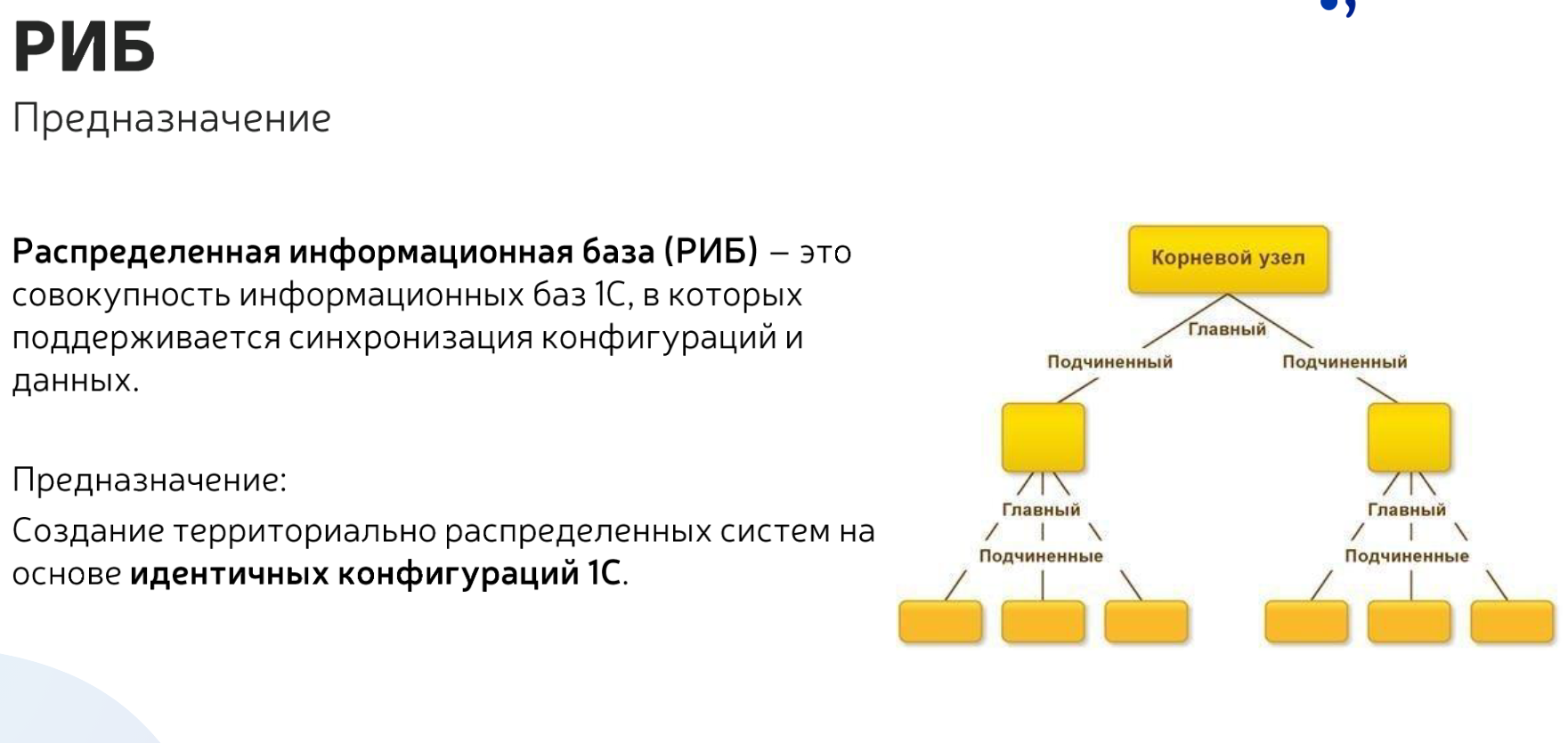
**Распределенная информационная база.**

Теория:

Распределенная информационная база(РИБ) – это совокупность информационных баз 1С, в которых поддерживается синхронизация конфигураций данных.

Предназначение – создание территориально распределенных систем на основе идентичных конфигураций.



Особенности РИБ:

1. Жесткая иерархия узлов:
   1. Один главный(корневой) узел
   2. Для каждой пары узлов определено отношение «главный – подчиненный».
2. Конфигурацию можно изменять только в корневом узле

2.1 Изменения конфигурации передаются в подчиненные узлы в сообщении обмена данными(включаются в состав сообщения)

*Разрешение коллизий.*

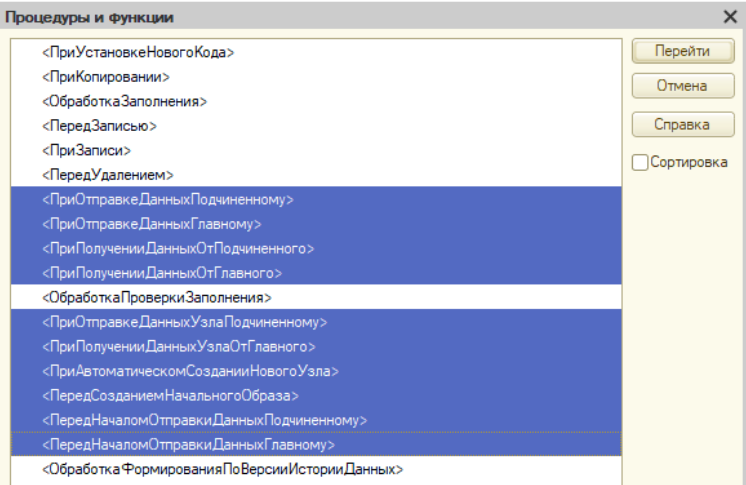
*Коллизия* в данном случае, это когда один объект меняется сразу в двух узлах.

Изменения в главном узле имеют более высокий приоритет относительно изменений в подчиненном узле.

То есть если в справочнике Номенклатура поменять описание какого-то товара и в главной и в подчиненной базе, то при загрузке данных из подчиненной базы ничего не изменится. А вот если мы из главной базы будем загружать данные, то в подчиненной все изменения применятся.

Сценарии обмена данными в РИБ.

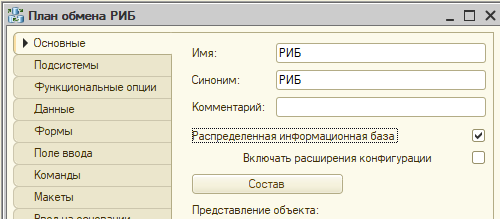
1. Поведение по умолчанию:
   1. Каждое изменение стремиться распространиться по всем узлам РИБ
   2. Разрешение коллизий производится на основании отношения «главный – подчиненный»
2. Распределение данных по подчиненным узлам
   1. Все данные хранятся в главном узле
   2. В подчиненные узлы передаются не все данные, а ограниченный набор
   3. Разрешение коллизий производится на основании отношения «главный – подчиненный»
3. Нестандартное разрешение коллизий
   1. Каждое изменение стремиться распространиться по всем узлам РИБ
   2. Разрешение коллизий производится на основании отношения «главный – подчиненный», но более высокий приоритет имеет подчиненный узел.



Задача: Организовать передачу данных в РИБ между Центральной базой и Магазином №1. Организовать передачу и в ручном режиме и при помощи регламентного задания.

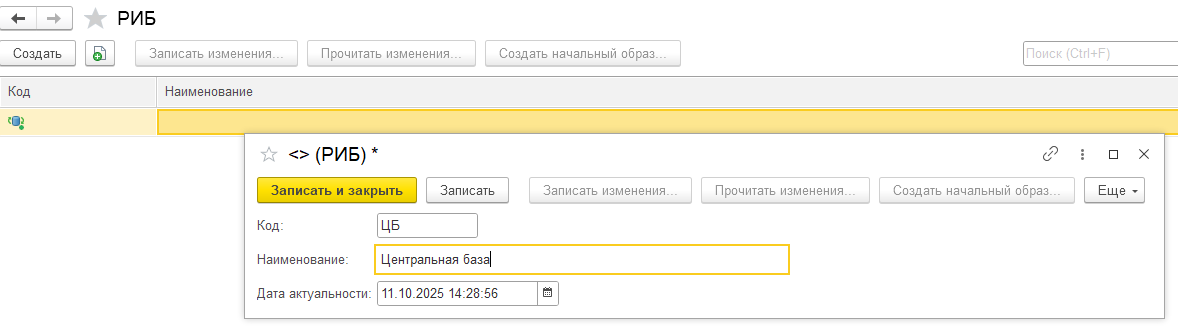
Ход работы:

Заходим в информационную базу и создаем «План обмена». Называем его РИБ и устанавливаем флаг «распределенная информационная база».

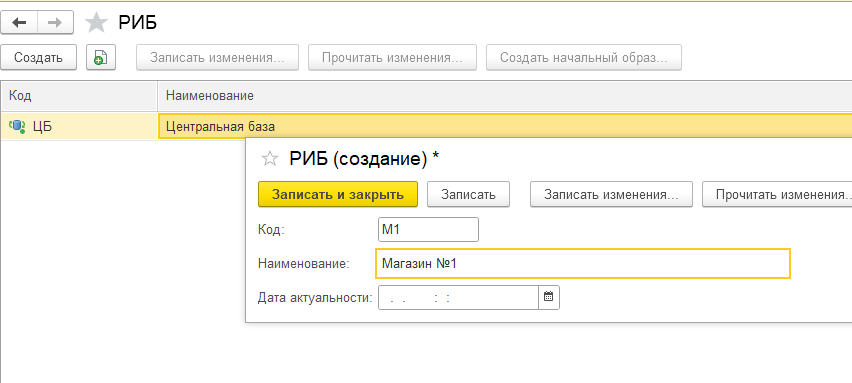


Обновляем изменения и переходим в режим 1С:Предприятие.

Заходим в РИБ и создаем узлы.

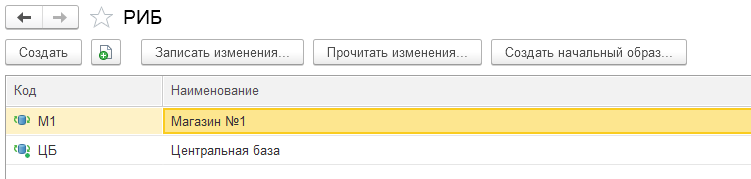


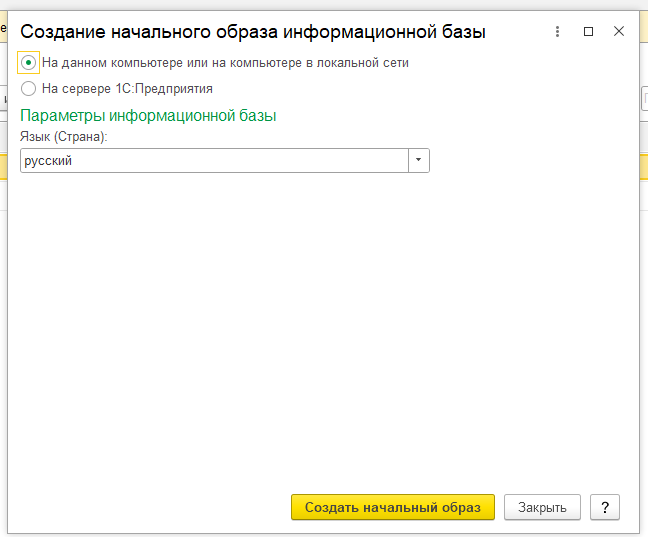
Первый узел – это наш корневой элемент, то есть центр нашей РИБ. От него уже пойдут все остальные элементы. В нашем случае ещё один элемент М1 Магазин1.

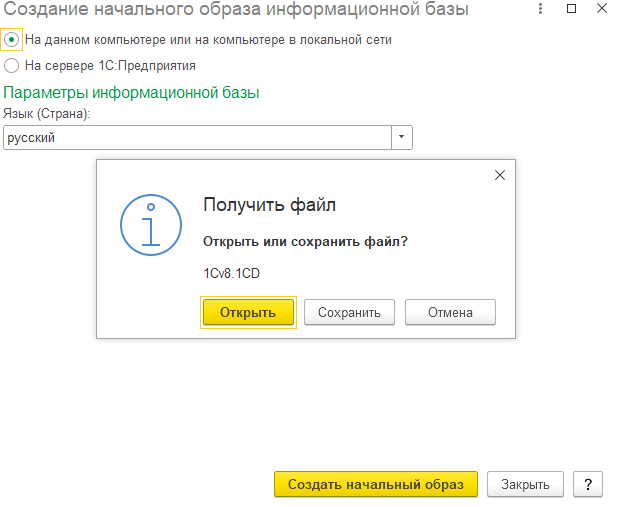


Далее нужно создать начальный образ нашей мазы для Магазина №1.

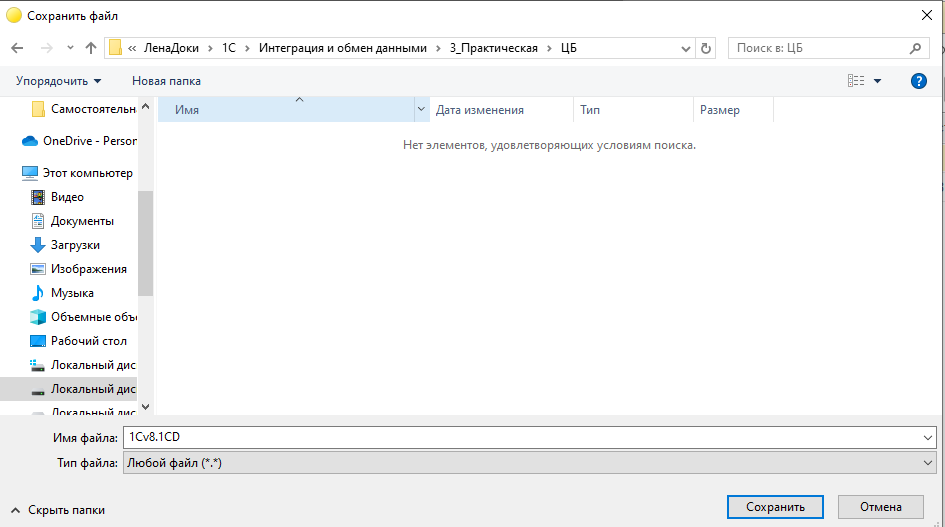
Нажимаем на кнопку Создать начальный образ при выделенном элементе Магазин №1.





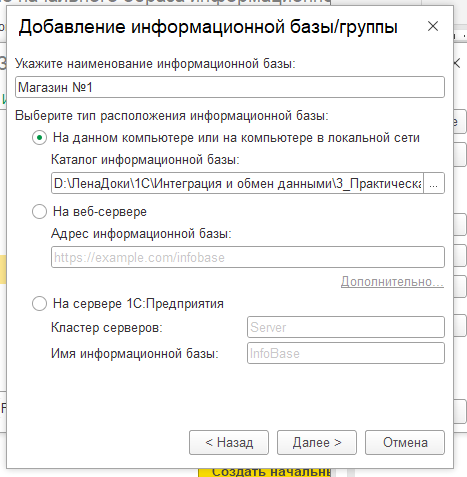


Нажимаем на кнопку Сохранить. И Выбираем путь сохранения.

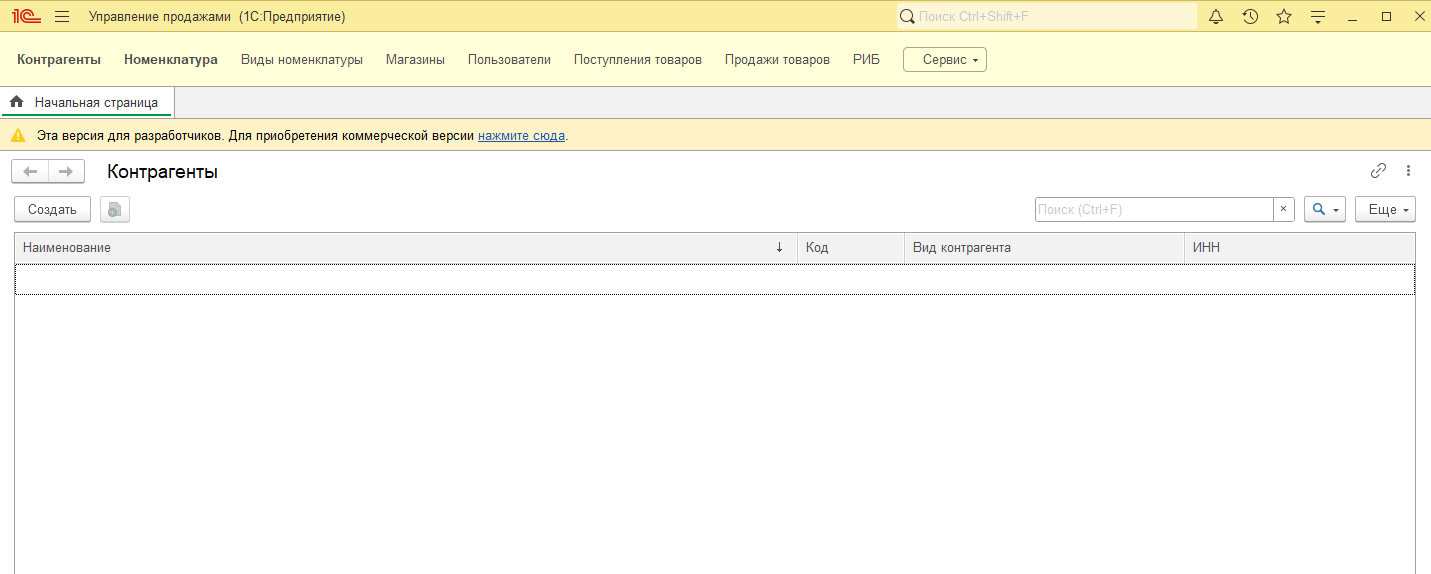


Теперь чтобы работать с этой базой нужно добавить её в список.

