

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЕЛЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. И.А. БУНИНА»

Д.В. Корниенко

**Особенности ведения учета
в типовой конфигурации
1С:ERP Управление предприятием 2**

Учебно-методическое
пособие

Елец – 2021

УДК 004.422
ББК 30.2-5-05
К 67

Печатается по решению редакционно-издательского совета
Елецкого государственного университета им. И.А. Бунина
от 28.01.2021, протокол № 1

Рецензенты:

Н.А. Трубицына, кандидат экономических наук, доцент кафедры
экономики и управления им. Н.Г. Нечаева
(ФГБОУ ВО «Елецкий государственный университет им. И.А. Бунина»);
О.Б. Гладких, кандидат физико-математических наук, доцент кафедры
медицинской информатики и кибернетики
(ФГБОУ ВО «Елецкий государственный университет им. И.А. Бунина»)

Д.В. Корниенко

К 67 Особенности ведения учета в типовой конфигурации 1С:ERP Управление предприятием 2: учебно-методическое пособие. – Елец: ФГБОУ ВО «Елецкий государственный университет им. И.А. Бунина», 2021. – 82 с.
ISBN 978-5-00151-245-5

Пособие посвящено описанию работы с расширениями для типовой конфигурации 1С:ERP Управление предприятием 2. Расширения конфигурации позволяют значительно упростить адаптацию типового прикладного решения к потребностям конкретного внедрения, конкретного заказчика. Стратегия, предлагаемая расширениями, заключается в том, что изменять типовую конфигурацию не нужно. Все изменения выполняются в расширении, которое, по сути, тоже является конфигурацией. Платформа автоматически, в режиме 1С:Предприятие, объединяет расширения с типовой конфигурацией. В результате заказчик работает с изменённым, по его желаниям, типовым решением. Когда поставщик выпускает новую версию типовой конфигурации, выполняется её автоматическое обновление, поскольку режим поддержки типовой конфигурации не менялся. Она осталась на полной поддержке поставщика. А при запуске обновлённого прикладного решения платформа снова автоматически объединит изменённую типовую конфигурацию с расширением. И заказчик вновь продолжит работать с изменённым, по его желаниям, типовым решением.

Учебно-методическое пособие предназначено для студентов математических и экономических направлений подготовки.

УДК 004.422
ББК 30.2-5-05

ISBN 978-5-00151-245-5

© ФГБОУ ВО «Елецкий государственный университет им. И.А. Бунина», 2021

Оглавление

Введение	4
Глава 1. Расширение конфигурации	13
1.1. Особенности и ограничения	17
1.2. Управление расширениями	19
1.3. Работа с расширением	21
1.4. Создание собственного объекта	28
1.5. Программные модули	32
1.6. Способы расширения	35
1.7. Редактирование программного модуля	41
1.8. Взаимодействие нескольких расширений	43
Глава 2. Форма	47
2.1. Заимствование формы	47
2.2. Работа с модулем формы	51
2.3. Построение результирующей формы	53
Глава 3. Права доступа и роли	57
3.1. Заимствованные роли	59
3.2. Собственные роли	61
3.3. Расширение прав пользователей и профили безопасности	64
3.4. Отчеты и обработки, XDTO-пакет, Web-сервис, HTTP-сервис	66
3.5. Табличный документ	68
3.6. Планы обмена, Справочники, Документы, Перечисления	70
3.7. Расширение и распределенная информационная база	73
Список используемых источников	82

ВВЕДЕНИЕ

Потребность в кастомизации программного обеспечения, т.е. его изменения под нужды конкретного пользователя, появилась, наверное, одновременно с самим программным обеспечением. Трудно написать программу, которая удовлетворит всех, а потому заложить в нее возможность изменений без привлечения производителя программы – хорошая идея. Особенно если дело касается бизнес-приложений, так как бизнес-процессы даже в одних и тех же областях могут отличаться в разных организациях.

Есть разные стратегии кастомизаций. Если приложение поставляется в исходных кодах, то самый очевидный подход – переписать исходный код под свои нужды. И самая очевидная в этом случае проблема – переход на новую версию приложения, т.к. он влечет слияние (merge) исходных кодов измененной на стороне клиента версии и новой версии от поставщика. А это может быть нетривиальной задачей, особенно если код на стороне клиента сильно кастомизирован.

Более безопасная в этом плане стратегия – плагины. Исходное приложение предоставляет плагину фиксированный набор интерфейсов, а также возможность зарегистрировать себя в приложении. При выходе новой версии приложения плагины, написанные для предыдущей версии, продолжают работать и в новой версии (при условии неизменности интерфейсов). Поведение плагинов в новой версии может отличаться от поведения в предыдущей, если поставщик ПО изменил поведение приложения. Концепция плагинов используется в самых разнообразных классах ПО – офисном и бизнес-софте, средах разработки (Visual Studio, Eclipse), графических и звуковых редакторах и т.п.

Еще одна технология кастомизации – возможность оформления подписки (subscription) на события в приложении и выполнения пользовательского кода на общеизвестном или проприетарном языке во время этих событий. События могут быть самого разного вида – открытие окна, загрузка изображения (для графического редактора), обработка заказа (для бизнес-системы).

Одна из разновидностей такого подхода – встраивание в основную программу возможности выполнять пользовательские скрипты на языках типа Visual Basic for Application (VBA). Кастомный код может, в частности, выполняться в ответ на события приложения. Тот же VBA показал себя очень мощным и гибким средством кастомизации; он встроен в Microsoft Office, AutoCAD, SolidWorks, CorelDRAW, WordPerfect, ESRI ArcGIS и другие продукты.

В платформе «1С:Предприятие» реализованы разные стратегии кастомизации. Поскольку прикладные решения 1С поставляются в исходных кодах, естественно, один из самых очевидных сценариев – изменение исходного кода.

Когда клиент меняет исходный код решения 1С под свои нужды, ему надо помнить, что поставщик приложения тоже не бездействует и выпускает новые версии, добавляя функциональность и исправляя ошибки. Чтобы при установке новой версии приложения не потерялись изменения, сделанные под потребности клиента, нужно каким-то образом произвести слияние (merge) измененной предыдущей версии приложения и новой версии.

Естественно, в «1С» уделяли большое внимание этой задаче и разработали механизм поставки и поддержки, облегчающий ее решение. Прежде чем рассказать, как он работает, опишем детали внутреннего устройства решений «1С».

Исходные коды и метаданные прикладного решения «1С» (конфигурации) хранятся в базе данных, в той же самой, в которой лежат данные самого приложения (проводки, данные справочников и документов и т.п.), т.е. программа хранится вместе с данными. База данных с конфигурацией (и данными приложения) в терминологии 1С называется информационной базой. В процессе разработки поставщик конфигурации определяет, какие объекты конфигурации (справочники, документы и т.п.) клиент может менять, а какие – нет.

Клиент на своей стороне с помощью этого механизма также может определять правила поддержки объектов внедренной конфигурации поставщика –

например, он может отказаться от поддержки поставщиком конкретного объекта, если возьмет на себя ответственность за дальнейшую модификацию этого объекта. А можно, наоборот, запретить редактирование объекта «своей» конфигурации (даже если поставщик разрешает это делать) с тем, чтобы застраховаться от случайного изменения.

Когда клиент начинает что-то менять в типовой конфигурации, в информационной базе создаются две конфигурации:

1. Оригинальная конфигурация поставщика.
2. Текущая конфигурация, измененная на стороне клиента.

И вот поставщик выпускает новую версию. Она может поставляться в виде полного приложения, а может – в виде пакета обновления с измененными объектами. При переходе на новую версию, у нас на стороне клиента имеется три конфигурации, на основании которых осуществляется так называемое трехстороннее слияние конфигураций:

1. Старая конфигурация от поставщика.
2. Текущая конфигурация клиента (старая конфигурация от поставщика плюс изменения, сделанные в ней клиентом).
3. Новая конфигурация от поставщика.

Понятно, что в ряде случаев объекты, измененные поставщиком, можно обновлять автоматически:

- Объекты, не измененные клиентом.
- Простые изменения объектов на стороне клиента (например, добавление дополнительных реквизитов к объекту).

В случае же, когда объект был изменен и на стороне клиента, и в новой версии от поставщика, необходимо ручное вмешательство. В платформе есть мощный механизм сравнения и объединения не только для модулей кода, но и для моделей (метаданных, форм, отчетов и др.), но даже с этим механизмом объединение конфигураций может быть нетривиальной задачей.

Другой механизм кастомизации, сравнительно безопасный с точки зрения перехода на новые версии – это механизм внешних отчетов и обработок. Как

следует из названия, оба типа объектов – внешние отчеты и внешние обработки – являются внешними по отношению к прикладному решению, хранятся в отдельных файлах и загружаются в прикладное решение в момент исполнения. Таким образом, процесс перехода на новую версию не затрагивает их вовсе. Но в случае, если в новой версии реквизиты какого-либо объекта были удалены или переименованы, а обработка или отчет обращаются к ним – на новой версии, то отчет или обработка без переделки не заработают.

Внешние отчеты и обработки по сути являются плагинами. Они хорошо подходят для «кастомных» отчетов и специфических операций над данными (например, для импорта информации из других систем), но, конечно, покрывают далеко не все сценарии кастомизации. Если клиенту нужно добавить специфический код, который должен выполняться при проведении документа, внешней обработкой тут не обойтись – надо править исходный код модуля документа.

С появлением облачной технологии 1CFresh задача кастомизации вышла на новый уровень. Дело в том, что в «облаке» пользователи прикладного решения из разных организаций могут физически работать с одной информационной базой (т.е. с одним экземпляром приложения), но, в то же время, благодаря механизму разделения данных, видят только данные своей организации. Кастомизация через изменение исходного кода здесь становятся неприемлемой – каждой организации нужны свои кастомизации, и совершенно не нужны кастомизации «соседей» по информационной базе.

В «облаке» для кастомизации применимо только использование внешних отчетов и обработок, но, как говорилось выше, внешние отчеты и обработки покрывают далеко не все нужные пользователям сценарии.

Итак, нам нужно было придумать механизм кастомизации, который бы удовлетворял следующим требованиям:

1. Позволял бы легко обновлять кастомизированное решение на новую версию, избегая ручной работы по объединению конфигураций.

2. Позволял включать кастомизацию при определенных условиях (например, если мы работаем в контексте определенной организации).

3. Снижал вероятность потери работоспособности кастомизации при переходе на новую версию исходной конфигурации.

4. Имел возможность отключения кастомизации в случае проблем для сохранения работоспособности приложения.

Такой механизм, помимо применения в облачных решениях, сильно облегчил бы жизнь при переходе на новую версию на внедрениях типовых конфигураций, где необходимы кастомизации.

Разработчики придумали такой механизм и назвали его расширения. Этот механизм в каком-то смысле совмещает в себе два подхода к кастомизации – идеологию плагинов и механизм подписок.

Расширения – это способ держать изменения конфигурации отдельно от самой конфигурации. Расширение, по сути, само является отдельной конфигурацией, содержащей измененные объекты. Оно так же, как и конфигурация, представляется в виде дерева объектов. Для работы с расширением используются те же приёмы работы, что и с обычной конфигурацией:

Если мы хотим задействовать в расширении объект из основной конфигурации (например, добавить новую форму к существующему в основной конфигурации документу) – нам вначале надо позаимствовать объект к себе в расширение через команду «Добавить в расширение». Сразу после добавления объекта в расширение он «пустой» в дереве объектов расширения – у него есть только те свойства, которые есть в основной конфигурации. Можно также позаимствовать из основной конфигурации уже существующую форму и, например, добавить к ней новую кнопку, выполняющую какое-либо специфическое действие. К объектам в расширениях пока нельзя добавлять новые реквизиты.

В расширении также есть аналог подписки на события – возможность обрабатывать события объектов расширяемой конфигурации, например, обработку записи. Можно указать, как именно будет вызываться наш код в расширении.

Можно перед стандартной процедурой записи документа вызвать наш код, который, например, проверит, заполнено ли поле ответственного за документ сотрудника, и, если нет – запишет в это поле текущего пользователя.

В новой версии конфигурации реализация записи документа может измениться, но наш код в расширении будет по-прежнему выполняться перед стандартным кодом записи документа и делать свою работу.

Во время исполнения типовая конфигурация и расширения (их может быть несколько) «складываются», давая в итоге новую, кастомизированную конфигурацию, с которой и работает конечный пользователь.

При разработке расширений следует помнить, что платформа не гарантирует одинакового порядка выполнения расширений при добавлении нескольких расширений к конфигурации. Разработчики сознательно отказались от явного задания порядка выполнения расширений, так как это усложняет настройку и в конечном счете приносит больше проблем, чем решает.

Если к конфигурации добавляются несколько расширений, в каждом из которых есть обработка проведения одного и того же документа с директивой «&После», то все обработчики будут выполнены, но платформа не гарантирует, что порядок их выполнения всегда будет одинаков. Это надо учитывать при разработке расширений.

В случае наличия в нескольких расширениях обработчика одного и того же события с директивой «&Вместо» будет выполнен только один обработчик, причем какой – заранее сказать нельзя. Об этом нужно помнить и отслеживать, чтобы к конфигурации не более одного расширения имели обработчик «&Вместо» для одного и того же объекта/события.

Можно позаимствовать в свое расширение форму объекта из конфигурации (например, форму документа). При этом в визуальном редакторе формы в расширении мы увидим форму такой же, как и в основной конфигурации. А в редакторе кода формы в расширении будет пусто – весь код для формы пока содержится только в основной конфигурации.

На форму можно добавить новую кнопку (или даже несколько). В случае если несколько расширений добавляют на одну и ту же форму свои кнопки – все они будут присутствовать на итоговой форме во время исполнения.

А вот удалять стандартные элементы с формы не рекомендуется – это может сломать существующий в оригинальной конфигурации код (если он обращается к элементам формы). Если уж есть такая нужда – лучше делать элементы невидимыми через свойство «Видимость».

Нужно учитывать, что приложение на «1С:Предприятии» – это не просто код на языке программирования. Большая часть приложения описывается в виде декларативных моделей. Причем для разных задач используются разные виды моделей (формы, отчеты, права и др.). Для каждого вида модели мы подбираем свой способ кастомизации в расширениях, обеспечивающий наиболее удобное изменение для типичных случаев.

Расширения имеют идеологическое отличие от механизма поставки и поддержки. В механизме поставки и поддержки разработчик правит конфигурацию поставщика, как хочет, как будто просто дорабатывает свою конфигурацию, а потом (при обновлении) разбирается с тем, как синхронизировать конфликтующие изменения. В расширениях разработчик сразу изначально разрабатывает именно расширение – в терминах добавляемой функциональности. Расширение хранится системой именно как дополнение и система заботится о максимально безопасном обновлении.

Когда поставщик выпускает новую версию типовой конфигурации, выполняется автоматическое обновление, поскольку режим поддержки типовой конфигурации не менялся – она осталась на полной поддержке поставщика. А при запуске обновлённого прикладного решения платформа снова автоматически объединит изменённую типовую конфигурацию с расширением. И клиент продолжит работать с изменённым под его потребности типовым решением.

Иногда все же после обновления версии типовой конфигурации может потребоваться адаптация расширения под новую версию, например, если в но-

вой версии переименованы объекты или реквизиты объектов, задействованных в расширении.

Первое очевидное преимущество расширений – все кастомизации, сделанные для клиента, лежат отдельно от типовой конфигурации, и теперь не надо сравнивать измененную конфигурацию с типовой, чтобы понять, что же именно было изменено.

Для того чтобы задействовать в расширении объект из основной конфигурации, его надо позаимствовать в расширение из основной конфигурации. Таким образом, в расширении появляется нечто вроде ссылки на объект из основной конфигурации.

При этом есть способ понять, какие заимствованные объекты в конфигурации действительно изменены, а какие позаимствованы в режиме read-only – например, для использования в отчетах. В дереве объектов расширения есть кнопка фильтра «Изменённые и добавленные в расширении», после нажатия которой в дереве остаются только заимствованные объекты, модифицированные в этом расширении, и новые объекты, созданные в этом расширении.

Предположим, мы позаимствовали в расширение справочник Контракты из основной конфигурации для использования его в отчете. Тем временем вышла новая версия типовой конфигурации, в которой справочник Контракты был переименован в Договоры. Естественно, после перехода на новую версию наш отчет в расширении работать не будет. Если бы мы использовали старую технологию кастомизации – внешний отчет, то ошибка возникла бы только в момент выполнения отчета. В случае же расширений у нас есть возможность проверить корректность расширений в design-time после обновления версии типовой конфигурации, и исправить все проблемы до того, как пользователи начнут работу. Особенно это актуально на внедрениях, где в одной информационной базе используется много расширений, а обновление версии конфигурации проводится централизованно (например, на облачных внедрениях, где разные организации, использующие одну и ту же конфигурацию, но работающие в разных областях данных, могут использовать разные наборы расширений). Адми-

нистратор может на тестовой базе обновить версию типовой конфигурации и проверить корректность всех пользовательских расширений относительно новой версии. В случае проблем он сообщит о них владельцам расширений, и переход рабочей базы на новую версию типовой конфигурации будет сделан только тогда, когда все расширения будут приведены в соответствие с новой версией конфигурации.

Таким образом развитие расширений является одним из главных направлений развития средств кастомизации в платформе «1С:Предприятие». Расширения, задуманные изначально для облегчения кастомизаций в облачном сервисе, были спроектированы так, чтобы облегчить и ситуации с кастомизациями и на необлачных внедрениях.

ГЛАВА 1. РАСШИРЕНИЕ КОНФИГУРАЦИИ

Механизм расширения конфигурации – это специальный механизм, предназначенный для доработки расширяемой конфигурации без изменения этой конфигурации (в том числе без снятия с поддержки).

При рассмотрении механизма расширения конфигурации будут использоваться следующие термины:

Расширяемая конфигурация – основная конфигурация информационной базы, для которой применяется расширение или для которой расширение разрабатывается.

Расширение конфигурации – набор объектов конфигурации, подключаемых к расширяемой конфигурации и содержащий набор объектов, добавляемых к расширяемой конфигурации. Расширение может включать в себя как объекты расширяемой конфигурации, так и объекты, которые отсутствуют в расширяемой конфигурации.

Собственный объект – самостоятельный объект конфигурации, который может находиться как в расширяемой конфигурации, так и в расширении.

Заемствованный объект – собственный объект, добавленный в расширение конфигурации.

Расширяемый объект – собственный объект, для которого в заимствованном объекте изменены какие-либо параметры (свойства, формы и т. д.).

Расширяющий объект – это заимствованный объект, в который внесены изменения относительно расширяемого объекта. Наличие в заимствованном объекте только контролируемых свойств не делает заимствованный объект расширяющим.

Результирующий объект – это собственный объект плюс объединение всех расширяющих объектов (если расширений несколько). Если для собственного объекта нет расширяющих объектов – он становится результирующим «без изменений». Т.е. в конфигурации, с которой работает пользователь - все

объекты являются результирующими, вне зависимости от наличия и количества установленных расширений.

Расширяющее свойство – свойство заимствованного объекта, которое изменяет одноименное свойство расширяемого объекта.

Контролируемое свойство – свойство заимствованного объекта, значение которого проверяется при подключении расширения к расширяемой конфигурации. Если при подключении расширения (в режиме 1С:Предприятие) значение контролируемого свойства в расширении не совпадет со значением этого же свойства в расширяемой конфигурации, расширение не будет подключено.

Проверяемое свойство – является аналогом контролируемого свойства. Отличие заключается в том, что несовпадение контролируемого свойства приведет к невозможности подключения расширения, а несовпадение проверяемого свойства приведет только к выводу предупреждения о несовпадении. Расширение в этом случае будет подключено и сохранит работоспособность.

