Приложение 4 в Договору оказания услуг № ГПХ

Внутренний стандарт разработки

Developers

Exported on Sep 28, 2022

Table of Contents

1 Общее представление 3

2 Правила создания объектов метаданных 4

2.1 Наименование 4

2.2 Синонимы 4

2.3 Префиксация 5

2.4 Сортировка 5

2.5 Регламентные задания 5

3 Правила создания ролей 6

3.1 Согласно должностной инструкции пользователя 6

3.2 По подсистемам конфигурации 6

3.3 По объектам конфигурации 6

3.4 При сервисной необходимости 6

3.5 Распределение ролей по группам 7

4 Правила внесения изменений в код 8

4.1 Оформление области изменения стандартных модулей 8

4.2 Пометка для будущего изменения в коде 8

4.3 Добавление новых проверок в объекты конфигурации 8

4.4 Оформление 8

4.5 Поддержание строгой инкапсуляции 11

5 Правила работы с собственными общими модулями 14

6 Правила применения механизма отправки уведомлений 15

7 Правила работы с отдельными элементами справочников 17

7.1 Правила работы с хранилищем предопределенных значений 17

8 Удаление и перенос устаревших объектов конфигурации 19

9 Отказ от использования модальности окон 20

# Общее представление

Внутренний стандарт представляет собой набор правил и соглашений, применяемых при создании объектов конфигурации и написании исходного кода. Цель стандарта - облегчить понимание и поддержку конфигурации.

Общая методика разработки и правила оформления исходного кода в 1С приведена на сайте ИТС. Для входа в систему требуется корпоративная учетная запись пользователя ИТС.

|  |
| --- |
| "Примечание" Логин и пароль учетной записи пользователя ИТС уточняются у руководителя НРБП, заместителя руководителя НРБП и у консультанта по бизнес-приложениям. |

|  |
| --- |
| совет" Для использования нижеприводимых шаблонов комментариев рекомендуется создание на основе пользовательских шаблонов текста. |

# Правила создания объектов метаданных

## Наименование

При именовании новых объектов метаданных конфигурации требуется придерживаться следующих правил:

* **Справочники**
Имена справочников задаются во множественном числе и образовываются от описания списка объектов, значения которых хранятся в справочнике

|  |
| --- |
| "Например" \* Валюты \* Пользователи |

* **Планы видов характеристик**
Имена планов характеристики требуется задавать во множественном числе и образовывать от описания списка объектов, которые перечисляются в плане видов характеристик

|  |
| --- |
| "Например" \* ВидыДоступа \* ВопросыДляАнкетирования |

* **Перечисления**
Имена перечислений в конфигурации задаются во множественном числе;
* **Документы**
Имена документов требуется задавать в единственном числе;
* **Журналы документов**
Имена журналов документов задаются во множественном числе и образовываются от описания списка объектов, которые содержатся в журнале

|  |
| --- |
| "Например" \* СкладскиеДокументы \* КорректировкиНДС |

* **Регистры сведений**, **регистры накоплений** Имена регистров сведений, регистров накопления требуется задавать во множественном числе и образовывать от описания списка записей, которые содержатся в регистре

|  |
| --- |
| "Например" \* ДокументыФизическихЛиц \* ДвиженияДенежныхСредств |

## Синонимы

В синоним объекта метаданных не допускается применение префикса (например, правильно - **Квалификационный акт разделки** (см. рис. 1), неправильно - **ПРС: Квалификационный акт разделки**).



**Рис. 1.** Именование объектов метаданных (за исключением роли)

Исключением из правила считаются синонимы ролей, для которых допускается содержать префикс в синониме (см. рис. 2).