Запускаем 1С, выбираем **добавление** существующей и указываем каталог, где шагом выше был создан Магазин №1.

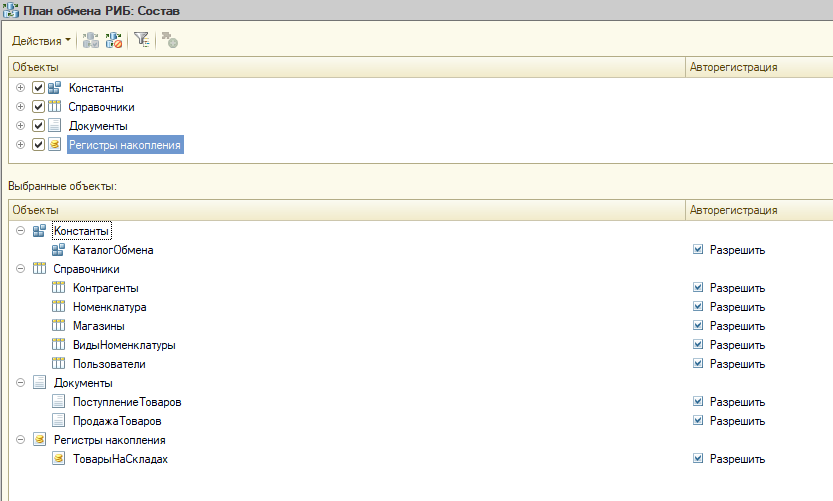


Заходим в режим Предприятие для данной базы и видим, что база создалась, но она пустая.

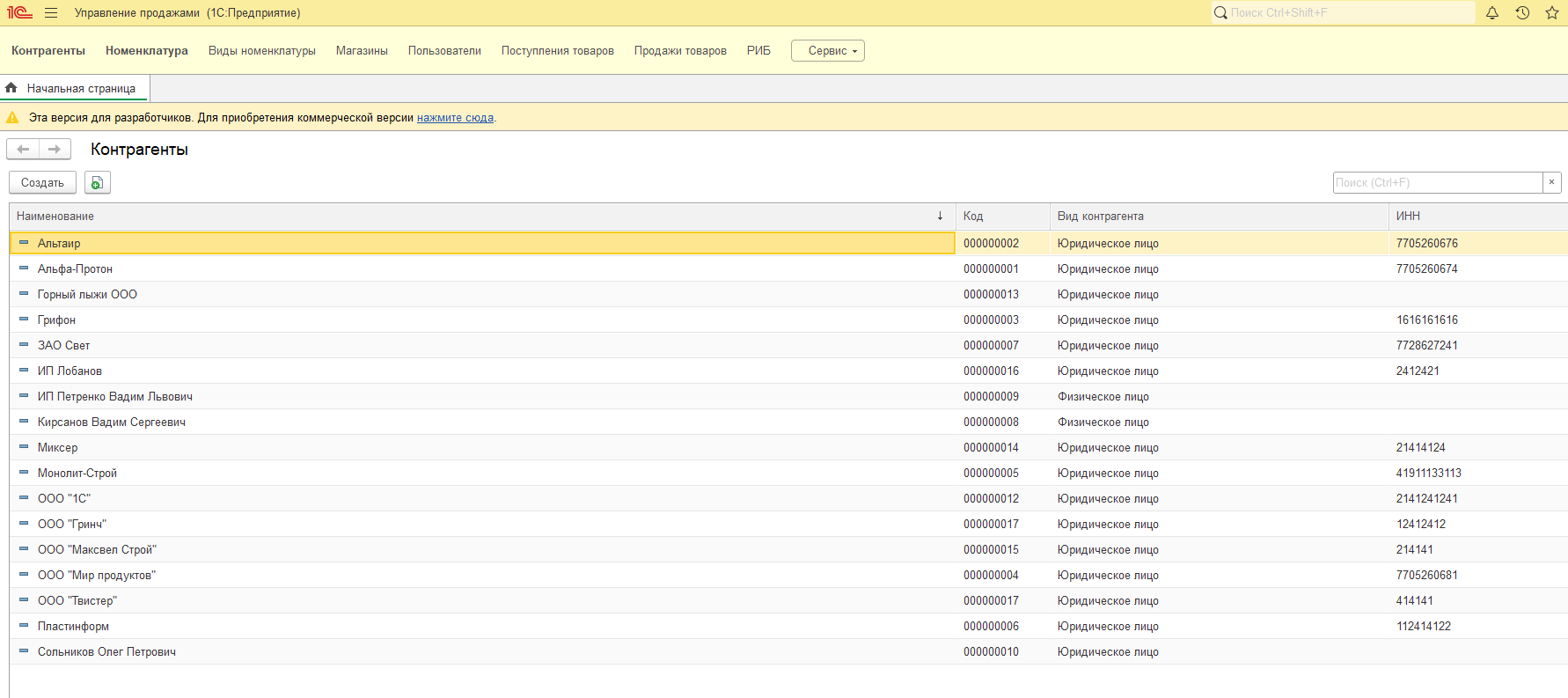


Чтобы это исправить, нужно вернуться в конфигуратор базы Центрального офиса(там где мы создавали план обмена) и отметить состав нашего плана обмена.

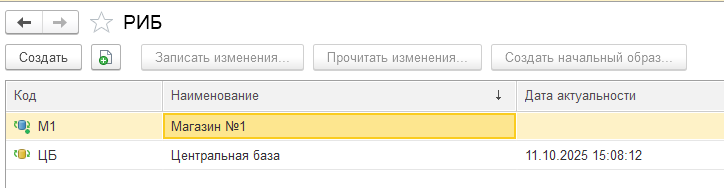
Добавляем все объекты и разрешаем авторегистрацию.



После чего переходим в пользовательский режим и создаем повторно начальный образ для нашей базы Магазин №1. Кстати, возможность создавать образы у нас появилась только благодаря тому, что при создании плана обмена мы установили флажок Распределенная информационная база. Сохраняем базу в ту же папку, в которую мы уже отправляли наш образ. После чего, открываем данную конфигурацию в пользовательском режиме и видим, что база содержит данные.



Откроем РИБ и посмотрим, что в этой базе он выглядит иначе.



ЦБ – отображается как корневой узел

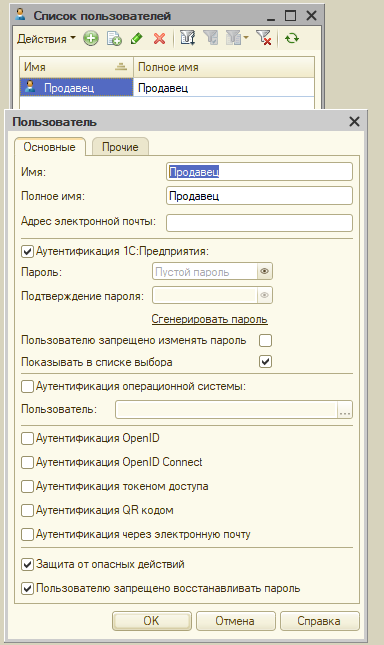
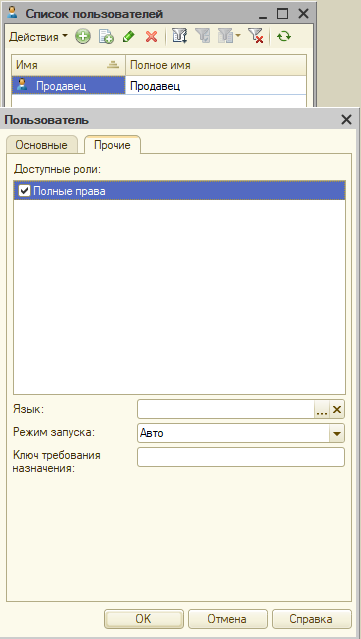
М1 – как предопределенный

И при желании сюда можно добавить ещё какие-то подчиненные узлы.

То есть, ***начальный образ — это копия базы корневого узла на момент создания этого начального образа***.

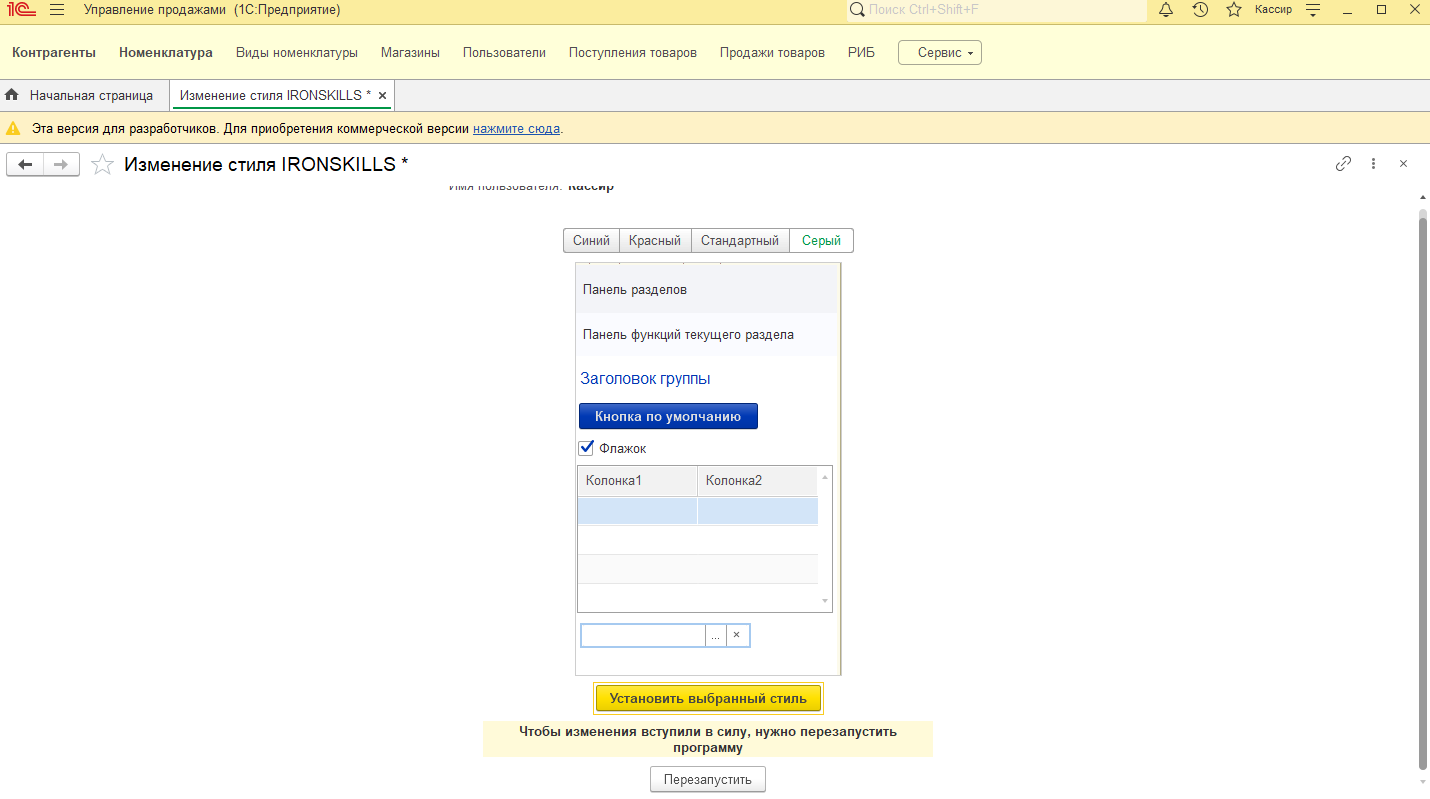
Работать в базах, которые выглядят абсолютно одинаково – очень сложно.

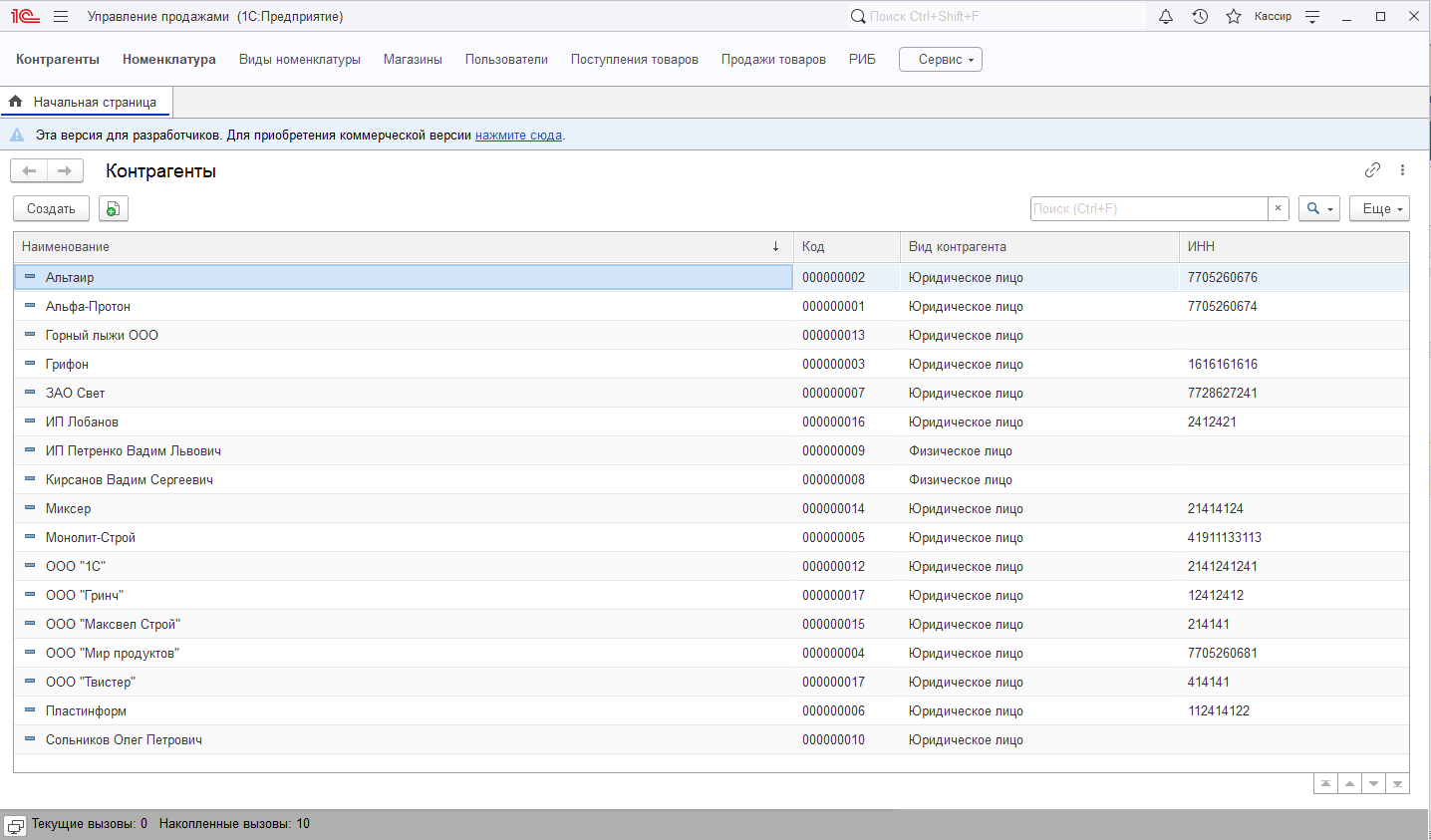
Теперь для того, чтобы не запутаться окончательно, где и какая база, поменяем стиль в базе Магазин №1. Прежде, чем это делать, откроем базу Магазин №1 в Конфигураторе и добавим нового Пользователя – Кассир.

