

## Предопределенные элементы в "1С:Предприятия 8.3"

При реализации алгоритмов разработчики часто опираются на определенные данные – элементы справочников, планов счетов, планов видов расчета и т.д. Во встроенном языке существуют методы для поиска данных, например, *НайтиПоКоду()* или *НайтиПоНаименованию()*.

Однако алгоритмы, опирающиеся на код или наименование, зачастую являются не надежными. Поскольку в пользовательском режиме код или наименование элемента справочника могут быть изменены, что может привести к неработоспособности алгоритмов.

Именно для решения этой проблемы и предназначены предопределенные данные – данные, созданные в конфигураторе, обратиться к которым возможно по имени, не прибегая к предварительному поиску элемента.

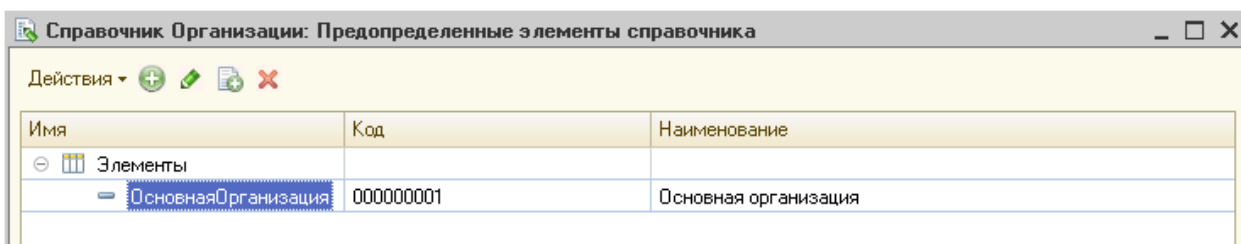
Таким образом, у предопределенных данных есть две “стороны”: во-первых, существует список предопределенных элементов, созданный в конфигураторе, а, во-вторых, для данных информационной базы указывается, является ли конкретный элемент предопределенным.

Предопределенные элементы могут быть созданы у:

- справочников;
- планов счетов;
- планов видов характеристик;
- планов видов расчета.

В статье рассмотрены новшества, касающиеся предопределенных данных на платформе 8.3, а также особенности работы с ними в распределенных базах (как центральных, так и периферийных) и в информационных базах в режиме разделения данных.

Для примера, создадим в справочнике *Организации* предопределенный элемент *ОсновнаяОрганизация*:

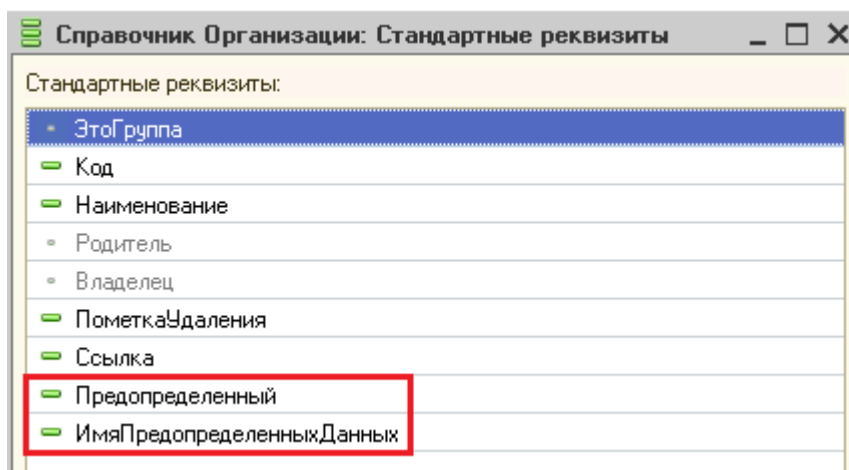


Обращение к этому элементу из программного кода будет следующим:

```
Справочники.Организации.ОсновнаяОрганизация
```

В платформе 8.3 реализована возможность связать предопределенные данные с элементами соответствующего типа.

Для этого у объектов, которые могут иметь предопределенные элементы (они указаны выше), добавлено новое свойство *ИмяПредопределенныхДанных*. Оно отображается в списке стандартных реквизитов:



Выберем при помощи запроса все поля из справочника *Организации*:

Ссылка	ВерсияДанных	ПометкаУдаления	Код	Наименование	ВалютныйУчет	Предопределенный	ИмяПредопределенныхДанных
ООО "Все для дома"	AAAAA0MAAA*	Нет	000000001	ООО "Все для дома"	Да	Нет	
ООО "Товары"	AAAAA0AAAA*	Нет	000000002	ООО "Товары"	Да	Нет	
ООО "1000 мячей"	AAAAA0AAAA*	Нет	000000003	ООО "1000 мячей"	Нет	Нет	
Основная организация	AAAAA0AAAA*	Нет	000000001	Основная организация	Нет	Да	ОсновнаяОрганизация

На рисунке видно, что в поле *ИмяПредопределенныхДанных* указан именно тот идентификатор, который мы ввели в режиме конфигуратора.