**Рис. 2.** Именование роли

## Префиксация

Новые объекты метаданных именуются с обязательным префиксом

|  |
| --- |
| "Примечание" Префикс **ПРС\_** применяется только при разработке в центральной базе "1С: УПП", в остальных базах требуется присутствие префикса в произвольной форме |

|  |
| --- |
| "Например" \* **ПРС\_** - префикс объект конфигурации "Управление производственным предприятием" \* **кпп** - префикс объектов конфигурации "Контрольно-пропускной пункт" \* **уоп** - объекты программы "Управление оперативным производством" \* **унг** - объекты программы "Управления номерным фондом" |

Объекты крупных подсистем, а также внедренные решения внешних разработчиков, допустимо дополнять собственным префиксом

|  |
| --- |
| "Например" \* **ПРС\_Судоремонт** - объекты блока подсистемы судоремонта \* **упл** - объекты внедренной подсистемы путевых листов). |

## Сортировка

После добавления нового объекта метаданных требуется выполнить сортировку в дереве объектов того же вида. Сортировка выполняется по принципу: собственные объекты располагать в конце списка и отсортировать по алфавиту.

|  |
| --- |
| "Внимание!" При сортировке объектов в дереве конфигурации недопустимо использовать автосортировку, так как при такой сортировке «самописные» объекты перемешиваются с типовыми. |

## Регламентные задания

При создании собственных регламентных заданий процедуры-обработчики заданий требуется размещать в общем модуле УправлениеРегламентнымиЗаданиями.

# Правила создания ролей

Перед предоставлением прав доступа к любым объектам конфигурации, требуется принять решение о детализации прав. Права предоставляются с разной степенью точности. Ниже описываются варианты и требования к именованию ролей.

## Согласно должностной инструкции пользователя

* Роль предоставляется при требовании ограничения доступа к объектам согласно должностным обязанностям целевого пользователя
* Имя содержит префикс и название должности

|  |
| --- |
| "Пример" **ПРС\_ГлавныйМеханик** |

## По подсистемам конфигурации

* Роль призвана обеспечить доступ конкретным блокам программы
* Имя содержит префикс и название целевой подсистемы. Допускается дополнение имени словами **Учет**, **Использование**

|  |
| --- |
| "Пример" **ПРС\_ИспользованиеКомпенсацийИЛьгот****ПРС\_УчетКоммерческихЗатрат****ПРС\_БизнесСобытияИПроцессы** |

## По объектам конфигурации

* Максимально детализированная роль, предоставляющая доступ на чтение, добавление, изменение к одному объекту метаданных (справочник, документ и т.д.)
* Имя содержит префикс, имя целевого объекта метаданных и слово, описывающее вид доступа:
	+ **Чтение** - доступ на чтение или просмотр объекта
	+ **Изменение** - модификация и просмотр данных
	+ **ДобавлениеИзменение** - права на просмотр, изменение и добавление новых объектов

|  |
| --- |
| "Пример" **ПРС\_СудоремонтДобавлениеИзменениеРабот****ПРС\_ДобавлениеИзмененийДолжностныхИнструкций** |

|  |
| --- |
| совет" Если в задаче требуется предоставить доступ только на добавление и изменение объектов, то необязательно предоставить сразу три вида доступа (Чтение, Изменение, ДобавлениеИзменение). В таком случае применяется роль ПРС\_ДобавлениеИзменение. |

## При сервисной необходимости

* Роль для предоставления служебных и сервисных прав
* Имя содержит префикс и ключевое слово **Право**

|  |
| --- |
| "Пример" **ПРС\_ПравоРаспределенияВнутреннихЗаказовПоИсполнителям** |

## Распределение ролей по группам

Крупные блоки ролей требуется разделить с помощью отдельных групп (см. рис. 3). Разделение делается с указанием группы конкретной роли в общем макете ОписаниеРолейКонфигурации.



**Рис. 3.** Распределение ролей по группам

|  |
| --- |
| "Важно" Роли, относящиеся к подсистеме судоремонта, требуют обязательной группировки. |

# Правила внесения изменений в код

## Оформление области изменения стандартных модулей

При изменении стандартных модулей конфигурации требуется обрамить измененную секцию кода. Пример обрамления представлен в листинг 3.

