

Формы интерфейса

Здесь и далее под формой понимается управляемая форма (часть кода на сервере, часть на клиенте).

Виды форм интерфейса и взаимосвязь параметров и Основного параметра реквизитов

Существуют формы объекта и произвольные формы.

Форма объекта - форма с заранее установленным реквизитом в соответствии с типом объекта (например в форме элемента это называется Объект и содержит реквизиты соответствующего элемента, в форме списка это Список).

В случае наличия возвращаемого значения (в форме выбора), возврат создается автоматически.

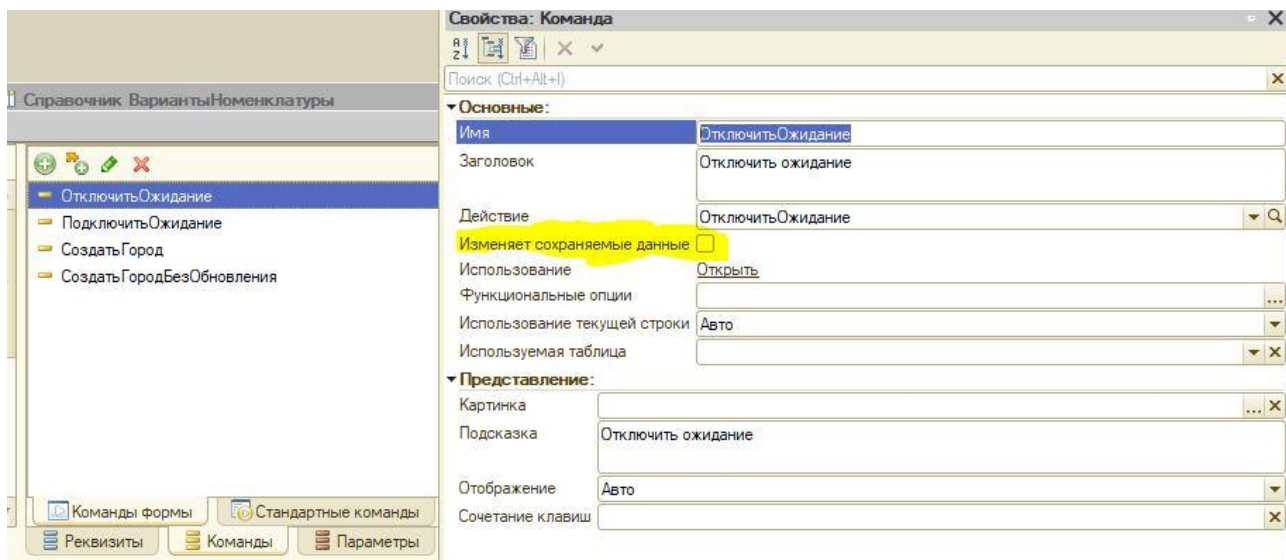
Поддерживается полная программная настройка жизненного цикла формы (Обработка входных параметров - Начальное заполнение - Обработка событий - Возврат выходных параметров).

Термин "заранее установленный реквизит" соответствует реквизиту формы с установленной опцией "Основной параметр".

Для структуры Параметры при программном открытии формы, ключ "Ключ" ссылается на основной параметр. Внутри формы он может называться как угодно, связь происходит по свойству "Основной параметр". В произвольной форме признак основной параметр можно установить, если в типе реквизита указать ...Объект, например СправочникОбъект.

[Статья о формах](#)

Есть критичная особенность обработки данных, связанная с [механизмом блокировок](#) и основным параметром. При установке в свойствах команды флага Изменяет сохраненные данные,



при попытке выполнения команды будут выполняться следующие действия:

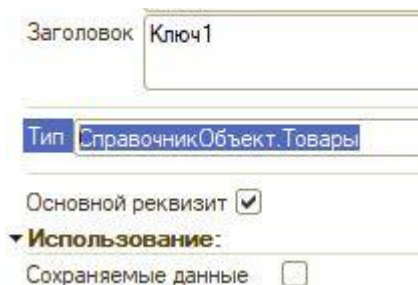
- Попытка заблокировать основной параметр. При неудаче команда не будет выполнена.
- Будет установлен признак изменения у формы (флаг «Модифицированность»).
- Для нового и несохранённого объекта будет произведена попытка записи. При этом пользователю будет задан вопрос о необходимости записи. Если ответ будет отрицательный — команда не будет выполнена.

