**Работа с конфигурацией**

Особенности использования "Загрузки измененной конфигурации"

При выполнении загрузки измененной конфигурации система контролирует правильность использования данной функции. Типичной ошибкой является попытка загрузки конфигурации, не являющейся потомком текущей конфигурации. Не смотря на то, что система допускает такое действие, ограничиваясь выдачей предупреждения, настоятельно не рекомендуется выполнять загрузку, если выдано сообщение о том, что "Выбранный файл конфигурации не является потомком…". Это может привести к достаточно серьезным последствиям , например, потере данных или нарушении работоспособности информационной базы.

Основным применением данной функции является модернизация конфигурации параллельно с работой пользователей с информационной базой. Чтобы корректно применять функцию загрузки измененной конфигурации следует придерживаться следующей последовательности действий:

Скопировать конфигурацию функционирующей в организации информационной базы (только файл 1CV7.MD, или всю информационную базу целиком).

Произвести необходимую модификацию конфигурации на скопированной информационной базе или созданной тестовой базе на основе скопированного файла 1CV7.MD.Данная модификация, разумеется, может выполняться многократно. Важно, чтобы в это время не выполнялось никаких изменений в конфигурации работающей в организации информационной базы, включая изменения состава наборов прав, интерфейсов и т.д. Заметим, что список пользователей не является частью конфигурации и, следовательно, может изменяться.

После окончания отладки сделанных в скопированной конфигурации изменений следует открыть конфигурацию работающей информационной базы и выполнить загрузку измененной конфигурации из модифицированной конфигурации. При этом не должно выдаваться сообщение "Выбранный файл конфигурации не является потомком…", так как конфигурация, в которую выполняется загрузка, ни разу не менялась с момента создания копии. После выполнения загрузки измененной конфигурации выполняется сохранение конфигурации с соответствующей реструктуризацией информационной базы.

Если, предполагается продолжать модификацию конфигурации на отдельной информационной базе, то необходимо выполнить повторное копирование полученной после загрузки измененной конфигурации информационной базы или файла 1CV7.MD. Последующую модификацию следует выполнять уже с вновь сделанной копией, а не с той конфигурацией, которая модифицировалась в прошлый раз. Это объясняется тем, что сам факт сохранения после загрузки измененной конфигурации также является изменением конфигурации и отмечается системой, и, следовательно, сделанная ранее копия уже не будет являться прямым потомком работающей конфигурации.

Именно невыполнение последнего пункта описанной последовательности действий приводит обычно к появлению при выполнении загрузки измененной конфигурации предупреждения. Заметим, что в версии 7.7 для решения аналогичной задачи можно использовать механизм объединения конфигураций. Он позволяет выполнять загрузку изменений, даже если конфигурация работающей информационной базы менялась. Однако, использование механизма объединения требует большей аккуратности, так как процесс объединения не столь однозначен, как загрузка измененной конфигурации, вследствие того, что внесенные в работающую конфигурацию и в ее копию изменения могут быть между собой логически не согласованы, что необходимо анализировать в процессе объединения.

**Использования режима объединения конфигураций**

Рекомендации по использованию режима объединения конфигураций

Основное назначение и дополнительные возможности, предоставляемые режимом объединения конфигураций

Одним из главных назначений режима объединения конфигураций является возможность обновления используемых типовых конфигураций при выпуске очередных релизов.

Кроме того, режим объединения конфигураций позволяет разработчикам прикладных решений, созданных на основе типовых конфигураций, достаточно просто вносить в собственную конфигурацию изменения, которым подверглась типовая конфигурация с момента начала разработки собственной. Ранее использовавшееся в таких случаях средство обновления конфигурации (режим загрузки измененной конфигурации) являлось практически неприменимым, так как после загрузки исчезали все изменения, внесенные разработчиком. Более подробно отличия этих режимов будут рассмотрены в разделе "Отличия режима объединения конфигураций от загрузки измененной конфигурации ".

Однако возможности режима объединения позволяют использовать его и для решения других задач, например, групповой разработки или наследования вновь создаваемой конфигурацией готовых решений, реализованных в других проектах. Интересные применения, например, при сопровождении конфигурации, может иметь также визуализированный список различий в конфигурациях.

**Подготовка к объединению конфигураций**

Безусловно, процесс объединения конфигураций (особенно существенно различающихся) может быть весьма не простым и потребовать подготовительных действий. В частности, он может быть разбит на несколько этапов, причем для каждого этапа разработчиком может быть выбрана своя стратегия объединения.