Предопределенный элемент в списке отображается специальной пиктограммой:

 Организации

Создать

Найти...

Отменить поиск

Наименование

- ООО "1000 мелочей"

- ООО "Все для дома"

- ООО "Товары"

-  Основная организация

Чтобы "отсоединить" элемент данных от элемента predetermined данных, нужно присвоить свойству *ИмяПредопределенныхДанных* пустую строку и записать элемент:

`&НаКлиенте``Процедура Отсоединить (Команда)``ОтсоединитьНаСервере ();``КонецПроцедуры``&НаСервереБезКонтекста``Процедура ОтсоединитьНаСервере ()``ОбъектДанных =``Справочники.Организации.ОсновнаяОрганизация.ПолучитьОбъект ();``ОбъектДанных.ИмяПредопределенныхДанных = "";``ОбъектДанных.Записать ();``КонецПроцедуры`

Пиктограмма в списке изменилась:

 Организации

Создать

Найти...


Отменить поиск

Наименование

- ООО "1000 мелочей"

- ООО "Все для дома"

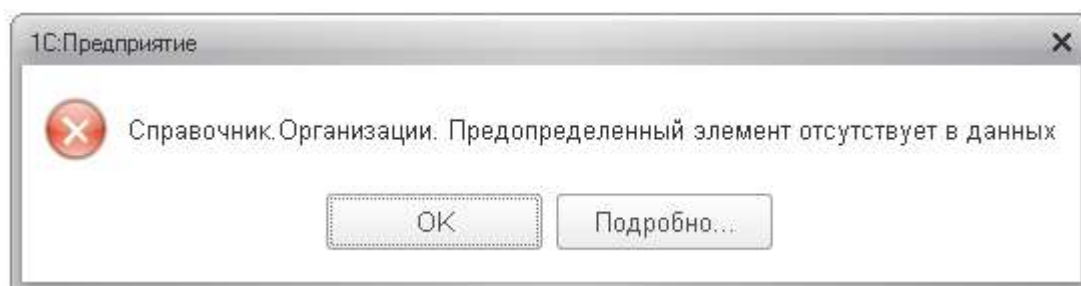
- ООО "Товары"

-  Основная организация

Теперь predetermined элемент существует только в конфигурации и в данных нет элемента, привязанного к идентификатору *ОсновнаяОрганизация*:

Ссылка	ВерсияДанных	Пиктограмма	Код	Наименование	Включен/Учет	Предопределенный	ИмяПредопределенныхДанных
ООО "Все для дома"	AAAAA0AAAAA	Нет	000000001	ООО "Все для дома"	Да	Нет	
ООО "Товары"	AAAAA0AAAAA	Нет	000000002	ООО "Товары"	Да	Нет	
ООО "1000 мякотей"	AAAAA0AAAAA	Нет	000000003	ООО "1000 мякотей"	Нет	Нет	
Основная организация	AAAAA0AAAAA	Нет	000000001	Основная организац.	Нет	Нет	

Обращение из программного кода к predetermined элементу вызовет исключение:



Чтобы связать predetermined элемент с новой записью, нужно присвоить свойству *ИмяПредопределенныхДанных* имя predetermined элемента:

&НаКлиенте

**Процедура** Привязать (Команда)

ПривязатьНаСервере ();

**КонецПроцедуры**

&НаСервереБезКонтекста

**Процедура** ПривязатьНаСервере ()

НайденнаяСсылка = Справочники.Организации.НайтиПоНаименованию ("ООО  
"Товары"");

ОбъектДанных = НайденнаяСсылка.ПолучитьОбъект ();

ОбъектДанных.ИмяПредопределенныхДанных = "ОсновнаяОрганизация";

ОбъектДанных.Записать ();

