

КОМПЛЕКСНАЯ СИСТЕМА УПРАВЛЕНИЯ ПАРУС
(ПАРУС 8)

КОНВЕРТЕР

8.5.6.1

ИНСТРУКЦИЯ

Оглавление

1 Введение.....	6
2 Описание разделов	8
2.1 Конвертер: Форматы обмена данными.....	8
2.1.1 Назначение раздела.....	8
2.1.2 Структура раздела.....	8
2.2 Конвертер: Источники данных.....	9
2.2.1 Назначение раздела.....	9
2.2.2 Структура раздела.....	10
2.3 Конвертер: Хранилище.....	10
2.3.1 Назначение раздела.....	10
2.3.2 Структура раздела.....	10
2.4 Конвертер: Алгоритмы загрузки	11
2.4.1 Назначение раздела.....	11
2.4.2 Структура раздела.....	11
2.5 Конвертер: Внешние ключи.....	14
2.5.1 Назначение раздела.....	14
2.5.2 Структура раздела.....	14
3 Использование разделов.....	15
3.1 Настройка и отладка	15
3.2 Работа пользователя.....	16
3.3 Сквозной пример.....	16
3.3.1 Книга Excel	16

3.3.2 Тестовый файл с разделителями	31
3.3.3 Текстовый файл с полями фиксированной длины	36
3.3.4 Файл XML.....	43
3.3.5 База данных Oracle.....	50
3.3.6 Форма сводной отчетности.....	60
4 Тиражирование настроек.....	67
5 Заключение	71

Условные обозначения:

Конвертер - преобразование данных из одного формата в другой с сохранением основного логически-структурного содержания информации. В контексте данного документа – заполнение базы данных ПАРУС 8 на основании внешних файлов или таблиц.

Мнемокод – краткое наименование (уникальное для системы поле).

СУБД – Система управления базами данных, совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

НСИ – нормативно-справочная информация.

Класс – это описание логических и физических сущностей. Соответствует понятию "Раздел". В классе описывается способ организации и визуализации данных в разделе.

Раздел – описание сущности предметной области. С точки зрения разработчика Системы под разделом понимают механизм, позволяющий манипулировать данными экземпляров (списков экземпляров) классов и вызывать методы классов, используя формы и представления классов. Основное назначение раздела – сгруппировать функционально связанные объекты СУБД.

Спецификация – подробное описание. В контексте данного документа. Detail–классы, включенные в выбранный Master–класс. В Системе многие разделы имеют заголовок и спецификации. Класс "Спецификация" регистрируется как Detail–класс для Master–класса "Заголовок".

Первичный ключ (англ. primary key) — в реляционной модели данных один из потенциальных ключей отношения, выбранный в качестве основного ключа (или ключа по умолчанию). Также термин «первичный ключ» вошёл в формулировку 12 правил Кодда как основной способ адресации любого значения отношения (таблицы) наряду с именем отношения (таблицы) и именем атрибута

(столбца). В ПАРУС 8 в каждой таблице существует регистрационный номер (registration number) записи, часто называемый просто RN записи (чаще всего имя поля – RN). Именно это поле используется в качестве первичного ключа. Поля аналогичного назначения есть, как правило, и в других реляционных базах. В рамках данного документа, для краткости, мы будем использовать термины "RN записи" и "внешний RN", для обозначения первичных ключей ПАРУС 8 и источников данных для конвертации.

1 Введение

В процессе эксплуатации информационной системы нередко возникают подобные задачи:

- загрузка или выгрузка документа в заданном формате;
- конвертация данных из внешней системы;
- формирование НСИ на основе электронных файлов.

Как правило, такие задачи решаются по мере возникновения каждый раз заново под конкретный формат или систему. Однако, так делается не всегда. В настоящее время в системе существует несколько настраиваемых механизмов, предназначенных для решения подобных задач без программирования.

- а) Пользовательское приложение "Универсальный импорт из Excel".
- б) Механизм формирования данных из формы сводной отчетности.
- в) Формирование произвольного XML файла.
- г) Загрузка произвольного XML в форму сводной отчетности.

Каждый из этих механизмов имеет серьезные недостатки, главный из которых – узкая область применения. Однако можно заметить, что любой из этих механизмов решает некие типовые задачи:

- описание формата источника;
- описание правил заполнения данных.

Причем в каждом механизме эти задачи решаются совершенно по-разному, что повышает трудоемкость при описании новых форматов.

Целесообразно было разработать единые подходы, которые позволяют использовать описания форматом типовым образом и легко тиражировать настроенные описания для прикладных задач. Таким образом, "Конвертер" предназначен для решения следующих задач:

- 1) Загрузка данных в систему из одного из форматов:
 - Книга Excel;
 - Файл XML;

- Тестовый файл с разделителями;
- Текстовый файл с полями фиксированной длины;
- Форма сводной отчетности;
- База данных Oracle;

2) Выгрузка данных в XML файл.

В данной инструкции мы подробно рассмотрим решение первой задачи, так как она не требует навыков программирования и рассчитана на широкий круг пользователей. Для создания выгрузки в XML необходимы знания PL SQL что сужает круг потенциальный пользователей этого механизма. Выгрузку в XML мы не будем затрагивать в рамках данной инструкции.

2 Описание разделов

2.1 Конвертер: Форматы обмена данными

2.1.1 Назначение раздела

Раздел предназначен для описания формата источника данных. Раздел является иерархическим, каждый уровень иерархии имеет свое назначение.

Тип понятия\уровень	1	2	3	4	5 и более
Книга Excel	Папка, группа форматов	Наименование формата и определение типа понятия	Листы	Колонки	Запрещен
Файл XML			Таг	Таг	Таг
Тестовый файл с разделителями			Строка	Поле	Поле
Текстовый файл с полями фиксированной длины			Строка	Поле (диапазон символов)	Поле (диапазон символов)
Форма сводной отчетности			Таблица	Показатель	Запрещен
База данных Oracle			Запрос	Поле	Запрещен

Таблица 1

Как видно из таблицы 1, для любого типа понятия в разделе должно быть четыре или более уровня вложенности. Для текстовых файлов и XML количество уровней не ограничено, так как вложенность понятий возможна любая.

2.1.2 Структура раздела

Заголовок:

- Порядковый номер – простой счетчик в рамках узла иерархии;
- Наименование – понятное, "пользовательское" название понятия, в зависимости от типа и уровня, например, имя листа EXCEL;
- Идентификатор – код этого понятия, используемый в алгоритмах;
- Примечание – поле для заметок;
- Тип понятия – одно из значений, перечень в первой колонке таблицы 1;
- Тип данных – строка, число, дата и.т.д.;
- Значение по умолчанию – используется, если значение не сформировалось иным способом;

- Запрос – SQL запрос, используется для выгрузки в XML или для загрузки из таблиц СУБД, формирует список используемых записей;

- Поле запроса для значения – SQL запрос, используется для выгрузки в XML, формирует значение тага.

Спецификация "параметры":

Используется только для выгрузки в XML

- Мнемокод;
- Наименование;
- Понятие значения параметра;
- Поле значения параметра.

2.2 Конвертер: Источники данных

2.2.1 Назначение раздела

Раздел предназначен для хранения данных, которые в дальнейшем будут подлежать конвертации. Например, при конвертации данных из внешней системы. Практически, возможны два режима загрузки:

- В два этапа, с использованием "Источников данных", сначала загружаем в источник, затем заполняем данные в разделах Системы;

- В один этап, разделы системы формируются сразу. Разумеется, все равно сначала заполняется промежуточное хранилище, а затем данные загружаются в разделы, но это происходит незаметно, и хранилище очищается после завершения конвертации. Для работы в данном режиме, раздел "Источники данных" не используется.

Тип понятия\уровень	Использование раздела "Источники данных"
Книга Excel	Используется при отладке алгоритмов или, если это сложный файл с большим количеством разнообразных данных загружаемых в разные разделы
Файл XML	Используется при отладке алгоритмов или, если это сложный файл с большим количеством разнообразных данных загружаемых в разные разделы
Тестовый файл с разделителями	Используется только при отладке алгоритмов

Тип понятия\уровень	Использование раздела "Источники данных"
Текстовый файл с полями фиксированной длины	Используется только при отладке алгоритмов
Форма сводной отчетности	Не используется, конвертация запускается специальной процедурой непосредственно из первичного отчета в Системе
База данных Oracle	Используется для вызова конвертации, данные в хранилище не зачисляются, берутся непосредственно из СУБД

Таблица 2

В таблице 2 описывается, для каких типов понятий используется раздел.

2.2.2 Структура раздела

- Тип документа – ссылка на "форматы обмена данными";
- Префикс документа;
- Номер документа;
- Дата документа;
- Файл – имя загруженного файла;
- Дата загрузки
- Примечание

Заголовок ссылается на «хранилище», в котором хранятся собственно данные после загрузки.

2.3 Конвертер: Хранилище

2.3.1 Назначение раздела

Раздел, в котором хранятся загруженные данные, разобранные по узлам формата обмена данными. Записи в хранилище могут либо храниться, при использовании "Источников данных", либо автоматически удаляться, при загрузке документов из разделов системы.

2.3.2 Структура раздела

- Наименование – ссылка на запись описания формата;
- Номер экземпляра – номер по порядку в рамках узла иерархии;
- Значение (число);
- Значение (строка);

- Значение (дата);
- Значение (текстовые данные);
- Значение (двоичные данные)

Заполняется одно из полей "значений", в зависимости от типа данных.

2.4 Конвертер: Алгоритмы загрузки

2.4.1 Назначение раздела

Раздел описывает правила заполнения данных в разделах Системы с использованием данных "хранилища". Кроме того, в этом же разделе настраивается вызов загрузки и ведётся протокол загрузки.

2.4.2 Структура раздела

Заголовок:

- Наименование
- Примечание

Спецификации первого уровня:

Буфер загрузки

В спецификации нет никаких действий, строки формируются при загрузке и используются для анализа ошибок

- Порядковый номер загрузки;
- Источник данных;
- Таблица источника данных;
- Организация;
- Номер загружаемой строки;
- Номер загружаемого раздела;
- Статус;
- Текст блока загрузки;
- Сообщение об ошибке.

Разделы вызова

- Наименование вызова;
- Маска файлов – если в одном разделе нам необходимы вызовы различных файлов, они должны отличаться по маске;
- Раздел – раздел Системы, в котором мы будем запускать загрузку данных
- Сохранить буфер в хранилище после загрузки – логическое поле, имеет смысл при загрузке из разделов. Если установить "галку", хранилище не будет удаляться после загрузки.

Загружаемые разделы

- Порядковый номер загрузки – счетчик в рамках узла иерархии;
- Раздел – выбор из списка классов;
- Таблица раздела – заполняется автоматически после выбора раздела;
- Представление раздела – заполняется автоматически после выбора раздела;
- Процедура – заполняется автоматически после выбора раздела процедурой добавления;
- Способ выполнения – выбор из списка;
- Обязательность – логическое поле;
- Таблица источника – ссылка на описание формата;
- Условие выполнения – формула, которая должна вернуть значение «истина» или «ложь», в формуле могут использоваться любые выражения PL SQL и таги (элементы макроподстановки в треугольных скобках <имя>, которые при конвертации заменяются на значения соответствующих полей источника или параметров);
- Примечание.

Спецификации второго уровня, подчиненная спецификации "Загружаемые разделы". Параметры:

- Мнемокод – код поля;
- Наименование – наименование поля;

- Тип параметра – входной или выходной;
- Тип данных – строка, число, дата и.т.д.;
- Длина данных;
- Поле таблицы;
- Обязательное – логическое, да или нет;
- Контекст – выбор из списка;
- Показатель источника – выбор из описания формата;
- Параметр – параметр процедуры добавления записи;
- Наименование параметра;
- Значение по умолчанию;
- Удалить крайние пробелы – логическое, да или нет;
- Поле представления;
- Входит в уникальный ключ – логическое, да или нет, пять полей для пяти уникальных ключей;
- Функция обработки значения – выражение PL SQL;
- Сгенерить уникальное значение – логическое, да или нет;
- Примечание;
- Выравнивать пробелами слева – логическое, да или нет.

Действия

Кроме стандартных, в разделе есть два важных, удобных действия, облегчающие процесс настройки и отладки:

- "Пересчитать список параметров";
- "Продолжить конвертацию".

Первое действие нужно использовать после обновления на новый релиз. В используемых процедурах могут появляться новые параметры. Действие "Пересчитать список параметров" позволит быстро, сразу по всем записям восстановить работоспособность алгоритмов, не вникая в то, какие параметры были добавлены в используемые в алгоритме процедуры.

Второе действие может быть полезно при отладке. В случае, если конвертация прекратилась в результате ошибки, мы можем внести изменения в алгоритм и продолжить конвертацию с того места, на котором она прервалась.

2.5 Конвертер: Внешние ключи

2.5.1 Назначение раздела

Раздел предназначен для сохранения "внешних RN" конвертируемых данных. Используется в случае, когда источником является реляционная база данных. Более подробно будет описано в разделе инструкции "Сквозной пример" → "База данных Oracle".

2.5.2 Структура раздела

- Внешний ключ – Первичный ключ (primary key) или "внешний RN" источника данных
- Код раздела – Код раздела, в котором создана запись;
- Рег.номер в системе Парус – RN созданной записи;
- Раздел – наименование раздела, в котором создана запись.

3 Использование разделов

3.1 Настройка и отладка

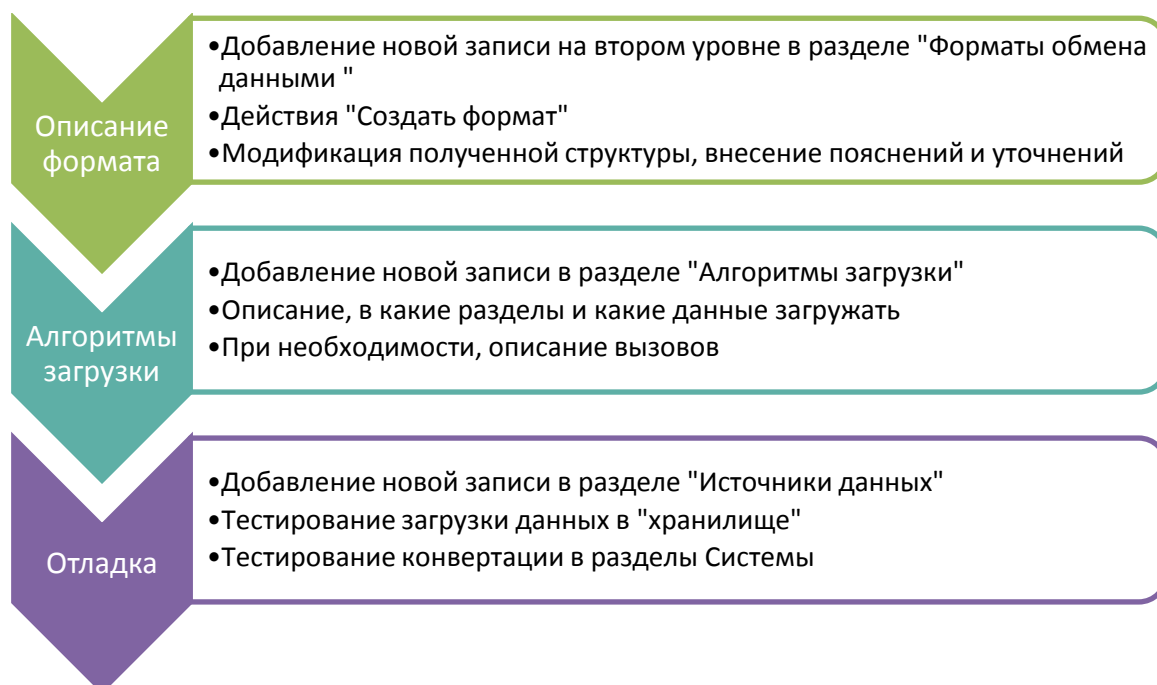


Рисунок 1

На рисунке 1 схематично показан процесс настройки и отладки процесса конвертации данных. Более подробно порядок выполнения действий удобно будет разобрать на примере (раздел инструкции "Сквозной пример").

Желательно настраивать алгоритмы под пользователем – владельцем схемы (чаще всего PARUS). В противном случае не видны описания методов и имя метода нельзя будет выбрать из списка, что не очень удобно.

3.2 Работа пользователя



Рисунок 2

На рисунке 2 показаны варианты запуска конвертации пользователями. Загрузка файла непосредственно из раздела является упрощенным способом, позволяющим избежать промежуточное действие по загрузке данных в хранилище. В любом случае, документ можно загрузить и вторым способом, особенно удобно это делать при отладке алгоритма. Информацию из "Сводной отчетности" можно загружать только специализированной процедурой, так как необходимо знать, какой именно первичный/сводный отчет используется в качестве источника данных.

3.3 Сквозной пример

Для того чтобы разобрать более подробно способы использования механизма конвертации, разберем сквозные примеры настройки и конвертации данных отдельно по каждому формату.

3.3.1 Книга Excel

Рассмотрим в качестве примера загрузку "Целевых статей" из файла

...example\TARGET_ITEM.xlsx. При этом, чтобы пример был не слишком тривиальным, одновременно будем заполнять подчиненный раздел "Направление расходов".

Добавление Формата обмена данными

Заходим в раздел Формат обмена данными. Первый уровень раздела используется для группировки записей удобным для пользователя способом, например, по форматам. Находим или создаем подходящую по смыслу запись, например, "Загрузки из Excel". Внутри ее выполняем действие "Создать формат". Заполняем поля в соответствии с нашим файлом - источником.

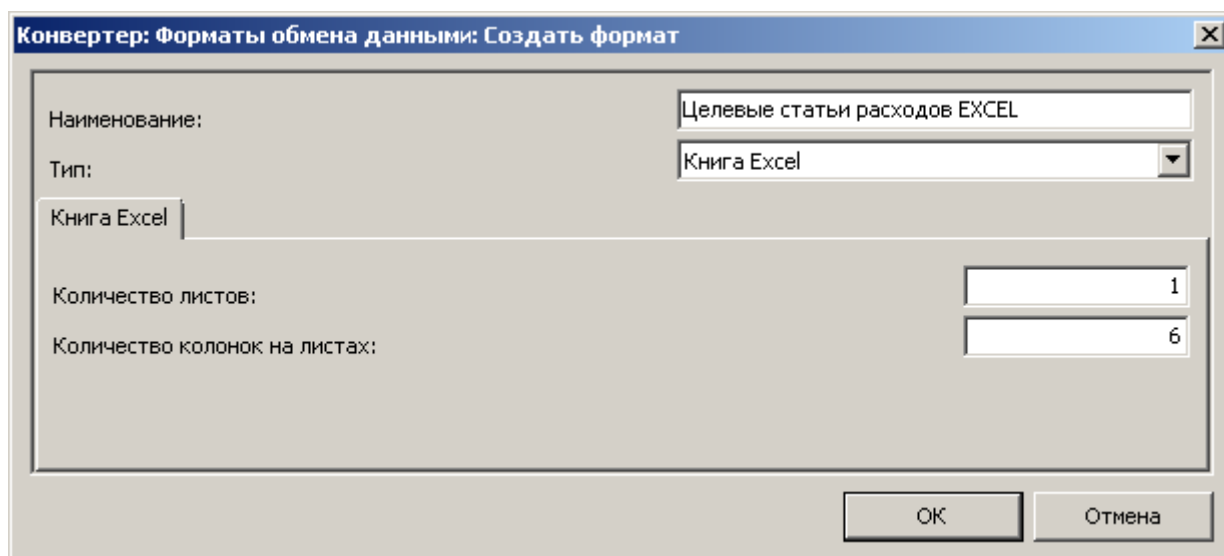


Рисунок 3

Система создаст дерево с именем листа и именами колонок. Для удобства можно заполнить наименование колонок понятными для себя значениями. Например, "А" заменить на "Код". При этом идентификатор колодки останется "А"

Конвертер: Форматы обмена данными: Исправление

Понятие | **Запрос**

Порядковый номер: 1 ✓

Наименование: Код

Идентификатор: A

Тип понятия: Колонка Excel

Тип данных: Строка

Значение по умолчанию:

Примечание:

⏪ ⏩ + + ✕

OK Отмена

Рисунок 4

В результате мы получили описание формата файла – источника.

ПАРУС-Бюджет В Конструктор отраслевых расширений - [Конвертер: Форматы обмена данными]

Файл | Учет | Форматы | Отчеты | Конвертер | Монитор | Словари | Офис | Справка

Структура

- Конвертер: Форматы обмена данными
 - Выгрузки в XML
 - Загрузки из Excel
 - Загрузка Excel с Контрагентами
 - Универсальный шаблон Excel
 - Целевые статьи расходов EXCEL
 - TARGET_ITEM**
 - Дата начала действия
 - Код
 - Код родителя
 - Наименование
 - Направление расходов
 - Полное наименование
 - Загрузки из XML
 - Загрузки из базы данных Oracle
 - Загрузки из сводной отчетности
 - Загрузки из текстовых файлов

Конвертер: Форматы обмена данными

Порядковый номер	Наименование	Идентификатор	Тип понятия
1	Код	A	Колонка Excel
2	Наименование	B	Колонка Excel
3	Полное наименование	C	Колонка Excel
4	Дата начала действия	D	Колонка Excel
5	Код родителя	E	Колонка Excel
6	Направление расходов	F	Колонка Excel

15:50:58 demo.beshosting@PARUS ГБУ ВЕСЕЛОВСКИЙ ММ116-2 NUM

Рисунок 5

На рисунке 5 мы видим результат этого этапа настройки.

