# [1С:Предприятие 8. Система стандартов и методик разработки конфигураций](https://its.1c.ru/db/v8std" \l "content:788:hdoc)

# Обзор новых систем стандартов разработки

## Цель доклада

Рассказать коллегам вкратце рассказать о важных нововведениях в систему стандартов разработки.

## Структура доклада

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Новые функции | **v** | **v** | **v** | **x** |
| Исправлены ошибки | **v** | **v** | **v** | **v** |

Здесь:

1. РР (редакция) – существенно нарушена совместимость (серьезные архитектурные или «знаковые» изменения в библиотеке);
2. ПП (номер подредакции) – нарушена совместимость (требуется отработать инструкцию по переходу на эту версию, иначе конфигурация будет неработоспособна), удалены устаревшие объекты метаданных;
3. ВВ (номер версии) – доступны только новые функции для пользователей и/или разработчиков (возможна «механическая» автозаливка метаданных библиотеки в конфигурацию-потребитель; инструкция по переходу на эту версию не обязательна, без ее отработки конфигурация сохранит работоспособность);
4. СС (номер сборки) – содержит только исправление ошибок (возможна «механическая» автозаливка метаданных библиотеки в конфигурацию-потребитель).

Удалять устаревшие объекты метаданных следует не чаще раза в год в версиях, запланированных к выпуску ближе к концу года, а удаляемые объекты должны существовать не менее двух лет. При этом обязательно нужно будет поменять номер подредакции

**Проверка прав доступа**

Для проверки прав доступа в коде следует использовать метод **ПравоДоступа**.

Такой подход позволяет повысить устойчивость кода к пересмотру состава ролей в конфигурации.

Но в тех случаях, где роль не дает никаких прав на объекты метаданных, а служит только для определения того или иного дополнительного права, следует использовать метод **РольДоступна**.

**Многократное выполнение однотипных запросов**

Всегда нужно стараться получить данные одним запросом, чтобы избежать запросов в цикле.

Допускается разделение запроса на серию запросов в следующих случаях:

* заметное усложнение разработки запроса для выбора сразу всех однотипных данных;
* деградация производительности запроса. Например, когда запрос содержит обращение к виртуальным таблицам, формируемым на определенную дату (как в случае с виртуальными таблицами остатков регистра накопления, срезом последних периодического регистра сведений и т.п.);
* порционная обработка большого объема данных
  + в обработчиках обновления;
  + в групповом проведении, изменении данных, удалении помеченных объектов, контроле ведения учета и т.п.;
  + при загрузке данных из файла или внешних источников, получении почты и т.п.

В этих случаях запросы в цикле:

* необходимы для порционности;
* либо они незначительно замедляют по сравнению с записью самих данных;
* либо они неустранимы на текущих средствах платформы (например, чтение и запись наборов регистра в цикле по значениям измерения).

**Особенности использования в запросах оператора ПОДОБНО**

1. Оператор ПОДОБНО в тексте запроса не следует формировать строку шаблона вычислениями или конкатенацией.
2. Длина строки шаблона не должна превышать 1024 символа
3. Сравнение выполняется без учета регистра символов.
4. Необходимо экранировать все спецсимволы (\_%[]^), чтобы они трактовались как часть искомого текста (а не как спецсимвол). Иначе если пользователь в строке поиска, например, введет текст со скобками [ ], то при выполнении запроса возникнет ошибка вида: ERROR: invalid regular expression: invalid character range (PostgreSQL).

Для экранирования необходимо:

* выбрать спецсимвол экранирования, например (~);
* установить спецсимвол перед символами операнда, например (~%). Чтобы экранировать выбранный спецсимвол, его необходимо задвоить, например (~~).
* и дополнить шаблон ключевым словом СПЕЦСИМВОЛ.

"100~%" СПЕЦСИМВОЛ "~"

**Стандартные роли в расширениях конфигурации**

При разработке расширений конфигурации не рекомендуется заимствовать и изменять роли **АдминистраторСистемы** и **ПолныеПрава**. Вместо этого необходимо добавить в расширение собственные роли по следующему принципу.