// --> PRS
<модифицированная секция кода>
// <-- PRS

**Листинг 1.** Оформление области изменения стандартных модулей

## Пометка для будущего изменения в коде

Для пометки предстоящих изменений в конкретном участке кода применяется конструкция комментария, приведенного на листинг 4.

// TODO:\_описание\_того\_что\_нужно\_сделать
<секция кода>

**Листинг 2.** Комментарий-пометка для будущего изменения в коде

## Добавление новых проверок в объекты конфигурации

В случае добавления новых проверок, запрещающих выполнять какие-либо действия над объектами (Запись\Проведение\etc), необходимо так же выполнить одно из следующих требований:

* Добавить обработчик обновления, который проведёт требуемые манипуляции с существующими объектами для того что бы действие над объектом (Запись\Проведение\etc) могло выполнятся так же как и до добавления новой проверки

|  |
| --- |
| "Например" Заполнение значения по-умолчанию в новом поле, обязательном для заполнения |

* Проверка должна быть опциональной и включатся с определённой даты, так что бы не воздействовать на уже существующие объекты

|  |
| --- |
| "Например" Добавить новый раздел учёта с датой начала проверки |

Ключевой смысл данного требования в том, что если объект был ранее записан или проведён, он продолжал записываться или проводится и после обновления.

## Оформление

* Длина строки в общем случае не должна превышать 120 символов.
* Для отступов необходимо использовать символы табуляции.
* Одна строка кода - одна управляющая конструкция.



* Следует отделять друг от друга пробелами ключевые слова, вызовы процедур и функций, параметры процедур и функций внутри скобок, операторы



* Для разделения на логические части внутри модуля следует использовать пустые строки
* При вызове функции с несколькими параметрами при переносе строк необходимо выравнивать параметры по первому



* Выравнивание однотипных операторов. При следовании друг за другом нескольких однотипных операторов допускается их выравнивание. Выравнивание следует выполнять с помощью пробелов.



* Параметр функции не должен возвращать значение. Иными словами не используйте входные параметры функций как дополнительный вывод. Весь вывод должен быть в возвращаемом значении. Если нужно возвращать несколько значений следует использовать такие типы как Структура, Массив и т.д.



* Предпочтительней использовать тернарный оператор для простых конструкций.



* Не допускайте использования вложенных тернарных операторов.
* Ключевое слово Тогда пишется на той же строке, что и последнее условие.
* Сложные условия (содержащие 3 конструкции и более) необходимо выносить в отдельные методы.



* Избегайте использования Йода-синтаксиса



* Избегайте неявного определения переменных



* Следуйте общему подходу именования



* Не используйте отрицание в именах переменных и методов



* К комментариям также относятся ограничения на длину строки в 120 символов.

## Поддержание строгой инкапсуляции

* Логика работы с объектами должна принадлежать самим объектами, код который занимается обработкой объекта следует располагать в МО\ММ самого объекта
* Реквизиты\Измерения\Ресурсы\etc объявляются деталями технической реализации объекта, обращение к этим деталям за пределами самого объекта запрещено.



* При необходимости использования данных объекта в запросах, требуется реализовать программный интерфейс, который сформирует необходимые временные таблицы для запроса:





* Для целей создания объектов требуется сделать специализированный программный интерфейс. Например:



* Так же допустимо заполнение через описание контракта и метода Заполнить().Например:



* Так же можно воспользоваться [строителем](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D0%BE%D0%B8%D1%82%D0%B5%D0%BB%D1%8C_%28%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%29). Например: 

# Правила работы с собственными общими модулями

При создании собственных общих модулей в заголовке модуля перед списком внесенных изменений указываются:

* Владелец;
* Имя модуля;
* Краткое описание модуля.

Шаблон оформления модуля приведен в листинге 3.

// -------------------------------------------------------------------------
// (С)<год> ГК «Доброфлот»
//
// Имя модуля:
// <имя модуля>
//
// Краткое описание:
// <описание модуля>
// -------------------------------------------------------------------------

**Листинг 3.** Оформление собственного общего модуля

Процедуры и функции общих модулей создаются с блоками описания параметров и описания возвращаемых значений. Шаблон и пример оформления представлены в листингах ниже.