Подготовка к объединению конфигураций, прежде всего, заключается в согласовании "терминологии", принятой при разработке конкретных конфигураций, в том, какой идентификатор дан тому или иному объекту метаданных, отражающих одну и ту же физическую сущность. Например, справочник "Сотрудники" в одной конфигурации может быть полным или частичным аналогом справочника "Работники" в другой. Поиск пар объектов, которые сравниваются и, при необходимости, замещаются или объединяются, происходит именно по совпадению идентификаторов. В связи с этим, перед выполнением объединения имеет смысл просмотреть объединяемые конфигурации для проверки того, что применяемые идентификаторы идентичны, при необходимости заменить идентификаторы в одной из конфигураций и внести изменения в тексты модулей и формулы там, где использовались измененные идентификаторы.

Существенную помощь на всех этапах слияния конфигураций могут оказать появившиеся в версии 7.7 возможность получения описания конфигурации в текстовом виде, а также специальные средства, вызов которых осуществляется непосредственно из окна "Объединение конфигураций", а именно кнопка "Сравнить", позволяющая получить развернутую информацию об измененном объекте, и кнопка "Отчет...", при помощи которой можно получить на экране или вывести в файл краткий или детальный отчет о различиях в объединяемых конфигурациях.

В процессе разработки плана объединения может выясниться, что одни объекты метаданных нужно перенести "как есть в загружаемой конфигурации", а для других нужно осуществить "слияние" свойств или выполнять обработку объектов в порядке, отличном, от используемого конфигуратором. В этих случаях процесс объединения разбивается на этапы, причем для каждого, как правило, выбирается своя стратегия. Под стратегией в данном случае понимается совокупность метода объединения (замещение или объединение объектов) и выбор приоритетности конфигураций (текущая или загружаемая).

При выборе стратегии объединения для каждого этапа следует четко понимать, каков будет результат того или иного выбора, т.к. от этого весьма существенно зависит алгоритм обработки каждого объекта.

Основные принципы работы режима объединения конфигураций

Наиболее простой для понимания является использование метода замещения объектов. При выборе этого метода просто происходит замена объекта, определенного в текущей конфигурации на соответствующий из загружаемой, если в текущей выбранный объект имеется, или в текущую конфигурацию выбранный объект добавляется, если объект имеется только в загружаемой. Заметим, что несмотря на то, что при замене объекта происходит, с точки зрения пользователя, удаление имеющегося и добавление нового объекта, это не совсем так, такая замена происходит с минимизацией потери данных, т.е. результат получается тот же, как если бы пользователь открыл объект метаданных для редактирования и вручную привел бы все в соответствие с аналогичным объектом из загружаемой конфигурации, а не удалил бы предварительно объект, а потом вставил, например, из буфера обмена - в этом случае потеря данных была бы полной. Приоритет конфигурации влияет на состав замещаемых объектов. Если в качестве приоритетной конфигурации выбрана текущая , то замещения измененных объектов не происходит, в противном случае они включаются в список замещаемых.

Суть стратегии при выборе метода объединения объектов можно свести к следующей фразе: "При выборе метода объединения объектов конфигуратор стремится к тому, чтобы в результате объединения объект содержал все, что было в нем в текущей конфигурации и все, что было в загружаемой". Слово "стремится" здесь употреблено потому, что для некоторых объектов метаданных (например, "Задача") объединение является бессмысленным, они могут только замещаться.

Следуя этому правилу можно достаточно просто представить себе, что происходит с большинством объектов метаданных при объединении. Такие свойства объекта метаданных, как "синоним", "комментарий" и др. берутся из приоритетной конфигурации, а списочные (такие, например, как список реквизитов) сливаются, то есть их список после объединения будет содержать набор реквизитов обоих конфигураций. Отдельно стоит упомянуть свойства, содержащие массив ссылок на элементы метаданных, такие, например как, список входящих в журнал документов. Подобно списочным свойствам, они сливаются, однако, при этом корректно обрабатывается ситуация, когда ссылка в результате оказывается неразрешенной. Примером может служить случай, если объединяемый (или переносимый) журнал документов содержит ссылки на документы, которые не переносятся из загружаемой конфигурации и отсутствуют в текущей. В этом случае ссылка уничтожается, о чем выдается соответствующая строка в окно сообщений.

Теперь более подробно рассмотрим случаи, когда применение сформулированного правила не дает полного представления о результате или дает неверное представление.

