

Универсальный интерфейс для подключения к модулю  
«RestoMarket PriceChecker»

из любых конфигураций для платформ 1С 7.7 или 1С 8.x (BAF 8.x),  
в т. ч. из любых внешних приложений

(COM-объект RRC\_back.dll)

## Назначение:

COM-объект обеспечивает для конфигураций 1С 7.7 и 1С 8.х (BAF 8.х), а так же для любого приложения, управление справочником товаров модуля «PriceChecker».

## Подключение к COM-объекту:

COM-объект представляет из себя библиотеку RRC\_back.dll , которая должна быть зарегистрирована в реестре. (например из командной строки: `regsvr32 C:\Minimarket\RRC_back.dll`)

COM-объект содержит класс "RRCback", находящийся в пространстве имён "RRC\_back".

Соответственно, чтобы подключить его к Вашему приложению следует создать в приложении объект, например:

```
для C++:      Front = new RRC_back.RRCback();
для VB:      Set Front = CreateObject("RRC_back.RRCback")
для 1С7.7:   Front = СоздатьОбъект("RRC_back.RRCback");
для 1С8.х:   Front = Новый СОМОбъект("RRC_back.RRCback");
```

При подключении (создании экземпляра COM-объекта) будет открыто соединение с сервером модуля «PriceChecker» в соответствии с сохранёнными ранее настройками источника данных ODBC, настройки доступа к которому хранятся в файле "C:/MINIMARKET.udl". На панели задач появится иконка, сигнализирующая о состоянии соединения.

Для корректного отключения от сервера необходимо просто разрушить объект (например : `delete Front; Set Front = Nothing` `Front = 0;` или `Front = Null;` соответственно) - при разрушении объекта он сам закроет ODBC соединение с сервером фронт-офисного модуля.

## Описание методов COM-объекта:

### AddPluChecker\_( cod, cod1, bar, name, price )

метод добавляет запись про товар на сервер модуля «PriceChecker», а если такая запись на сервере уже существует, то обновляет ее; ключевыми полями для записи про товар являются поля cod и cod1; количество записей про товар для каждого товара соответствует количеству штрихкодов этого товара, если штрихкода нет, то запись одна.

метод **принимает** параметры:

**cod** (число, тип long от 1 до 999999) – код товара

**cod1** (число, тип byte) – порядковый номер записи, если у товара нет штрихкода или всего

один штрихкод: cod1=1, если штрихкодов у товара несколько, то у каждой записи свой cod1= порядковый номер штрихкода, принадлежащего товару.

**bar** (строка, длина 20) – штрихкод товара, если его нет — пустая строка

**name** (строка, длина 50) – наименование товара

**price** (число, тип long) – розничная цена товара \* 100 (в копейках)

**ответ** — (тип boolean) True или False в зависимости от результата

### UpdPrice\_( cod, price )

метод предназначен для обновления цен для существующих товаров на сервере модуля «PriceChecker»

метод **принимает** параметры:

**cod** (число, тип long от 1 до 999999) – код товара

**price** (число, тип long) – розничная цена товара \* 100 (в копейках)

**ответ** — (тип boolean) True или False в зависимости от результата

## DeletePlu\_(cod)

метод удаляет все записи про товар с выбранным кодом на сервере модуля «PriceChecker»

метод **принимает** параметр: **cod** (число, тип long от 1 до 999999) – код товара, который удаляем  
**ответ** — (тип boolean) True или False в зависимости от результата

## DeleteAllPlu\_()

метод удаляет все записи про ВСЕ товары на сервере модуля «PriceChecker». (можно вместо него использовать метод DeletePlu\_(0))

**ответ** — (тип boolean) True или False в зависимости от результата

## РЕКОМЕНДАЦИИ:

В наших конфигурациях в справочнике Товары есть кнопки:

**ДобавитьВсеТовары** для перезаписи всего справочника товаров: сначала удаляем все товары DeleteAllPlu\_, затем через цикл добавляем по-одному с помощью метода AddPluChecker\_

**ДобавитьТовар** для добавления (обновления) текущего товара с помощью метода AddPluChecker\_

**УдалитьТовар** для удаления текущего товара с помощью DeletePlu\_

## Пример использования:

1С7.7: процедура обновления на сервере модуля «PriceChecker» справочника товаров из табличной части документа (метод AddPluChecker\_, на товаре несколько штрихкодов)

```
Процедура ОбновитьНаКассах(Контекст) Экспорт
    Контекст.ВыбратьСтроки();
    Пока Контекст.ПолучитьСтроку() = 1 Цикл
        cod = Контекст.Товар.ВнутрКод;
        name = Контекст.Товар.Наименование;
        price = глПолучитьЦену(Контекст.Товар,Константа.РозничныйСклад.КатегорияЦен,ТекущаяДата())*100;
        ШК = СоздатьОбъект("Справочник.ШтрихКоды");
        ШК.ИспользоватьВладельца(Контекст.Товар);
        ШК.ВыбратьЭлементы();
        ЧислоШтрихКодов = 0;
        Пока ШК.ПолучитьЭлемент() = 1 Цикл // перебираем штрихкоды из подчинённого справочника
            ЧислоШтрихКодов = ЧислоШтрихКодов + 1;
            Front.AddPluChecker_(cod,ЧислоШтрихКодов,ШК.ШтрихКод,name,price);
        КонецЦикла;
        Если ЧислоШтрихКодов = 0 Тогда // если штрихкодов нет, создаём строку с пустым штрихкодом
            Front.AddPluChecker_(cod,1,"",name,price);
        КонецЕсли;
    КонецЦикла;
КонецПроцедуры
```