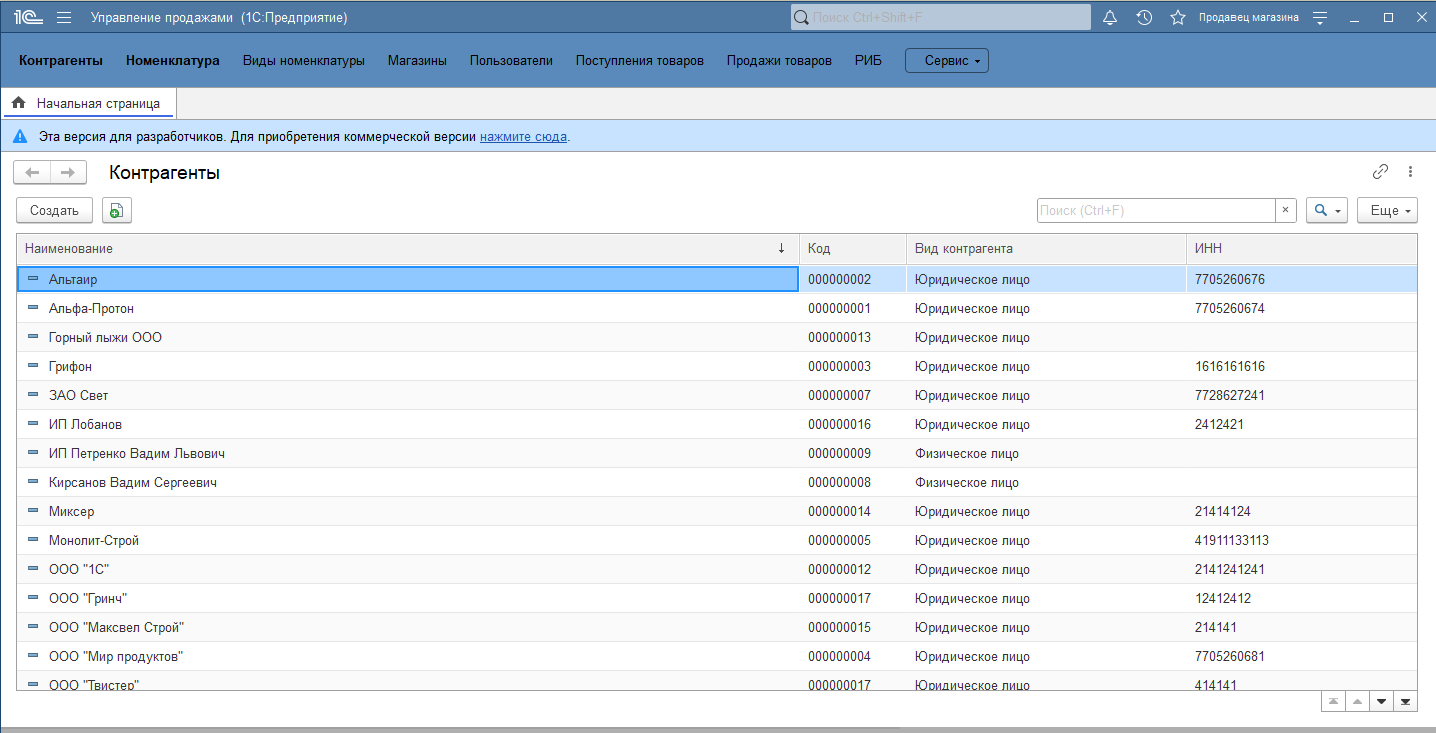
Переходим в пользовательский режим и открываем через Сервис – обработку Изменение стиля и выбираем Серый.

Появляется сообщение о том, что нужно перезапустить программу – перезапускаем.

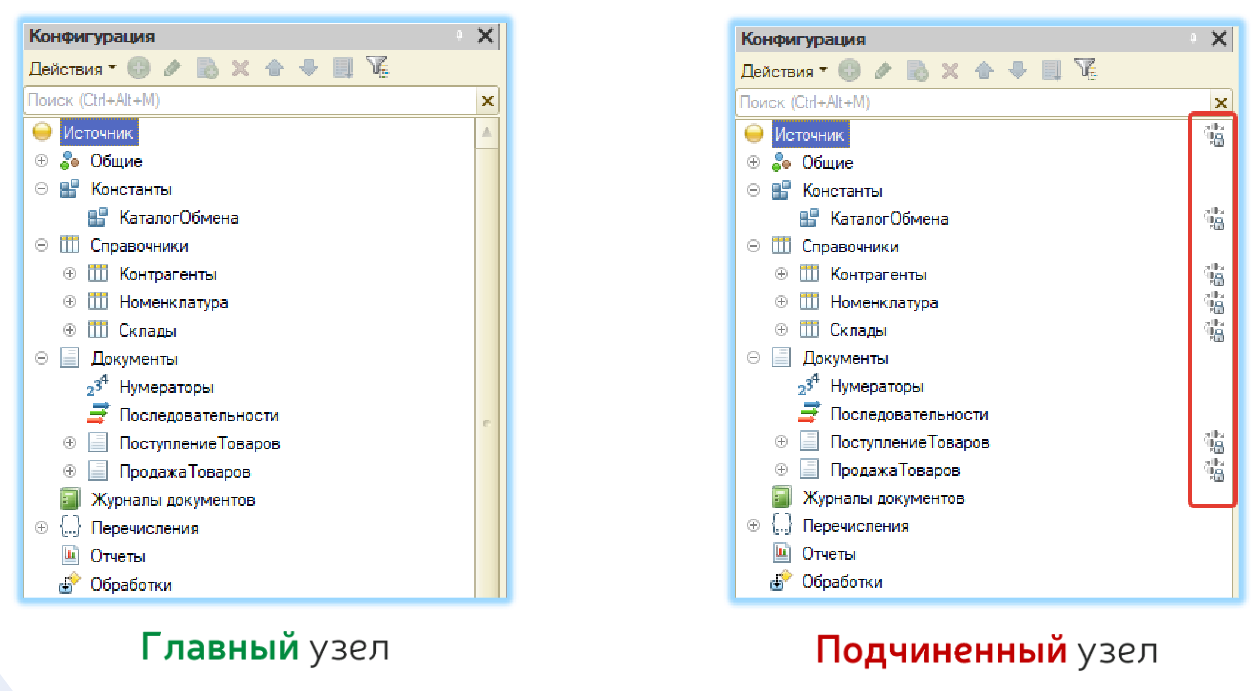




Аналогично добавляем базу для Магазина №2. Добавляем пользователя Продавец магазина и выбираем стиль Синий.

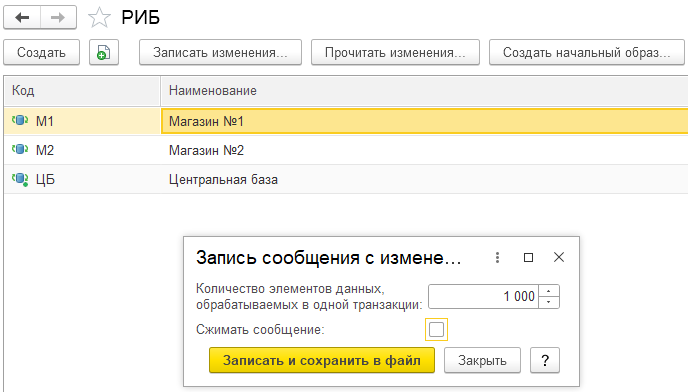


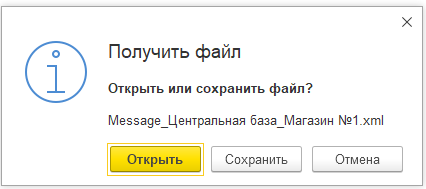
Редактирование конфигурации доступно только в базе корневого узла.



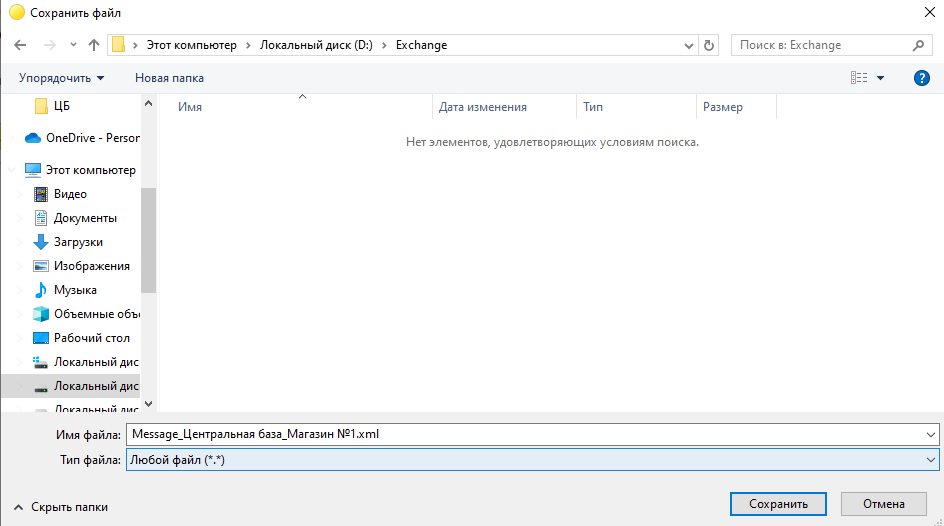
При использовании плана обмена для распределенной базы есть возможность автоматически осуществлять передачу данных.

Допустим, находясь в центральной базе заходим в РИБ и выбираем для какой базы мы хотим сформировать сообщение обмена. Выбираем Магазин №1 и нажимаем кнопку Записать изменения. В появившемся окне снимаем флажок в Сжимать сообщение(он нужен для того чтобы формировать архив – нам это не нужно).

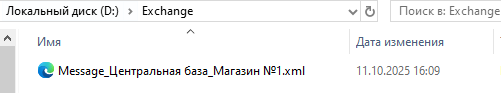


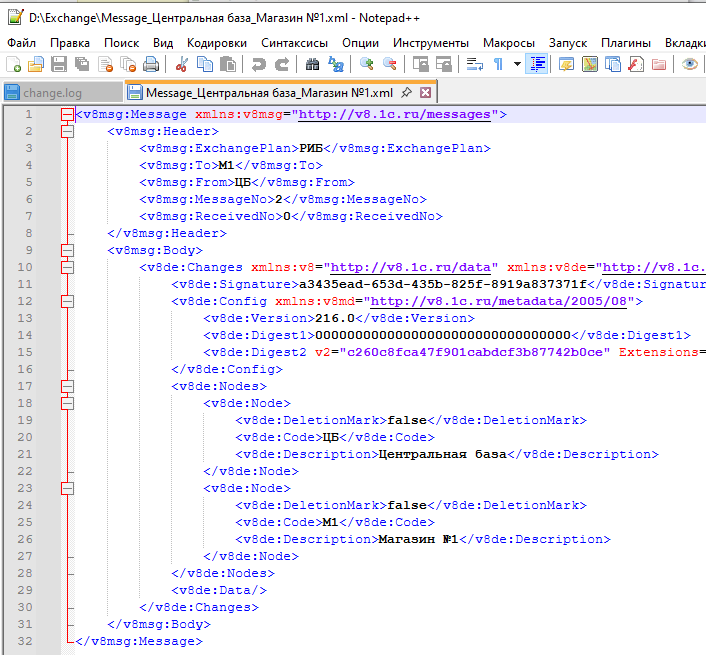


Дальше мы указываем в какой каталог нужно сохранить.

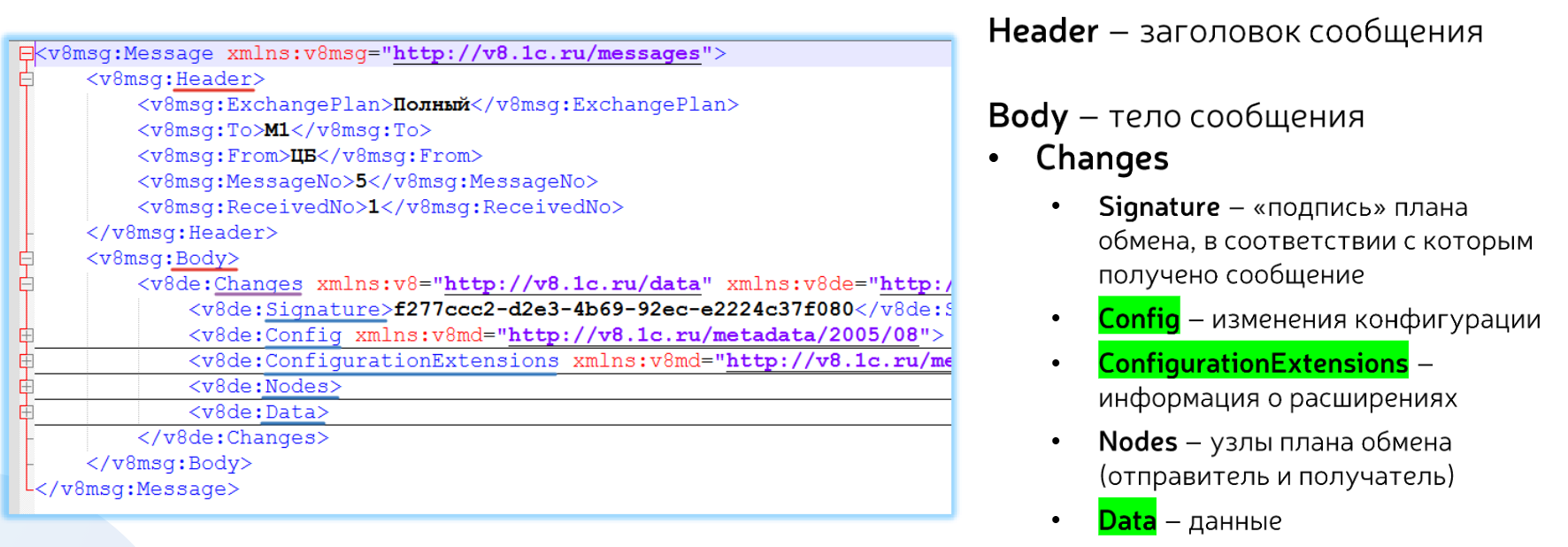


Если зайдём в каталог, то увидим сообщение обмена.



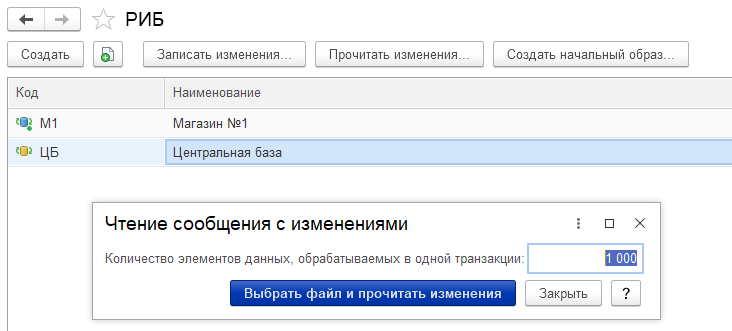


Как устроено сообщение обмена?

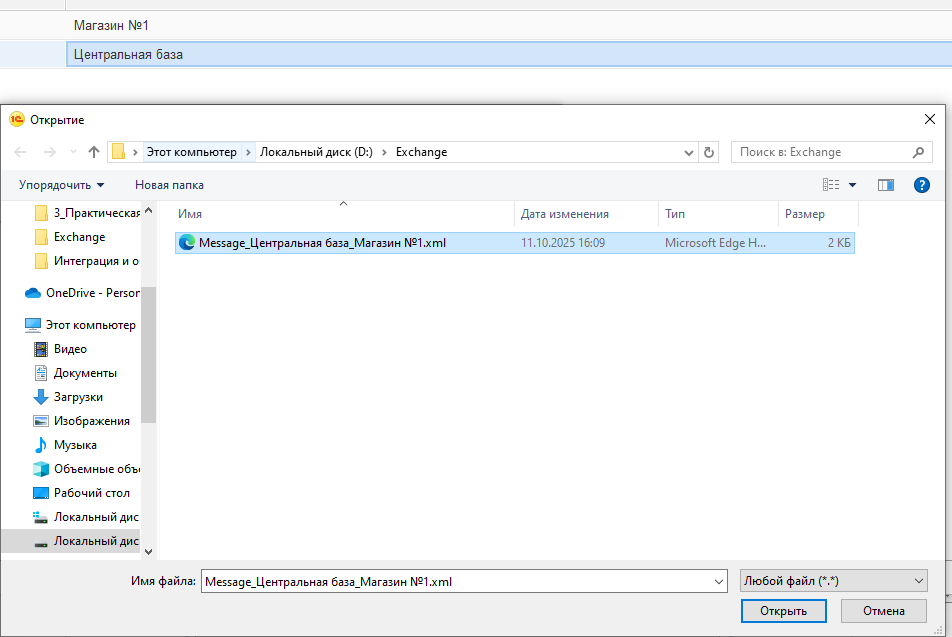


Чтобы загрузить эти данные идем в базу Магазин №1 и открываем план обмена РИБ.

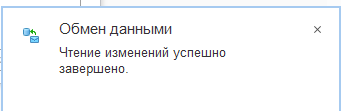
Выбираем узел из которого мы читаем сообщения и нажимаем прочитать изменения.



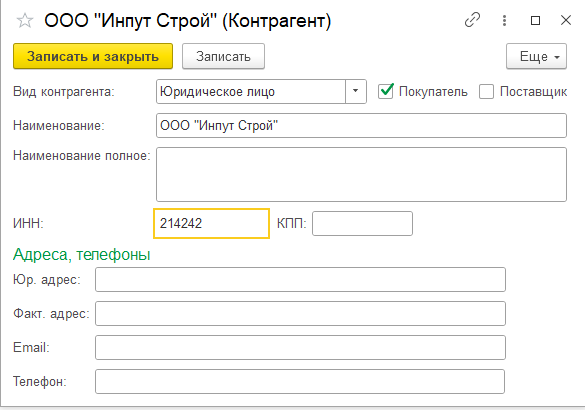
Выбираем файл и нажимаем Открыть.



Получаем сообщение, что данные прочитаны.