Регистры. Специфика этого объекта метаданных состоит в том, что для него существенным является не только состав списочных свойств - измерений, ресурсов, реквизитов, но порядок их следования в списке. Для слияния этих списков используется следующий алгоритм: сначала берутся все элементы списка из приоритетной конфигурации в порядке их следования, затем к списку добавляются не вошедшие ранее элементы из списка неприоритетной конфигурации в порядке их следования в нем.

Формы диалогов. Результирующая форма диалога имеет все слои, которые имели обе исходные формы. В качестве основы берется форма диалога приоритетной конфигурации, элементы диалога, имеющиеся только в неприоритетной конфигурации, добавляются на тот слой, который был использован для каждого конкретного элемента в неприоритетной конфигурации.

Интерес представляет также способ, которым определяются пары соответствующих друг другу элементов диалога, т.к. они не являются элементами метаданных и, соответственно, способ сравнения идентификатора в этом случае неприменим. Поиск в форме элемента диалога, соответствующего данному элементу производится следующим образом: если присутствует идентификатор элемента диалога (который не следует путать с идентификатором элемента метаданных) происходит сравнение по нему. В противном случае производится поиск элемента того же типа (текст, кнопка и т.д.), имеющий такую же формулу (не пустую) или такой же заголовок (не пустой). Заметим, что данный алгоритм обеспечивает определение пары элементов диалога, соответствующих, с точки зрения совпадения идентификаторов, реквизитам отображаемого формой объекта метаданных, так как совпадение идентификаторов реквизитов гарантирует совпадение идентификаторов соответствующих им элементов формы диалога.

Модули (тексты). За основу берется текст модуля приоритетной конфигурации. В нем в виде комментариев вставляются все отличия модуля неприоритетной конфигурации фрагменты текста. В начало каждой строки вставляемого фрагмента текста добавляется двойной слеш (два символа косой черты), что в тексте программного модуля делает эти строки закомментированными. Кроме того, для облегчения дальнейшей работы с результирующим текстом, в него вставляются специальные строчки-комментарии.

Измененные фрагменты обрамляются сверху и снизу строками " //{{MRG[ <-> ] ",

добавленные " //{{MRG[ <-- ] ",

удаленные " //{{MRG[ - -> ] ".

Таблицы. Форма, получившаяся в результате объединения, будет содержать все таблицы, имевшиеся в формах объединяемых конфигураций. В случае, если в обеих формах есть одинаково названные таблицы, таблице неприоритетной конфигурации присваивается новое, уникальное в пределах результирующей формы имя.

Права и интерфейсы. Независимо от установки метода объединения ("Объединять объекты" или "Замещать объекты"), пользовательские интерфейсы всегда объединяются, а права конкретного набора прав для конкретного объекта метаданных замещаются полностью в соответствии с выбранным приоритетом конфигурации. При этом, замещение прав выполняется при условии, что в окне объединения включен признак объединения и у объекта метаданных и у набора прав. При объединении интерфейсов за основу берется интерфейс приоритетной конфигурации, и в него добавляются отсутствующие в нем элементы интерфейса из неприоритетной конфигурации.

Таким образом, режим объединения конфигураций при выборе объединения объектов полностью сохраняет функциональность объекта из приоритетной конфигурации и опционально включает в результирующий объект функциональность из неприоритетной конфигурации. Как правило, для включения опциональной функциональности требуется корректировка конфигурации вручную.

Отличия режима объединения конфигураций от загрузки измененной конфигурации

В процессе работы с конфигурацией каждый вновь образующийся элемент метаданных получает уникальный в пределах данной конфигурации внутренний идентификатор, который сохраняется за ним весь цикл жизни конфигурации. Загрузка измененной конфигурации для определения пар объектов, соответствующих друг другу в текущей и загружаемой конфигурации, использует именно внутренний идентификатор, благополучно отрабатывая ситуацию, когда объект метаданных переименован. Кстати, именно поэтому так важно, чтобы загружаемая конфигурация являлась потомком текущей, так как нет никакой возможности "согласовать" словарь внутренних идентификаторов двух различных конфигураций при использовании загрузки измененной конфигурации.

Напротив, режим объединения конфигураций использует задаваемые пользователем идентификаторы объектов метаданных при определении пар соответствующих друг другу объектов. Это дает возможность объединять конфигурации, не имеющие общего родителя. Хотя следует учитывать, что объединение сильно различающихся конфигураций может потребовать существенной переработки полученной конфигурации для согласования структур данных, алгоритмов, а, зачастую, и самой концепции унаследованной от обеих конфигураций.

