

# Инструкции по доработке конфигурации

## Возможные способы доработки конфигурации:

1. пользовательские раскладки клавиатуры
2. [пользовательские кнопки в форме супервизор](#)
3. [расширенные отчеты](#)
4. модуль логики
5. изменение конфигурации и типовых внешних обработок

Способ №1 (использование собственных раскладок клавиатуры) рекомендуется использовать когда клиенту нужна функция (кнопка или горячая клавиша), которая выполнит типовые действия, которые отсутствуют в предопределенных раскладках клавиатуры. Например, применить определенную (или из списка) дисконтную карту, товар, продавца и т.п. Т.е. использовать набор функций формы регистрации.

Способ №2 используется когда нужно добавить дополнительную кнопку в супервизор.

Способ №3: Расширенные отчеты используются для вывода какой-то информации на экран или принтер, могут вызываться пользователем в произвольный момент времени и, опционально, автоматически при снятии отчета с гашением

[Способ №4](#) решает большинство задач без необходимости доработок конфигурации и типовых обработок с максимальной совместимостью при обновлении конфигурации.

[API логических модулей](#)

[Список обработчиков в защищенной компоненте](#)

Ниже можно скачать архивы с заготовкой модуля логики. В конце модуля основной формы каждой заготовки есть краткая пошаговая инструкция.

ПустаяЗаготовка

- пустой модуль логики, обработчики для добавления закладки подключатся автоматически при заполнении переменных `ИмяМодуля` и `ЗаголовокЗакладкиНастроек`.

ЗаготовкаМакетЧека

- заготовка модуля логики, который позволяет добавить дополнительные поля в макет чека.

ЗаготовкаОсновныеОбработчики

- заготовка модуля с наиболее часто используемыми обработчиками. Каждый обработчик прокомментирован.

Переопределение активности автоскидок

- модуль логики позволяет программно переопределять активность автоматических скидок.

Вариант №5 крайне не рекомендуется использовать по следующим причинам:

- Изменение базовой версии конфигурации невозможно.
- При каждом обновлении конфигурации потребуются переносить изменения вручную.
- В новой версии конфигурации возможно изменение кода, которое потребует дополнительное время на актуализацию доработок.



```
МенеджерОбъектов.ПолучитьОбщийОбъект("Регистрация.ФормаЗапросКоличества");  
Если ФормаЗапросКоличества = Неопределено Тогда  
    ФормаЗапросКоличества =  
МенеджерОбъектов.СоздатьОбщуюФорму("Регистрация.ФормаЗапросКоличества" ,  
"MAIN" );  
КонецЕсли;  
НеДробноеЧисло = ФормаЗапросКоличества.ПолучитьЧисло(10, 0, 2); // ввод недробного  
двухзначного числа, значение по-умолчанию 10
```

**ВыбратьДату, ВыбратьВремя, ВыбратьДатуИВремя** - функции общей формы ВыборДаты открывает форму для ввода даты/времени/даты и времени, тип возвращаемого значения Булево.

1: Произвольный - в эту переменную запишется дата, которую указал пользователь.

2: Дата (Неопределено) - значение по-умолчанию.

```
УИД_Формы = Новый УникальныйИдентификатор();  
ФормаВыбораДаты = МенеджерОбъектов.СоздатьОбщуюФорму("ОбщаяФорма.ВыборДаты",  
УИД_Формы);  
Результат = Неопределено;  
ДатаРождения = Дата(2000, 1, 1);  
  
Если ФормаВыбораДаты.ВыбратьДату(Результат, ДатаРождения) Тогда  
    ДатаРождения = Результат;  
КонецЕсли;  
  
МенеджерОбъектов.ВыгрузитьОбщийОбъект("ОбщаяФорма.ВыборДаты", УИД_Формы);
```

## Работа с составом чека

Как получить форму регистрации в коде:

```
ФормаРегистрации = МенеджерОбъектов.ПолучитьОбщийОбъект("Регистрация.Форма");
```

