

# Бизнес процессы и задачи.

- [Общая информация](#)
- [Бизнес процессы](#)
- [Создание бизнес процесса](#)

# Общая информация

## Определения

Руководство разработчика, глава 13

### [Пример настройки БП](#)

## Общее взаимодействие

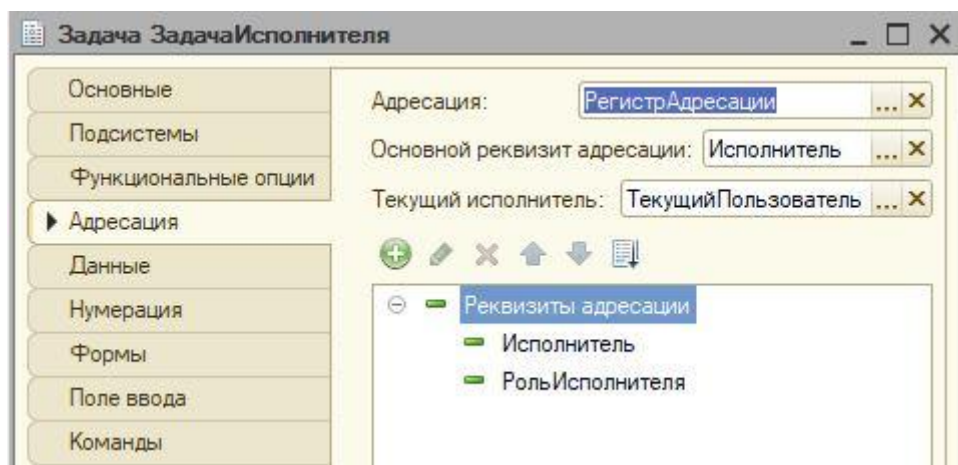
Бизнес процесс берет из задачи правила распределения (Адресацию в терминах 1С). Т е в бизнес процессе мы прописываем последовательность действий. Из объекта Задачи, привязанной к БП, берется алгоритм адресации. К тому же, для каждого шага БП создается одна и та же несколько модифицированная задача. В ней можно отобразить реквизиты БП, но изменять эти реквизиты можно только внутри БП. Взаимодействие БП - Задача довольно мутное, задачу нужно рассматривать как некий шаблон, из которого будет создаваться на каждом шаге конкретная задача.

## Система адресации

Назначение – обеспечить возможность персональной и ролевой адресации задач участникам бизнес-процессов.

Ролевая адресация (ролевая маршрутизация) – набор правил, позволяющий определить конечного адресата (исполнителя) исходя из реквизитов адресации.

Определение конкретного исполнителя (разыменованное) осуществляется с помощью свойств задачи – Адресация, Основной реквизит адресации и Текущий исполнитель.



Адресация – ссылка на непериодический независимый регистр сведений, который хранит актуальную на текущий момент информацию о соответствии исполнителей (основной реквизит адресации) всем остальным реквизитам адресации задач.

Реквизиты адресации - независимый элемент настройки, может не совпадать по названию с элементами регистра адресации. Есть настройка "Тип" (например, СправочникСсылка.Сотрудники). Сопоставление со значениями регистра сведений идет по свойству "Измерение адресации" Реквизита адресации. Т е возможна (и ошибок не выдается) ситуация, когда в регистре адресации измерение ссылается на один справочник, а реквизит адресации, ссылаясь на другой справочник, сопоставляется с этим измерением регистра. И оно вроде будет работать, но не правильно.

Основной реквизит адресации: конкретный сотрудник – исполнитель заданий. Должен ссылаться на реквизит адресации (не на измерение регистра, указанного в разделе Адресация).

Текущий исполнитель - переменная параметра сеанса. Может браться из справочника, например Сотрудники. В предлагаемых алгоритмах работы с заданиями / бизнес процессами корректность связи этой переменной с системным пользователем полностью лежит на разработчике. В ней просто ссылка. Можно установить и конкретное число или строку. Однако нет логической проверки корректности.