Однако может оказаться полезным даже из сильно отличающейся конфигурации выделить и перенести в текущую один или несколько объектов, реализующих ту или иную функцию, не поддерживаемую текущей конфигурацией. Метод объединения конфигураций дает такую возможность, благодаря реализованной выборочности переноса, чего не поддерживает загрузка измененной конфигурации.

Наконец, еще одним отличием режима объединения конфигураций является наличие метода объединения объектов, т.к. загрузка измененной конфигурации способна лишь замещать объекты.

Фактически, режим загрузки измененной конфигурации используется исключительно для обеспечения возможности изменения копии конфигурации параллельно с работой пользователей с информационной базой, и затем однозначного внесения сделанных в копии изменений в "рабочую" конфигурацию.

Из сказанного выше можно сделать вывод, что режим объединения конфигураций является по сравнению с загрузкой измененной конфигурации более развитым, дающим новые возможности способом обновления конфигураций, однако, его применение предъявляет к пользователю повышенные требования с точки зрения понимания механизма объединения.

**Сравнение внешних отчетов**

Режим сравнения файлов Конфигуратора (меню "Файл - Сравнить файлы") позволяет сравнивать любые файлы на полное совпадение, а для текстовых файлов предоставляет возможность сравнения с показом отличий в тексте.

При сравнении внешних отчетов может быть выполнено сравнение самих файлов на полное совпадение, а может быть выполнено сравнение текстов модулей с показом отличий в тексте.

Для принудительного сравнения модулей внешних отчетов следует при выборе файлов выбрать тип файла "Внешний отчет (обработка)"

Контроль целостности файла конфигурации в процессе разработки и внесения изменений в конфигурацию

Аппаратные сбои некачественного компьютера во время сохранения файла конфигурации (1cv7.md) могут привести к частичному или полному его разрушению. Это подтверждает и анализ статистики обращений на линию консультаций по поводу разрушения файла конфигурации. Например, проведенное однажды исследование причин появления ошибки в файле конфигурации показало, что разрушение произошло на компьютере с неисправной оперативной памятью.

Приведенные ниже простые рекомендации помогут Вам избежать последствий аппаратных сбоев во время сохранения конфигурации:

проводите редактирование конфигурации и реструктуризацию только на компьютере, в качестве которого Вы уверены;

постоянно имейте резервную копию информационной базы, проверенную на отсутствие ошибок в файле конфигурации.

Для проверки отсутствия ошибок в файле конфигурации при применении версии 7.7 можно использовать встроенный в Конфигуратор режим тестирования и исправления ИБ. При этом все флажки в диалоге режима для ускорения процесса можно убрать и выбрать режим "Только тестирование". В этом случае будет проверяться только файл конфигурации. Для версии 7.5 проверку можно осуществить при помощи режима "Поиск во всех текстах" со всеми взведенными флажками, причем какую именно строку искать - неважно, существенным является сам факт нормального завершения поиска.

После окончания проверки можно сделать новую резервную копию взамен существующей и хранить ее до следующего исправления конфигурации – это должно защитить Вас и Ваших клиентов от неприятностей, связанных с внезапной потерей работоспособности системы.

Еще одной причиной появления ошибок в файле конфигурации может быть попытка внесения изменений вручную, при помощи различных редакторов. Подобные попытки, скорее всего, приведут к тяжелым последствиям, вплоть до полной невозможности восстановить хотя бы частично файл конфигурации. Для внесения исправлений в конфигурацию пользуйтесь предназначенным для этого средством - режимом запуска "Конфигуратор".

В заключение, хотелось бы отметить еще один аспект, связанный с целостностью информационной базы в целом. Файл конфигурации, а точнее состав содержащихся в нем метаданных, полностью определяет структуру таблиц информационной базы (но не наоборот). Можно сказать, что файл конфигурации является основой ИБ. Поэтому совершенно недопустима попытка копирования файла, содержащего "желаемую" конфигурацию, в каталог, содержащий "желаемые" данные, находящиеся в таблицах ИБ. Копируемый файл подразумевает некую структуру таблиц, которая практически наверняка не совпадет с реальной. Это приведет к нарушению целостности ИБ и, как следствие, к потере работоспособности, хотя и сам файл конфигурации и таблицы по отдельности могут не содержать ни единой ошибки. К сожалению, такое случается, когда разработка конфигурации идет параллельно с работой пользователей. Разработчик конфигурации приносит новый вариант и просто копирует его в каталог информационной базы, а это недопустимо по изложенным выше причинам. Для осуществления параллельной разработки реализован специальный режим Конфигуратора - "Загрузка измененной конфигурации".

