться по разным правилам.

Теперь если мы выгрузим наших контрагентов, мы увидим, что выгрузился только первый объект из выборки и его родитель. Это получилось потому, что в ПКО снят флаг «Не запоминать выгруженные», и ключ выгружаемых данных не определен. Обычно ключ выгружаемых данных – это адрес ссылки, но поскольку в этом примере используется выборка из результатов запроса, то поиск элементов будет проводиться по полям поиска, а ключ выгружаемых данных надо задать отдельно. Ключ должен содержать текстовую строку, уникальную для каждого элемента (и группы) справочника Контрагенты. Например:

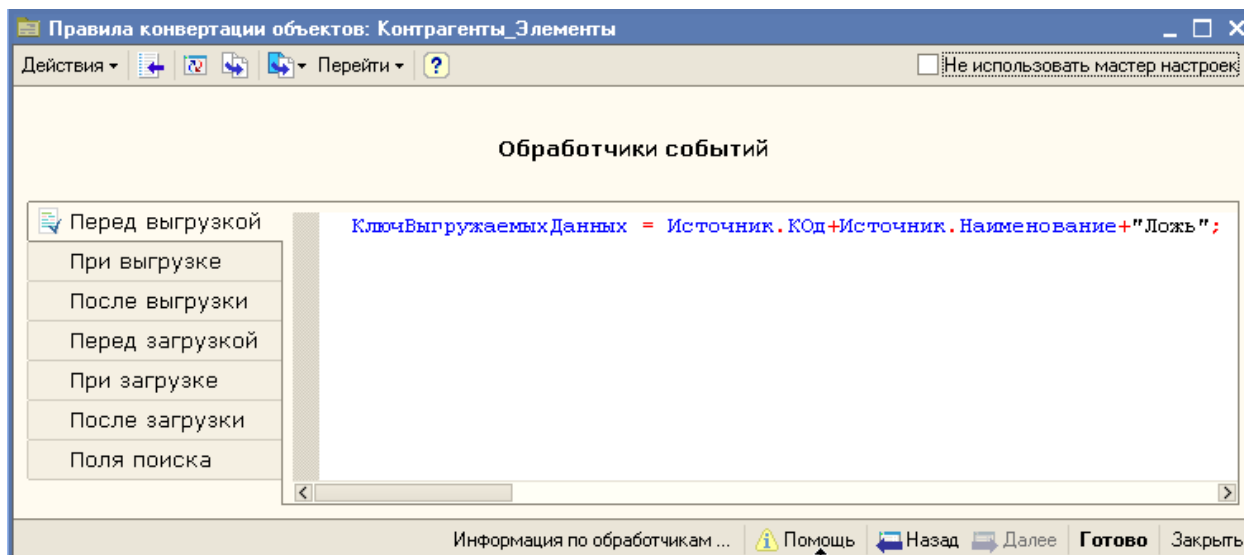


Рис. 44.

Вот теперь выгрузятся все элементы выборки.

Надо помнить, что, независимо от состояния флажка «Искать по внутреннему идентификатору», поиск элементов в нашем примере будет проводиться по полям поиска, так как ссылка на элемент не используется. А вот группы, если флажок включен, будут синхронизироваться по идентификатору.

Пример: Конвертация Характеристик Номенклатуры в Номенклатуру

Если выполняется обмен данными между БП и УТ или между БП и УПП, то встает задача как-то перенести номенклатуру вместе с характеристиками. Если, например, переносим УПП в БП, то, скорее всего, придется переносить элементы номенклатуры как группы, а характеристики – как элементы справочника номенклатура.

Для этого создаем два ПКО: Для переноса групп справочника и для переноса элементов.

Правило для групп довольно простое, настраиваем ПКС только для тех свойств, которые определены для группы в Бухгалтерии.

В ПКС для ЭтоГруппа нужно жестко задать Значение = Истина. Это надо сделать до того, как программа попытается получить значение из источника – то есть в событии Перед Выгрузкой.

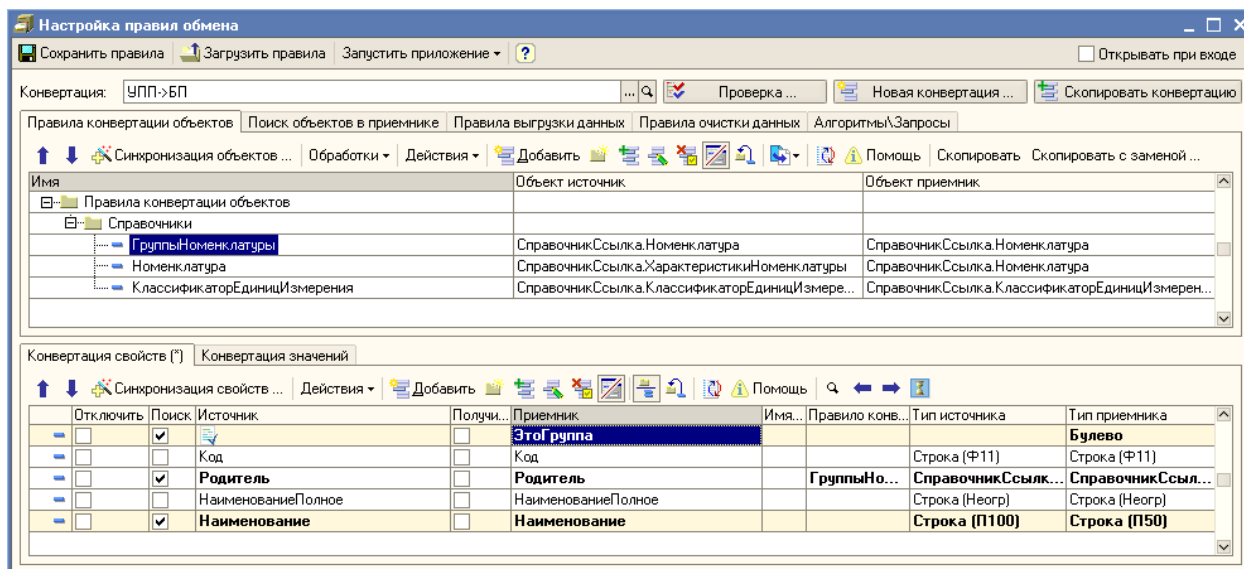


Рис. 45.

Второе ПКО – для переноса каждой характеристики как элемента справочника Номенклатура. Здесь настраиваем главное – ПКС для переноса владельца характеристики в группу-Родитель. Правило указываем то, которое создали для групп.

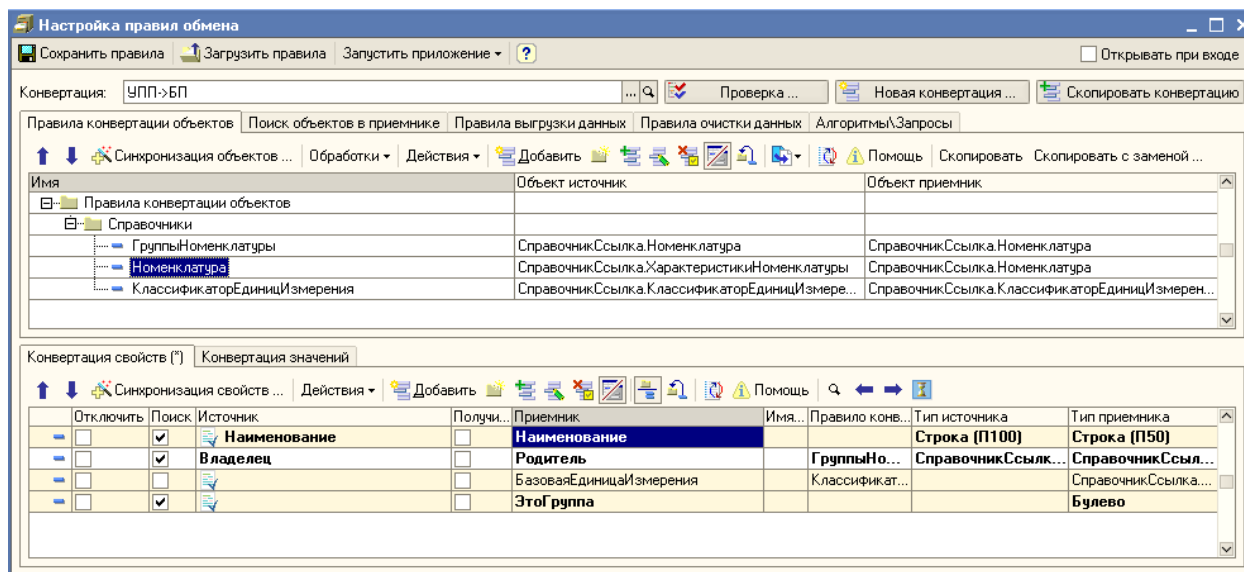


Рис. 46.

Здесь ЭтоГруппа жестко устанавливаем в Ложь в том же событии Перед Выгрузкой.

У номенклатуры в Бухгалтерии есть два обязательных для заполнения реквизита, их надо загрузить – это Наименование и Единица измерения. Поиск в обоих правилах должен быть по одному и тому же набору полей. Для номенклатуры и характеристик можно оставить поиск по уникальному идентификатору.

Наименование загрузим по принципу Наименование Номенклатуры + Наименование Характеристики. Для этого в ПКС для Наименования в событии При Выгрузке, когда уже в переменной значение содержится наименование характеристики.

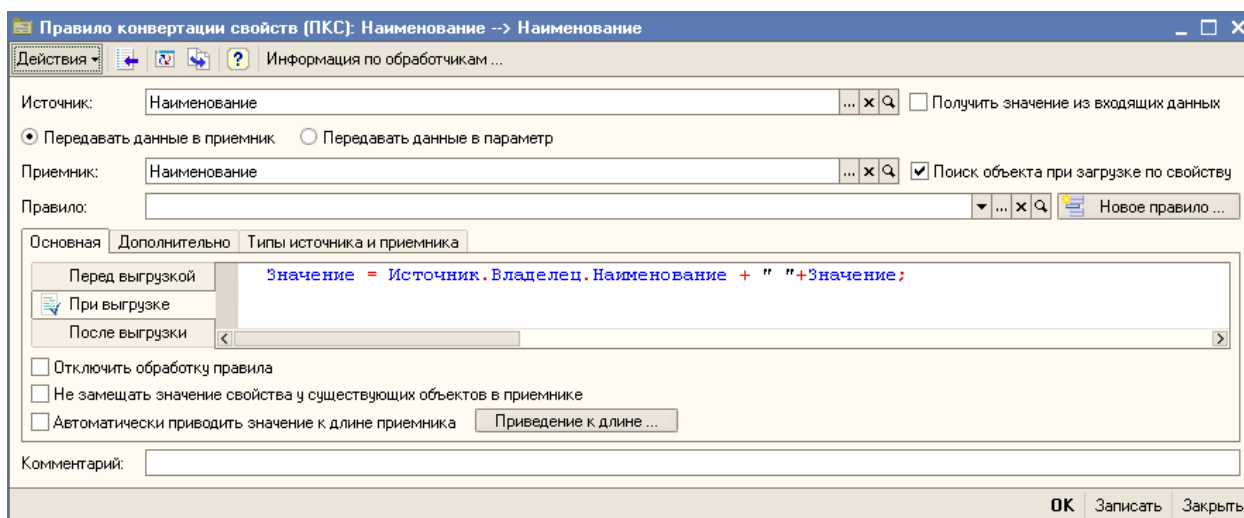


Рис. 47.

Единицу возьмем из владельца характеристики – справочника Номенклатура.

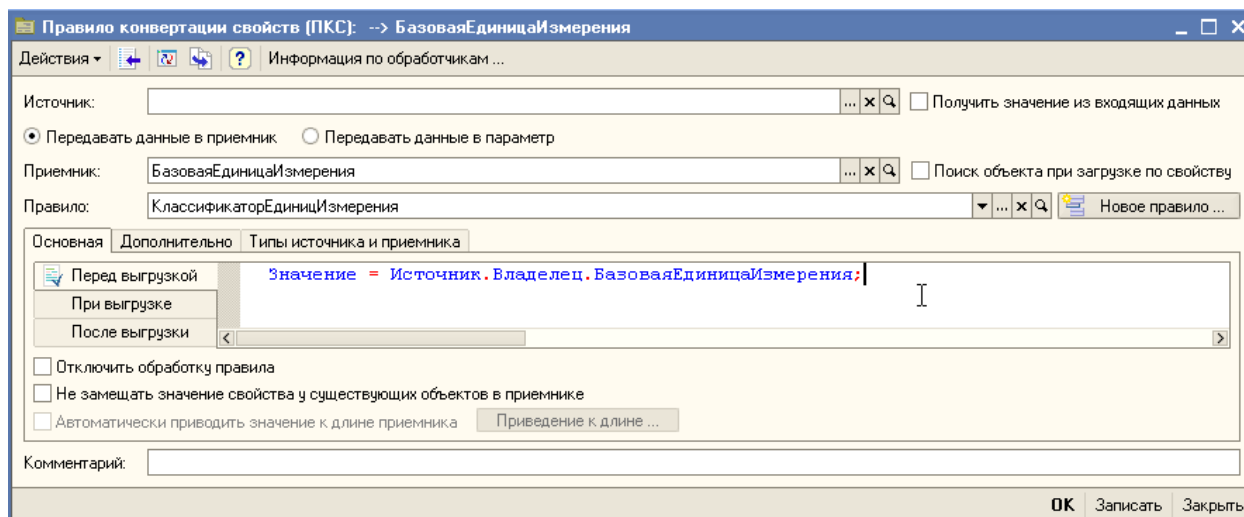


Рис. 48.

Теперь осталось только настроить ПКО для Классификатора Единиц Измерения – здесь сложностей нет. Для Классификатора Единиц, поскольку он обычно заполнен даже в пустой базе, поиск либо только по полям поиска, либо по полям поиска, если не найдено по идентификатору.

Пример: Операция (бухгалтерский учет). Конвертация таблицы значений в документ

Допустим, поставлена задача переноса неких произвольных данных, из которых была динамически сформирована таблица значений, как документ Операция. Таблицу будем формировать в конфигурации Бухгалтерия Предприятия, переносить в конфигурацию УПП.

По ссылкам из операции необходимо перенести также субконто, которые имеют разные типы данных. Для того, чтобы в нашу базу-приемник переносились значения субконто, надо настроить правила конвертации объектов (ПКО) для всех типов данных, какие могут встретиться нам в наших Операциях. Для простоты решим, что переносим проводки, в которых используются только счета 10 и 60. Для всех типов субконто, которые возможны на данных счетах, мы настроили ПКО: Номенклатура, Склады, Контрагенты и Договоры. Также мы настроили ПКО для справочника Организации, Валюты, а также Единицы измерения, Виды номенклатуры, Классификатор единиц измерения и прочие справочники, на которые ссылаются обязательные для заполнения реквизиты справочников, ПКО для которых мы уже настроили.

Теперь необходимо настроить ПКО непосредственно для документа Операция (бухгалтерский учет).