// Описание процедуры/функции
//
// Параметры
// ИмяПараметра1 – ТипПараметра1 – ОписаниеПараметра1
// ИмяПараметра2 – ТипПараметра2 – ОписаниеПараметра2
...
// ИмяПараметраN – ТипПараметраN – ОписаниеПараметраN
//
// Возвращаемое значение:
// ТипВозвращаемогоЗначения – ОписаниеВозвращаемогоЗначения
//

**Листинг 4.** Оформление заголовков процедур и функций

// Сохраняет файл на диск, исключая ошибки, связанные с не поддерживаемыми в
// файловой системе именами файлов.
//
// Параметры:
// Данные – ДвоичныеДанные – данные файла, которые создаются в хранилище
// ИмяФайла – Строка – имя сохраняемого файла
//
// Возвращаемое значение:
// Строка – полное имя файла
//
Функция ЗаписатьФайлНаДискБезопасно(Данные, ИмяФайла) Экспорт

**Листинг 5.** Пример оформления заголовка функции

# Правила применения механизма отправки уведомлений

Механизм отправки уведомлений основан на методах общего модуля УправлениеУведомлениями. Примеры возникающих ситуаций, связанных с отправкой уведомлений и имена используемых методов общего модуля УправлениеУведомлениями приведены ниже:

1. Требуется интерактивно (на стороне клиента) отправить уведомление по произвольному ссылочному объекту:
	* СоздатьКнопкуОтправкиУведомленияЗаинтересованнымЛицам - динамически создает кнопку отправки уведомлений в произвольном объекте (см. рис. 4);



**Рис. 4.** Отправка уведомления произвольному объекту

* + ОтправитьУведомлениеЗаинтересованнымЛицам - переопределяет получателей, тему или текст уведомления (см. рис. 5). 
	**Рис. 5.** Отправка уведомления с указанием получателей и произвольным текстом
1. Требуется отправить автоматическое уведомление по объекту при наступлении событий проведения, изменения значимых реквизитов и т.д:
	* ОтправитьУведомлениеПоОбъекту - дополнительное включение в параметрах функции хранения истории отправки уведомлений и установление проверки на повторную отправку одного и того же уведомления.
2. Требуется отправить уведомление по объекту в ручном режиме:
	* ОтправитьУведомлениеПоОбъекту - если перед отправкой требуется интерактивно изменить список получателей и (или) добавить уведомление комментариями, то для этого используется функция РедактироватьУведомлениеИнтерактивно.
3. Требуется отправить уведомление, которое не привязано к конкретному объекту программы:
	* ОтправитьПроизвольноеУведомление.

В качестве получателей уведомлений выступают элементы следующих справочников:

* Пользователи
* ФизическиеЛица
* ПодразделенияОрганизаций
* Склады
* ПРС\_ПолучателиУведомлений

|  |
| --- |
| "Примечание" Справочник ПРС\_ПолучателиУведомлений используется для создания предопределенных элементов, в табличной части которых в виде строки указаны адреса получателей (справочник заменит отдельные элементы справочника ПРС\_ХранилищеПредопределенныхЗначений). |

# Правила работы с отдельными элементами справочников

Для получения элементов или наборов элементов отдельных справочников используется следующий подход:

1. В справочнике ПРС\_ХранилищеПредопределенныхЗначений требуется создать предопределенный элемент (например, ГруппаНоменклатурыКонсервы);
2. Создать функцию-обертку для получения предопределенного значения в общем модуле РаботаСХранилищемПредопределенныхЗначений. Для получения используется сервисная функция ПолучитьИнформациюИзХранилища, которая расположена в том же модуле.
3. В коде вызвать функцию-обертку для получения нужного значения.

|  |
| --- |
| "Внимание!" При написании кода недопустимо применение конструкции вида Справочник.ИмяСправочника.НайтиПоКоду или НайтиПоНаименованию. |

|  |
| --- |
| "Примечание" Часть значений, касающихся периодов учета и признака необходимости ведения учета перенесена в регистр сведений Настройки разделов учета. |

## Правила работы с хранилищем предопределенных значений

В хранилище предопределенных значений добавляются значения, которые остаются неизменными, состоят из одного и того же значения или изменяются редко и не поддерживаются в актуальном состоянии.