**КонецПроцедуры**

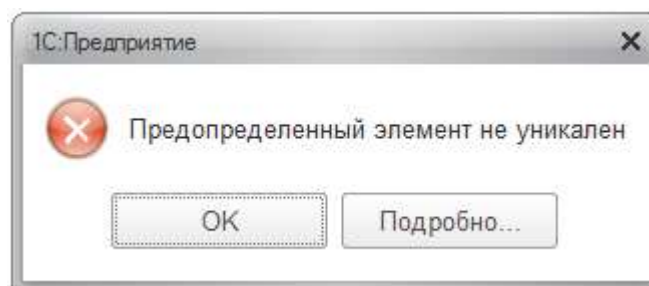
Теперь с пиктограммой predetermined элемента выводится другой элемент справочника:

 Организации

Создать Найти... Отменить поиск

Наименование
ООО "1000 мелочей"
ООО "Все для дома"
ООО "Товары"
Основная организация

При записи объекта проверяется, не существует ли в информационной базе другого элемента данных с таким же идентификатором predeterminedанных данных. В случае его обнаружения будет выдано сообщение об ошибке:



При помощи запроса обратимся к полю *ИмяПредопределенныхДанных*.

```
Запрос = Новый Запрос;
```

```
Запрос.УстановитьПараметр ("ИмяПредопределенныхДанных",  
ИмяПредопределенныхДанных);
```

```
Запрос.Текст =
```

```
"ВЫБРАТЬ
```

```
| Организации.Ссылка,
```

```
| Организации.Наименование,
```

```
| Организации.Код
```

```
| ИЗ
```

```
| Справочник.Организации КАК Организации
```

```
| ГДЕ
```

```
| Организации.ИмяПредопределенныхДанных = &ИмяПредопределенныхДанных";
```

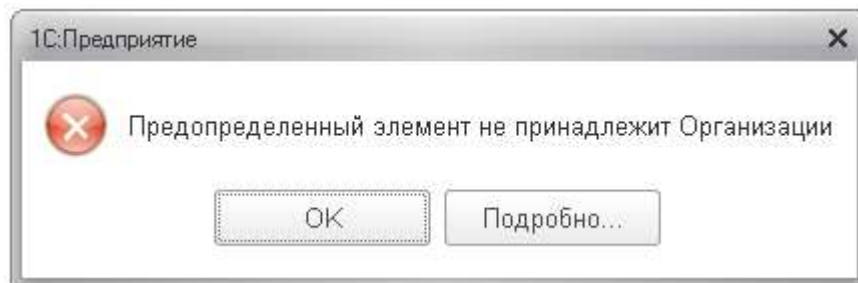
```
Выборка = Запрос.Выполнить().Выбрать();
```

```
Если Выборка.Следующий() Тогда
```

```
Сообщение = Новый СообщениеПользователю;
```

```
Сообщение.Текст = Выборка.Наименование;
```

```
Сообщение.Сообщить ();
КонецЕсли;
```



При использовании в конфигурации *Библиотеки стандартных подсистем (БСП)* версии 2.1.4 и выше рекомендуется использовать функцию *ПредопределенныйЭлемент* общего модуля *ОбщегоНазначенияКлиентСервер*, которая возвращает *Неопределено* для несуществующих в информационной базе предопределенных элементов:

```
Организация =
ОбщегоНазначенияКлиентСервер.ПредопределенныйЭлемент ("Справочник.Организации.
ОсновнаяОрганизация");
```

Платформа позволяет получить список только предопределенных элементов, присутствующих в данных. Это можно сделать, например, при помощи запроса.

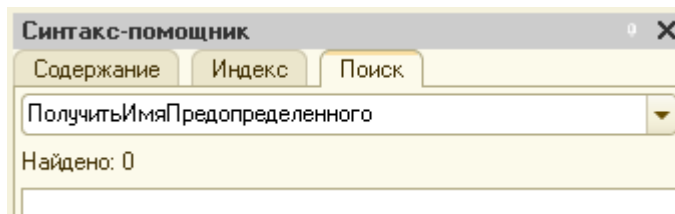
```
ОбъектДанных = Метаданные.Справочники.Организации;
ТекстЗапроса = "
| ВЫБРАТЬ
|     ИмяОбъекта.Ссылка
| ИЗ
|     " + ОбъектДанных.ПолноеИмя () + " КАК ИмяОбъекта
|
| ГДЕ
|     ИмяОбъекта.Предопределенный;
```

```
Запрос = Новый Запрос (ТекстЗапроса);
Выборка = Запрос.Выполнить ().Выбрать ();
```