Нам нужно загрузить таблицу значений, значит, источника у правила не будет. Создаем правила для всех необходимых нам свойств

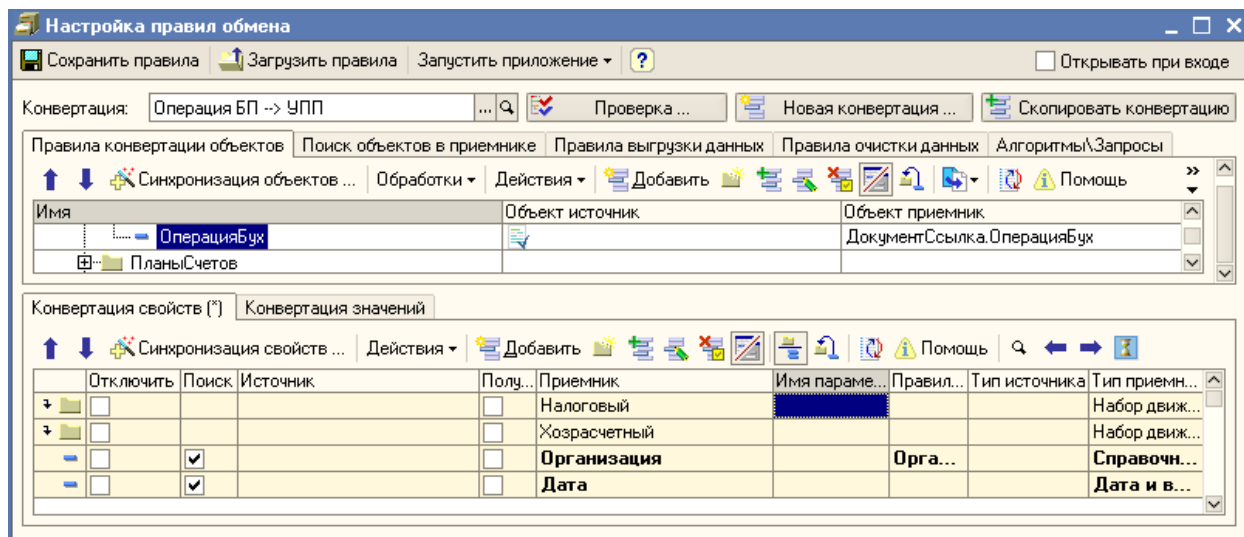


Рис. 49.

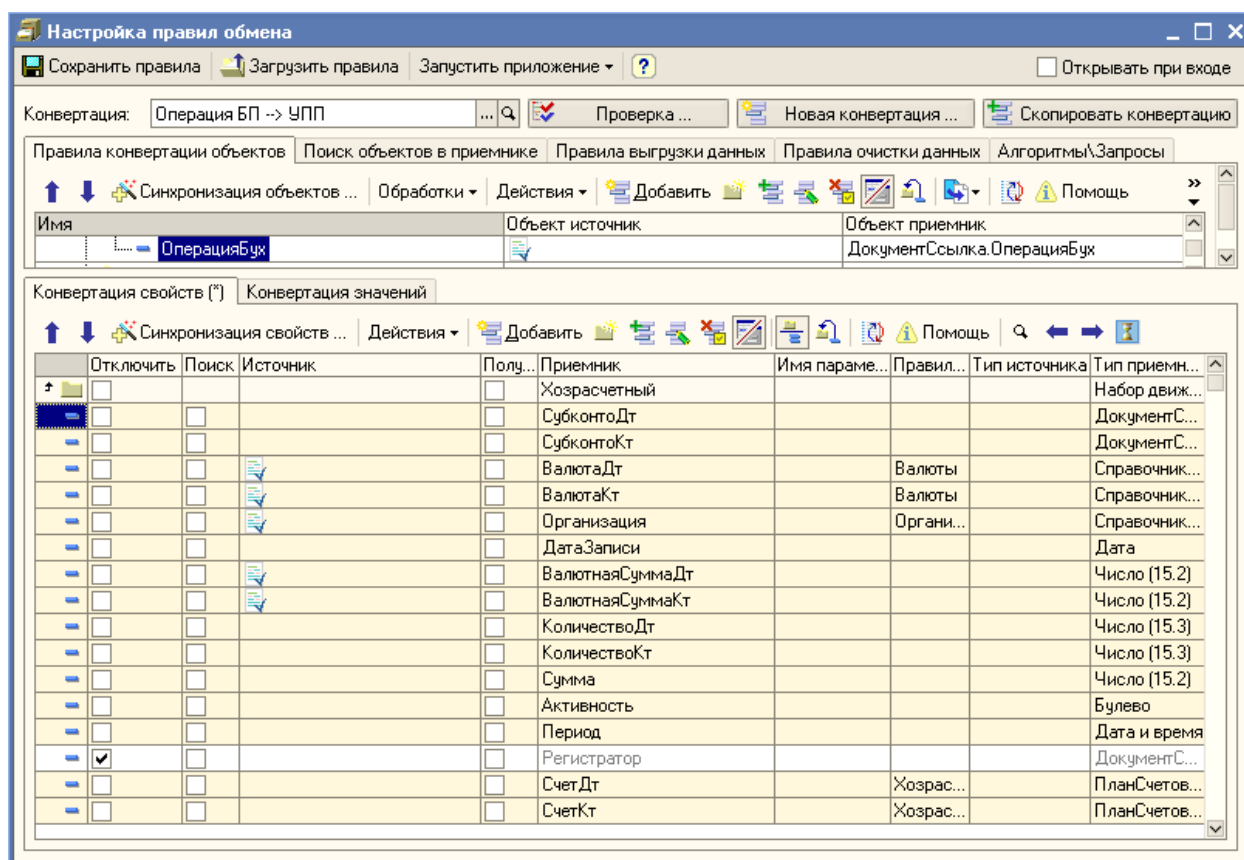


Рис. 50.

Чтобы Валюта попадала только в случае валютных счетов, пишем обработчик события перед выгрузкой ПКС ВалютаДт и ВалютаКт, а также соответствующих валютных сумм.

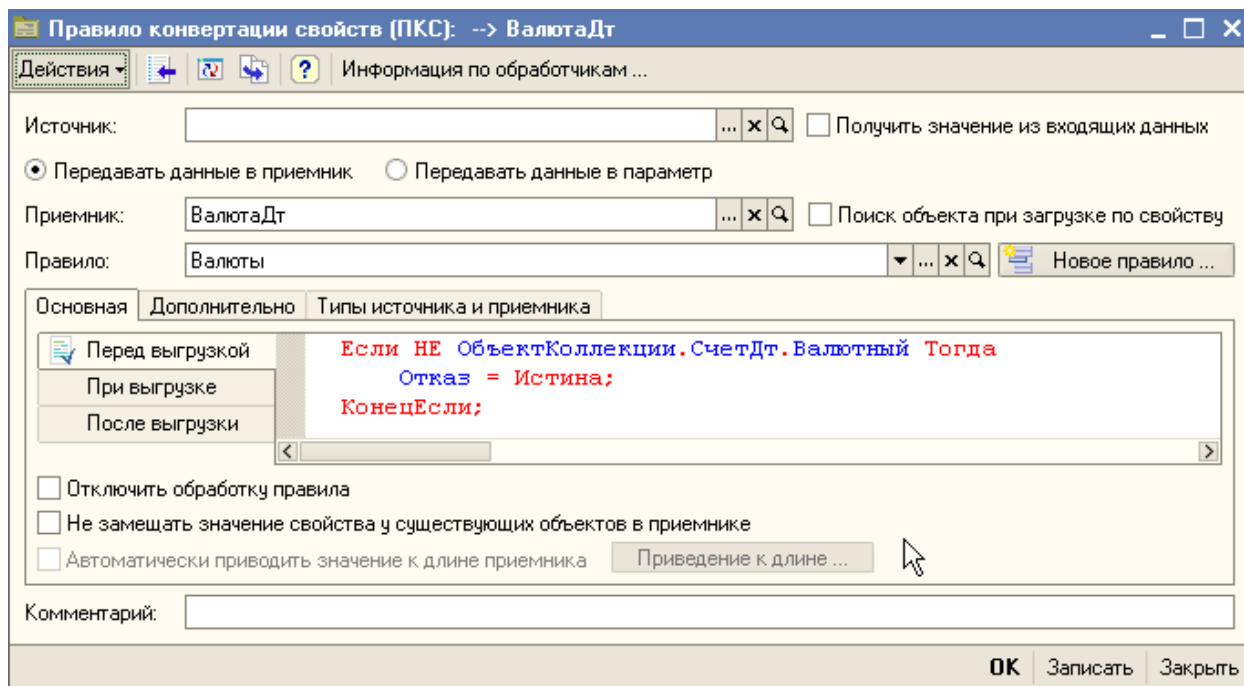


Рис. 51.

ПКС для поля Регистратор необязательно, он будет фигурировать в проводках и так.

И осталось только запустить механизм выгрузки, сформировать нужную нам таблицу и записать ее в переменную ВыборкаДанных.

Таким образом, на момент исполнения ПКО ОперацияБух в переменной ВыборкаДанных должна быть таблица. Каждая строка этой таблицы будет загружена как документ Операция. Для этого в таблице должно быть 4 колонки, названные так же, как ПКС в нашем ПКО («Налоговый», «Хозрасчетный», «Организация», «Дата»). В колонках «Хозрасчетный» и «Налоговый» должны содержаться таблицы, которые будут загружены как наборы движений регистров Хозрасчетный и Налоговый.

В событии «Перед обработкой» Правила выгрузки данных мы напишем алгоритм, формирующий две таблицы значений – в одной будут проводки по регистру Хозрасчетный, во второй – по регистру Налоговый. В таблице для регистра Хозрасчетный должны быть колонки с теми же именами, что и ПКС, входящие в группу ПКС «Хозрасчетный» - СубконтоДт, СубконтоКт, ВалютаДт, ВалютаКт, Организация, ДатаЗаписи, ВалютнаяСуммаДт, ВалютнаяСуммаКт, КоличествоДт, КоличествоКт, Сумма, Активность, Период, СчетДт, СчетКт.

СубконтоДт, СубконтоКт – это соответствия, где ключ – Вид субконто, Значение – значение субконто.

Вот вариант алгоритма заполнения таблицы для примера:

```

ТабХозр = Новый ТаблицаЗначений;
ТабХозр.Колонки.Добавить ("СубконтоДт");
ТабХозр.Колонки.Добавить ("СубконтоКт");
ТабХозр.Колонки.Добавить ("ВалютаДт");
ТабХозр.Колонки.Добавить ("ВалютаКт");
ТабХозр.Колонки.Добавить ("Организация");
ТабХозр.Колонки.Добавить ("ДатаЗаписи");
ТабХозр.Колонки.Добавить ("ВалютнаяСуммаДт");
ТабХозр.Колонки.Добавить ("ВалютнаяСуммаКт");
ТабХозр.Колонки.Добавить ("КоличествоДт");
ТабХозр.Колонки.Добавить ("КоличествоКт");
ТабХозр.Колонки.Добавить ("Сумма");
ТабХозр.Колонки.Добавить ("Активность");

```

```

ТабХозр.Колонки.Добавить ("Период") ;
ТабХозр.Колонки.Добавить ("СчетДт") ;
ТабХозр.Колонки.Добавить ("СчетКт") ;

// проводка Дт 10 Кт 60
СтрТаб = ТабХозр.Добавить () ;
СоотвСубк = Новый Соответствие ;
СоотвСубк.Вставить (ПланыВидовХарактеристик.ВидыСубконтоХозрасчетные.Номенклатура,Справочники.Номенклатура.НайтиПоКоду ("00024")) ;
СоотвСубк.Вставить (ПланыВидовХарактеристик.ВидыСубконтоХозрасчетные.Склады,Справочники.Склады.НайтиПоКоду ("00001")) ;
СтрТаб.СубконтоДт = СоотвСубк ;
СоотвСубк = Новый Соответствие ;
СоотвСубк.Вставить (ПланыВидовХарактеристик.ВидыСубконтоХозрасчетные.Контрагенты,Справочники.Контрагенты.НайтиПоКоду ("00042")) ;
СтрТаб.СубконтоКт = СоотвСубк ;
СтрТаб.Организация = "" ;
СтрТаб.ДатаЗаписи = ТекущаяДата () ;
СтрТаб.КоличествоДт = 2 ;
СтрТаб.Сумма = 2000 ;
СтрТаб.Активность = Истина ;
СтрТаб.Период = ТекущаяДата () ;
СтрТаб.СчетДт = ПланыСчетов.Хозрасчетный.СырьеИМатериалы ;
СтрТаб.СчетКт = ПланыСчетов.Хозрасчетный.РасчетыСПоставщикамиВал ;

// проводка Дт 10 Кт 60 в валюте
СтрТаб = ТабХозр.Добавить () ;
СоотвСубк = Новый Соответствие ;
СоотвСубк.Вставить (ПланыВидовХарактеристик.ВидыСубконтоХозрасчетные.Номенклатура,Справочники.Номенклатура.НайтиПоКоду ("00018")) ;
СоотвСубк.Вставить (ПланыВидовХарактеристик.ВидыСубконтоХозрасчетные.Склады,Справочники.Склады.НайтиПоКоду ("00001")) ;
СтрТаб.СубконтоДт = СоотвСубк ;
СоотвСубк = Новый Соответствие ;
СоотвСубк.Вставить (ПланыВидовХарактеристик.ВидыСубконтоХозрасчетные.Контрагенты,Справочники.Контрагенты.НайтиПоКоду ("00029")) ;
СтрТаб.СубконтоКт = СоотвСубк ;
СтрТаб.ВалютаКт = Справочники.Валюты.НайтиПоКоду ("840") ; СтрТаб.Организация = "" ;
СтрТаб.ДатаЗаписи = ТекущаяДата () ;
СтрТаб.ВалютнаяСуммаКт = 20 ;
СтрТаб.КоличествоДт = 10 ;
СтрТаб.Сумма = 3000 ;
СтрТаб.Активность = Истина ;
СтрТаб.Период = ТекущаяДата () ;
СтрТаб.СчетДт = ПланыСчетов.Хозрасчетный.СырьеИМатериалы ;
СтрТаб.СчетКт = ПланыСчетов.Хозрасчетный.РасчетыСПоставщиками ;

// здесь должен быть аналогичный алгоритм заполнения таблицы для налогового
// регистра ТабНал

ВыборкаДанных = Новый ТаблицаЗначений ;
ВыборкаДанных.Колонки.Добавить ("Хозрасчетный") ;
ВыборкаДанных.Колонки.Добавить ("Налоговый") ;
ВыборкаДанных.Колонки.Добавить ("Организация") ;
ВыборкаДанных.Колонки.Добавить ("Дата") ;
ВыборкаДанных.Колонки.Добавить ("Номер") ; СтрВыб
= ВыборкаДанных.Добавить () ; СтрВыб.Дата =
ТекущаяДата () ; СтрВыб.Хозрасчетный = ТабХозр ;
СтрВыб.Налоговый = ТабНал ; СтрВыб.Организация =
"Организация1" ; СтрВыб.Номер = "00001" ;

```

Поскольку одна и та же операция на протяжении одной выгрузки не может встретиться дважды по ссылкам, то надо поставить флажок «Не запоминать выгруженные объекты» в ПКО для Операций.

Хотя в ПКО «ОперацияБух» не заполнено поле «Источник», во всех обработчиках событий на стороне выгрузки переменная Источник не будет пустой, в ней будет строка таблицы значений ВыборкаДанныхх.

Чтобы сделать аналогичный пример для переноса из 7.7 в 8.1, необходимо переписать все обработчики событий на стороне выгрузки на языке 7.7. Поскольку в языке 7.7 нет такого типа как соответствие, то СубконтоДт и СубконтоКт здесь – таблицы значений, в которых две колонки – «Ключ» и «Значение». В колонке «Ключ» - вид субконто, в колонке «Значение»- значение субконто.

Пример: Перенос курсов валют 8 – 8

Для примера перенесем Курсы валют из Бухгалтерии Предприятия в Управление Производственным предприятием. В обеих конфигурациях регистр сведений имеет одинаковую структуру. Так же, как и для справочников, настроим все нужные ПКС – для измерений, ресурсов и реквизитов.