Модифицируемое свойство – свойство заимствованного объекта, значение которого в результирующем объекте будет получаться из расширения.

Расширение данных – это возможность расширять структуру хранимых данных расширяемой конфигурации с помощью заимствованных и собственных объектов расширения.

Свойство заимствованного объекта не может быть одновременно контролируемым и модифицируемым.

Основное назначение расширения конфигурации – это доработка прикладного решения при внедрении (или в «облаке») под нужды клиента. При этом дорабатываемую конфигурацию не надо снимать с поддержки. В результате сохраняется простота обновления типового прикладного решения, стоящего на поддержке, с необходимостью выполнять доработки. При разработке расширения следует понимать некоторые особенности функционирования расширения конфигурации. Так, расширяемая конфигурация в любой момент может быть изменена, например, в результате обновления. При этом разработчик

расширения не может никак повлиять на возможность или невозможность обновления. Также следует учитывать тот факт, что в системе могут функционировать более одного расширения и автор каждого расширения (в общем случае) не знает, как функционирует другое расширение.

Объекты, размещенные в расширении конфигурации, могут изменять поведение существующих объектов расширяемой конфигурации, не изменяя структуру данных, которые хранятся в информационной базе. В тоже время объекты, расположенные в расширении конфигурации, имеют возможность дополнять структуру хранения расширяемых объектов конфигурации (добавлять реквизиты) и создавать собственные объекты расширения, которые будут хранить «свои» данные в информационной базе расширяемой конфигурации.

Сопоставление объектов расширяемой конфигурации и расширения выполняется по имени объекта. Это означает, что изменение имени объекта в расширяемой конфигурации вызовет невозможность подключения расширения к такой конфигурации. Аналогичная ситуация (невозможность подключения расширения) возникнет при изменении в расширении имени заимствованного объекта при неизменности имени расширяемого объекта в расширяемой конфигурации. Однако платформа предлагает инструменты для упрощения поиска таких проблем, которые будут описаны далее в этой главе.

Каждое расширение имеет свое назначение (свойство расширения Назначение расширения конфигурации). Назначение расширения конфигурации описывает, для какой цели создается это расширение. Расширение может иметь одно из следующих назначений:

- Исправление – такое расширение предназначено для исправления ошибок в прикладном решении. В таких расширениях предполагается использование потенциально «опасных» возможностей расширений, например, применение расширения метода с помощью аннотации `Вместо`. Исправления предназначено для определенной версии прикладного решения. При выходе следующей версии этого решения автор расширения должен проводить анализ применимости созданных расширений в новой версии. Допускается наличие несколь-

ких расширений с таким назначением, но необходимо обеспечить отсутствие конфликтов между такими расширениями, например, несколько таких расширений не должны расширять один и тот же метод с разными целями. Такие расширения могут не учитывать наличия расширений другого назначения.

- Адаптация – такое расширение предназначено для адаптации прикладного решения под условия конкретного клиента. В таких расширениях рекомендуется не использовать потенциально «опасных» возможностей, т. е. тех возможностей, которые могут привести к конфликту расширений при их совместной работе или которые зависят от порядка подключения расширений. Тем не менее, допускается аккуратное использование «опасных» возможностей, при условии, что автор расширения берет на себя полную ответственность за обеспечение корректного функционирования результирующей конфигурации в новых версиях прикладных решений, и с учетом расширений, имеющих назначение Исправление. Предполагается, что в каждый момент времени в информационной базе существует минимальное количество таких расширений. Если в одно расширение невозможно включить весь набор изменений, то рекомендуется расширения с назначением Адаптация разбивать по максимально крупным блокам расширяемого прикладного решения.

- Дополнение – такое расширение предназначено для реализации новых возможностей прикладных решений, которые минимально привязаны к конкретной версии прикладного решения. Примером такого расширения может служить создание нового набора отчетов, который отсутствует в расширяемом прикладном решении. Предполагается, что такие расширения должны корректно работать в случае обновления расширяемого прикладного решения. При этом расширение с назначением Дополнение не должно учитывать в своей работе возможное наличие расширений с другим назначением. Предполагается, что таких расширений может быть произвольное количество.

Если расширение подключается к разделенной информационной базе, то для такого расширения имеется возможность указать область действия расширения: вся информационная база или область данных.

Расширение создается в конфигураторе, хранится в информационной базе и может быть сохранено в файл. Для добавления (подключения) расширения, сохраненного в файл, в прикладное решение конкретного клиента нет необходимости использовать конфигуратор. Подключить расширение можно с помощью специальной стандартной функции (Функции для технического специалиста - Стандартные - Управление расширениями конфигурации). Подключить расширение также можно с помощью инструментария прикладного решения, которое использует программный интерфейс, предоставляемый платформой. Подключение расширения (интерактивно или из встроенного языка) возможно в следующих случаях:

- 1) в небезопасном режиме;
- 2) в том случае, когда профиль безопасности, под управлением которого работает сеанс, разрешает доступ к подключаемому расширению.

1.1. Особенности и ограничения

В расширении конфигурации не поддерживается создание следующих собственных объектов:

- Общие реквизиты.
- Регламентные задания.
- Определяемые типы.
- Хранилища настроек.
- Языки.
- Журналы документов.
- Бизнес-процессы и задачи.
- Внешние источники данных.

Не поддерживается расширение следующих объектов:

- Нельзя создавать подчиненные подсистемы.
- Если объект не может быть собственным объектом расширения – его расширение также не поддерживается.

Не поддерживается добавление реквизитов и табличных частей для:

- плана видов характеристик;
- плана счетов;
- плана видов расчетов.

Не поддерживается изменение структуры регистров всех видов. Поддерживается только расширение состава регистраторов.

Документы расширения (собственные и заимствованные) могут выполнять движения по собственным регистрам расширения. Другими словами, документы расширения могут выступать в роли регистраторов для собственных регистров расширения.

Для собственных регистров накопления не поддерживается использование агрегатов.

Среди контролируемых свойств следует особо выделить:

- Состав плана обмена;
- Предопределенные элементы для справочников, планов видов характеристик, планов счетов и планов видов расчетов.

При расширении данных следует помнить о следующих особенностях и ограничениях:

- В расширении не поддерживается использование типа ЛюбаяСсылка. Тип ЛюбаяСсылка, использованный в расширяемой конфигурации, не включает в себя типы, которые предоставлены подключенными расширениями.

- В расширении конфигурации не поддерживается использование заимствованных определяемых типов.

- Особенности поведения системы с включенным разделением.

- После подключения расширения конфигурации, выполняющего расширение данные, в такой информационной базе не поддерживается использование фоновой реструктуризации.

- Тестирование и исправление обрабатывает только структуры данных расширяемой конфигурации.

- В базовой конфигурации расширение может быть использовано только в том случае, если расширение подписано тем же ключом, что и сама базовая конфигурация.

- В мобильном клиенте с автономным режимом поддерживается использование расширений, однако не поддерживается включение расширений в состав автономной конфигурации.

1.2. Управление расширениями

Для управления расширениями в конфигураторе предназначен диалог Расширения конфигурации (рис. 1). Доступ к этому диалогу осуществляется через меню Конфигурация – Расширения конфигурации. С помощью данного диалога можно создать новое расширение, удалить существующее, открыть конфигурацию расширения, сохранить расширение в файл, а также выполнять другие операции.

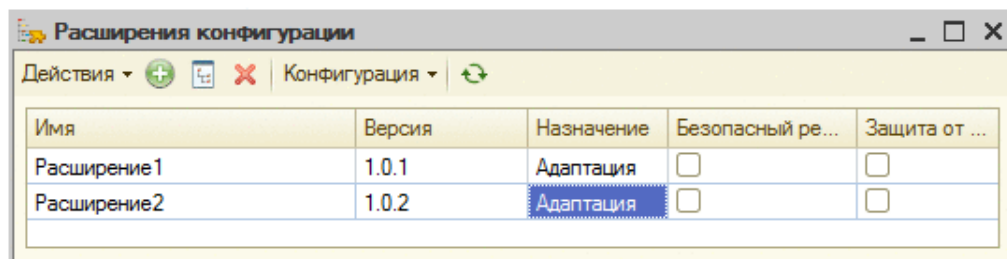


Рис. 1. Управление расширениями в конфигураторе

В общем случае получить доступ к командам работы с расширением конфигурации можно следующими способами:

1. С помощью команд командная панель панели окна Расширения конфигурации.

2. С помощью контекстного меню выбранного расширения в окне Расширения конфигурации.

3. С помощью меню Действия - Конфигурация окна с расширением конфигурации.

Команды, представленные в этих меню, по смыслу совпадают с командами, применяемыми для основной конфигурации.

При создании расширения необходимо указать имя расширения, его синоним и префикс имен методов и объектов, создаваемых в расширении (рис. 2). Этот префикс будет автоматически добавляться к новым отчетам, обработкам и подсистемам, которые созданы в расширении, а также к обработчикам событий, которые создаются в расширяющих формах. В дальнейшем этот префикс можно посмотреть в свойстве расширения конфигурации Префикс имен.

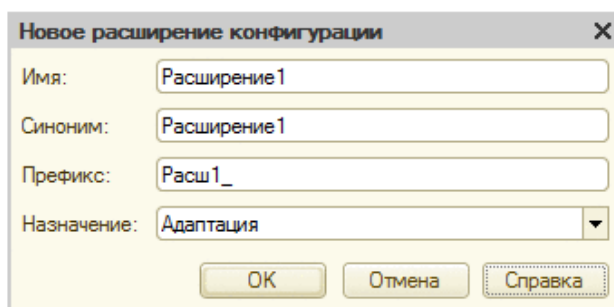


Рис. 2. Новое расширение

При создании нового расширения из расширяемой конфигурации автоматически копируются следующие свойства: Основной режим запуска, Назначение использования, Основной язык, Режим совместимости интерфейса и Режим совместимости. Эти же свойства помечаются как контролируемые.

При создании расширения автоматически создается роль с именем, которое формируется из префикса расширения и слова ОсновнаяРоль. Для расширения с префиксом Расш1_, имя автоматически созданной роли будет Расш1_Основная Роль. В созданной роли автоматически включается доступ к собственным объектам расширения. Созданная роль автоматически устанавливается в качестве значения свойства ОсновныеРоли расширения.

Для установки расширения в какое-либо прикладное решение, необходим файл с расширением конфигурации. Для выполнения этого действия следует выбрать в списке расширений конфигуратора нужное расширение и использовать команду Конфигурация - Сохранить конфигурацию в файл командной панели диалога Расширения конфигурации или контекстного меню этого окна.

В результате выполнения этого действия будет сформирован файл с расширением .cfe. Этот файл будет необходимо подключить к прикладному решению.

Для управления расширениями в режиме 1С:Предприятия предназначена стандартная функция Управление расширениями конфигурации.

Интерактивное удаление расширений (в конфигураторе или в клиентском приложении) различает удаление расширения, которое не расширяет данные и расширения, которое данные расширяет. В первом случае будет задан стандартный вопрос-подтверждение, после чего расширение будет удалено.

Во втором случае удаление выполняется в два действия.

Удаление расширения из встроенного языка выполняется без дополнительного контроля со стороны системы (метод `РасширениеКонфигурации.Удалить()`). Однако, разработчик может выполнить все необходимые проверки самостоятельно. Для проверки активности расширения предназначено свойство `РасширениеКонфигурации.Активно`. Проверить, расширяет ли расширение данные или нет, можно с помощью метода `РасширениеКонфигурации.ИзменяетСтруктуруДанных()`.

1.3. Работа с расширением

Для того, чтобы выполнить заимствование объекта, необходимо воспользоваться командой `Добавить` в расширение контекстного меню объекта конфигурации. После выполнения данной команды в расширении создается заимствованный объект. При создании заимствованного объекта заимствуются также все родительские объекты, если они еще не заимствованы в данном расширении. Подчиненные объекты не заимствуются. Поясним поведение на примерах.

Предположим, что в расширяемой конфигурации существует справочник `Товары`, который представлен некоторой структурой, которая представлена на рис. 3.

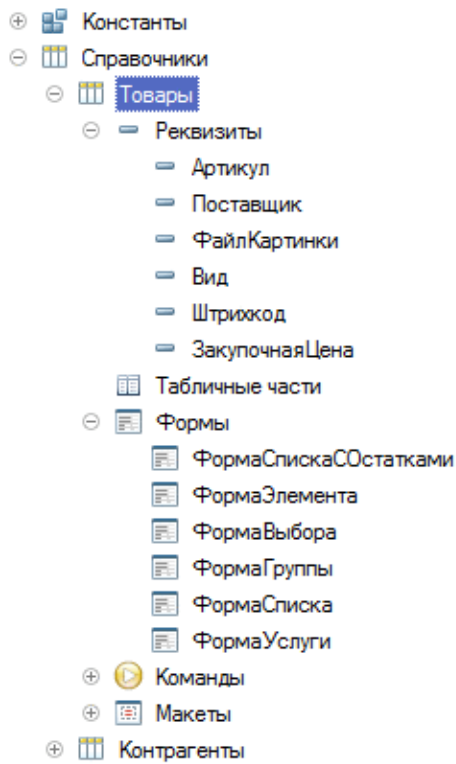


Рис. 3. Структура справочника Товары

Если добавить в расширение непосредственно сам справочник, то никакие подчиненные объекты добавлены не будут, что видно на рис. 4.

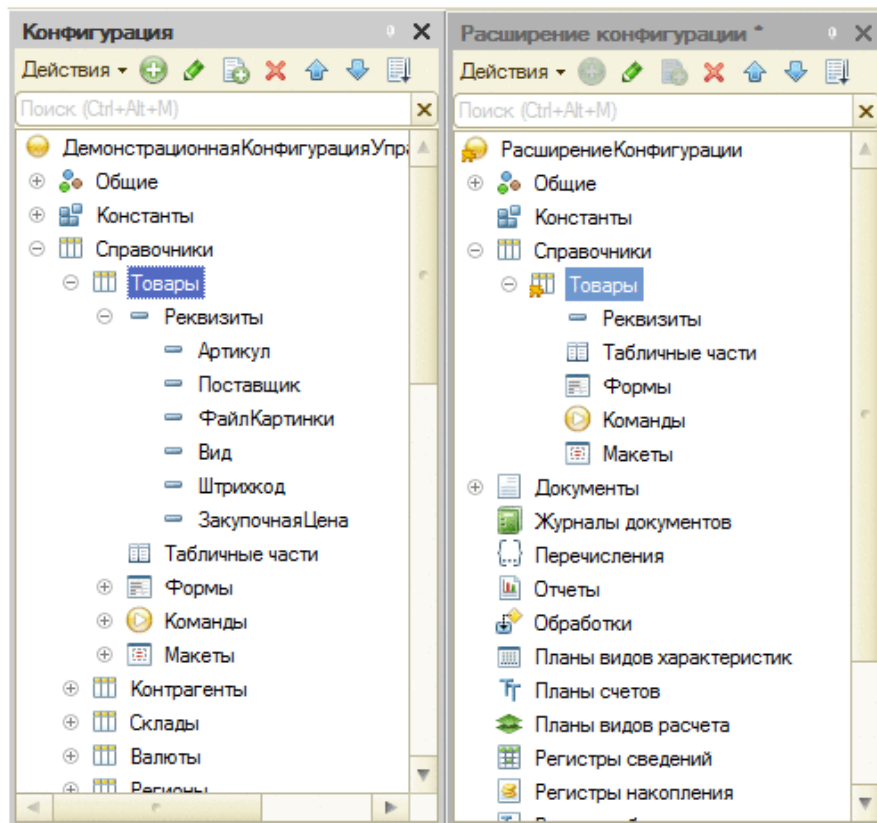


Рис. 4. После добавления объекта

Если теперь добавить форму справочника Товары, например форму элемента, и сам справочник уже заимствован, то будет заимствована только форма. Если заимствуется форма для объекта, которого нет в расширении - то будет заимствован сам объект и его форма. В самой форме будут также заимствованы все элементы этой формы, но не более.

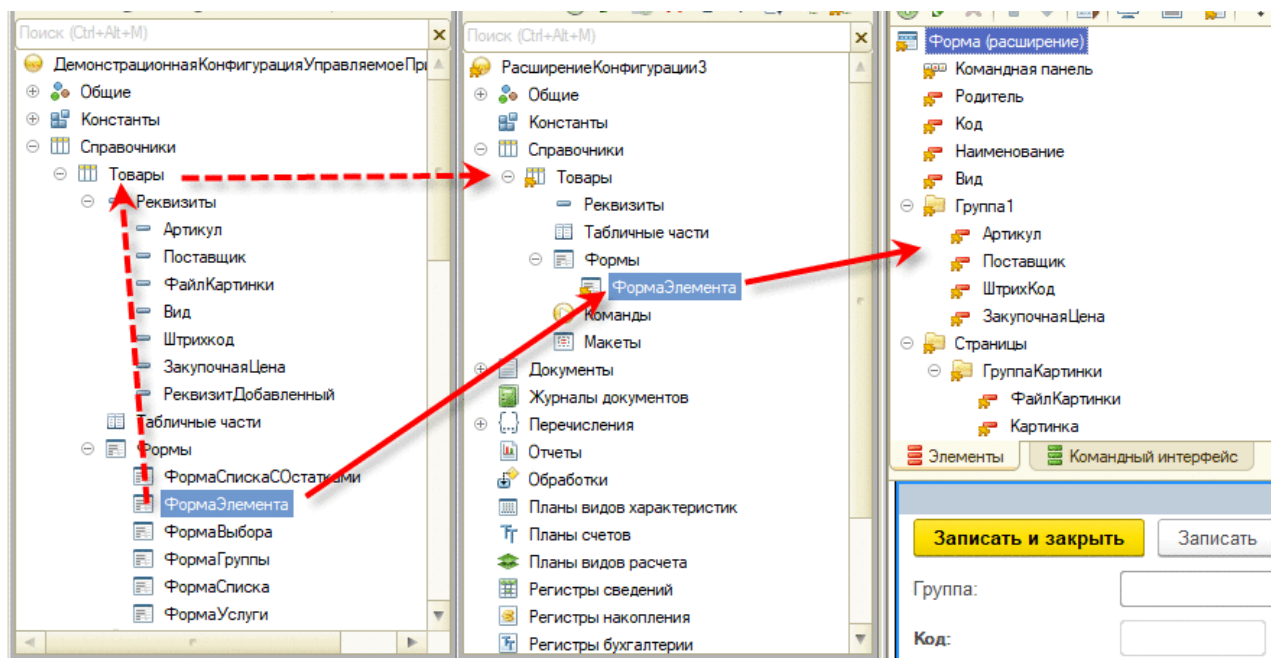


Рис. 5. После добавления объекта с родителями

На рис. 5 видно, что кроме добавляемой формы ФормаЭлемента, никаких «лишних» заимствований не выполнено.

При заимствовании составного типа поведение зависит от характеристик такого типа:

- Если в составной тип может входить тип Произвольный, то любой составной тип для такого реквизита будет заимствован как составной тип, состоящий из единственного типа Произвольный.
- Если в составной тип не может входить тип Произвольный, то такой составной тип будет заимствован по следующим правилам:
- Если составной тип включает в свой состав только примитивные типы, то будут заимствованы все типы, образующие составной тип.

- Если в составной тип входит хотя бы один ссылочный тип (СправочникСсылка, ДокументСсылка и т. д.), то такой тип будет заимствован как простой тип ЛюбаяСсылка.

Однако, если добавить в расширение, например, основной реквизит формы, то картина изменится (рис. 6).