Создание Алгоритма загрузки

Добавляем новую запись в разделе "Алгоритмы загрузки"

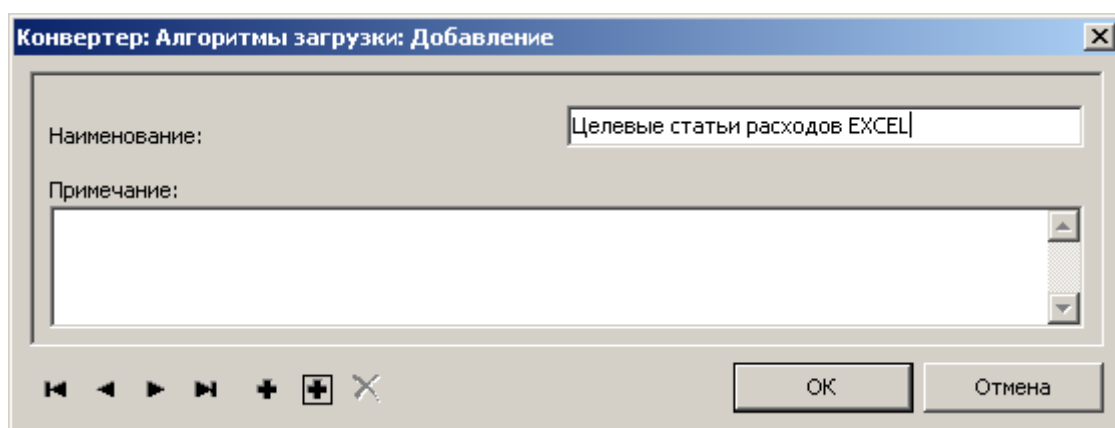


Рисунок 6

На втором уровне добавляем строку добавив наш, только что созданный формат источника.

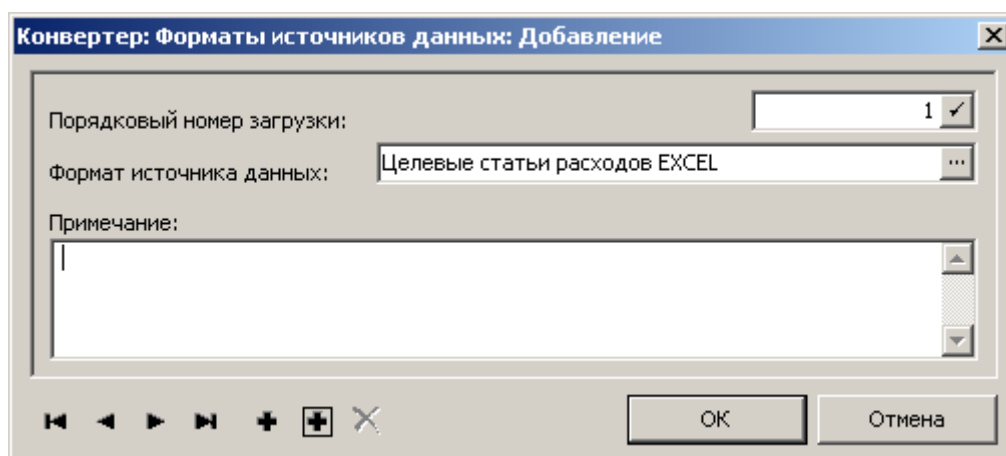


Рисунок 7

Конвертер позволяет добавлять для одного алгоритма любое количество источников. Это необходимо для сложных конвертаций, например, из других систем. В нашем примере строка с форматом источника будет всего одна.

В спецификации третьего уровня, мы добавляем разделы системы, которые будем заполнять при конвертации. Разделы надо заполнять с нижнего уровня. Например, мы начнем с раздела "Направления расходов", так как он должен быть уже заполнен при создании целевой статьи. При выборе раздела, поля "таблица", "представление", "Процедура добавления", "Способ выполнения" заполняться автоматически. Не будем ничего менять в этих полях.

Конвертер: Загружаемые разделы: Добавление

Порядковый номер загрузки: 1 ✓

Таблица источника: TARGET_ITEM ...

Раздел: Направления расходов ...

Таблица раздела: GOVPROG ...

Представление раздела: V_GOVPROG ...

Процедура: P_GOVPROG_INSERT ...

Способ выполнения: Поиск, запуск процедуры (если не нашли) для вставки ▾

☐ Обязательность (ошибка, если запись не найдена)

Условие выполнения (формула):

Параметры спецификации и показатели, например:

<NRN> is not null

<AGNABBR> is not null

Примечание:

⏪ ⏩ + + X OK Отмена

Рисунок 8

Однако, поле "Условие выполнения" необходимо заполнить формулой $\langle \text{ROWNUM} \rangle \geq 2$. То есть строка, больше или равна 2. Это необходимо, так как в первой строке нашего файла-источника содержится заголовок, а данные начинаются со второй строки.

Спецификация параметров заполнится автоматически. Нам лишь необходимо заполнить поле "Показатель источника" для тех параметров процедуры добавления, которые мы будем заполнять из нашего файла.

Конвертер: Параметры: Исправление

Параметр

Порядковый номер: 3

Мнемокод параметра: SCODE

Наименование параметра: Мнемокод

Тип параметра: Входной

Поле представления: SCODE

Тип данных: Строка Длина: 20 ☒ Обязательное

Источник значения

Контекст: Нет

Показатель источника: Направление расходов

Параметр:

Значение по умолчанию:

Обработка значения

Функция обработки значения:

Примеры:
substr(<VALUE>,1,3)
upper(<VALUE>)
<VALUE>*20/100

☒ Удалить крайние пробелы ☐ Выравнивать пробелами слева (префикс, номер)

Уникальность

Входит в уникальный: ☒ ключ 1 ☐ ключ 2 ☐ ключ 3 ☐ ключ 4 ☐ ключ 5

☐ Сгенерить уникальное значение (для обхода физического ключа уникальности)

Примечание

ОК Отмена

Рисунок 9

"Мнемокод" и "Код" мы будем заполнять значением колонки "Направление расходов", а "Наименование" и "Полное наименование" одноименными колонками.

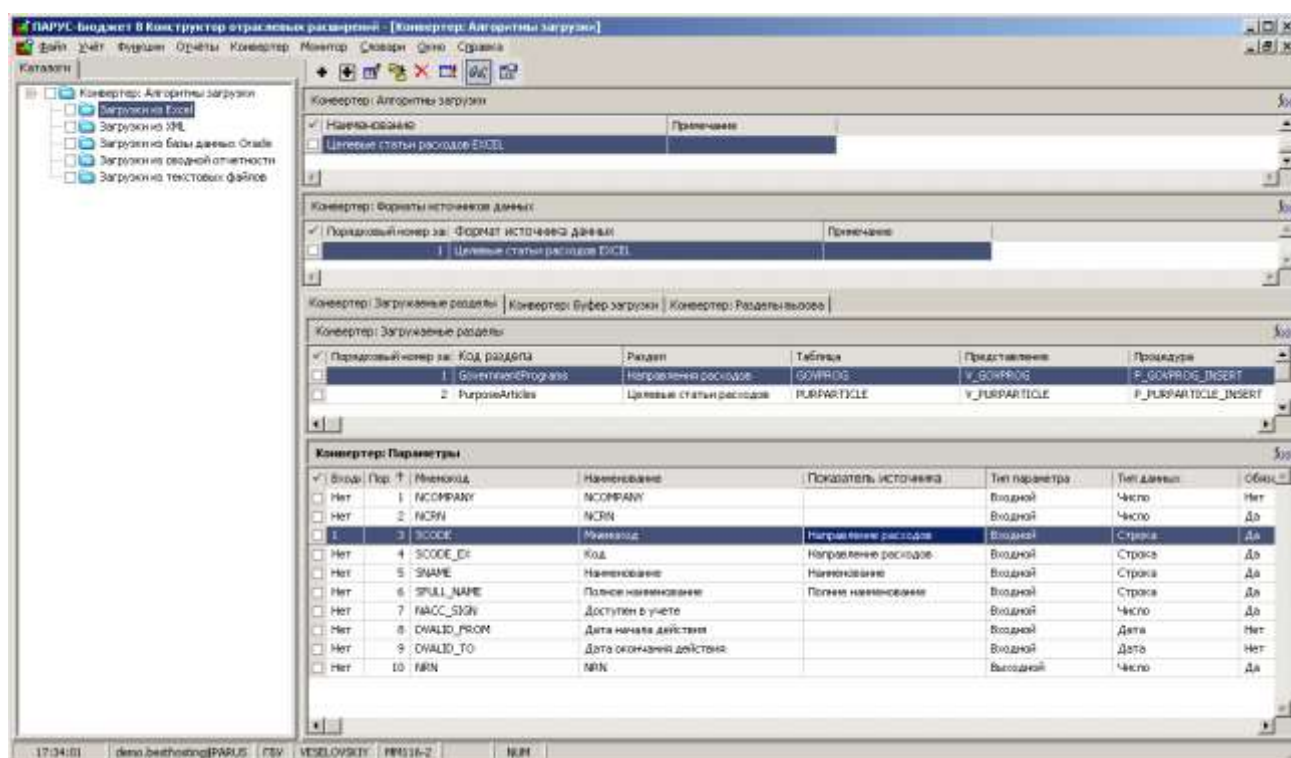


Рисунок 10

Теперь необходимо добавить раздел "Целевые статьи расходов". Но тут есть некая особенность. Раздел иерархический, при добавлении нужен параметр "Родитель в иерархии". Поэтому мы опишем раздел в алгоритме дважды. В первый раз, чтобы найти родителя, а второй раз уже собственно для добавления записи.

Итак, добавляем раздел "Целевые статьи расходов", но на этот раз меняем "способ выполнения" на "Только поиск"

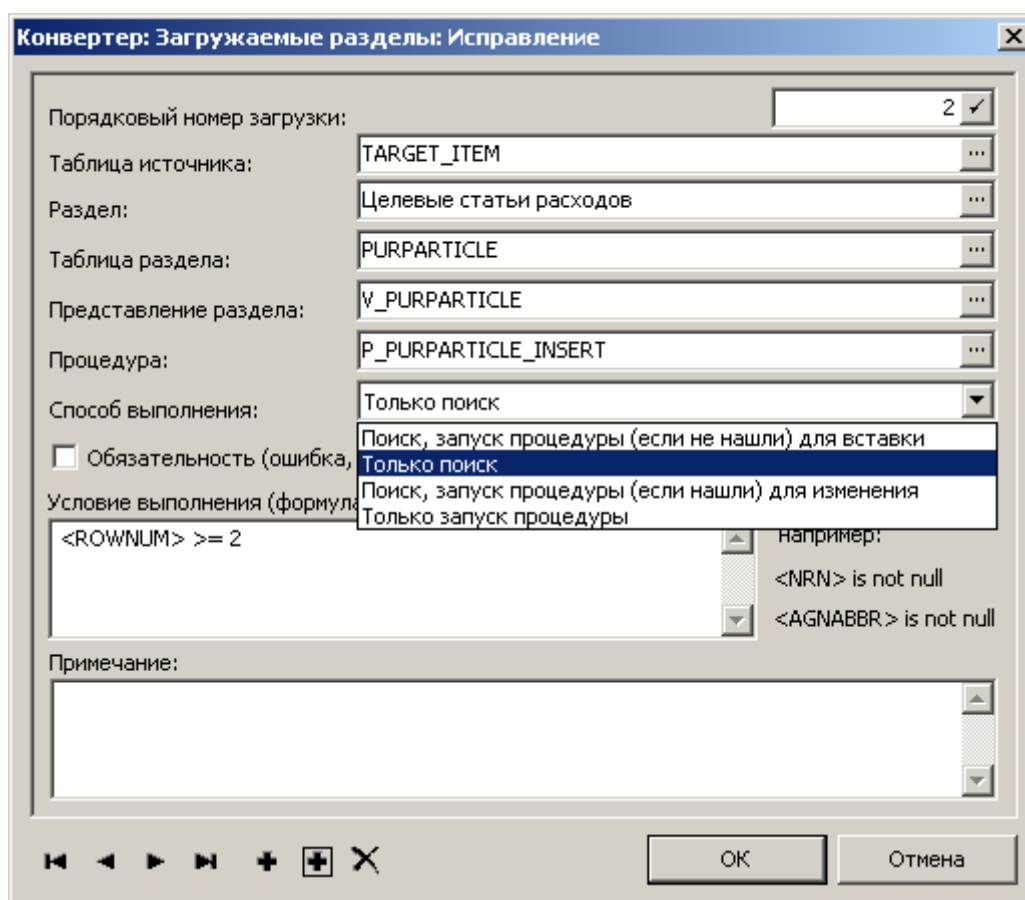


Рисунок 11

"Показатель источника" заполняем только для поля "Мнемокод" значением "Код родителя". То есть, мы хотим найти строки, которые будут родительскими в дальнейшем.

Конвертер: Параметры: Исправление

Параметр

Порядковый номер: 3

Мнемокод параметра: SCODE

Наименование параметра: Мнемокод

Тип параметра: Входной

Поле представления: SCODE

Тип данных: Строка Длина: 20 ☒ Обязательное

Источник значения

Контекст: Нет

Показатель источника: Код родителя

Параметр:

Значение по умолчанию:

Обработка значения

Функция обработки значения:

Примеры:
substr(<VALUE>,1,3)
upper(<VALUE>)
<VALUE>*20/100

☒ Удалить крайние пробелы ☐ Выравнивать пробелами слева (префикс, номер)

Уникальность

Входит в уникальный: ☒ ключ 1 ☐ ключ 2 ☐ ключ 3 ☐ ключ 4 ☐ ключ 5

☐ Сгенерить уникальное значение (для обхода физического ключа уникальности)

Примечание

OK Отмена

Рисунок 12

Ну и, наконец, добавляем еще раз этот же раздел, уже для добавления записи.

Конвертер: Загружаемые разделы: Исправление

Порядковый номер загрузки: 3 ✓

Таблица источника: TARGET_ITEM ...

Раздел: Целевые статьи расходов ...

Таблица раздела: PURPARTICLE ...

Представление раздела: V_PURPARTICLE ...

Процедура: P_PURPARTICLE_INSERT ...

Способ выполнения: Поиск, запуск процедуры (если не нашли) для вставки

☐ Обязательность (ошибка, если запись не найдена)

Условие выполнения (формула): <ROWNUM> >= 2

Параметры спецификации и показатели, например:
<NRN> is not null
<AGNABBR> is not null

Примечание:

⏪ ⏩ ⏴ ⏵ + + ×

OK Отмена

Рисунок 13

Далее заполняем спецификацию параметров. Остановимся на некоторых новых моментах.

При заполнении поля "Родитель в иерархии" используем "Параметр", то есть найденную на предыдущем шаге запись. То есть "Проваливаемся" из поля "Параметр" в список и выбираем найденный на прошлом шаге RN записи

Конвертер: Загружаемые разделы

Конвертер: Загружаемые разделы

✓	Порядковый номер за	Код раздела	Раздел	Только поиск	Обязательность	Представление
<input type="checkbox"/>	1	GovernmentPrograms	Направления расходов	Поиск, запуск процед:	Нет	V_GOVPROG
<input type="checkbox"/>	2	PurposeArticles	Целевые статьи расходов	Только поиск	Нет	V_PURPARTICLE
<input type="checkbox"/>	3	PurposeArticles	Целевые статьи расходов	Поиск, запуск процед:	Нет	V_PURPARTICLE

Конвертер: Параметры

Конвертер: Параметры

✓	Порядковый номер	МНЕМОКОД	Наименование	Тип параметра	Тип данных	Длина данных
<input type="checkbox"/>	1	NCOMPANY	NCOMPANY	Входной	Число	
<input type="checkbox"/>	2	NPRN	Родитель в иерархии	Входной	Число	
<input type="checkbox"/>	3	SCODE	Мнемокод	Входной	Строка	
<input type="checkbox"/>	4	SCODE_EX	Код	Входной	Строка	
<input type="checkbox"/>	5	SNAME	Наименование	Входной	Строка	2
<input type="checkbox"/>	6	SFULL_NAME	Полное наименование	Входной	Строка	2
<input type="checkbox"/>	7	SCODE_ST	Код программной (непрограмм	Входной	Строка	
<input type="checkbox"/>	8	SGOVPROG	Направление расходов	Входной	Строка	
<input type="checkbox"/>	9	SCODE_NR	Код направления расходов	Входной	Строка	
<input type="checkbox"/>	10	NACC_SIGN	Доступна в учёте	Входной	Число	
<input type="checkbox"/>	11	NEXT_SIGN	Расширение	Входной	Число	
<input type="checkbox"/>	12	DVALID_FROM	Дата начала действия	Входной	Дата	
<input type="checkbox"/>	13	DVALID_TO	Дата окончания действия	Входной	Дата	
<input type="checkbox"/>	14	NRN	NRN	Выходной	Число	

OK Отмена Справка

Рисунок 14

Результат показан на рисунке 15. Это можно считать примером использования в качестве параметра функции добавления выходного параметра, рассчитанного на предыдущем шаге.

Конвертер: Параметры: Исправление

Параметр

Порядковый номер: 2

Мнемикод параметра: NPRN

Наименование параметра: Родитель в иерархии

Тип параметра: Входной

Поле представления:

Тип данных: Число Длина: ☐ Обязательное

Источник значения

Контекст: Нет

Показатель источника: ...

Параметр: 2.PURPARTICLE.NRN

Значение по умолчанию:

Обработка значения

Функция обработки значения:

Примеры:
substr(<VALUE>,1,3)
upper(<VALUE>)
<VALUE>*20/100

☒ Удалить крайние пробелы ☐ Выравнивать пробелами слева (префикс, номер)

Уникальность

Входит в уникальный: ☐ ключ 1 ☐ ключ 2 ☐ ключ 3 ☐ ключ 4 ☐ ключ 5

☐ Сгенерить уникальное значение (для обхода физического ключа уникальности)

Примечание

ОК Отмена

Рисунок 15

Рассмотрим еще порядок заполнения параметра "Код программной (непрограммной) статьи". Этот код соответствует первым четырем символам поля "Код". Эта задача решается при помощи функции `substr(<VALUE>,1,5)`

Конвертер: Параметры: Исправление

Параметр

Порядковый номер: 7

Мнемикод параметра: SCODE_ST

Наименование параметра: Код программной (непрограммной) статьи

Тип параметра: Входной

Поле представления: SCODE_ST

Тип данных: Строка Длина: 20 ☒ Обязательное

Источник значения

Контекст: Нет

Показатель источника: Код

Параметр:

Значение по умолчанию:

Обработка значения

Функция обработки значения: substr(<VALUE>,1,5)

Примеры:
substr(<VALUE>,1,3)
upper(<VALUE>)
<VALUE>*20/100

☒ Удалить крайние пробелы ☐ Выравнивать пробелами слева (префикс, номер)

Уникальность

Входит в уникальный: ☐ ключ 1 ☐ ключ 2 ☐ ключ 3 ☐ ключ 4 ☐ ключ 5

☐ Сгенерить уникальное значение (для обхода физического ключа уникальности)

Примечание

OK Отмена

Рисунок 16

Не будем рассматривать заполнение остальных параметров, так как они очевидны. Полностью настроенный алгоритм показан на рисунке 17

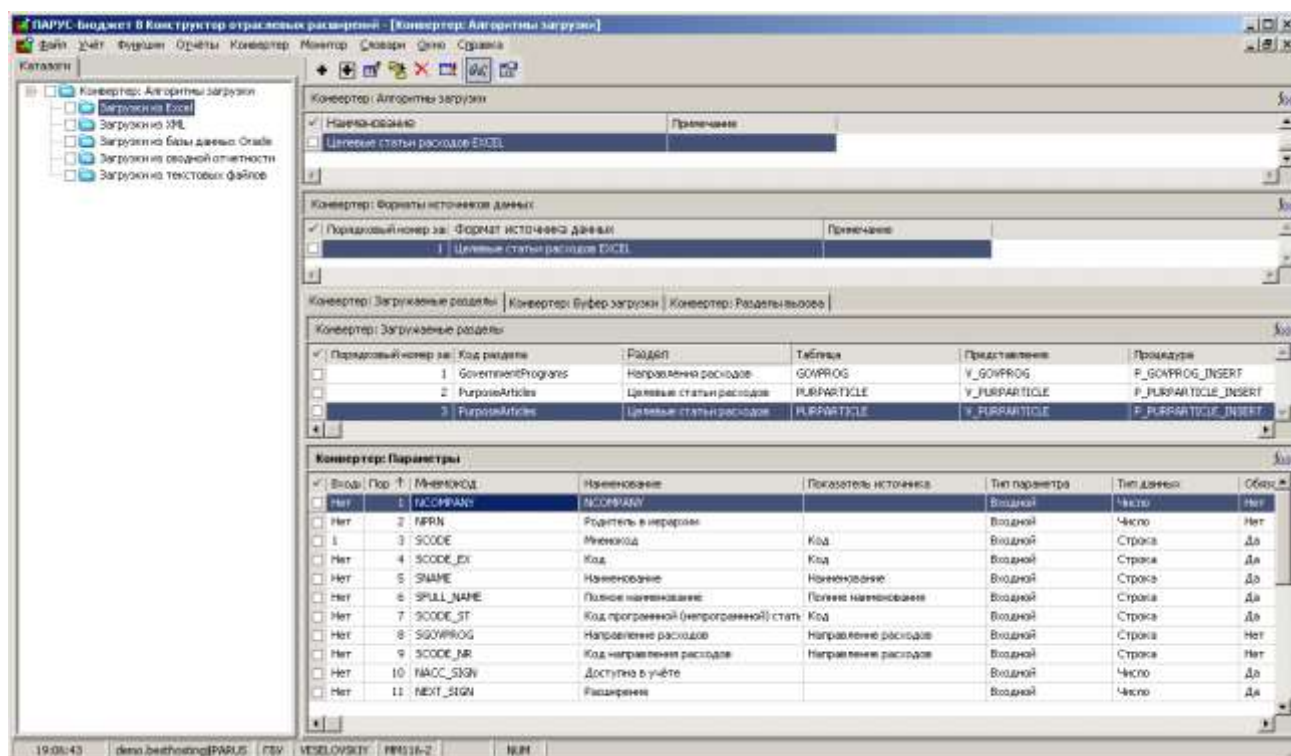


Рисунок 17

Удобно будет в дальнейшем выполнять конвертацию непосредственно из раздела "Целевые статьи расходов". Для этого нужно добавить запись в спецификацию "Разделы вызова".