Как получить состав чека формы регистрации:

```
ФормаРегистрации = МенеджерОбъектов.ПолучитьОбщийОбъект("Регистрация.Форма");  
КоличествоСтрок = ФормаРегистрации.СоставЧека.Количество();
```

Как получить выделенную строку состава чека:

```
ФормаРегистрации = МенеджерОбъектов.ПолучитьОбщийОбъект("Регистрация.Форма");  
ТекущаяСтрокаСоставаЧека = ФормаРегистрации.элтСоставЧека.ТекущаяСтрока;
```

Как добавить товар в состав чека по коду или артикулу (для добавления по артикулу нужно в 2-й параметр передать «Артикул»):

```
ФормаРегистрации = МенеджерОбъектов.ПолучитьОбщийОбъект("Регистрация.Форма");  
ЗК = МенеджерОбъектов.ПолучитьОбщийОбъект ("AddIn.CashContext" );
```



## Работа с ККМ

Пример печати произвольного текста на ККМ (из формы регистрации, считаем что ККМ подключен):

```
//ф-я КонвертироватьВМассивДляПечати принимает на вход макет чека, можно использовать тэги форматирования
//<CENTER> - выравнивание по центру
//<LEFT> - выравнивание по левому краю (по-умолчанию)
//<RIGHT> - выравнивание по правому краю
//Слева<HR> </HR>Справа - разделить слова пробелами на всю ширину
//<HR>.</HR> - повторить на всю ширину
//<s> - штрихкод ean13
//<s2d> - qr-код
//<CUT> - отрезка
ПС = Символ(182);
Текст = "<CENTER>Строка по центру" + ПС +
"Строка слева" + ПС +
" <RIGHT>Строка справа" + ПС +
" <HR>*</HR>" + ПС +
" <s>1234567890123<s>" + ПС +
" <s2d>http://host<s2d>" + ПС +
" " + ПС +
" " + ПС +
" <CUT>";
ЗК = МенеджерОбъектов.ПолучитьОбщийОбъект ("AddIn.CashContext");
МассивДляПечати = ЗК.КонвертироватьВМассивДляПечати(Текст);
СтрокаККМ = Кассир5_DataAccess.ПолучитьККМ();
РезультатПечати_бул = ЗК.FnККМ_PrintStringBlock(СтрокаККМ, МассивДляПечати);
```

## Работа со справочниками транзакций

В Кассир 5 есть два типа справочников с транзакциями:

1. Справочник Временные транзакции - хранит информацию о текущем чеке, при закрытии (или отмене) чека переносятся в справочник Транзакции и удаляются.
2. Справочник Транзакции - хранит информацию об основных действиях кассира, закрытых или отмененных чеках и пр.

Справочник выгружается в файл отчета по [формату Кассир 5](#).

Элементы удаляется только автоматически по регламенту.

## Работа со справочником Временные транзакции

Перед работой с временными транзакциями рекомендуется ознакомиться со статьей [Таблица временных транзакций кассира](#).

Если нужно записать произвольную транзакцию, которая относится к текущему чеку, тогда нужно записать временную транзакцию.



```
Кассир5_DataAccess .ПолучитьВремТранзакцииПоТипу(ТипТранзакцииДопИнфо) Цикл
    Кассир5_DataAccess .УдалитьВремТранзакцию(Транз.Код);
КонецЦикла;
// запишем транзакцию на чек, код установится автоматически, дата транзакции =
ТекущаяДата(), секция = 1, код товара = "2", цена = 3, количество = 4, сумма = 5, номер ккм
= по-умолчанию
Кассир5_DataAccess .ПисатьВремТранзакцию(0, ТипТранзакцииДопИнфо, ТекущаяДата(), 1,
"2", 3, 4, 5, Кассир5_DataAccess .ИспользоватьККМ, "");
```

См. также в модуле обработки Кассир5\_DataAccess:

**ИзменитьВремТранзакциюПоСтруктуре**

**ИзменитьВремТранзакцию**

**УдалитьВремТранзакцию**

**ПолучитьВыборкуСкидок**

**ПолучитьВремТранзакцииЧека**

## Работа со справочником Транзакции

Для записи транзакции напрямую в справочник Транзакции (транзакции, не относящиеся к конкретному чеку) используется процедура модуля обработки Кассир5\_DataAccess

**ЗаписатьТранзакции.**

Параметры:

1: Таблица значений (неопределено) - таблица значений, получаемая функцией модуля формы регистрации или модуля обработки Кассир5\_DataAccess **ДобавитьТранзакциюВТаблицу.**

Пример записи одной транзакции с типом 777:

```
Кассир5_DataAccess .ЗаписатьТранзакции(Кассир5_DataAccess .ДобавитьТранзакциюВТаблицу(Неопределено,
Неопределено, // код кассира
777, // тип транзакции
Неопределено, // дата транзакции
0, // секция
"", // код товара
0, // цена
0, // количество
0)); // сумма
```

Пример записи нескольких транзакций:

```
ТаблицаТранзакций = Неопределено;
Кассир5_DataAccess .ДобавитьТранзакциюВТаблицу(ТаблицаТранзакций, Неопределено,
777, Неопределено, 0, "", 0, 0, 0);
Кассир5_DataAccess .ДобавитьТранзакциюВТаблицу(ТаблицаТранзакций, Неопределено,
778, Неопределено, 0, "", 0, 0, 0);
Кассир5_DataAccess .ЗаписатьТранзакции(ТаблицаТранзакций);
```

## Работа с правами пользователя

**ПравоДоступно** - функции модуля обработки Кассир5\_DataAccess, выполняет проверку наличия права у пользователя, тип возвращаемого значения булево.

1: Строка - имя права.

2: Булево (Истина) - если у текущего пользователя право отсутствует, но существует пользователь с набором прав, в котором это право присутствует - тогда откроется форма для считывания карты пользователя с этим правом.

3: Булево (Истина) - выводить сообщение об отсутствии права.

4: Число (Неопределено) - если параметр указан, тогда будет проверяться пользователь с кодом, указанным в этом параметре, иначе текущий пользователь.

5: Число (Неопределено) - в этот параметр запишется код пользователя, который подтвердил право.

```
Если НЕ Кассир5_DataAccess.ПравоДоступно("Право", Ложь, Ложь) Тогда  
// действие при отсутствии права без запроса карты и без вывода сообщения  
КонецЕсли;
```

**ПравоОтсутствует** - функции модуля обработки Кассир5\_DataAccess, выполняет проверку отсутствия права у пользователя, тип возвращаемого значения булево.

1: Строка - имя права.

2: Булево (Истина) - если у текущего пользователя право присутствует, но существует пользователь с набором прав, в котором это право отсутствует - тогда откроется форма для считывания карты пользователя без этого права.

3: Булево (Истина) - выводить сообщение при наличии права.

4: Число (Неопределено) - если параметр указан, тогда будет проверяться пользователь с кодом, указанным в этом параметре, иначе текущий пользователь.

```
Если НЕ Кассир5_DataAccess.ПравоДоступно("Право", Ложь, Ложь) Тогда  
// действие при отсутствии права без запроса карты и без вывода сообщения  
КонецЕсли;
```

## Отладка

Большинство действий пользователя (нажатие клавиш, события сканера и ридера магнитных карт) обрабатывается в конфигурации как внешнее событие.

Иногда нужно проверить - правильно ли отработает программа при определенных входящих данных от оборудования, а возможности использовать реальное оборудования нет.

Для этого можно (в модуле логики или назначить на кнопку) эмулировать внешнее событие методом GenerateEvent компоненты AddIn.ShtrihContext:

Параметры:

1: название источника события (для большинства событий не анализируется)

2: название события

3: данные события

Пример:

эмуляция работы со сканером:

```
МенеджерОбъектов.ПолучитьОбщийОбъект("AddIn.ShtrihContext").GenerateEvent("Scanner", "ScannerEvent", "123");
```