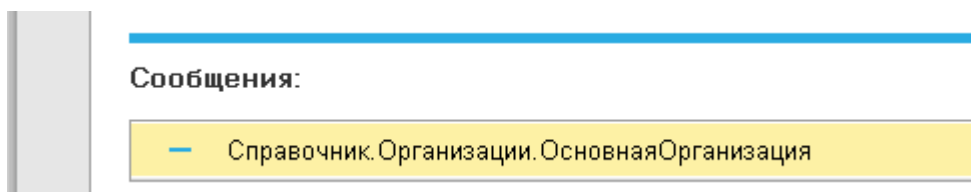
```
Пока Выборка.Следующий () Цикл
    ИмяПредопределенного =
Справочники [ОбъектДанных.Имя].ПолучитьИмяПредопределенного (Выборка.Ссылка);
    Сообщить (ИмяПредопределенного);
КонецЦикла;
```

Получение списка предопределенных элементов, заданных в метаданных, при помощи встроенного языка – невозможно.

Однако начиная с платформы 8.3.3, метод *ПолучитьИмяПредопределенного()* является устаревшим и поддерживается для совместимости, поэтому даже в *Синтакс-помощнике* платформы 8.3.4 он не отображается.



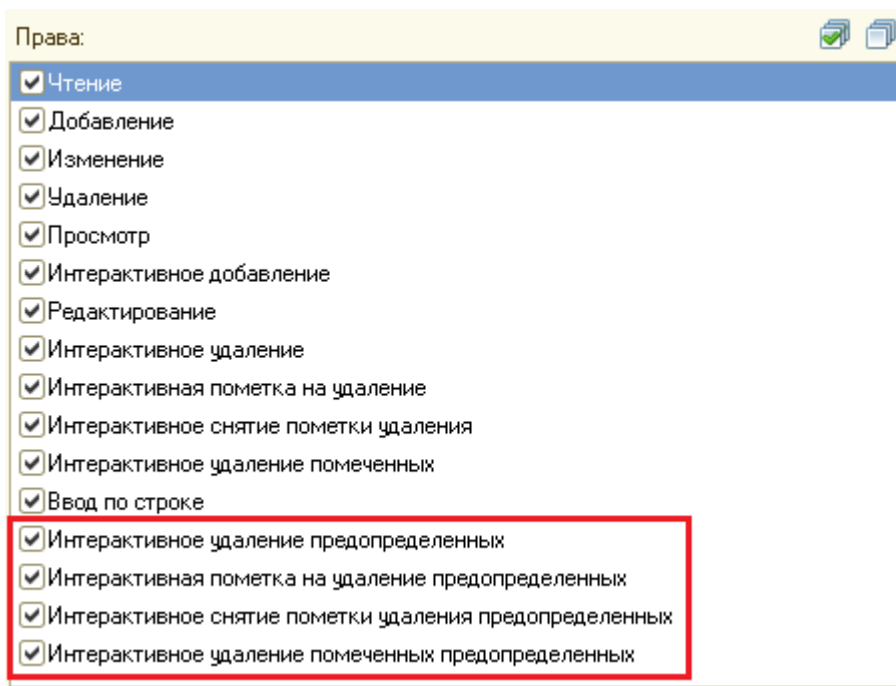
В платформе 8.3 существует метод глобального контекста *ПолучитьПолноеИмяПредопределенногоЗначения()*. Этот метод выполняет действие, обратное методу *ПредопределенноеЗначение()*, – по ссылке на предопределенный элемент возвращает полное имя предопределенного значения.



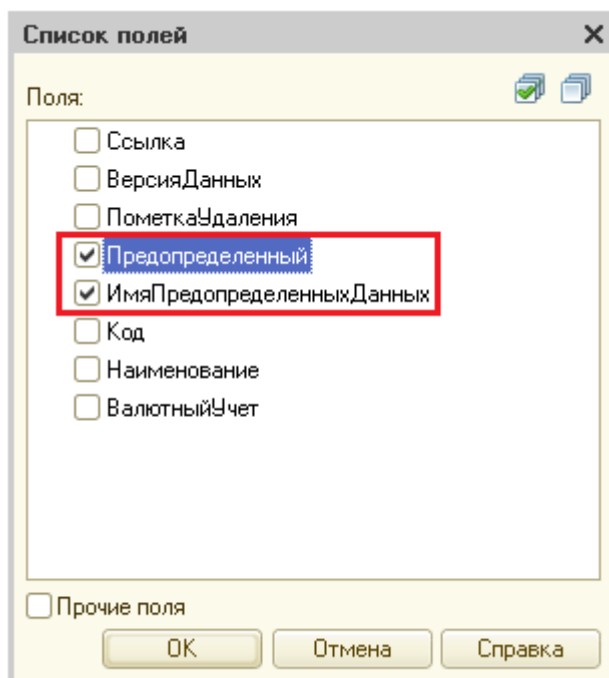
В платформе 8.3 разрешено пометать предопределенные данные на удаление и производить удаление предопределенных данных. Если же удаление предопределенных элементов в разрабатываемой конфигурации недопустимо, то следует настроить права доступа.