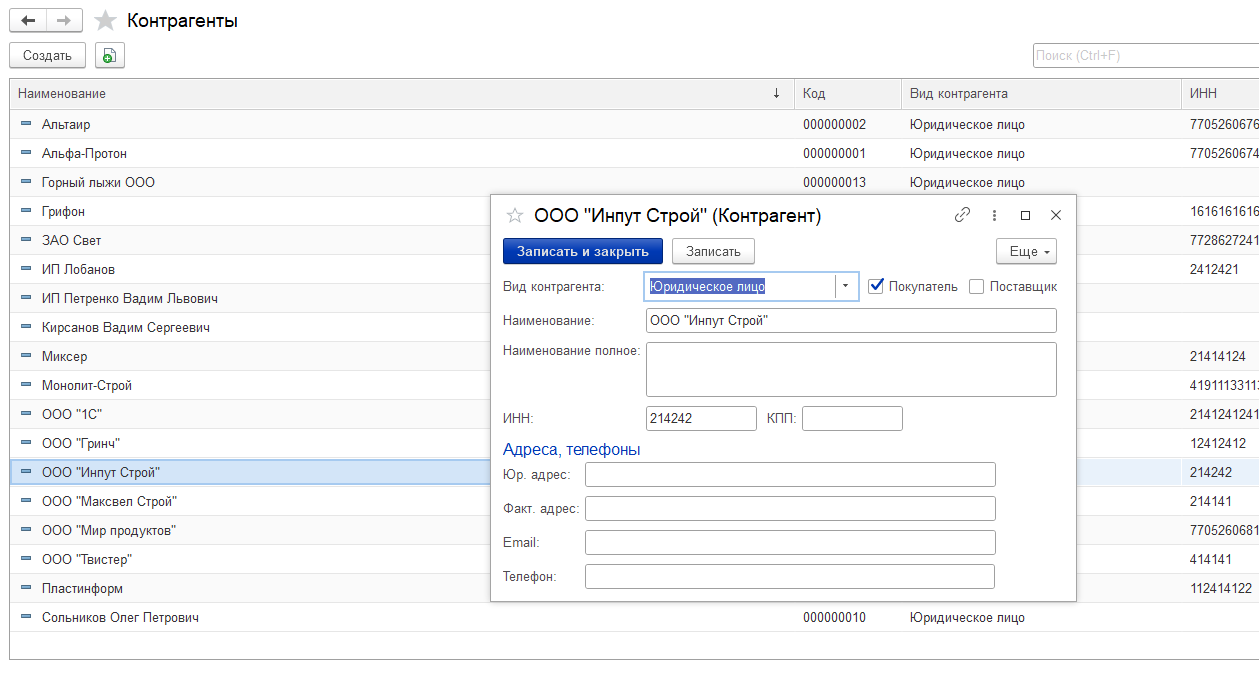


Теперь проверим на примере. Добавим в центральную базу ООО «Инпут Строй».

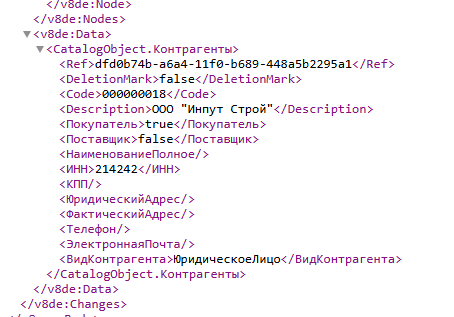


Теперь повторяем все те же действия, что были сделаны выше по сохранению сообщения в Центральной базе и чтению в Магазине №1.

Откроем справочник Контрагенты в базе Магазина №1 и увидим, что ООО «Инпут Строй» появился в списке Контрагентов.

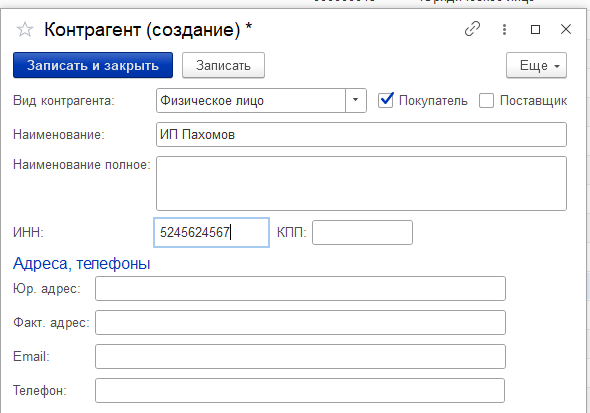
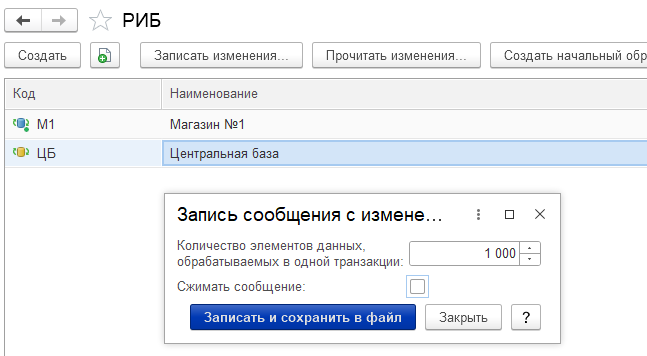


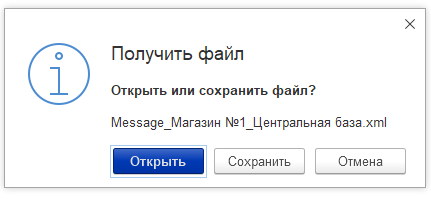
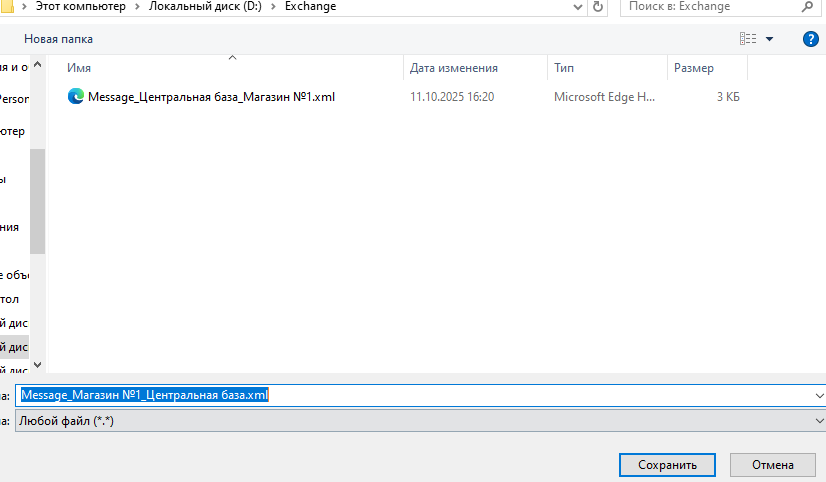
Если открыть сообщение обмена, то можно увидеть, что тут теперь появились сообщения об изменении.



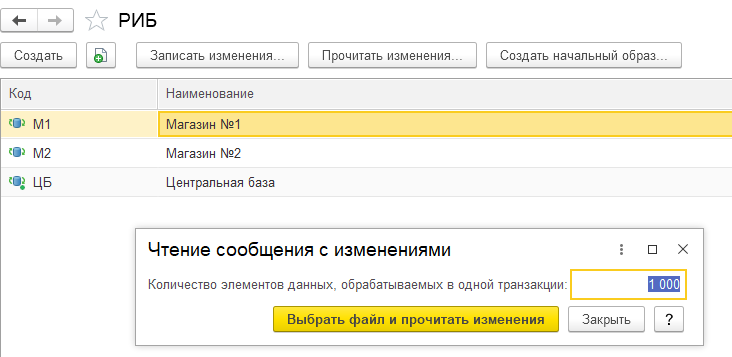
Эти данные выгружаются за счет механизма ***сериализации*** данных.

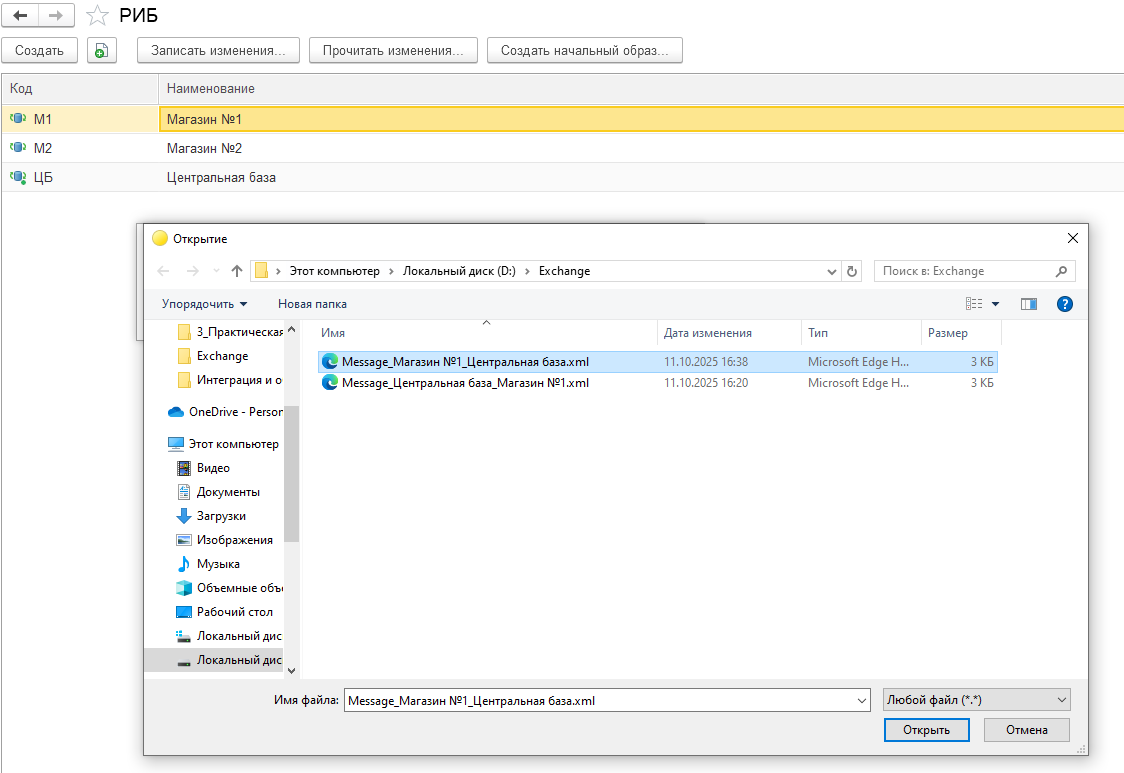
Тоже самое происходит и в обратном направлении. Если данные в Магазине №1 изменились, то их точно также можно выгрузить в Центральную базу. Добавим произвольного контрагента в базе Магазина №1 и выгрузим их в Центральную базу.

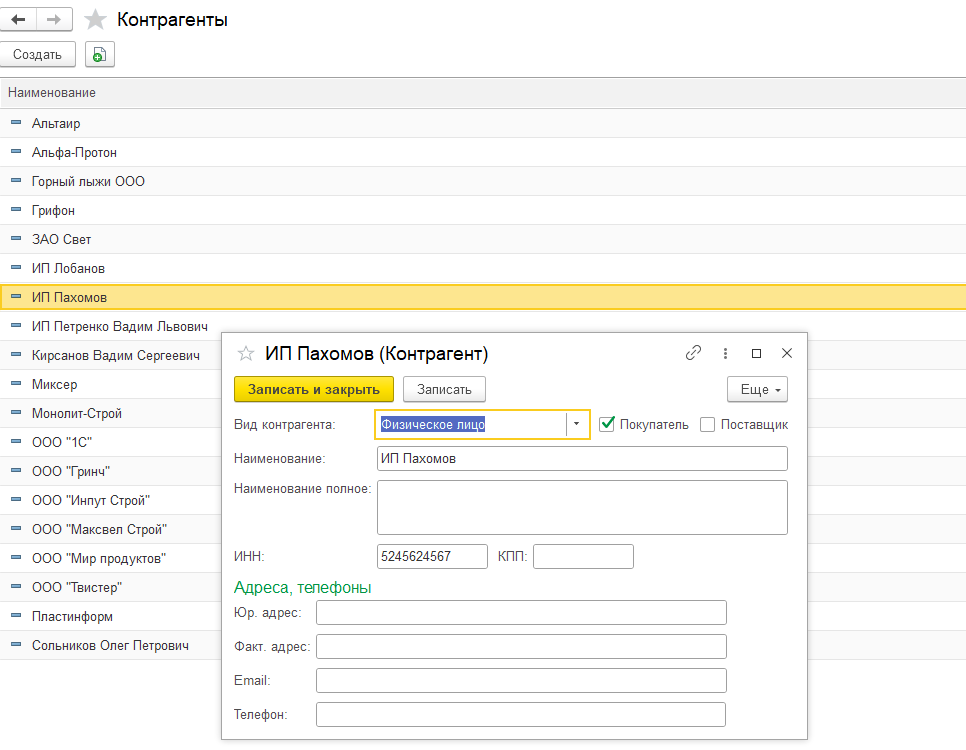
 





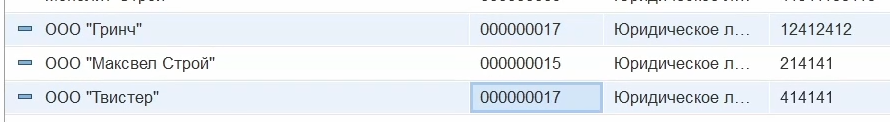




Также в центральную базу добавился ещё и пользователь, потому что он был добавлен в базу Магазина №1.

То есть, как вы видите мы не написали ни одной строки кода, а механизм передачи данных у нас уже есть, и он весьма успешно работает. Но, как и у всего, у штатного механизма есть недостатки:

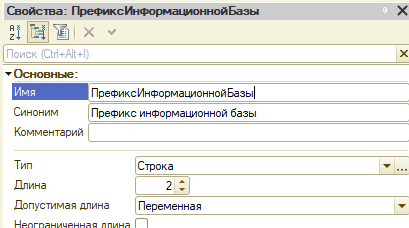
1. Не контролируется ссылочная целостность. То есть если был удален контрагент, на которого был уже оформлен документ, то при передаче данных о его удалении в документе будет просто «битая ссылка».
2. Дублирование кодов. То есть если одновременно в разные базы добавлено по одному Контрагенту, то оба контрагента запишутся в базы с одним и тем же кодом. И при обмене информацией получится, что в справочнике Контрагенты, будет два Контрагента с одним кодом. С номерами документов будет такая же история. Это на самом деле очень плохо.



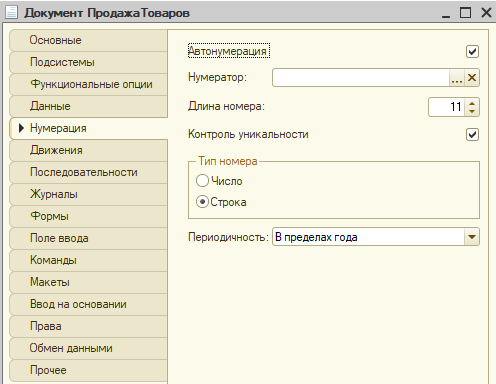
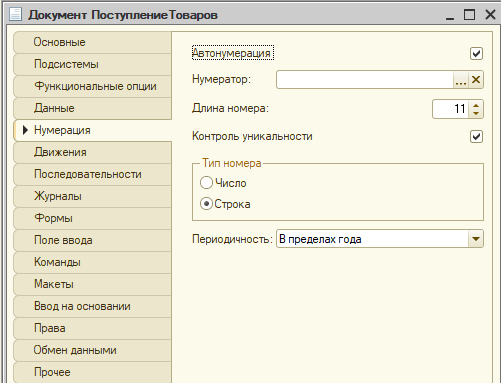
Решается последняя проблема путем префиксации документов или справочников.

Этим мы сейчас и займёмся.

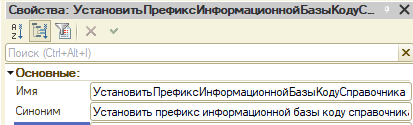
Открываем центральную базу. Добавляем константу ПрефиксИнформационнойБазы тип Строка, длина 2 символа.



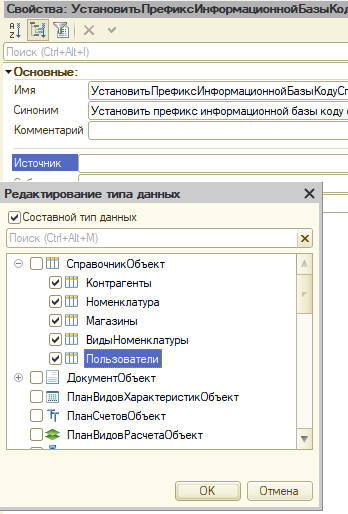
У документов увеличиваем длину номера до 11.



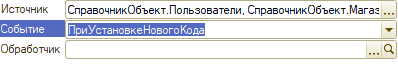
Саму префиксацию нужно реализовать через подписки на события.