Размер файла внешнего отчета

Данный раздел представляет интерес, прежде всего, для разработчиков, создающих внешние отчеты для массового тиражирования. В разделе приводится рекомендация, каким образом можно уменьшить размер файла внешнего отчета.

Диалоги форм внешних отчетов, как и любые другие диалоги форм, могут содержать поля ввода, имеющие не только базовый тип (строка, число, дата), но и тип любого типообразующего элемента метаданных (справочника, документа, перечисления и т.д.), из числа определенных в конфигурации, являющейся текущей во время создания этого элемента диалога. В связи с этим файл внешнего отчета содержит необходимую информацию о типообразующих элементах метаданных текущей конфигурации. Эта информация используется при загрузке формы внешнего отчета во время работы с конфигурацией, отличной от той, в которой внешний отчет создавался. Она служит для привязки типа данных, связанного с полем ввода диалога к одноименному, т.е. имеющему совпадающий идентификатор, элементу метаданных текущей конфигурации. Размер информации о типообразующих элементах метаданных конфигурации напрямую зависит от их количества в конфигурации и иногда может приводить к необоснованному увеличению размера файла внешнего отчета. Чтобы этого избежать, существует возможность создавать внешние отчеты, загрузив или

специально создав конфигурацию, имеющую лишь те типообразующие элементы метаданных, которые используются этим внешним отчетом. Причем создавать внутреннюю структуру элемента метаданных, т.е. создавать реквизиты для справочников, значения для перечислений и т.д., не является обязательным, т.к. в данном случае, значение имеет лишь идентификатор типообразующего элемента метаданных.

Например, форма диалог формы внешнего отчета имеет несколько полей ввода простых типов и одно поле ввода типа справочник Товары. Для того, чтобы внешний отчет имел минимальный размер и, в то же время, при загрузке во время работы в любой конфигурации, имеющей справочник Товары, тип данных поля ввода автоматически привязывался к данному справочнику, следует выполнить следующие действия:

Создать новую информационную базу в пустом каталоге

Войти в нее в режиме запуска "Конфигуратор"

Создать новый справочник, задав ему идентификатор Товары, или скопировать из другой конфигурации.

Создать новый внешний отчет и сохранить его.

Аналогично можно поступить и с уже существующим внешним отчетом, загрузив его в сокращенной (или пустой) конфигурации, формально что-то изменить, чтобы взвести флаг модифицированности, и затем сохранить.

Для внешних отчетов, не имеющих полей ввода, жестко привязанных к типу (например, тип "справочник неопределенного" вида не является жестко типизированным) или имеющих только поля базовых типов, рекомендуется сохранить внешний отчет, воспользовавшись запуском системы в режиме Конфигуратора в пустой конфигурации.

**Размер файла конфигурации**

Данный раздел представляет интерес, прежде всего, для разработчиков, создающих дистрибутив тиражируемой конфигурации.

Периодически на линию консультаций поступают обращения, в которых пользователи 1С:Предприятия выражают удивление по поводу труднопрогнозируемого изменения размера файла конфигурации (1cv7.md). Можно процитировать такую, типичную выдержку из письма : "После удаления из модуля достаточно большого фрагмента текста и сохранения конфигурации, мы ожидали некоторого уменьшения размера файла 1cv7.md. Однако, размер увеличился (!) на 350 байт. После этого в тот же модуль было добавлено несколько строк, но после сохранения размер файла не изменился. Тщательный просмотр текста модуля и запуск в режиме "1С:Предприятие" показывают, что изменения в тексте сохранены правильно, однако такое поведение файла 1cv7.md наводит на мысли о возможных ошибках в нем. Как вы можете прокомментировать такую ситуацию?".

Такое "поведение" файла конфигурации является НОРМАЛЬНЫМ. Дело в том, что используемая для работы с файлом конфигурации в 1С:Предприятии технология позволяет рассматривать его, как набор фрагментов, каждый из которых хранит свою часть данных о конфигурации. Такой подход позволяет в ряде случаев сократить время на сохранение конфигурации, т.к. переписывается не весь файл целиком, а только его фрагменты, хранящие информацию о тех частях конфигурации, которые реально были изменены в процессе редактирования. Перезапись фрагмента состоит из записи обновленной информации в специально отведенное дополнительно под нее место в файле конфигурации и отметке того места, где хранилась устаревшая информация, как свободного. Таким образом, внутри файла 1cv7.md образуются "пустоты", которые могут быть использованы при последующих записях информации об этой же или какой-либо другой части конфигурации, т.е. происходит фрагментация файла конфигурации.