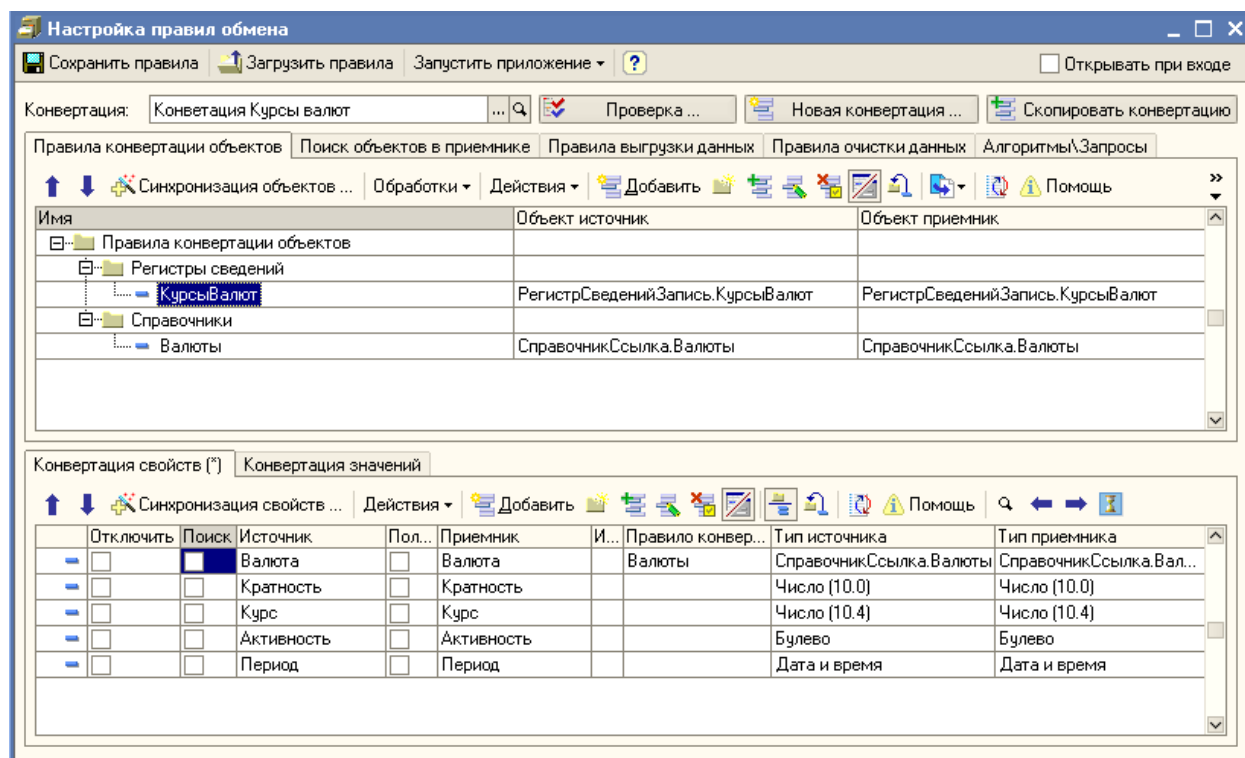


Рис. 52.

Для переноса этого регистра нам нужно одно дополнительное ПКО – для справочника Валюты. Настроим его.

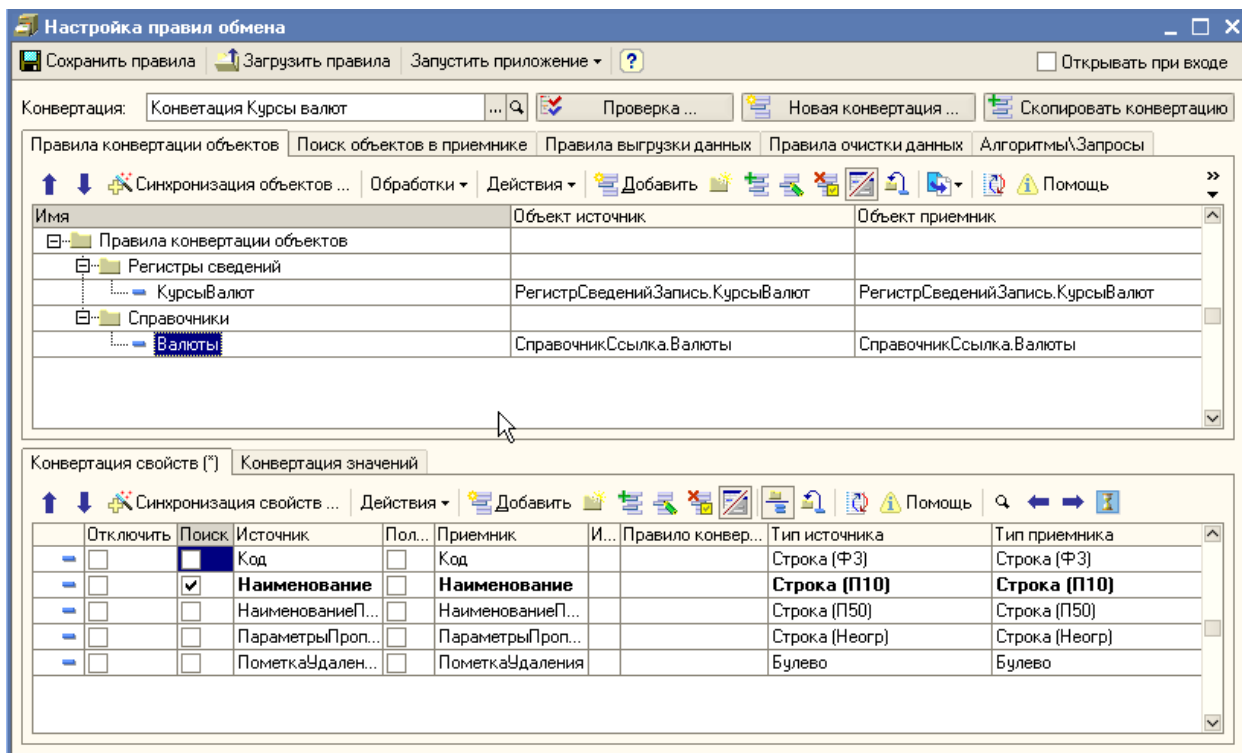


Рис. 53.

Осталось создать ПВД для регистра сведений.

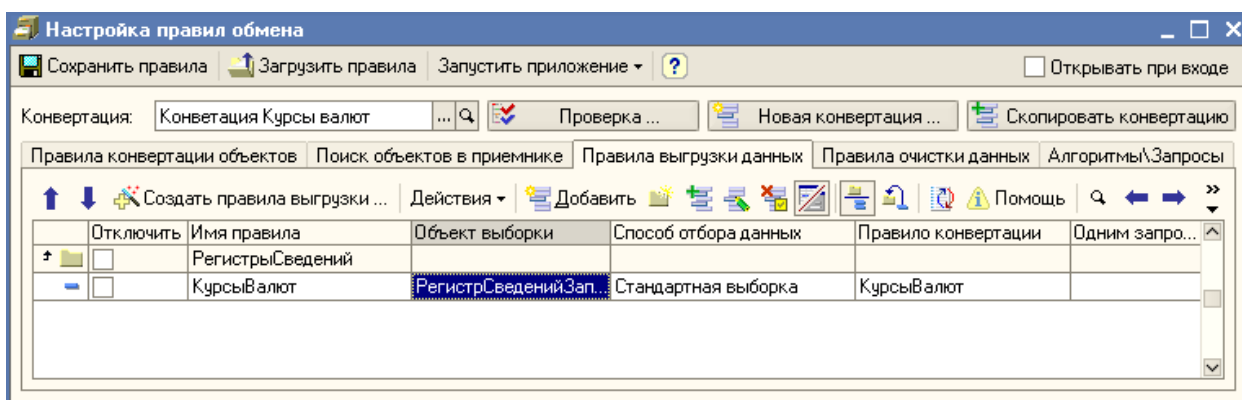


Рис. 54.

Флаги полей поиска ставить не нужно, поскольку записи регистров сведений синхронизируются только по значениям измерений.

Если перенос происходит в базу, где валюты были уже введены, то в ПКО для справочника Валюты надо либо снять флаг поиска по внутреннему идентификатору, либо установить флаг «Продолжить поиск по полям поиска, если по идентификатору объект-приемник не найден». Потому что иначе значения валют удвоятся.

Все, правила настроены, перенос будет происходить без ошибок.

Если пользователь в форме выгрузки укажет период выгрузки, то записи регистра сведений будут выбраны только за этот период.

Пример: Перенос курсов валют 7.7 – 8

По сравнению с предыдущим примером здесь добавляется одна сложность – получить курсы валют из периодического реквизита справочника Валюты в семерке и придать им форму регистра сведений «Курсы валют».

Сначала создадим ПКО «Курсы валют». Объект-источник в этом правиле выбрать нельзя, источником будет строка таблицы значений, которую потом сформируем, выбрав данные из периодического реквизита в семерке.

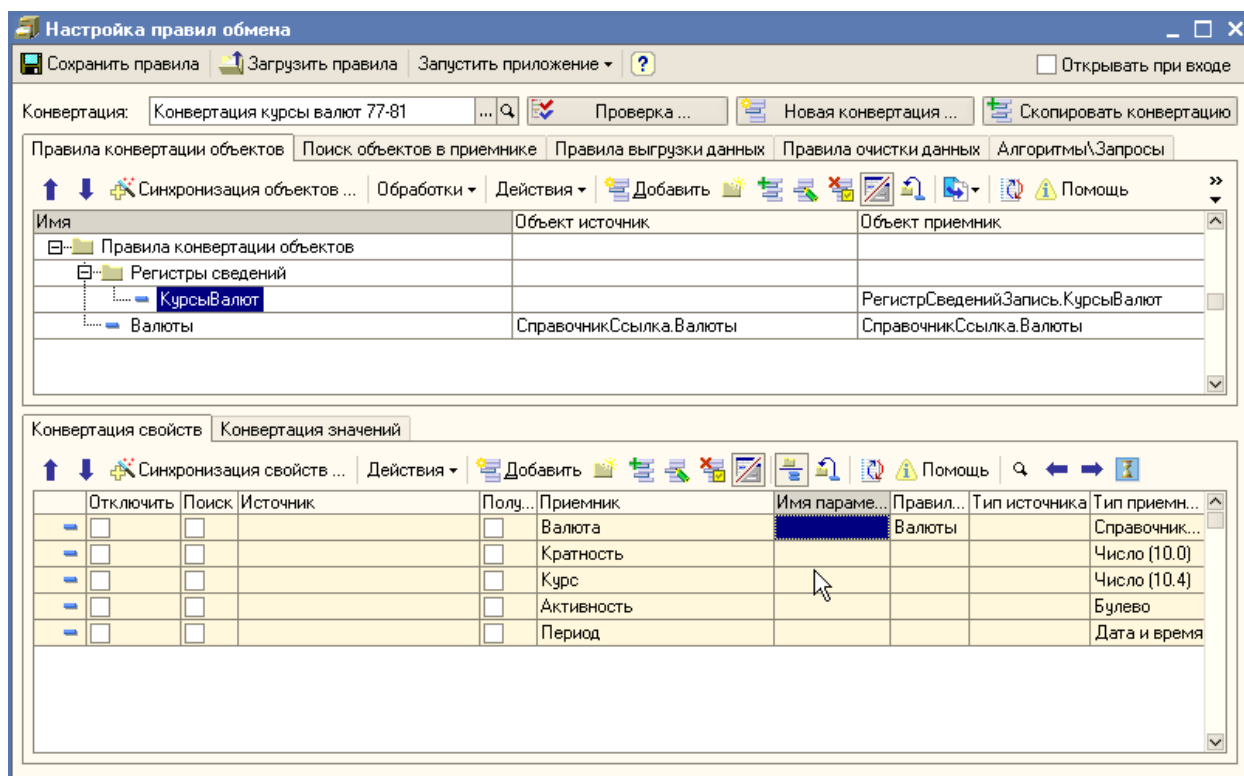


Рис. 55.

Флаги полей поиска ставить не нужно, поскольку записи регистров сведений синхронизируются только по значениям измерений.

Теперь надо сформировать ту таблицу, которую передадим в это ПКО. Проще всего для этого в ПВД в событии «Перед обработкой» написать алгоритм получения для каждой валюты всех дат изменения и значений курсов. Чтобы выбирались только те даты, которые входят в интервал выгрузки, устанавливаемый пользователем в форме выгрузки, в алгоритме можно использовать переменные ДатаНачала и ДатаОкончания. Данные необходимо записать в переменную ВыборкаДанных в виде таблицы, колонки которой называются так же, как свойства ПКО «Курсы валют».

Пример алгоритма:

```

ВыборкаДанных = СоздатьОбъект ("ТаблицаЗначений") ;
ВыборкаДанных.НоваяКолонка ("Период") ;
ВыборкаДанных.НоваяКолонка ("Валюта") ;
ВыборкаДанных.НоваяКолонка ("Курс") ;
ВыборкаДанных.НоваяКолонка ("Кратность") ;
ВыборкаДанных.НоваяКолонка ("Активность") ;
СпрВалюты = СоздатьОбъект ("Справочник.Валюты") ;
СпрВалюты.ВыбратьЭлементы () ;

Пока СпрВалюты.ПолучитьЭлемент () > 0 Цикл
    КурсВалюты = СоздатьОбъект ("Периодический") ;
    КурсВалюты.ИспользоватьОбъект ("Курс", СпрВалюты.ТекущийЭлемент () ) ;

    Если ПустоеЗначение (ДатаОкончания) = 0 Тогда
        КурсВалюты.ВыбратьЗначения (ДатаНачала, ДатаОкончания) ;
    Иначе
        КурсВалюты.ВыбратьЗначения (ДатаНачала) ;
    КонецЕсли ;

```

```

Пока КурсВалюты.ПолучитьЗначение() > 0 Цикл
    ВыборкаДанных.НоваяСтрока(); ВыборкаДанных.Период
    = КурсВалюты.ДатаЗнач; ВыборкаДанных.Валюта =
    СпрВалюты.ТекущийЭлемент(); ВыборкаДанных.Курс =
    КурсВалюты.Значение; ВыборкаДанных.Кратность =
    СпрВалюты.ТекущийЭлемент().Кратность.Получить(КурсВалюты.ДатаЗнач);
    ВыборкаДанных.Активность = 1;
КонецЦикла;
КонецЦикла;

```

Поскольку за период одной выгрузки одна запись регистра не может встретиться дважды, в ПКО «Курсы валют» следует поставить флажок «Не запоминать выгруженные объекты».

Разбор типичных примеров из демонстрационной базы

Два в один

В этом примере разобран случай, когда два разных реквизита, один из которых имеет числовой тип, а другой – тип перечисление, должны быть преобразованы в один реквизит типа перечисление в приемнике.

В элементе-источнике реквизит «Количество продаж в день» имеет целое числовое значение, а реквизит «Уровень покупателей» - тип перечисление со значениями «Эконом-класс», «Средний класс» и «Бизнес класс». В элементе-приемнике реквизит «Средние продажи» имеет тип перечисление со значениями «Эконом класс мало», «Эконом класс много», «Средний класс мало», «Средний класс много», «Бизнес класс мало» и «Бизнес класс много». Задача в том, чтобы реквизит «Средние продажи» в приемнике заполнялся в зависимости от значений двух реквизитов источника: уровень покупателей должен соответствовать уровню покупателей, а «мало» или «много» определяется количеством продаж – меньше или больше 30.

Для этого создаем правило, где источником будет любой из наших реквизитов-источников (а можно вообще без источника), а приемником – средние продажи.