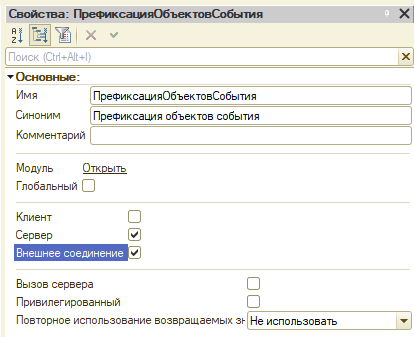
В качестве источника устанавливаем составной тип данных и выбираем все справочники



А событие при установке нового кода

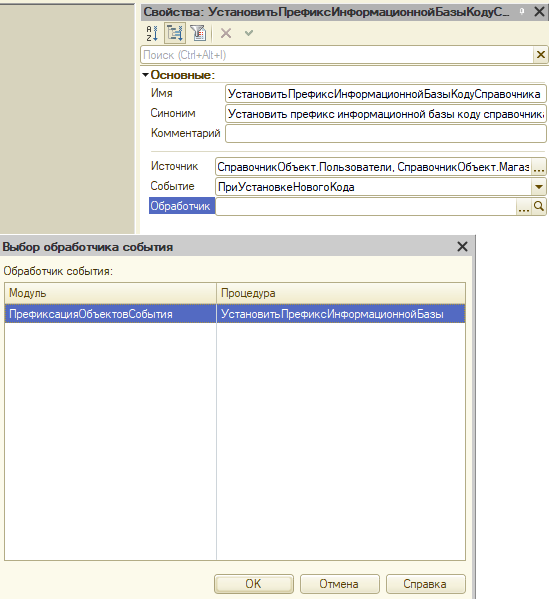


Добавляем общий модуль ПрефиксацияОбъектовСобытия

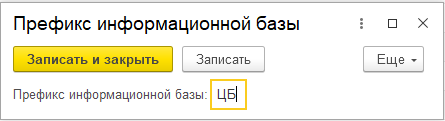


Вставляем в него код из Приложения!

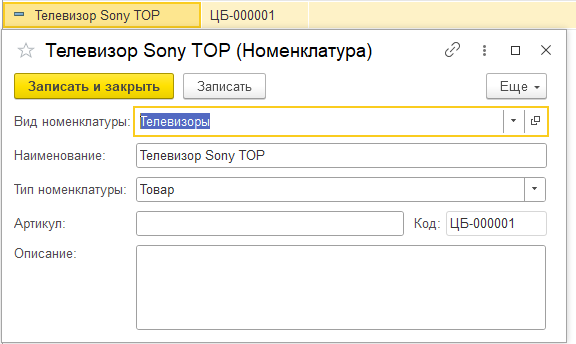
И назначаем его как обработчик из подписки.



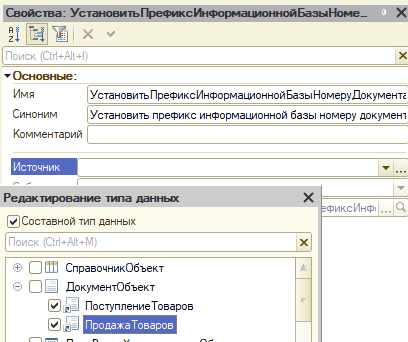
Проверим что получилось. Переходим в пользовательский режим и заполняем константу. Допустим, ЦБ.



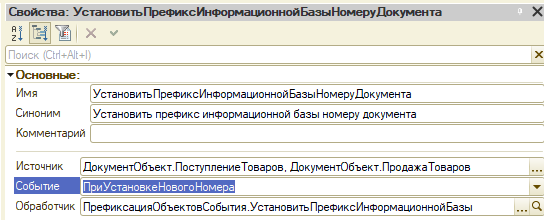
Открываем справочник Номенклатура и добавляем новый товар. После записи товара в коде проставится код с префиксом.



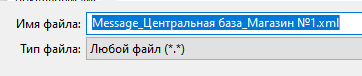
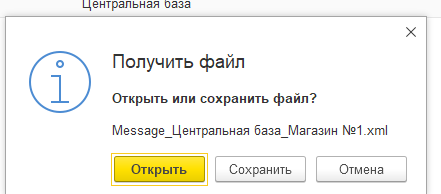
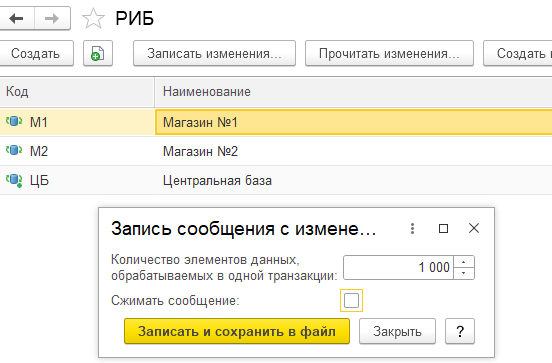
Теперь тоже самое нужно создать для документов. Добавляем отдельную подписку УстановитьПрефиксИнформационнойБазыНомеруДокумента.



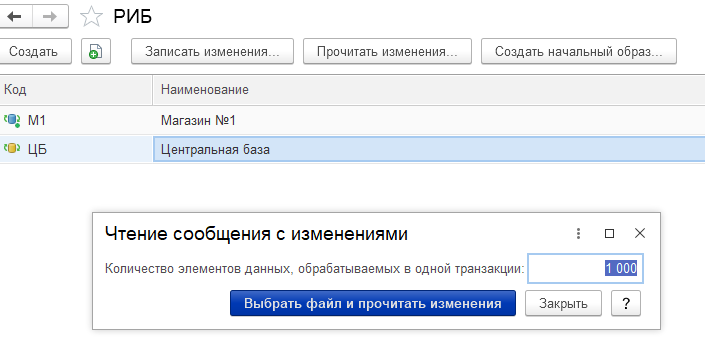
Событие выбираем ПриУстановкеНовогоНомера. Обработчик используем тот же самый.



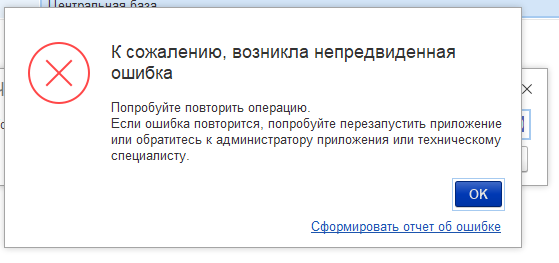
Теперь перейдем в пользовательский режим и сформируем сообщение обмена из Центральной базы в Магазин №1.



Заходим в пользовательский режим базы Магазин №1 и читаем сообщение обмена.



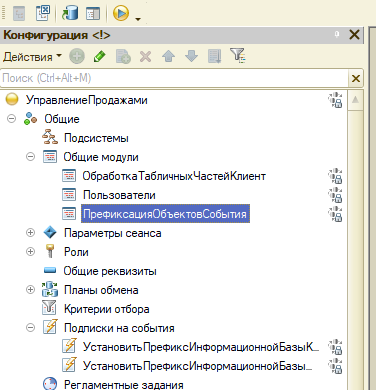
И появляется ошибка.



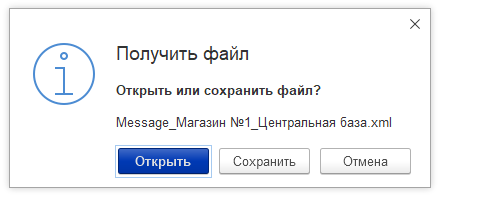
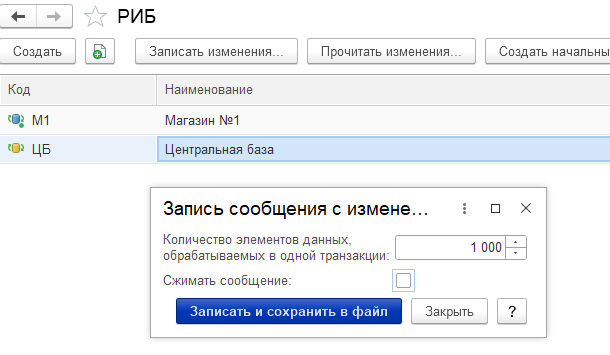
Но на самом деле это не ошибка!

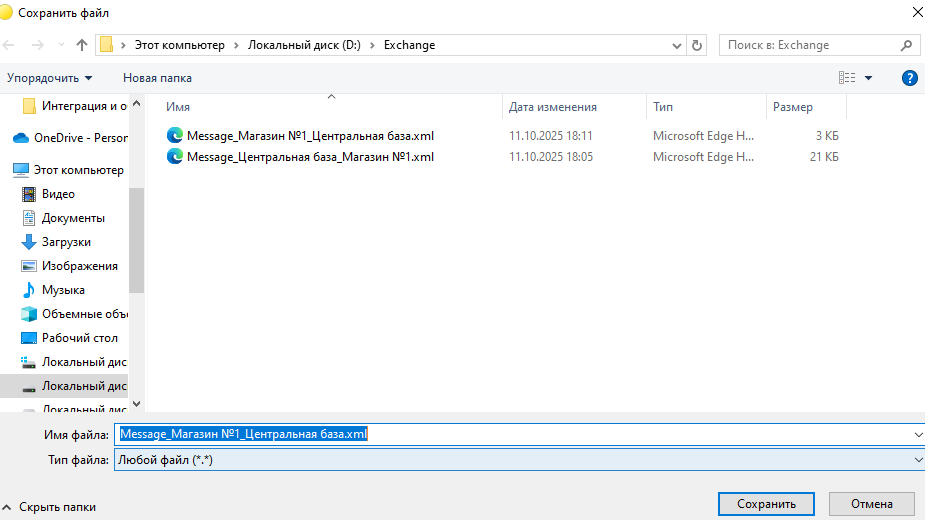
Нужно закрыть пользовательский режим базы Магазин №1 и зайти в конфигуратор.

Мы видим, что в ветке Общие всё появилось, но также горит значок, что конфигурация не обновлена.

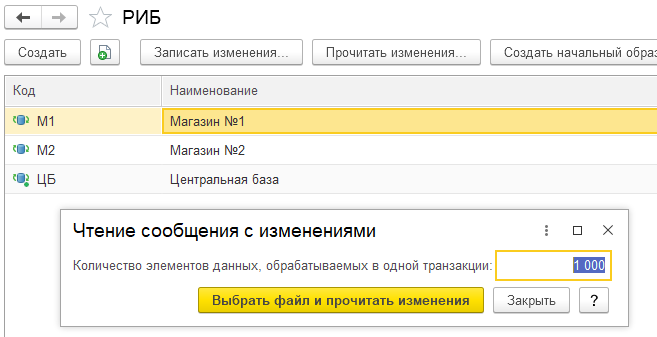


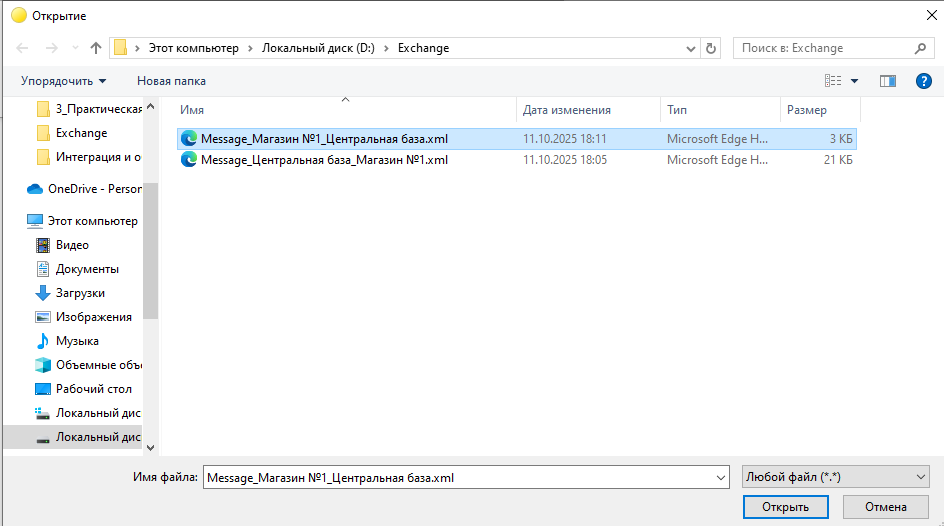
То есть нужно конфигурацию обновить и перейти в пользовательский режим для того, чтобы сформировать ответное сообщение и прошло квитирование данного сообщения обмена.





После чего в пользовательском режиме Центральной базы читаем это сообщение, для дальнейшего квитирования.



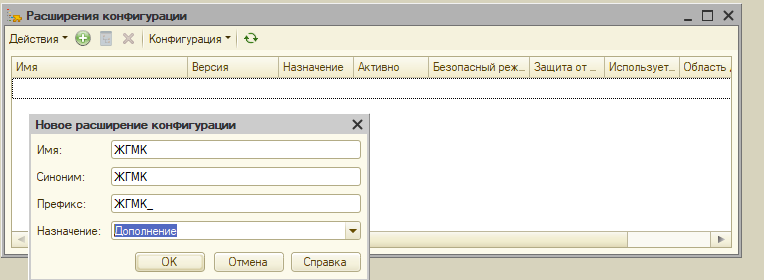


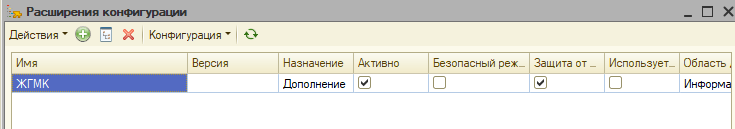
И несмотря на то, что внешне ничего не изменилось, но квитирование прошло.

Теперь рассмотрим ещё вопрос о том, как передаются расширения конфигураций. Как вы знаете, все изменения в типовых конфигурациях происходят через расширения, поэтому очень важно знать как происходит передача расширений.

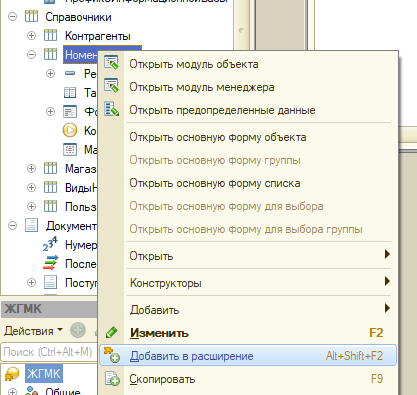
Допустим задача: Необходимо в справочник Номенклатура добавить какое-то поле с отображением идентификатора на сайте. Чтобы не сломать основную конфигурацию, все изменения желательно вносить чем расширения.

Добавим расширение ЖГМК. И снимаем флажок «Безопасный режим».

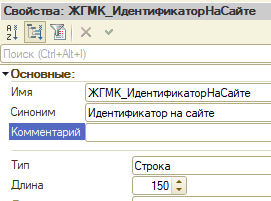




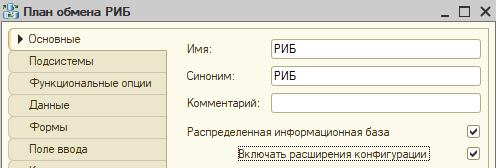
Открываем расширение. Заимствуем справочник Номенклатура.



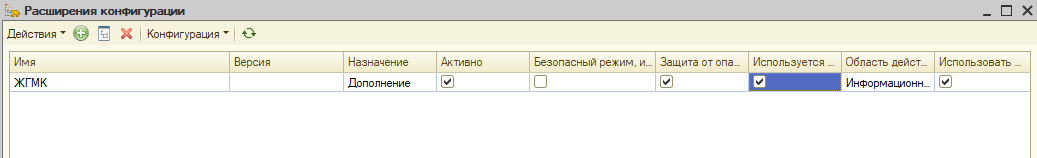
Добавляем в этот справочник в расширении, реквизит Идентификатор на сайте



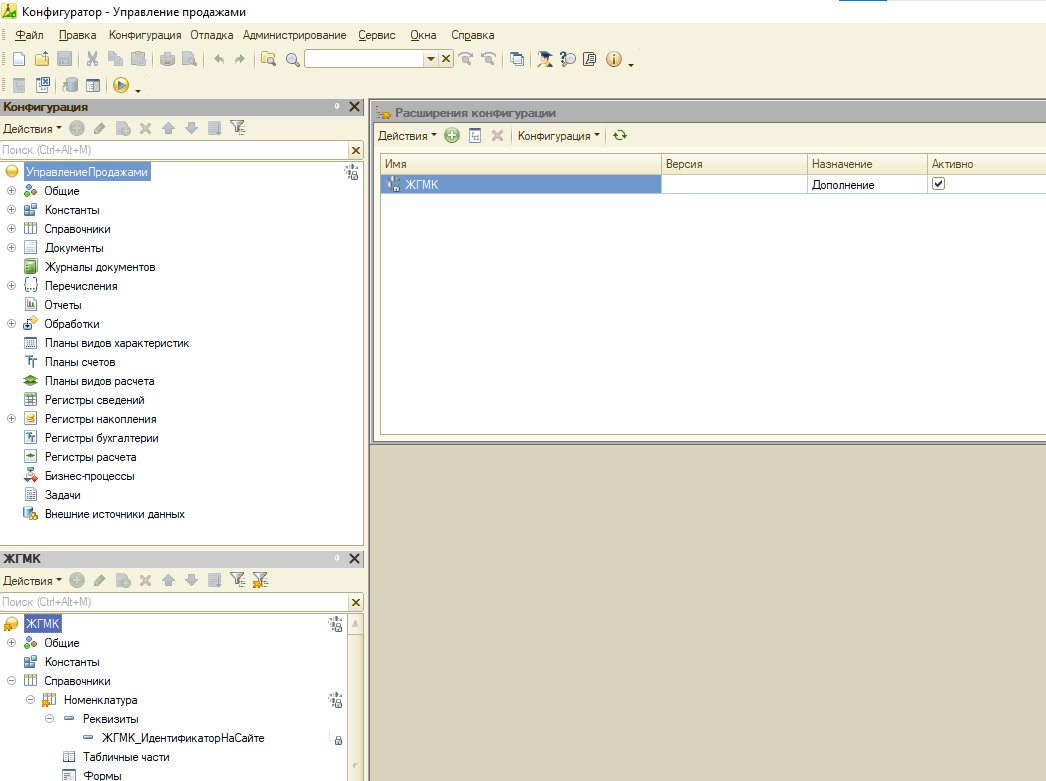
Обновляем конфигурацию и открываем наш план обмена, в котором отмечаем Включать расширения конфигураций.