Заметим, что, если при редактировании конфигурации были внесены изменения, которые могли потребовать реорганизацию информационной базы, то при сохранении файл конфигурации создается "с нуля" и в результате не является фрагментированным. Характерным признаком этого является появление сообщения о том, что "при сохранении метаданных будет произведен анализ сделанных изменений и, в случае необходимости, реорганизация информационной базы".

Для того, чтобы принудительно дефрагментировать файл конфигурации, рекомендуется сделать загрузку измененной конфигурации, указав в качестве загружаемого файл конфигурации текущей информационной базы.

**Использование синонимов объектов метаданных**

При создании объекта метаданных рекомендуется сразу указывать синоним.

В качестве синонима обычно используется краткое, но понятное пользователю обозначение объекта метаданных. Синоним следует задавать, если идентификатор объекта метаданных имеет большую длину или включает несколько слов. В последнем случае в синониме обычно те же слова, которые заданы в идентификаторе слитно, указываются через пробел или через точку.

Ввод синонима сразу при создании объекта метаданных целесообразен потому, что система использует введенный синоним при автоматическом формировании различных элементов интерфейса конфигурации.

Например, если создать новый вид документов и указать для него синоним, то при включении в меню пользовательского интерфейса команды ввода этого документа в качестве названия пункта меню будет подставлен именно синоним. А если синоним не будет указан, то в качестве названия пункта меню будет подставлен идентификатор и вероятнее всего название придется исправлять. Аналогично, если при создании реквизита документа указать синоним, то он будет использован системой при вставке реквизита в форму документа для подстановки имени реквизита шапки или заголовка колонки реквизита табличной части.

**Использование шаблонов фрагментов модулей**

Рекомендации по использованию шаблонов фрагментов модулей

При интенсивной разработке конфигураций весьма полезным инструментом является механизм шаблонов. Он позволяет создать заготовки фрагментов модулей, которые затем будут использоваться при написании алгоритмов. В этом разделе приводятся некоторые рекомендации по созданию и использованию шаблонов.

Для специалиста, работающего с несколькими конфигурациями, рекомендуется установить единый файл шаблонов для различных конфигураций. Это имя файла указывается в параметрах системы (меню "Сервис - Параметры" – закладка "Синтакс-Помощник" - поле "Файл шаблонов"). Если значение этого параметра установлено, то для всех конфигураций на данном компьютере будет использоваться единый набор шаблонов. В этом случае создание нового шаблона для часто используемой конструкции будет более оправдано, так как его можно будет использовать во всех конфигурациях.

При разработке шаблонов рекомендуется разбивать их на группы. Команда "Новая группа" вызывается из меню "Действия" или из контекстного меню окна редактирования списка шаблонов.

Для некоторых шаблонов имеет смысл задать строку в поле "Заменять". В этом случае шаблон будет автоматически вызываться при наборе в тексте указанного слова. Однако эту установку следует использовать осмотрительно, так как если задать слово, которое часто применяется в различных конструкциях, которые не имеют отношения к данному шаблону, то автоматический вызов будет мешать набору модулей.

Из всех шаблонов можно выделить 10 шаблонов, которые будут включаться в контекстное меню при редактировании текстов модулей. Разумеется, признак "Включать в меню" имеет смысл устанавливать для часто используемых шаблонов. Признак "Включать в меню" задается только для тех шаблонов, для которых указана строка "Заменять".

При создании шаблонов с переменной частью рекомендуется использовать управляющие конструкции, которые позволяют организовать ввод переменных частей шаблона при его использовании. Для вставки в шаблон управляющей конструкции следует воспользоваться контекстным меню, меню "Действия" или клавишами Ctrl+Shift+C. Основная управляющая конструкция "Произвольный запрос" позволяет ввести строку, которая будет помещена в том месте шаблона, в котором начинается управляющая конструкция. Рекомендуется задавать в управляющей конструкции подсказку. Это облегчит использование шаблона. При использовании конструкции запроса в шаблоне существует возможность указать в тексте шаблона два и более раз одну и ту же конструкцию (с одной и той же подсказкой). В этом случае запрос будет выдан только один раз, а его результат будет помещен во все места шаблона, в которых указан данный запрос.

