

Простой тест с помощью механизма «Сценарное тестирование».

Статья не в коей мере не претендует на истину в последней инстанции и имеет целью только рассказать о знакомстве автора со сценарным тестированием.

**Для начала немного теории.**

Сценарного тестирования – инструмент, входящий в состав КИП. Общую информацию можно получить здесь - <http://1c.ru/news/info.jsp?id=17985>, а на ИТС содержится довольно подробное руководство.

Сценарное тестирование версии 3.0.8.2, с которым экспериментировали, использует механизм автоматизированного тестирования платформы (почитать о нем можно здесь <http://v8.1c.ru/overview/Term_000000816.htm> и вот здесь <http://infostart.ru/public/262904/>).

Те сценарии, что можно найти как примеры использования Сценарного тестирования, обычно работают с эталонными данными – то есть предполагается, что предварительно в эталонной базе оформляем справочники/документы так, как должно быть. После, в тестируемой базе, полностью копируем эталонные документы и сравниваем результаты – реквизиты, движения, результаты отчетов и прочее. Такой подход, конечно, полезен, но есть ситуации, когда нужно несколько другое. К примеру, на форму добавлены свои обработчики заполнения ТЧ с помощью внешних обработок, в расширении конфигурации у формы добавлены новые команды или создана новая роль с ограниченными правами. В этом случае, после каждого обновления релиза конфигурации необходимо руками нажимать каждую добавленную\измененную команду на форме для проверки ее работоспособности. В случае с добавлением нетиповой роли, приходится каждый раз заходить под пользователем с ограниченными правами и воспроизводить его типовые действия при работе в 1С, например, создание нового контрагента, оформление заказа, его оплата и последующая отгрузка и т.д.

**Простой сценарий создания контрагентов.**

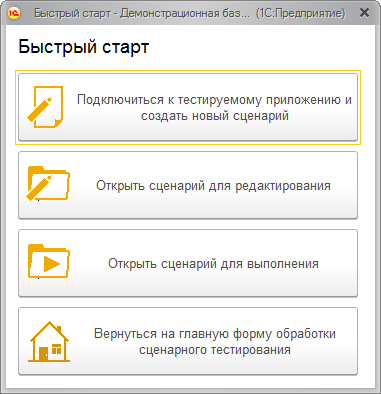
Для примера работы со сценарным тестированием разработаем простой сценарий проверки создания контрагентов ба демо-базе Управление торговлей 11.2.3.163. Далее этот сценарий можно расширять, добавляя блоки создания заказов клиента и их обработки, проверяя какие-то механизмы передачи созданных контрагентов в другие системы.

Для разработки нового сценария понадобится обработка «Сценарное тестирование». Запускать обработку нужно в пользовательском режиме 1С:Предприятие, запущенном как менеджер тестирования.

Запустить 1С как менеджера тестирования можно:

* Указав в настройках базы в списке баз параметр запуска «/TESTMANAGER»
* Через конфигуратор –в окне «Сервис» - «Параметры» - закладка «Запуск 1С:Предприятия» - закладка «Дополнительно» - установить параметр «Автоматизированное тестирование» в значение «Запускать как менеджер тестирования».

Запускаем обработку Сценарное тестирование. При запуске сразу предлагается выбрать один из вариантов:



Выбираем «Подключиться к тестируемому приложению и создать новый сценарий».

*Если нужно открыть уже существующий сценарий для редактирования, лучше сразу запустить клиент тестирования. Так будет возможно, во-первых, заполнять новые шаги данными из клиента тестирования, во-вторых, сразу выполнять шаги и наблюдать результат.*

Небольшое отступление:

В сценарии есть 2 вида шагов – групповые шаги и обычные шаги сценария.

Групповые шаги могут служить для объединения нескольких действий с одним объектом, а также для условного или циклического выполнения действий. Примеры групповых шагов – Условие, Цикл, Форма (внутри нее должны располагаться все шаги, связанные с действиями над конкретной формой), Таблица формы.

Обычные шаги отвечают за какое-то конкретное действие – нажать кнопку, заполнить поле, перейти по ссылке и прочие.

В зависимости от того, находясь в каком месте вы добавляете шаг, могут быть доступны разные шаги. Например, находясь в группе формы, можно добавить групповой шаг работы с таблицей формы.

И вернемся к написанию нашего сценария. Требования по тестированию контрагентов:

* необходимо проверить создание разных контрагентов – юр./физ. лицо, с/без электронной почты;
* у каждого контрагента должно быть уникальное наименование.