Открываем само расширение и ставим флажок в Используется в распределенной ИБ



Перейдем в пользовательский режим и выгрузим сообщение обмена из Центральной базы в Магазин №1. Добавлять сейчас всю последовательность картинок не буду – уже выше делала это неоднократно! Так как перестраивается структура через добавление расширения, то тоже будет сообщение об «ошибке». Нужно закрыть пользовательский режим и открыть конфигуратор у базы Магазин №1 и обновить конфигурацию. В результате добавится расширение.

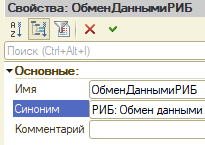


Как вы видите по значкам возле конфигурации расширения, эти данные тоже можно изменять только в корневом узле.

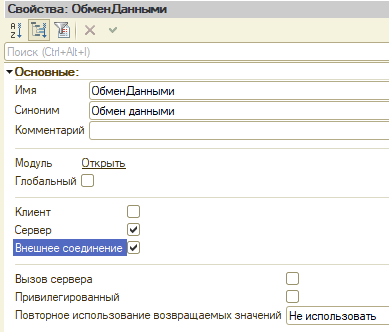
Программная работа с начальным образом информационной базы.

Используется для того, чтобы организовать регламентное задание, а не постоянно нажимать на кнопки.

Добавляем в Центральной базе регламентное задание ОбменДаннымиРИБ.

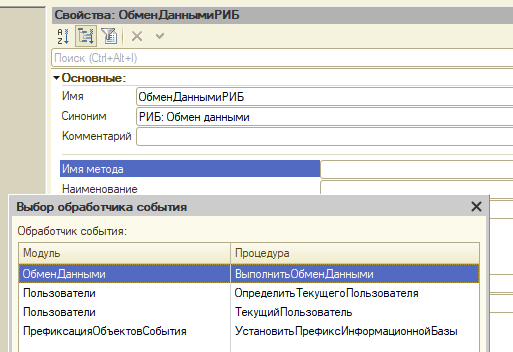


Для организации работы регламентного задания добавляем Общий модуль ОбменДанными.

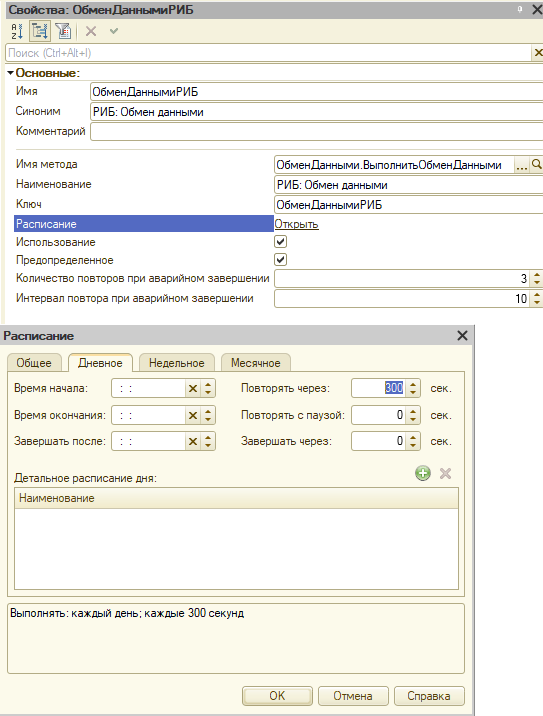


Текст модуля будет в Приложении!

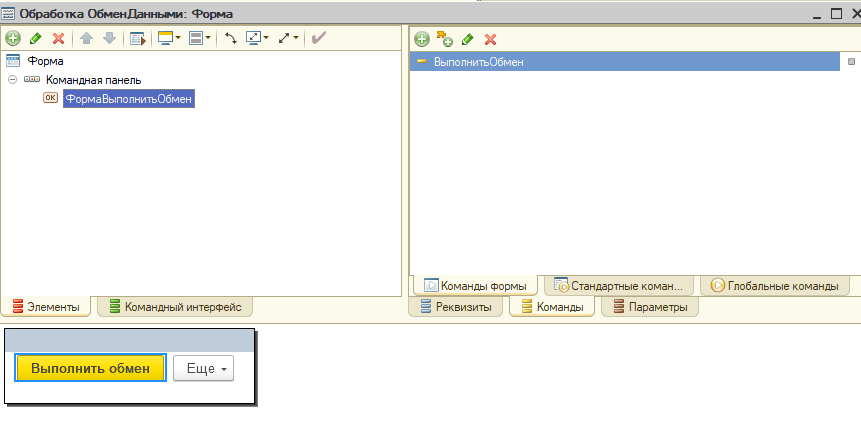
Возвращаемся к регламентному заданию и выбираем метод ВыполнитьОбменДанными.



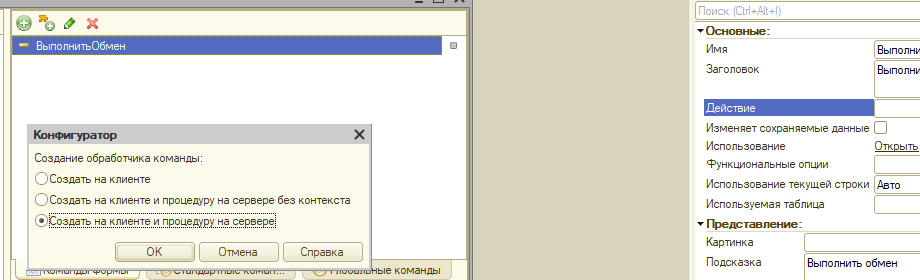
Внешний вид регламентного задания представлен ниже.



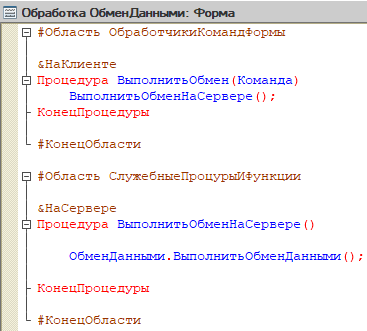
Создаем обработку ОбменДанных. Создаем форму. На форму добавляем команду ВыполнитьОбмен. Перетаскиваем на форму и отмечаем, что кнопка по умолчанию.



Создаем действие команды с вызовом процедуры на сервере.

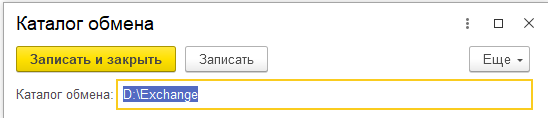
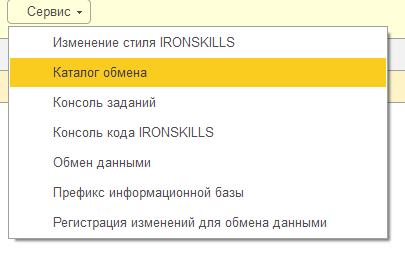


Пишем модуль

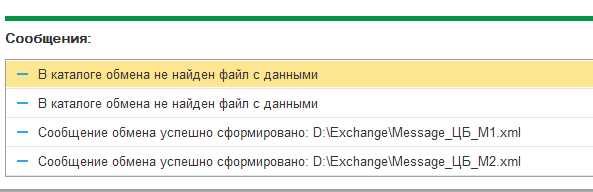


Обновляем конфигурацию и переходим в пользовательский режим.

Проверяем каталог обмена, что он у вас записан верно.



После чего вызываете созданную обработку и проверяете, что у вас сообщения формируются.

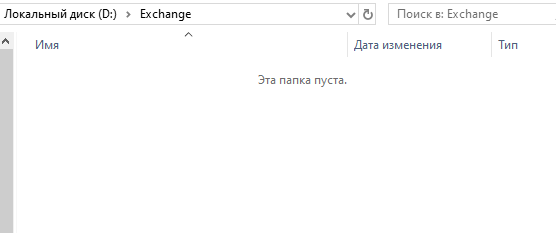


Первые две строки это из-за того, что в каталоге не было сообщений обмена от баз Магазина №1 и Магазина №2(которую мы не создавали!).

Теперь **вручную(через РИБ)** нужно создать сообщение обмена для Магазина №1. Прочитать это сообщение в базе Магазина №1. Опять будет «ошибка», по которой нужно обновить конфигурацию Магазина №1. После этого сделать вручную сообщение для Центральной базы и прочитать его в центральной базе. Удаляем пока что узел Магазин №2 из списка РИБ в центральной базе. Он нам был нужен только для теста того, что сообщения формируются по всем узлам.

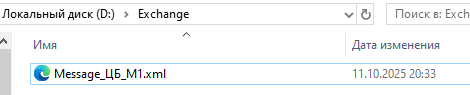
Теперь протестируем действия при использовании обработки ОбменДанными.

Очистим наш каталог обмена.

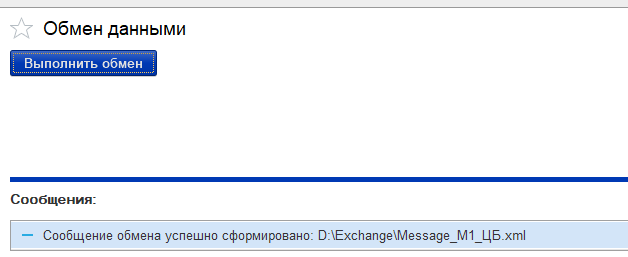


Меняем данные в центральной базе, допустим контрагент ЗАО «Свет» стал ПАО «Свет». Запускаем обмен данными в центральной базе.

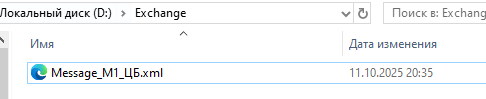
В результате сформировался файл.



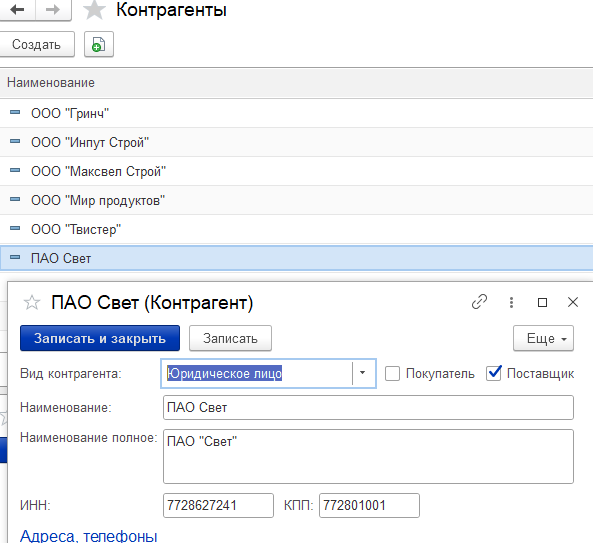
В базе Магазин №1 проверяем адрес каталога обмена и выполняем обмен.



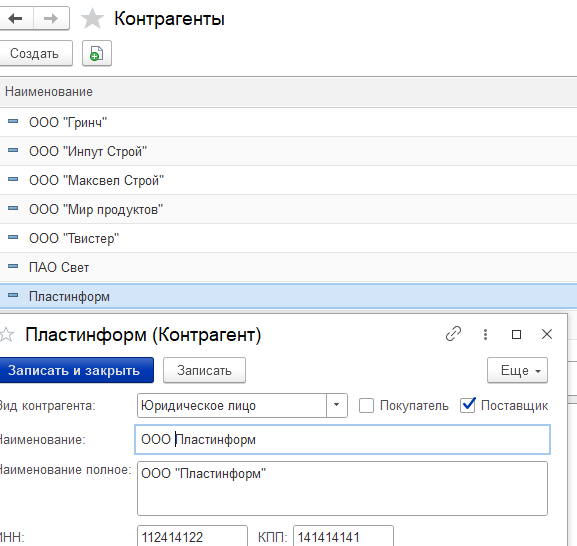
Видим, что сообщение заменилось.



Проверяем контрагента и видим что теперь в базе Магазина №1 у нас ПАО Свет

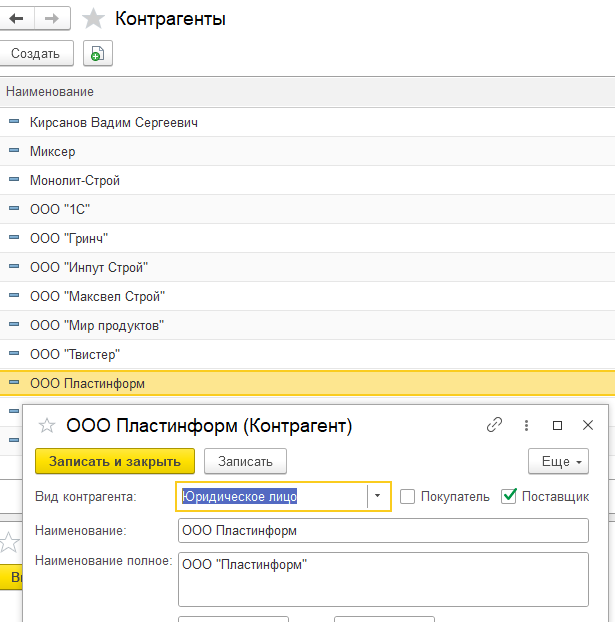


Теперь в базе магазина №1 меняем контрагента Пластинформ, на ООО Пластинформ.

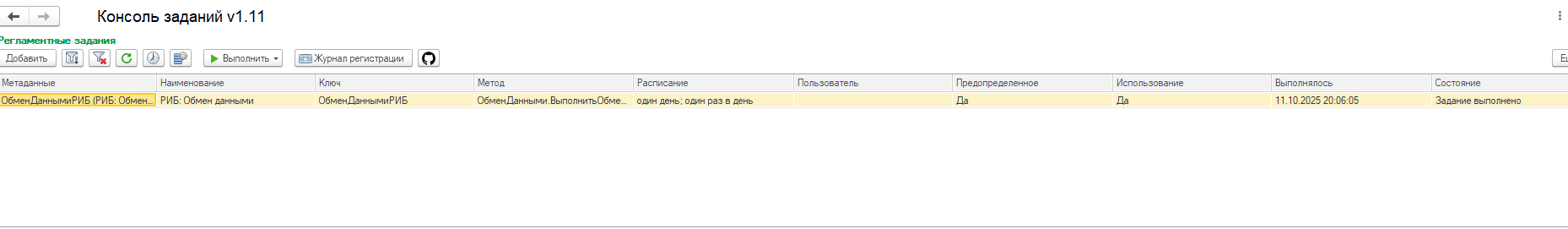


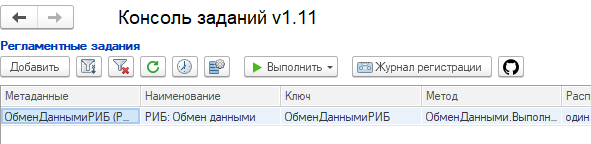
Нажимаем выполнить обмен в Магазине №1 и потом в Центральной базе.

Проверяем контрагентов в центральной базе и видим, что данные по Пластинформу сменились.



Осталось проверить через консоль заданий, что регламентное задание работает. Проверяем и в Центральной базе и в Магазине №1.

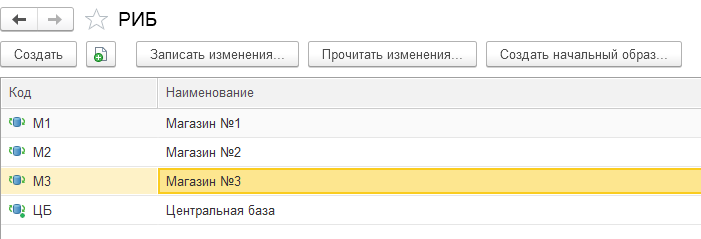




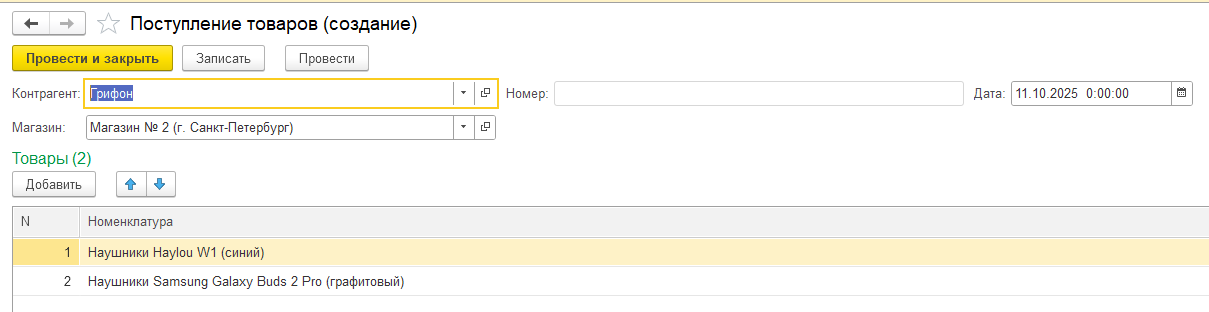
Управление миграцией данных.