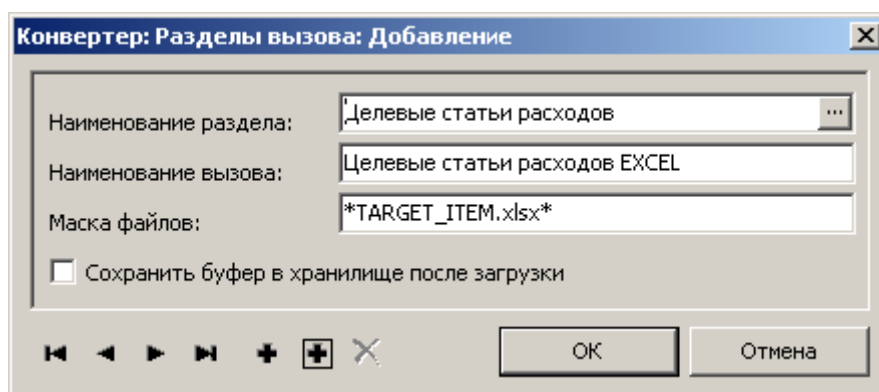


Рисунок 18

На этом настройку алгоритма загрузки можно считать законченной.

Отладка

Мы выполнили настройку таким образом, чтобы была возможность запускать загрузку Excel файла из раздела "Целевые статьи расходов". Но мы

помним, что существует режим конвертации в два этапа из раздела "Источники данных". Этот режим является единственно возможным при проведении сложных конвертаций из многих источников во многие разделы. Но и для самых простых загрузок целесообразно пользоваться именно этим режимом на этапе отладки алгоритма, так как это позволяет легче находить допущенные при настройке ошибки.

Добавление новой записи в разделе "Источники данных" – первый шаг на этапе отладки алгоритма. В качестве "типа" выбираем настроенный ранее нами формат источника.

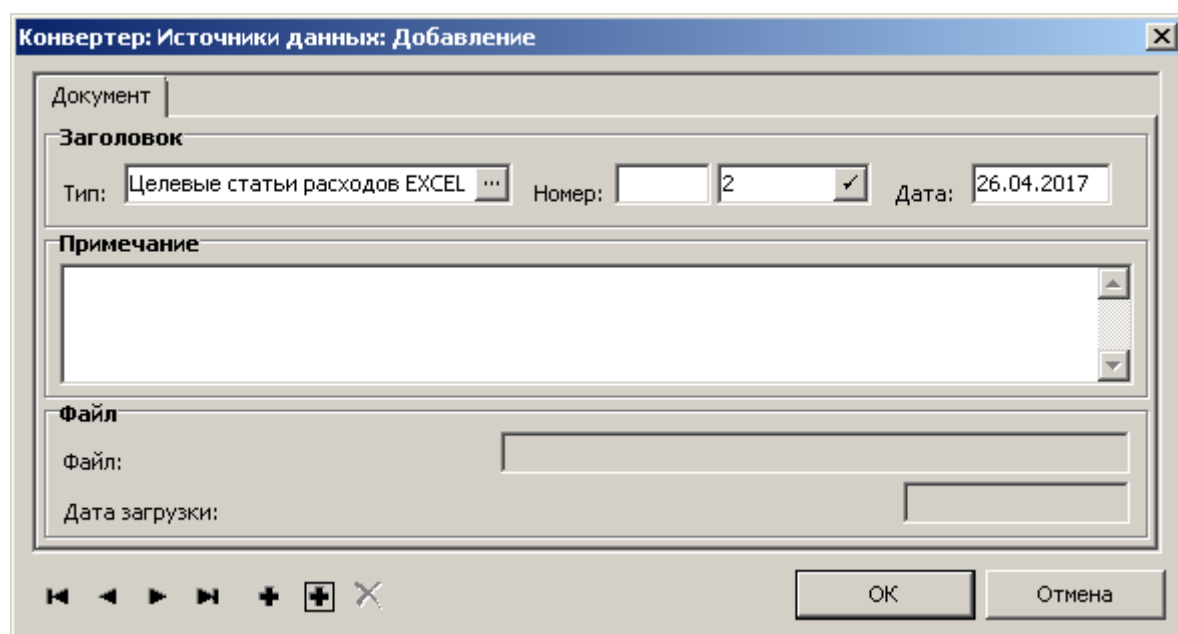


Рисунок 19

Стоя на созданной строке, выполним пользовательское приложение "Загрузить файл". На этом этапе данных из Excel загрузятся в хранилище, в соответствии с описанным форматом. Посмотреть содержание хранилища можно с помощью пункта контекстного меню "Хранилище данных".

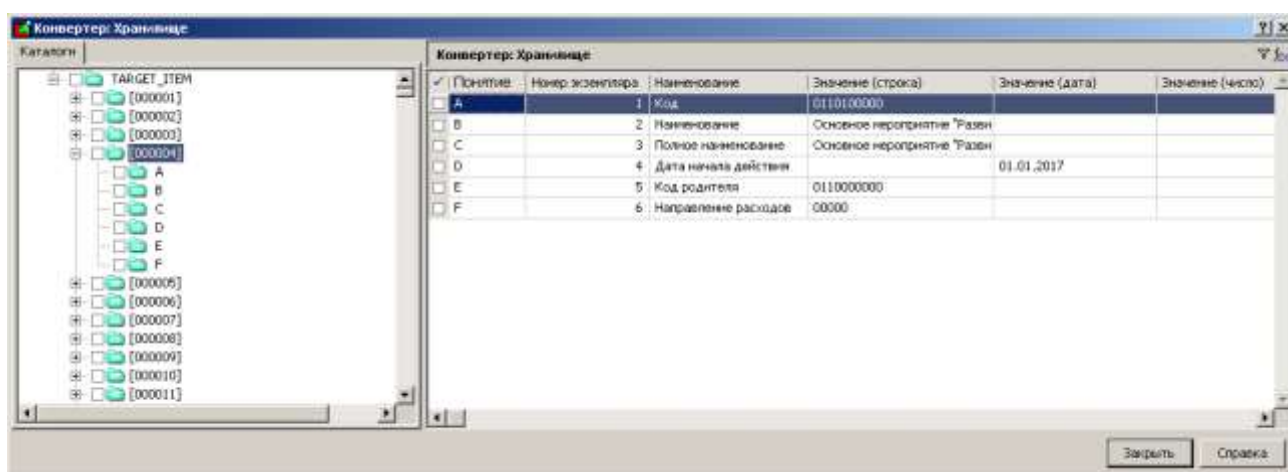


Рисунок 20

На этом этапе возможно проконтролировать визуально, насколько правильно "легли" данные в базу данных.

Далее, выполняем действие "Выполнить конвертацию". На этом этапе выполняется настроенный нами алгоритм. В случае если возникнет ошибка, ее подробное содержание можно посмотреть в "алгоритмах загрузки" в спецификации "Буфер загрузки".

После того, как алгоритм будет полностью отлажен, запись в "Источниках данных" можно удалить и пользоваться в дальнейшем загрузкой непосредственно из раздела "Целевые статьи расходов".

3.3.2 Тестовый файл с разделителями

Рассмотрим пример с загрузкой целевых статей из файла с разделителем. Загружать будем опять же целевые статьи расходов из файла TARGET_ITEM.spr, который был скачен с сайта "КейСистемс" <http://www.keysystems.ru/services/nsi/BudKlass2017.aspx>. Нетрудно заметить, что excel, который использовался ранее в примере, был создан из этого файла. Настроим теперь загрузку непосредственно из текстового источника.

Добавление Формата обмена данными

Воспользуемся действием "Создать формат".

В качестве разделителя используем символ "|". Если бы у нас было описание формата, мы бы могли вставить его в поле "Формат текстового файла" и описание создалось бы автоматически. Однако у нас описания формата нет, и нам придется его создать.

Скопируем строки из файла, хотя бы по одной каждого типа.

KS/2013.03/

TYPE/Dictionary_TARGET_ITEM/

BK|0110000000|Подпрограмма "Профилактика заболеваний и формирование здорового образа жизни. Развитие первичной медико-санитарной помощи"|Подпрограмма "Профилактика заболеваний и формирование здорового образа жизни. Развитие первичной медико-санитарной помощи"|01.01.2017|||||0100000000|||||

TOTAL/2418/

Далее, заменим значения показателей именами полей. Пустые значения заменим любыми именами, например, просто цифрами. Получим такой текст:

KS|Вексия/

TYPE|Тип/

BK|Код| Наименование| Полные наименование| Дата начала действия|5|6|7|8| Код родителя|||||

TOTAL|Всего/

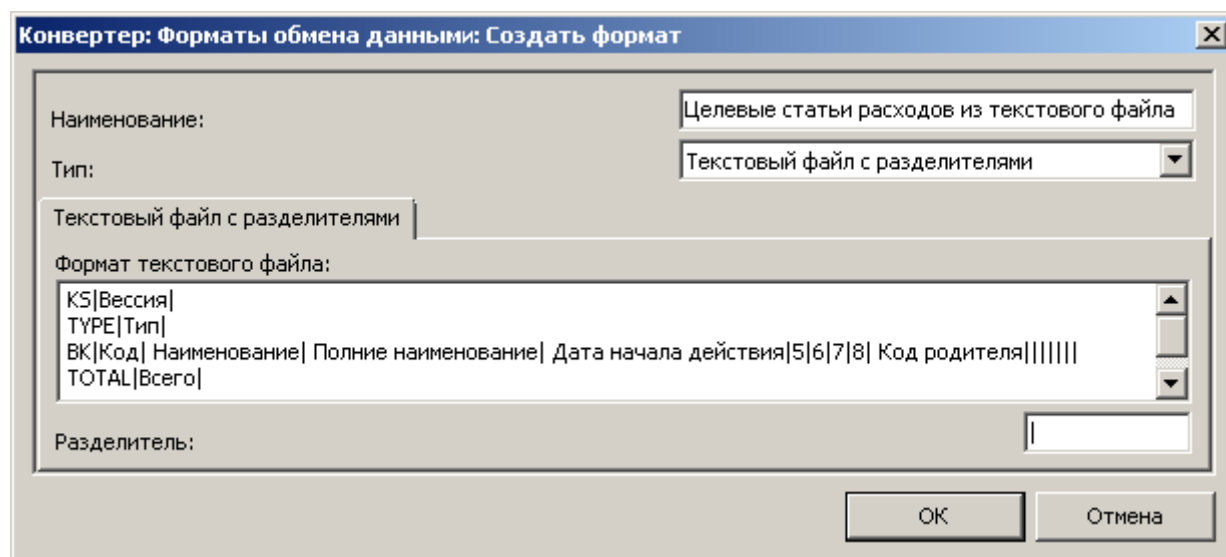


Рисунок 21

Описание формата сформировано. Структура файла описана. Для того чтобы убедиться, что описание верно, целесообразно сразу создать запись в "Источниках данных" и проверить правильность загрузки содержимого файла в "Хранилище". Обратите внимания, что для текстовых файлов типы всех полей формата "строка", даже если в них хранится число или дата.

Создание Алгоритма загрузки

Так как мы загружаем те же разделы, что в примере с Excel, алгоритм будет очень похож. Добавляем запись "Целевые статьи расходов из текстового файла", и спецификацию первого уровня, со ссылкой на формат "Целевые статьи расходов из текстового файла". Добавит также сразу спецификацию "Раздел вызова".

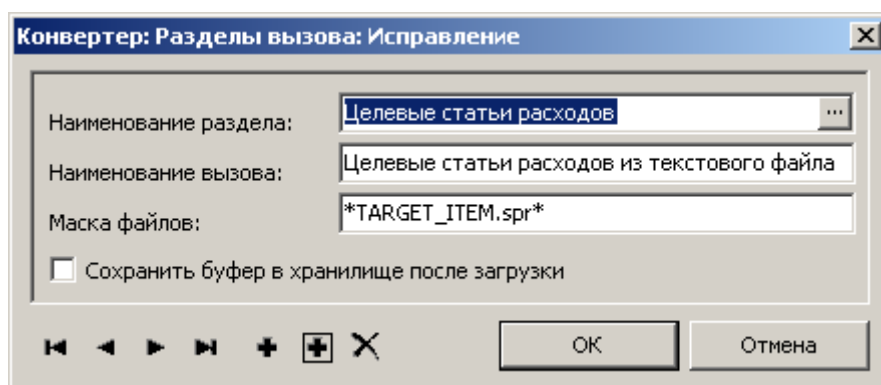


Рисунок 22

Спецификация "Загружаемы разделы" почти полностью повторяет алгоритм, созданный нами ранее для загрузки их Excel. Разница в том, что в Excel у нас было поле "Направление расходов", в которое мы брали последние 5 символов поля "Код". В текстовом поле у нас поля "Направление расходов" нет, поэтому нужно воспользоваться формулой.

Конвертер: Параметры: Исправление

Параметр

Порядковый номер: 3

Мнемокод параметра: SCODE

Наименование параметра: Мнемокод

Тип параметра: Входной

Поле представления: SCODE

Тип данных: Строка Длина: 20 ☒ Обязательное

Источник значения

Контекст: Нет

Показатель источника: Код

Параметр:

Значение по умолчанию:

Обработка значения

Функция обработки значения: substr(<VALUE>,6,5)

Примеры:
substr(<VALUE>,1,3)
upper(<VALUE>)
<VALUE>*20/100

☒ Удалить крайние пробелы ☐ Выравнивать пробелами слева (префикс, номер)

Уникальность

Входит в уникальный: ☐ ключ 1 ☐ ключ 2 ☐ ключ 3 ☐ ключ 4 ☐ ключ 5

☐ Сгенерить уникальное значение (для обхода физического ключа уникальности)

Примечание

OK Отмена

Рисунок 23

Больше нет никаких отличий в настройке алгоритма, результат настройки
виден на рисунке 24

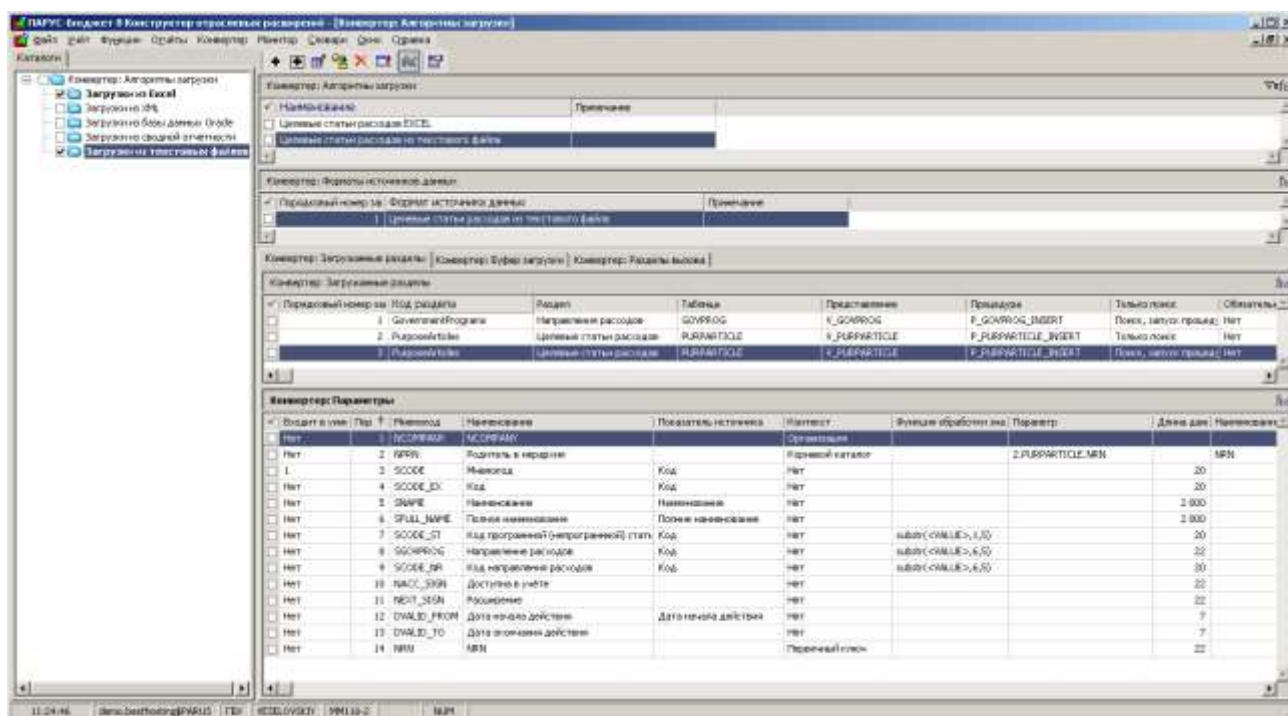


Рисунок 24

Алгоритм настроен и готов к работе. Для проверки и отладки целесообразно пользоваться действием "Выполнить конвертацию" в разделе "Источники данных". В дальнейшем удобнее загружать файл непосредственно из загружаемого раздела.

Загрузка файла

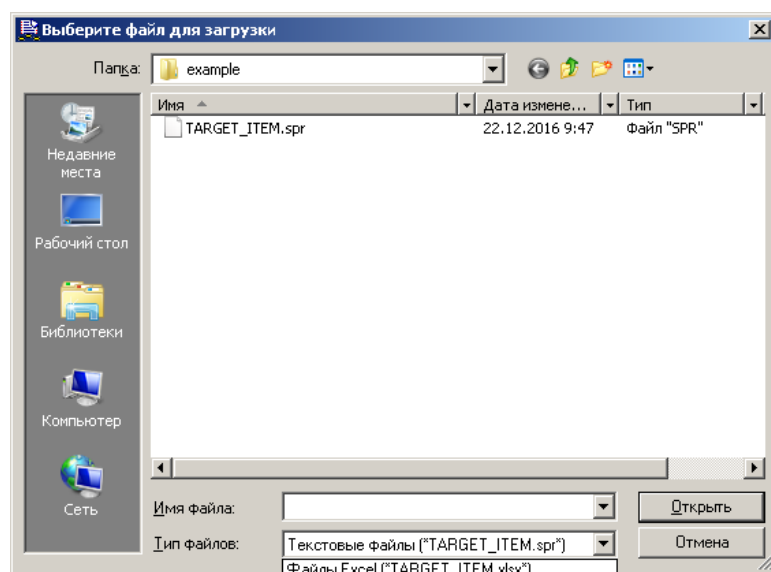


Рисунок 25

Следует обратить внимание на то, что к разделу подвязано лишь одно пользовательское приложение "Загрузить файл", а у нас существует два алгоритма по загрузке данных, привязанных к этому разделу. Система определяет, какой алгоритм использовать по маске файла.

3.3.3 Текстовый файл с полями фиксированной длины

Работа с текстовыми файлами полями фиксированной длины очень мало отличается от работы с текстовыми файлами с разделителями. Фактически, отличие только в том, что полем считается часть строки, с определенного символа начиная и определенной длины, а не часть строки между разделителями. Поэтому, чтобы не повторять ранее рассмотренные приемы, создадим новый пример с подчиненными понятиями в рамках файла.

Создадим текстовый файл DS.txt для загрузки дополнительных словарей следующего формата:

```
List,20,80,20  
Meanings,20,240
```

Это описание формата означает, что есть секция заголовка (List), с полями "Мнемокод", "Наименование" и "Формат данных" каждое из которых фиксированной длины в символах, и есть секция значений (Meanings), с полями "Значение" и "Примечание". В отличие от предыдущих примеров, *Meanings* является подчиненной по отношению к *List*. На этом примере разберем особенности использования в качестве источника данных текстовых файлов полями фиксированной длины и особенности описания иерархических форматов.

Добавление Формата обмена данными

Создадим запись "Дополнительные словари из текстового файла". С использованием действия "создать". Заполним формат нашим описанием.

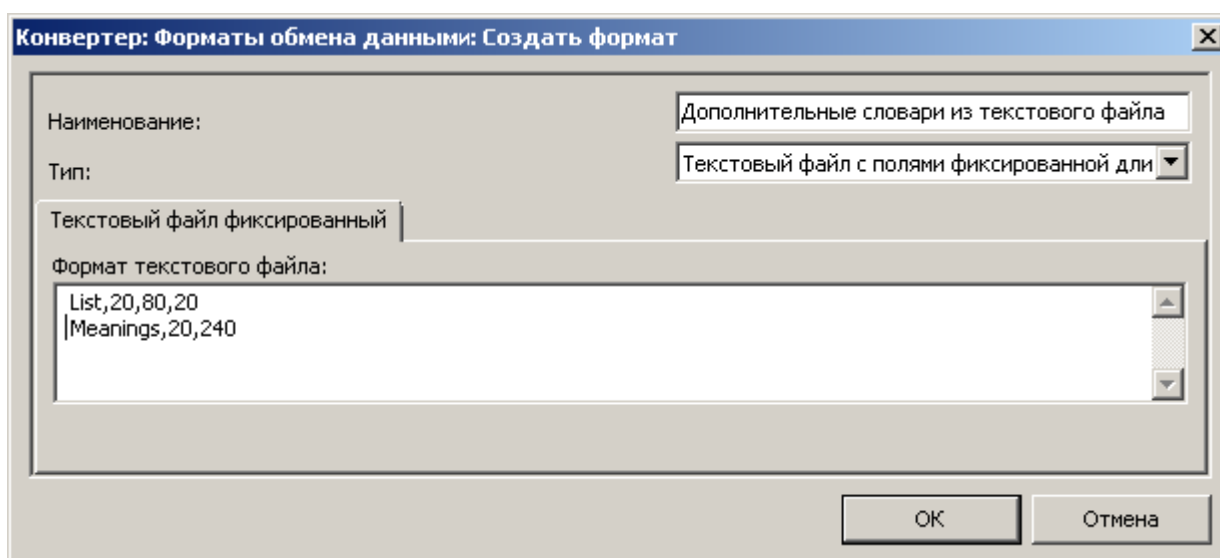


Рисунок 26

Заметим, что в "полях" в качестве идентификатора и наименование используется диапазон символов, например, "006:025". Наименования можно исправить на понятные нам понятия, чтобы было удобно работать с форматом, идентификатор изменять нельзя.