В событии Перед выгрузкой этого ПКС – там, где еще не получено значение свойства-источника, пишем код:

```

Если Источник.УровеньПокупателей =
Перечисления.УровеньПокупателей.ЭкономКласс Тогда
    Если Источник.КоличествоПродажВДень > 30 Тогда
        Значение = "ЭкономКлассМало";
    Иначе
        Значение = "ЭкономКлассМного";
    КонецЕсли;
ИначеЕсли Источник.УровеньПокупателей =
Перечисления.УровеньПокупателей.СреднийКласс Тогда
    Если Источник.КоличествоПродажВДень > 30 Тогда
        Значение = "СреднийКлассМало";
    Иначе
        Значение = "СреднийКлассМного";
    КонецЕсли;
ИначеЕсли Источник.УровеньПокупателей =
Перечисления.УровеньПокупателей.БизнесКласс Тогда
    Если Источник.КоличествоПродажВДень > 30 Тогда
        Значение = "БизнесКлассМало";
    Иначе
        Значение = "БизнесКлассМного";
    КонецЕсли;
КонецЕсли;

```

В переменной Источник у нас здесь выгружаемый элемент (номенклатура) и через точку из этого элемента можем получить любой его реквизит. Записываем требуемое значение в переменную Значение – в текстовом виде, при загрузке текстовая строка (написанная без ошибок) правильно перенесется в значение перечисления.

Осталось настроить ПВД, чтобы запустить процесс выгрузки.

Два объекта в один

В этом примере рассмотрена ситуация, когда часть реквизитов элемента справочника Пользователи в базе-приемнике должны заполняться из элемента справочника Физические лица, а другая часть - из справочника Пользователи в источнике.

Для этого создадим два ПКО: первое – для конвертации справочника Пользователи в справочник Пользователи, второе – справочник Физические лица в справочник Пользователи. Для каждого ПКО настраиваем ПКС для необходимых реквизитов.

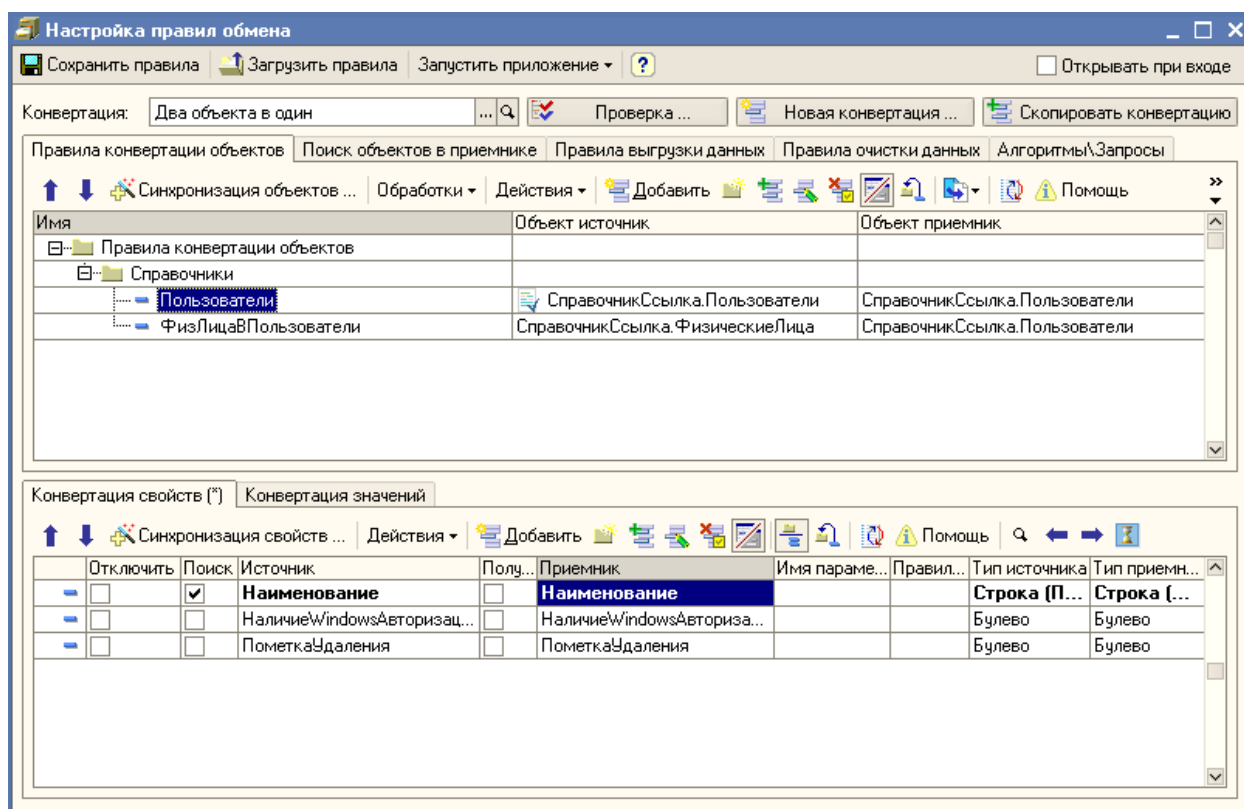


Рис. 56.

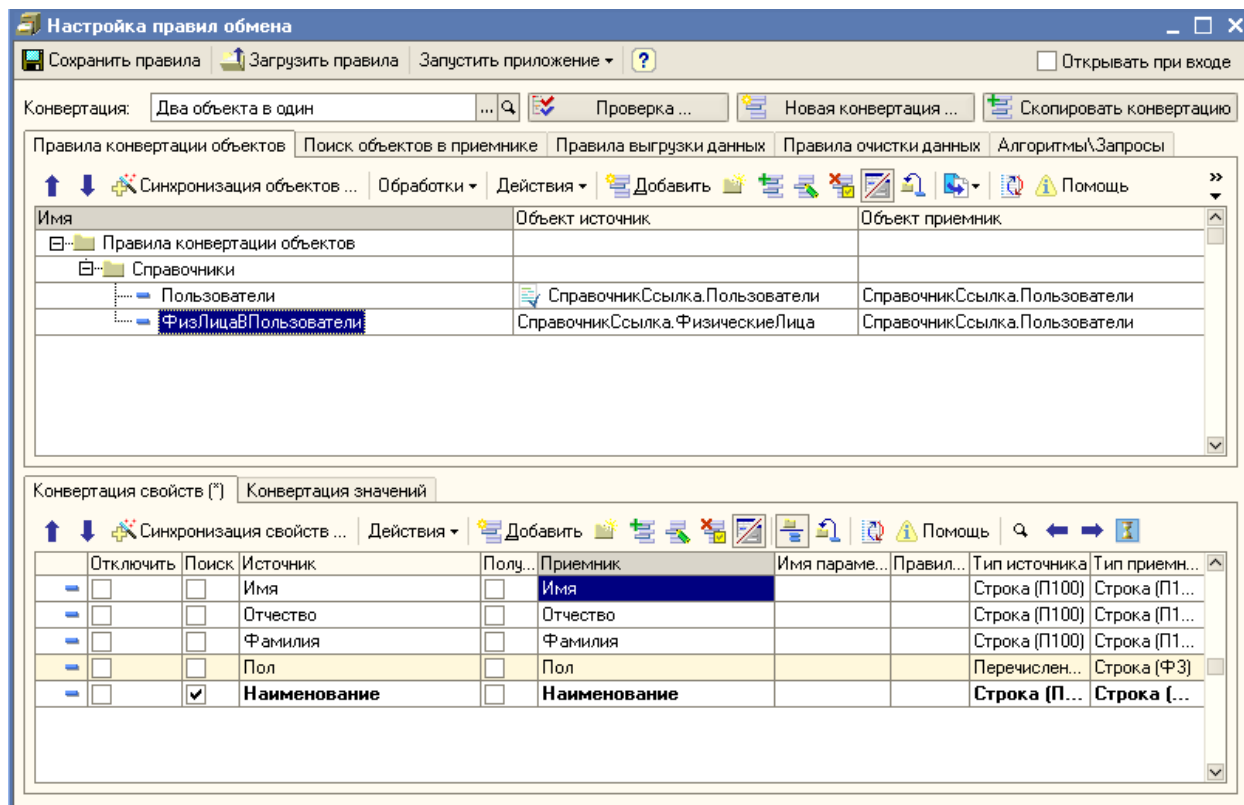


Рис. 57.

Теперь необходимо сделать так, чтобы в базе-приемнике выгрузка двух разных ПКО происходила не в два разных, а в один и тот же элемент справочника Пользователи. В нашей базе-источнике элементы справочников Пользователи и ФизЛица, которые надо совместить в один, имеют одинаковое наименование. Поэтому в базе-приемнике необходимо сделать поиск именно по наименованию в обоих ПКО. Для этого обязательно надо снять флаг поиска по внутреннему идентификатору.

В принципе, можно создать ПВД для обоих ПКО, и они будут работать правильно, но в этом случае могут быть созданы как элементы, заполненные только из элемента Пользователи, так и только из элемента Физические лица. Чтобы то ФизЛицо, для которого не создано Пользователя, не переносилось в базу-приемник как Пользователь, создадим ПВД только для ПКО Пользователи. В этом ПКО в событии после выгрузки, когда уже узел с этим элементом записан в файл данных, найдем соответствующий элемент из справочника ФизЛица и передадим управление во второе созданное ПКО:

```

СсылкаНаФизЛиц =
Справочники.ФизическиеЛица.НайтиПоНаименованию(Источник.Наименование);
Если НЕ СсылкаНаФизЛиц.Пустая() Тогда
    ВыгрузитьПоПравилу(СсылкаНаФизЛиц,
        Неопределено,
        Неопределено,
        Неопределено,
        "ФизЛицаВПользователи",
        ,
        ,
        );
КонецЕсли;

```

Здесь для передачи управления в другое ПКО используется функция ВыгрузитьПоПравилу().

(Про использование этой функции см. раздел **Функция ВыгрузитьПоПравилу()**)

Один объект в два

Здесь разобрана ситуация, когда один элемент справочника Физические Лица должен быть выгружен в элемент справочника Пользователи, а если это физическое лицо является сотрудником, то и в элемент справочника Сотрудники.

Для этого необходимо создать два ПКО и настроить для них ПКС:

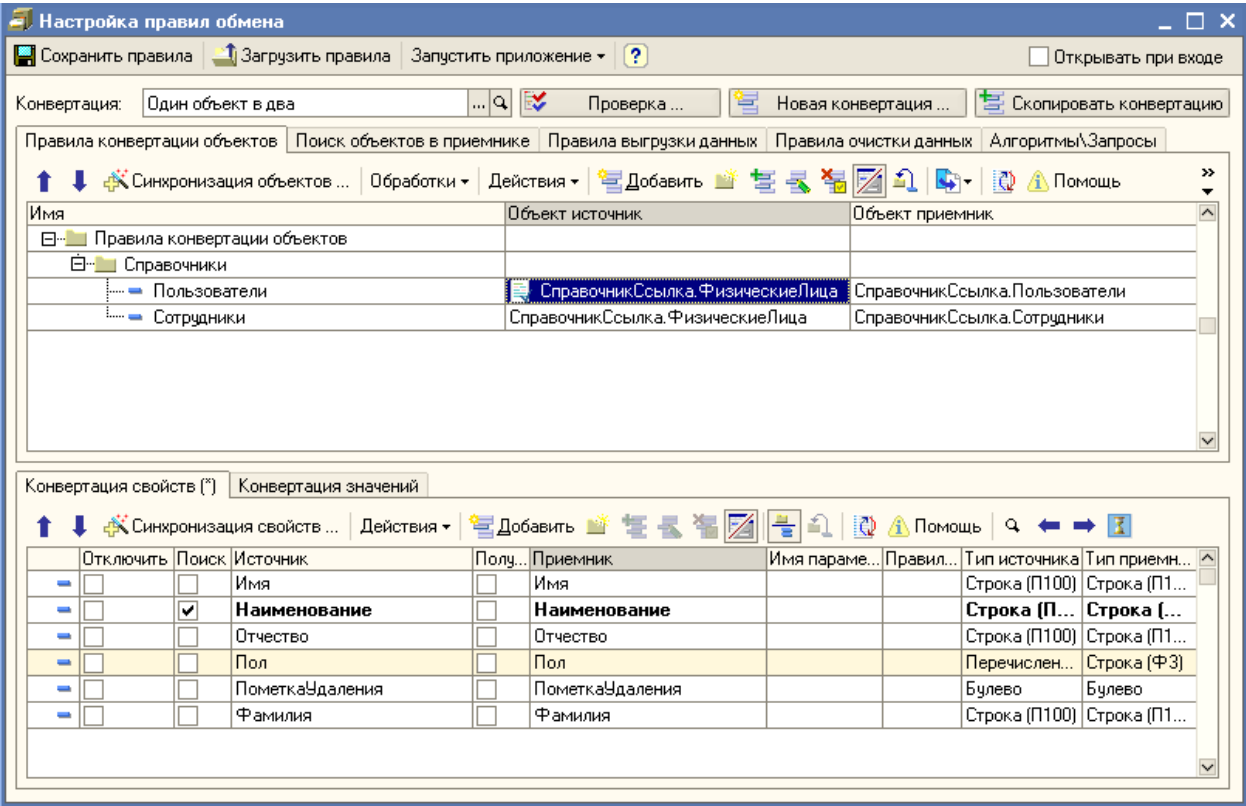


Рис. 58.

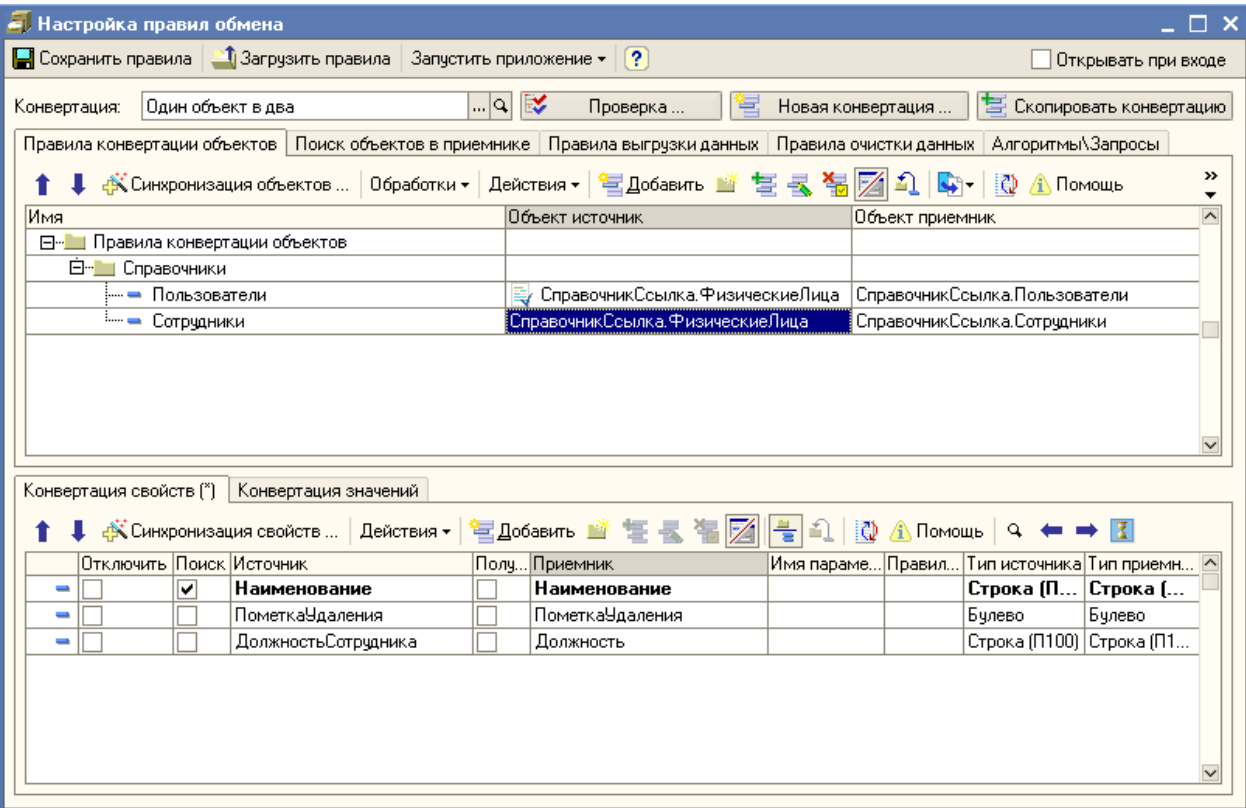


Рис. 59.