Теперь в платформе присутствуют новые права доступа, определяющие возможность редактирования предопределенных данных:

- ИнтерактивноеУдалениеПредопределенныхДанных,
- ИнтерактивнаяПометкаУдаленияПредопределенныхДанных,
- ИнтерактивноеСнятиеПометкиУдаленияПредопределенныхДанных,
- ИнтерактивноеУдалениеПомеченныхПредопределенныхДанных.

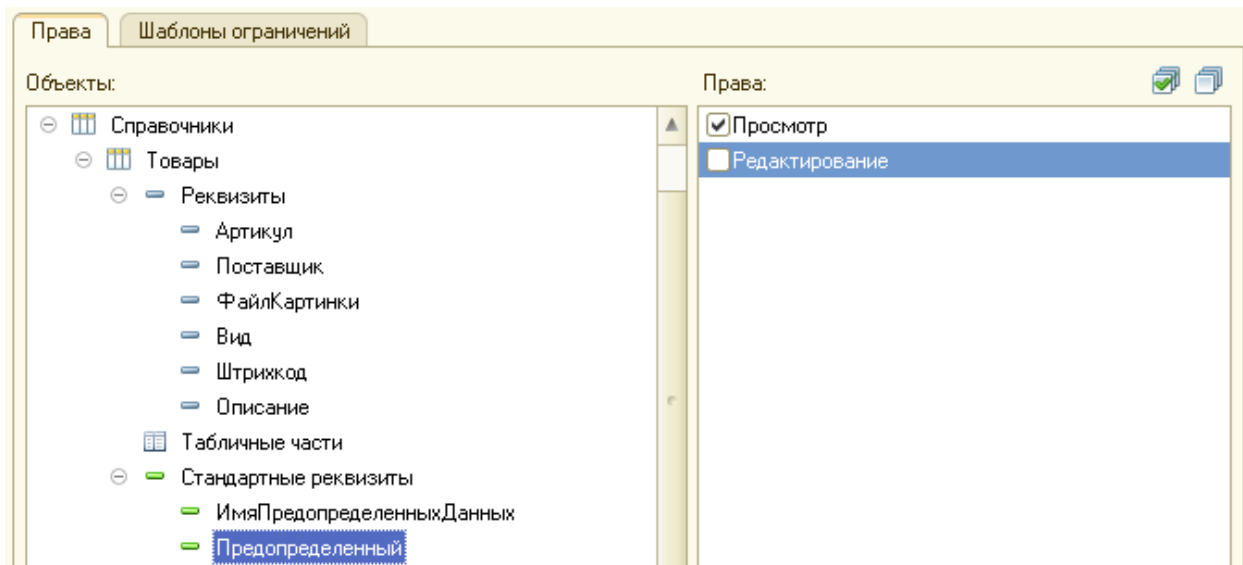


Ограничения доступа к данным одновременно назначаются на поля *Предопределенный* и *ИмяПредопределенныхДанных* (оба флажка устанавливаются и снимаются одновременно):





Однако “интерактивные” права на просмотр и редактирование стандартных реквизитов *Предопределенный* и *ИмяПредопределенныхДанных* можно установить по отдельности:



Для корректного функционирования системы необходимо, чтобы предопределенные элементы не только были созданы в конфигураторе, но и в данных информационной базы существовали связанные с ними элементы.

Для указания стратегии создания или изменения предопределенных данных было реализовано системное перечисление *ОбновлениеПредопределенныхДанных*, содержащее следующие значения:

- *Авто* – необходимость создания предопределенных элементов и обновление существующих будет определяться системой автоматически;
- *НеОбновлятьАвтоматически* – не будет выполняться автоматическое создание предопределенных элементов и обновление существующих значений. В случае отсутствия элемента данных, связанного с предопределенным элементом, будет сгенерировано исключение при попытке обращения к нему;
- *ОбновлятьАвтоматически* – будет выполняться автоматическое создание предопределенных элементов и обновление существующих значений.

Чтобы принять решение, нужно ли создавать и обновлять предопределенные данные, система последовательно анализирует следующие значения:

- значение, установленное для объекта в данных;
- значение, установленное для объекта метаданных в конфигураторе;
- значение, установленное в целом для информационной базы;
- вид информационной базы - центральная (либо не распределенная база) или периферийная база.