**Важно!** Сопоставление между Реквизитами адресации, установленными в Задаче, и измерениями РегистраАдресации настраиваются в свойствах каждого реквизита адресации, в разделе Адресация.

**Последовательность настройки.** Создается параметр сеанса, связанный со справочником сотрудников. При авторизации в параметр сеанса устанавливается ссылка, связанная с созданным нами справочником сотрудников.

**Важно!** Не системным Пользователи, а созданным вручную справочником.

Созданный параметр сеанса устанавливается в Задача - Текущий исполнитель. В нем ссылка на Справочники.Сотрудники.

В Адресации устанавливается регистр, в котором будет храниться информация о текущих сопоставлениях Сотрудник-Роль.

**Эксперимент.** Поскольку было неясно, каким образом сопоставляются значения роли и исполнителя, то к настроенной "по учебнику" схеме (Справочник.Сотрудники, регистр адресации, переменная ТекущийПользователь) был добавлен справочник Пользователи. В Задаче у реквизита адресации Исполнитель был заменен тип со СправочникСсылка.Сотрудники на СправочникСсылка.Пользователи. Распределение задач по пользователям осталось корректным, это повлияло лишь на поле выбора исполнителя, оно бралось из справочника Пользователи.

Номер:	000000009
Наименование:	Ввод приема на работу в с
Дата:	24.01.2025 13:48:23
Бизнес-процесс:	Тестовый процесс 000000004 от 24.01.2025 13:48:11
Точка маршрута:	Ввод приема на работу в статусе проект
Исполнитель задания:	
Роль исполнителя новая:	Старший кадровик

Т е если бы был добавлен алгоритм автоматического заполнения на основе ТекущийПользователь и данное поле было бы скрытым для пользователя, то появилась бы ошибка.

# Бизнес процессы

Бизнес-процесс стартует при вызове метода Старт() или нажатии кнопки "Стартовать и закрыть" в форме объекта. Бизнес-процесс может быть записан, но не стартован.

```
БП = БизнесПроцессы.Согласование.СоздатьБизнесПроцесс();
БП.Дата = ТекущаяДата();
БП.Записать();
БП.Старт();
```

При вызове метода ВыполнитьЗадачу() осуществляется проверка выполнения, после которой задача помечается как выполненная и об этом оповещается бизнес-процесс. Если все необходимые условия выполнены, то бизнес-процесс осуществляет переход на следующую точку маршрута.

Свойство Групповая в Адресации создает задание каждому исполнителю в группе, поэтому, чтобы шаг был выполнен, необходимо выполнение всеми участниками группы.

## Работа с точками процесса.

### События

В свойствах задачи на схеме процесса настраиваются события.

"ПередСозданиемЗадач" – вызывается когда задачи еще не созданы, можно создать новые задачи и полностью заполнить их свойства.

"ПриСозданииЗадач" – задачи уже созданы их можно отредактировать.

### Заполнение бизнес процессом реквизитов задачи

В связи с наличием одной формы для всех задач, может использоваться для детализации формы для пользователя.

```
Процедура ОтсутствиеСвязиССерверомИиЗПриСозданииЗадач(ТочкаМаршрутаБизнесПроцесса,
ФормируемыеЗадачи, Отказ)
□Для каждого Задача Из ФормируемыеЗадачи Цикл
    Задача.ПингДоступен = ПингДоступен;
КонецЦикла;
КонецПроцедуры
```

### Изменение реквизитов бизнес процесса при выполнении задачи

```
Процедура ОтсутствиеСвязиССерверомИиЗПриВыполнении(ТочкаМаршрутаБизнесПроцесса, Задача,
Отказ)
[ЭтотОбъект.ПингДоступен = Истина;
[ЭтотОбъект.Записать();
КонецПроцедуры
```