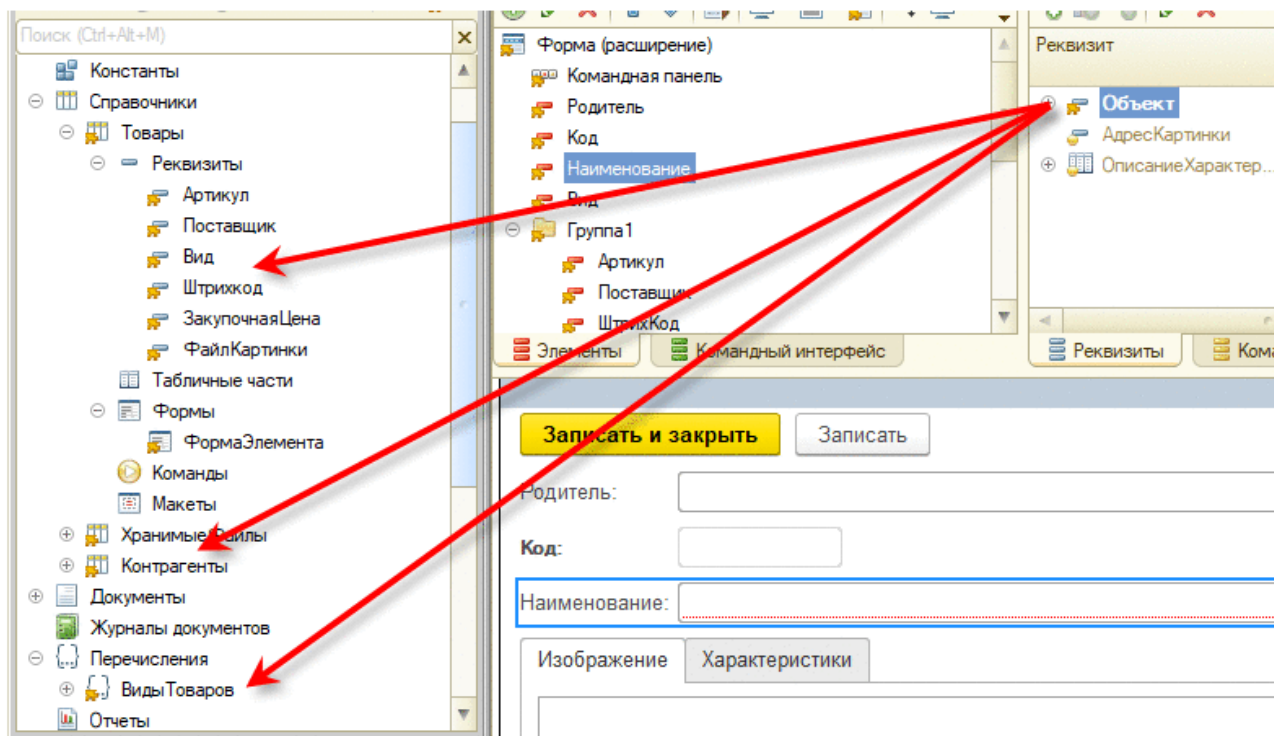


Рис. 6. Добавили объект со связями

На рисунке видно, что после добавления реквизита формы Объект (можно считать, что это тип СправочникСсылка.Товары), в расширение были добавлены все реквизиты справочника Товары и все объекты конфигурации, которые указаны в качестве типов реквизитов этого справочника.

При заимствовании параметрической команды не происходит заимствование параметра команды и, как следствие, объекта конфигурации, определяющий тип параметра. В случае необходимости следует самостоятельно выполнить заимствование объекта (или объектов, если тип составной) конфигурации, определяющего тип параметра команды и затем указать этот объект в качестве типа параметра заимствованной команды.

Работа с заимствованным объектом имеет некоторые особенности.

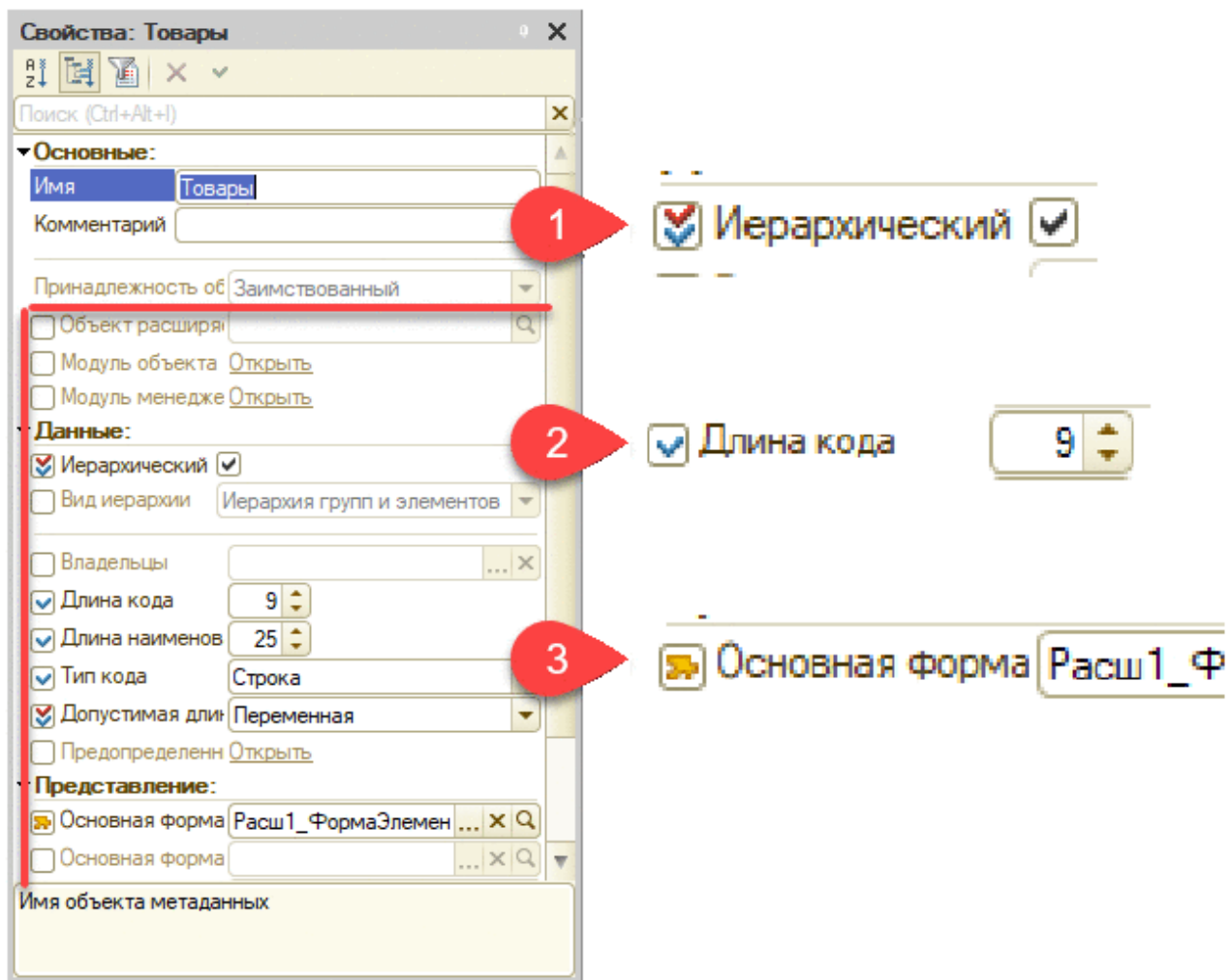


Рис. 7. Свойства заимствованного объекта

Во-первых, у заимствованного объекта присутствуют только те свойства, которые могут быть установлены контролируемыми, проверяемыми или модифицируемыми. Возможность установка признака свойства является второй особенностью панели свойств заимствованного объекта. Различные признаки свойств объекта по-разному отображаются в панели свойств для заимствованного объекта, что показано на рис. 7:

- Цифрой 1 обозначен признак контролируемого свойства. Несовпадение таких свойств между расширением и расширяемой конфигурацией приведет к невозможности применения расширения. Видно, что для справочника Товары указаны следующие контролируемые свойства: Иерархический и Допустимая длина кода.

- Цифрой 2 на рисунке обозначено проверяемое свойство. Несовпадение таких свойств между расширением и расширяемой конфигурацией вызовет

формирование предупреждений о несовпадении свойств, но не будет препятствовать применению расширения. На рисунке указаны следующие проверяемые свойства для справочника Товары: Длина кода, Длина наименования, Тип кода.

- Цифра 3 на рисунке обозначает модифицируемое свойство. На рисунке видно, что свойство Основная форма отмечено как модифицируемое.

В результирующем объекте модифицируемое свойство будет получено из расширения. Если в информационной базе (или области данных) подключено несколько расширений – модифицируемое свойство получит значение из расширения, указанного последним в списке расширений.

Если значение модифицируемого свойства не указано, то поведение системы аналогично поведению системы с неуказанным значением свойства в обычной конфигурации (без расширений). Так, например, если в качестве модифицируемого указано свойство конфигурации Основная форма отчета, а собственно форма не указана, то в качестве основной формы отчета будет выступать форма, автоматически сгенерированная системой.

Следует помнить, что свойство объекта конфигурации Имя всегда является контролируемым реквизитом. Если в расширяемой конфигурации, на момент создания расширения, справочник имел имя Товары, а потом был переименован в Номенклатура, то расширение перестанет работать до тех пор, пока для заимствованного справочника в расширении также не будет установлено имя Номенклатура. То же самое верно для значений других контролируемых свойств.

Проблема переименованных объектов расширяемой конфигурации может сильно усложнить поддержку расширения. Чтобы упростить решение этой проблемы, система предлагает возможность отслеживать соответствие объектов не только по имени, но и по внутреннему идентификатору типа объекта конфигурации. Данная возможность включается с помощью свойства расширения Поддерживать соответствие объектам расширяемой конфигурации по внутренним идентификаторам и для новых расширений включается автоматически. Если возможность включена, то для каждого заимствованного объекта

конфигурации (включая подчиненные) заполняется свойство Объект расширяемой конфигурации, которое автоматически устанавливается в контролируемое состояние. При переименовании расширяемого объекта, расширение не будет применено, т. к. изменилось имя объекта, которое является контролируемым свойством.

Однако в конфигураторе можно запустить проверку применимости расширения. В результатах проверки будет указана проблема отсутствия объекта с прежним наименованием и система предлагает несколько вариантов решения (которые можно применить для нескольких выделенных строк в окне применимости). В зависимости от ситуации, предлагаются различные комбинации из следующих вариантов решения проблемы:

- Переименовать, сохранив соответствие – объекту присваивается имя объекта из расширяемой конфигурации с внутренним идентификатором типа, ранее записанным в свойстве Объект расширяемой конфигурации, если такой существует и типы метаданных совпадают.

- Сохранить имя, изменив соответствие – свойству Объект расширяемой конфигурации устанавливается внутренний идентификатор объекта расширяемой конфигурации с тем же именем, что и у заимствованного объекта.

- Выбрать соответствие – в свойство Объект расширяемой конфигурации устанавливается внутренний идентификатор типа объекта конфигурации, выбранный вручную.

- Отключить проверку – отключается свойство Объект расширяемой конфигурации.

- Очистить соответствие – объекту задаётся служебное значение свойства Объект расширяемой конфигурации. Свойство с таким значением не обновляется при сохранении расширения.

- Установить значение из объекта конфигурации – свойству присваивается значение свойства расширяемого объекта.

- Удалить объект – объект удаляется из расширения.

Если для решения выбрано несколько проблем, то для выбора будут предложены только те варианты, которые применимы для всех выделенных проблем.

Для того чтобы комментировать объекты расширения, предназначено свойство Комментарий объекта расширения. Свойство не является многоязычным. Комментарий не используется в процессе применения расширения.

1.4. Создание собственного объекта

В расширении можно не только создавать заимствованные объекты, но и собственные. Создание и редактирование собственных объектов расширения ничем не отличается от аналогичных действий в расширяемой конфигурации. Контроль уникальности имен собственных объектов выполняется:

- При подключении расширения – среди имен расширяемой конфигурации и всех расширений;
- При разработке расширения – среди имен расширяемой конфигурации и того расширения, в котором создается собственный объект.

Если в расширяемой конфигурации, например, существует обработка ПоискВДанных, то ни в одном расширении не может существовать одноименной обработки.

При создании собственного объекта, этот объект получает префикс, который задается при создании расширения.

В связи с тем, что список расширений является единым и для доступа из конфигуратора и для доступа из «1С:Предприятия», система не позволяет выполнять взаимоисключающие действия. Другими словами, если в Конфигураторе открыто окно с расширением, то его невозможно удалить из списка расширений в режиме «1С:Предприятия». Расширение, во время записи в информационную базу в режиме «1С:Предприятия», в конфигураторе является недоступным для любых операций с ним.

Разработку расширения можно выполнять с использованием хранилища конфигурации. Каждое расширение считается отдельной конфигурацией и для него создается отдельное хранилище конфигурации. Не поддерживается:

- Хранение расширений в том же хранилище конфигурации, что и расширяемая конфигурация.
- Хранение в одном хранилище конфигурации нескольких расширений.
- Возможность подключения расширяемой конфигурации к хранилищу расширения и наоборот.

В общем, работа с хранилищем конфигурации для расширения ничем не отличается от работы с хранилищем конфигурации для расширяемой конфигурации. Разработчику предоставляется возможность захватывать объекты, помещать их в хранилище, формировать отчеты по хранилищу, сравнивать объекты из основной конфигурации расширения и хранилища и т. д.

Команды работы с расширением (рис. 8) расположены в меню Конфигурация – Хранилище конфигурации окна Расширения конфигурации.

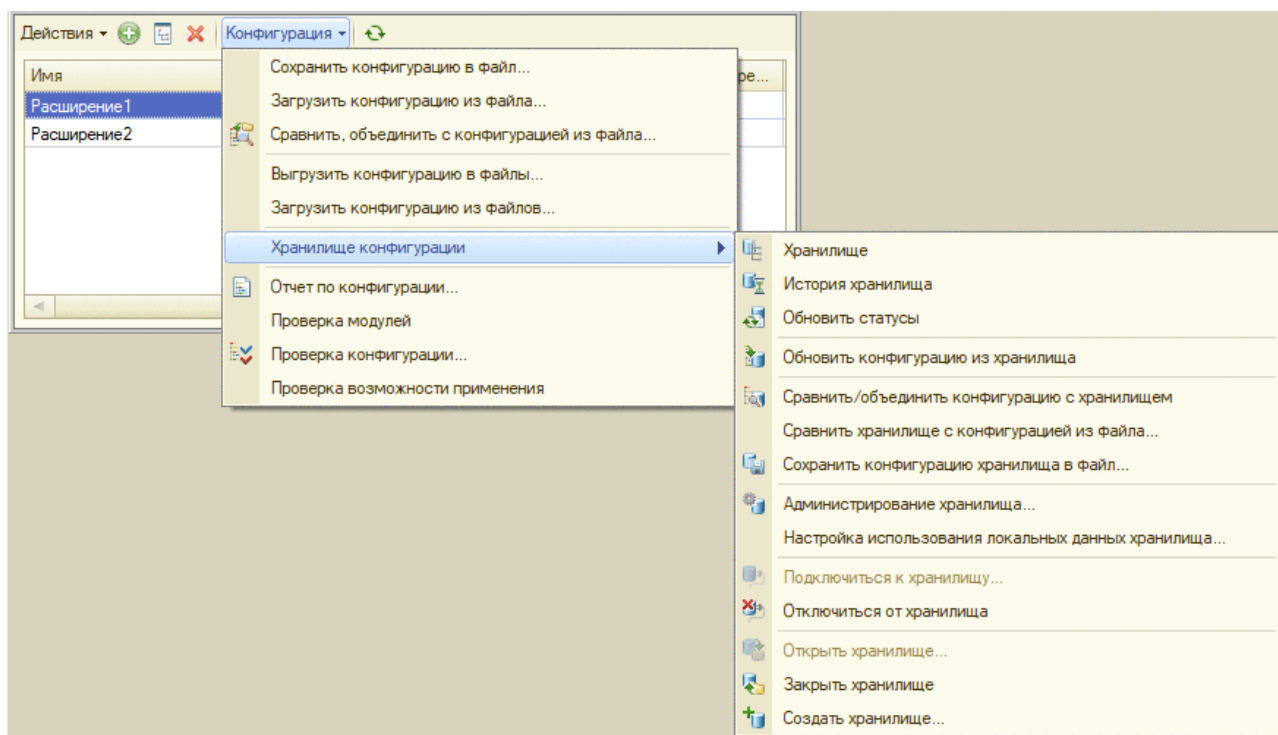


Рис. 8. Доступ к командам работы с хранилищем конфигурации для расширения

В окне с деревом метаданных расширения, команды работы с хранилищем доступны в контекстном меню, аналогично дереву основной конфигурации.

Расширение конфигурации обладает набором свойств, подобным свойствам расширяемой конфигурации. Но существуют и специальные свойства, которые имеются только у расширения конфигурации или поведение которых отличается от поведения в расширяемой конфигурации. Описание свойств Назначение расширения конфигурации и Префикс имени приведено в кратком описании механизма. Свойство Поддерживать соответствие объектам расширяемой конфигурации по внутренним идентификаторам предназначено для облегчения разработчику расширения поиска проблем, возникающих при переименовании объекта в расширяемой конфигурации.

Для описания расширения предоставляются возможности, аналогичные расширяемой конфигурации. Для этого у расширения существуют набор свойств:

- Краткая информация и Подробная информация позволяют задать описание расширения.
- Авторские права позволяет указать информацию о разработчике расширения.
- Адрес информации о поставщике позволяет задать URL, перейдя по которому можно будет получить более подробную информацию о поставщике расширения.
- Адрес информации о конфигурации позволяет задать URL, перейдя по которому можно получить более подробную информацию о данном расширении.

Также имеется возможность заменить следующие картинки основной конфигурации: картинка основного раздела, логотип и заставка. При этом надо помнить о следующих особенностях такой замены:

- Картинки из конфигурации полностью замещаются картинками из расширения.

- Если несколько расширений замещают картинки, то будет использована одна картинка для каждого свойства. Такой картинкой будет картинка из последнего примененного расширения.

Расширение конфигурации предоставляет возможность изменить оформление расширяемой конфигурации. Для этого имеется возможность расширить свойство Основной стиль. При этом реализована возможность как расширять существующие стили и элементы стиля, так и создавать собственные стили и элементы стиля в расширении. При расширении элементов стиля поддерживается только изменение значения цвета, но не вида элемента стиля.

Свойство Основные роли позволяет указать роли, которые будут определять доступность объектов расширения в том случае, если в расширяемой конфигурации отсутствуют пользователи. Свойство аналогично свойству Основные роли расширяемой конфигурации. Для роли, которая добавляется в качестве основной роли расширения, действует следующий набор ограничений:

- Роль не должна быть заимствованной. В качестве основной роли может выступать только собственная роль расширения.

- Роль не может предоставлять доступ к заимствованным объектам расширения. Основная роль расширения управляет доступом только к собственным объектам расширения.

Свойство Режим совместимости расширения конфигурации по своему назначению аналогично свойству Режим совместимости расширяемой конфигурации. Данное свойство управляет возможностями, которые предоставляет система в расширении конфигурации. Если при редактировании расширения используется какая-либо возможность, которая несовместима с установленным значением свойства Режим совместимости расширения конфигурации, то сохранение конфигурации информационной базы будет невозможно, а при попытке выполнить эту операцию - будет выдано диагностическое сообщение с объяснением причины невозможности выполнения операции.

1.5. Программные модули

Система позволяет расширить практически любой программный модуль, относящийся к управляемому приложению. Невозможно расширять глобальные серверные модули. Также следует помнить, что расширение привилегированного общего модуля будет выполняться в непривилегированном режиме (если иное не разрешено профилем безопасности). Кроме того, предоставляется возможность создавать собственные общие модули, которые не могут быть привилегированными и глобальными серверными. Работа с собственным общим модулем в расширении ничем не отличается от работы с общим модулем обычной конфигурации.