Как можно решить задачу:

* Можно создать набор шагов по созданию контрагента и скопировать его 4 раза, заменяя заполнение реквизитов контрагента.
* Можно создать макрошаг, который будет включать набор шагов по созданию контрагента. И создать 4 обращения к макрошагу, передавая в него разные параметры.
* Можно создать цикл, в зависимости от номера итератора цикла, заполнять контрагента разными данными.

Воспользуемся вторым вариантом, так как (субъективно) для понимания и дальнейшего сопровождения сценария это наиболее удобный вариант.

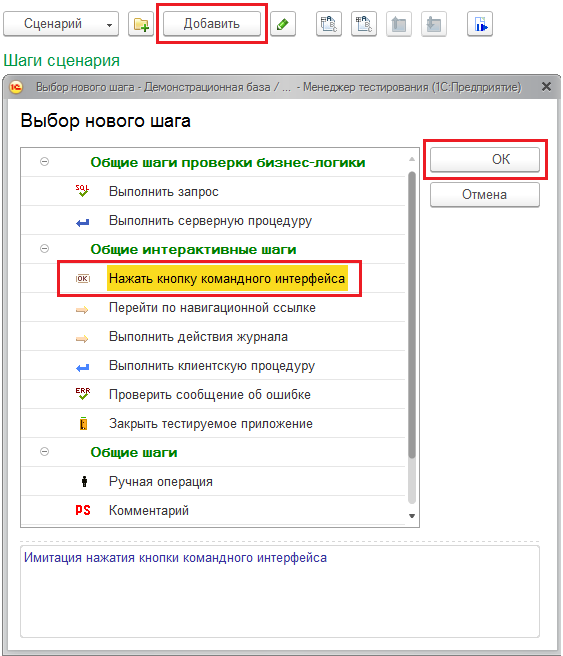
В результате должна получиться подобная структура сценария:

* Определяем параметры заполнения контрагента физ. лицо, с email.
* Создаем контрагента с указанными параметрами.
* Определяем параметры заполнения контрагента физ. лицо, без email.
* Создаем контрагента с указанными параметрами.
* Определяем параметры заполнения контрагента юр. лицо, с email.
* Создаем контрагента с указанными параметрами.
* Определяем параметры заполнения контрагента юр. лицо, без email.
* Создаем контрагента с указанными параметрами.

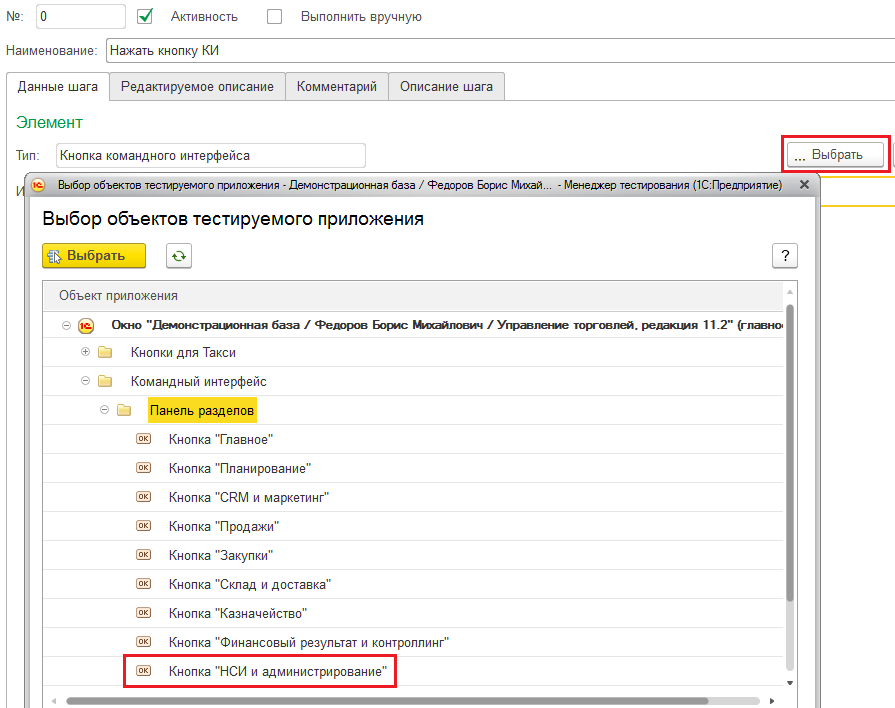
Сначала создадим набор шагов для создания первого контрагента – физ.лицо с заполненным email (на его основании после создадим остальных контрагентов).