Отсутствие этого флага не влияет на возможности модификации, но забота о целостности и непротиворечивости данных ложится на плечи разработчика.

Эксперимент. Бессмысленный с практической стороны, только для демонстрации взаимосвязи ключа "Ключ" и основного параметра формы. Справочник Города. В форму элемента добавлена кнопка и реквизит ссылка на товары. Обработка формы:

```
&НаКлиенте
Процедура КомандаОткрытияТоваров(Команда)
□Парам = Новый Структура;
□Парам.Вставить("Ключ", НужныйТоварСсылка);
□ОткрытьФорму("Справочник.Города.Форма.ФормаПростоТак", Парам);
КонецПроцедуры
```

В ФормеПростоТак такая настройка:



Заголовок Ключ1

Тип СправочникОбъект.Товары

Основной реквизит ☒

▼ **Использование:**

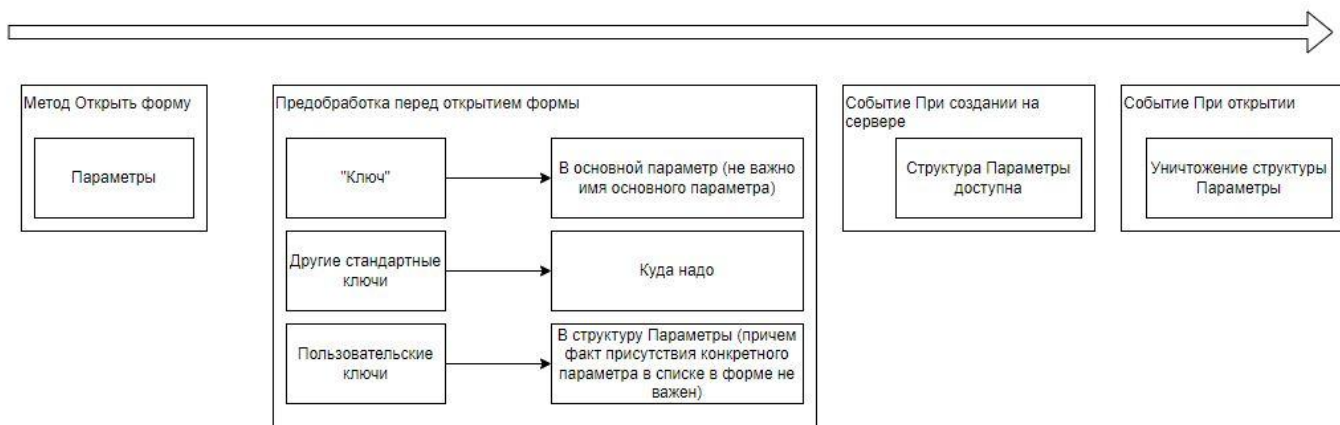
Сохраняемые данные ☐

Имена параметров не совпадают, но когда он основной реквизит - выбранный товар открывается. Похоже, в этом случае ссылка автоматически преобразуется в объект. Если не основной - не открывается. А далее было получено парадоксальное (на момент написания этой строки) поведение: если реквизит основной, но в обработке формы меняю имя параметра на совпадающее название (Ключ1) - данные не подгружаются. После раскуривания умной книжки опять пришлось выдохнуть, улыбнуться, понять и простить разработчиков за использование одного и того же слова в разных контекстах и скрытое преобразование данных:

Взаимосвязь между структурой Параметры в методе Открыть форму (см. раздел Открытие другой формы), реквизитами и параметрами формы. Перед созданием формы, структура Параметры в методе Открыть форму сопоставляется с параметрами в разделе Параметры в форме и стандартными ключами.