## Условия и выбор варианта

Для условного ветвления есть обработчик проверки условия, наличие которого обязательно и контролируется при проверке карты маршрута перед сохранением бизнес-процесса. Если обработчик отсутствует, то будет выдано предупреждение: Точка условия не имеет обработчика события "Проверка условия". Этот обработчик должен вернуть результат проверки условия, от которого будет зависеть выбор следующей точки маршрута. Если результат Истина, то бизнес-процесс пойдет по ветке Да, в противном случае – по ветке Нет. По умолчанию результат устанавливается равным значению Ложь.

```
Процедура ОграничениеСкидкиПроверкаУсловия(ТочкаМаршрутаБП, Результат)
Если ПолучитьСкидкуПоСчету() > ПолучитьОбычнуюСкидку() Тогда
    Результат = Истина;
Иначе
    Результат = Ложь;
КонецЕсли;
КонецПроцедуры
```

Для выбора одного из нескольких возможных путей используется точка выбора варианта. Для каждого варианта обязателен обработчик выбора варианта, контролируется при проверке карты маршрута. Если обработчик отсутствует, то будет выдано предупреждение: Точка выбора варианта не имеет обработчика события Выбор варианта. Этот обработчик должен установить параметр Результат равным одному из предусмотренных вариантов.

```
Процедура ВыборВарианта (ТочкаВыбораВарианта, Результат)
Если ВидОплаты = Перечисления.ВидОплаты.Наличная Тогда
    Результат = ТочкаВыбораВарианта.Варианты.Наличная;
ИначеЕсли ВидОплаты = Перечисления.ВидОплаты.Безналичная Тогда
    Результат = ТочкаВыбораВарианта.Варианты.Безналичная;
ИначеЕсли ВидОплаты = Перечисления.ВидОплаты.Взаимозачет Тогда
    Результат = ТочкаВыбораВарианта.Варианты.Взаимозачет;
ИначеЕсли ВидОплаты = Перечисления.ВидОплаты.Кредит Тогда
    Результат = ТочкаВыбораВарианта.Варианты.Кредит;
КонецЕсли;
КонецПроцедуры
//ВидОплаты – реквизит бизнес-процесса.
```

## Шаблоны кода

### Программный старт БП из формы нового БП с дополнительным изменением параметров

```
&НаСервере  
Процедура СтартоватьБПНаСервере()  
□ТекОбъект = Объект.Ссылка.ПолучитьОбъект();  
□ТекОбъект.Старт();  
КонецПроцедуры  
  
&НаКлиенте  
Процедура СтартоватьБП(Команда)  
□Если Не ЗначениеЗаполнено(Объект.Дата) Тогда  
□□Объект.Дата = ТекущаяДата();  
□КонецЕсли;  
□Объект.СервисДоступен = Ложь;  
□ЭтаФорма.Записать();  
□СтартоватьБПНаСервере();  
КонецПроцедуры
```

### Проверка и выполнение задачи.

Например, если задача предусматривает проведение документа, то автоматическая процедура слежения за такими задачами может определять, что нужный документ уже проведен, и помечать задачу как выполненную путем вызова у нее метода Выполнить(). Для организации такого рода автоматизированных процедур предназначен метод ПроверкаВыполнения() у задачи и соответствующие ему обработчики у точек маршрута.

```
Если Задача.ПроверитьВыполнение() Тогда  
    Задача.ВыполнитьЗадачу();  
КонецЕсли
```

### Получить ссылки на процессы

```
СписокБП = БизнесПроцессы.ТестовыйПроцесс.Выбрать();  
Пока СписокБП.Следующий() Цикл  
□Если Не СписокБП.Завершен Тогда  
□□Сообщить(СписокБП.Номер);  
□КонецЕсли;  
КонецЦикла;
```