В этих ПКО обязательно снять флаги поиска по уникальному идентификатору и установить флаги поиска у нужных ПКС.

Для обоих ПКО можно было бы настроить отдельные ПВД, но это затормозит процесс выгрузки, так как одну и ту же выборку придется получать и перебирать дважды. Чтобы оптимизировать его, создаем ПВД только для ПКО Пользователи, а в этом ПКО в событии перед выгрузкой пишем алгоритм, передающий тот же элемент в другое ПКО.

```
Если Источник.ЭтоСотрудникФирмы Тогда
    ВыгрузитьПоПравилу(Источник,
        Неопределено,
        Неопределено,
        Неопределено,
        "Сотрудники",
        ,
        ,
        );
КонецЕсли;
```

Про использование функции `ВыгрузитьПоПравилу()` см. раздел **Функция `ВыгрузитьПоПравилу()`**.

Один реквизит в два

В примере рассмотрена ситуация, когда в базе-источнике у элемента справочника Номенклатура есть реквизит «Дополнительная о товаре» типа справочник «Дополнительная информация о номенклатуре» (Использование: Для элемента). Этот справочник имеет реквизиты «Вес товара» и «Описание товара». А в базе-приемнике у номенклатуры нет реквизита «Дополнительная о товаре», а есть непосредственно реквизиты «Вес» и «Описание товара». Необходимо из реквизита «Дополнительная о товаре» заполнить реквизиты «Вес» и «Описание товара».

Для этого создаем два ПКС:

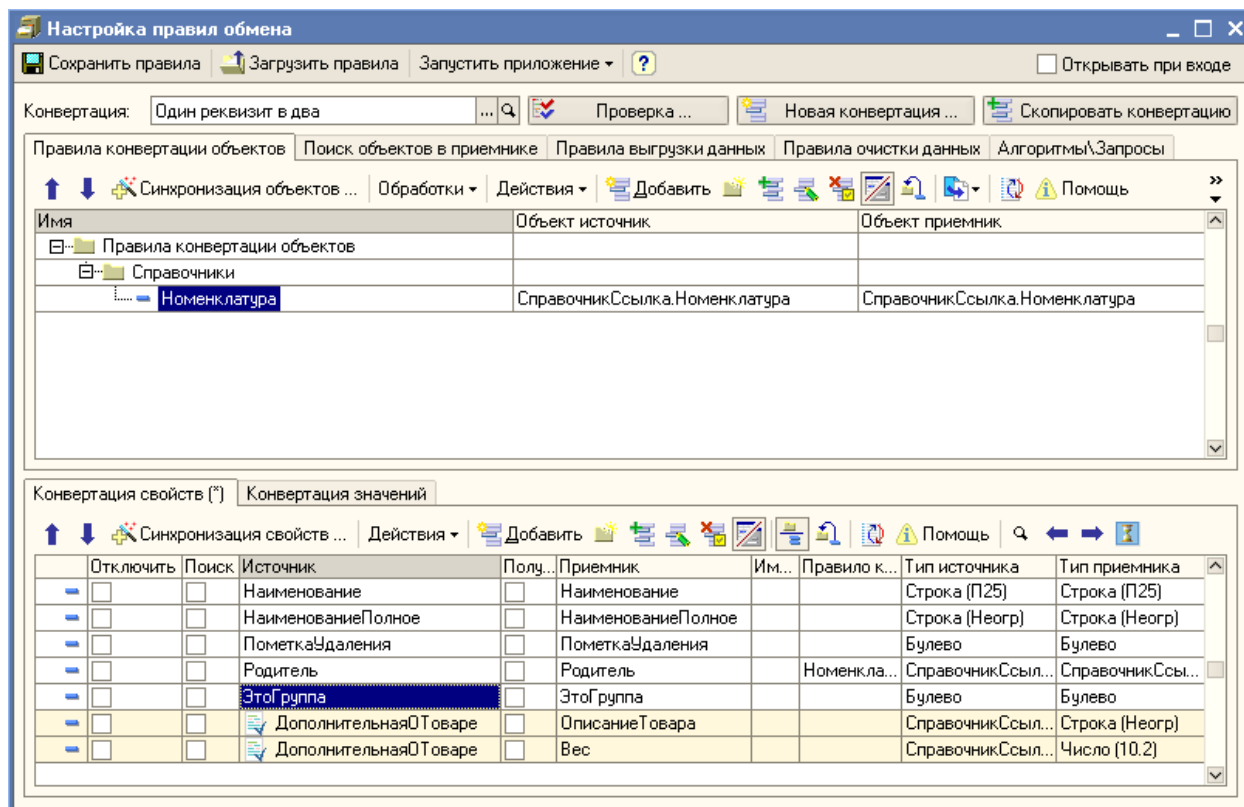


Рис. 60.

Теперь в каждом из этих ПКС необходимо получить из реквизита «Дополнительная о товаре» нужную нам информацию и записать ее в переменную Значение. Причем необходимо учесть, что этот реквизит может быть и не заполнен. Лучше всего это сделать в событии перед выгрузкой - до того, как в переменную Значение будет получена ссылка на элемент справочника «Дополнительная информация о номенклатуре» - для ускорения процесса.

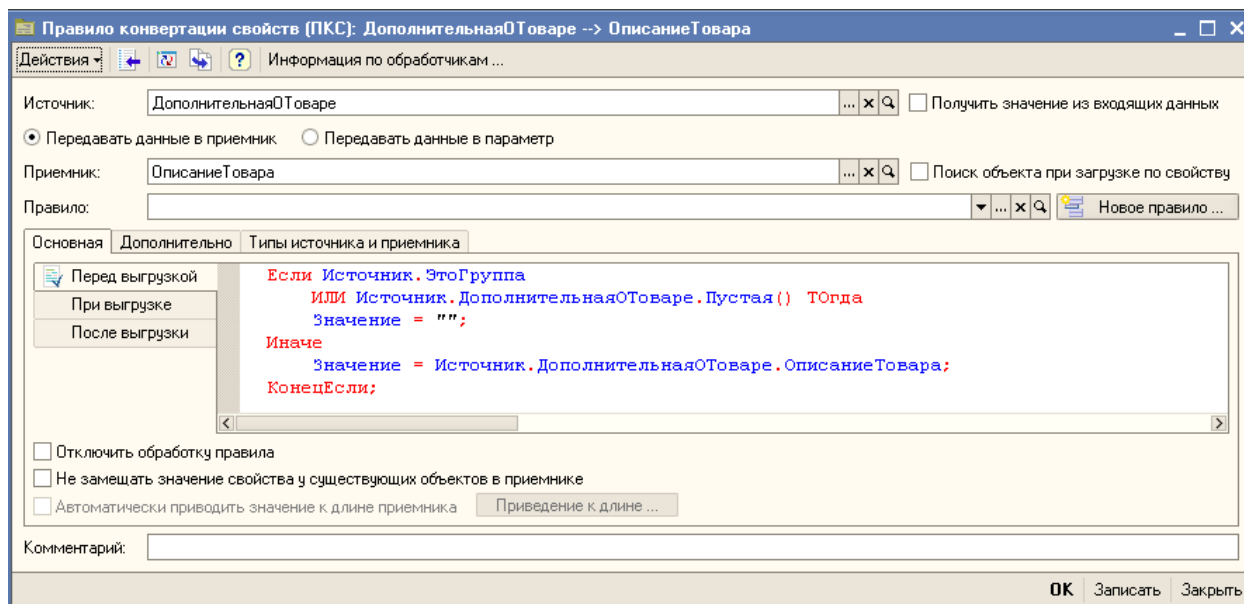


Рис. 61.

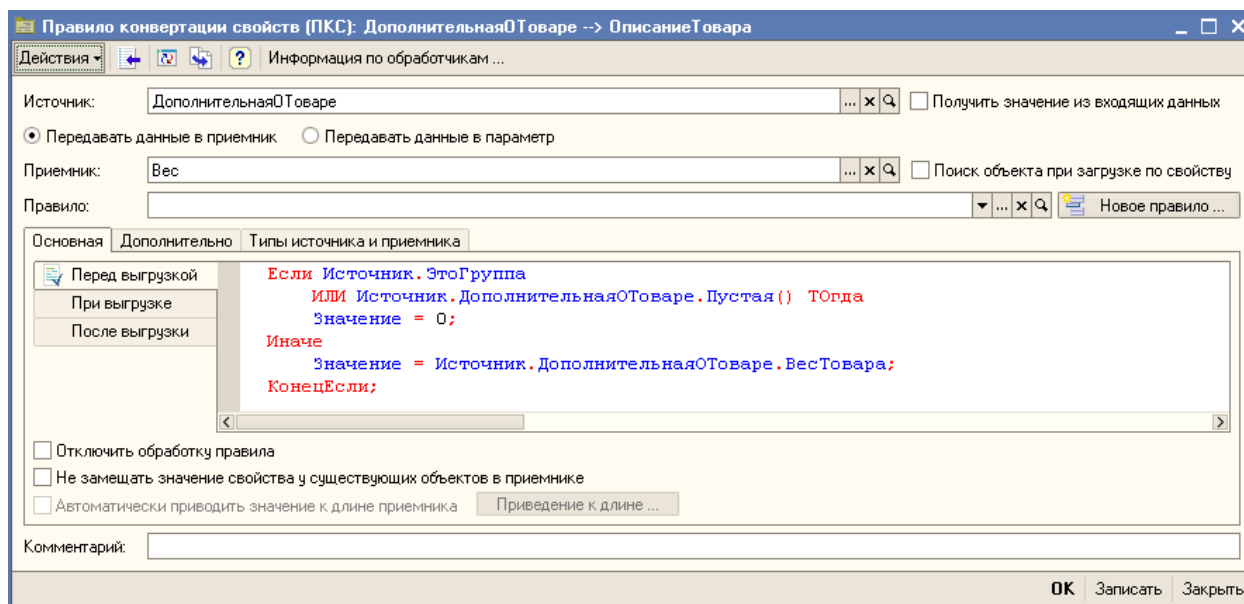


Рис. 62.

В данных ПКС в переменной Источник находится ссылка на элемент справочника Номенклатура.

Перечисление в справочник (на этапе выгрузки)

Решается следующая задача: в конфигурации-источнике у Номенклатуры есть реквизит «Ставки НДС» – типа перечисление «Ставки НДС» со значениями «НДС10», «НДС18» и «НДС20». В конфигурации-приемнике у Номенклатуры тоже есть реквизит «Ставки НДС», но типа справочник «Ставки НДС». Необходимо перенести значение перечисления в справочник, причем так, чтобы в справочнике происходила

синхронизация значений – то есть каждый элемент перечисления переносился в один и тот же элемент справочника, а не в новый элемент при каждой загрузке.

В справочнике «Ставки НДС» есть два predetermined значения, соответствующих значениям перечисления. Соответствие этих значений настраиваем на закладке «Конвертация значений». Однако у перечисления есть и такие значения, аналогов которым нет среди predetermined элементов справочника. Для них надо настраивать ПКС на закладке «Конвертация свойств».

Поскольку реквизит-приемник имеет ссылочный тип, то прямо в ПКС преобразовать значение не получится, и необходимо создать ПКО «Ставки НДС».

В справочнике «Ставки НДС» необходимо заполнить код, наименование и реквизит «Процент ставки» типа число.

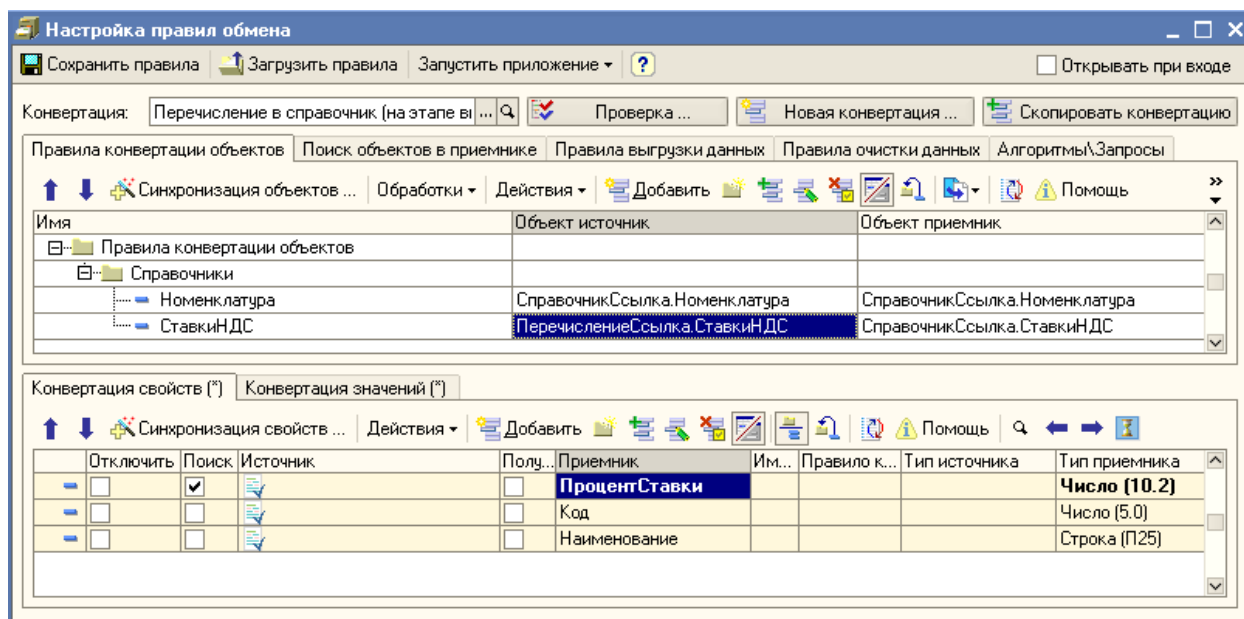


Рис. 63.

У перечисления никаких свойств нет, поэтому придется нужные значения получать в обработчике Перед выгрузкой из переменной Источник, где находится элемент перечисления.