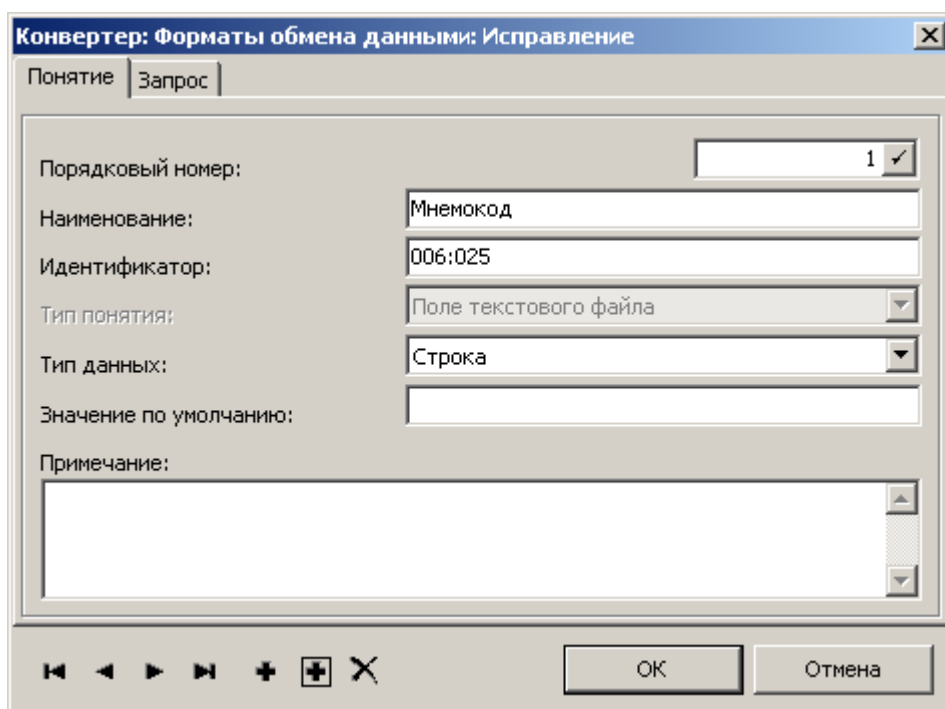


Рисунок 27

Из описания формата не следует, что формат содержит заголовок и спецификацию. Поэтому система создала понятия на одном уровне. Однако для правильной работы алгоритмов конвертации, существенно важно, что таблица Meanings является подчиненной по отношению к List. Скорректируем описание формата, перенеся понятие Meanings внутрь List.

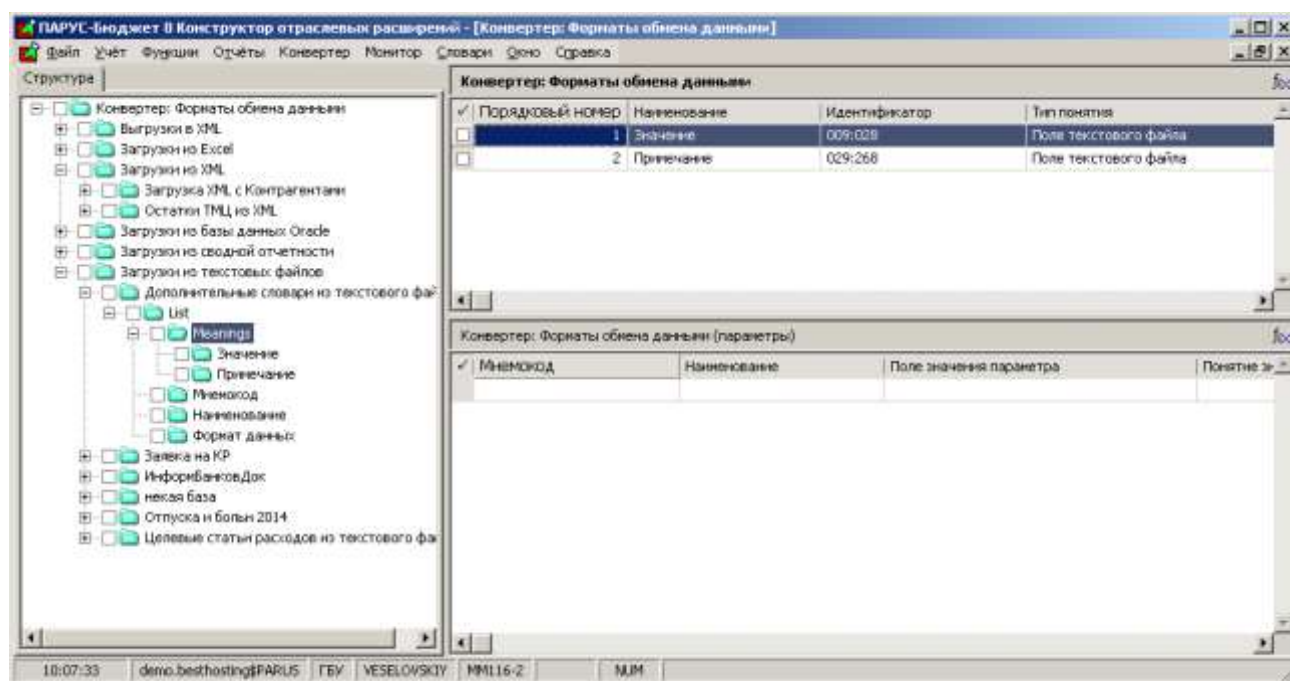


Рисунок 28

Теперь, когда подчиненность понятий выстроена правильно, можно считать процесс описание формата завершенным.

Создание Алгоритма загрузки

Добавляем запись в разделе "Дополнительные словари из текстового файла" и спецификацию первого уровня с форматом "Дополнительные словари из текстового файла". При необходимости можно также добавить вызов из раздела "Дополнительные словари". Далее, заполняем спецификацию "Загружаемые разделы" по тем же принципам, что и в предыдущих примерах.

При добавлении заголовка мы столкнулись с новой ситуацией. Поля "Мнемокод" и "Наименование" заполняются одноименными полями из источника. А вот с полем "Формат данных" чуть сложнее, так как это поле с "перечисляемым

значением". То есть, значение в источнике может быть "Строка", "Число" или "Дата". А значение параметра "Формат данных" у процедуры добавления числовое. В связи с этим, возникает два вопроса:

- 1) Как узнать, какое число соответствует какому значению?
- 2) Как заменить текстовые значения числами?

Ответ на первый вопрос, строго говоря, не такой простой. Но на практике проще всего использовать очень простой прием, который не дает стопроцентной гарантии, но чаще всего его вполне достаточно.

- заходим в раздел, в котором есть записи, которые мы хотим загружать, в нашем случае "Дополнительные словари";
- выполняем "отбор по колонке" на интересующем нас поле.

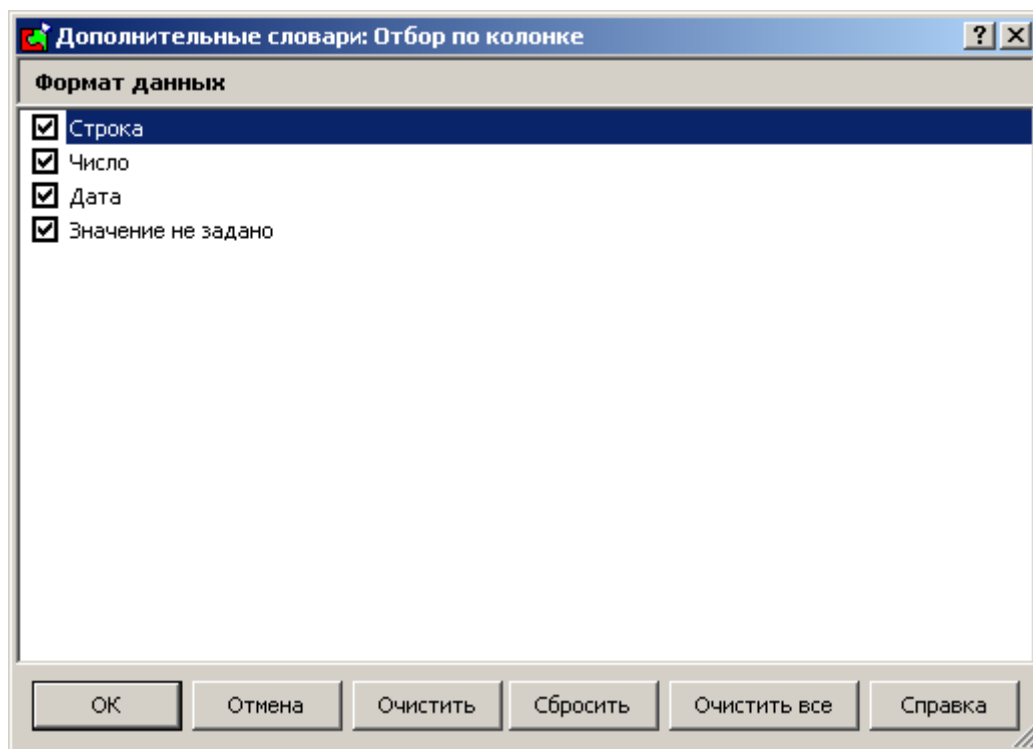


Рисунок 29

Скорей всего, значения выстроены по порядку, начиная с 0. То есть, для нашего случая формат "Строка" соответствует 0, "Число" - 1 а "Дата" – 2. Возьмем за рабочую гипотезу, что это так, проверив, разумеется, правильность загружаемых данных при отладке алгоритма.

Что же касается замены значений, то удобнее всего воспользоваться функцией "decode". Например, в нашем случае следует в поле "Функция обработки значения" написать выражение `decode(<VALUE>,'Число',1,'Дата',2,0)`. Таким образом, если значение источника "Число", оно заменится на 1, если "Дата", то на 2, а любом другом случае – 0, то есть считаем, что формат данных "строка".

Конвертер: Параметры: Исправление

Параметр

Порядковый номер: 5

Мнемикод параметра: NFORMAT

Наименование параметра: Формат данных

Тип параметра: Входящий

Поле представления: NFORMAT

Тип данных: Число Длина: 22 ☒ Обязательное

Источник значения

Контекст: Нет

Показатель источника: Формат данных

Параметр:

Значение по умолчанию: 0

Обработка значения

Функция обработки значения: `decode(<VALUE>,'Число',1,'Дата',2,0)`

Примеры:
`substr(<VALUE>,1,3)`
`upper(<VALUE>)`
`<VALUE>*20/100`

☒ Удалить крайние пробелы ☐ Выравнивать пробелами слева (префикс, номер)

Уникальность

Входит в уникальный: ☐ ключ 1 ☐ ключ 2 ☐ ключ 3 ☐ ключ 4 ☐ ключ 5

☐ Сгенерить уникальное значение (для обхода физического ключа уникальности)

Примечание

OK Отмена

Рисунок 30

При добавлении записей в раздел "Дополнительные словари (значения)" тоже есть свои особенности. Дело в том, что в разделе три разных поля:

SSTR_VALUE Строковое значение

NNUM_VALUE Числовое значение

DDATE_VALUE Значение даты

А в источнике у нас одно значение и еще есть тип в заголовке. То есть, нужно в зависимости от типа заполнять разные поля.

Первая задача определить, есть ли уже такая запись. Но мы не знаем, по какому полю искать значение, так как это зависит от формата значения. Поэтому, мы сделаем три ключа поиска.

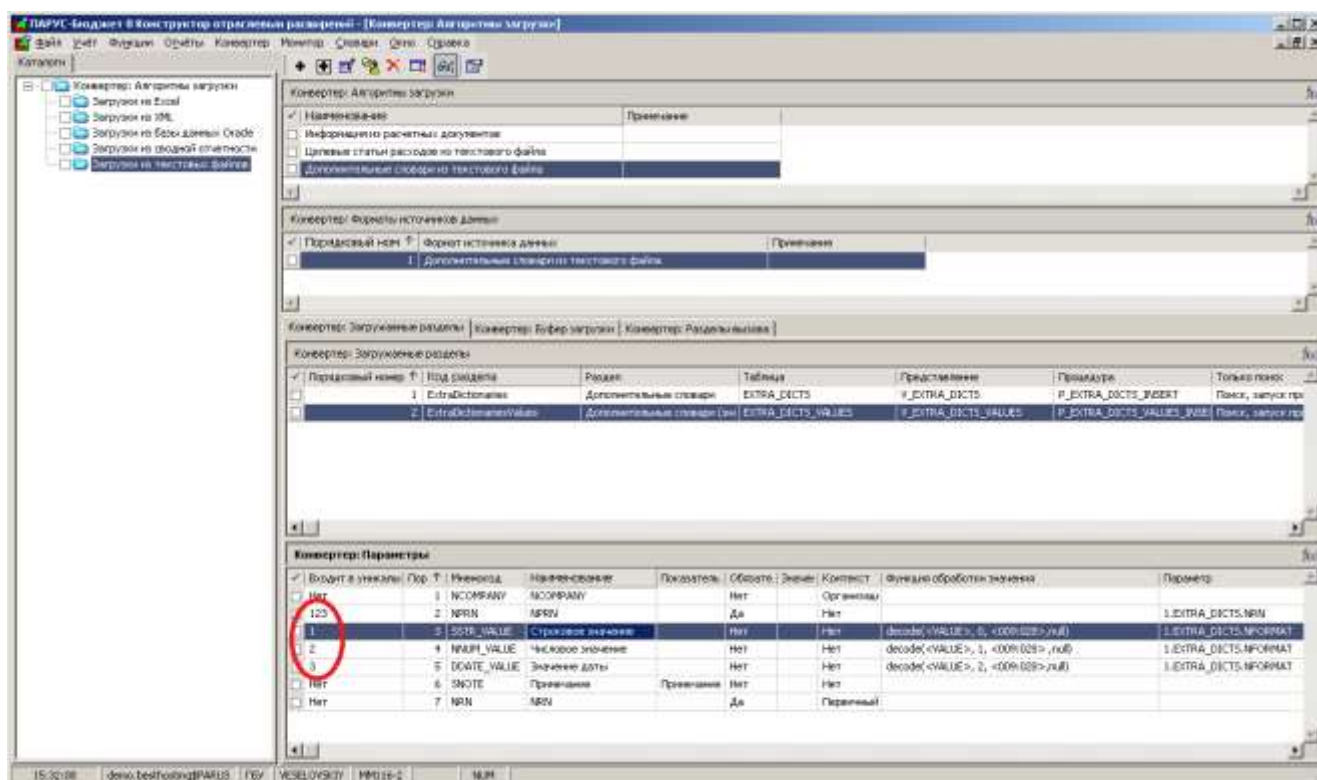


Рисунок 31

При такой настройке система будет искать запись по ключу NPRN+SSTR_VALUE, если не нашла, то по ключу NPRN+NNUM_VALUE, а если не нашла по нему, то по NPRN+DDATE_VALUE. И только если запись не была найдена по всем трем ключам, будет добавляться новая запись.

Следующая проблема, это выбрать, какое из полей SSTR_VALUE, NNUM_VALUE или DDATE_VALUE заполнять. Воспользуемся для этого уже знакомой нам функцией decode. Например:

Конвертер: Параметры: Исправление

Параметр

Параметр

Порядковый номер: 3

Мнемокод параметра: SSTR_VALUE

Наименование параметра: Строковое значение

Тип параметра: Входящий Домен: TNOTE

Поле представления: SSTR_VALUE

Тип данных: Строка Длина: 240 ☐ Обязательное

Источник значения

Контекст: Нет

Показатель источника: ...

Параметр: 1.EXTRA_DICTS.NFORMAT

Значение по умолчанию:

Обработка значения

Функция обработки значения: decode(<VALUE>, 0, <009:028>,null)

Примеры: substr(<VALUE>,1,3)
upper(<VALUE>)
<VALUE>*20/100

☒ Удалить крайние пробелы ☐ Выравнивать пробелами слева (префикс, номер)

Уникальность

Входит в уникальный: ☒ ключ 1 ☐ ключ 2 ☐ ключ 3 ☐ ключ 4 ☐ ключ 5

☐ Сгенерить уникальное значение (для обхода физического ключа уникальности)

Примечание

OK Отмена

Получается, что если значение поля NFORMAT равно 0, то параметр равен значению поля с идентификатором 009:028, то есть полю "Значение", в противном случае NULL. То есть параметр будет заполнен только при формате "строка". Аналогичная настройка выполняется и для параметров NNUM_VALUE и

DDATE_VALUE. Соответственно, формула в них будет `decode(<VALUE>, 1, <009:028>, null)` и `decode(<VALUE>, 2, <009:028>, null)`.

На этом настройку алгоритма можно считать законченной, загрузка выполняется стандартным способом, и мы не будем на этом останавливаться.

3.3.4 Файл XML

Завершаем описание работы с файлами различного формата, пожалуй, самым популярным в настоящее время текстовым форматом XML. В качестве примера возьмем "ТМЦ.xml". Файл получен путем выгрузки данных из 1С и содержит остатки ТМЦ.

Добавление Формата обмена данными

XML файл содержит в себе всю необходимую для описания формата информацию. Поэтому создание формата предельно приятное, мы только создаем строку "Остатки ТМЦ из XML" в разделе и указываем файл, по которому сформировать описание.

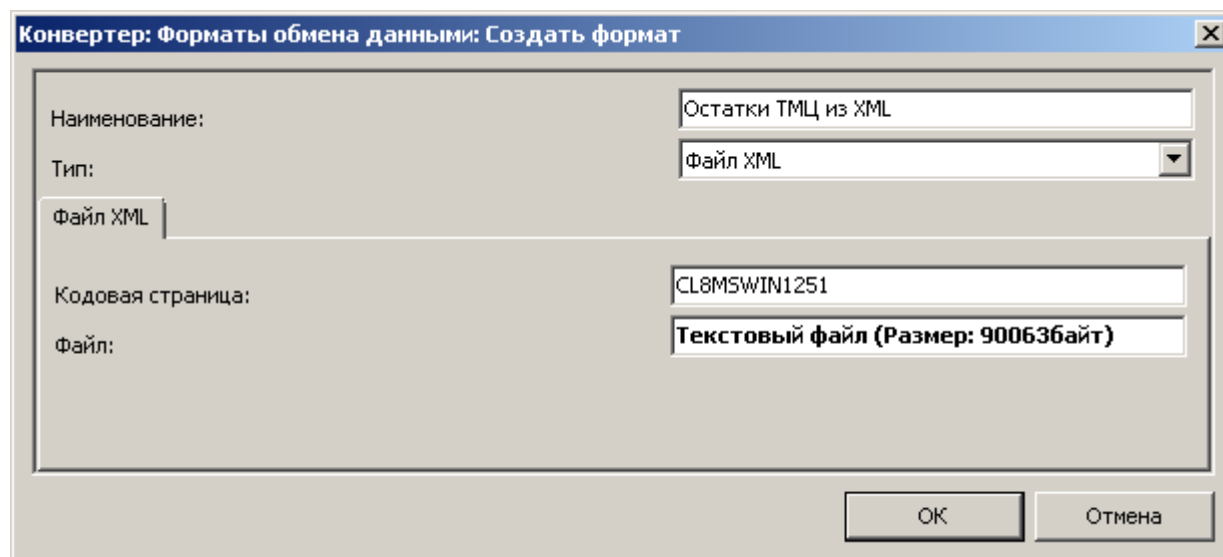


Рисунок 32

Все описание формируется автоматически и не требует правки. Единственное, что стоит сделать, это поменять типы данных у некоторых полей. Это позволит нам при создании алгоритмов загрузки не думать о таких вещах, как

формат представления даты или разделитель дробной части в числовых полях. В частности, у тагов "КОЛИЧЕСТВО" и "СУММА" имеет смысл тип "строка" поменять на "число", а у "ПЕРИОДС" и "ПЕРИОДПО" на "дату".

Для проверки правильности описания формата можно создать запись в "Источниках данных" и загрузить содержимое файла в хранилище.