В большинстве шаблонов имеет смысл также разместить управляющую конструкцию "Установка курсора". Она будет определять то положение, в которое будет помещен текстовый курсор после ввода шаблона. Использование этой конструкции позволит избежать дополнительных манипуляций курсором при использовании шаблона.

Весьма эффективным средством при использовании шаблонов является применение управляющих конструкций "Выбор метаданного". Это позволяет создавать шаблоны, при использовании которых будут выбираться объекты метаданных и подставляться в формируемый шаблоном фрагмент модуля. Например, это позволяет сделать шаблон, выполняющий обход справочника. При использовании данного шаблона достаточно будет выбрать из списка справочников нужный вид и в модуль будет вставлен фрагмент, выполняющий обход справочника выбранного вида.

Отдельной управляющей конструкцией является "Форматная строка". Она позволяет существенно упростить задание форматной строки, так как представляет все возможные варианты ее заполнения в диалоговой форме. Эту конструкцию используют в основном для создания шаблона, формирующего собственно вызов функции "Формат".

При создании шаблонов следует учитывать, что управляющие конструкции позволяют вставить непосредственно вводимую или выбираемую строку, а все "окружение" вставляемой конструкции должно быть задано непосредственно в шаблоне. Например, если вставляется управляющая конструкция для подстановки вида документа в функцию "СоздатьОбъект", то в шаблоне должно быть указано и слово "Документ" и точка и двойные кавычки, а вставляться будет только собственно идентификатор вида документа.

Заметим, что некоторые конструкции языка можно не выносить в шаблоны, а использовать соответствующие пункты Синтакс-Помощника. Например, из Синтакс-Помощника можно вставить практически все предопределенные процедуры. Создание аналогичных шаблонов целесообразно либо для сокращения времени поиска необходимой конструкции, либо для дополнения вставляемого фрагмента по отношению к тому, который содержится в Синтакс-Помощнике.

**Расположение файла внешнего отчета**

При открытии в Конфигураторе внешнего отчета через список последних открытых файлов в меню "Файл", иногда требуется убедиться, что для редактирования открывается именно тот файл, который необходим. Если посмотреть на заголовок окна открытого внешнего отчета, то до первого сохранения этого отчета в данном сеансе работы с ним в заголовке окна выдается полный путь к файлу отчета (после сохранения - только имя файла).

**Поиск нужного объекта в пользовательском интерфейсе**

Одна из задач, стоящих перед разработчиком конфигурации, - создание пользовательского интерфейса (или набора таких интерфейсов для разных категорий пользователей).

При разработке пользовательских интерфейсов для сложных конфигураций, содержащих большое число объектов, одна из типичных проблем, которую приходится решать, - определение того, включен ли уже какой-либо объект метаданных в пользовательский интерфейс, или еще нет.

Простым просмотром меню или панелей инструментов такую задачу зачастую не решить. Дадим несколько советов, которые позволят облегчить процесс поисков.

Чтобы определить, включен ли интересующий вас объект метаданных - справочник, документ, отчет, журнал документов - в какой-либо интерфейс, попробуйте создать в этом же интерфейсе команду для работы с этим объектом метаданных.

Удобно это делать в редакторе меню. Откройте любое меню верхнего уровня и дважды щелкните мышью по служебному элементу <новый…>. В палитре свойств заполните поле "Объект", выбрав интересующий вас объект метаданных. Если для выбранного объекта метаданных существует несколько команд для работы с ним - укажите одну из них в поле "Команда". Теперь нажмите кнопку "Обновить".

Если указанный объект уже присутствует в меню, будет выдано сообщение "Данное меню уже содержит аналогичную команду". Можно нажать кнопу "Отмена", чтобы отказать от добавления нового элемента меню: мы убедились что интересующий нас объект метаданных уже присутствует в пользовательском интерфейсе.

Однако, если для объекта метаданных предусмотрены несколько вариантов заполнения поля "Команда" (несколько команд вызова из меню), то успешное включение объекта метаданных в меню может еще ничего не означать - возможно, что в меню уже присутствует другая команда для работы с этим же объектом метаданных.

Чтобы определить это, воспользуйтесь другим приемом.

Для пользовательского интерфейса, в котором требуется "искать" объект метаданных, создайте временный подчиненный интерфейс с любыми свойствами, и откройте его для редактирования. Подчиненный интерфейс содержит все команды из интерфейса родителя, отсортированные в алфавитном порядке: сначала по типам объектов метаданных, внутри типов - по идентификаторам объектов, затем - по командам для работы с объектом, а если все это одинаковое - то по параметрам команд.