Работа с расширенным программным модулем имеет существенные особенности, которые будут рассмотрены в данном разделе.

При разработке расширяющего модуля следует учитывать, что этот модуль будет находиться в одном пространстве имен с расширяемым модулем. В связи с этим, из расширяющего модуля имеется возможность использовать расширяемый контекст (переменные и методы) напрямую. Также следует помнить, что созданные в расширении экспортные методы и переменные автоматически попадают в публичный контекст расширяемого модуля.

В расширяющем модуле имеется возможность:

1. Создавать собственные методы и переменные (если это допускает расширяемый модуль).

2. Назначать собственные обработчики на события, которые не обрабатываются в расширяемой конфигурации.

3. Перехватывать любой метод расширяемого модуля (в том числе и методы обработчиков событий), при этом имеется возможность:

- полностью заменить оригинальный метод собственным;
- создать методы, которые будут вызваны перед или после расширяемого метода.

Создание собственного метода или переменной в расширяющем модуле ничем не отличается от создания метода или переменной в расширяемой кон-

фигурации, при работе без расширений. Однако при создании собственных методов или переменных рекомендуется предварять их имена префиксом, который позволит однозначно идентифицировать принадлежность метода или переменной тому или иному расширению, а также избежать конфликта имен с расширяемой конфигурацией.

В расширении нельзя расширить тело модуля и область переменных, но можно определить свои переменные и написать свое тело модуля. Тело модуля расширения будет выполняться после модуля расширяемой конфигурации.

Перехват методов расширяемой конфигурации (включая назначение обработчиков) реализуется с помощью специального механизма аннотаций. Имеется несколько различных возможности перехвата вызова метода, которые позволяет реализовать практически любую схему исполнения расширяемого и расширяющего программного кода.

В общем случае, аннотированный метод из программного модуля расширения выглядит следующим образом:

```
&Аннотация("ИмяМетода")
Процедура Префикс_Имя()
    // текст метода
КонецПроцедуры
```

В данном примере Аннотация – одна из возможных аннотаций, ИмяМетода – имя расширяемого метода, Префикс_Имя – имя расширяющего метода. Префикс_ – это префикс расширения, который желательно указывать для всех методов расширения. Расширяться может как процедура, так и функция. Имя расширяющего метода должно подчиняться обычным правилам формирования имен процедур и функций. В частности, оно должно быть уникально.

Если расширяемый метод содержит какие-либо параметры, то:

1. Все расширяющие методы обязаны иметь в точности такое же описание, как и расширяемый метод, с точностью до ключевых слов `Знач` в описаниях параметров методов.

2. Значения параметров разделяются между всеми расширяющими методами из всех расширений и собственно расширяемым методом. Это значит, что, если какой-либо расширяющий метод изменит значение параметра, все методы, которые получают управление **после** этого метода, получают измененное значение этого параметра.

3. В расширяющем методе не имеет смысла указывать значения по умолчанию для параметров расширяемого метода. Значения по умолчанию будут определяться из описания расширяемого метода.

Также следует отметить, что если расширяется обработчик события, то расширяющий метод выполнится до того, как сработают подписки на «расширяемое» событие.

Если программный модуль расширен, то перед исполнением результирующего программного кода (расширяемая конфигурация и все подключенные расширения), выполняются следующие действия:

1. Выполняется синтаксический контроль модуля расширяемой конфигурации. Если во время этого контроля обнаруживается ошибка, пользователь получает сообщение об ошибке и текущее действие прерывается.

2. Последовательно выполняется синтаксический контроль расширяющих модулей из всех подключенных расширений. Если во время этого контроля обнаруживается ошибка (например, в расширяющем модуле используется метод, отсутствующий в расширяемой конфигурации), то пользователь получает сообщение об ошибке и текущее действие прерывается.

3. Выполняется проверка применимости расширяющих модулей: для каждого расширяющего метода ищется расширяемый метод, проверяется соответствие имен и состав параметров. Если проверка применимости не выполняется, то расширение конкретного метода с помощью конкретного расширения не выполняется, пользователю выводится предупреждение.

4. Если все проверки выполнены успешно - все программные модули компилируются и объект, в состав которого входят эти модули, готов к использованию.

Из вышеперечисленной последовательности действий следуют два вывода:

1) если синтаксический контроль обнаруживает проблемы для модулей, которые выполняются при старте прикладного решения (модуль приложения, модуль сеанса), то такие ошибки не позволят начать работу прикладному решению;

2) если синтаксический контроль обнаруживает проблемы при инициализации какого-либо объекта (например, при попытке открыть форму какого-либо справочника), то объект будет недоступен для использования, но работа прикладного решения не будет прервана. Проверка будет выполняться при каждой попытке использования любого объекта конфигурации.

1.6. Способы расширения

Если метод аннотирован таким образом:

Исполнение до расширяемого метода (аннотация Перед)

это означает, что вначале будет выполнен метод расширения, а затем - расширяемый метод (рис. 9).

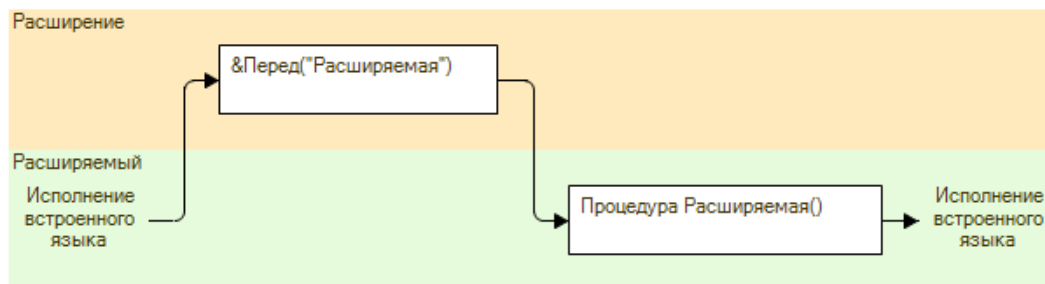


Рис. 9. Схема исполнения «Перед»

Если метод аннотирован таким образом:

Исполнение после расширяемого метода (аннотация После)

это означает, что вначале будет выполнен расширяемый метод, а затем - метод расширения (рис. 10).

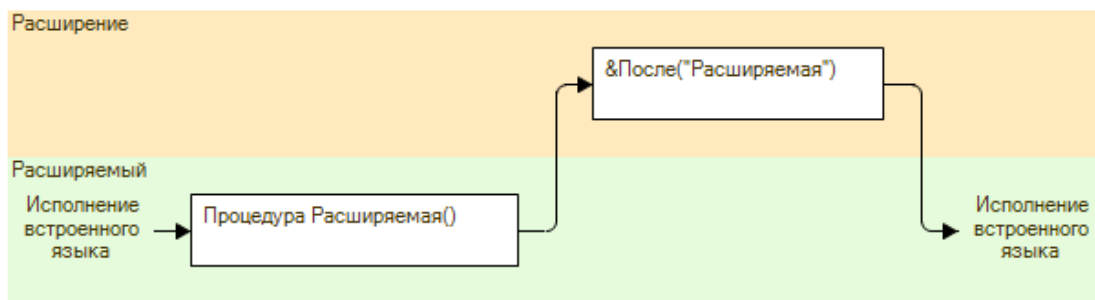


Рис. 10. Схема исполнения «После»

Если в расширяющем модуле созданы расширяющие методы, которые отмечены аннотациями Перед и После, то это означает, что вначале будет вызван метод, который отмечен аннотацией Перед, затем расширяемый метод и затем метод расширения, отмеченный аннотацией После (рис. 11).

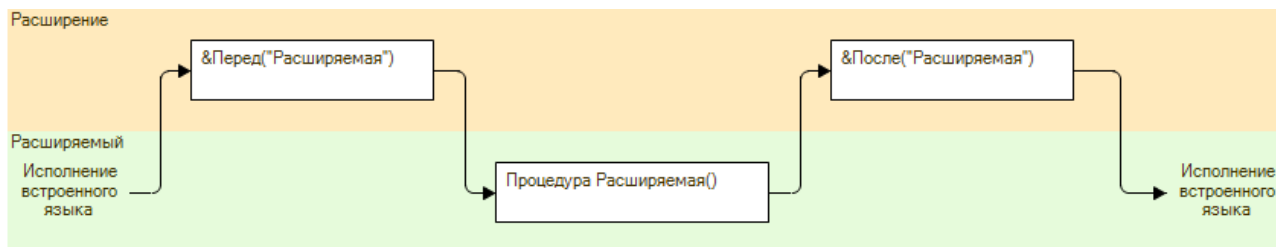


Рис. 11. Схема исполнения «Перед» и «После»

Под перехватом вызова метода понимается ситуация, когда метод, созданный в расширении, полностью замещает собой расширяемый метод. Другими словами, вызов метода расширяемой конфигурации приведет к исполнению метода расширения. Имя замещаемого метода указывается в качестве параметра аннотации. Рекомендуется использовать данный способ только в том случае, когда нет возможности использовать другие способы расширения методов.

Для расширяемых функций возможно применение только этого способа расширения (рис. 12).

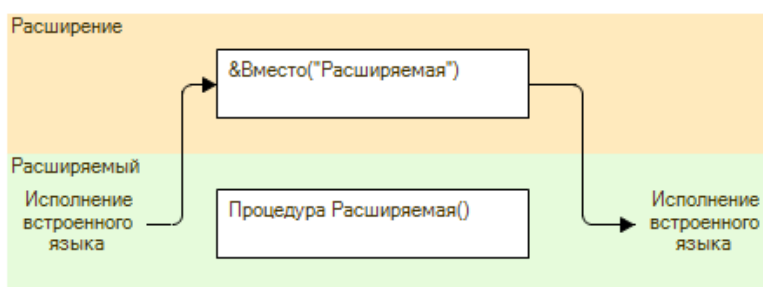


Рис. 12. Схема исполнения «Вместо»

Чтобы иметь возможность модифицировать результат работы расширяемого метода, предусмотрен метод глобального контекста ПродолжитьВызов(). В качестве параметров метода должны быть указаны фактические параметры, которые переданы в расширяющий метод (рис. 13).

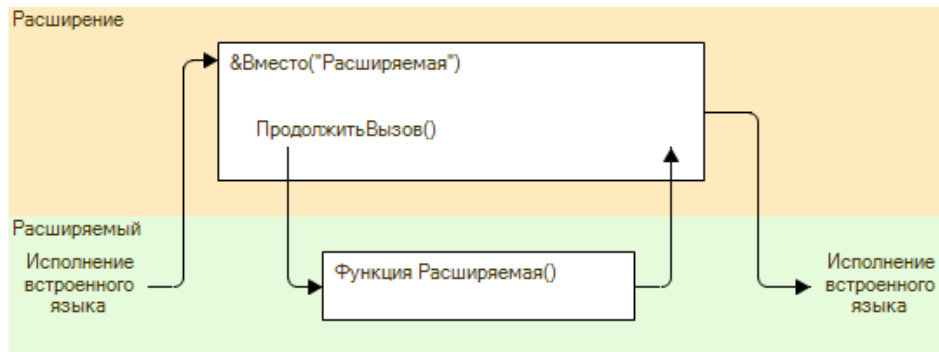


Рис. 13. Схема исполнения «Вместо» совместно с ПродолжитьВызов()

В исходном тексте пример будет выглядеть следующим образом:

```
&Вместо("Расширяемая")
```

```
Процедура Расш1_Расширяющая(Парам1, Парам2)
```

```
    // текст метода
```

```
    ПродолжитьВызов(Парам1, Парам2);
```

```
    // текст метода
```

```
КонецПроцедуры
```

Данный способ расширения позволяет точно изменять текст расширяемого модуля. Во время заимствования весь модуль копируется в расширение. Затем, с помощью директив препроцессора #Вставка ... #КонецВставки и #Удаление ... #КонецУдаления отмечаются фрагменты текста, которые необходимо добавить в расширяемый модуль и, соответственно, удалить из него.

Рассмотрим пример. Допустим, у нас есть метод возведения в степень простым умножением. При создании метода мы допустили ошибку и вместо умножения написали операцию сложения. Получился следующий метод:

```
Функция ВозведениеВСтепень(Основание, Степень) Экспорт
```

```
    Результат = 1;
```

```
    Для Индекс = 1 По Степень Цикл
```

Результат = Результат + Основание;

КонецЦикла;

Возврат Результат;

КонецФункции

Ошибку обнаружили уже «на месте». Для исправления ошибки можно использовать аннотацию ИзменениеИКонтроль. Исправленный метод в расширении будет выглядеть следующим образом:

```
&ИзменениеИКонтроль("ВозведениеВСтепень")
```

```
Функция Расш1_ВозведениеВСтепень(Основание, Степень)
```

```
    Результат = 1;
```

```
    Для Индекс = 1 По Степень Цикл
```

```
#Удаление
```

```
    Результат = Результат + Основание;
```

```
#КонецУдаления
```

```
#Вставка
```

```
    Результат = Результат * Основание;
```

```
#КонецВставки
```

```
    КонецЦикла;
```

```
    Возврат Результат;
```

```
КонецФункции
```

Как только разработчик конфигурации исправит ошибку (и расширяемая конфигурация будет обновлена), наше расширение перестанет выполняться с диагностикой следующего вида:

Расширение1: Обычная: Ошибка применения модуля "Расширение1 ОбщийМодуль.ОбщийМодуль1.Модуль". Текст модуля для метода "ВозведениеВСтепень" изменился.

Во время проверки применимости игнорируются пробелы и символы табуляции в начале и конце каждой строки метода, а также пустые строки (или строки, состоящие только из символов пробел и табуляция) в теле метода.

Если проверка применимости расширений выполняется в конфигураторе, то результатом проверки будет сообщение об ошибке и возможность восстановить соответствие с методом конфигурации (рис. 14).

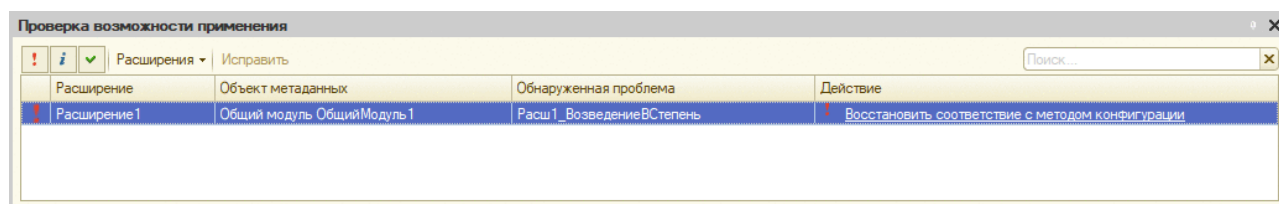


Рис. 14. Ошибка применимости для аннотации «ИзменениеИКонтроль»

Для восстановления следует нажать гиперссылку в колонке Действие. В результате будет открыта форма трехстороннего объединения модуля. В объединении будут участвовать:

- Оригинальный текст метода, который был в расширяемой конфигурации. Файл будет иметь имя `ИмяМетод_RESTORED_`.
- Текст метода из расширения (с директивами препроцессора). Файл будет иметь имя `ПрефиксРасширенияИмяМетода`.
- Текст метода, который сейчас находится в расширяемой конфигурации. Файл будет иметь имя `ИмяМетод`.

Результат объединения будет помещен в расширение. Если в конфигурации не настроено использование внешней программы для трехстороннего объединения – восстановить соответствие с методом конфигурации будет невозможно. Внешняя программа, необходимая для объединения, указывается в свойстве Режим "Объединить с помощью внешней программы" в диалоге настройки параметров конфигуратора (Главное меню – Сервис – Параметры – Сравнение/объединение – группа Модули – группа Обновление конфигурации на поддержке и актуализация расширения конфигурации). Это свойство не должно быть установлено в значение Не использовать.

Теперь более подробно рассмотрим работу такого способа расширения.

При заимствовании метод полностью копируется в расширение, но помечается аннотацией. Затем необходимо отметить код, который удаляется из расширения, директивами препроцессора `#Удаление ... #КонецУдаления` (нельзя

просто удалить «неправильный» код из модуля в расширении). Это позволит системе контролировать эквивалентность расширяемого метода и метода в расширении. Тот код, который необходимо вставить в расширяемый метод, необходимо помещать внутрь директив препроцессора #Вставка ... #КонецВставки (недостаточно просто вставить «правильный» код в нужные места метода) (рис. 15).

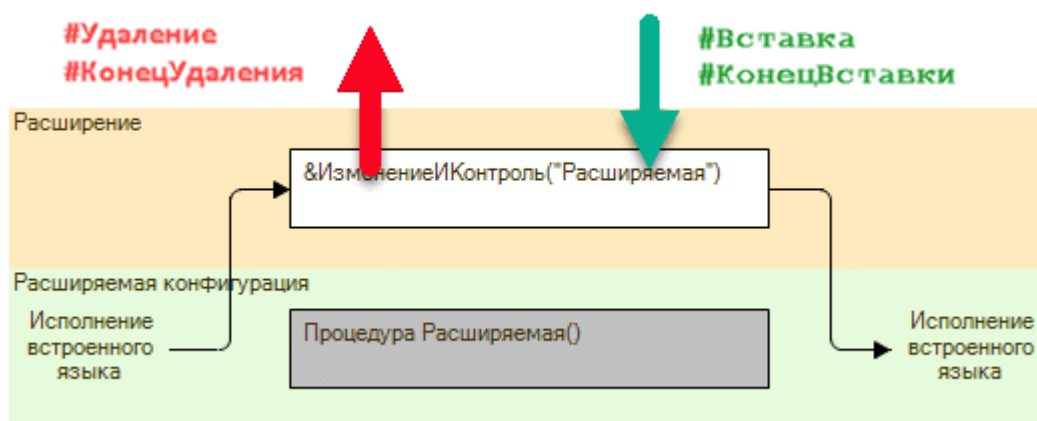


Рис. 15. Схема исполнения «ИзменениеИКонтроль»

С точки зрения работы конфигурации можно сказать, что метод, расширенный с помощью аннотации `ИзменениеИКонтроль`, полностью заменяет собой расширяемый метод конфигурации. Остальные директивы будут вызываться относительно исправленного метода. Таким образом, если в конфигурации используется еще одно расширение, где этот же метод (в примере выше – `ВозведениВСтепень()`) расширяется с помощью аннотации `Вместо` (и методом `ПродолжитьВызов()`), расширяться будет тот метод, который получится в результате модификации расширяемого метода из расширения с помощью аннотации `ИзменениеИКонтроль`.

Из этого следует одно очень важное замечание: каждый метод конфигурации может быть расширен с помощью аннотации `ИзменениеИКонтроль` ровно один раз. Если в конфигурацию подключается еще одно расширение, которое расширяет тот же метод с помощью аннотации `ИзменениеИКонтроль`, то такое расширение не пройдет проверку применимости.

1.7. Редактирование программного модуля

В общем случае, редактирование программного модуля в расширении мало отличается от редактирования модуля в расширяемой конфигурации или модуля в конфигурации, где вообще нет расширений. В качестве основных отличий можно указать процесс заимствования метода из какого-либо программного модуля расширяемой конфигурации.

Для того чтобы добавить какой-либо метод в расширение, необходимо поместить курсор в требуемый метод (включая строку с именем метода) и выбрать команду Добавить в расширение. Если при выполнении данной команды в конфигураторе открыто одно расширение, то именно это расширение будет использовано. Во всех остальных случаях будет предложен выбор из расширений, которые добавлены для данной информационной базы.

При выполнении команды Добавить в расширение происходит следующее:

1. Если объект, из модуля которого добавляется метод, отсутствует в выбранном расширении – этот объект автоматически добавляется в расширение.