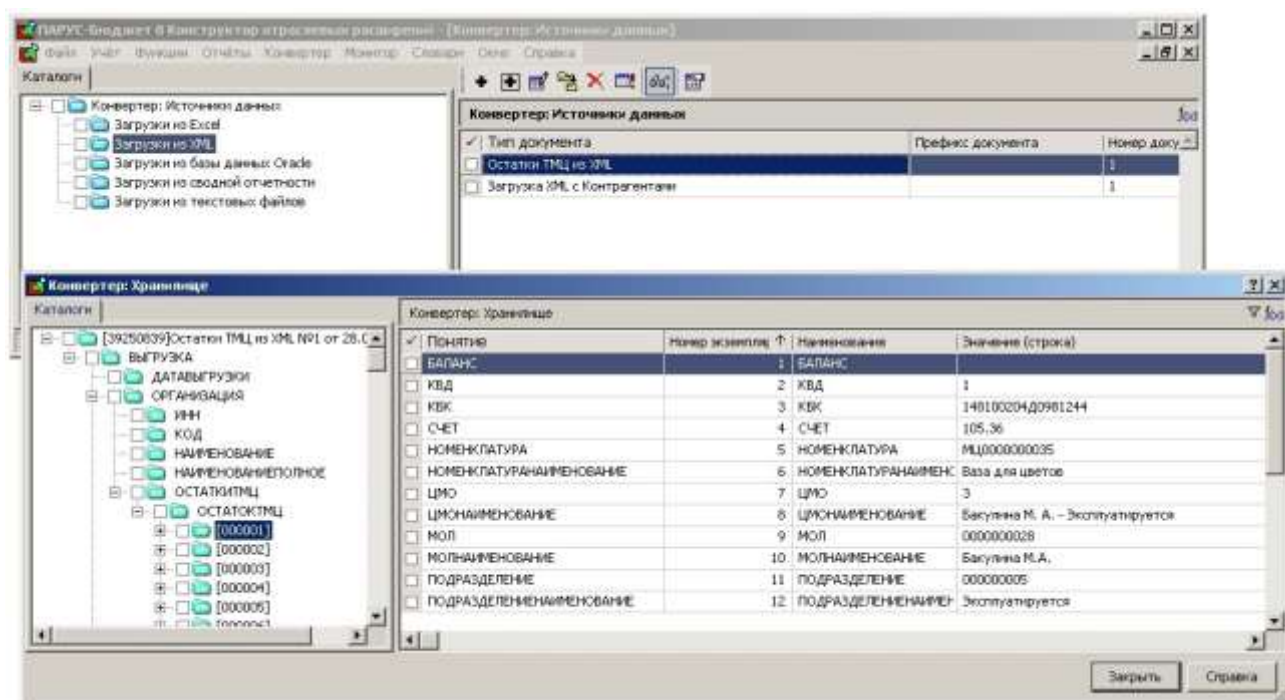


Рисунок 33

Создание Алгоритма загрузки

Как обычно, начнем с добавления заголовка и формата "Остатки ТМЦ из XML". Задача загрузки остатков, как правило, является разовой. Поэтому заполнять "Разделы вызова" нецелесообразно, логичнее пользоваться загрузкой из "Источников данных". Остановимся на заполнении "Загружаемых разделов".

Сама по себе технология заполнения параметров загружаемых разделов ничем не отличается от загрузки из файлов других форматов. Поэтому мы не будем подробно описывать последовательность действий, а ограничимся описанием нескольких приемов, с которыми мы не сталкивались ранее. Причем, эти особенности скорее связаны не с форматом XML как таковым, а с характером

загружаемых данных.

В разделе "Остатки товарно-материальных ценностей" существуют ссылки на синтетические и аналитические счета, номенклатор и контрагентов. Будем считать, что синтетические счета у нас в базе все есть. Аналитику, номенклатуру и контрагентов будем добавлять "на лету". Кроме того, в разделе остатки хранятся как в заголовке, так и в спецификации в разрезе аналитических счетов.

№	Код раздела	Раздел	Таблица	Представление	Процедура	Только поиск	Таблица источника
1	AGNLIST	Контрагенты	AGNLIST	V_AGNLIST	P_AGNLIST_INSERT	Поиск, запуск процедуры (если не нашли) для вставки	ОСТАТОКТМЦ
2	Nomenclator	Номенклатор	DICNOMNS	V_DICNOMNS	P_DICNOMNS_INSERT	Поиск, запуск процедуры (если не нашли) для вставки	ОСТАТОКТМЦ
3	AccountsPlan	План счетов	DICACCS	V_DICACCS	P_DICACCS_INSERT	Только поиск	ОСТАТОКТМЦ
4	AnalyticPlan	План счетов (аналитические счета)	DICANLS	V_DICANLS	P_DICANLS_INSERT	Поиск, запуск процедуры (если не нашли) для вставки	ОСТАТОКТМЦ
5	ValuesRemnants	Остатки товарно-материальных ценностей	VALREMNS	V_VALREMNS	P_VALREMNS_INSERT	Поиск, запуск процедуры (если не нашли) для вставки	ОСТАТОКТМЦ
6	ValuesRemnants	Остатки товарно-материальных ценностей	VALREMNS	V_VALREMNS	P_VALREMNS_UPDATE	Поиск, запуск процедуры (если нашли) для изменения	ОСТАТОКТМЦ
7	ValuesRemnantsAnalytics	Остатки товарно-материальных ценностей (аналитика)	VALREMNSANL	V_VALREMNSANL	P_VALREMNSANL_INSERT	Поиск, запуск процедуры (если не нашли) для вставки	ОСТАТОКТМЦ

Таблица 3

Таким образом, как мы видим по таблице 3, мы вызываем 7 разделов. Первые 4 – справочники. Причем любые записи кроме синтетического счета, мы добавляем, если не нашли подходящее значение.

В таблице 4 укажем, для примера все параметры, заполняемые для раздела "Контрагенты". Обратите внимания, что в связи с тем, что в источнике нет мнемкода, в качестве ключа для поиска мы используем полное наименование. А в качестве мнемкода берем первые 20 символов наименования и обеспечиваем уникальность полученного значения.

Входит в уникальный ключ	Мнемкод	Наименование	Показатель источника	Обязательное	Контекст	Функция обработки значения	Сгенерить уникальное значение
Нет	NCOMPANY	NCOMPANY		Нет	Организация		Нет
Нет	CRN	CRN		Да	Корневой каталог		Нет
Нет	AGNABBR	Мнемкод	МОЛНАИМЕНОВАНИЕ	Да	Нет	substr(<VALUE>,1,20)	Да
Нет	AGNTYPE	Тип		Да	Нет		Нет
1	AGNNAME	Наименование	МОЛНАИМЕНОВАНИЕ	Да	Нет		Нет

Входит в уникальный ключ	Мнемокод	Наименование	Показатель источника	Обяза тельн ое	Контекст	Функция обработки значения	Сгенерить уникальное значение
Нет	AGNIDNUMB	Идентификационный номер налогоплательщика		Нет	Нет		Нет
Нет	ECONCODE	Код ОКОНХ (только для юр. лиц)		Нет	Нет		Нет
Нет	ORGCODE	Код ОКПО		Нет	Нет		Нет
Нет	AGNFAMILYN AME	Фамилия		Нет	Нет		Нет
Нет	AGNFIRSTNA ME	Имя		Нет	Нет		Нет
Нет	AGNLASTNA ME	Отчество		Нет	Нет		Нет
Нет	AGNFAMILYN AME_TO	Фамилия (кому)		Нет	Нет		Нет
Нет	AGNFIRSTNA ME_TO	Имя (кому)		Нет	Нет		Нет
Нет	AGNLASTNA ME_TO	Отчество (кому)		Нет	Нет		Нет
Нет	AGNFAMILYN AME_FR	Фамилия (от кого)		Нет	Нет		Нет
Нет	AGNFIRSTNA ME_FR	Имя (от кого)		Нет	Нет		Нет
Нет	AGNLASTNA ME_FR	Отчество (от кого)		Нет	Нет		Нет
Нет	AGNFAMILYN AME_AC	Фамилия (винительный падеж)		Нет	Нет		Нет
Нет	AGNFIRSTNA ME_AC	Имя (винительный падеж)		Нет	Нет		Нет
Нет	AGNLASTNA ME_AC	Отчество (винительный падеж)		Нет	Нет		Нет
Нет	AGNFAMILYN AME_ABL	Фамилия (творительный падеж)		Нет	Нет		Нет
Нет	AGNFIRSTNA ME_ABL	Имя (творительный падеж)		Нет	Нет		Нет
Нет	AGNLASTNA ME_ABL	Отчество (творительный падеж)		Нет	Нет		Нет
Нет	EMP	Сотрудник		Да	Нет		Нет
Нет	EMPPOST	Должность		Нет	Нет		Нет
Нет	EMPPOST_FR OM	Должность (от кого)		Нет	Нет		Нет
Нет	EMPPOST_TO	Должность (кому)		Нет	Нет		Нет
Нет	EMPPOST_AC	Должность (винительный падеж)		Нет	Нет		Нет
Нет	EMPPOST_AB L	Должность (творительный падеж)		Нет	Нет		Нет
Нет	AGNBURN	Дата рождения физического лица		Нет	Нет		Нет
Нет	PHONE	Телефон 1		Нет	Нет		Нет
Нет	PHONE2	Телефон 2		Нет	Нет		Нет
Нет	FAX	Факс		Нет	Нет		Нет
Нет	TELEX	Телекс		Нет	Нет		Нет
Нет	MAIL	Электронная почта		Нет	Нет		Нет
Нет	IMAGE	Изображение		Нет	Нет		Нет
Нет	DDISCDATE	Дата оформления дисконта		Нет	Нет		Нет
Нет	AGN_COMME NT	Примечание		Нет	Нет		Нет
Нет	NSEX	Пол		Да	Нет		Нет
Нет	SPENSION_N BR	Номер пенсионного свидетельства		Нет	Нет		Нет
Нет	SMEDPOLICY _SER	Серия полиса ОМС		Нет	Нет		Нет
Нет	SMEDPOLICY _NUMB	Номер полиса ОМС		Нет	Нет		Нет
Нет	SPROPFORM	Форма собственности		Нет	Нет		Нет
Нет	SREASON_C ODE	КПП (код причины постановки на учет)		Нет	Нет		Нет
Нет	NRESIDENT_ SIGN	Нерезидент		Да	Нет		Нет
Нет	STAXPSTATU S	Бюджетный статус налогоплательщика		Нет	Нет		Нет

Входит в уникальный ключ	Мнемокод	Наименование	Показатель источника	Обяза тельн ое	Контекст	Функция обработки значения	Сгенерить уникальное значение
Нет	SOGRN	ОГРН - основной государственный регистрационный номер		Нет	Нет		Нет
Нет	SPRFMLSTS	Семейное положение		Нет	Нет		Нет
Нет	SPRNATION	Национальность		Нет	Нет		Нет
Нет	SCITIZENSHIP	Гражданство		Нет	Нет		Нет
Нет	ADDR_BURN	Место рождения		Нет	Нет		Нет
Нет	SPRMLREL	Отношение к военной службе		Нет	Нет		Нет
Нет	SOKATO	Код OKATO		Нет	Нет		Нет
Нет	SPFR_NAME	Наименование ТО ПФР		Нет	Нет		Нет
Нет	DPFR_FILL_DATE	Дата заполнения заявления		Нет	Нет		Нет
Нет	DPFR_REG_DATE	Дата регистрации		Нет	Нет		Нет
Нет	SPFR_REG_NUMBER	Номер регистрации		Нет	Нет		Нет
Нет	SFULLNAME	Полное наименование		Нет	Нет		Нет
Нет	SOKFS	Код ОКФС		Нет	Нет		Нет
Нет	SOKOPF	Код ОКОПФ		Нет	Нет		Нет
Нет	STFOMS	Регистрационный номер в ТФОМС		Нет	Нет		Нет
Нет	SFSS_REG_NUMBER	Регистрационный номер в ФСС		Нет	Нет		Нет
Нет	SFSS_SUBCODE	Код подчинённости ФСС		Нет	Нет		Нет
Нет	NCOEFFIC	Районный коэффициент		Да	Нет		Нет
Нет	DAGNDEATH	Дата смерти		Нет	Нет		Нет
Нет	NOLD_RN	NOLD_RN		Нет	Нет		Нет
Нет	SOKTMO	Код ОКТМО		Нет	Нет		Нет
Нет	SINN_CITIZENSHIP	ИНН в стране регистрации		Нет	Нет		Нет
Нет	DTAX_REG_DATE	Дата постановки на учёт в налоговом органе		Нет	Нет		Нет
Нет	SORIGINAL_NAME	Оригинальное наименование		Нет	Нет		Нет
Нет	NIND_BUSINESSMAN	Индивидуальный предприниматель		Да	Нет		Нет
Нет	SFNS_CODE	Код ФНС		Нет	Нет		Нет
Нет	SCTZNSHP_TYPE	Тип гражданства		Нет	Нет		Нет
Нет	NRN	NRN		Да	Первичный ключ		Нет

Таблица 4

Для номенклатора у нас в источнике есть значения для мнемокода и наименования, но не хватает целого ряда обязательных полей. Продеться заполнить их одинаковыми значениями по умолчанию.

Входит в уникальный ключ	Мнемокод	Наименование	Показатель источника	Обязательное	Значение по умолчанию
1	SNOMEN_CODE	Мнемокод	НОМЕНКЛАТУРА	Да	
Нет	SNOMEN_NAME	Наименование	НОМЕНКЛАТУРАНАИМЕНОВАНИЕ	Да	
Нет	SMN_NAME	Международное непатентованное наименование		Нет	
Нет	SORIGINAL_NAME	Оригинальное наименование		Нет	
Нет	SUMEAS_MAIN	Основная единица измерения		Да	шт
Нет	SUMEAS_ALT	Дополнительная единица измерения		Нет	

Входит в уникальный ключ	Мнемокод	Наименование	Показатель источника	Обязательное	Значение по умолчанию
Нет	NEQUAL	Коэффициент пересчёта		Да	0
Нет	NSIGN_ACNT	Доступен в учёте		Да	1
Нет	NSIGN_DOCS	Доступен в документах		Да	1
Нет	SGROUP_CODE	Группа		Да	Общая
Нет	STAX_GROUP	Налоговая группа		Нет	

Таблица 5

Счета мы не добавляем, только ищем. Необходимо только преобразовать счет в формат "Паруса" из формата, используемого в 1С функцией <КВД>||substr(<СЧЕТ>,1,3)||substr(<СЧЕТ>,5,2)||'000'.

Например, если в источнике

КВД 1

СЧЕТ 105.36

то в Парусе в плане счетов будет счет "110536000".

Аналитический счет первого уровня ищем по совпадению с полем КБК, в случае, если такого нет, добавляем.

Далее два раза подряд вызываем раздел "Остатки товарно-материальных ценностей", в первый раз для добавления записи, заполняя количество и сумму нулевыми значениями (см. таблицу 6).

Ключ	Мнемокод	Наименование	Показатель источника	Контекст	Функция обработки значения	Параметр
1	NCOMPANY	NCOMPANY		Организация		
Нет	SJUR_PERS	Принадлежность		Основное юридическое лицо		
1	DREMN_DATE	Остаток на...	ПЕРИОДС	Нет		
1	SACCOUNT	Счёт		Нет	<КВД> substr(<СЧЕТ>,1,3) substr(<СЧЕТ>,5,2) '000'	
Нет	SBALUNIT	ПБЕ		Нет		
Нет	SCURRENCY	Валюта		Базовая валюта		
1	SAGENT	МОЛ		Нет		1.AGNLIST.AGNABBR
1	SNOMEN	Номенклатура	НОМЕНКЛАТУРА	Нет		
Нет	SPARTNO	Номер партии		Нет		
Нет	DINDATE	Дата поступления		Нет		
Нет	NACNT_REMN_SUM	Сумма (бухгалтерская оценка)		Нет		
Нет	NACNT_REMN_BASE_SUM	Сумма эквивалента (бухгалтерская оценка)		Нет		
Нет	NACNT_REMN_ACC_SUM	Сумма в валюте отчётности (бухгалтерская оценка)		Нет		
Нет	NACNT_REMN_QUANT	Количество (основная ЕИ, бухгалтерская оценка)		Нет		

Ключ	Мнемокод	Наименование	Показатель источника	Контекст	Функция обработки значения	Параметр
Нет	NACNT_ALT_QUANT	Количество (дополнительная ЕИ, бухгалтерская оценка)		Нет		
Нет	NCTRL_REMN_SUM	Сумма (управленческая оценка)		Нет		
Нет	NCTRL_REMN_BASE_SUM	Сумма эквивалента (управленческая оценка)		Нет		
Нет	NCTRL_REMN_ACC_SUM	Сумма в валюте отчётности (управленческая оценка)		Нет		
Нет	NCTRL_REMN_QUANT	Количество (основная ЕИ, управленческая оценка)		Нет		
Нет	NCTRL_ALT_QUANT	Количество (дополнительная ЕИ, управленческая оценка)		Нет		
Нет	NRN	NRN	СЧЕТ	Первичный ключ		

Таблица 6

Второй раз вызываем уже процедуру исправления записи, прибавляя к текущим значениям количество и сумму (см. таблицу 7).

Ключ	Мнемокод	Наименование	Показатель источника	Контекст	Функция обработки значения	Параметр
1	NCOMPANY	NCOMPANY		Организация		
Нет	NRN	NRN		Первичный ключ		
Нет	SJUR_PERSON	Принадлежность		Основное юридическое лицо		
1	DREMNDATE	Остаток на...	ПЕРИОДС	Нет		
1	SACCOUNT	Счёт		Нет	<КВД> substr(<СЧЕТ>,1,3) substr(<СЧЕТ>,5,2) '000'	
Нет	SBALUNIT	ПБЕ		Нет		
Нет	SCURRENCY	Валюта		Базовая валюта		
1	SAGENT	МОЛ		Нет		1.AGNLIST.AGNABBR
1	SNOMEN	Номенклатура	НОМЕНКЛАТУРА	Нет		
Нет	SPARTNO	Номер партии		Нет		
Нет	DINDATE	Дата поступления		Нет		
Нет	NACNT_REMN_SUM	Сумма (бухгалтерская оценка)		Нет	nvl(<VALUE>,0)+nvl(<СУММА>,0)	5.VALREMNS.NACNT_REMN_SUM
Нет	NACNT_REMN_BASE_SUM	Сумма эквивалента (бухгалтерская оценка)		Нет	nvl(<VALUE>,0)+nvl(<СУММА>,0)	5.VALREMNS.NACNT_REMN_BASE_SUM
Нет	NACNT_REMN_ACC_SUM	Сумма в валюте отчётности (бухгалтерская оценка)		Нет		
Нет	NACNT_REMN_QUANT	Количество (основная ЕИ, бухгалтерская оценка)		Нет	nvl(<VALUE>,0)+nvl(<КОЛИЧЕСТВО>,0)	5.VALREMNS.NACNT_REMN_QUANT
Нет	NACNT_ALT_QUANT	Количество (дополнительная ЕИ, бухгалтерская оценка)		Нет		
Нет	NCTRL_REMN_SUM	Сумма (управленческая оценка)		Нет		
Нет	NCTRL_REMN_BASE_SUM	Сумма эквивалента (управленческая оценка)		Нет		
Нет	NCTRL_REMN_ACC_SUM	Сумма в валюте отчётности (управленческая оценка)		Нет		
Нет	NCTRL_REMN_QUANT	Количество (основная ЕИ, управленческая оценка)		Нет		
Нет	NCTRL_ALT_QUANT	Количество (дополнительная ЕИ, управленческая оценка)		Нет		

Таблица 7

Почему именно так? Мы предполагаем, что в источнике может быть более одной строк по одному МОЛ и наименованию ТМЦ, но по разным КБК. То есть, в спецификации может быть более одной строки. В результате такого подхода числовые значения заголовком остатков накапливаются. Кстати, из-за такой реализации повторная загрузка тех же остатков приведет к неправильному результату: суммы и количества удвоятся. Следовательно, необходимо удалить остатки перед повторной загрузкой.

Последний раздел в алгоритме – "Остатки товарно-материальных ценностей (аналитика)", то есть спецификация остатков в разрезе КБК.

Ключ	Мнемокод	Наименование	Показатель источника	Контекст	Параметр
1	NCOMPANY	NCOMPANY		Организация	
1	NPRN	NPRN		Нет	5.VALREMNS.NRN
1	SANALYTIC1	Аналитика 1 уровня	КБК	Нет	
Нет	SANALYTIC2	Аналитика 2 уровня		Нет	
Нет	SANALYTIC3	Аналитика 3 уровня		Нет	
Нет	SANALYTIC4	Аналитика 4 уровня		Нет	
Нет	SANALYTIC5	Аналитика 5 уровня		Нет	
Нет	NACNT_REMN_QUANT	Количество (основная ЕИ)	КОЛИЧЕСТВО	Нет	
Нет	NACNT_ALT_QUANT	Количество (дополнительная ЕИ)		Нет	
Нет	NACNT_REMN_SUM	Сумма (бухгалтерская оценка)	СУММА	Нет	
Нет	NACNT_REMN_BASE_SUM	Сумма эквивалента (бухгалтерская оценка)	СУММА	Нет	
Нет	NACNT_REMN_ACC_SUM	Сумма в валюте отчётности (бухгалтерская оценка)		Нет	
Нет	NCTRL_REMN_QUANT	Количество (основная ЕИ, управленческая оценка)		Нет	
Нет	NCTRL_ALT_QUANT	Количество (дополнительная ЕИ, управленческая оценка)		Нет	
Нет	NCTRL_REMN_SUM	Сумма (управленческая оценка)		Нет	
Нет	NCTRL_REMN_BASE_SUM	Сумма эквивалента (управленческая оценка)		Нет	
Нет	NCTRL_REMN_ACC_SUM	Сумма в валюте отчётности (управленческая оценка)		Нет	
Нет	NRN	NRN		Первичный ключ	

Таблица 8

Теперь можно конвертировать остатки из раздела "Источники данных".

3.3.5 База данных Oracle

Введение

До сих пор мы рассматривали загрузку информации из внешних файлов. Такая загрузка всегда происходит в два этапа:

- загрузка данных в хранилище;

- конвертация данных в разделы ПАРУС 8.