Если значения заводятся для контрольных функций, то такие настройки рекомендуется выносить в сами объекты созданием новых реквизитов у объектов.

|  |
| --- |
| "Например" Для значений **Подразделения, для которых контролируются сроки исполнения внутренних заказов** или **Список подразделений, для которых осуществляется контроль в заявках на снабжение** лучше завести реквизиты **Контролировать срок исполнения во внутреннем заказе** в справочнике **Подразделений**. |

В наименовании значения требуется четко отражать цель предназначения. Наименование **Список подразделений, для которых осуществляется контроль в заявках на снабжение** считается неверным, так как непонятно что контролировать в подразделениях.

Запрещено добавлять в хранилище значения:

1. Касающиеся периодов начала учета или признака о необходимости учета. Такие данные требуется добавлять через регистр сведений **Настройки разделов учета**;
2. Пользователей, которые получают уведомления. Для этого требуется применение справочника **Получатели уведомлений**;
3. Пользователи, имеющие права доступа над конкретными действиями. Для этого используются:
	* **Роль конфигурации** - для ограничения прав доступа к объектам (справочникам, документам и т.д.). Роль определяет доступность действий пользователя над конкретными объектами метаданных.
	* **Дополнительные права** - если требуется ограничить доступ к отдельной функциональности в рамках одного объекта метаданных (например, запретить выполнение конкретного действия или видеть информацию в интерактивном режиме).
	* **Исполнителей ролей** - рекомендуется использовать для участия пользователя в бизнес-логике (например, доступность перевода статусов в документах, либо для разделения доступа внутри одного документа).
4. По справочникам. Предопределенный элемент требуется добавить в самом справочнике (например, **Вид корреспонденции Запрос уточнение**).
5. Используемые только для внешних отчетов. Такие значению требуется добавлять через параметры отчета.

Главное правило по работе с хранилищем - быть культурным и убирать за собой. Значение, которое добавлено и в дальнейшем стало неактуальным, подлежит удалению вместе с функцией из общего модуля РаботаСХранилищемПредопределенныхЗначений. Перед удалением рекомендуется обязательная проверка использования удаляемого значения в основной конфигурации, во внешних обработках и в правилах по обмену.

# Удаление и перенос устаревших объектов конфигурации

Если при изменении структуры метаданных конфигурации планируется удалить объект метаданных (реквизит, измерение, ресурс и прочее), который связан с записями информационной базы, то требуется принять решение об удалении или переносе данных удаляемого объекта в новые структуры. Перенос данных в другие объекты выполняется соблюдением следующих правил:

1. Устаревшие объекты метаданных и реквизиты не удаляются из конфигурации безвозвратно, а помечаются как устаревшие. Для пометки к именам объектов добавляется префикс Удалить

|  |
| --- |
| "Например" Реквизит ОтказаноВПриеме переименовывается в УдалитьОтказаноВПриеме |

1. После обновления структуры метаданных требуется написать обработку переноса данных из устаревших реквизитов в новую структуру метаданных конфигурации.
2. Периодически выполнять окончательное удаление неиспользуемых объектов.

|  |
| --- |
| "Примечание" В синоним удаляемого объекта требуется добавить префикс (не использовать)Например: (не использовать) Основной договорТакже требуется отсортировать устаревшие реквизиты в самый низ дерева метаданных. |

# Отказ от использования модальности окон

Отказ от использования модальности окон в разработке обусловлен вызываемыми проблемами при тестировании, дружелюбности интерфейса для пользователя (не блокируется всё приложение), а так же в целях более грамотной утилизации ресурсов клиента.

|  |
| --- |
| "Внимание!" При реализации решений запрещается применять модальные окна. |