Таким образом, можно легко найти и интересующий вас объект метаданных, и увидеть все команды для работы с этим объектом.

Следует напомнить, что редактировать вы все равно будете интерфейс-родитель, а подчиненный интерфейс, пока не закончена разработка, можно не удалять, а использовать "для контроля".

**Определение количества строк модулей в конфигурации**

Иногда у разработчика конфигурации или специалиста, анализирующего какую либо конфигурацию, возникает необходимость оценки размера конфигурации. В данном случае под размером подразумевается не размер файла 1cv7.md, а объем прикладного решения, которое представляет собой конфигурация. Например, это может быть необходимо для косвенной оценки сложности конфигурации или трудоемкости ее поддержки и модификации.

Наиболее очевидным показателем объема конфигурации является количество объектов метаданных, однако это не всегда позволяет оценить сложность их реализации. Во многом сложность модификации и сопровождения конфигурации зависит от объема алгоритмов написанных на встроенном языке 1С:Предприятия. Для того чтобы оценить объем алгоритмов можно воспользоваться следующей методикой.

В режиме запуска "Конфигуратор" следует воспользоваться режимом поиска во всех текстах (меню "Конфигурация - Поиск во всех текстах"). В поле "Искать в" следует включить только флажок "Модулях", а в поле "Искать:" ввести символ ";". Далее следует собственно выполнить поиск. При этом будут проанализированы все модули конфигурации и подсчитано число символов ";".

Разумеется, не все строки модулей кончаются символом ";", а иногда в одной строке может быть несколько таких символов. Однако полученное число можно считать приблизительной оценкой количества строк и использовать для оценки объема конфигурации.

Следует заметить, режим поиска во всех текстах имеет ограничение по количеству отображаемых вхождений строки поиска. То есть с некоторого момента (после 32000 вхождений) найденные строки не будут отображаться, однако общее число вхождений будет подсчитано и выдано в окне результатов поиска.

**Особенности работы с базовыми версиями 1С:Предприятия**

Среди продуктов системы программ 1С:Предприятие существуют Базовые версии:

1С:Налогоплательщик 7.7

1С:Аспект 7.7 Компактная торговая система

1С:Бухгалтерия 7.7 (базовая версия)

1С:Бухгалтерия 7.7 Базовая версия для Украины

1С:Зарплата и кадры (базовая версия)

1С:Платежные документы 7.7

1С:Деньги 7.7

1С:Воинская часть 7.7

Особенностью данных продуктов является отсутствие режима конфигурирования.

Технологически это организовано следующим образом.

В комплект поставки данных продуктов входят специальные конфигурации, в которых проставлен признак, обеспечивающий возможность работы 1С:Предприятия с этими конфигурациями в качестве Базовой версии. Данный признак проставляется фирмой 1С. Работая с такой конфигурацией 1С:Предприятие не выполняет стандартную процедуру проверки ключа аппаратной защиты. Для некоторых конфигураций вместо этого применяется защита по ключевым словам документации. При работе 1С:Предприятия в режиме базовой версии для изменение конфигураций не допускается. Исключение составляет загрузка измененной конфигурации, полученной из фирмы 1С в качестве обновления.

При использовании данных конфигураций с обычной версией 1С:Предприятия (при установленном ключе аппаратной защиты) они могут быть изменены также, как и обычные конфигурации. Однако при записи конфигурации признак базовой конфигурации будет стерт и конфигурация не будет в дальнейшем работать с базовой версией 1С:Предприятия.

Таким образом, после покупки базовой версии пользователь в любой момент может приобрести версию 1С:Предприятия допускающую возможность конфигурирования, и начать модифицировать имеющуюся у него конфигурацию.

В качестве базовой версии 1С:Предприятия выступают программные файлы обычной несетевой версии (1cv7l.exe). Функционирование в режиме базовой версии обеспечивается при работе со специальной конфигурацией при отсутствии ключа аппаратной защиты.

С организациями, выполняющими разработку тиражных конфигураций фирма 1С может заключить специальный договор, предоставляющий возможность продажи одной из базовых версии 1С:Предприятия с конфигурацией разработанной этой организацией. Этот договор рассчитан на случаи, когда разработанная конфигурация предназначена для массового использования с базовой версией 1С:Предприятия.