Обратите внимание на первые два пункта. При некотором внешнем сходстве формулировок между этими пунктами есть принципиальная разница: второе значение устанавливается разработчиком для объекта метаданных в конфигураторе, а первое хранится в данных информационной базы. Подробнее будет рассмотрено ниже.

Такое большое количество анализируемых значений было реализовано для возможности полноценно управлять созданием и обновлением предопределенных данных.

Например, в периферийной базе предопределенные данные не должны создаваться автоматически, а в случае отключения базы от РИБа нельзя допустить автоматического создания элементов.

Кроме того, разработчик в зависимости от логики прикладного решения должен иметь возможность управлять процессом создания предопределенных элементов.

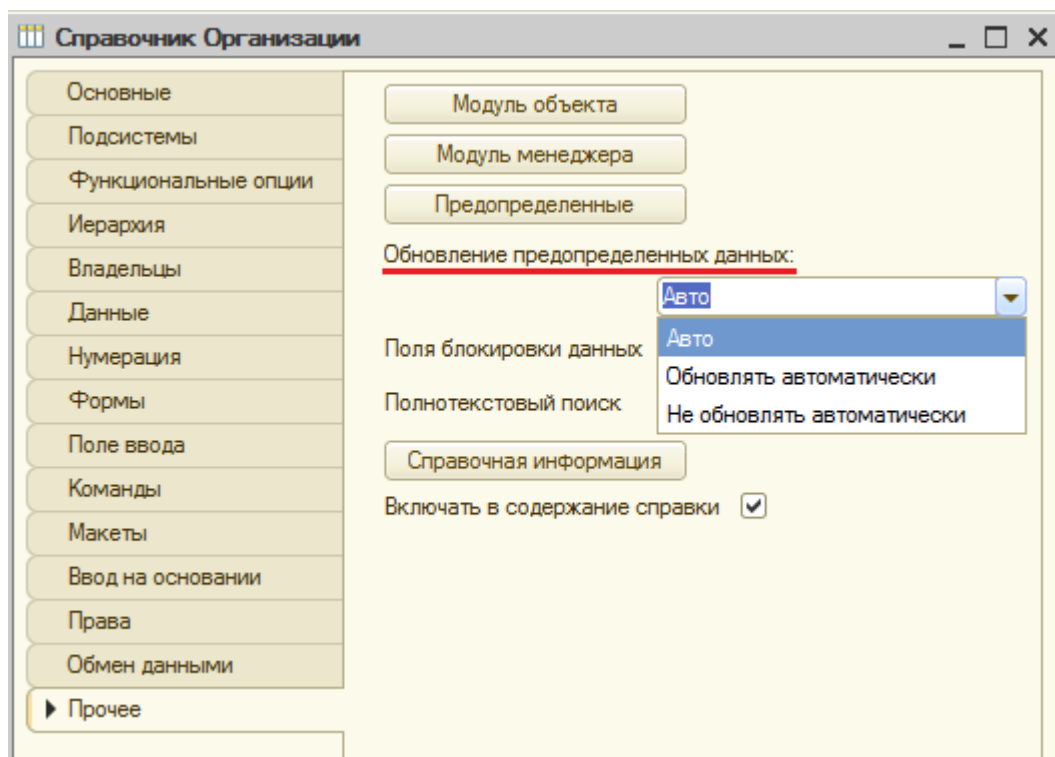
Теперь рассмотрим последовательность действий, которые выполняет система, чтобы определить, нужно ли создавать или обновлять предопределенные данные.

1. Если для объекта метаданных в информационной базе (т.е. в данных) установлен режим обновления, отличный от Авто, то используется это значение.

Получить и установить значение этого свойства для каждого объекта метаданных в пользовательском режиме можно с помощью методов *ПолучитьОбновлениеПредопределенныхДанных()* и *УстановитьОбновлениеПредопределенныхДанных()*. Эти методы доступны у объектов типа *СправочникМенеджер*, *ПланСчетовМенеджер*, *ПланВидовХарактеристикМенеджер*, *ПланВидовРасчетаМенеджер*.

Следует учитывать, что метод *УстановитьОбновлениеПредопределенныхДанных()* невозможно использовать во время фоновой реструктуризации базы.

- Иначе если для объекта метаданных в конфигурации установлен режим обновления, отличный от *Авто*, то используется это значение.



- Иначе если для информационной базы установлен режим обновления, отличный от *Авто*, то используется это значение.