**Первый шаг.** Нам необходимо открыть раздел «НСИ и Администрирование», выбрать пункт «Партнеры».

Добавляем шаг сценария «Нажать кнопку командного интерфейса»:



В форме шага нужно указать нажимаемую кнопку. Чтобы не заполнять имя вручную, можно нажать кнопку «…Выбрать» и в появившемся окне выбрать нужную кнопку:



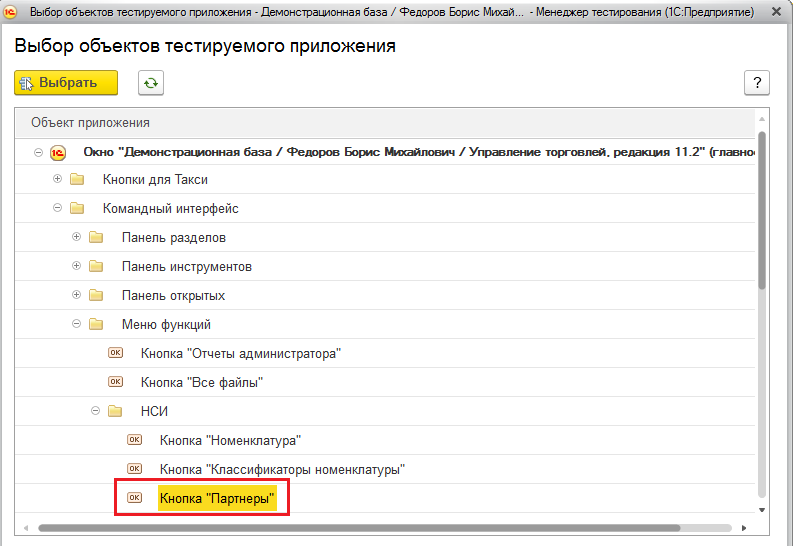
*Данное окно появляется только если менеджер тестирования подключен к тестируемому приложению. Если окно не появляется – воспользуйтесь кнопками в основном окне сценария: «Сценарий» - «Запустить тестируемое приложение» и «Завершить работу тестируемого приложения».*

При желании можно поменять наименование шага. Сохраняем шаг.

Теперь шаг сценария выполним в тестируемом приложении. После выполнения шага мы получаем доступ к результатам выполнения (т.е. тем кнопкам и формам, который были открыты\использованы в тестируемом приложении) и можем использовать их в дальнейших шагах тестирования. Выполняем шаг по кнопке F11 или кнопкой на панели:



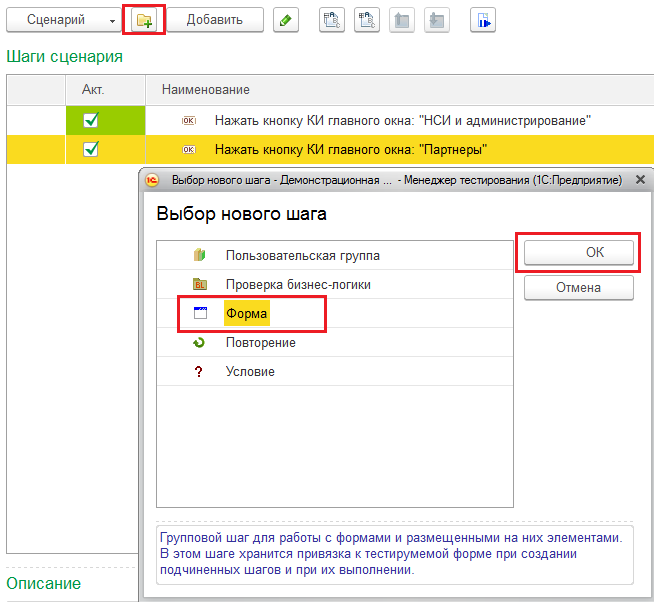
**Второй шаг.** Создаем новый шаг «Нажать кнопку командного интерфейса», по кнопке «…Выбрать» в окне шага теперь доступны новые кнопки, которые видны в тестируемом приложении:



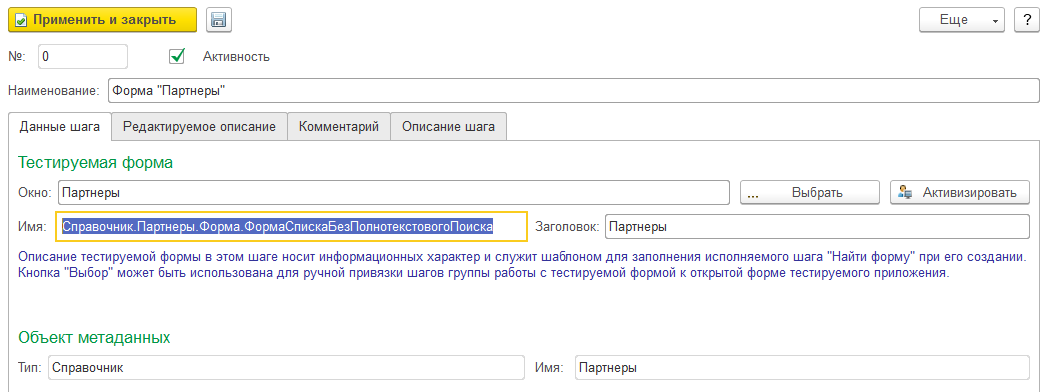
Выбираем кнопку Партнеры и сохраняем шаг. Выполним его, чтобы в тестируемом приложении открылось окно «Партнеры». *Если окно будет открыто не через выполнение шага в сценарии, а, например, вручную, с ним невозможно будет работать из сценария.*

**Третий шаг.** Дальше работать будем с окном формы списка партнеров, для этого в сценарий нужно добавить групповой шаг «Форма», внутри которого будут шаги работы с этой формой.

Добавляем групповой шаг «Форма»:

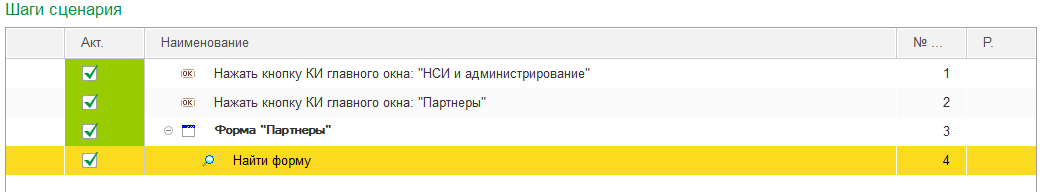


Шаг «Форма» сразу будет заполнен данными открытой формы списка «Партнеры»:



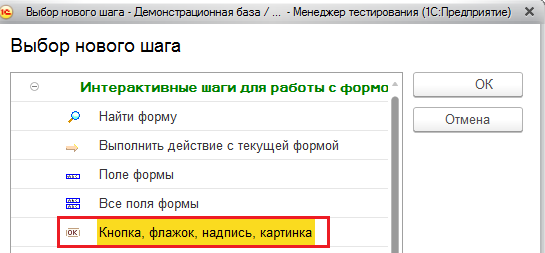
Шаг можно сохранить и закрыть без изменений.

Для группового шага «Форма» по умолчанию создается шаг «Найти форму», который ищет форму по наименованию/заголовку или реквизитам среди всех открытых форм. В общем случае его менять не нужно, в особых случаях – можно изменить правила поиска формы (например, только по наименованию, без учета заголовка).

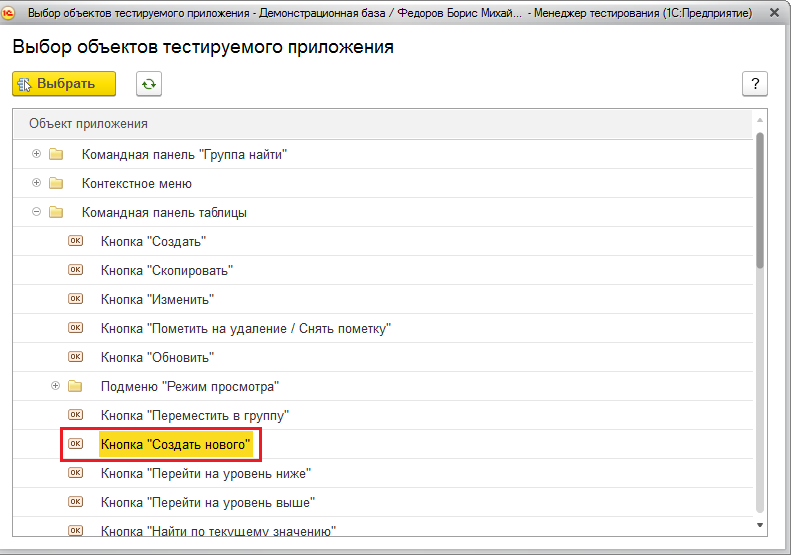


Выполнить и эти новые шаги тоже.