При обработке события "При создании на сервере" доступна структура Параметры, в которой будут находиться все переданные параметры. Факт наличия конкретного параметра в списке параметров формы не важен (понять и простить). Указание в списке параметров влияет только на отображение этого параметра в списке контекстной подсказки, но при указании несуществующего параметра ошибки не происходит, и если этот параметр был передан - он считается.

Параметр "Ключ" передается в реквизит со свойством "Основной параметр" напрямую и это свойство может быть выставлено только у одного реквизита. Если в обработке события "При создании на сервере" не сделать заполнение реквизитов (или каких-либо других действий), все пользовательские параметры будут проигнорированы и затем структура Параметры очищается.



&НаСервере

Процедура СообщитьИнфу()

□Сообщить(Параметры.Реквизит3);

КонецПроцедуры

&НаКлиенте

Процедура Команда1(Команда)

□СообщитьИнфу();

КонецПроцедуры

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

□Реквизит1 = Параметры.Реквизит3;

КонецПроцедуры

Команда1 привязана к кнопке на форме. В параметрах формы - Реквизит2. Реквизит3 в параметрах формы отсутствует. В реквизитах формы - Реквизит1. Следующий код корректно открывает вышеуказанную форму и заполняет Реквизит1 значением 5:

&НаКлиенте

Процедура КомандаОткрытияФирм(Команда)

□Парам = Новый Структура;

□Парам.Вставить("Реквизит3", 5);

□ОткрытьФорму("Справочник.Города.Форма.ФормаПростоТак", Парам);

КонецПроцедуры

При выполнении процедуры СообщитьИнфу выдается ошибка отсутствия поля Реквизит3. То есть структура Параметры после создания формы остается, но пустая. Если перенести Сообщить(Параметры.Реквизит3); в Команда1, то ошибка такая же.

Передача реквизита формы в серверную процедуру

Описание процедур

Необходимость в данной технологии возникает, когда есть данные на форме, и нужно провести с ними манипуляции. Т е есть еще не записанные в базу данные и есть модули (общие, ...) и нужно передать туда данные без записи, модифицировать и отдать обратно в форму.

РеквизитФормыВЗначение() компилируется &НаСервере, контекст формы необходим. На клиенте не работает, поскольку в результате получаем прикладной объект.

ЗначениеВРеквизитФормы() для возврата данных на форму. Вторым параметром является тип значения, необязательный. Если обрабатываемый реквизит не является составным типом, то тип будет автоматически получен из реквизита формы. Иначе генерируется исключение времени выполнения.

&НаСервере

Процедура ЗаполнитьТЧПоДаннымПоследнегоДокумента(Контрагент)

ДокОбъект = РеквизитФормыВЗначение("Объект");

ДокОбъект.ЗаполнитьТЧПоДаннымПоследнегоДокумента(Контрагент);

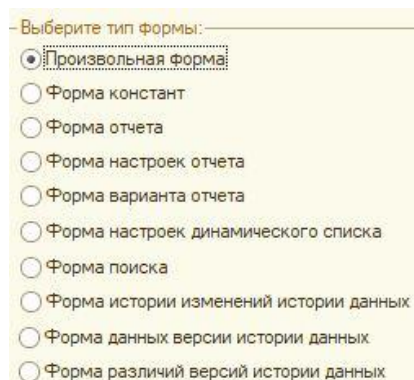
ЗначениеВРеквизитФормы(ДокОбъект, "Объект");

КонецПроцедуры

Точки хранения форм

Формы могут храниться в Общие - Общие формы и в объектах конфигурации, раздел Формы. Произвольную форму можно создать во всех точках хранения.

В общих формах можно создать формы констант, отчетов, динамических списков, поиска, истории данных.