2. Если расширяется метод из модуля, отличного от модуля формы:

- Разработчику предлагается выбрать аннотацию для метода (Перед, После, Вместо или ИзменениеИКонтроль). При этом полужирным шрифтом выделяются те расширения метода, которые уже существуют в расширении.

- Если выбран существующий метод в расширении, то выполняется переход к этому методу. При этом возможна корректировка объявления метода, если оно стало отличаться от расширяемого метода.

- Если выбран несуществующий способ расширения, то будет создан новый метод в расширении, который будет предваряться соответствующей аннотацией.

- Для функций недоступны аннотации Перед и После.

- Для процедуры недоступно:

- Аннотации Вместо и ИзменениеИКонтроль, если уже существуют методы с аннотациями Перед или После;

- Другая аннотация, если уже существует метод с аннотацией Вместо.

3. Если расширяется метод из модуля формы:

- Если расширяемый метод является обработчиком одного события или одной команды, то предлагается выбрать, каким образом выполнить расширение выбранного метода: как расширение для обработчика события/команды или как расширение обычного метода.

- Если расширяемый метод является обработчиком для нескольких событий или команд, то будет сформировано предупреждение о том, что данный метод не может быть расширен как обработчик события/команды, и он будет расширен только как обычный метод. Затем будет предложено выбрать способ расширения метода.

- Для обычного метода, не являющегося обработчиком какого-либо события, будет выполнено расширение метода с использованием аннотаций.

Следует учитывать, что если расширяемый метод обрамлен инструкциями препроцессора, то эти инструкции не будут перенесены в расширение.

При проверке соответствия описания расширяемого метода и метода расширения проверяются следующие характеристики методов:

- Количество параметров и признак передачи параметров «по значению» (ключевое слово Знач).

- Метод является процедурой или функцией. Если определение метода изменяется с процедуры на функцию, а до этого процедура была расширена с применением аннотаций Перед/После, то аннотация будет заменена на аннотацию Вместо.

- Если в модуле расширения присутствуют несколько методов с одинаковым именем, но которые обрамлены разными инструкциями препроцессора, то в расширении будет обновляться всегда первый из перечня таких методов.

Еще одной особенностью редактирования программного модуля в расширении является то, что имеется возможность перехода к расширяемому методу

непосредственно из расширения. Для этого следует поместить курсор на имя расширяемого метода в аннотации, а затем воспользоваться стандартной командой редактора **Перейти к определению**.

1.8. Взаимодействие нескольких расширений

При разработке расширений следует исходить из следующих предположений:

1. Расширение должно разрабатываться как автономный продукт, не опирающийся на наличие или отсутствие других расширений.

2. Одновременно может быть подключено более одного расширения, которое расширяет один и тот же объект расширяемой конфигурации.

3. Порядок исполнения программных модулей в результирующей конфигурации определяется порядком регистрации расширений в информационной базе расширяемой системы. Первым используется расширение, которое зарегистрировано последним.

4. Необработанное исключение, возникающее в любом из расширений (или расширяемой конфигурации), прерывает исполнение всей цепочки методов расширений и распространяется в расширяемой конфигурации.

При рассмотрении вопроса взаимодействия расширений и расширяемой конфигурации будет использоваться пример из двух расширений (Расширение1 и Расширение2), зарегистрированных в расширяемой системе в указанном порядке. При этом чем раньше зарегистрировано расширение, тем меньше расширений находится между этим расширением и расширяемой конфигурацией. Так, Расширение1 будет расширять непосредственно расширяемую конфигурацию, а Расширение2 будет расширять конфигурацию, которая является объединением Расширения1 и оригинальной расширяемой конфигурации. Таким образом, в самом низу этой слоистой конструкции расположена расширяемая конфигурация, откуда всегда начинается исполнение встроенного языка.

Далее будут рассмотрены несколько примеров работы результирующей конфигурации при различных способах расширения.

Пример 1.

Если оба расширения одинаково обрамляют (аннотации Перед и После) расширяемый метод, то схема исполнения встроенного языка будет выглядеть следующим образом (рис. 16):

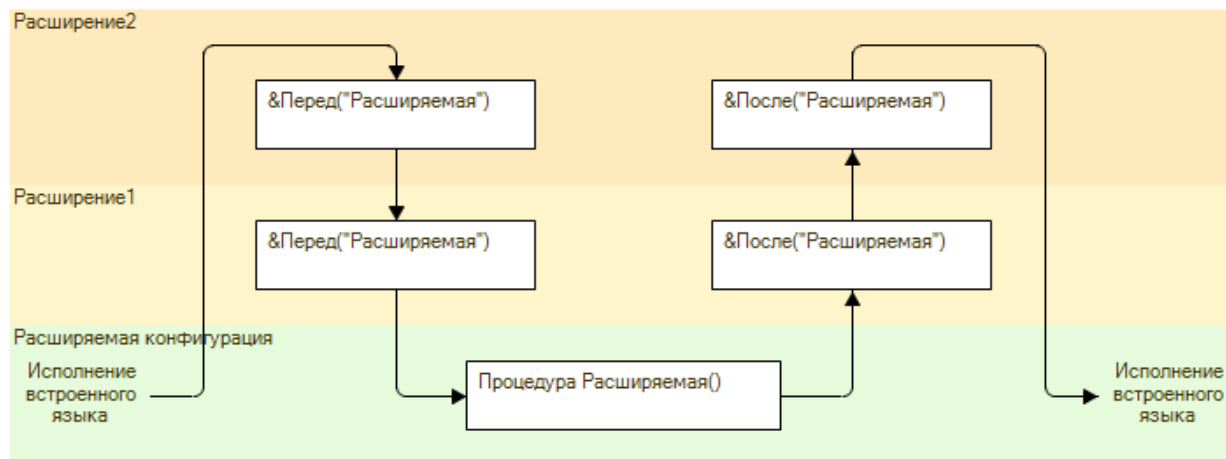


Рис. 16. Схема взаимодействия. Пример 1

При попытке вызвать метод Расширяемая() (в основной конфигурации), встроенный язык будет исполняться в следующем порядке:

1. Будет вызван метод, отмеченный аннотацией Перед("Расширяемая") из Расширения2.
2. Затем будет вызван метод, отмеченный аннотацией Перед ("Расширяемая") из Расширения1.
3. Затем будет вызван метод Расширяемая() из расширяемой конфигурации.
4. Затем, в обратном порядке (относительно списка расширений, т. е. Расширение1 и Расширение2), будут вызваны методы, отмеченные аннотациями После("Расширяемая").

То есть вначале вызываются те методы расширений, которые разработчик отметил как вызываемые **до** исполнения расширяемого метода, затем исполняется собственно расширяемый метод, а затем вызываются те методы, которые разработчик отметил как вызываемые **после** исполнения расширяемого метода.

Пример 2.

Если в каждом расширении расширяемый метод перехватывается полностью, то схема выполнения будет выглядеть следующим образом (рис. 17):

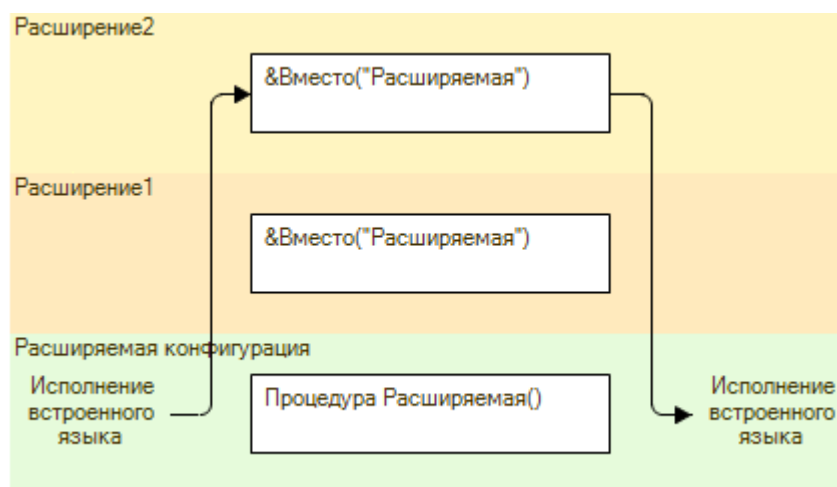


Рис. 17. Схема взаимодействия. Пример 2

Так произошло потому, что метод, отмеченный аннотацией `Вместо`, полностью заменяет собой расширяемый метод. Следует особо отметить, что если в этом примере поменять порядок регистрации расширений (первым будет `Расширение2`, а последним – `Расширение1`), то единственным выполнившимся методом будет метод из `Расширение1`.

В связи с таким поведением «замещающего» метода расширения, рекомендуется не допускать ситуаций, когда какой-то метод расширяемой конфигурации расширяется с замещением (аннотация `Вместо`) более чем в одном одновременно работающем расширении. Другими словами, если в расширении, в модуле документа `Документ1` замещается обработчик события `ОбработкаПроведения`, то рекомендуется не допускать ситуаций, когда в информационной базе работает еще хотя бы одно расширение, которое каким-либо образом перехватывает обработчик события `ОбработкаПроведения` в документе `Документ1`. Однако, если в «замещающем» методе расширения есть **безусловный** вызов метода `ПродолжитьВызов()`, то одновременное существование нескольких расширений с таким способом расширения вполне допустимо.

Факт того, что расширение успешно подключено к расширяемой конфигурации, не означает, что все расширяющие методы, которые находятся в расширении, будут выполняться. Работоспособность методов расширения может быть изменена настройками профиля безопасности, который использует информационная база.

Если прикладное решение работает в файловом варианте или в клиент-серверном варианте без профилей безопасности, то при подключении расширения:

- В обычном режиме исполнения встроенного языка - допустимо расширять как клиентские, так и серверные методы. Без ограничения места расположения методов.

- В безопасном режиме исполнения встроенного языка - будут расширяться только клиентские методы и серверные обработчики форм, которые установлены через панель свойств. К остальным серверным методам (аннотированные серверные методы модулей формы и серверные общие модули) расширение применяться не будет.

Возможность расширения серверных общих модулей в клиент-серверном варианте регулируется профилем безопасности.

ГЛАВА 2. ФОРМА

2.1. Заимствование формы

При заимствовании формы происходит заимствование только элементов формы. Реквизиты, команды и параметры формы отображаются в редакторе формы расширения, но не являются фактически заимствованными. Заимствованная форма выглядит точно так, как эта форма выглядит в расширяемой конфигурации в момент добавления ее в расширение. Модуль заимствованной формы не содержит никаких методов. В момент добавления конфигуратор «запоминает» добавляемую форму в расширении. В дальнейшем такая форма будет называться **сохраненной**. До тех пор, пока разработчик не изменит свойства элемента заимствованной формы, отображение элемента определяются расширяемой конфигурацией. Для того чтобы изменить свойства элемента в заимствованной форме, никаких специальных действий предпринимать не требуется. Однако, для того чтобы изменить свойства реквизита, команды или параметра формы - необходимо соответствующий элемент добавить в расширение (аналогично любому другому элементу конфигурации).

Логика обновления компонентов формы следующая:

- Если реквизит, параметр или команда не добавлены в расширение, то изменения свойств этих компонентов, выполненные в расширяемой конфигурации, автоматически отражаются в расширении.
- Реквизиты, параметры и команды, которые добавлены в расширяемой конфигурации, автоматически отражаются в расширении.
- Если в расширяемой конфигурации изменены свойства элементов формы, то такие изменения будут отражены в расширении только после обновления сохраненной формы. При этом результирующая форма (в режиме «1С:Предприятие») будет сформирована с **учетом** всех изменений, которые выполнены в расширяемой конфигурации.
- Реквизиты, параметры и команды, которые удалены в расширяемой конфигурации, будут автоматически удалены в расширении, не дожидаясь об-

новления сохраненной формы. Однако, если удаляемый компонент формы был добавлен в расширение, то в расширении он останется (до обновления сохраненной формы), но в результирующей форме (в режиме «1С:Предприятие») будет отсутствовать.

- В том случае, если изменяются свойства любого объекта формы, который добавлен в расширение, то системой будет выполнено трехстороннее сравнение для получения результирующего объекта.

Редактор формы отображает расширение редактируемой формы. Имя расширения отображается в квадратных скобках в заголовке редактора формы (рис. 18).

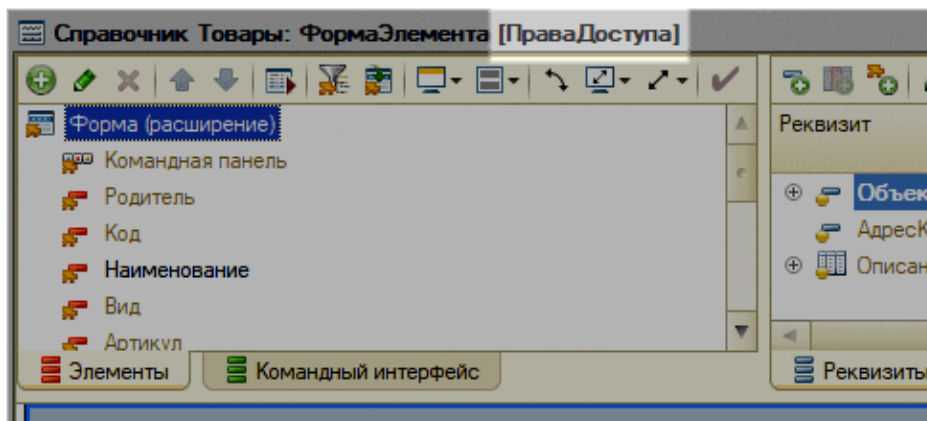


Рис. 18. Отображение имени расширения

Команду Обновить расширение формы (рис. 19) командной панели редактора формы следует использовать в том случае, если необходимо перенести в расширение изменения и дополнения, которые выполнены в расширяемой конфигурации после создания расширения формы.

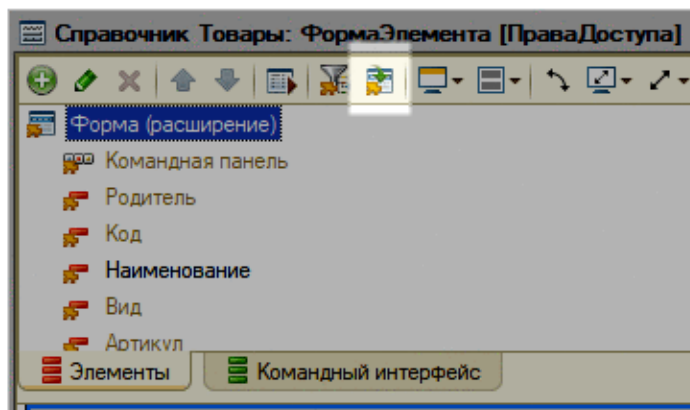


Рис. 19. Обновить расширение формы

Кнопка Показывать только измененные (рис. 20) командной панели редактора формы позволяет скрыть элементы формы, свойства которых не изменялись в данной форме (относительно расширяемой формы).

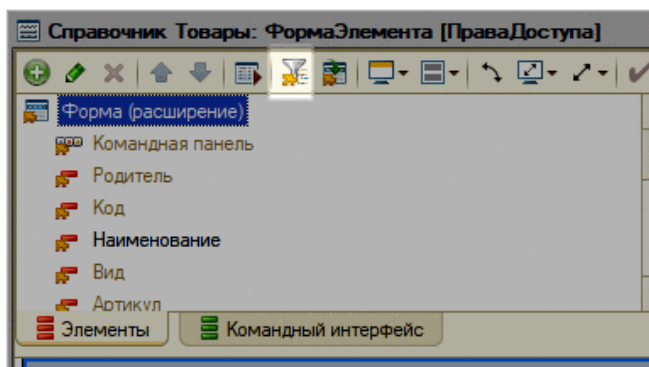


Рис. 20. Показать только измененные элементы

Если кнопка нажата, то в списке элементов формы отображаются только те элементы формы, свойства которых изменены в данной форме расширения. Если отжата – отображаются все элементы данной формы.

В том случае, когда в расширяемой форме будет изменен состав элементов формы, в редакторе расширяющей формы этот факт будет определен и пользователю будет предложено выполнить обновление формы в расширении (рис. 21).

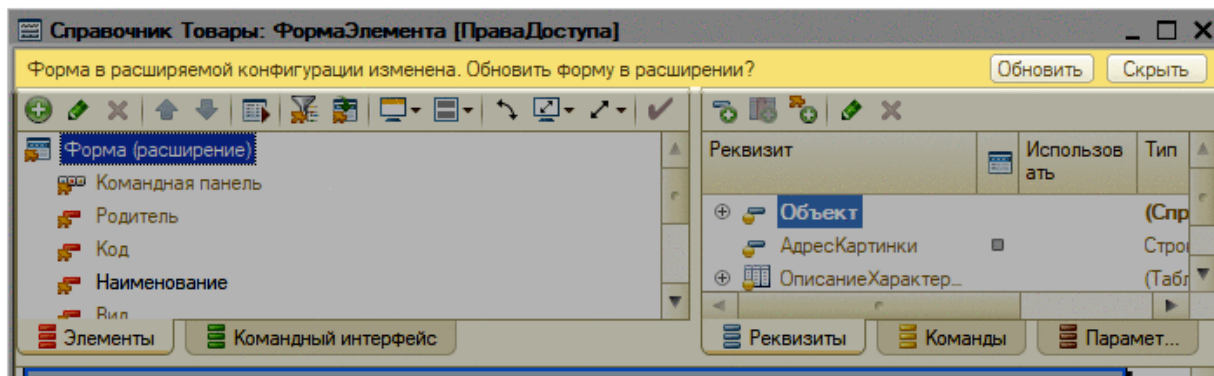


Рис. 21. Обновить форму в расширении

Такое предложение будет сделано как при первом открытии редактора формы расширения, так и в процессе работы (если форма расширяемой конфигурации будет изменена в своем окне). Аналогичное предложение будет сделано в том случае, если состав элементов формы будет изменен в самом расширении.

Отображение компонентов формы также позволяет определить, в каком состоянии находится компонент формы (рис. 22).

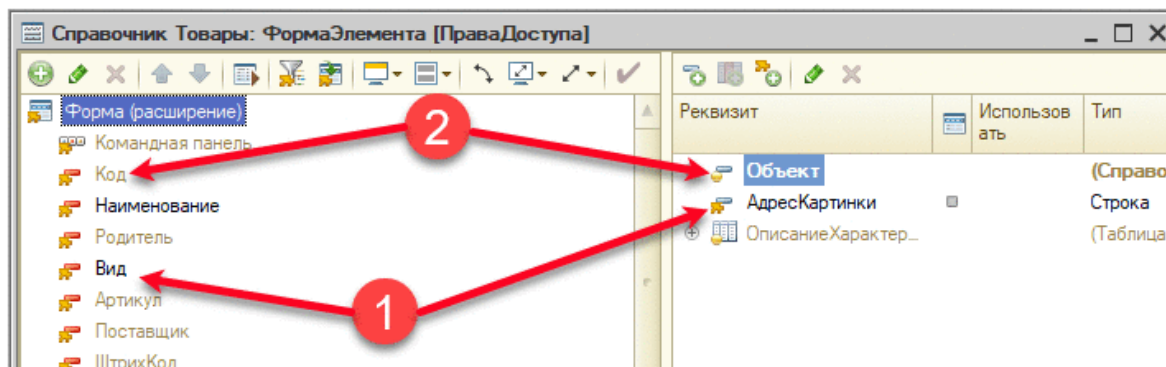


Рис. 22. Статусы компонентов

На рис.22 показаны различные состояния компонентов формы. Цифры на рисунке означают следующее:

- Цифра 1 – компоненты формы, заимствованные в расширение. Элементы формы всегда являются заимствованными. Реквизиты, команды и параметры необходимо заимствовать явным образом при необходимости.