**Четвертый шаг.** Теперь приступим к созданию нового партнера. Нужно нажать кнопку в списке контрагентов «Создать нового». Для этого создаем новый шаг «Кнопка, флажок, надпись, картинка». Этот шаг позволит работать с элементами, расположенными на форме:

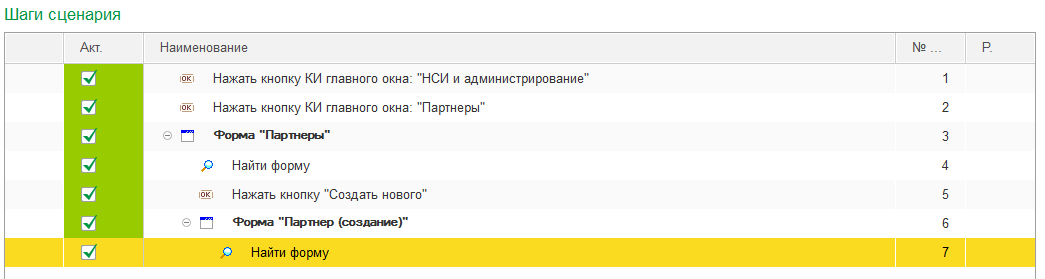


Через кнопку «…Выбрать» находим нужную нам кнопку из командной панели:



Выполняем шаг. В тестируемом приложении открывается форма нового партнера.

**Пятый шаг.** Теперь мы будем работать с формой партнера, поэтому нужно добавить групповой шаг «Форма» для этой формы. Добавляем его по аналогии с формой списка партнеров. В результате получаем:



**Шестой шаг.** Теперь нам нужно заполнить партнера данными. Вспоминаем, что нам нужно будет создавать 4 разных партнера, и чтобы не плодить шаги заполнения с константными данными, воспользуемся возможностью заполнять данные форм параметрами.

Для этого воспользуемся глобальными параметрами сценария. Это структура, доступная практически из любого шага сценария. Она используется как для внутренних нужд сценария (в ней хранятся значения итераторов цикла, переменные, создающиеся при ошибках выполнения шагов), так и может использоваться нами.

Для начала создадим и заполним новые параметры, содержащие наименование контрагента, вид контрагента (юр./физ. лицо) и email. Для этого нам потребуется шаг «Выполнить серверную процедуру». Добавляем шаг в сценарий (на верхний уровень – этот шаг нам понадобится выполнять в самом начале). В шаге прописываем:

НаименованиеКонтрагента = Строка(Формат(ТекущаяДата(), "ДФ=ddMMyyyyHHmmss"));

ГлобальныеПараметры.Вставить("НаименованиеКонтрагента", НаименованиеКонтрагента);

ГлобальныеПараметры.Вставить("ВидКонтрагента", "Частное лицо");

Email = НаименованиеКонтрагента + "@mailforspam.com";

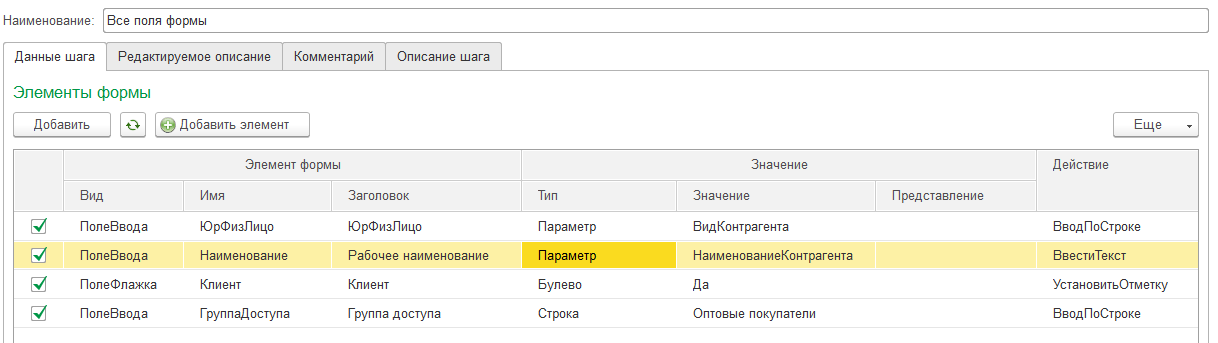
ГлобальныеПараметры.Вставить("Email", Email);

В результате выполнения этого шага появятся 3 новых параметра сценария. Все существующие параметры и их значения можно просмотреть из главного окна сценария по кнопке «Сценарий» - «Параметры».