Даже если мы пользуемся загрузкой данных непосредственно из раздела, и это выглядит для пользователя как одно простое действие, все равно эти два этапа существуют, просто они делаются автоматически (точнее, даже три этапа, так как хранилище потом удаляется, и это можно считать отдельным этапом). При загрузке данных из таблиц Oracle, этапа загрузки данных в хранилище нет, так как данные уже находятся в базе. По большому счету, это единственное отличие от ранее описанных форматов. Если бы мы сейчас рассмотрели пример, полностью аналогичный примеру загрузки целевых статей, описанному в разделе "Сквозной пример" → "Книга Excel", только вместо книги Excel у нас бы была таблица базы с аналогичными полями, этот пример почти ничем бы не отличался от варианта с Excel. Однако, на практике, загрузка из Oracle, как правило, происходит не из одной таблицы, а из реляционной базы данных, то есть это полноценная конвертация из многих таблиц логически связанных между собой. Например, при переходе с Парус 7 на Парус 8 конвертация проходит так:



Рисунок 34

При решении подобных задач возникает необходимость сохранения "внешних RN" в базе данных. То есть нужна отдельная таблица, ставящая в соответствие "внешний RN" и RN создаваемых записей. Во-первых, это необходимо потому, что часто RN является единственным ключом, связывающим две таблицы. Например, спецификация документа связана с заголовком только по RN. А во-вторых, при проведении конвертации полезно сохранять внешние RN для того, чтобы можно было подгружать данные и знать при этом, какие записи уже были загружены, а какие нет.

В качестве примера загрузки данных из таблиц, рассмотрим загрузку данных дополнительного словаря из одной компании в другую. Разумеется, это не совсем та задача, для которой предназначен «Конвертер», зато этот пример позволит нам не создавать «чужих» таблиц в нашей базе данных, но показать, тем не менее, как использовать раздел с внешними ключами.

Добавление Формата обмена данными

Для добавления формата, как обычно, воспользуемся действием "Создать формат", указав таблицы (или выюхи) заголовков и значений дополнительных словарей (V_EXTRA_DICTS; V_EXTRA_DICTS_VALUES)

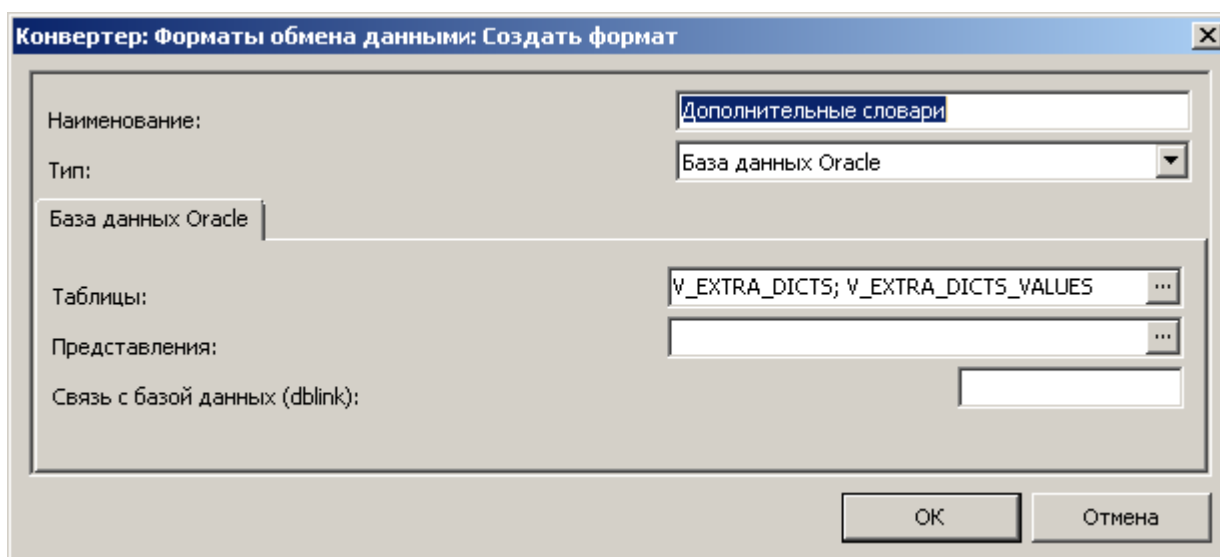


Рисунок 35

Формат сформируется автоматически. В дальнейшем идентификаторы полей можно было бы заменить на более понятные, русские, чтобы легче было настраивать алгоритм загрузки. Например, SCODE заменить на «мнемокод».

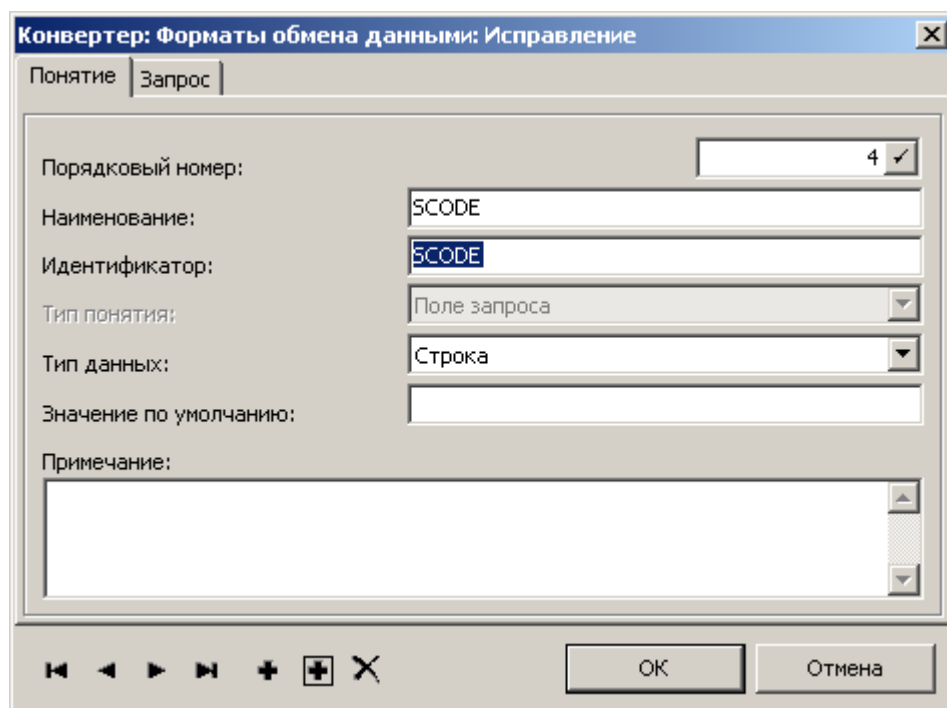


Рисунок 36

Однако мы не будем этого делать, так как алгоритм у нас будет очень простой, и смысла тратить время на описание полей никакого нет. Таким образом, описание формата закончено.

Создание Алгоритма загрузки

Добавим алгоритм для формата «Дополнительные словари». Он будет состоять из двух разделов, собственно «Дополнительные словари» и «Дополнительные словари (значения)». Начнем с заголовка. Заполнение всех полей достаточно стандартно. Обязательно заполнить «условие выполнения», ведь мы хотим перегрузить не все словари. Условие «<SCODE>='ПроверкаОстатковСч_' and <NVERSION>=107389» означает, что мы перегрузим словарь с мнемокодом «ПроверкаОстатковСч» из версии «107389».

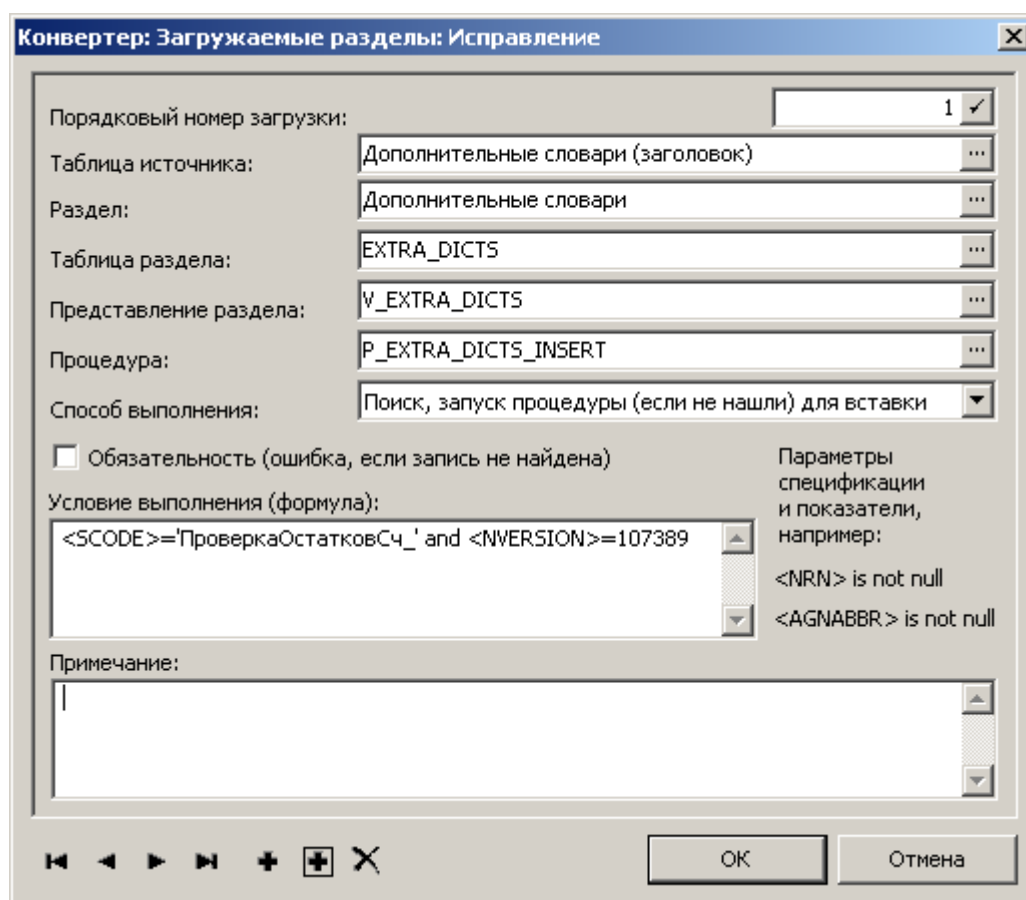


Рисунок 37

Те, кто работает на эталонной базе данных, версию раздела могут легко узнать с помощью удобной процедуры «get_info».

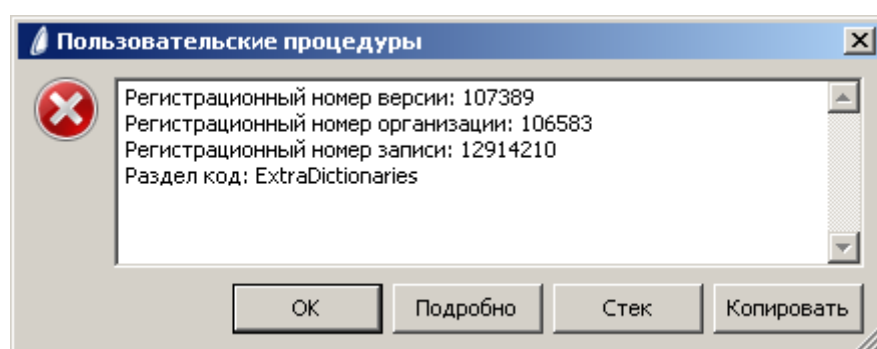


Рисунок 38

Заполнение спецификации параметров довольно тривиально, так как параметры процедуры добавления соответствуют полям источника. Например, «наименование» во вьюхе поле SNAME и имя параметра точно такое же.

Конвертер: Параметры: Исправление

Параметр

Порядковый номер: 4

Мнемокод параметра: SNAME

Наименование параметра: Наименование

Тип параметра: Входной Домен: TNAME

Поле представления: SNAME

Тип данных: Строка Длина: 160 ☒ Обязательное

Источник значения

Контекст: Нет

Показатель источника: SNAME

Параметр:

Значение по умолчанию:

Обработка значения

Функция обработки значения:

Примеры:
substr(<VALUE>,1,3)
upper(<VALUE>)
<VALUE>*20/100

☒ Удалить крайние пробелы ☐ Выравнивать пробелами слева (префикс, номер)

Уникальность

Входит в уникальный: ☐ ключ 1 ☒ ключ 2 ☐ ключ 3 ☐ ключ 4 ☐ ключ 5

☐ Сгенерить уникальное значение (для обхода физического ключа уникальности)

Примечание

ОК Отмена

Рисунок 39

Особенность алгоритма в том, что нам необходимо «запомнить» RN созданной записи словаря. Ведь ее необходимо будет найти, когда мы будем загружать значения словаря. Это делается очень просто, для выходного

параметра NRN мы должны указать, какое поле источника является внешним ключом.

Конвертер: Параметры: Исправление

Параметр

Порядковый номер: 9

Мнемокод параметра: NRN

Наименование параметра: NRN

Тип параметра: Входной/выходной Домен: TRN

Поле представления: NRN

Тип данных: Число Длина: 22 ☒ Обязательное

Источник значения

Контекст: Первичный ключ

Внешний ключ источника: NRN

Параметр:

Значение по умолчанию:

Обработка значения

Функция обработки значения:

Примеры:
substr(<VALUE>,1,3)
upper(<VALUE>)
<VALUE>*20/100

☒ Удалить крайние пробелы ☐ Выравнивать пробелами слева (префикс, номер)

Уникальность

Входит в уникальный: ☐ ключ 1 ☐ ключ 2 ☐ ключ 3 ☐ ключ 4 ☐ ключ 5

☐ Сгенерить уникальное значение (для обхода физического ключа уникальности)

Примечание

OK Отмена

Рисунок 40

В результате такого приема система в разделе «внешние ключи» будет сохранять соответствие созданных записей с уникальными внешними ключами, и

мы сможем использовать эту таблицу соответствий при загрузке значений словаря.

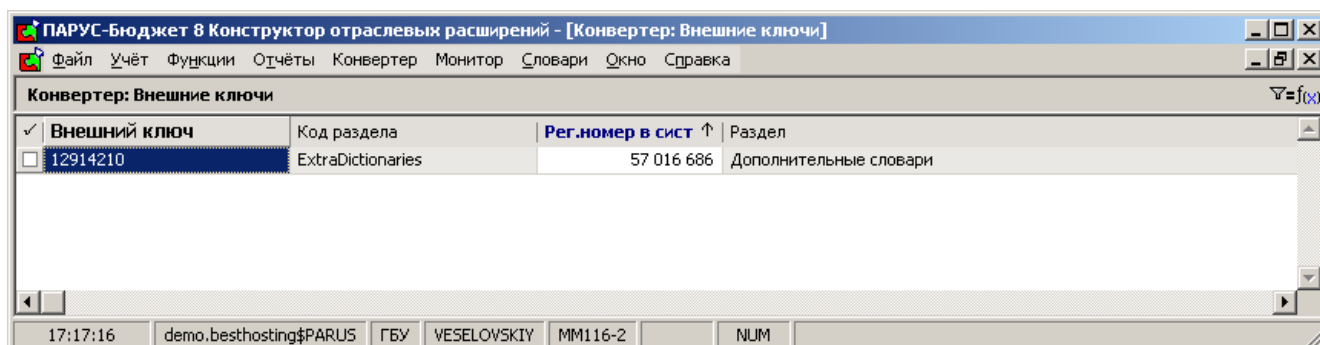


Рисунок 41

При добавлении записей в раздел «Дополнительные словари (значения)» в качестве RN заголовка, нужно будет подставить не поле источника, а соответствующий ему RN созданной записи. Для этого служит контекст «Регистрационный номер по внешнему ключу».

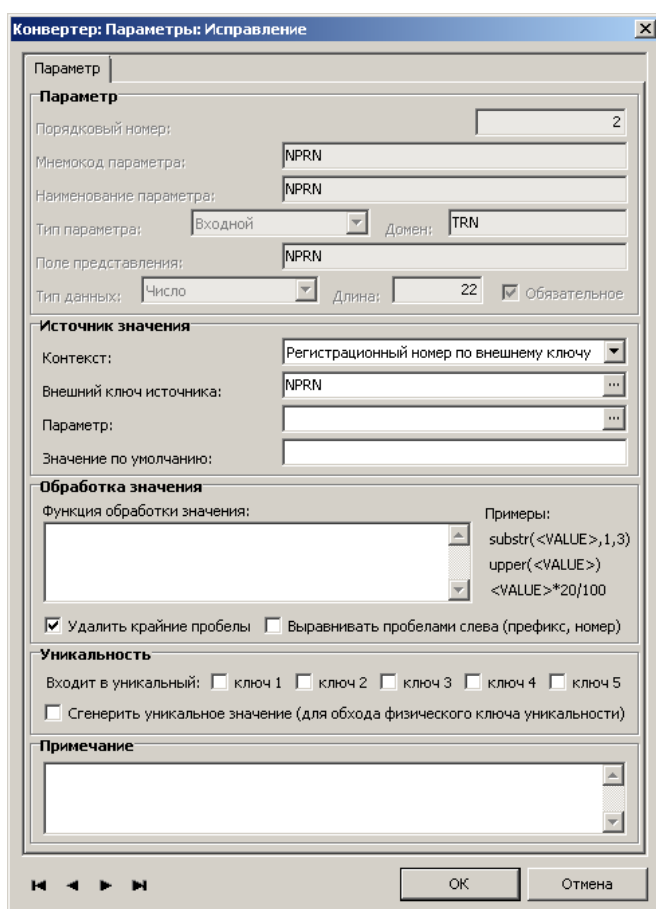


Рисунок 42

В этом случае в качестве параметра NPRN процедуре добавления будет передано не значение поля NPRN из таблицы источника, а созданный RN записи из раздела «внешние ключи», соответствующий полю NPRN источника.

Еще несколько слов о поле «условие выполнения» при добавлении значений в словарь. Понятно, что загружать значения словаря нам нужно не по всем записям, а только по тем записям, по которым мы загружали заголовки (сами словари). То есть записи, по которым есть записи в разделе «внешние ключи». Другими словами, «условие выполнения» – `<NPRN> is not null` (параметр NPRN существует) должно было бы решить задачу. Однако, в данном примере не так все просто. Используемые в формуле таги (элементы макроподстановки в треугольных скобках, которые при конвертации заменяются на значения соответствующих полей источника или параметров) заменяются значениями в следующем порядке:

- 1) поле источника с заданным именем;
- 2) имя параметра используемого метода.

То есть, приоритет за полями источника. В нашем примере по невероятной случайности имя параметра NPRN совпало с именем поля в таблице источника. В этом случае, условие `<NPRN> is not null` всегда вернет значение «истина», так как поле NPRN есть во всех строках источника. А это совсем не тот результат, который нам необходим. Поэтому, необходимо использовать чуть более сложное выражение:

`<P_EXTRA_DICTS_VALUES_INSERT.NPRN> is not null`

То есть, необходимо явно указать, что нас интересует NPRN именно в качестве параметра процедуры «P_EXTRA_DICTS_VALUES_INSERT».

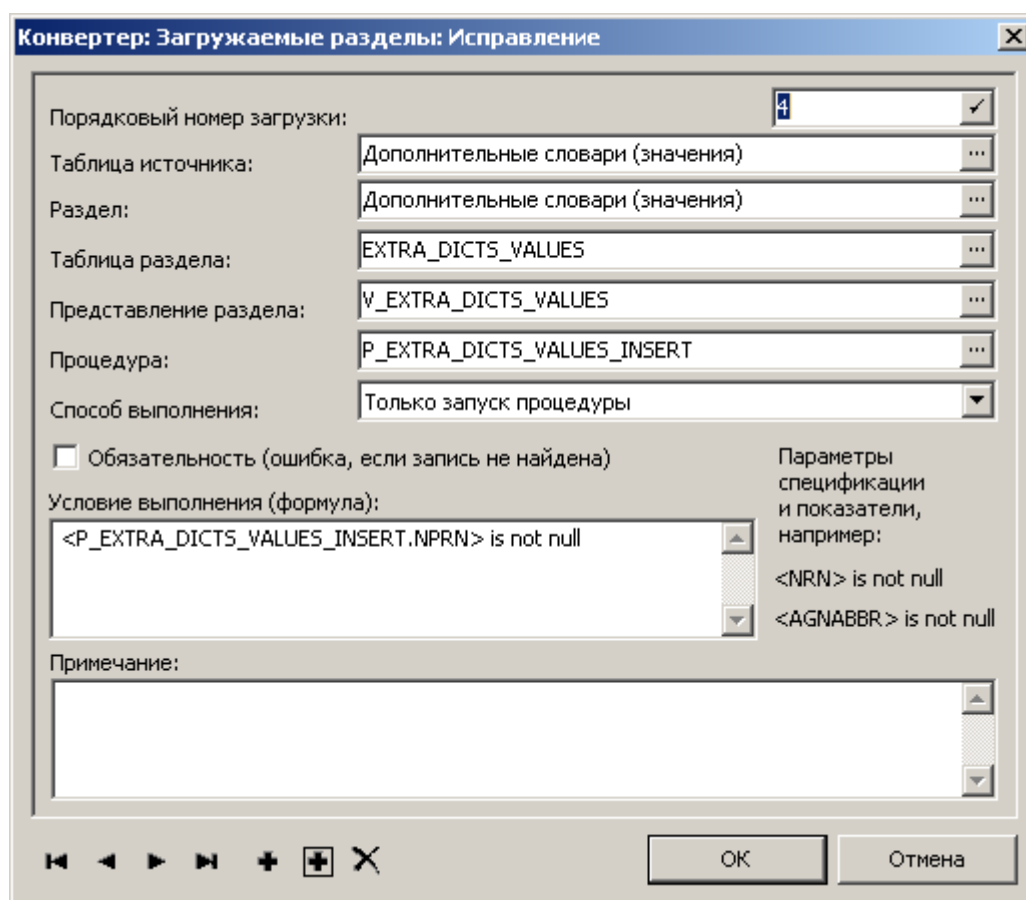


Рисунок 43

На этом настройку алгоритма загрузки можно считать законченной.

Отладка

Как обычно, добавляем «источник данных» и запускаем конвертацию.