## Получить имена всех бизнес процессов

```
&НаСервереБезКонтекста
Процедура ВывестиВыполненныеПроцессыНаСервере()
    // Получаем коллекцию всех объектов типа БизнесПроцессы
    КоллекцияБизнесПроцессов = Метаданные.БизнесПроцессы;
    Для Каждого БизнесПроцесс Из КоллекцияБизнесПроцессов Цикл
        //БизнесПроцесс.Имя; БизнесПроцесс.Синхронизация; БизнесПроцесс.Комментарий;
        □□Сообщить(БизнесПроцесс.Имя);
        КонецЦикла;;
    КонецПроцедуры

//В предыдущем скрипте можно заменить имя для просмотра в цикле
//ИмяШаблонаБП = БизнесПроцесс.Имя;
//СписокБП = БизнесПроцессы[ИмяШаблонаБП].Выбрать();
```

## Получение текущей задачи для бизнес процесса

```
ВЫБРАТЬ
    Задача.Ссылка
ИЗ
    Задача.Задача КАК Задача
ГДЕ
    (НЕ Задача.Выполнена)
    И Задача.ГлавныйБП = &ГлавныйБП
    И Задача.ГлавныйБП.Стартован
```

## Получение всех точек бизнес процесса:

```
&НаСервереБезКонтекста
Процедура СписокТочекПроцессаНаСервере()
    □Запрос = Новый Запрос;
    □Запрос.Текст = "ВЫБРАТЬ
        |□□ПроцессМониторингаИиЗТочки.Ссылка КАК Ссылка,
        |□□ПроцессМониторингаИиЗТочки.Порядок КАК Порядок
        |ИЗ
        |□□БизнесПроцесс.ПроцессМониторингаИиЗ.Точки КАК ПроцессМониторингаИиЗТочки";
    □РезультатЗапроса = Запрос.Выполнить();
    □ТЗ = РезультатЗапроса.Выгрузить();
    □Для Каждого СтрокаТЗ из ТЗ Цикл
```



```
□□Сообщить(Строка(СтрокаТЗ[1]) + ": " + Строка(СтрокаТЗ[0]));
```

```
□КонецЦикла
```

```
КонецПроцедуры
```

## Получение следующих точек для конкретной точки

```
&НаСервереБезКонтекста
```

```
Функция ПолучитьИсходящиеТочки(пТочкаМаршрута)
```

```
    Запрос = Новый Запрос;
```

```
    Запрос.Текст =
```

```
        "ВЫБРАТЬ
```

```
        |   ПроцессМониторингаИиЗТочки.Ссылка
```

```
        |ИЗ
```

```
        |   БизнесПроцесс.ПроцессМониторингаИиЗ.Точки КАК ПроцессМониторингаИиЗТочки";
```

```
    РезультатЗапроса = Запрос.Выполнить();
```

```
    ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
```

```
    лРезультат = Новый Массив;
```

```
    Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
```

```
        лВходящиеТочки = ВыборкаДетальныеЗаписи.Ссылка.ПолучитьВходящиеТочки();
```

```
        Если лВходящиеТочки.Найти(пТочкаМаршрута) <> Неопределено Тогда
```

```
            лРезультат.Добавить(ВыборкаДетальныеЗаписи.Ссылка);
```

```
        КонецЕсли;
```

```
    КонецЦикла;
```

```
    Возврат лРезультат;
```

```
КонецФункции
```



# Создание бизнес процесса

**Начало:** База пустая.

**Шаг 1.** Создание системной роли и системного пользователя. Должен быть хотя бы один пользователь с правами администратора. Сейчас должна быть доступна авторизация пользователя с правами администратора.

**Шаг 2.** Настройка хранения ссылки на профиль пользователя в параметре сеанса. [Детали о пользователях и ролях](#)

Создать справочник Сотрудники, он будет содержать профиль пользователя. Для начала без реквизитов, по необходимости можно будет добавить.