Выберите тип формы:

- ☒ Произвольная форма
- ☐ Форма констант
- ☐ Форма отчета
- ☐ Форма настроек отчета
- ☐ Форма варианта отчета
- ☐ Форма настроек динамического списка
- ☐ Форма поиска
- ☐ Форма истории изменений истории данных
- ☐ Форма данных версии истории данных
- ☐ Форма различий версий истории данных

В объектах конфигурации - набор форм в соответствии с типом объекта. Создание произвольной формы в объекте конфигурации технически возможно, актуально если логически данная форма выполняет действия, связанные с этим объектом. Должны быть аргументы для размещения произвольной формы в объекте, в общем случае логичнее произвольную форму разместить в общих формах. Плюс размещения произвольной формы в

объекте - возможность выбора этой формы в качестве формы чего-либо. Однако, стандартная форма ничем не хуже произвольной, кроме уже настроенного объекта и привязанного к нему свойству Основной реквизит.

Модальные и обычные окна

Модальные окна в 1С - это такой тип окна, вызываемый в среде 1С, который блокирует остальной интерфейс 1С 8. Сейчас устаревшая технология.

Способы взаимодействия между формами

[Посмотреть интерактивное взаимодействие и оповещения](#)

1. ОткрытьМодально, параметр закрытия, метод Закреть. - устаревший способ, не использовать.

```
//В первой форме
ВозвращаемоеЗначение = ДругаяФорма.ОткрытьМодально().
//В другой форме
ЭтаФорма.Закреть(ВозвращаемоеЗначениеВПервуюФорму)
```

2. Реквизиты формы (или свойства расширения формы). Реквизиты формы видимы снаружи как свойства объекта «Форма».

```
//В первой форме
ДругаяФорма.ИмяРеквизита = Значение.
```

3. Через ЭлементыФормы.

```
//В первой форме
ДругаяФорма = ДокументСсылка.ПолучитьФорму().
ДругаяФорма.ЭлементыФормы.ИмяПоля.Значение = Значение.
```

4. Оповещения.

```
//В первой форме оповещаются все открытые формы
Оповестить("ИмяСобытия", "Форма1", Данные);

//В другой форме следует назначить обработчик события оповещения.
Процедура ОбработкаОповещения(Событие, Источник, Данные)
    Если Событие = "ИмяСобытия" И Источник = "Форма1" Тогда
        КонечЕсли;
    КонечПроцедуры
```

5. Экспортные переменные модулей форм. Они становятся видимы снаружи примерно как реквизиты форм.

```
//В первой форме
ДругаяФорма.ИмяПеременной = Значение;
//В другой форме
Перем ИмяПеременной Экспорт;
```

6. Параметры сеанса.

```
//В первой форме
ПараметрыСеанса.ИмяПараметра = Значение
//В другой форме
Если ПараметрыСеанса.ИмяПараметра = Значение Тогда КонецЕсли
```

Работа с элементами формы

Доступ к выбранному элементу табличной части. Контактные лица - имя табличной части.

```
&НаКлиенте
Процедура СообщитьВыбраннуюСтроку(Команда)
□ВыбранныйКонтакт = Элементы.КонтактныеЛица.ТекущиеДанные;
□Если ВыбранныйКонтакт = Неопределено Тогда
□□Сообщить("Контактов не существует");
□Иначе
□□Сообщить(ВыбранныйКонтакт.Телефон);
□КонецЕсли;
КонецПроцедуры
```

Проверка заполнения поля. Важный момент, потратил минут 30. ! Код нужно тестировать !

```
ЗначениеЗаполнено(Поле)
```

Открытие формы.

[Статья по открытию форм](#), но стоит проверить данные в статье.

Существуют объекты ссылочного типа (Справочники, элементы справочника и т д) и нессылочного, например элементы регистра. Естественно, если в форме присутствует ссылка на объект, их можно передать в создаваемую форму в качестве параметра.

