

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.

Вариант №5 крайне не рекомендуется использовать по следующим причинам:

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

Основные функции

При доработке конфигурации рекомендуется использовать типовые методы, которые можно найти в коде конфигурации.

Наиболее часто используемые методы описаны ниже.

Формат описания методов:

Имя метода - описание метода.

Параметры:

Номер параметра: Тип параметра (значение по-умолчанию) - описание параметра.

Пример кода.

Диалоги

ИнтерфОшибка - процедура выводит сообщение пользователю.

Параметры:

- 1: Стока - строка с текстом сообщения
 - 2: Стока («Ошибка») - заголовок формы

ИнтерфОшибка("Текст сообщения", "Внимание!");

ИнтерфВопросДаНет - глобальная функция открывает окно с переданным текстом и двумя кнопками «Да» и «Нет», возвращает нажатую кнопку («Да» или «Нет»).

Параметры:

- 1: Страна - строка с текстом вопроса
 - 2: Страна («Нет») - кнопка по-умолчанию («Да» или «Нет»)

```
Если ИнтерфВопросДаНет("Текст вопроса") = "Да" Тогда
    // обработка ответа на вопрос
КонецЕсли;
```

ИнтерфВопросДаНетОтмена - глобальная функция открывает окно с переданным текстом и кнопками «Да», «Нет», «Отмена», при нажатии кнопки возвращает строку («Да», «Нет», «Отмена»).

Параметры:

- ## 1: Стока - строка с текстом вопроса

```
Если ИнтерфВопросДаНетОтмена("Текст вопроса") = "Да" Тогда
    // обработка ответа на вопрос
КонецЕсли;
```

ПолучитьКоличество, ПолучитьСекцию, ПолучитьСумму и др. - функции формы

Регистрация.ФормаЗапросКоличества открывают окно для ввода числового значения (все возможные функции и их параметры см. в модуле Регистрация.ФормаЗапросКоличества).

```
ФормаЗапросКоличества =
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
```

Выбордаты - функция общей формы ВыборДаты открывает форму для ввода даты.

```
УИД_Формы = Новый УникальныйИдентификатор();
ФормаВыбораДаты = МенеджерОбъектов.СоздатьОбщуюФорму("ОбщаяФорма.ВыборДаты",
УИД_Формы);
Результат = Неопределено;
ДатаВышедшая = Дата(2000, 1, 1);
```

Если Форма Выбира Дату Выбрать Дату/Результат Дата/Рождения) Тогда

ДатаРождения = Результат;
КонецЕсли;

МенеджерОбъектов.ВыгрузитьОбщийОбъект("ОбщаяФорма.ВыборДаты", УИД_Формы);

Работа с составом чека

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 оборудование 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");
```

Эмуляция работы с ридером магнитных карт:

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