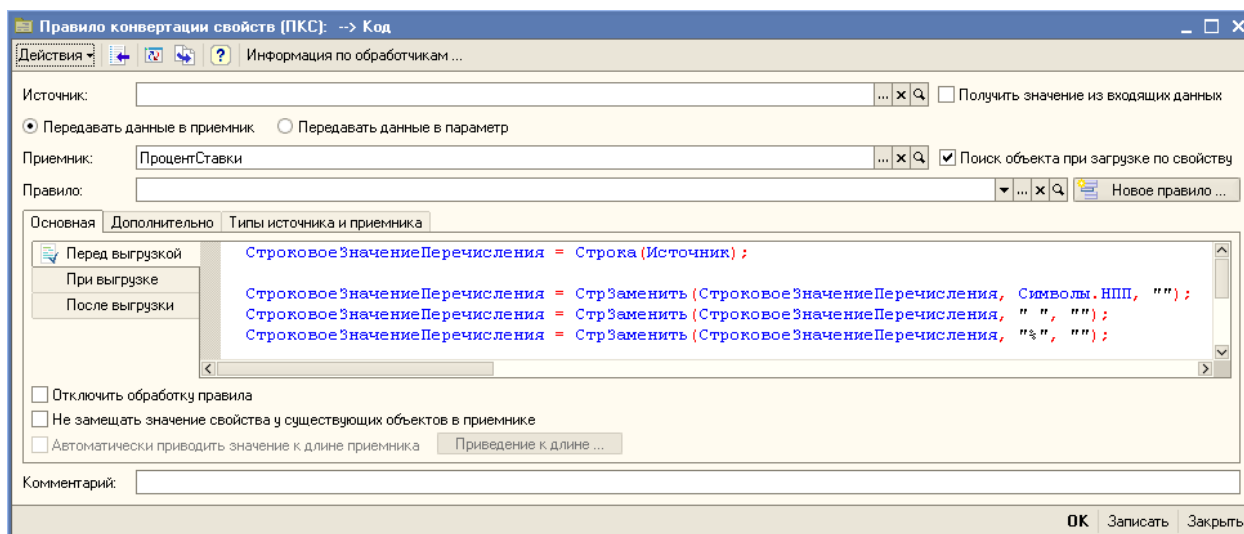


Рис. 64.

Здесь Символы.НПП – это служебный символ «Неразрывный пробел».

Вообще говоря, в значениях этого перечисления нет знака «%», а зато есть буквы «НДС», которые в данном обработчике почему-то не принимаются во внимание.

Реквизит-приемник имеет тип число, поэтому лучше всего явно преобразовать имя элемента перечисления в число, то есть заменить буквы «НДС» на пустую строку и применить функцию Число().

В результате работы данного обработчика каждый элемент перечисления однозначно преобразуется в уникальное значение, поэтому по данному свойству можно проводить синхронизацию справочника «Ставки НДС».

К коду в данном случае есть только требование уникальности, поэтому его заполняем аналогично реквизиту «Процент ставки», а к наименованию никаких требований нет, его заполняем произвольно. В данном случае – просто преобразуя имя элемента перечисления в строку.

Перечисление в справочник (на этапе загрузки)

Задача та же, что и в предыдущем примере: конвертация перечисления «Ставки НДС» в справочник «Ставки НДС».

Решение задачи выполнено путем формирования узлов и загрузки данных практически «вручную».

Здесь создается ПКО, где в первую очередь настраивается соответствие значений перечисления и предопределенных значений справочника. Но никаких ПКС в этом правиле не создается. Для переноса остальных элементов перечисления в созданном ПКО в обработчике события после выгрузки вручную создается узел, куда и записывается необходимая информация о будущем элементе справочника.

Поскольку это событие После выгрузки, то к моменту, когда выполняется обработчик, уже сформирован пустой узел ссылки на объект «Ставки НДС», и узел-Приемник, в который добавлен этот узел ссылки. Других подчиненных узлов в нем нет, поскольку никаких ПКС в нашем ПКО не настроено. Узел-Премник уже полностью сформирован, но еще не записан в файл данных.

В обработчике сначала из элемента перечисления получают процент ставки, наименование и код элемента справочника, потом создается узел, в который эти данные записываются как атрибуты, а потом он добавляется как дочерний в Приемник (узел объекта «Ставки НДС»).

СтроковоеЗначениеПеречисления = Строка (Источник) ;

СтроковоеЗначениеПеречисления = СтрЗаменить (СтроковоеЗначениеПеречисления, Символы.НПП, "");

СтроковоеЗначениеПеречисления = СтрЗаменить (СтроковоеЗначениеПеречисления, " ", "");

СтроковоеЗначениеПеречисления = СтрЗаменить (СтроковоеЗначениеПеречисления, "%", "");

ПозицияНДС = Найти (СтроковоеЗначениеПеречисления, "НДС") ;

ЗначениеСтавки = Число (Сред (СтроковоеЗначениеПеречисления, ПозицияНДС + 3)) ;

УзелСтрока = СоздатьУзел ("ДопИнформация") ;

УстановитьАтрибут (УзелСтрока, "Код", ЗначениеСтавки) ;

УстановитьАтрибут (УзелСтрока, "Наименование", Строка (Источник)) ;

УстановитьАтрибут (УзелСтрока, "Процент", ЗначениеСтавки) ;

УзелСтрока.ЗаписатьКонецЭлемента () ;

УзелСтрока = УзелСтрока.Закрыть () ;

ДобавитьПодчиненный (Приемник, УзелСтрока) ;

После выгрузки получается вот такая картина:

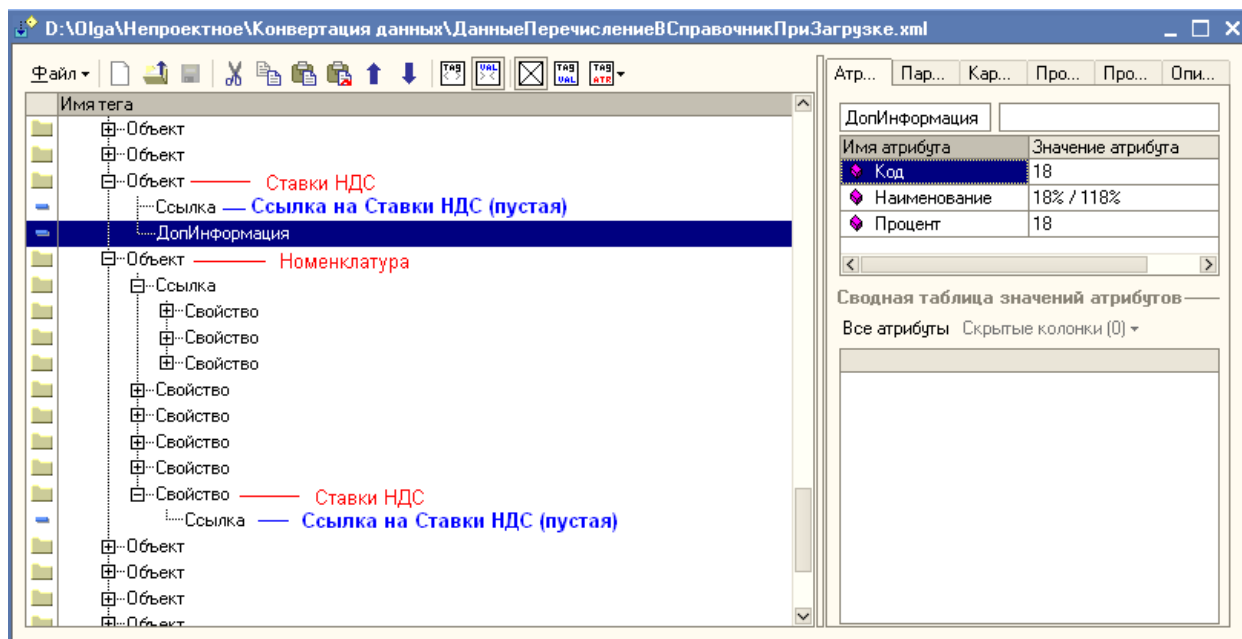


Рис. 65.

Теперь, чтобы узел ДопИнформацию не вызвал ошибку, придется перехватить управление процессом в событии Перед загрузкой и написать там альтернативный механизм чтения этого объекта. Узел объекта «Ставки НДС» в формате xml выглядит так:

```
<Объект Нпп="50" Тип="СправочникСсылка.СтавкиНДС" ИмяПравила="СтавкиНДС">
    <Ссылка Нпп="50" />
    <ДопИнформация Код="18" Наименование="18% / 118%" Процент="18" />
</Объект>
```

То есть при чтении этого объекта будут попадаться типы тегов: «Объект» - открывающий тег, «Ссылка», «ДопИнформация», «Объект» - закрывающий тег. Но когда управление передается в ПКО, открывающий тег узла «Объект» уже прочитан, так что его при чтении не встретится. Закрывающий тег узла «Объект» читать не следует, потому что его более грамотно обработает собственный код механизма выгрузки (Когда встречается тег конца объекта, происходит запись загруженного объекта и т.д.). То есть нам необходимо обработать теги «Ссылка» и «ДопИнформация», и когда прочтем тег «ДопИнформация», прервать чтение файла и вернуть управление в собственный код обработки.

В примере обработчик события «Перед загрузкой» выглядит следующим образом:

```
Код = "";
Наименование = "";
Процент = "";

НомерПоПорядку = 0;

Пока ФайлОбмена.Прочитать () Цикл

    ИмяУзла = ФайлОбмена.ЛокальноеИмя;
    ТипУзла = ФайлОбмена.ТипУзла;

    Если (ИмяУзла = "ДопИнформация") И (ТипУзла =
        одТипУзлаXML_КонецЭлемента) Тогда

        СсылкаНаОбъект =
            Справочники.СтавкиНДС.НайтиПоРеквизиту ( "ПроцентСтавки",
                Число (Процент) );
```

```

Если СсылкаНаОбъект.Пустая() Тогда

    Объект = Справочники.СтавкиНДС.СоздатьЭлемент();

Иначе

    Объект = СсылкаНаОбъект.ПолучитьОбъект();

КонецЕсли;

Объект.Код = Код;
Объект.Наименование = Наименование;
Объект.ПроцентСтавки = Процент;

Объект.Записать();

Если НомерПоПорядку <> 0 Тогда
    ЗагруженныеОбъекты[НомерПоПорядку] = Объект.Ссылка;
КонецЕсли;

Прервать;

ИначеЕсли ИмяУзла = "ДопИнформация" Тогда

    Код = одАтрибут(ФайлОбмена, одТипСтрока, "Код");
    Наименование = одАтрибут(ФайлОбмена, одТипСтрока, "Наименование");
    Процент = одАтрибут(ФайлОбмена, одТипСтрока, "Процент");

ИначеЕсли ИмяУзла = "Ссылка" И (ТипУзла =
одТипУзлаXML_НачалоЭлемента) Тогда

    НомерПоПорядку = одАтрибут(ФайлОбмена, одТипЧисло, "Нпп");

КонецЕсли;

КонецЦикла;

```

Мне представляется, что тут есть ошибка, так как предполагается, что сформированы открывающий и закрывающий теги «ДопИнформация». На мой взгляд, этот обработчик надо изменить так:

```

Код = "";
Наименование = "";
Процент = "";

НомерПоПорядку = 0;

Пока ФайлОбмена.Прочитать() Цикл

    ИмяУзла = ФайлОбмена.ЛокальноеИмя;
    ТипУзла = ФайлОбмена.ТипУзла;

    Если (ИмяУзла = "ДопИнформация") Тогда

        Код = одАтрибут(ФайлОбмена, одТипСтрока, "Код");
        Наименование = одАтрибут(ФайлОбмена, одТипСтрока, "Наименование");
        Процент = одАтрибут(ФайлОбмена, одТипСтрока, "Процент");

        СсылкаНаОбъект =
        Справочники.СтавкиНДС.НайтиПоРеквизиту(
        "ПроцентСтавки", Число(Процент));

        Если СсылкаНаОбъект.Пустая() Тогда

            Объект = Справочники.СтавкиНДС.СоздатьЭлемент();

```

```

        Иначе

            Объект = СсылкаНаОбъект.ПолучитьОбъект ( ) ;

        КонецЕсли;

        Объект.Код = Код;
        Объект.Наименование = Наименование;
        Объект.ПроцентСтавки = Процент;

        Объект.Записать ( ) ;

        Если НомерПоПорядку <> 0 Тогда
            ЗагруженныеОбъекты[НомерПоПорядку] = Объект.Ссылка;
        КонецЕсли;

        Прервать;

        ИначеЕсли ИмяУзла = "Ссылка" Тогда

            НомерПоПорядку = одАтрибут(ФайлОбмена, одТипЧисло, "Нпп");

        КонецЕсли;

    КонецЦикла;

```

Метод Прочитать() выбирает из файла данных следующий тег со всеми его атрибутами. Получить атрибуты текущего узла можно с помощью функции одАтрибут(ФайлОбмена, Тип, ИмяАтрибута), где параметр ФайлОбмена – это читаемый файл данных, Тип – один из примитивных типов, к которому будет приведено полученное значение атрибута, ИмяАтрибута – это имя атрибута, который необходимо получить.

Когда встречается тег «Ссылка» происходит получение его единственного атрибута «Нпп», а при обработке тега «ДопИнформация» происходит добавление элемента в структуру ЗагруженныеОбъекты.

ЗагруженныеОбъекты – это структура, где сохраняются ссылки на объекты, которые были уже в данном сеансе загрузки получены из файла данных и загружены в информационную базу. Причем не важно, были эти объекты созданы, или найдены в базе. В теге «Ссылка» каждого объекта в файле данных есть атрибут «Нпп», который служит ключом для записи ссылки в структуру ЗагруженныеОбъекты. Когда в каком-либо свойстве или субконто в файле данных встретится ссылка на уже загруженный объект, из тега этой ссылки опять будет получен атрибут «Нпп», по нему в структуре ЗагруженныеОбъекты будет найдена ссылка на нужный объект, и эта ссылка будет записана в свойство или субконто.

Число в перечисление

Здесь рассмотрен пример, когда числовой реквизит номенклатуры «Скидка для дистрибьютора» должен в зависимости от принимаемого значения конвертироваться в один из трех элементов перечисления «Скидки для дистрибьютора» в базе-приемнике. Для этого в ПКО для номенклатуры настраиваем ПКС для реквизита «Скидки для дистрибьютора». ПКО для этого ПКС создавать не нужно, поскольку прямого соответствия значения настроить нельзя. В обработчике события «Перед выгрузкой» пишем код:

```

Если Источник.СкидкаДляДистрибьютора = 10 Тогда

    Значение = "Скидка10";

```

ИначеЕсли Источник.СкидкаДляДистрибьютора = 20 Тогда

Значение = "Скидка20";

ИначеЕсли Источник.СкидкаДляДистрибьютора = 30 Тогда

Значение = "Скидка30";

КонецЕсли;

Минус данного решения в том, что если в базе-источнике скидка для дистрибьютора какой-то номенклатуры принимает значение, не равное 10, 20 или 30, то в базе-приемнике у этой номенклатуры скидка для дистрибьютора заполнена не будет.

Особенности работы с платформой 7.7

Загрузка или выгрузка из базы на платформе 7.7 имеет ряд отличий.

Из-за невозможности в семерке выполнить программный код, записанный в текстовом виде в xml, обработчики событий должны быть добавлены в текст обработки выгрузки или загрузки. Поэтому, если в процессе настройки правил обмена данными были изменены какие-либо обработчики событий, после настройки правил при сохранении правил необходимо сохранить также текстовый файл с модулем обработки выгрузки (загрузки). На этот текст надо заменить весь модуль обработки, поставляемый вместе с конфигурацией Конвертация данных.

При необходимости можно внести изменения непосредственно в тело модуля так, чтобы они сохранялись при каждой автоматической генерации кода модуля. Для этого надо изменить шаблоны частей модуля, хранящиеся в макетах обработки «Выгрузка конвертации», встроенной в конфигурацию. Например, если вставить в шаблон новую процедуру или функцию, ее можно будет вызывать из любого обработчика событий. То же самое можно сделать и с помощью создания алгоритма, но если необходимо, например, написать две функции, вызывающие друг друга рекурсивно, то проще вставить их в шаблон модуля, чтобы самостоятельно определить параметры, передаваемые в каждую из них.

Если база-приемник или база-источник семерочная, то недоступен поиск загружаемого объекта по идентификатору.

Если выгрузка ведется из семерочной базы, нельзя использовать Ключ Выгружаемых Данных.

Если загрузка ведется в семерочную базу, недоступен обработчик события Поля Поиска в ПКО.