Нужно создать параметр сеанса Текущий пользователь, тип - СправочникСсылка.Сотрудники. В модуле сеанса создается процедура, сохраняющая в этот параметр сеанса ссылку на нужную запись в справочнике.

```
Процедура УстановкаПараметровСеанса(ТребуемыеПараметры)
□ИмяПольз = ИмяПользователя();
□ТекПользователь = Справочники.Сотрудники.НайтиПоНаименованию(ИмяПольз, Истина);
□Если Не ЗначениеЗаполнено(ТекПользователь) Тогда
    □НовыйПользователь = Справочники.Сотрудники.СоздатьЭлемент();
    □НовыйПользователь.Наименование = ИмяПольз;
    □НовыйПользователь.Код = ИмяПольз;
    □НовыйПользователь.Записать();
    □ТекПользователь = НовыйПользователь.Ссылка;
□КонецЕсли;
□ПараметрыСеанса.ТекущийПользователь = ТекПользователь;
КонецПроцедуры
```

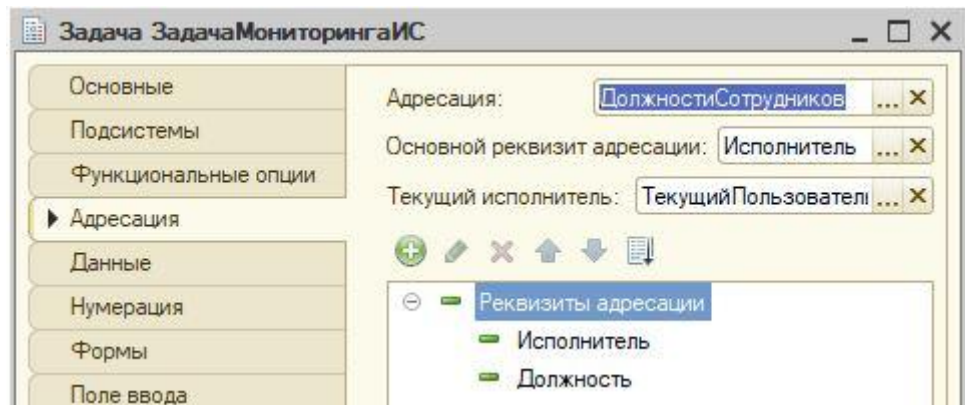
Сейчас после входа в систему с пустым справочником Сотрудники, в нем должен создаваться пользователь. Для проверки можно сделать обработку и кнопку "Вывести", которая отобразит запись из справочника о текущем пользователе.

**Шаг 3.** Создание справочника (структурные подразделения) и регистра сведений (Должности сотрудников). Нужно для организации модели распределения заданий бизнес процесса по должностям, а не по именам. Структурные подразделения - справочник с иерархией групп, группы - названия структурных подразделений, элементы - должности в структурных подразделениях. Структурные подразделения и должности должны быть

предопределенными элементами. В регистре сведений два измерения: Должность и Сотрудник.

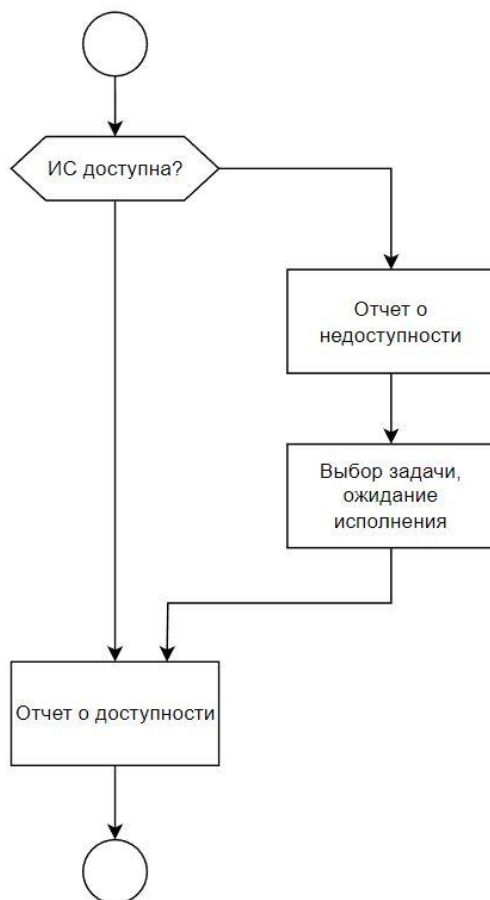
Сейчас можно добавить структурные подразделения, сотрудников и их должности.