Как сделать так, чтобы для Магазина №1 из Центральной базы выгружались данные, только касающиеся Магазина №1 и не загружались по Магазину № 2. Или есть одна ваша центральная база и несколько юрлиц, которым нужно загружать только их данные.

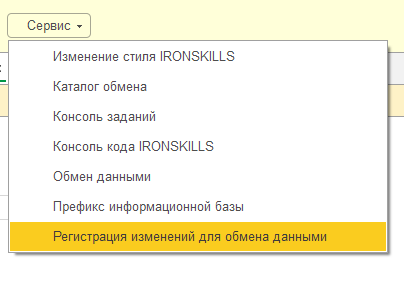
Заходим в Центральную базу и добавляем ещё Магазин №2 и Магазин №3.



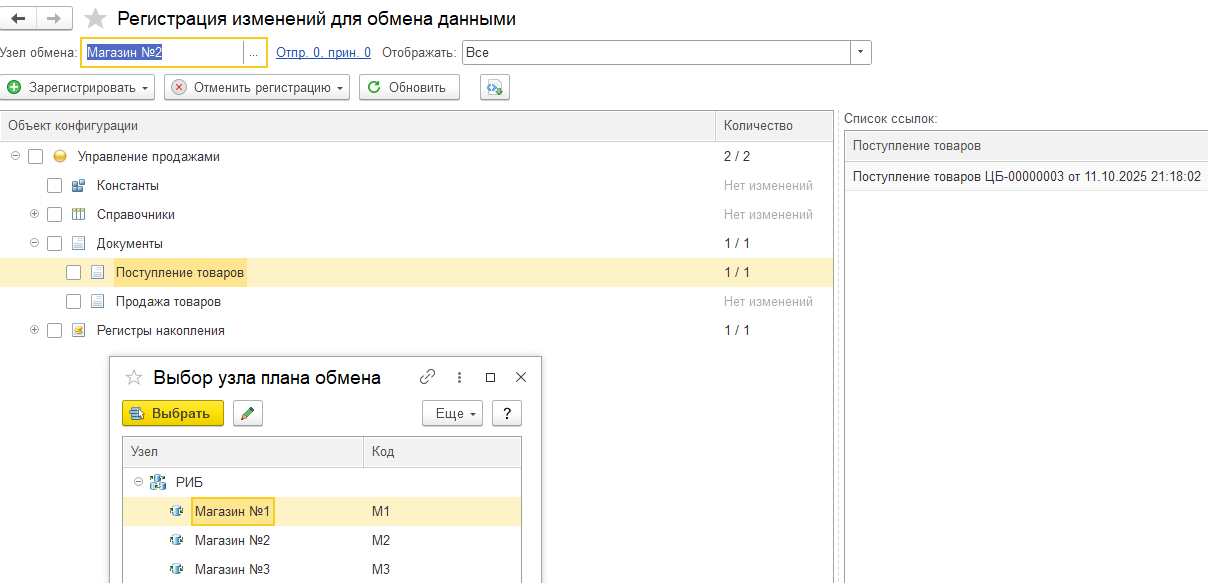
Откроем-закроем или добавим копированием любой из документов.

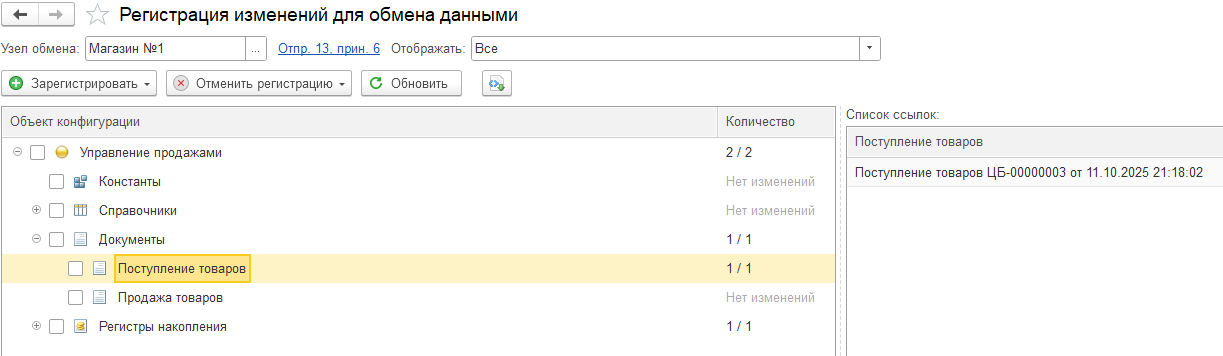


Откроем обработку регистрация изменений для обмена данными.



И если мы проверим по всем узлам обмена, то получится, что этот документ, который должен быть адресован Магазину №2, отображается как измененные данные и для Магазина №1, и для Магазина №2, и для Магазина №3.



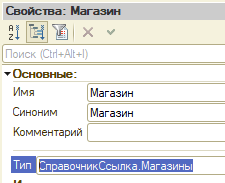
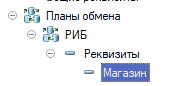


А хотелось бы, чтобы только для Магазина №2.

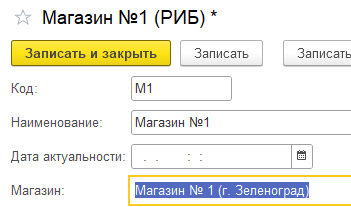
Как это реализовать?

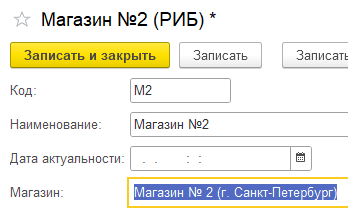
Для этого нам потребуется отказаться от автоматической регистрации и регистрацию выполнять программно.

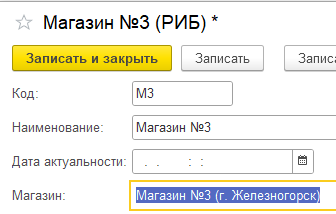
Добавляем в план обмена реквизит Магазин с типом данных СправочникСсылка.Магазины.



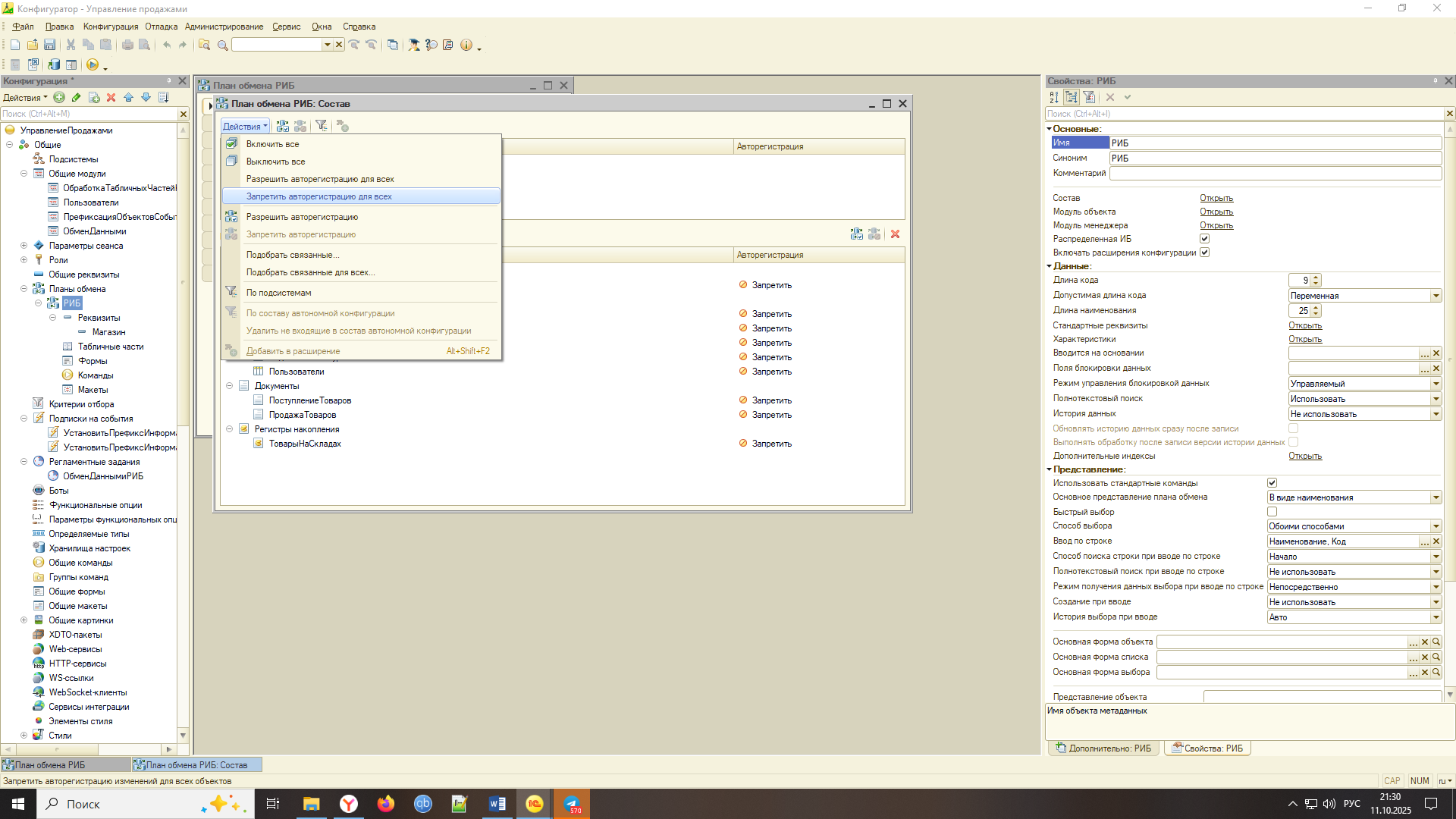
Обновить конфигурацию и перейти в пользовательский режим. Открыть РИБ и для Магазина №1 указать Магазин №1, для Магазина №2 указать Магазин №2 и тд.

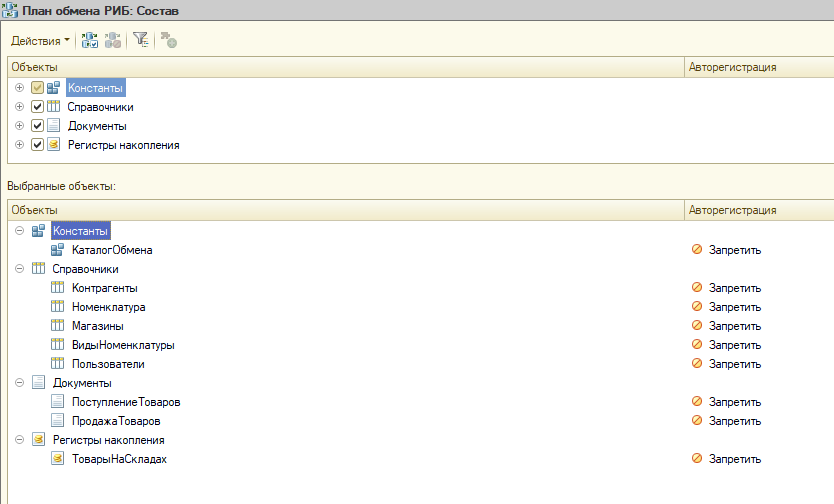




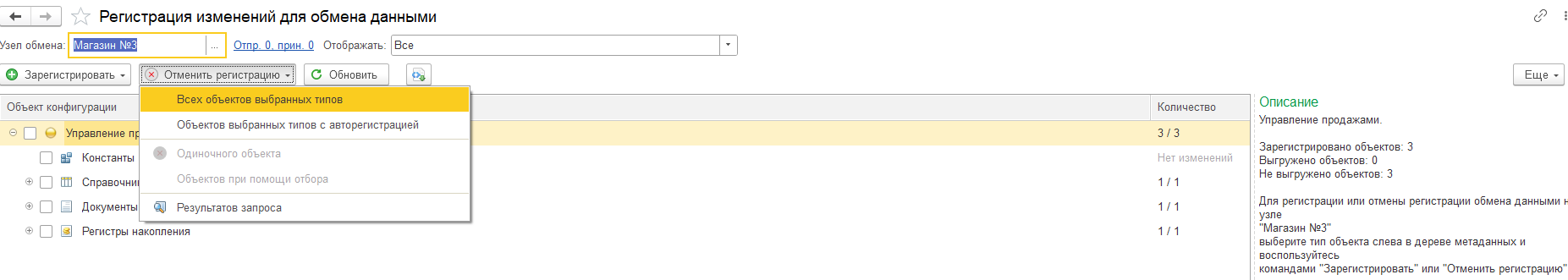


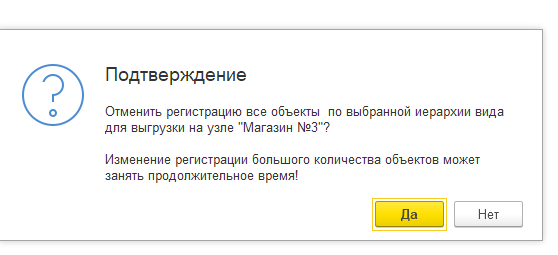
Открываем состав РИБ в конфигураторе и запрещаем авторегистрацию для всех.

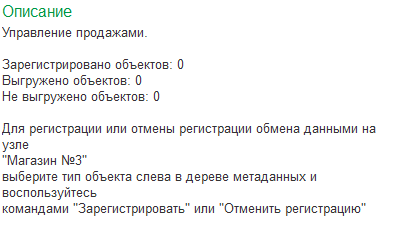




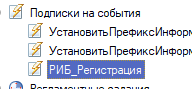
Обновляем конфигурацию и переходим в пользовательский режим чтобы при помощи обработки Регистрация изменений для обмена данных, удалить регистрацию всех объектов. Делаем для каждого Магазина в отдельности.



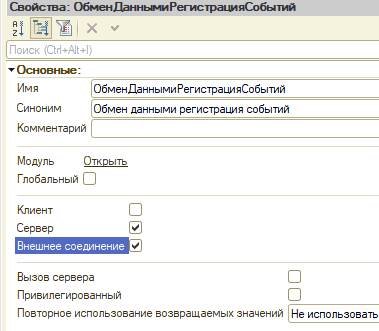




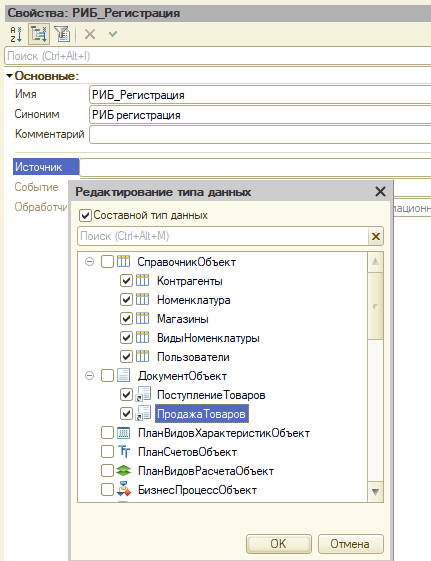
Это можно сделать при помощи подписки. Создаем подписку РИБ\_Регистрация.



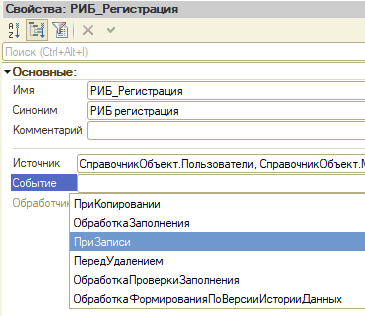
Добавляем общий модуль ОбменДаннымиРегистрацияСобытий.



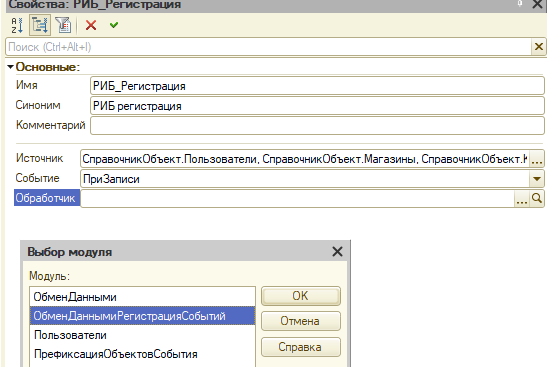
Вернемся к Подписке и в качестве источников выберем все справочники и все документы.

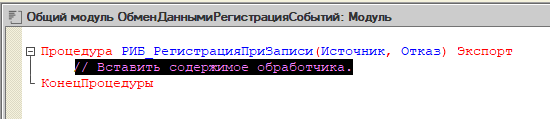


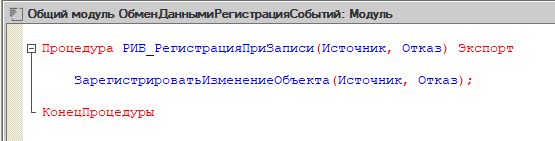
Выберем событие ПриЗаписи.