Необходимо помнить две особенности конвертации данных их Oracle:

- 1) Возможен только один способ загрузки – из раздела «Источники данных», так как мы не выбираем никакого файла и использование пользовательского приложения «Конвертер: Источники данных. Загрузить данные из файла» невозможно и не имеет смысла;

2) Действие «Загрузить файл» в разделе «Источники данных» не используется по той же причине. Загружать ничего не требуется. Данные уже находятся в базе данных, алгоритм конвертации не использует «хранилище».

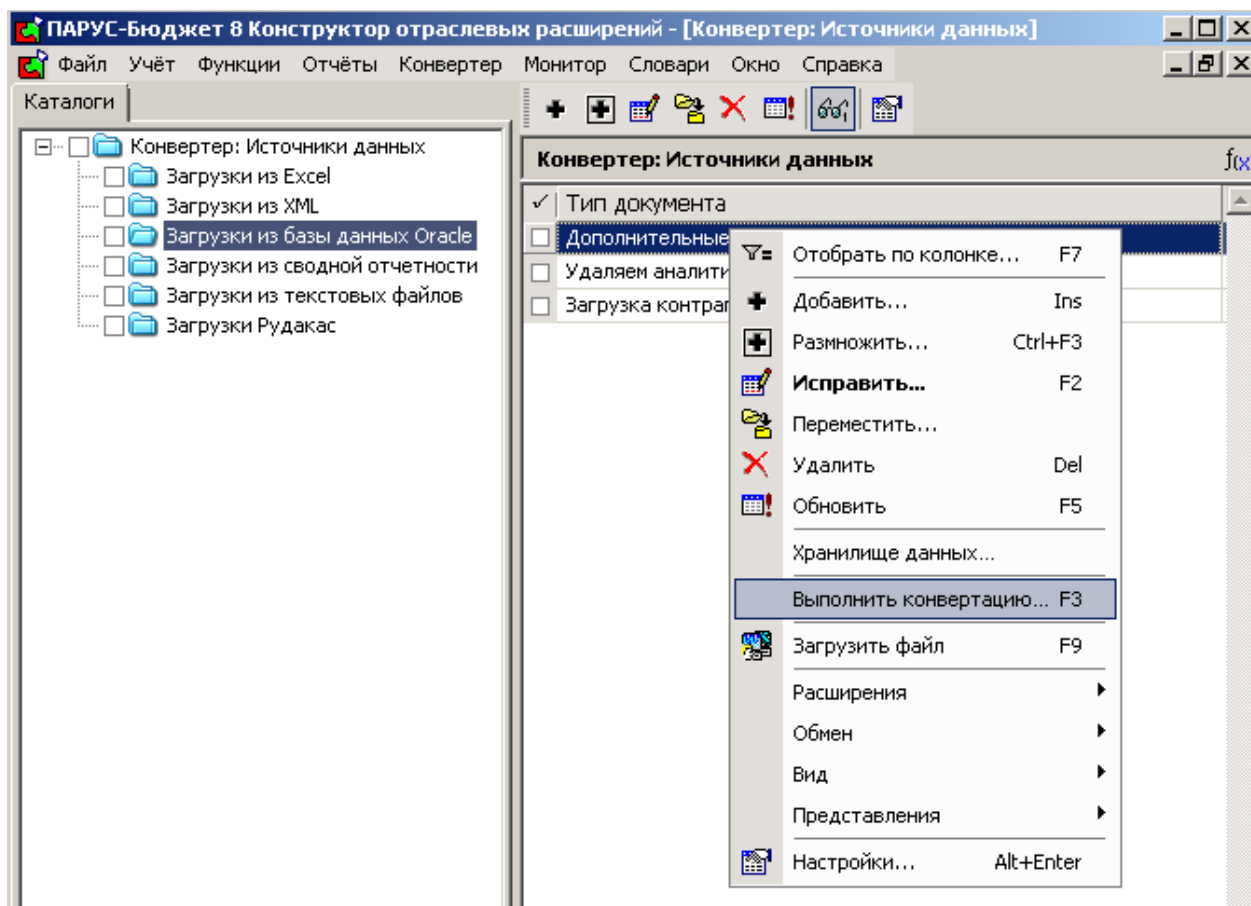


Рисунок 44

3.3.6 Форма сводной отчетности

Введение

Последний еще не описанный формат представления данных для конвертации – форма сводной отчетности. Эти методики имеют много общего с предыдущим разделом, ведь данные уже находятся в базе данных и этап загрузки данных в базу отсутствует. Однако, есть и некоторое количество существенных отличий, которые мы и рассмотрим ниже. Но, прежде чем углубиться в

технические детали, остановимся чуть подробнее на вопросе практического использования подобных методик.

Начнем с рассмотрения приблизительной схемы «стандартной» конвертации данных их внешней системы.

Конвертация из внешних систем «Как обычно»

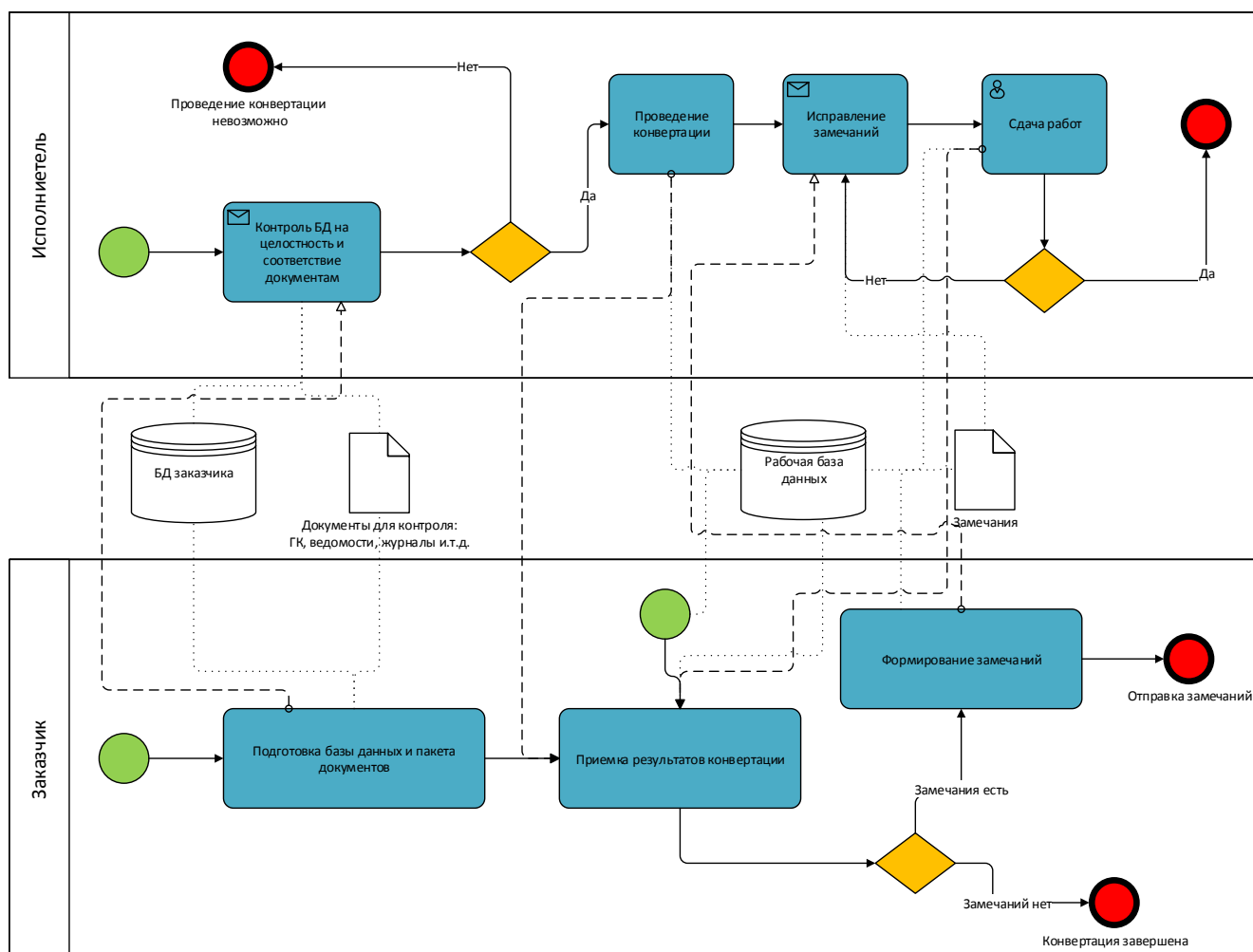


Рисунок 45

Нетрудно заметить, что в этом процессе много узких мест:

- сложно формализовать причины, по которым базу нельзя взять в работу; приходится указывать на конкретные ошибки и проблемы, после их исправления повторно проводить проверку базы;

- не нормализуется процесс исправления замечаний, количество замечаний и количество циклов в общем случае бесконечны, процесс имеет склонность к заикливанию;

- сложности с юридической стороной дела, почти невозможно однозначно определить виновного в затягивании работ.

В результате, конвертация данных из внешних систем становится едва ли не самым рискованным этапом контракта.

Конвертация из внешних систем «Используя своды»

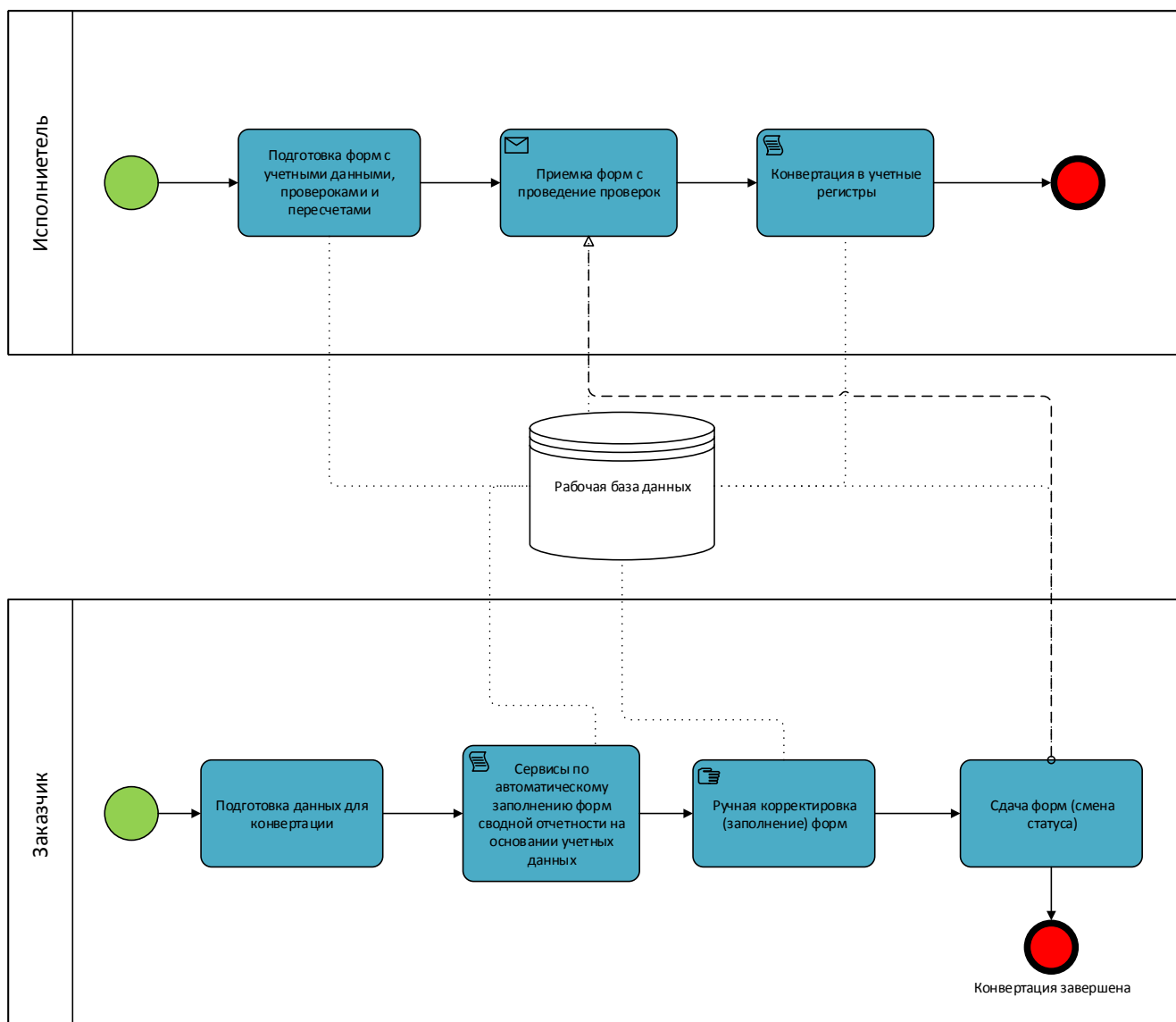


Рисунок 46

Рассмотрим теперь порядок конвертации с использованием модуля сводной отчетности. В этом случае процесс становится значительно проще и прозрачнее. Зацикливание невозможно, процесс строго последовательный и на каждом этапе нет никаких сомнений в том, кто ответственен за текущую задачу. А точнее сказать, за все задачи отвечает заказчик, исполнитель лишь обеспечивает работоспособность системы и правильную настройку форм. Весь процесс конвертации можно представить в виде простого последовательного перечня мероприятий:

№	Мероприятие	Ответственный
1.0	<i>Подготовительные мероприятия</i>	
1.1	Подготовка методики конвертации, с отражением загружаемых разделов и контрольных соотношений	Исполнитель
1.2	Согласование методики конвертации	Заказчик
1.3	Настройки сервисов, форм, проверок и алгоритмов загрузки	Исполнитель
1.4	Проведение пробной конвертации, защита макета у заказчика	Исполнитель
1.5	Подготовка данных для конвертации, закрытие периода, сдача отчетности	Заказчик
2.0	<i>Проведение конвертации</i>	
2.1	Заполнение с использованием сервисом и (или) в ручном режиме форм в «сводах», проверки и, при необходимости, корректировка.	Заказчик
2.2	«Сдача» форм (смена статуса). Возможна только при отсутствии критичных ошибок	Заказчик
2.3	Проведение конвертации	Заказчик

Обратите внимание, что исполнитель выполняет работы на подготовительном этапе, а в период проведения конвертации, когда сроки критичны, все работы выполняются заказчиком, ответственность за сроки лежит на нем и легко контролируется, благодаря статусной модели модуля «Сводь». На исполнителе лежит только функция обеспечения работоспособности системы.

Теперь, когда область использования рассматриваемой возможности стала более понятной, рассмотрим технические особенности настройки конвертации.

Добавление Формата обмена данными

Предположим в «Сводах» создана форма такого содержания:

Код	Наименование	Полное наименов	Дата начала	Код родителя	Направление расх
0100000000	Государственная прог	Государственная прог	01.01.2017		00000
0110000000	Подпрограмма "Проф	Подпрограмма "Проф	01.01.2017	0100000000	00000
0110100000	Основное мероприяти	Основное мероприят	01.01.2017	0110000000	00000
0110200000	Основное мероприяти	Основное мероприят	01.01.2017	0110000000	00000
0110300000	Основное мероприяти	Основное мероприят	01.01.2017	0110000000	00000
0110351790	Иные межбюджетные	Иные межбюджетны	01.01.2017	0110300000	51790
0110400000	Основное мероприяти	Основное мероприят	01.01.2017	0110000000	00000
0110500000	Основное мероприяти	Основное мероприят	01.01.2017	0110000000	00000

Нетрудно заметить, что форма полностью идентично файлу TARGET_ITEM.xlsx для которого мы уже настраивали алгоритм загрузки ранее (пункт инструкции 3.3.1 Книга Excel). Это сделано намеренно, чтобы не было нужды рассматривать особенности алгоритмов, независящих собственно от формата представления данных.

Как обычно, воспользуемся действием «создать формат».

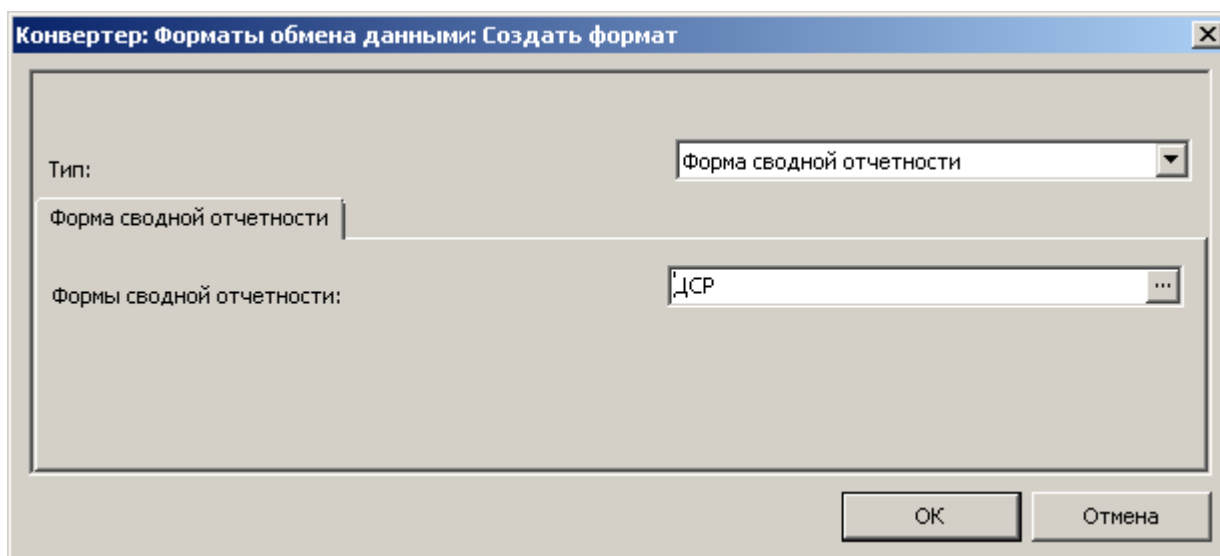


Рисунок 47

Собственно, на этом создание формата практически закончено. Целесообразно еще дать понятные имена полям, это упростит в дальнейшем настройку алгоритма загрузки. Разумеется, идентификаторы мы оставляем в том виде, как они создались, в соответствии с показателями формы.

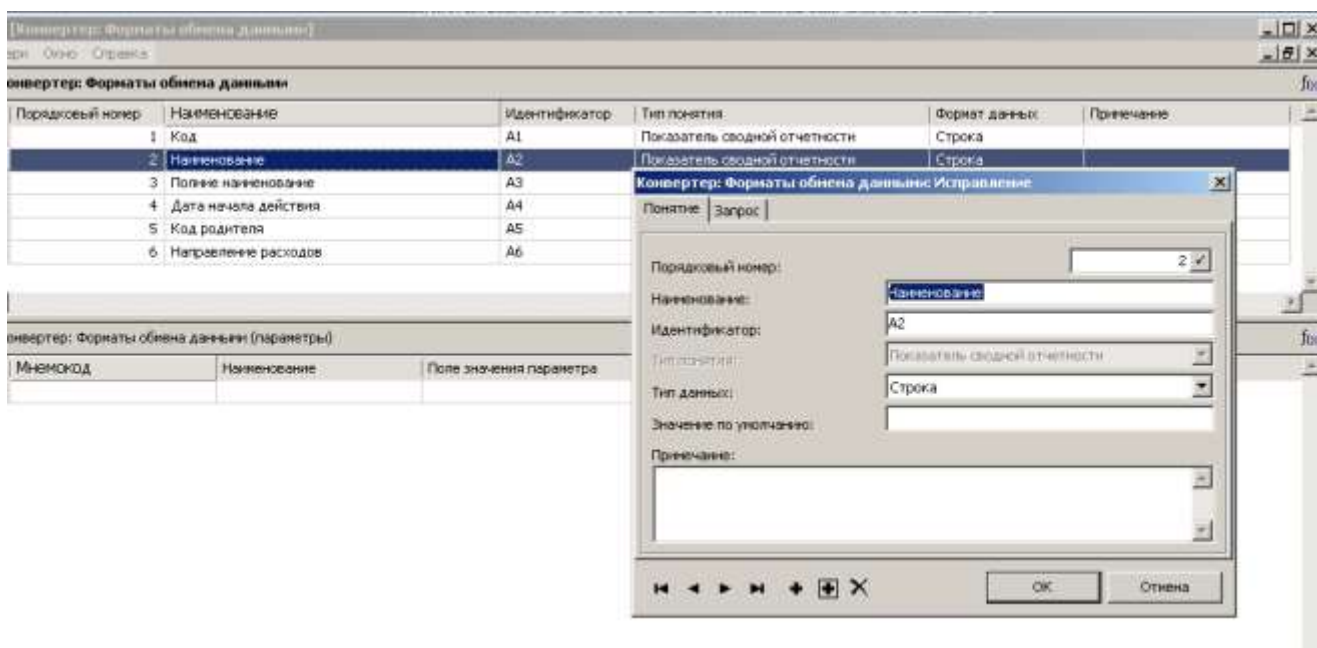


Рисунок 48

Создание Алгоритма загрузки

Алгоритм загрузки, который необходимо настроить для загрузки целевых статей, практически полностью идентичен алгоритму, который мы создавали для файла excel, только показатели выбираем из другого формата. Не будем останавливаться на этом, так как тут нет никаких новых особенностей.

Запуск конвертации

В начале данной инструкции, мы уже рассматривали особенности запуска конвертации для «сводного» отчета. На рисунке 2 показаны варианты запуска конвертации пользователями. Информацию из "Сводной отчетности" можно загружать только специализированной процедурой, так как необходимо знать, какой именно первичный/сводный отчет используется в качестве источника данных.