Получить и установить значение этого свойства для каждого объекта метаданных в пользовательском режиме можно с помощью методов *ПолучитьОбновлениеПредопределенныхДанныхИнформационнойБазы()* и *УстановитьОбновлениеПредопределенныхДанныхИнформационнойБазы()*.

- Иначе анализируется вид узла распределенной базы. Если это периферийный узел, то predeterminedные данные не будут обновлены (соответствует значению *НеОбновлятьАвтоматически*). Если проверка выполняется для центрального узла распределенной базы или для базы, не являющейся распределенной, обновление predeterminedных данных будет выполнено (соответствует значению *ОбновлятьАвтоматически*).

Если база находится в режиме совместимости "*Версия 8.3.3*", то алгоритм определения необходимости обновления predeterminedных данных выглядит немного по-другому.

Обновление не будет производиться:

- если в метаданных или в данных установлено *НеОбновлятьАвтоматически*;
- если в метаданных или в данных установлено *Авто* и текущий узел является периферийным.

В противном случае predetermined данные будут обновлены.

Создание в информационной базе данных, связанных с predetermined, может быть выполнено следующими способами:

1. При реструктуризации информационной базы в случае, когда разрешено автоматическое создание и обновление predetermined данных, а также predetermined данные ранее создавались в этой области данных или информационной базе.
2. При первом обращении к таблице с данными объекта конфигурации. Создание predetermined элементов будет выполняться при условии, что их создание не запрещено.
3. С помощью кода на встроенном языке, в котором заполняется значение свойства *ИмяПредопределенныхДанных*. Этот способ можно использовать, если автоматическое создание predetermined элементов запрещено в свойствах объектов конфигурации.

Обновление данных, связанных с predetermined, будет выполняться, если:

- разрешено автоматическое обновление predetermined данных;
- predetermined данные связаны с данными информационной базы;
- predetermined данные изменены в конфигураторе.

Во всех остальных случаях информация о predetermined элементах из конфигуратора не будет перенесена в информационную базу.

Если в конфигураторе удалить predetermined элемент, то будут выполнены следующие действия:

- элемент данных, связанный с predetermined элементом, будет помечен на удаление;
- в списке субконто элемента плана счетов будут удалены соответствующие записи;

- в списках ведущих, базовых и вытесняющих видов расчета будут удалены соответствующие записи.

Установим для справочника *Организации* в конфигураторе свойство *ОбновлениеПредопределенныхДанных* в *НеОбновлятьАвтоматически* и удалим предопределенный элемент *ОсновнаяОрганизация*.

В пользовательском режиме у элемента справочника, связанного с предопределенным, значение свойства *ИмяПредопределенныхДанных* стало равным *#708c7031-c7bd-45d4-a8c9-65b2fecfde62*, где символы после знака “#” – это уникальный идентификатор удаленного элемента метаданных.

Строка	ВоскрДанные	ПометДанные	Код	Наименование	ВалютныйЧет	Предопределенный	ИмяПредопределенныхДанных
Основная организация	AAAAQAAAAA	Нет	00000001	Основная организация	Нет	Да	#708c7031-c7bd-45d4-a8c9-65b2fecfde62
ООО "Все для дома"	AAAAA9AAAAA	Нет	00000001	ООО "Все для дома"	Да	Нет	
ООО "Товары"	AAAAWAAAAA	Нет	00000002	ООО "Товары"	Да	Нет	
ООО "1000 мелочей"	AAAA8AAAAA	Нет	00000003	ООО "1000 мелочей"	Нет	Нет	

При использовании распределенной информационной базы предопределенные элементы из конфигурации передаются вместе с сообщением, содержащим изменения конфигурации.

А элементы данных, связанные с предопределенными, регистрируются в планах обмена так же, как и обычные данные. Автоматического создания предопределенных данных в периферийной базе не происходит.

В распределенной базе иногда могут возникать ошибки, которые исправляются после отключения от *РИБа*, загрузки в периферийную базу конфигурации центральной базы и дальнейшего подключения к распределенной базе.

При работе на платформе 8.3 следует учитывать следующую особенность. База, у которой главный узел стал не установлен (базу отключили от *РИБа*), не является периферийной, поэтому при запуске этой базы в пользовательском режиме может произойти обновление предопределенных данных согласно алгоритму, описанному выше.

Чтобы не допустить такого поведения, можно использовать метод глобального контекста *УстановитьОбновлениеПредопределенныхДанныхИнформационнойБазы()* или использовать параметр командной строки в пакетном режиме запуска конфигуратора */SetPredefinedDataUpdate*, который предназначен для указания режимов обновления предопределенных данных.