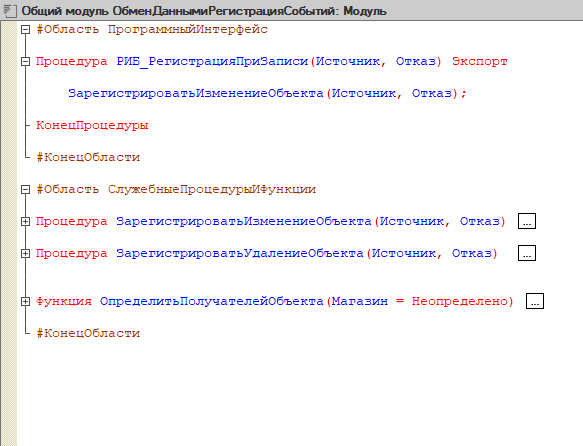
Событие создадим в модуле ОбменДаннымиРегистрацияСобытий. Нажимаем на «увеличительное стекло» и выбираем нужный модуль.



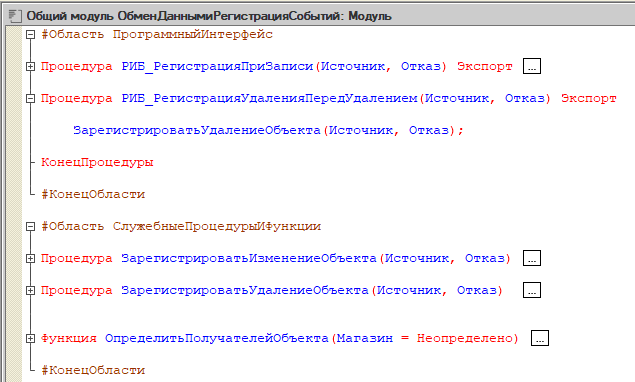




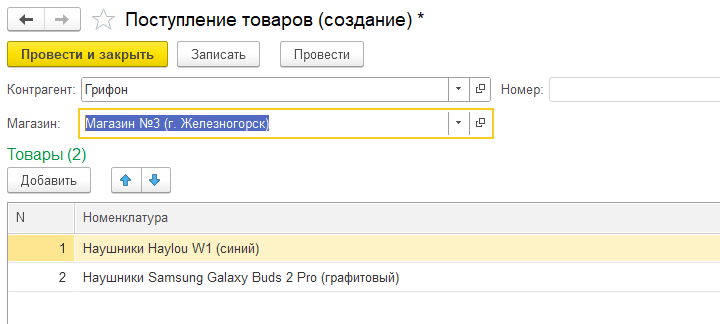
Ниже добавляем процедуры из Приложения и общий модуль будет выглядеть так:



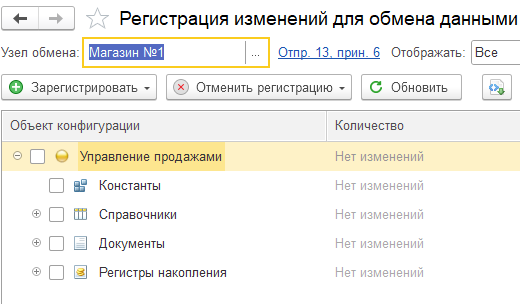
В области программного интерфейса осталось добавить задел на будущее, когда мы будем описывать ещё одну подписку.

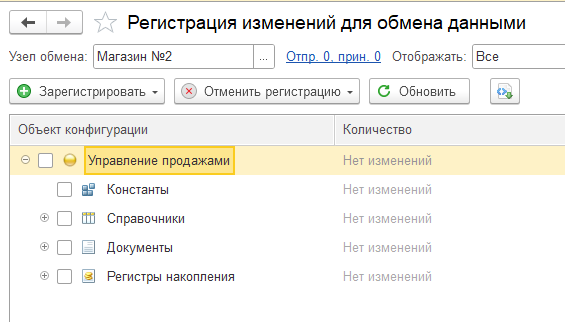


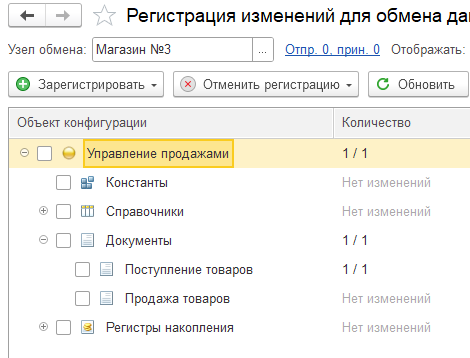
Обновляем конфигурацию и переходим в пользовательский режим. Добавляем копированием документ и меняем магазин на Магазин №3.



Через обработку для регистрации изменений проверяем все магазины.





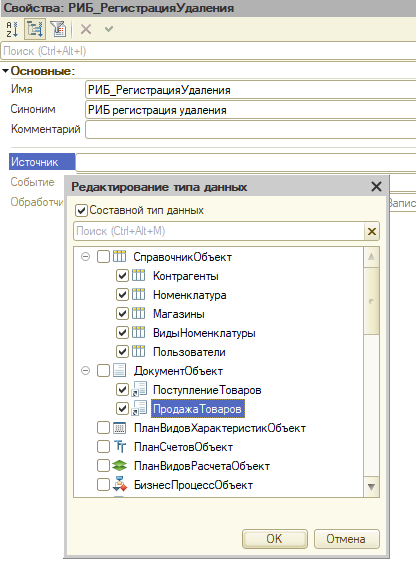


Видим, что изменения зарегистрировались только для магазина №3.

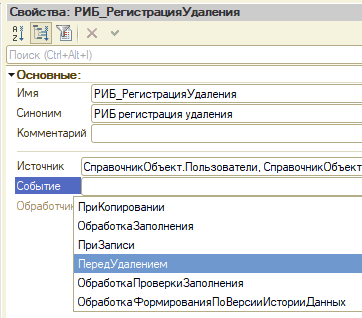
Осталось доделать регистрацию удаления объектов.

Добавляем подписку на события РИБ\_РегистрацияУдаления.

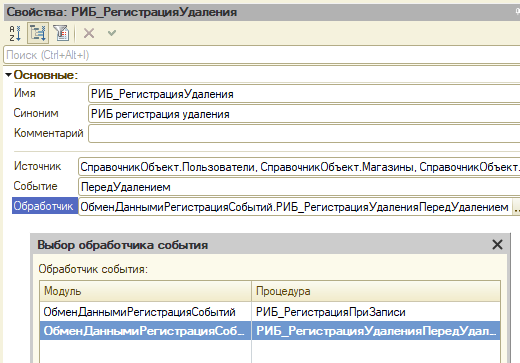
Источником будут все те же самые объекты.



Событие перед удалением.



А в качестве обработчика выберем уже существующий модуль.



Теперь можно обновить конфигурацию и перейти в пользовательский режим, чтобы проверить, что всё работает как надо.

Приложение

#Область ПрограммныйИнтерфейс

// Устанавливает префикс источника подписки в соответствии с префиксом информационной базы.

// Ограничения на реквизиты источника не накладываются.

//

// Параметры:

// Источник - Произвольный - источник события подписки.

// Любой объект из множества [Справочник, Документ, План видов характеристик, Бизнес процесс, Задача].

// СтандартнаяОбработка - Булево - флаг стандартной обработки подписки.

// Префикс - Строка - префикс объекта, который нужно изменить.

//

Процедура УстановитьПрефиксИнформационнойБазы(Источник, СтандартнаяОбработка, Префикс) Экспорт

ПрефиксИнформационнойБазы = Константы.ПрефиксИнформационнойБазы.Получить();

Если ПустаяСтрока(ПрефиксИнформационнойБазы) Тогда

ПрефиксИнформационнойБазы = "00";

КонецЕсли;

ШаблонПрефикса = "[ИБ]-[Префикс]";

ШаблонПрефикса = СтрЗаменить(ШаблонПрефикса, "[ИБ]", ПрефиксИнформационнойБазы);

ШаблонПрефикса = СтрЗаменить(ШаблонПрефикса, "[Префикс]", Префикс);

Префикс = ШаблонПрефикса;

КонецПроцедуры

#КонецОбласти

#Область ПрограммныйИнтерфейс

Процедура ВыполнитьОбменДанными() Экспорт

ЗагрузитьДанные();

ВыгрузитьДанные();

КонецПроцедуры

#КонецОбласти

#Область СлужебныеПроцедурыИФункции

Процедура ЗагрузитьДанные()

КаталогОбмена = Константы.КаталогОбмена.Получить();

Если ПустаяСтрока(КаталогОбмена) Тогда

ТекстОшибки = "Не заполнен каталог обмена. Нужно заполнить каталог в настройках";

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = ТекстОшибки;

Сообщение.Сообщить();

ЗаписьЖурналаРегистрации(ИмяСобытияЖР(), УровеньЖурналаРегистрации.Ошибка,,,ТекстОшибки);

Возврат;

КонецЕсли;

Каталог = Новый Файл(КаталогОбмена);

Если НЕ Каталог.Существует() Тогда

ТекстОшибки = "Выбранный каталог обмена не существует на сервере";

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = ТекстОшибки;

Сообщение.Сообщить();

ЗаписьЖурналаРегистрации(ИмяСобытияЖР(), УровеньЖурналаРегистрации.Ошибка,,,ТекстОшибки);

Возврат;

КонецЕсли;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| РИБ.Ссылка КАК Ссылка,

| РИБ.Код КАК Код

|ИЗ

| ПланОбмена.РИБ КАК РИБ

|ГДЕ

| НЕ РИБ.ПометкаУдаления

| И НЕ РИБ.ЭтотУзел";

РезультатЗапроса = Запрос.Выполнить();

Выборка = РезультатЗапроса.Выбрать();

ПараметрыОбмена = Новый Структура;

ПараметрыОбмена.Вставить("КаталогОбмена", КаталогОбмена);

ПараметрыОбмена.Вставить("ЭтотУзел", ПланыОбмена.РИБ.ЭтотУзел());

Пока Выборка.Следующий() Цикл

ПараметрыОбмена.Вставить("УзелОбмена", Выборка.Ссылка);

ПрочитатьСообщениеОбмена(ПараметрыОбмена);

КонецЦикла;

КонецПроцедуры

Процедура ВыгрузитьДанные()

КаталогОбмена = Константы.КаталогОбмена.Получить();

Если ПустаяСтрока(КаталогОбмена) Тогда

ТекстОшибки = "Не заполнен каталог обмена. Нужно заполнить каталог в настройках";

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = ТекстОшибки;

Сообщение.Сообщить();

ЗаписьЖурналаРегистрации(ИмяСобытияЖР(), УровеньЖурналаРегистрации.Ошибка,,,ТекстОшибки);

Возврат;

КонецЕсли;

Каталог = Новый Файл(КаталогОбмена);

Если НЕ Каталог.Существует() Тогда

ТекстОшибки = "Выбранный каталог обмена не существует на сервере";

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = ТекстОшибки;

Сообщение.Сообщить();

ЗаписьЖурналаРегистрации(ИмяСобытияЖР(), УровеньЖурналаРегистрации.Ошибка,,,ТекстОшибки);

Возврат;

КонецЕсли;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| РИБ.Ссылка КАК Ссылка,

| РИБ.Код КАК Код

|ИЗ

| ПланОбмена.РИБ КАК РИБ

|ГДЕ

| НЕ РИБ.ПометкаУдаления

| И НЕ РИБ.ЭтотУзел";

РезультатЗапроса = Запрос.Выполнить();

Выборка = РезультатЗапроса.Выбрать();

ПараметрыОбмена = Новый Структура;

ПараметрыОбмена.Вставить("КаталогОбмена", КаталогОбмена);

ПараметрыОбмена.Вставить("ЭтотУзел", ПланыОбмена.РИБ.ЭтотУзел());

Пока Выборка.Следующий() Цикл

ПараметрыОбмена.Вставить("УзелОбмена", Выборка.Ссылка);

ЗаписатьСообщениеОбмена(ПараметрыОбмена);

КонецЦикла;

КонецПроцедуры

Процедура ЗаписатьСообщениеОбмена(ПараметрыОбмена)

// Формируем имя файла сообщения обмена

УзелОтправитель = ПараметрыОбмена.ЭтотУзел;

УзелПолучатель = ПараметрыОбмена.УзелОбмена;

КаталогОбмена = ПараметрыОбмена.КаталогОбмена;

ИмяФайлаСообщения = КаталогОбмена + СтрШаблон("\Message\_%1\_%2.xml",

УзелОтправитель.Код, УзелПолучатель.Код);

// Открываем файл для записи

ЗаписьXML = Новый ЗаписьXML();

ЗаписьXML.ОткрытьФайл(ИмяФайлаСообщения);

// Создаем запись сообщения

ЗаписьСообщения = ПланыОбмена.СоздатьЗаписьСообщения();

ЗаписьСообщения.НачатьЗапись(ЗаписьXML, УзелПолучатель);

// Записываем изменения в файл

ПланыОбмена.ЗаписатьИзменения(ЗаписьСообщения);

ЗаписьСообщения.ЗакончитьЗапись();

// Закрываем файл

ЗаписьXML.Закрыть();

Текст = "Сообщение обмена успешно сформировано: " + ИмяФайлаСообщения;

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = Текст;

Сообщение.Сообщить();

ЗаписьЖурналаРегистрации(ИмяСобытияЖР(), УровеньЖурналаРегистрации.Информация,,,Текст);

КонецПроцедуры

Процедура ПрочитатьСообщениеОбмена(ПараметрыОбмена)

// Формируем имя файла сообщения обмена

УзелОтправитель = ПараметрыОбмена.УзелОбмена;

УзелПолучатель = ПараметрыОбмена.ЭтотУзел;

КаталогОбмена = ПараметрыОбмена.КаталогОбмена;

ИмяФайлаСообщения = КаталогОбмена + СтрШаблон("\Message\_%1\_%2.xml",

УзелОтправитель.Код, УзелПолучатель.Код);

Файл = Новый Файл(ИмяФайлаСообщения);

Если НЕ Файл.Существует() Тогда

Текст = "В каталоге обмена не найден файл с данными";

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = Текст;

Сообщение.Сообщить();

ЗаписьЖурналаРегистрации(ИмяСобытияЖР(), УровеньЖурналаРегистрации.Информация, ,, Текст);

Возврат;

КонецЕсли;

// Открываем файл для чтения

ЧтениеXML = Новый ЧтениеXML;

ЧтениеXML.ОткрытьФайл(ИмяФайлаСообщения);

// Создаем чтение сообщения

ЧтениеСообщения = ПланыОбмена.СоздатьЧтениеСообщения();

СообщениеПрочитано = Ложь;

// Чтение сообщения обмена

Попытка

ЧтениеСообщения.НачатьЧтение(ЧтениеXML);

ПланыОбмена.ПрочитатьИзменения(ЧтениеСообщения);

СообщениеПрочитано = Истина;

Исключение

ТекстОшибки = ОбработкаОшибок.КраткоеПредставлениеОшибки(ИнформацияОбОшибке());

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = ТекстОшибки;

Сообщение.Сообщить();

КонецПопытки;

// Завершить чтение сообщения

ЧтениеСообщения.ЗакончитьЧтение();

ЧтениеXML.Закрыть();

Если СообщениеПрочитано Тогда

УдалитьФайлы(ИмяФайлаСообщения);

КонецЕсли;

КонецПроцедуры

Функция ИмяСобытияЖР()

Возврат "ОбменДанными.РИБ";

КонецФункции

#КонецОбласти

#Область СлужебныеПроцедурыИФункции

Процедура ЗарегистрироватьИзменениеОбъекта(Источник, Отказ)

Если Отказ Тогда

Возврат;

КонецЕсли;

ЭтоДокумент = Документы.ТипВсеСсылки().СодержитТип(ТипЗнч(Источник.Ссылка));

Магазин = ?(ЭтоДокумент, Источник.Магазин, Неопределено);

УзлыОбмена = ОпределитьПолучателейОбъекта(Магазин);

ПланыОбмена.ЗарегистрироватьИзменения(УзлыОбмена, Источник);

КонецПроцедуры

Процедура ЗарегистрироватьУдалениеОбъекта(Источник, Отказ)

Если Отказ Тогда

Возврат;

КонецЕсли;

ЭтоДокумент = Документы.ТипВсеСсылки().СодержитТип(ТипЗнч(Источник.Ссылка));

Магазин = ?(ЭтоДокумент, Источник.Магазин, Неопределено);

УзлыОбмена = ОпределитьПолучателейОбъекта(Магазин);

ПланыОбмена.ЗарегистрироватьИзменения(УзлыОбмена, Новый УдалениеОбъекта(Источник.Ссылка));

КонецПроцедуры

Функция ОпределитьПолучателейОбъекта(Магазин = Неопределено)

УзлыОбмена = Новый Массив;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| РИБ.Ссылка КАК Ссылка

|ИЗ

| ПланОбмена.РИБ КАК РИБ

|ГДЕ

| НЕ РИБ.ПометкаУдаления

| И НЕ РИБ.ЭтотУзел";

Если Магазин <> Неопределено Тогда

Запрос.Текст = Запрос.Текст + "

|И РИБ.Магазин = &Магазин";

Запрос.УстановитьПараметр("Магазин", Магазин);

КонецЕсли;

РезультатЗапроса = Запрос.Выполнить();

Выборка = РезультатЗапроса.Выбрать();

Пока Выборка.Следующий() Цикл

УзлыОбмена.Добавить(Выборка.Ссылка);

КонецЦикла;

Возврат УзлыОбмена;

КонецФункции

#КонецОбласти