В простейшем случае, если права на все объекты расширения одинаковы для всех категорий пользователей, то создать одну роль с указанным именем, настроить в ней необходимые права на объекты расширения и включить ее в основные роли расширения:

* <ПрефиксРасширения>**ОбщиеПрава**

Если права на объекты расширения различаются для администратора и пользователей, то создать две роли с именами:

* <ПрефиксРасширения>**ПолныеПрава** (включить в основные роли расширения)
* <ПрефиксРасширения>**БазовыеПрава**

Если права на объекты расширения различаются для администратора, пользователей и внешних пользователей, то создать три роли с именами:

* <ПрефиксРасширения>**ПолныеПрава** (включить в основные роли расширения)
* <ПрефиксРасширения>**БазовыеПрава**
* <ПрефиксРасширения>**БазовыеПраваВнешнихПользователей**

**Использование журнала регистрации**

**Строковый идентификатор типа события.** Как правило, список типов событий в конфигурации может быть сколь угодно большим, поэтому типы событий рекомендуется группировать по функциональному признаку: «Название группы событий.Название события».

Тип события предназначен для отбора однотипных записей из журнала регистрации, поэтому неправильно включать в него любую избыточную информацию – конкретные значения из информационной базы или внешних систем. Например, неправильно:

* Не удалось отменить проведение документа <Реализация КП00-00002 от 11>
* В информационной базе не найден сотрудник Найденова, банковская карта 3333

Правильно помещать метаданные в параметр **Метаданные**, а данные – в параметр **Данные** метода **ЗаписьЖурналаРегистрации**, и использовать более общие типы событий:

* Проведение документов.Не удалось отменить проведение
* Сотрудники.В информационной базе не найден сотрудник

**Использование объектов типа Структура**

Не рекомендуется в разных частях программного кода добавлять в структуру свойства и затем проверять их наличие в структуре. Следует сразу создавать структуру со значением свойств по умолчанию. Помимо повышения читаемости, это позволяет эффективнее применять различные средства автоматической проверки и обработки кода.

Исключением можно считать структуры, формат которых не фиксирован:

* Структуры, полученные из внешних источников: результат HTTP-запроса, данные со сканера штрихкодов, терминала сбора данных и т.п.
* Системные параметры формы (\*), параметры выбора в параметрах формы.

**Открытие параметризированных форм**

Параметры формы следует объявлять явно на закладке **Параметры** редактора формы. В таком случае в коде обработчика **ПриСозданииНаСервере** не требуется проверять наличие свойств у структуры **Параметры**, а сам состав параметров формы явно задекларирован (поэтому их не требуется восстанавливать, изучая весь код обработчика **ПриСозданииНаСервере**).

Например, неправильно:

&НаСервере  
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)  
 ФамилияИмяОтчество = Неопределено;  
 Если Параметры.Свойство("ФамилияИмяОтчество", ФамилияИмяОтчество) Тогда  
 Объект.Наименование = ФамилияИмяОтчество;  
 КонецЕсли;  
КонецПроцедуры

правильно:

&НаСервере  
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)  
 Объект.Наименование = Параметры.ФамилияИмяОтчество;  
КонецПроцедуры

**Транзакции: правила использования**

В рамках транзакции нужно стремиться выполнять минимум действий – только те, которые нельзя в соответствии с бизнес-логикой выполнять вне транзакции. В частности:

* сложные, ресурсоемкие расчеты нужно стремиться делать до начала транзакции, если это позволяет бизнес-логика;
* если расчет должен выполняться в транзакции, то нужно стремиться сделать его как можно более простым. Например, контроль остатков можно делать уже после записи простым запросом к записываемому регистру;
* проверка заполнения объекта должна делаться вне транзакции;
* запросы, перед выполнением которых не нужно устанавливать блокировку данных, нужно стремиться выполнять до начала транзакции;
* запросы, выполняемые в рамках транзакций нужно стремиться оптимизировать
* не следует в транзакции вызывать внешние ресурсы (сетевые папки, интранет- и интернет-сайты, почтовые, FTP-, HTTP-, веб-сервисы и т.п.):
  + обращение к внешнему ресурсу может быть длительным;
  + ресурс может быть недоступен;
  + может произойти ошибка тайм-аута.

Обязательное использование транзакции:

1. В случае, если для ускорения операции записи в регистр используется отключение итогов, такую операцию вместе с отключением и включением итогов необходимо выполнять в транзакции, иначе в других сеансах может возникнуть ошибка при получении среза последних.

### Выводы

На сайте ИТС есть раздел с нововведениями в стандарты разработки, которые необходимо время от времени проверять, чтобы у клиентов всегда был продукт который работает быстро, а также его легко поддерживать