Способы открыть форму:

ОткрытьФорму	ПолучитьФорму	ПоказатьЗначение
Основной универсальный способ. Сразу открывает форму.	Получаем форму без открытия, затем метод Открыть. Как ОткрытьФорму.	Типа упрощение, негибкое

Метод ОткрытьФорму. Универсальный способ. Полный список параметров метода «ОткрытьФорму»: ИмяФормы, Параметры, Владелец, Уникальность, Окно

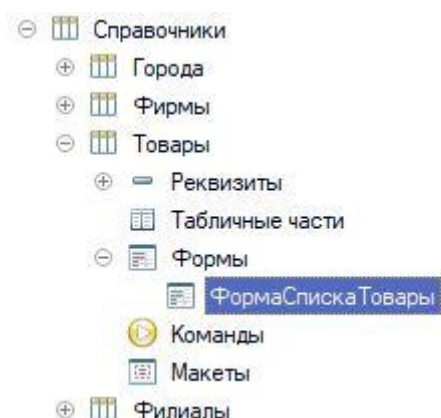
Свойство ИмяФормы. Пример открытия формы с передачей только имени формы. Параметром передается строка, подсказки нет, поэтому внимательно к именам (Справочник в единственном числе, Форма в единственном числе).

```
ОткрытьФорму("Справочник.Товары.ФормаЭлемента");//Основная форма
ОткрытьФорму("Справочник.Товары.Форма.ПроизвольнаяФормаТовара");// + слово Форма для
неосновной формы
```

Нюансы: формы константы нет, нужно сначала создать общую форму констант.

Пример. Кнопка, к которой привязано действие, размещена в форме элемента справочника Города.

```
&НаКлиенте
Процедура КомандаОткрытияТоваров(Команда)
□ОткрытьФорму("Справочник.Товары.Форма.ФормаСпискаТовары");
КонецПроцедуры
```



Свойство «Параметры» Тип Структура. Передать туда можно все, что можно передать с клиента на сервер.

Если объект ссылочный (например конкретный документ, на который есть ссылка), то необходимо в поле Ключ передать ссылку.

Пример: есть справочник, в котором есть табличная часть КонтактныеЛица. У табличной части есть реквизит ДокументТестовый типа ДокументСсылкаПоступлениеТоваров. Процедура обработки нажатия на кнопку:

```
&НаКлиенте
Процедура ОткрытьДокумент(Команда)
□ВыбранныйКонтакт = Элементы.КонтактныеЛица.ТекущиеДанные;
□Если ВыбранныйКонтакт = Неопределено Тогда
□□Сообщить("Контактов не существует");
□Иначе
□□Док = ВыбранныйКонтакт.ДокументТестовый;
□□Если ЗначениеЗаполнено(Док) Тогда
□□□Парам = Новый Структура;
□□□Парам.Вставить("Ключ", Док);
□□□ОткрытьФорму("Документ.ПоступлениеТоваров.ФормаОбъекта", Парам);
□□Иначе
□□□Сообщить("Документ не привязан.");
□□КонецЕсли;
□КонецЕсли;
КонецПроцедуры
```

При передаче пустого ключа создастся новый документ.

В случае, если нужно создать новый объект на основании существующего, то нужно добавить параметр "Основание" со ссылкой на объект, на основании которого нужно сделать новый документ. В этом случае вызывается процедура «ОбработкаЗаполнения» модуля объекта. Эта процедура имеет единственный параметр «Основание». Если мы в параметры вставим ключ «Основание», то он будет передан в процедуру «ОбработкаЗаполнения».

Если нужно открыть форму создания группы, то в параметрах указать "ЭтоГруппа" Истина.

События формы

[Интересная статья](#)

События при создании формы (сверху вниз)

Модуль формы клиент	Модуль формы сервер	Модуль объекта сервер
	При чтении на сервере (существующий объект)	Обработка заполнения (новый объект)
	При создании на сервере	
При открытии		

Прикладные задачи

Передача параметров в произвольный запрос динамического списка. При открытии формы передаются конкретные значения параметров, в обработчике события формы При создании на сервере эти значения устанавливаются параметрами запроса в динамическом списке.