- Цифра 2 – компоненты формы, которые не заимствованы в расширение. Эти компоненты выступают в виде «ссылок» на компоненты расширяемой формы. Чтобы изменить значения свойств таких компонентов, их надо предварительно добавить в расширение.

Для элемента формы, которые заимствованы в расширение, имеется возможность отобразить только те свойства, которые были изменены в расширении. Для этого в панели свойств заимствованного элемента формы существует кнопка Показывать только измененные (рис. 23).

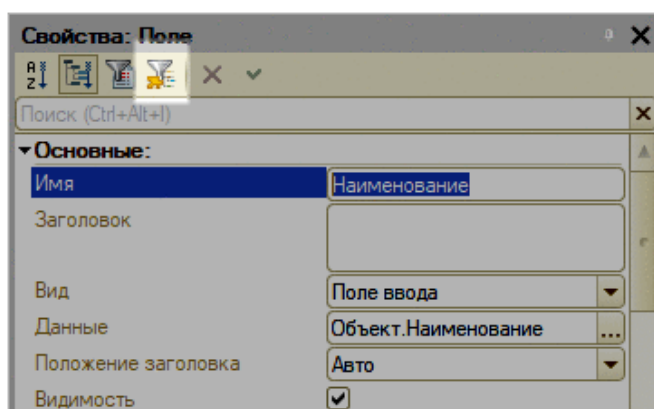


Рис. 23. Только измененные свойства элемента формы

При нажатой кнопке, панель свойств показывает только те свойства элемента формы, которые изменены в расширении. Если элемент формы не захвачен в расширении – кнопка отсутствует в командной панели.

Реквизиты, команды и параметры формы можно как добавить в расширение, так и «удалить» из него. «Удаление» выполняется нажатием кнопки Del на соответствующем компоненте формы. В этом случае компонент изменит свое состояние на то, которое отображается цифрой 2 на рис.22. При этом все изменения, которые были выполнены в расширении, будут утеряны. После «удаления» поведение возвращается к тому поведению, которое заложено в данном компоненте в расширяемой конфигурации. Это поведение не касается элементов формы. Не поддерживается удаление заимствованных элементов формы и возврат элемента формы к настройкам «как в расширяемой конфигурации».

Для того чтобы убрать элемент из результирующей формы, рекомендуется для такого элемента в расширении установить свойство Доступность в значение Ложь (сбросить флажок).

2.2. Работа с модулем формы

Расширение модуля формы несколько отличается от расширения любого другого модуля. В данном разделе описаны особенности расширения модулей в варианте модуля формы. Первой особенностью является то, что создавать обработчики в расширяющей форме следует с помощью панели свойств соответствующего компонента формы (форма, элемент, команда). При попытке создать обработчик, конфигуратор уточнит, когда должен быть вызван создаваемый обработчик и как он (обработчик) будет создаваться (рис. 24).

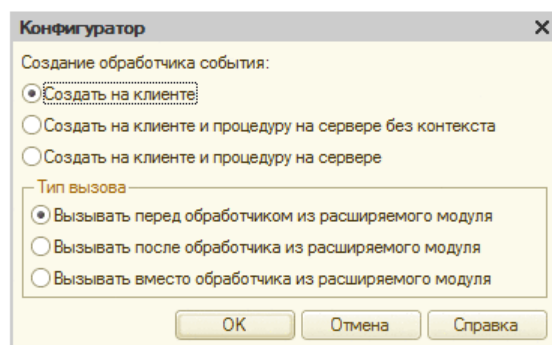


Рис. 24. Расширение обработчика

При этом в палитре свойств, рядом с именем расширяющего обработчика, появятся картинки, которая отображает тип расширения (рис. 25). Также в том случае, если выбран тип расширения Перед или После - автоматически будет показано еще одно поле ввода для указания парного расширяющего метода: если создан расширяющий метод с типом расширения Перед, то будет создано еще одно поле для расширяющего метода с типом расширения После и наоборот. Для замещающего обработчика (с типом расширения Вместо) подобного поведения не будет.

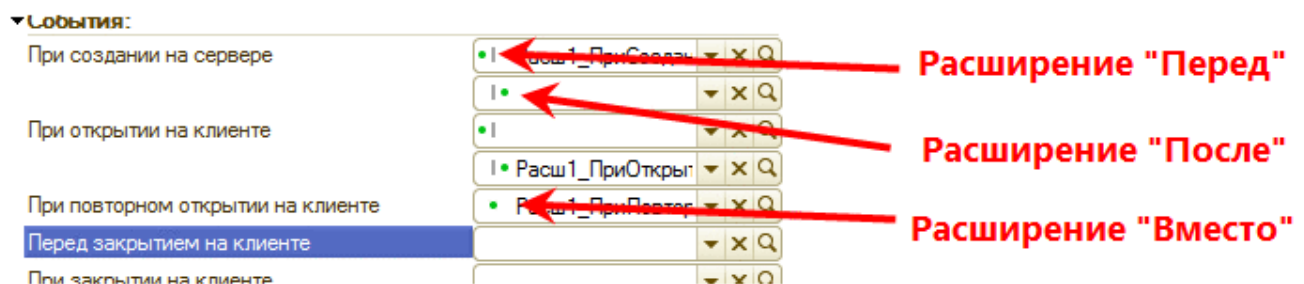


Рис. 25. Обозначения обработчиков

В том случае, если требуется изменить тип расширяющего обработчика – следует удалить существующий обработчик и создать его заново.

В коде обработчика события с типом расширения После запрещено менять значения параметров Отказ (для всех событий с таким формальным параметром), Выполнение (для события НачалоПеретаскивания) и ОтменаРедактирования (для события ПриОкончанииРедактирования). В обработчиках с типом расширения Перед не будут вызываться обработчики в следующих расширениях и обработчик в расширяемой форме в следующих случаях:

- установка параметра Отказ в значение Истина;
- установка параметра Выполнение в значение Ложь (для события Начало-Перетаскивания);
- установка параметра ОтменаРедактирования в значение Истина (для обработчиков событий ПриОкончанииРедактирования).

Следующей особенностью является то, что редактор управляемой формы создает расширяющие обработчики без использования аннотаций.

2.3. Построение результирующей формы

В режиме 1С:Предприятие для получения результирующего объекта происходит объединение расширяемого и всех расширяющих объектов. Процесс объединения работает следующим образом:

1. Выполняется объединение расширяемой конфигурации и первого (по порядку регистрации) расширения.

2. Объединение повторяется для каждого следующего зарегистрированного расширения, при этом в качестве расширяемой конфигурации выступает результат предыдущего объединения. С точки зрения каждого следующего объединения, результат предыдущего объединения будет считаться расширяемым объектом.

В процессе объединения каждой компоненты (элемент, реквизит, параметр или команда) управляемой формы анализируются изменения в расширяемой и расширяющей формах относительно сохраненной. По результатам анализа принимается решение, какая компонента будет отображаться в результирующей форме:

1. Если изменений нет – компонента берется из расширяемого объекта;

2. Если изменение есть только в расширяемой форме - компонента берется из расширяемой формы;

3. Если изменение есть только в расширяющей форме - компонента берется из расширяющей формы;

4. Если изменение есть и в расширяемой форме, и в расширяющих формах - компонента берется из расширяющей формы.

Сопоставление компонентов управляемой формы выполняется по имени соответствующего элемента.

Вышеописанный режим не действует:

1. Для условного оформления. В этом случае происходит объединение элементов условного оформления из расширяемой формы и всех расширяющих форм аналогично тому, как это происходит при объединении конфигураций.

2. Командный интерфейс результирующей формы формируется путем добавления элементов командного интерфейса из расширяющей формы к командному интерфейсу расширяемой формы. При этом возможна ситуация, когда на форме будут несколько команд с одинаковым представлением, т. к. они образованы командами из разных расширений, но с одинаковым представлением.

При определении порядка элементов результирующей управляемой формы вначале определяется возможность расположить элементы без конфликтов размещения. Если такая возможность есть, то порядок элементов результирующей формы определяется следующим образом:

- Элементы формы, добавленные в начало списка (не важно, в расширении или в расширяемой конфигурации), останутся в начале списка независимо от изменения порядка остальных элементов. Сначала будут идти элементы формы из расширений (в порядке следования), а потом из расширяемой конфигурации.

- Элементы формы, добавленные в конец списка (не важно, в расширении или в расширяемой конфигурации), останутся в конце списка независимо от изменения порядка остальных элементов. Сначала будут идти элементы формы из расширяемой конфигурации, потом из расширений (в порядке следования).

- Также существует возможность располагать рядом с элементом расширяемой формы другие элементы формы. Для этого следует создать в расширении группу, в которую поместить элемент расширяемой формы, и те элементы, которые необходимо рядом. Тогда при смене местоположения оригинального элемента в расширяемой форме, в результирующей форме на новом месте окажется и вся группа, в которую был помещен этот элемент.

Если без конфликтов порядок определить нельзя, используется следующий алгоритм:

- Исходным выступает список элементов расширения в порядке, который установлен в расширении.

- Для определения положения элемента, добавленного в расширяемой конфигурации, выполняется поиск элемента, существующего в сохраненной конфигурации и расположенного перед добавленным элементом (в порядке расширяемой конфигурации):

- В соответствии с порядком элементов в расширении, определяется, какой элемент является следующим - существовавший или добавленный в расширении:

- Если существовавший, то элемент основной конфигурации добавляется после найденного элемента.

- Если элемент или элементы, добавленные в расширении, то элемент расширяемой конфигурации добавляется после них.

- Если в расширяемой конфигурации было добавлено несколько элементов, то последующие элементы добавляются после предшествующих добавленных, даже если предшествующие уже были добавлены расширением.

- Если элемент, после которого элементы были добавлены, не найден в расширении (например, удален), то ищется предыдущий существующий в порядке основной конфигурации. Если при поиске достигнуто начало списка элементов, то новый элемент (или группа новых элементов) добавляется в начало результирующего списка, но после новых элементов расширения, добавленных в начало списка.

- Новые элементы расширяемой конфигурации, добавленные в начало списка, добавляются сразу после новых элементов расширения, добавленных в начало списка.

- Новые элементы расширяемой конфигурации, добавленные в конец списка, добавляются перед новыми элементами расширения, добавленными в конец списка.

- Если при добавлении оказывается, что элемент уже был добавлен расширением, то такой элемент пропускается.

- Удаляются элементы, которые были удалены в основной конфигурации по сравнению с сохраненной версией.

Командный интерфейс формы дополняется командными интерфейсами расширений. Команды, добавляемые в командный интерфейс формы вручную, будут отображаться после команд расширяемой конфигурации (в режиме «1С:Предприятие»). Следствием этого является то, что в результирующей форме могут одновременно существовать несколько команд с одинаковым представлением.

Удалить команду расширяемой конфигурации невозможно.

Необходимо учитывать, что в редакторе формы в конфигураторе, разработчик не будет видеть фрагменты командного интерфейса, которые предоставляются расширяемой конфигурацией.

Для объектов конфигурации имеется возможность изменять основные и вспомогательные формы заимствованных объектов конфигурации. Для этого необходимо создать в расширении форму, которой будет заменена существующая форма объекта конфигурации, затем отметить свойство конфигурации, описывающее заменяемую форму, как модифицируемый реквизит и указать новую форму в этом реквизите. Для того чтобы изменять свойства конфигурации, связанные с формами (свойства Основная форма отчета, Основная форма поиска и т. д.) нет необходимости заимствовать корневой элемент дерева конфигурации, достаточно работать со свойствами корневого элемента расширения.

Подсистема может быть как заимствованным, так и собственным объектом расширения.

Из состава заимствованной подсистемы нельзя удалить объекты конфигурации, которые добавлены туда в расширяемой конфигурации. Но в состав заимствованной подсистемы можно добавить другие объекты: как собственные, так и заимствованные (относительно расширения).

Собственная подсистема расширения может включать как собственные объекты расширения, так и заимствованные объекты.

Для заимствованных объектов имеется возможность изменить картинку подсистемы, с помощью изменения модифицируемого реквизита Картинка.

ГЛАВА 3. ПРАВА ДОСТУПА И РОЛИ

Рассматриваемое ниже поведение актуально только в том случае, если свойство расширения Режим совместимости расширения конфигурации установлено в значение Версия 8.3.14 и выше (включая Не использовать в версии 8.3.14).

С помощью расширения конфигурации предоставляется возможность расширить уже существующую роль, а также создать собственную роль в расширении.

В простейшем случае, когда роли не заданы как в расширяемой конфигурации, так и в расширении - пользователь имеет доступ не только ко всем объектам основной конфигурации, но и ко всем объектам расширения. Для расширения отсутствие ролей является, в некотором смысле, вырожденной ситуацией, т. к. при создании расширения автоматически создается роль, дающая права на все собственные объекты расширения.

При подключении расширения к информационной базе есть возможность установить признак Использовать основные роли для всех пользователей. Данный флажок работает следующим образом:

- Если флажок установлен и в расширении задано свойство ОсновныеРоли, то минимальный набор прав будет определяться ролями, указанными в свойстве ОсновныеРоли. При этом администратор имеет возможность расширить состав ролей конкретного пользователя другими ролями расширения (не входящими в состав основных ролей расширения).

- Если флажок сброшен и в расширении задано свойство ОсновныеРоли, то минимальный набор прав будет определяться ролями, указанными в свойстве ОсновныеРоли только в том случае, если в конфигурации, к которой подключено расширение, отсутствуют пользователи.

- Во всех остальных случаях права на объекты расширения будут определяться администратором системы, который должен предоставить каждому

пользователю доступ к объектам расширения (с помощью конфигуратора или других инструментов).

Нужно понимать, что если при подключении для расширения установлено свойство `Использовать основные роли для всех пользователей`, то лишить пользователя прав, предоставленных основными ролями, невозможно. Для установки свойства `Использовать основные роли для всех пользователей` необходимо иметь право администрирования расширений конфигурации.

Если в расширении отсутствуют собственные роли, то доступностью объектов расширения можно управлять с помощью ролей расширяемой конфигурации.

Доступность собственных объектов расширения в результирующей конфигурации определяется состоянием свойства `Устанавливать права для новых объектов роли расширяемой конфигурации`. Т.е. результирующая конфигурация «считает» все собственные объекты расширения своими новыми объектами. Таким образом, собственные объекты расширения будут доступны в тех собственных ролях, в которых будет установлено свойство роли `Устанавливать права для новых объектов`.

Флажок роли `Устанавливать права для реквизитов и табличных частей по умолчанию расширяемой конфигурации` обеспечивает доступность реквизитов и табличных частей в собственных объектах расширения. В ролях, где флажок установлен – в собственных объектах расширения будут все права на реквизиты и табличные части.

Исходя из вышесказанного, можно определить следующую общую схему работы с правами доступа:

- Заимствовать роль следует только в тех случаях, когда необходимо изменить права доступа (или ограничения доступа к данным) относительно расширяемой конфигурации.
- В расширении без собственных ролей, доступ к объекту расширения определяется настройками ролей расширяемой конфигурации.

- Собственные роли расширения определяют права для собственных и заимствованных объектов расширения. Собственная роль расширения не может изменить права для объектов расширяемой конфигурации, которые не добавлены в расширение.

3.1. Заимствованные роли

В заимствованной роли имеется возможность изменять права доступа, указанные для объектов расширяемой конфигурации. Изменять можно не только сами права доступа, но и ограничения доступа к данным. Также предоставляется возможность создавать (и использовать) шаблоны ограничения доступа к данным в заимствованном объекте роли. В заимствованной роли не предоставляет возможности изменять права доступа, которые задаются для корневого объекта конфигурации (административные права, возможность использования того или иного клиентского приложения и т. д.).

В заимствованной роли все права, по умолчанию, соответствуют правам из расширяемой конфигурации. В настройке прав доступа этому соответствует специальное значение флажка. Однако имеется возможность установить или сбросить любое право. Таким образом, с помощью заимствованной роли имеется возможность как расширить права пользователя, так и сузить их.

Ограничения доступа к данным, заданные в расширении, будут использоваться только в том случае, если для права доступа явно указаны права в заимствованной роли. В этом случае ограничения доступа к данным из расширения полностью заместят собой ограничения доступа к данным из расширяемой конфигурации (для данного права и данной роли). Если право доступа заимствуется из расширяемой конфигурации – ограничения доступа к данным, заданные в расширении, не будут использоваться.

При редактировании заимствованной роли, появляется возможность установить «третье состояние» для права доступа: использовать значение из расширяемой роли (рис. 26).

На данном рисунке цифрой 1 обозначены права, измененные в расширении, а цифрой 2 - права, задаваемые расширяемой ролью.

В заимствованной роли предоставляется возможность управлять автоматическим назначением прав только для собственных объектов расширения.

За это отвечает флажок Устанавливать права для новых собственных объектов.

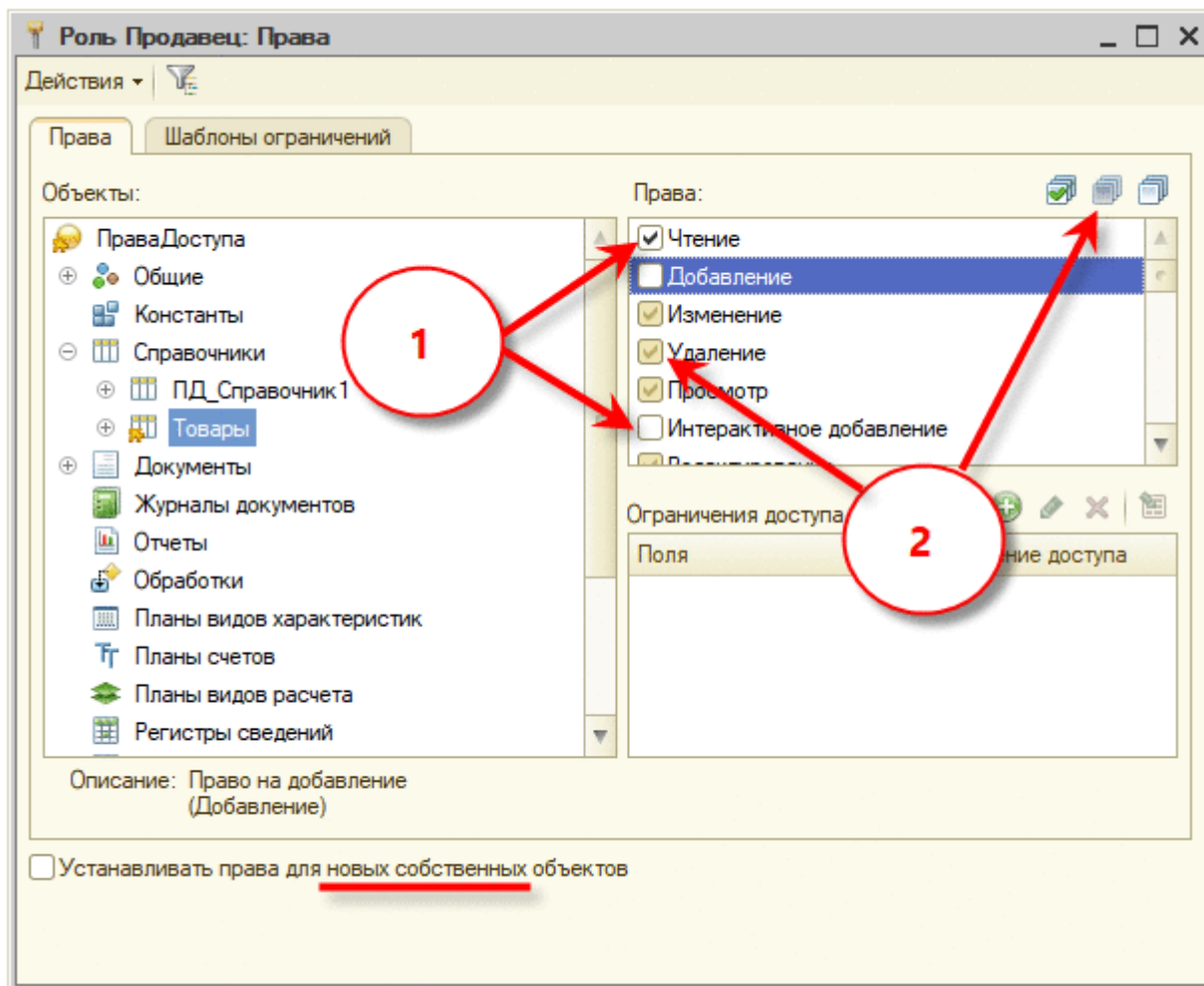


Рис. 26. Редактор заимствованной роли

За права на реквизиты и табличные части объектов расширения отвечает свойство Устанавливать права для реквизитов и табличных частей по умолчанию расширяемой роли. Возможность устанавливать независимые права на подчиненные объекты конфигурации в расширяющей роли определяется флажком Независимые права подчиненных объектов расширяемой роли. Другими словами, права на реквизиты и табличные части, а также возможность задания

независимых прав на подчиненные реквизиты определяются настройками расширяемой роли.