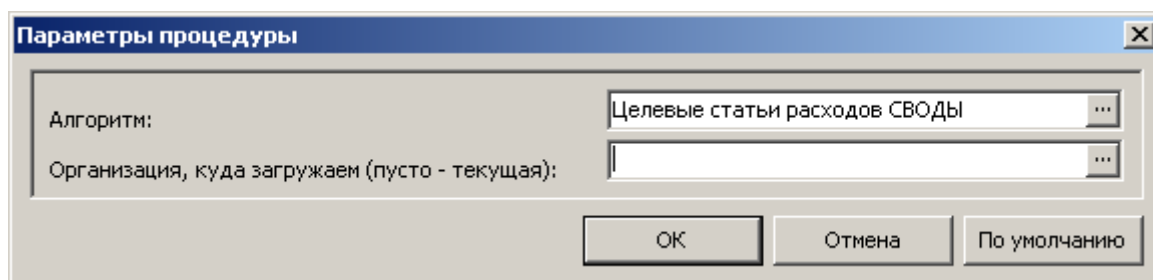


Рисунок 49

Выбираем настроенный алгоритм и, если необходимо загружать данные не в ту компанию, в которой у нас находятся первичный/сводный отчет, «организацию». На этом работа с конвертором завершена.

4 Тиражирование настроек

Для переноса настроек из одной базы данных в другую, удобнее всего воспользоваться специальными пользовательскими процедурами:

- Импорт раздела из XML-файла;
- Экспорт раздела в XML-файл.

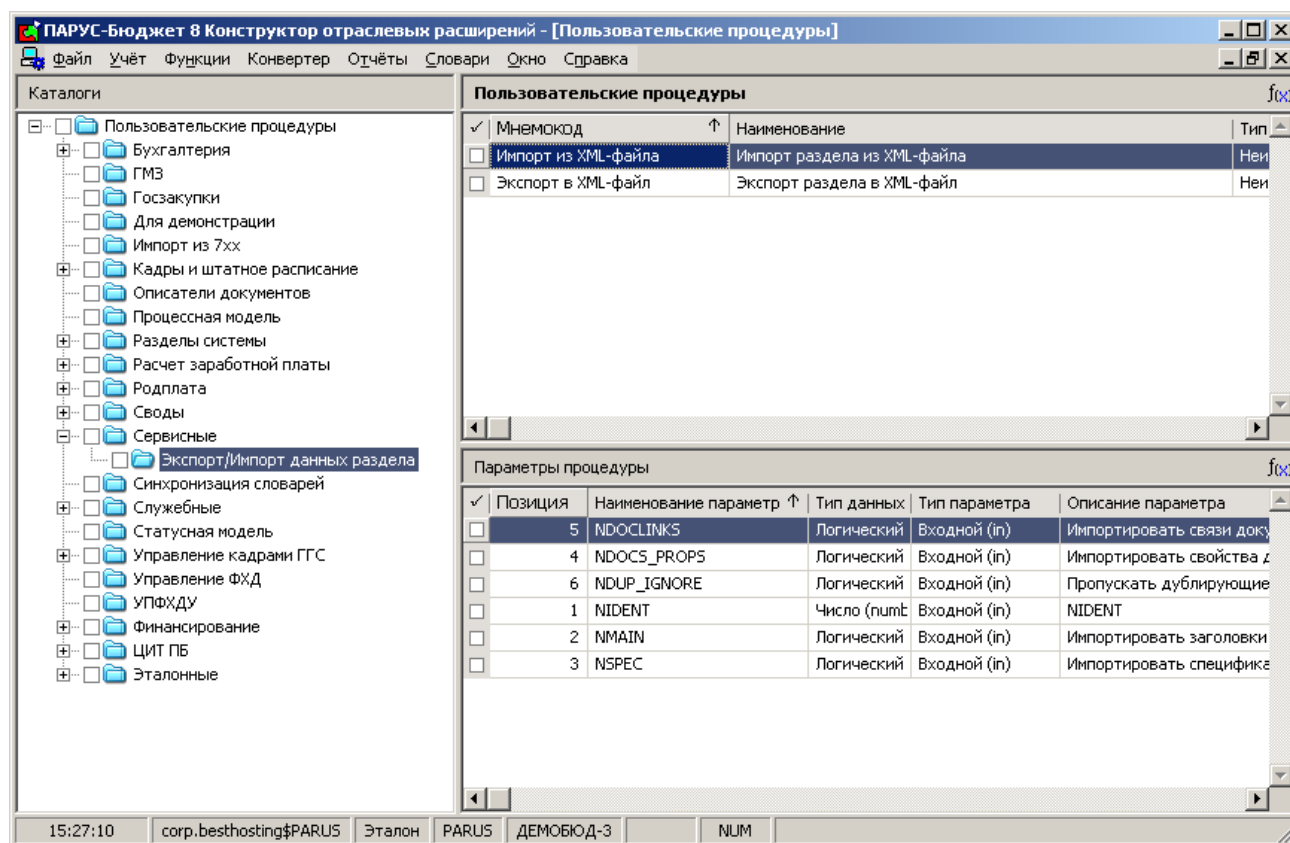


Рисунок 50

Данные процедуры предназначены для экспорта и импорта данных через XML в любом разделе ПАРУС 8, который имеет полное и типовое описание раздела в КОР. В данном случае мы будем использовать процедуру для передачи из базы в базу Описания форматов и Алгоритмов загрузки.

Сначала опишем процедуру загрузки, так как ее необходимо использовать, чтобы получить описание примеров в своей базе данных.

Действие производится стандартным способом, через «обмен»

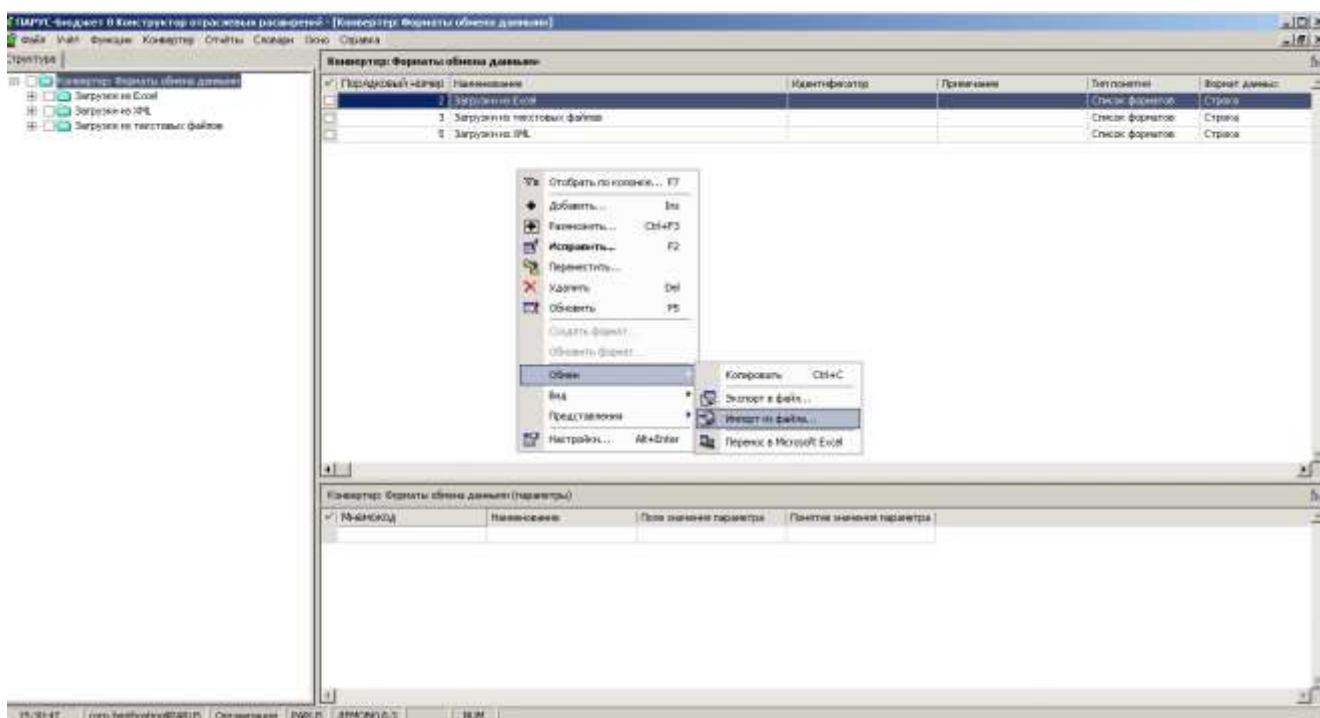


Рисунок 51

Выбираем процедуру и файл dump_udosconcept.xml, содержащий примеры описания форматов.

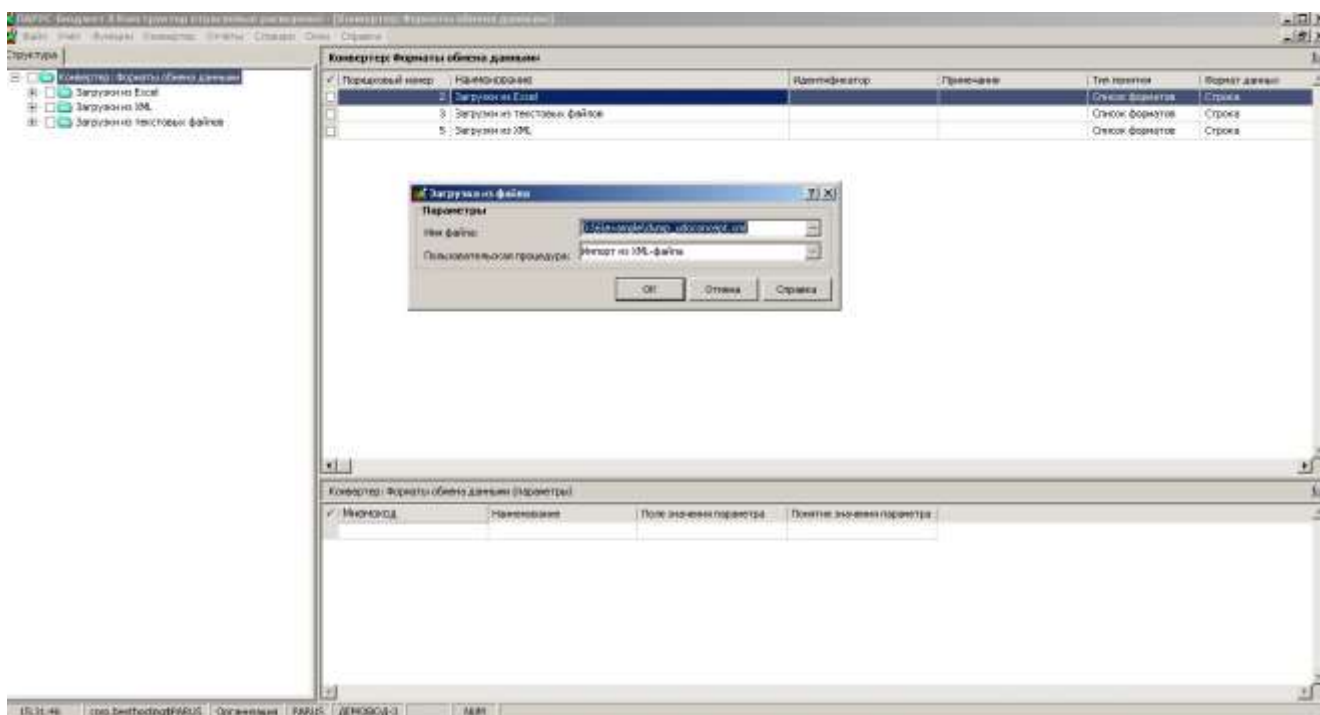


Рисунок 52

Аналогичным образом загружаем алгоритмы, содержащиеся в файле dump_udoconv.xml

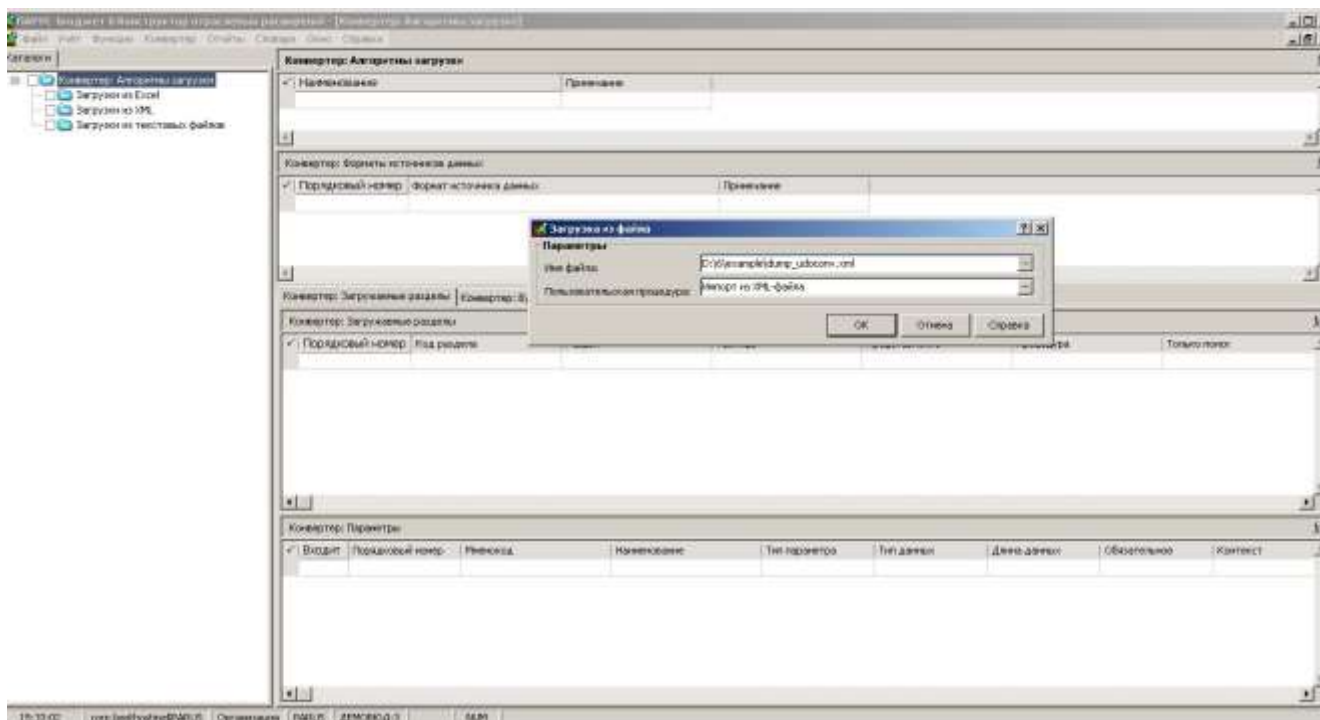


Рисунок 53

В случае если вам необходимо выполнить обратную операцию, то есть выгрузить свои настройки для использования в другой базе данных, используйте процедуру «Экспорт раздела в XML-файл».

При этом необходимо выгрузить в XML как сами алгоритмы

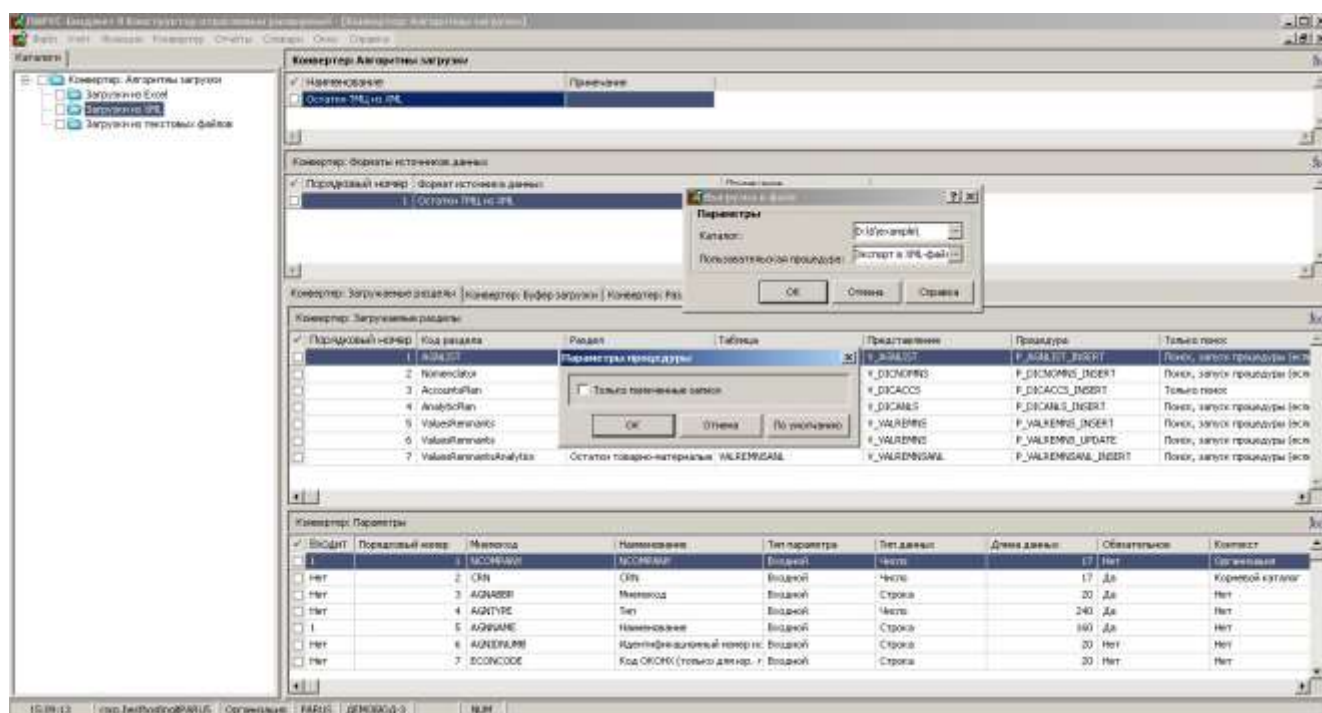


Рисунок 54

так и форматы, используемые в этих алгоритмах.

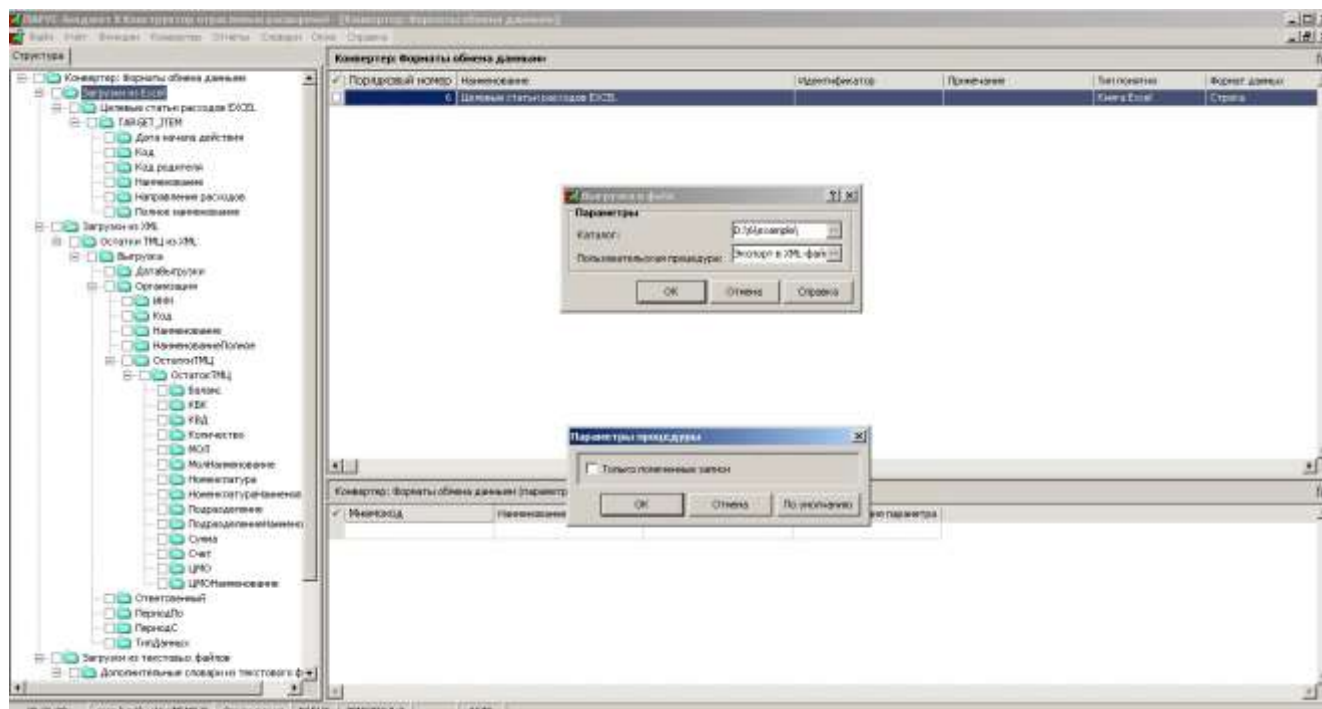


Рисунок 55

5 Заключение

Все рабочие базы данных, обслуживаемые группой компаний ЦНТ «Плюс»¹ уже содержат все объекты «Конвертора». Таким образом, для того, чтобы можно было начать его использовать, достаточно добавить соответствующие пункты меню. Для этого в модуле «Администратор» или КОР нужно зайти в список «приложений» и воспользоваться пунктом «Редактор меню». Про работу с редактором меню можно прочитать в системе помощи.

Для прочих организаций необходимо воспользоваться специально подготовленным пакетом и установить его на рабочую базу данных, используя прилагаемую к нему инструкцию. Кроме того, необходимо приобрести специальную лицензию.

¹ Кроме тех, которые по какой-либо причине не переведены на «эталон».