Все переменные, имеющие в восьмерке тип Структура (ВходящиеДанные, ИсходящиеДанные, Параметры) в семерке имеют тип СписокЗначений.

Из-за особенностей языка семерки, вместо типа Булево используется числовой тип (0-ложь, 1-истина).

Для работы с файлами xml из семерки нужна компонента v7plus.dll относительно свежей версии. (Версия 7.70.0.11 - работает). Компонента должна быть зарегистрирована и доступна с компьютера, откуда выполняется выгрузка.

Кроме этой компоненты на компьютере должен быть установлен XML-парсер Microsoft (компонента msxml.dll) не ниже третьей версии. (В последние версии Интернет Эксплорера он включен автоматически).

Формат XML. Что это такое?

Запись в файлы правил и данных. Общая схема

Формат XML представляет собой текст, структурированный тегами. Тег – это набор символов, заключенных в угловые скобки:

`<ПравилаОбмена>`

Теги бывают открывающие, закрывающие и пустые.

Открывающий тег:

`<ПравилаОбмена>`

Закрывающий тег:

`</ПравилаОбмена>`

Пустой тег:

`<ПравилаОбмена />`

Между открывающим и закрывающим тегом содержится определенный раздел, он называется узел. Пустой тег совмещает в себе открывающий и закрывающий тег, и означает, что узел этот пуст.

В открывающем и пустом теге кроме названия могут содержаться атрибуты. Атрибуты – это дополнительная информация о данном узле.

`<Свойство Поиск="true">`

`<Источник Имя="ПометкаУдаления" Вид="Свойство" Тип="Булево" />`

Здесь «Поиск», «Имя», «Вид», «Тип» - это атрибуты. А «true», «ПометкаУдаления», «Свойство», «Булево» - это значения этих атрибутов. Тег «Свойство» открывающий, тег «Источник» - пустой.

Узлы могут быть многократно вложены друг в друга. Вложенный узел называется подчиненным.

С помощью этих тегов в файл xml в текстовом виде записываются сначала правила переноса, а потом данные.

Пустые правила конвертации данных выглядят следующим образом:

`<ПравилаОбмена>`

`<ВерсияФормата>2.01</ВерсияФормата>`

`<Ид>72229421-2783-4b9e-aac3-18c856cee7a1 </Ид>`

`<Наименование>Имя конвертации</Наименование>`

`<ДатаВремяСоздания>2007-12-25T19:15:29</ДатаВремяСоздания>`

`<Источник ВерсияПлатформы="8.0">БухгалтерияПредприятия</Источник>`

`<Приемник ВерсияПлатформы="8.0">БухгалтерияПредприятия</Приемник>`

`<Параметры/>`

`<Обработки/>`

`<ПравилаКонвертацииОбъектов>`

`...`

`</ПравилаКонвертацииОбъектов>`

`<ПравилаВыгрузкиДанных>`

`...`

`</ПравилаВыгрузкиДанных>`

`<ПравилаОчисткиДанных/>`

`<Алгоритмы/>`

`<Запросы/>`

`</ПравилаОбмена>`

Вот те же самые правила в виде дерева

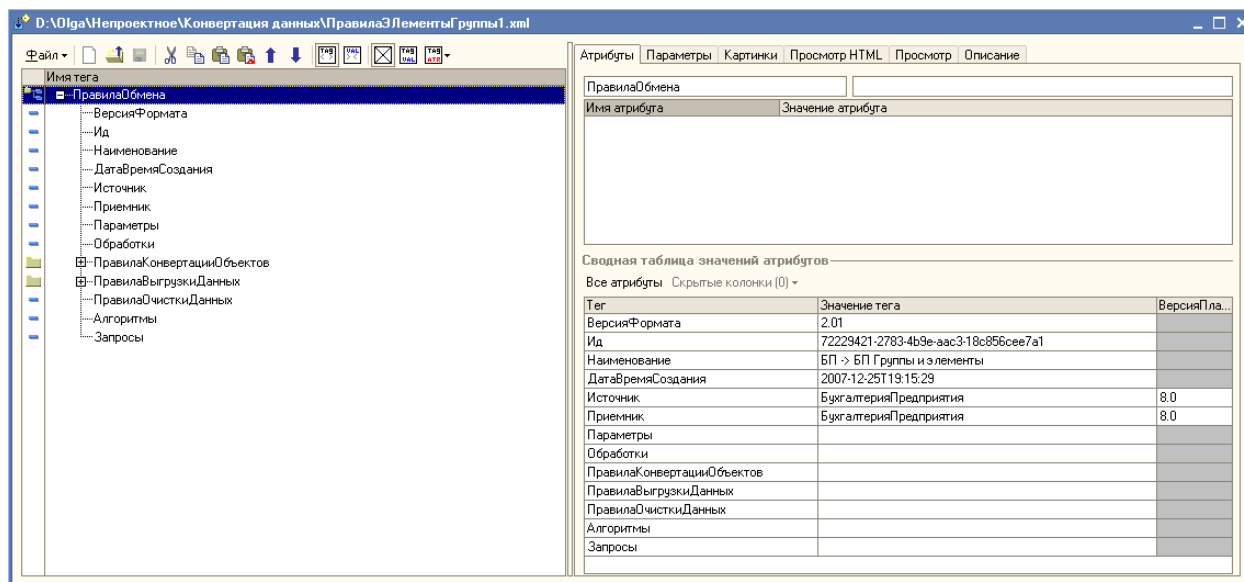


Рис. 66.

Сами ПКО, которые были настроены нами, хранятся в узле ПравилаКонвертацииОбъектов. Выглядит правило приблизительно так:

```

<Правило>
  <Код>КурсыВалют</Код>
  <Наименование>РегистрСведенийЗапись.КурсыВалют --> Валют</Наименование>
  <Порядок>50</Порядок>
  <НеЗапоминатьВыгруженные>true</НеЗапоминатьВыгруженные>
  <Источник>РегистрСведенийЗапись.КурсыВалют</Источник>
  <Приемник>РегистрСведенийЗапись.КурсыВалют</Приемник>
  <Свойства>
    <Свойство>
      <Код>1</Код>
      <Наименование>Валюта --> Валют</Наименование>
      <Порядок>50</Порядок>
      <Источник  Имя="Валюта "
                          Вид="Измерение "
                          Тип="СправочникСсылка.Валюты"/>
      <Приемник  Имя="Валюта "
                          Вид="Измерение "
                          Тип="СправочникСсылка.Валюты"/>
      <КодПравилаКонвертации>Валюты
    </КодПравилаКонвертации>
    </Свойство>
    <Свойство>
      ...
    </Свойство>
  </Свойства>
</Правило>

```

В виде дерева правило выглядит так:

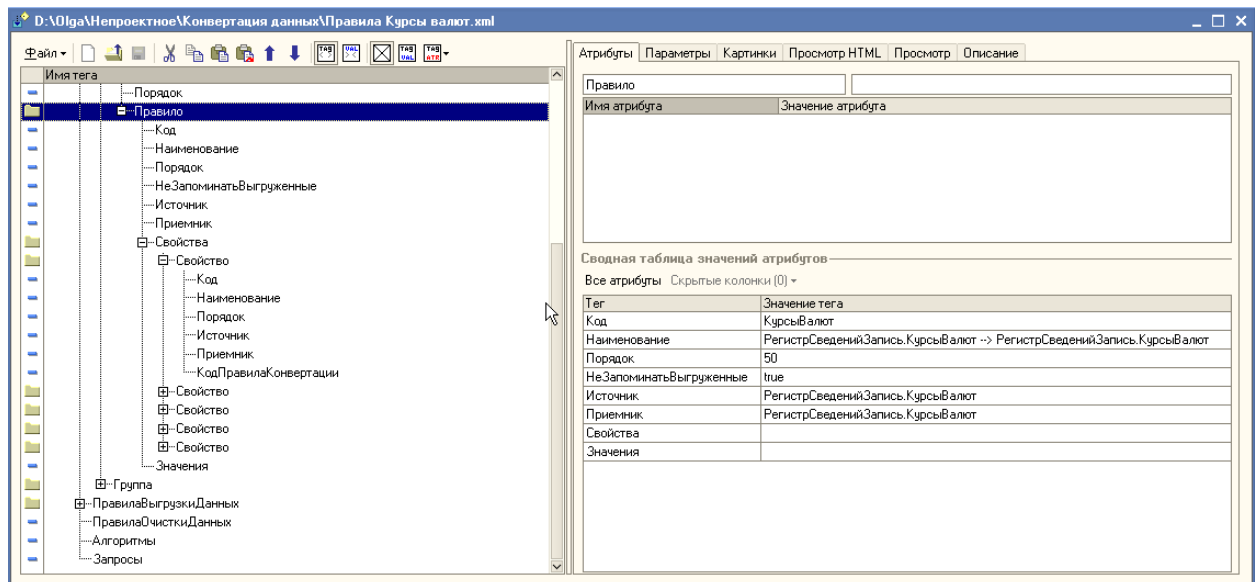


Рис. 67.

В узле Свойства хранятся ПКЗ, настроенные нами, в узле Значения – ПКЗ.

Структура файла данных выглядит приблизительно следующим образом

```
<?xml version="1.0" encoding="UTF-16"?>
<ФайлОбмена ВерсияФормата="2.0"
  ДатаВыгрузки="2007-12-28T18:05:56"
  НачалоПериодаВыгрузки="0001-01-01T00:00:00"
  ОкончаниеПериодаВыгрузки="0001-01-01T00:00:00"
  ИмяКонфигурацииИсточника="БухгалтерияПредприятия"
  ИмяКонфигурацииПриемника="УправлениеПроизводственнымПредприя
тием" ИдПравилКонвертации="efdc4359-bb70-4749-b7f1-
2fb37c8d045a" Комментарий="">
  <ПравилаОбмена>
    <ВерсияФормата>2.01</ВерсияФормата>
    <Ид>efdc4359-bb70-4749-b7f1-2fb37c8d045a</Ид>
    <Наименование>Конветация Курсы валют</Наименование>
    <ДатаВремяСоздания>2007-12-28T18:04:06</ДатаВремяСоздания>
    <Источник>БухгалтерияПредприятия</Источник>
    <Приемник>УправлениеПроизводственнымПредприятием</Приемник>
    <Параметры/>
    <Обработки/>
    <ПравилаКонвертацииОбъектов>
      <Правило>
        <Код>КурсыВалют</Код>
        <Источник>РегистрСведенийЗапись.КурсыВалют
        </Источник>
        <Приемник>РегистрСведенийЗапись.КурсыВалют
        </Приемник>
      </Правило>
      <Правило>
        ...
      </Правило>
    </ПравилаКонвертацииОбъектов>
    <ПравилаОчисткиДанных/>
    <Алгоритмы/>
    <Запросы/>
  </ПравилаОбмена>
  <Объект Нпп="2"
    Тип="СправочникСсылка.Валюты"
    ИмяПравила="Валюты">
    <Ссылка Нпп="2">
      <Свойство Имя="{УникальныйИдентификатор}"
```

```

        Тип="Строка">
        <Значение>329208a4-f656-11d8-b105-
        00055d80a2b9</Значение>
    </Свойство>
    <Свойство Имя="Наименование" Тип="Строка">
        <Значение>USD</Значение>
    </Свойство>
</Ссылка>
<Свойство Имя="Код" Тип="Строка">
    <Значение>840</Значение>
</Свойство>
<Свойство Имя="НаименованиеПолное" Тип="Строка">
    <Значение>Доллар США</Значение>
</Свойство>
</Объект>
<Объект ... >
... </Объект>
</ФайлОбмена>

```

Графическое представление:

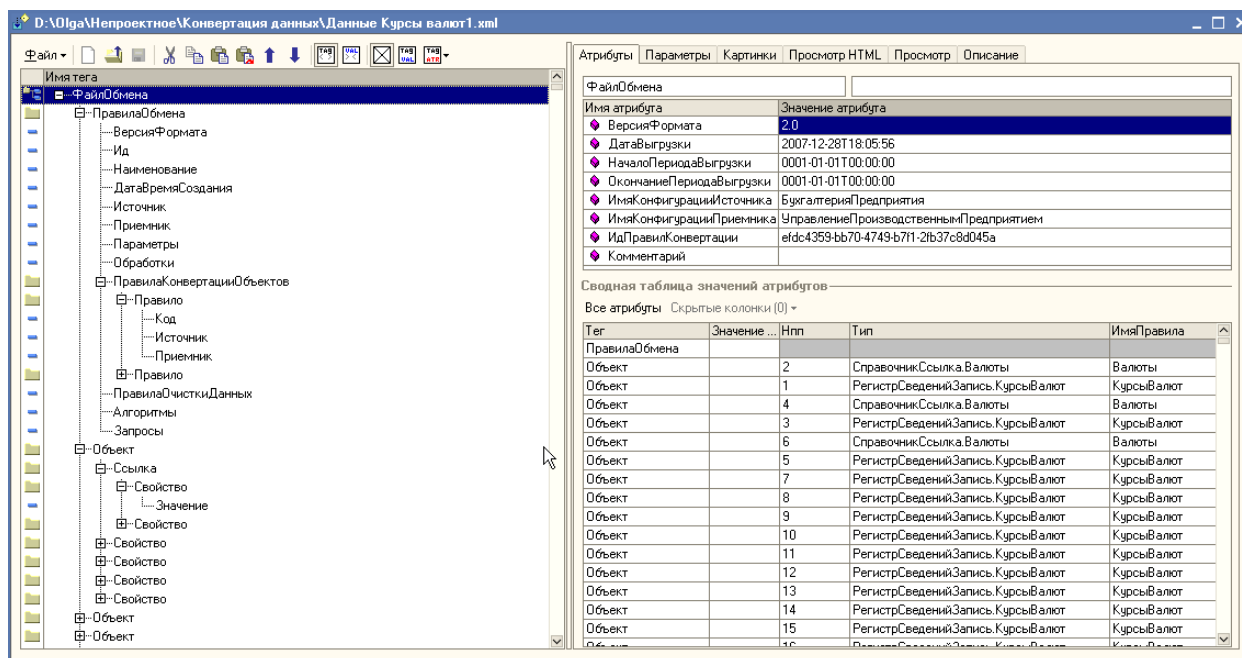


Рис. 68.

Здесь в узле ПравилаОбмена находятся правила, по которым будет производиться загрузка объектов. В подчиненном ему узле ПравилаКонвертацииОбъектов для каждого типа объектов есть свое правило. Сами объекты находятся после правил в узлах с именем Объект. В узле Ссылка находятся подчиненные узлы свойств, это те свойства, по которым будет произведена синхронизация этого объекта. То есть при загрузке объекта свойства из узла Ссылка будут получены первыми и по ним будет произведен поиск (с учетом настроек поиска по внутреннему идентификатору). Остальные свойства, стоящие вне узла Ссылка будут получены позднее, после того, как данный объект будет найден в базе или создан новый объект.

В формате xml есть несколько служебных символов, которые не могут быть непосредственно помещены в файл xml. Вместо этих символов используются коды этих символов. Обработка, формирующая xml-файл, сама производит конвертацию служебных символов.

Символ	Код символа
<	<
>	>
&	&

Файлы в формате xml можно открывать Интернет Эксплорером, там есть возможность сворачивать узлы для более удобного просмотра. Но там нельзя редактировать текст. Другой вариант – открывать с помощью блокнота, но там файл не будет структурирован, и его очень трудно читать. Оптимальный вариант – xml-редактор.

Запись объектов в файл данных. Как это происходит в программе?