3.2. Собственные роли

Собственная роль не предоставляет возможность устанавливать права доступа, которые привязаны к корневому элементу конфигурации (административные права, использование клиентских приложений, использование монопольного режима и т. д.). Собственные роли расширений могут быть добавлены пользователям информационной базы как интерактивно, так и с помощью встроенного языка. После удаления расширения все собственные роли этого расширения удаляются из списков ролей пользователей.

Пользователь, роли которого дополнены расширением, отображается в конфигураторе со специальным значком.

Собственная роль, автоматически создаваемая при создании расширения, может быть произвольным образом переименована или удалена. Если изменяется префикс расширения, то предлагается выполнить переименование автоматически созданной роли, если выполняются следующие условия:

- Имя роли сформировано как <ПрефиксРасширения>ОсновнаяРоль.
- Роль с таким (как в указано выше) именем установлена в свойство расширения ОсновныеРоли.
- В расширении отсутствует роль <НовыйПрефиксРасширения>ОсновнаяРоль.

Соответственно, предлагается заменить имя роли <СтарыйПрефиксРасширения>ОсновнаяРоль на имя <НовыйПрефиксРасширения>ОсновнаяРоль. Никаких ограничений на модификацию автоматически созданной собственной роли системой не накладывается.

Для редактирования собственной роли расширения предназначен специальный редактор. По внешнему виду и возможностям он аналогичен редактору роли расширяемой конфигурации, однако имеет некоторые отличия как в со-

ставе редактируемых прав (что обусловлено спецификой расширения), так и в поведении флажков роли (рис. 27).

Флажок Устанавливать права для реквизитов и табличных частей по умолчанию предоставляет (если установлен) полные права на реквизиты и табличные части как для собственных объектов расширения, так и для объектов расширяемой конфигурации. Также этот флажок предоставляет права на реквизиты и табличные части, которые добавлены в результирующую конфигурацию другими расширениями.

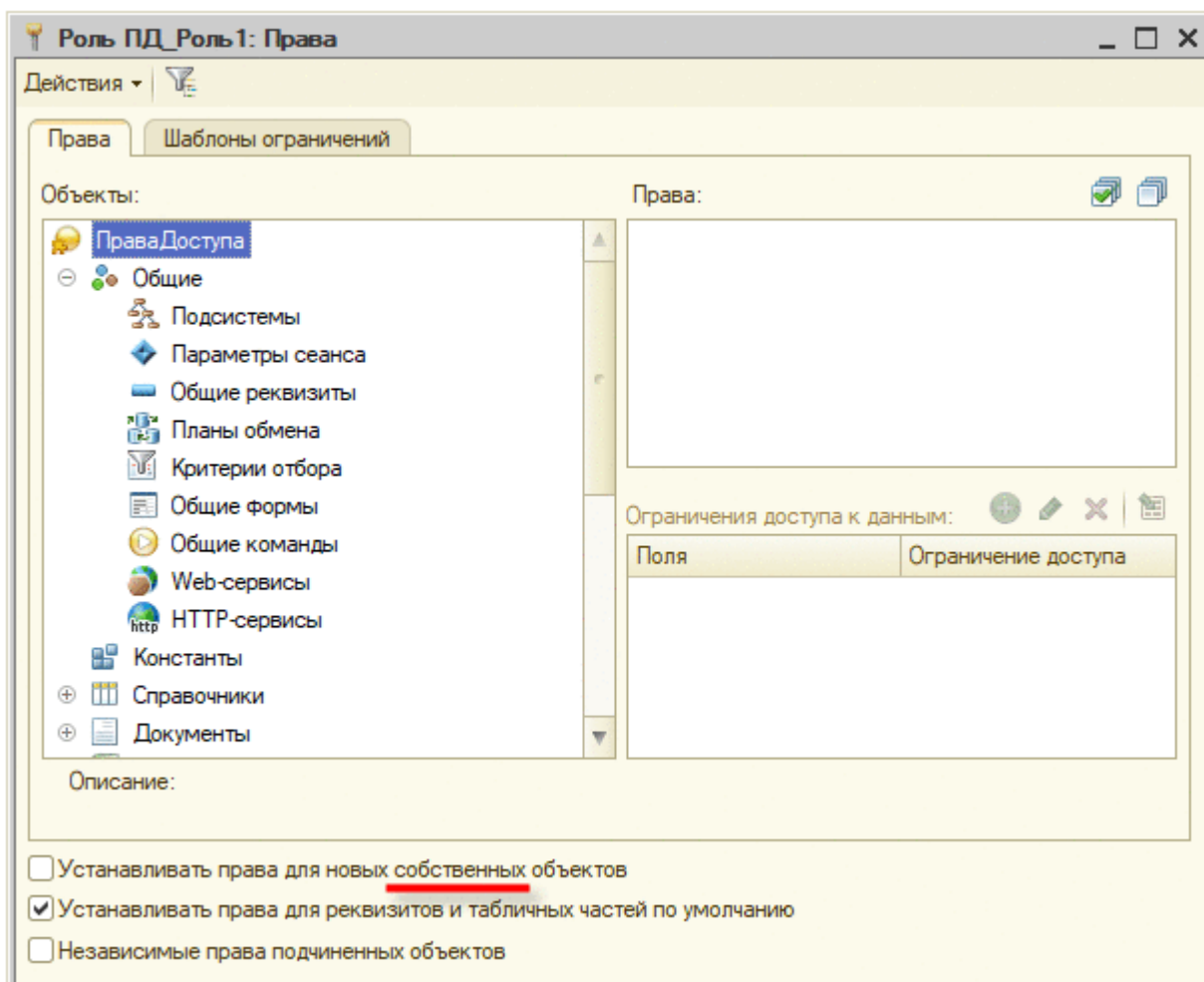


Рис. 27. Редактор собственной роли

Флажок Устанавливать права для новых собственных объектов действует только на собственные объекты расширения.

Флажок Независимые права подчиненных объектов работает полностью эквивалентно такому же флажку в роли конфигурации.

В том случае, когда расширение становится неактивным (в результате обнаружения проблем применимости), собственная роль расширения продолжает быть доступной в коллекции ПользовательИнформационнойБазы.Роли как обычный объект конфигурации. Такая роль может быть добавлена или удалена из списка ролей любого пользователя информационной базы. Однако, функция ПравоДоступа() для такой роли всегда будет возвращать значение Ложь. Если выполняется получение ролей для пользователя из другой области данных, то все роли, добавленные расширениями, становятся невидимыми (их нельзя получить, но при сохранении пользователя они не удаляются).

При разработке расширений, не содержащих заимствованных объектов, рекомендуется придерживаться следующих рекомендаций по управлению правами доступа на объекты расширения:

- Если расширение небольшое и не предполагает ролевого доступа к частям расширения, то достаточно автоматически созданной основной роли. Эта роль предоставит доступ ко всем объектам расширения.

- Если в расширении предполагается ролевой доступ к частям расширения (например, расширение формирования и отправки отчетов, где все пользователи могут формировать отчет, а отправлять должен ответственный пользователь), рекомендуется в основной роли (или ролях) указать права для большинства пользователей. Роль ответственного пользователя (кто может отправлять отчеты) администратор должен будет указать явным образом.

- Если расширение подключается с установленным флажком Использовать основные роли для всех пользователей, то рекомендуется обратить особое внимание на права, которые предоставляются ролями, входящими в состав свойства расширения ОсновныеРоли. Это связано с тем, что все пользователи информационной базы получают доступ в соответствии с правами основной роли (или ролей).

- Если права к каким-либо объектам расширения необходимо предоставлять выборочно, не всем пользователям, то рекомендуется:

- Создать для таких объектов расширения отдельную роль.

- Не указывать эту роль в свойстве ОсновнаяРоль.
- Исключить доступ к «приватным» объектам расширения из ролей, указанных в свойстве ОсновныеРоли.

3.3. Расширение прав пользователей и профили безопасности

Изменение прав доступа для объектов расширяемой конфигурации зависит от варианта работы информационной базы:

- Если в файловом варианте расширение подключается в безопасном режиме, то не поддерживается возможность расширять права доступа на собственные объекты расширяемой конфигурации. Предоставляется возможность только ограничить права доступа.

- В клиент-серверном варианте изменение прав доступа регулируется наличием профиля безопасности и его настройками:

- Если прикладное решение работает без использования профиля безопасности, то расширение ролей не допускается.

- Если прикладное решение работает с использованием профиля безопасности, то ограничивать права доступа можно в любом случае, а расширять права доступа можно только при соответствующей настройке профиля безопасности.

Если в свойствах профиля безопасности установлен флажок к расширению прав доступа (в группе Разрешен полный доступ), то расширять права доступа можно для любого объекта расширяемой конфигурации.

В том случае, если этот флажок сброшен, то расширять права доступа невозможно. Если при этом заполнено свойство профиля безопасности Роли, ограничивающие расширение прав доступа (роли разделяются символом «;»), то система работает следующим образом: роль из расширения конфигурации может расширить право доступа только в том случае, если хотя бы в одной из ролей, перечисленных в свойстве профиля безопасности Роли, ограничивающие расширение прав доступа, установлено расширяемое право доступа. При этом необязательно, чтобы роли, перечисленные в свойстве профиля безопасности,

использовались для пользователей информационной базы прикладного решения. Пример такого поведения будет рассмотрен ниже.

- Для собственных объектов расширения права доступа предоставляются в соответствии с настройками ролей расширения (собственными или заимствованными), без взаимодействия с профилем безопасности. Это поведение не зависит от настроек профиля безопасности.

Рассмотрим расширение права доступа на примере. Допустим, в конфигурации имеется документ РасходнаяНакладная. В расширении необходимо расширить право доступа Запись. Для этого используется роль расширения ПродавецРасширение. Также в расширяемой конфигурации присутствуют роли ПродавецШаблон1 и ПродавецШаблон2. В свойстве профиля безопасности Роли, ограничивающие расширение прав доступа указаны обе роли: ПродавецШаблон1 и ПродавецШаблон2.

Пример 1.

В расширяемой конфигурации для документа РасходнаяНакладная право Запись указано следующим образом:

- ПродавецШаблон1 – право Запись разрешено;
- ПродавецШаблон2 – право Запись запрещено.

В этом случае работа происходит следующим образом: хотя бы одна из ролей, указанных в свойстве Роли, ограничивающие расширение прав доступа право Запись разрешено (это роль ПродавецШаблон1). Это означает, что роль из расширения ПродавецРасширение **может** предоставить право Запись для документа РасходнаяНакладная.

Пример 2.

В расширяемой конфигурации для документа РасходнаяНакладная право Запись указано следующим образом:

- ПродавецШаблон1 – право Запись запрещено;
- ПродавецШаблон2 – право Запись запрещено.

В этом случае работа происходит следующим образом: никакая роль из списка ролей, указанных в свойстве Роли, ограничивающие расширение прав

доступа, не предоставляет право Запись. Это означает, что роль из расширения ПродавецРасширение не может предоставить право Запись для документа РасходнаяНакладная.

Другими словами, алгоритм определения возможности расширить права доступа из расширения, можно описать так: состояние требуемого права из всех ролей, перечисленных в свойстве Роли, ограничивающие расширение прав доступа, объединяется «по ИЛИ». С результатом сложения «по И» объединяется право из роли расширения.

3.4. Отчеты и обработки, XDTO-пакет, Web-сервис, HTTP-сервис

Создание нового отчета или обработки не отличается от таковой в конфигурации, не содержащей расширения. В тоже время, имеется возможность расширить существующие формы отчетов или обработок расширяемой конфигурации, выполнив заимствование необходимого объекта.

Для заимствованного объекта имеется возможность создавать реквизиты объектов, табличные части и реквизиты табличных частей. Работа с этими объектами не отличается от таковой в конфигурации, не содержащей расширений.

Для заимствованного объекта Отчет можно переопределить основную схему компоновки данных. Для этого необходимо добавить отчет в расширение, затем отметить свойство Основная схема компоновки данных как изменяемое и затем создать нужную схему компоновки данных для отчета. В том случае, если требуется изменить (а не создать заново) схему компоновки данных заимствованного отчета, то рекомендуется скопировать схему компоновки данных из основной конфигурации в расширение, указать эту схему в качестве основной в заимствованном отчете и затем модернизировать эту схему. Заимствование объектов, входящих в схему компоновки данных, следует выполнять вручную.

XDTO-пакет может быть как заимствованным, так и собственным объектом расширения.

Заимствовать можно не только целиком пакет, но и какую-то часть этого пакета, если для нормального функционирования расширения не требуется весь XDTO-пакет. Если заимствуемый элемент уже существует в расширении, пользователю задаётся запрос на его обновление.

При заимствовании переносится:

- Все установленные свойства элемента, включая списочные и агрегатные;
- Все установленные свойства создающихся родительских элементов, кроме списочных и агрегатных.

При подключении расширения, для заимствованного XDTO-пакета выполняется контроль на строгое подмножество, т. е. все заимствованные элементы пакета должны входить в пакет из расширяемой конфигурации с точным совпадением количества и значения свойств.

Публикация собственного Web-сервиса расширения возможна двумя способами:

1. Безусловным разрешением использовать любой Web-сервис из любого подключенного расширения. Для этого при публикации следует воспользоваться флажком диалога публикации Опубликовать Web-сервисы расширений по умолчанию или аналогичным по смыслу атрибутом `publishExtensionsByDefault` элемента `ws` файла `default.vrd`.

2. Вручную указав Web-сервисы расширений, которые могут быть использованы, в файле `default.vrd`, в элементе `point`.

Использование Web-сервисов из расширений доступно по значению свойства Web-сервиса `Имя`, так и значению свойства Web-сервиса `Имя` файла публикации. Если в расширениях существует несколько Web-сервисов с одинаковыми значениями свойства `Имя` файла публикации, то при обращении к Web-сервису с использованием значения этого свойства, будет использоваться первый найденный Web-сервис.

При подключении расширения к информационной базе, выполняется контроль уникальности значения свойства `Корневой URL` собственных HTTP-

сервисов расширения. Расширения, в которых существуют HTTP-сервисы, у которых значение свойства Корневой URL нарушает уникальность, не подключаются.

3.5. Табличный документ

При редактировании заимствованного табличного документа предоставляются некоторые возможности, облегчающие редактирование табличного документа. Данные возможности собраны в меню Главное меню - Таблица - Расширение (и соответствующая кнопка командной панели редактирования табличного документа).

Команда Способ расширения позволяет указать, каким образом будет расширяться табличный документ: объединением расширяемой таблицы и таблицы из расширения (способ Объединять) или таблица расширения будет целиком замещать собой расширяемый табличный документ (способ Замещать).

Команда Отображать изменения позволяет цветом выделять области табличного документа (рис. 28).

		1	2	3	4	5	6							
Заголовок	1	Расход товара												
	2													
Шапка	3													
	4													
	5								Номер	<Номер>				
	6								Дата	<Дата>				
	7								Покупатель	<Покупатель>				
8	Склад	<Склад>												
9														
ТоварыШаг	10													
	11	Товар	Цена	Артикул	Количество	Сумма								
Товары	12	<Товар>	<Цена>	<Артикул>	Количество>	<Сумма>								
	13													
	14													

Рис. 28. Отображение изменений табличного документа

В приведенном на рисунке табличном документе серым цветом фона отображаются ячейки расширяемого табличного документа, которые не были изменены в расширении. Зеленым цветом фона отображаются ячейки, которые были добавлены в расширении. А для ячеек табличного документа, свойства

которых были изменены в расширении – используется текущий цвет фона табличного документа.

Так, в документе, приведенном на рис.28, можно отметить следующие изменения:

- в строках 11 и 12 добавлены ячейки в колонке 4 (отображение колонки Артикул в табличной части документа Расход товара);
- в колонке 6 добавлены ячейки в строках 1-4 и 9-10. Это явилось следствием добавления колонки с артикулом.
- в ячейке, расположенной во 2 колонке 12 строки изменено значение некоторых свойств ячейки, поэтому ячейка отображается с цветом фона, отличным от серого цвета.

Чтобы увидеть различия в значениях свойств ячейки, предназначена специальная кнопка командной панели палитры свойств ячейки (рис. 29).

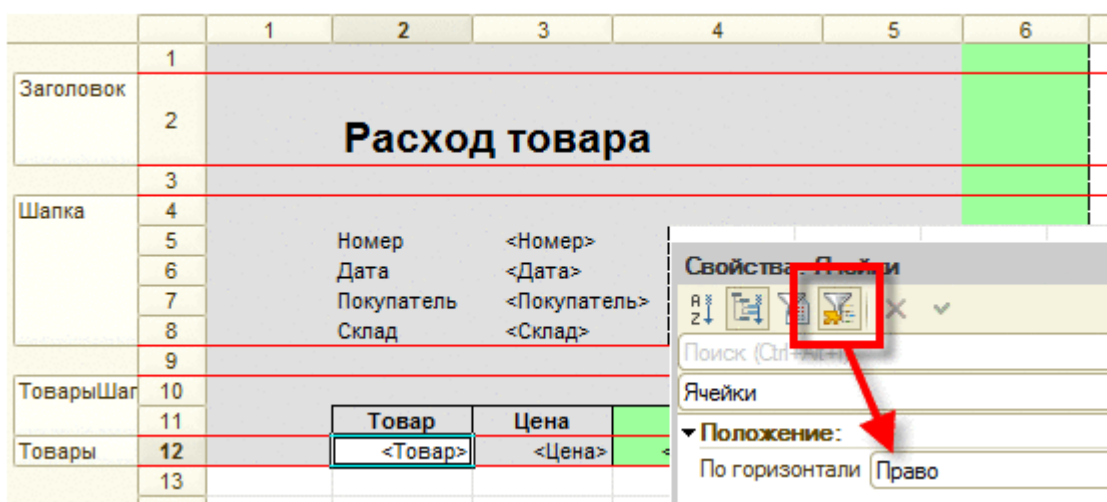


Рис. 29. Изменения свойств ячейки

Кнопка Показывать только измененные доступна только в тех ячейках заимствованного табличного документа, в которых изменены свойства ячеек. При нажатии на эту кнопку, палитра свойств начинает отображать только измененные свойства. Так, в примере на рис.29, видно, что в ячейке, которая находится на во 2 колонке 12 строки, изменено горизонтальное выравнивание. Остальные свойства этой ячейки остались без изменений.