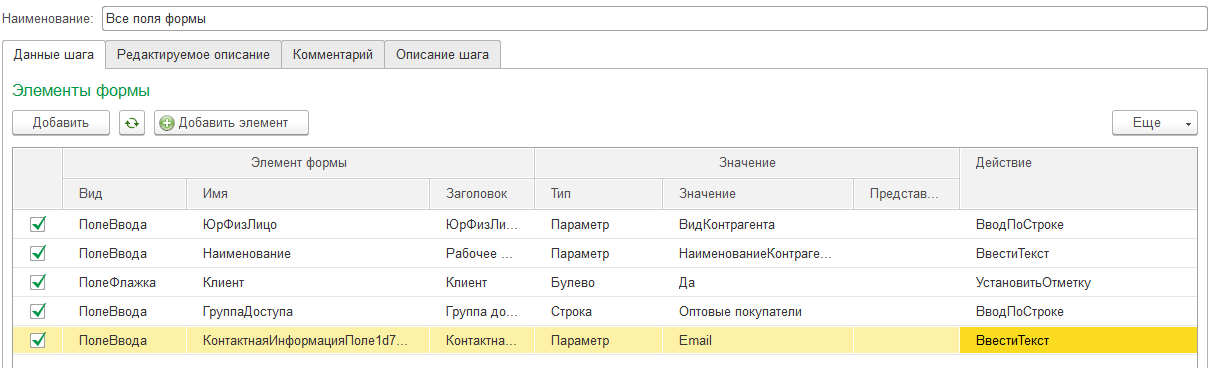
**Седьмой шаг.** Далее заполним этими параметрами нашего контрагента. Для этого добавим шаг «Все поля формы». На закладке «Данные шага» оставим только нужные нам поля – ЮрФизЛицо, Наименование, Клиент, ГруппаДоступа (обязательное для заполнения поле).

Так как вид контрагента и наименование заполняются из параметров, в колонке «Тип» для этих строк выбираем вариант «Имя параметра», в колонке «Значение» - пишем имя параметров. Отдельное внимание нужно уделить колонке «Действие». В зависимости от типа поля нужно выбирать соответствующее действие: если это просто строка и нет никаких выпадающих списков – «ВвестиТекст», если есть выпадающий список – «ВводПоСтроке» (возможны другие варианты, проще определить подбором).

Остальные поля заполняем константами:

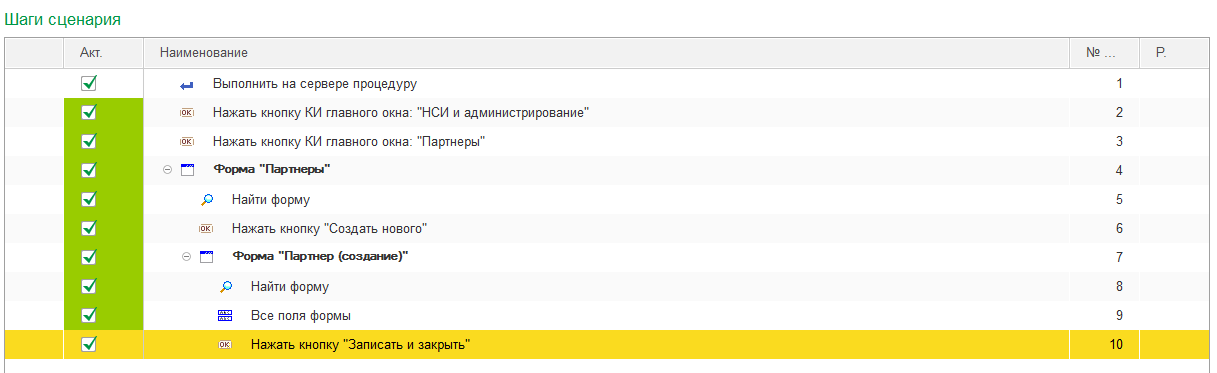


Обращаем внимание, что здесь не видно поля email. Его можно добавить по кнопке «Добавить элемент» - находим поле email (названия полей контактной информации не информативны, чтобы точно определить нужное поле – в клиенте тестирования встаем на нужное поле, в сценарии при добавлении элемента текущее поле будет выделено жирным). Выбираем тип «Параметр», значение «Email» и действие «ВвестиТекст»:



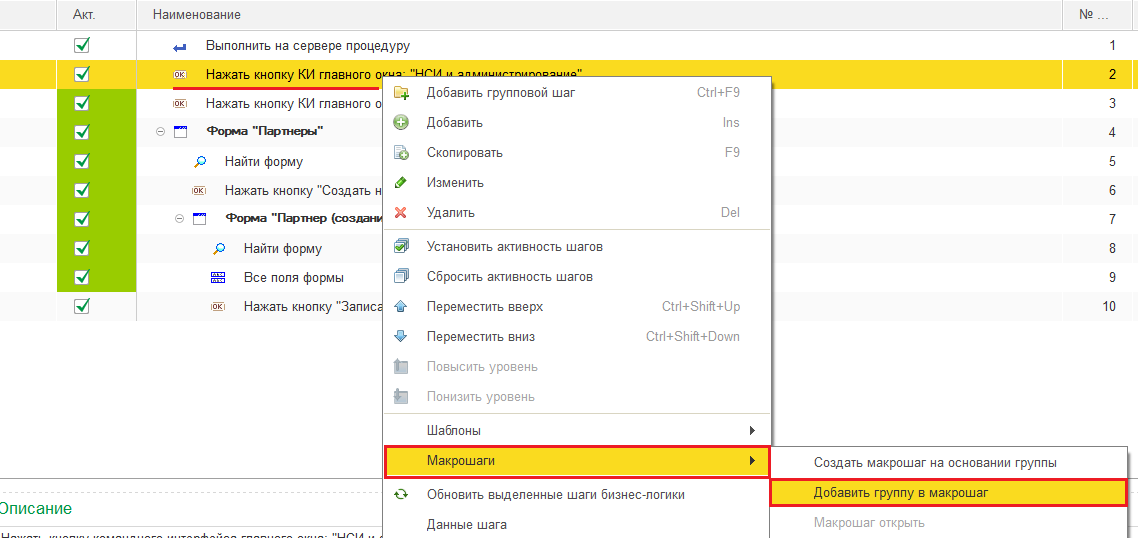
Теперь все правила заполнения полей формы созданы, сохраняем и выполняем шаг.

**Восьмой шаг.** После этого нужно сохранить нового партнера – создадим шаг для нажатия кнопки «Записать и закрыть». Добавляем шаг «Кнопка, флажок, надпись, картинка». Шаг заполняется данными кнопки формы по умолчанию – в данном случае это кнопка «Записать и закрыть» - то, что нам и нужно. Сохраняем и выполняем шаг.



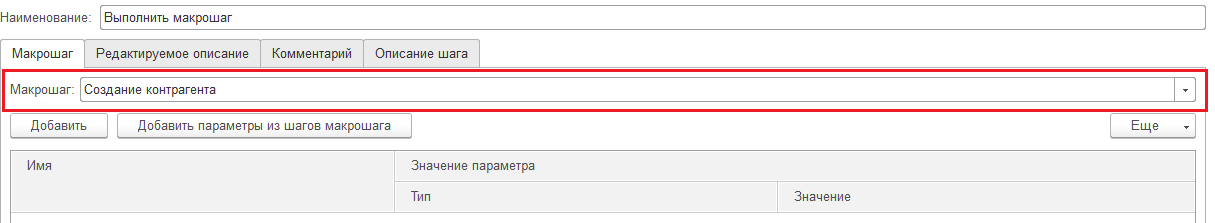
Итак, мы создали набор шагов сценария, который создает 1 контрагента. Теперь видоизменим его, чтобы создавали 4 разных контрагента.

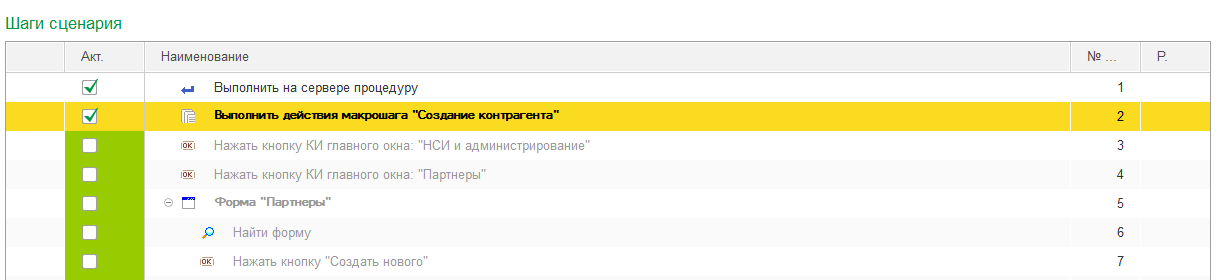
**Девятый шаг.** Для начала на основании уже имеющихся шагов создадим макрошаг. Для этого сначала создаем макрошаг – «Сценарий» - «Макрошаги» - «Добавить», даем ему наименование (например, «Создание контрагента») и сохраняем. Далее для каждого шага через контекстное меню выберем «Макрошаги» - «Добавить группу в макрошаг» и выбираем созданный ранее макрошаг.