После создания файла для записи данных, в первую очередь в него записывается узел с правилами обмена, согласно которым будет осуществляться загрузка. Потом из файла правил будет получена таблица ПВД, для первого ПВД будет получена выборка объектов и первый объект из этой выборки будет передан в функцию ВыгрузитьПоПравилу().
Дальше последовательность следующая

- 1) В функции ВыгрузитьПоПравилу() в переменной УзелСсылки создается узел с именем «Ссылка», куда будут записаны те свойства объекта, по которым будет производиться поиск.
- 2) Далее в процедуре ВыгрузитьСвойства() в этот узел записывается либо имя предопределенного элемента, либо подчиненные узлы «Свойство» для каждого свойства из тех, по которым будет производиться поиск. Эти узлы создаются в переменной УзелСвойства. Если свойство ссылочного типа, то для него рекурсивно вызывается функция ВыгрузитьПоПравилу().
- 3) В переменной Приемник создается узел с именем "Объект", туда добавляется в качестве дочернего уже созданный узел ссылки
- 4) В процедуре ВыгрузитьСвойства() в приемник записываются остальные свойства, не вошедшие в узел ссылки.
- 5) Узел Приемник записывается в файл данных. Таким образом, если функция ВыгрузитьПоПравилу() была вызвана рекурсивно для выгрузки свойства объекта по ссылке, то в файле сначала будет записан объект-свойство, а потом материнский объект.
- 6) Функция ВыгрузитьПоПравилу() возвращает узел ссылки выгруженного объекта. Если эта функция была вызвана для выгрузки объекта по ссылке из свойства, то возвращенный узел ссылки записывается в узел свойства.

Получается следующая картина:

Работа над ошибками

При настройке более- менее сложных правил, как правило, возникают ошибки. Как показывает практика, ошибки делятся на те, причину которых можно найти, хорошо подумав, и те, из-за которых приходится лезть в отладчик.

Совет 1

Прежде чем лезть в отладчик, можно узнать, какие значения находятся в каких переменных на разных этапах процессов выгрузки и загрузки. Для этого в обработчиках событий можно использовать процедуру Сообщить() для вывода информации в окно сообщений.

Совет 2

Чтобы понять, на каком этапе проблема, можно открыть xml-файл с выгруженными данными. Если там все в порядке – проблема на стороне загрузки. Если все равно непонятно, что происходит, открываем отладчик.

Поскольку конвертация происходит на этапе выгрузки, чаще всего отлаживать приходится именно выгрузку. Вот общая схема процесса выгрузки для упрощения отладки:

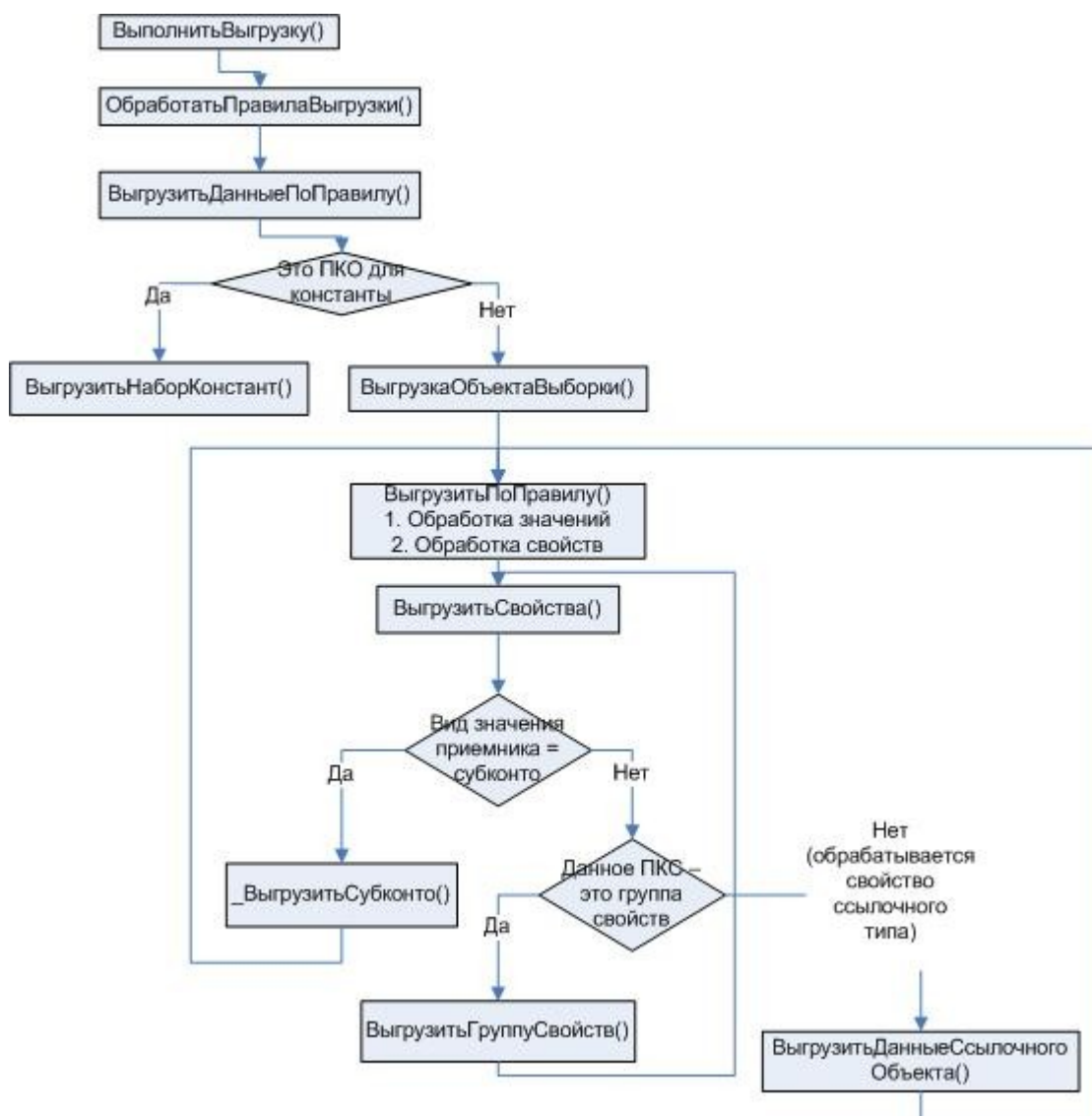


Рис. 71.

Здесь процедура `ВыгрузитьПоПравилу()` представляет собой конвертацию объекта согласно ПКО. Если необходимо из любого обработчика события вызвать выгрузку любого объекта, надо использовать именно эту процедуру. В нее передаются параметры:

`ВыгрузитьПоПравилу(Источник, Приемник, ВходящиеДанные, ИсходящиеДанные, ИмяПКО, УзелСсылки, ТолькоПолучитьУзелСсылки, ПКО, ЭтоПравилоСГлобальнойВыгрузкойОбъектов, ВыборкаДляВыгрузкиДанных)`. Все параметры необязательные, кроме параметра `ИмяПКО`. Если ПКО не определено, в процедуре будет выполнен поиск ПКО по имени ПКО.

Выгрузка проходит рекурсивно, при конвертации каждого объекта по ссылке получается свойство, и если оно имеет ссылочный тип и его надо выгружать, происходит передача управления в ПКО и выгрузка объекта-свойства.

Некоторые часто возникающие ошибки. При выгрузке

1. Объект не выгружается, хоть ошибок не выдает.

Это, скорее всего, значит, что перед передачей управления в ПКО в переменной `Значение` содержалось пустое значение. Тогда ошибки не возникает, но не возникает и передачи управления в ПКО. Если этот объект должен выгружаться по ссылке, то надо смотреть его ПКС, а если из выборки – ПВД. Для таких случаев полезно бывает вызывать в обработчиках событий процедуру `Сообщить()` и смотреть, что будет написано в окне сообщений в процессе выгрузки и загрузки.

2. `parentNode.appendChild(childNode);`

{<Путь к файлу обработки и номер строки>}: msxml3.dll: Параметр задан неверно.

Эта ошибка возникает при выгрузке из семерки. Причина ее в несоответствии версии парсера и создаваемого ДОМ-Документа.

Как исправлять: в автоматически создаваемом модуле обработки есть строка

```
_DOMDocument = СоздатьОбъект("Msxml2.DOMDocument.4.0");
```

Все будет работать, если в ней убрать номер версии ДОМ документа:

```
_DOMDocument = СоздатьОбъект("Msxml2.DOMDocument");
```

3. "Ошибка при загрузке правил обмена"

Правила обмена должны начинаться с тега `ПравилаОбмена`. Если это не так, возникает ошибка.

4. "Ошибка формата правил обмена"

Правила обмена найдены, но с содержанием этих правил что-то не в порядке.

5. "Не корректно указано имя файла для выгрузки данных"

Ошибка возникла при попытке открыть файл для выгрузки данных.

6. "Ошибка получения значения свойства объекта (по имени свойства приемника)"

Ошибка возникла при выгрузке в процедуре `ПолучитьЗначениеСвойства()`. В некоторое ПКО был передан объект. Программа начала выполнение ПКС этого ПКО. В одном из ПКС не было указано свойство-источник, но свойство-приемник указано. У переданного объекта свойства с таким именем нет.

7. "Ошибка получения значения свойства подчиненного объекта (по имени свойства источника)"

Ошибка возникла при выгрузке в процедуре `ПолучитьЗначениеСвойства()`. В некоторое ПКО был передан объект. В данном ПКО имеется группа ПКС (табличная часть, движения регистра или просто группа). Программа начала выполнение ПКС из этой группы. В одном из ПКС было указано такое свойство-источник, которого в переданном объекте нет.

8. `одСообщения.Вставить(17, "Ошибка получения значения свойства подчиненного объекта (по имени свойства приемника)");`

Ошибка возникла при выгрузке в процедуре `ПолучитьЗначениеСвойства()`. В некоторое ПКО был передан объект. В данном ПКО имеется группа ПКС (табличная часть, движения регистра или просто группа). Программа начала выполнение ПКС из этой группы. В одном из ПКС не было указано свойство-источник, но свойство-приемник указано. У переданного объекта свойства с таким именем нет.

9. "Ошибка в обработчике события"

Ошибка может быть как непосредственно в коде обработчика события, так и в процедуре или функции, которая вызвана из этого обработчика.

10. "Ошибка получения значения свойства объекта (по имени свойства источника)"

Ошибка возникла при выгрузке в процедуре `ПолучитьЗначениеСвойства()`. В некоторое ПКО был передан объект. Программа начала выполнение ПКС этого ПКО. В одном из ПКС было указано такое свойство-источник, которого в переданном объекте нет.

11. "Ошибка при выполнении алгоритма, содержащегося в файле обмена"

Ошибка возникла при выполнении алгоритма, написанного на закладке `Алгоритмы` в окне настройки правил обмена в конвертации или в вызванной из него процедуре или функции.

12. "Ошибка получения коллекции подчиненных объектов из входящих данных "

В данном ПКО есть группа ПКС, отмеченная флагом «Получить Из Входящих Данных». Ошибка произошла при попытке получить эту группу из структуры `Входящие Данные`.

13. "Ошибка получения свойства подчиненного объекта из входящих данных "

В ПКО есть группа ПКС. В группе есть ПКС, отмеченное флагом «Получить из входящих данных». В процессе выгрузки группа была получена, при выгрузке очередного элемента этой группы была произведена неудачная попытка получить указанное свойство из входящих данных.

14. "Ошибка получения свойства объекта из входящих данных "

Одно из ПКС данного ПКО отмечено флагом «Получить из входящих данных». На этапе выполнения данного ПКС в структуре `ВходящиеДанные` элемента с таким именем найдено не было.

15. "Не найдено соответствие для значения Источника"

В ПКО нет поиска по уникальному идентификатору и полей поиска, поиск ведется только по ПКЗ. Но для данного значения из источника ПКЗ не найдено.

16. Выгружается только один объект

Ошибка возникает, если в ПКО нет источника и объект получен из входящих данных, либо если ПКО вызвано из ПВД, в котором задействована переменная ВыборкаДанных. В обоих этих случаях программа некорректно определяет ключ выгружаемых данных, он получается неуникальный для разных объектов выборки. Чтобы этого не происходило, необходимо задать ключ выгружаемых данных самостоятельно (**См. раздел "Ключ выгружаемых данных"**), либо поставить в ПКО флажок «Не запоминать выгруженные».

При загрузке

17. Сообщение в окне сообщений при загрузке: «Ошибка записи регистра: Поле Количество (Валюта, Валютная сумма) должно быть пустым».

Возникает из-за того, что в запись регистра с неколичественным счетом подставляется количество. Чтобы подобные ошибки не возникали, при переносе движений регистра бухгалтерии в ПКС для полей КоличествоДт, КоличествоКт, ВалютнаяСуммаДт, ВалютнаяСуммаКт, ВалютаДт, ВалютаКт следует сразу писать в обработчике ПередВыгрузкой что-то вроде такого:

```
Если НЕ ОбъектКоллекции.СчетДт.Валютный Тогда  
    Отказ = Истина;  
КонецЕсли;
```

18. Объекты перегружаются по ссылке, но ссылки на эти объекты утрачиваются.

В событии После выгрузки свойства переменная Отказ установлена в Истина.

19. Ошибка при установке значения атрибута контекста: Реквизит недоступен для группы"

Причина – при загрузке происходит попытка присвоить группе справочника реквизит, недоступный для группы. Чтобы ошибка не возникала, во всех ПКС, недоступных для группы в приемнике, в событии Перед Выгрузкой необходимо вызвать проверку:

```
Отказ = Источник.ЭтоГруппа;
```

20. Для номенклатуры не переносятся реквизиты «Единица хранения остатков» и «Единица для отчетов», для контрагентов не переносится реквизит «Основной договор»

Такое бывает, даже если все правильно настроено. Это происходит в тех случаях, когда подчиненный справочник одновременно фигурирует в одном из реквизитов. В этом случае элементы подчиненного справочника тянут по ссылке Владельца, и сами из владельца тянутся по ссылке. Есть два способа исправить эту ошибку:

- 1) Если единицы измерения выгружаются прежде, чем Номенклатура (Договоры контрагентов прежде, чем контрагенты), тогда путем копирования еще одно ПКО "ЕдиницыИзмерения" (ДоговорыКонтрагентов), назвать его "ЕдиницыИзмерения1" (ДоговорыКонтрагентов1). В ПКО номенклатура, ПКО НоменклатурныеГруппы (Контрагенты) указать для нужных реквизитов новое созданное правило.
- 2) Можно поменять местами ПВД – Договоры контрагентов и единицы измерения поставить после Номенклатуры и Контрагентов.

21. Из семерки не переносятся реквизиты номера и даты документов

Эта ошибка исправлена в последних версиях программы Конвертация данных. Связана она была с тем, что при загрузке метаданных семерочной базы номер документа назывался Номер вместо НомерДок, а дата документа Дата вместо ДатаДок.

22. "Ошибка распаковки файла обмена. Файл заблокирован."

Может возникнуть при выгрузке, если правила были в формате .aml, или при загрузке, если файл с данными был в формате .aml.

23. "Ошибка формата файла обмена"

Файл с данными для загрузки должен начинаться с тега ФайлОбмена. Если это не так, программа выдает ошибку

Принятые сокращения

БП – Бухгалтерия предприятия
ПВД – Правило выгрузки данных
ПВХ – План видов характеристик
ПКЗ – Правило конвертации значений
ПКО – Правило конвертации объектов
ПКС – Правило конвертации свойств
УПП – Управление производственным предприятием
УТ – Управление торговлей

Источники информации

- 1) Встроенный учебник «Быстрое освоение программы»
- 2) Материалы, поставляемые на диске Информационно-Технического сопровождения
- 3) Форум для разработчиков <http://partners.v8.1c.ru/forum/>

Контакты

Вопросы, предложения и отзывы отправлять по адресу
fineosun@mail.ru
icq 258182003
Ольга Кузнецова