С помощью команды Показать сохраненную таблицу можно увидеть расширяемый табличный документ, в котором отмечены области, которые уда-

лены в табличном документе, который расположен в расширении. Такие области отмечены красным цветом фона (рис. 30). Этот табличный документ отображается в режиме «только просмотр».

		1	2	3	4	5
	1					
Заголовок	2	Расход товара				
	3					
Шапка	4					
	5	Номер	<Номер>			
	6	Дата	<Дата>			
	7	Покупатель	<Покупатель>			
	8	Склад	<Склад>			
	9					
	10					
ТоварыШаг	11					
	12	Товар	Цена	Количество	Сумма	
Товары	13	<Товар>	<Цена>	<Количество>	<Сумма>	
	14					
	15					

Рис. 30. Удаленные ячейки

Видно, что в расширяемом табличном документе между областями Шапка и ТоварыШапка были две строки: 9 и 10. Однако, в расширении одна строка (с номером 10) была удалена.

Команда Показать результат отображает таблицу (в режиме «только просмотр»), которая представляет собой табличный документ, который получится в результате применения расширения, в котором находится расширяемый табличный документ.

Команда Обновить расширение таблицы позволяет обновить расширяемый табличный документ в том случае, если он изменен в расширяемой конфигурации.

3.6. Планы обмена, Справочники, Документы, Перечисления

В расширении имеется возможность управлять контролируемостью состава плана обмена. Для этого необходимо установить признак проверки значения при подключении расширения у свойства Состав плана обмена. Также имеется возможность заимствовать в расширение конкретный элемент состава плана обмен. При этом в расширение переносятся:

- Признак участия объекта конфигурации в плане обмена;
- Состояние авторегистрации изменений.

Для заимствованного плана обмена предоставляется возможность добавлять реквизиты, табличные части и реквизиты табличных частей, формы, команды и макеты. В состав заимствованного плана обмена можно включать как собственные объекты расширения, так и объекты из расширяемой конфигурации (которые ранее не входили в состав этого плана обмена). Не поддерживается возможность исключать объекты расширяемой конфигурации из состава плана обмена. Другими словами, расширение не может уменьшить объем информации, передаваемой планом обмена.

При подключении расширения проверяется:

- Совпадение признака участия объекта конфигурации в плане обмена.
- Совпадение состояния авторегистрации изменений.

В расширении имеется возможность создать собственный план обмена. Такой план обмена будет обладать следующими особенностями:

- Собственный план обмена расширения не может участвовать в распределенной информационной базе.

- Система не предоставляет инструментов для автоматической синхронизации состава расширений между узлами обмена данными. Эту синхронизацию необходимо выполнять отдельно.

Справочники могут быть как заимствованными, так и собственными. В заимствованных справочниках имеется возможность добавлять реквизиты, табличные части и реквизиты табличных частей.

Документы могут быть как заимствованными, так и собственными. В заимствованных документах имеется возможность добавлять реквизиты, табличные части и реквизиты табличных частей. Для собственных документов расширения не поддерживается использование нумераторов и последовательностей.

Документ может выступать регистратором только для того регистра, которые находится в той же «части» прикладного решения, что и сам документ. Другими словами, документ из расширения может выступать регистратором

для собственных и заимствованных регистров расширения и не может выступать регистратором для собственных регистров расширяемой конфигурации. Также и наоборот: собственный документ конфигурации не может являться регистратором для собственного регистра расширения.

Документ расширения (как собственный, так и заимствованный) может выступать регистратором для любого заимствованного регистра расширения, кроме обратного регистра накопления с включенным режимом агрегатов.

Перечисление может быть заимствованным и собственным. В заимствованном перечислении имеется возможность создавать новые значения.

При этом следует помнить о следующих особенностях:

- В том случае, если в информационной базе используются значения из расширяющего объекта, то в ситуации неподключенного расширения такие значения отображаются аналогично ссылке на несуществующий объект. После подключения расширения представление восстанавливается.

- При удалении расширения из информационной базы ссылки на значения перечисления из расширяющего объекта будут очищены и заменены пустой ссылкой с типом расширяемого перечисления.

- Возможна ситуация, когда одинаковые ссылки на значение перечисления из разных расширений имеют разный смысл. Такое возможно в том случае, если переименовать собственное значение перечисления в расширении, созданном копированием. В этом случае запись такого значения в неразделенные данные приведет к логической проблеме для пользователя прикладного решения. Например, значение перечисления с идентификатором (внутренним) ИД1 будет иметь в одном расширении представление Самолет, а в другом расширении – Пароход. Очевидно, что пользователю будет сложно понять, что же имеется в виду для данного значения перечисления. Из этого следует две особенности:

- Не рекомендуется сохранять собственные значения для заимствованных перечислений в расширениях, созданных копированием. Рекомендуется все собственные значения создавать вручную.

- Не рекомендуется указывать в неразделенных данных собственные значения заимствованных перечислений расширений, область действия которых установлена как «Разделение данных».

- Если делается попытка установить расширение конфигурации, в котором используется идентификатор значения перечисления, равный идентификатору значения перечисления расширяемой конфигурации или другого расширения, то выполнить реструктуризацию информационной базы будет невозможно.

- В том случае, если внутренний идентификатор перечисления в расширяемой конфигурации совпадает с внутренним идентификатором перечисления расширения, то:

- реструктуризация информационной базы не блокируется;

- расширение, в котором размещено конфликтующее значение перечисления не подключается.

Для восстановления работоспособности информационной базы (или области) следует удалить конфликтующее значение перечисления в расширении и загрузить в информационную базу новую версию. Существующие ссылки на значение перечисления из расширения будут разрешены значением перечисления из расширяемой конфигурации.

3.7. Расширение и распределенная информационная база

Расширение конфигурации можно использовать в распределенной информационной базе. В этом случае расширение будет передаваться между узлами распределенной информационной базы с помощью сообщений обмена, тем самым обеспечив единый состав расширений в рамках распределенной системы.

Для указания того, что план обмена может передавать расширения, необходимо установить у этого плана обмена флажок Включать расширения конфигурации. Этот флажок можно установить только для планов обмена, которые участвуют в распределенной информационной базе.

В тоже время, каждое расширение, подключаемое к информационной базе, имеет свойство Используется в распределенной информационной базе. Установка данного свойства означает, что данное расширение будет мигрировать в подчиненные информационные базы. Возможность изменения данного свойства не связана с фактическим наличием в информационной базе планов обмена, включенных в распределенную информационную базу.

У объекта РасширениеКонфигурации существует свойство ГлавныйУзел, которое позволяет определить, что расширение добавлено в текущем узле (свойство имеет значение Неопределено) или поступило с помощью механизма обмена данными (содержит ссылку на главный узел распределенной информационной базы).

При создании начального образа периферийной базы, который поддерживает передачу расширений конфигурации, в него переносятся все расширения, которые подключены с установленным флагом Используется в распределенной информационной базе. Создание начального образа будет невозможно, если в сеансе, из которого выполняется создание начального образа:

- Существуют неподключенные расширения с флагом Используется в распределенной информационной базе.
- Существуют расширения, изменяющие данные, для которых установлен флаг Используется в распределенной информационной базе и которые не являются активными.
- Существуют расширения, изменяющие данные, для которых не установлен флаг Используется в распределенной информационной базе.
- Существуют расширения с флагом Используется в распределенной информационной базе, для которых загружены новые версии расширений.

В момент формирования сообщения обмена на сеанс, из которого выполняется формирование, накладываются те же требования, что и при формировании начального образа.

В периферийном узле распределенной информационной базы запрещено редактировать свойства расширений и удалять расширения, которые добавлены

в главном узле. Эти действия запрещены как в интерактивном режиме, так и с помощью методов встроенного языка. Признаком того, что расширение получено с помощью механизма обмена данными, отображается в интерфейсе стандартной функции Управление расширениями конфигурации.

При старте сеанса расширения применяются в следующем порядке:

- расширения, добавленные центральным узлом в том же порядке, в котором они применяются в центральном узле;
- расширения, добавленные в текущем узел, в своем порядке.

Если в узел распределенной информационной базы приходит сообщение о том, что для расширения сброшен флажок Используется в распределенной информационной базе, то такое расширение удаляется.

Если в информационной базе имеются расширения, которые поступили с помощью сообщений обмена распределенной информационной базы, то в такой информационной базе с помощью метода ГлавныйУзел() менеджера планов обмена можно установить только тот узел, из которого поступили расширения или отключить информационную базу от распределенной системы (установить значение Неопределено). В частности, это означает, что без удаления расширений невозможно переключить информационную базу из одного распределенной информационной базы в другую с помощью смены главного узла.

При подключении расширения к прикладному решению имеется возможность указать, будет расширение выполняться в безопасном режиме или нет.

Это можно сделать:

1. Явным указанием флажка Безопасный режим. В этом случае расширение будет выполняться в безопасном режиме или небезопасном режиме (в зависимости от состояния флажка).

2. Указанием имени профиля безопасности, в рамках которого будет выполняться расширение. Данная возможность доступна только в том случае, если прикладное решение используется в клиент-серверном варианте.

Указание профиля безопасности при подключении расширения указывает, что режим работы расширения определяется настройками указанного про-

филя безопасности. В профиле безопасности доступ к расширению указывается в настройках элемента Внешние модули.

Если в профиле безопасности запрещено небезопасное использование внешних модулей, а расширение требует такое использование, то его (расширение) можно внести в список исключений. Чтобы получить значение контрольной суммы расширения для добавления в список исключений, следует воспользоваться возможностями диалогов работы с расширениями или программного интерфейса работы с расширениями. Полученное значение контрольной суммы следует ввести в соответствующее поле в консоли администрирования кластера.

При подключении расширения проверяется возможность включения небезопасного режима (если требуется) и, если текущее расширение не соответствует требованиям указанного профиля безопасности - подключение расширения не происходит и выдается ошибка.

Для изменения настроек безопасного режима при подключении расширения необходимо, чтобы пользователь, выполняющий эти настройки, обладал правом Администрирование (в дополнение к праву АдминистрированиеРасширенийКонфигурации, которое позволяет выполнять остальные действия с расширениями).

Подключение расширения к сеансу пользователя выполняется перед вызовом обработчика события УстановкаПараметровСеанса и остается подключенным до завершения сеанса. Обновление расширения (с помощью инструментов администрирования) не приводит к обновлению расширения в существующих сеансах. Существующие сеансы продолжают работу с той «версией» расширения, которое было доступно в момент старта сеанса. Для того чтобы пользовательский сеанс начал работать с новой «версией» расширения – необходимо выполнить перезапуск сеанса.

Если расширение конфигурации использует расширение данных, то создание новых таблиц (для хранения данных расширения) в информационной базе выполняется:

- В конфигураторе – при выполнении реструктуризации информационной базы.

- В режиме 1С:Предприятия – во время добавления или обновления расширения. Фактически, изменение структуры базы данных выполняется во время выполнения метода Записать() объекта РасширениеКонфигурации.

При удалении расширения порядок действий аналогичный, только вместо создания таблиц выполняется их удаление.

Порядок подключения расширений определяется назначением расширения и его областью действия. Расширения подключаются в следующем порядке:

1. Исправление;
2. Адаптация;
3. Дополнение.

При этом в рамках каждого назначения вначале подключаются расширения с областью действия Информационная база, а затем – Область данных. Если существует несколько расширений с одинаковым назначением и областью действия – порядок подключения таких расширений в общем случае не определен.

При попытке подключения расширений, каждое расширение последовательно проверяется на соответствие контролируемых реквизитов между расширением и конфигурацией информационной базы. Если хотя бы один контролируемый реквизит не проходит проверку, пользователю выдается сообщение об ошибке и это расширение не подключается. При проверке свойств с несколькими значениями (составной тип, состав общего реквизита) значение из расширения контролируется на вхождение – все значения, которые есть в расширении, должны быть в расширяемой конфигурации, однако часть состава расширяемой конфигурации может не входить в контролируемое свойство. Как следствие, все объекты расширения перестают участвовать в формировании результирующей конфигурации.

Если при подключении расширений обнаружены проблемы, то пользователю выводится диалог с информацией о том, с какими расширениями возникли проблемы, а детальный список проблем доступен по нажатию на гиперссылку [Информация для технической поддержки](#), доступная в этом диалоге.

Кроме интерактивной проверки применимости, платформа «1С:Предприятие» предоставляет возможность выполнять проверку применимости из кода на встроенном языке. Имеется возможность выполнить проверку для конкретного расширения с помощью метода `РасширениеКонфигурации.ПроверитьВозможностьПрименения()`. Также имеется возможность выполнить проверку всех расширений, подключенных в данный момент. Для этого следует использовать метод `РасширенияКонфигурации.ПроверитьВозможностьПримененияВсех()`.

В том случае, если при старте сеанса подключены не все расширения, расширяющие данные, прикладное решение может без ошибок читать данные расширяемой конфигурации успешно подключенных расширений. При попытке записи объект, данные которого дополняются отключенным расширением, поведение будет зависеть от используемого объекта:

Поддерживается создание новых, удаление существующих и запись изменений в существующих объектах. Поля отключенных расширений при создании объектов заполняются значениями по умолчанию. Сохраняются значения полей из отключенных расширений в модифицируемых объектах. При копировании объекта значения полей отключенных расширений в новый объект не копируются.

При редактировании строк табличной части (программном или интерактивном, в том числе при изменении порядка строк) сохраняют данные в полях отключенных расширений. Поля отключенных расширений при создании строк заполняются значениями по умолчанию. При копировании строк, значения полей отключенных расширений в новую строку не копируются.

Для регистров, для которых список регистраторов расширен неактивным расширением, поддерживается запись наборов записей только для тех реги-

страторов, которые предоставлены расширяемой конфигурацией. Чтение доступно для любого набора записей. Однако, регистратор, который предоставлен неактивным расширением, в таком наборе записей будет представлен значением Неопределено. Как следствие, такой набор записей нельзя изменить и удалить.

В том случае, когда тип какого-либо реквизита (измерения, ресурса и т. д.) прикладного решения модифицирован отключенным расширением, то:

- При чтении объекта с таким реквизитом: значения, хранимые в информационной базе будут преобразовываться к типам конфигурации без учета неактивных расширений. Конвертация подчиняется логике работы метода ОписаниеТипов.ПривестиЗначение(). Ссылки на объекты неактивных расширений предварительно преобразуются в значения Неопределено.

- При записи объекта с таким реквизитом: обновляются только изменившиеся значения (из встроенного языка или интерактивно). Поля информационной базы, которые не изменены в записываемом объекте, могут сохранять значения, установленные при активных расширениях конфигурации. Записываемое значение будет приводиться к типам конфигурации, без учета неактивных расширений.

- При копировании объекта с такими реквизитами: значения реквизитов, модифицированных неактивным расширением, копируются в новый объект со значением, приведённым к типу расширяемой конфигурации без учета неактивных расширений.

- В процессе выполнения запроса - в выражения, условиях отбора, группировках и при упорядочивании будут использоваться непосредственно значения, хранимые в информационной базе.

При указании в условиях и группировках, ссылки на объекты из неактивных расширений считаются заполненными. При использовании таких ссылок в условиях, они не проходят сравнение со значениями Неопределено и NULL. В остальных случаях такие ссылки преобразуются в значение Неопределено.

Результатом обращения к реквизитам ссылок («через точку») на объекты из неактивных расширений является NULL.

При считывании результата запроса в объекты встроенного языка значения будут преобразовываться к типам конфигурации без учета неактивных расширений.

Для случаев, когда пользователь начал сеанс в области с расширением данных, а затем переключился в область данных с другим набором расширений, либо обновил или добавил расширение, меняющее структуру информационной базы, запрет на чтение и запись таблиц, модифицированных в начальной области – остаётся.

В случае необходимости реализации собственных инструментов работы с расширениями следует использовать специализированный программный интерфейс. Его методы доступны в том случае, если текущий пользователь обладает правом АдминистрированиеРасширенийКонфигурации. Текущий сеанс должен работать в небезопасном режиме в том случае, если будет использоваться метод Записать() объекта РасширениеКонфигурации.

Программный интерфейс предоставляется посредством специального менеджера расширений конфигурации, который представлен свойством глобального контекста РасширенияКонфигурации. Подробное описание методов приведено в синтакс-помощнике.

Из встроенного языка имеется возможность выполнить проверку того, что объект конфигурации расширен в данном сеансе работы. Для этого предназначен метод объекта конфигурации <Объект метаданных>.ЕстьИзмененияРасширениямиКонфигурации(). Данный метод вернет значение Истина в том случае, если в текущем сеансе выбранный объект конфигурации модифицирован расширениями одним из следующих способов:

- модифицировано свойство объекта конфигурации;
- добавлен новый дочерний объект объекта конфигурации (форма, макет, команда и т. д.);
- модифицировано свойства одного из дочерних объектов.

С помощью метода `<Объект метаданных>`. `РасширениеКонфигурации()` имеется возможность определить, какому расширению конфигурации (или расширяемой конфигурации) принадлежит объект конфигурации.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Абрашина, Е. Использование механизма расширенной аналитики в «1С: Управление производственным предприятием» / Е. Абрашина, И. Емельянов. – М.: 1С-Паблишинг, 2016. – 184 с.
2. Асатрян, А. Методическое пособие по эксплуатации крупных информационных систем на платформе 1С: Предприятие 8 / А. Асатрян, А. Голиков, А. Морозов. – М.: ООО «1С-Паблишинг»; СПб.: Питер, 2017. – 220 с.
3. Корниенко, Д.В. Автоматизация бизнес-процессов в 1С:ERP Управление предприятием 2: учебно-методическое пособие / Д.В. Корниенко. – Елец: ФГБОУ ВО «Елецкий государственный университет им. И.А. Бунина», 2019. – 85 с.
4. Корниенко, Д.В. Примеры разработки прикладных решений на базе 1С:Предприятие 8 / Д.В. Корниенко. – Елец: Елецкий государственный университет им. И.А. Бунина, 2018. – 80 с.
5. Мишина, С. В. Примеры реализации механизма криптографии в системе 1С:предприятие 8 / С. В. Мишина // Continuum. Математика. Информатика. Образование. – 2020. – № 4(20). – С. 85-93.
6. Мишина, С. В. Ценообразование и ценовая политика в организации / С. В. Мишина // Continuum. Математика. Информатика. Образование. – 2020. – № 3(19). – С. 109-116.
7. Печерских, В. Внедрение ERP-решений на платформе 1С: Предприятие 8 / В. Печерских, Г. Бельцев. – СПб. : БХВ-Петербург, 2015. – 146 с.
8. Хрусталева, Е.Ю. Методическое пособие по эксплуатации крупных информационных систем на платформе 1С: Предприятие 8 / Е. Ю. Хрусталева. – М.: ООО «1С-Паблишинг»; СПб.: Питер, 2017. – 288 с.
9. Информационно-технологическое сопровождение пользователей 1С [Электронный ресурс]. – Режим доступа: <https://its.1c.ru/> (20.10.2021).
10. Хабр. Сообщество IT-специалистов [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/1c/blog/320018/> (17.08.2021).

Учебное издание

Корниенко Дмитрий Васильевич

**Особенности ведения учета
в типовой конфигурации
1С:ERP Управление предприятием 2**

Учебно-методическое
пособие

*Технический редактор – О.А. Ядыкина
Техническое исполнение – В.М. Гришин*

Лицензия на издательскую деятельность
ИД № 06146. Дата выдачи 26.10.01.
Формат 60 x 84 /16. Гарнитура Times. Печать трафаретная
Печ.л. 5,2 Уч.-изд.л. 5,1
Тираж 300 экз. Заказ 93

Отпечатано с готового оригинал-макета на участке оперативной полиграфии
Елецкого государственного университета им. И.А. Бунина

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Елецкий государственный университет им. И.А. Бунина»
399770, г. Елец, ул. Коммунаров, 28,1