**Шаг 4.** Создание задачи (напомню: из задачи в бизнес процесс берется алгоритм адресации).



Исполнитель и Должность должен браться из соответствующих справочников.

**Шаг 5.** Настраиваем бизнес процесс следующего вида:



В данных бизнес процесса создаем реквизит СервисДоступен, булево, по умолчанию истина.

Создаем форму бизнес процесса и выполняем следующее:

- добавляем реквизит СервисДоступен на форму, добавляем обработчик

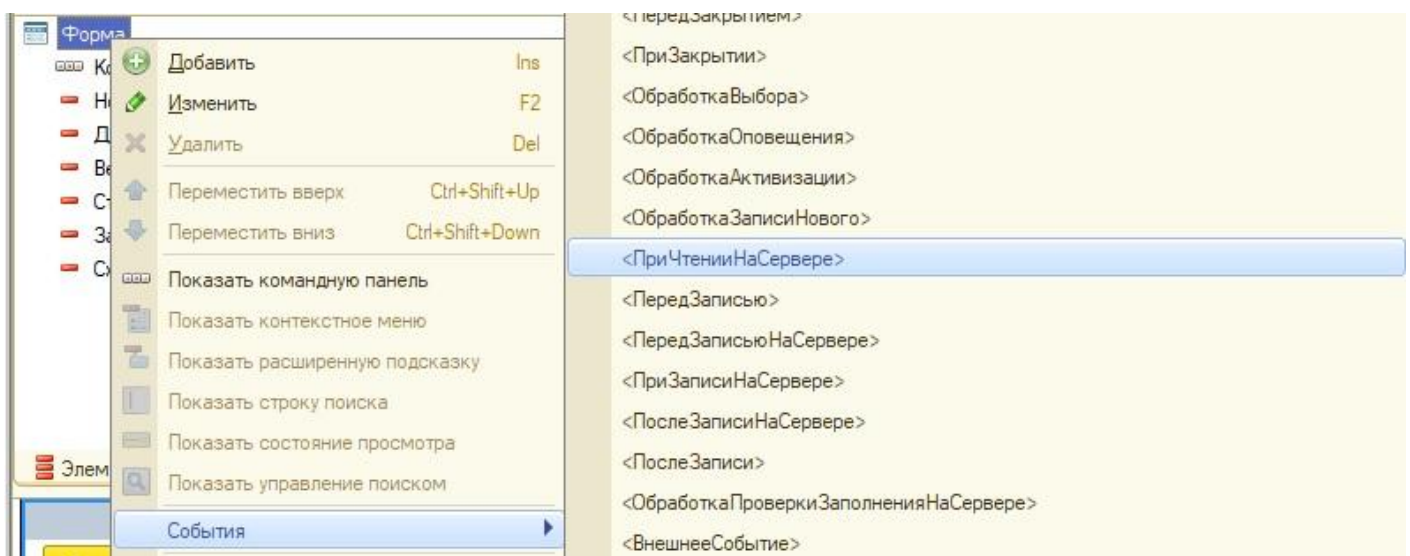
&НаКлиенте

Процедура СервисДоступенПриИзменении(Элемент)

□Элементы..Видимость = Истина;

КонецПроцедуры

- добавляем реквизит Графическая схема. Чтобы на форме отображалось текущее состояние процесса, создаем обработчик события ПриЧтенииНаСервере



В модуле формы нужно создать следующие процедуры:

&НаСервере

Процедура ОбновитьКартуМаршрута()

ОбъектБП = РеквизитФормыВЗначение("Объект");

СхемаПроцесса = ОбъектБП.ПолучитьКартуМаршрута(); //СхемаПроцесса - реквизит графическая  
схема

КонецПроцедуры

&НаСервере

Процедура ПриЧтенииНаСервере(ТекущийОбъект)

□ОбновитьКартуМаршрута();

КонецПроцедуры