Группа «Форма Партнеры» добавляется целиков, со всеми вложенными шагами.

Теперь удаляем шаги сценария, которые были добавлены в макрошаг (либо отключаем у них активность). Вместо них добавляем шаг сценария «Выполнить макрошаг» и указываем наш макрошаг «Создание контрагента»:





**Десятый шаг.** Теперь по аналогии с первым контрагентом создадим 3 оставшихся. Для этого нам нужно будет скопировать существующую пару шагов и изменить в каждом случае шаг «Выполнить серверную процедуру».

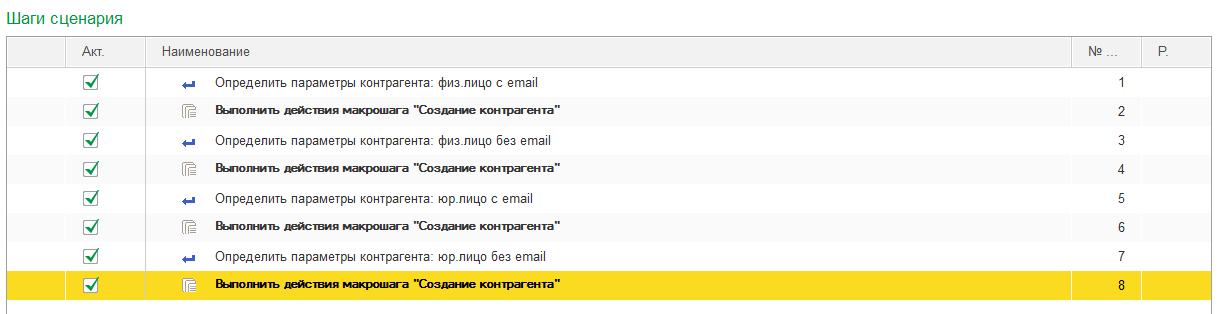
К примеру, для контрагента юр.лицо без email шаг будет содержать следующий код:

НаименованиеКонтрагента = Строка(Формат(ТекущаяДата(), "ДФ=ddMMyyyyHHmmss"));

ГлобальныеПараметры.Вставить("НаименованиеКонтрагента", НаименованиеКонтрагента);

ГлобальныеПараметры.Вставить("ВидКонтрагента", "Компания");

«Причешем» немного наш сценарий – дадим шагам понятные наименования, удалим все лишнее – и получится:



В результате у нас получился сценарий, который умеет создавать 4 разных контрагентов с уникальными наименованиями. Причем, если будет необходимость поменять механизм создания контрагента, менять мы будем только макрошаг, а не все места его использования.

Естественно, что данный пример создания контрагентов является тривиальным и может быть реализован и другими средствами. Однако это дает нам понимание хотя бы небольшой части возможностей, предоставляемых сценарным тестированием. В нашем случае, подобное создание контрагентов было использовано в рамках большого сценария тестирования процесса продажи-закупки в базе УТ и связки её со сторонней программой. В результате мы получили сценарий, который воспроизводит все типовые пользовательские действия на тестовой базе перед/после каждого обновления систем (1С:УТ и сторонней программы).

Некоторые моменты, которые не удалось логически включить в сценарий, но хочется рассказать:

1. Глобальные параметры. К ним возможно обращение не только из серверной процедуры, но и из клиентской. Для этого в шаге «Выполнить клиентскую процедуру» нужно прописать, к примеру:   
   &Вставить("<ИмяПараметра>", <ЗначениеПараметра>);  
   То есть вместо конструкции «ГлобальныеПараметры.» пишем просто «&»
2. В шаге «Выполнить клиентскую процедуру» можно обращаться к элементам формы. Например, на форме расположено поле «Код объекта в другой системе», из которого нужно получить значение и проверить, заполнено ли оно:

СтраницыФормы = ПеремТестируемаяФорма.ПолучитьПодчиненныеОбъекты()[1];

ОсновнаяСтраница = СтраницыФормы.ПолучитьПодчиненныеОбъекты()[0];

ГруппаИдентификации = ОсновнаяСтраница.ПолучитьПодчиненныеОбъекты()[1];

ПолеКодОбъекта = ГруппаИдентификации.ПолучитьПодчиненныеОбъекты()[1];

КодОбъекта = ПолеКодОбъектаВШине.ПолучитьПредставлениеДанных();  
Может быть, код не самый красивый, но принцип должен быть понятен.