Этот параметр может принимать одно из трех значений:

- *UpdateAutomatically* – при реструктуризации информационной базы будет выполняться автоматическое создание predetermined элементов и обновление существующих значений;
- *DoNotUpdateAutomatically* – при реструктуризации информационной базы не будет выполняться автоматическое создание новых predetermined элементов и обновление их значений;
- *Auto* – фактическое значение вычисляется автоматически. Для главного узла информационной базы значение будет равно *UpdateAutomatically*, для периферийного узла информационной базы будет равно *DoNotUpdateAutomatically*.

Командная строка запуска конфигуратора с этим параметром будет выглядеть следующим образом:

```
"C:\Program Files\1cv8\8.3.4.437\bin\1cv8.exe DESIGNER /F"ПутьКБазе" /N"Администратор" /P"Пароль" /SetPredefinedDataUpdate -DoNotUpdateAutomatically"
```

Таким образом возможно задавать признак обновления predetermined данных, когда информационная база отключается от РИБ, например, для восстановления работоспособности.

Так же можно поступить в случае, если с помощью обмена без использования РИБ взаимодействуют две конфигурации, где predetermined данные задаются в одной базе, а в другую они должны попасть после обмена.

Для информационной базы, где создаются predetermined элементы, следует использовать значение *UpdateAutomatically*, а для второй информационной базы – *DoNotUpdateAutomatically*.

В платформе 8.3 для объектов конфигурации, которые могут содержать predetermined данные (справочники, планы счетов, планы видов характеристик, планы видов расчета), реализована возможность устанавливать свойство *ИмяПредопределенныхДанных* в значение специального вида, состоящее из символа “#” и текстового представления уникального идентификатора.

Таким образом можно создавать обработки загрузки predetermined данных из файла.

В платформе 8.2.14 появился механизм разделения данных. При использовании разделенной базы predetermined элементы создавали ряд ограничений. Так, было

невозможно использовать предопределенные данные для объектов конфигурации, входящих в состав разделителя в режиме “*Независимо и совместно*”.

Теперь в платформе 8.3 предопределенные данные можно использовать для объектов, входящих в состав любых разделителей. Ссылки на предопределенные элементы более не являются одинаковыми, в том числе и в различных областях (даже для объектов, входящих в состав разделителя в режиме *Независимо*).

Например, рассмотрим предопределенный элемент *ОсновнаяОрганизация* в справочнике *Организации*.

Пусть в базе есть две области: с кодами *01* и *02*. Значит, уникальный идентификатор элемента данных, связанного с элементом *ОсновнаяОрганизация* в области с кодом *01*, не будет равен уникальному идентификатору элемента данных, связанного с элементом *ОсновнаяОрганизация* в области *02*.

Подведем итоги. Рассмотрим, для чего было внесено столько нововведений в работу с предопределенными данными. Прежний вариант использования предопределенных элементов накладывал ограничения при работе в режиме разделенной информационной базы.

Так, невозможно было использовать предопределенные данные для объектов конфигурации, входящих в состав разделителя в режиме “*Независимо и совместно*”. Невозможно было связать существующий объект данных с предопределенными данными. Один предопределенный элемент имел одинаковую ссылку во всех областях данных.

При обмене данными с использованием *РИБа* предопределенные данные передавались только с изменениями конфигурации прикладного решения. Управлять созданием и обновлением предопределенных данных было невозможно.

Таким образом, рассмотренные нововведения направлены на развитие возможностей платформы по работе в режиме разделенной информационной базы, что особенно актуально в рамках модели *SaaS* (“программы как сервис”), которую также называют работой “в облаке”.

**Ханевич Василий**

г. Калининград

## Дополнительные материалы

Все статьи проекта Курсы-по-1С.рф: <http://курсы-по-1с.рф/blog/articles/>

## Курсы по программированию в 1С v.8

Базовый и Продвинутой курсы по Программированию на Платформе 1С 8  
<http://www.Spec8.ru/>



### Базовый курс по программированию в 1С v.8

Курс про **готовые приемы и решения**  
**90% задач** по программированию в 1С



### Продвинутой курс по программированию в 1С v.8

Больше, чем Вы можете себе представить  
Детальнее требований на **1С:Специалист**

«Курс по подготовке к Аттестации по Платформе 1С 8.2 / 8.3»

<http://курсы-по-1с.рф/dev-attestation/>



### Подготовка к Аттестации по Платформе 1С v.8

Аттестация по Платформе – **с первого раза**  
**Экономия 100 - 150 часов** подготовки