

Общая информация

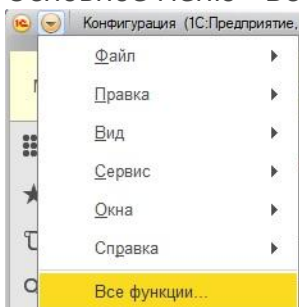
Получение информации о конфигурации

Через отчет по метаданным конфигурации (в режиме Конфигуратор меню Конфигурация – Отчет по конфигурации). Имена, показанные в отчете, соответствуют именам свойств и коллекций объектов.

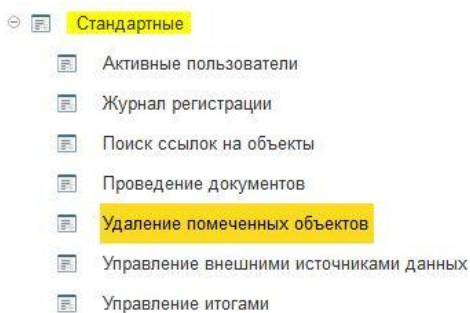
Ссылочное удаление (пометка удаления)

В конфигурации могут существовать связанные свойства у разных объектов. Предлагаемый алгоритм удаления:

- Объект помечается как удаляемый путем установки пометки на удаление (кнопка Del или контекстное меню). Все связанные объекты автоматически помечаются на удаление.
- Основное меню - Все функции



- Стандартные - Удаление помеченных объектов



- Можно посмотреть удаляемые объекты выбрав Выборочное удаление.

Объекты, помеченные на удаление (всего объектов: 2):



- Можно выбрать удаляемые объекты и удалить.

Удаление элемента родительского справочника.

Есть элемент (Подчиненный) в подчиненном справочнике, у которого выставлен в качестве родителя удаляемый нами элемент (Основной).

- В случае удаление через "Удалить" (Shift-Del) оба элемента тихо удалятся.
- В случае установки пометки можно увидеть все удаляемые элементы.
- При снятии пометки удаления с Основного, с Подчиненного тоже удаляется метка.
- Однако проверка ссылочной целостности происходит только в момент установки пометки. То есть если пометить на удаление Основной, затем создать второй Подчиненный, будет выдано предупреждение "Вы хотите установить в качестве владельца элемент с пометкой удаления", но пометка удаления на втором подчиненном элементе не установится. При попытке удаления через предлагаемую процедуру появится ошибка и объект не будет удален.

Результаты удаления:

Удалено объектов: 0
Невозможно удалить объектов: 1, т.к. в информационной базе на них ссылаются другие объекты.
Для просмотра списка таких объектов нажмите кнопку Далее >>

Но при удалении через "Удалить" (Shift-Del) все элементы (и помеченный, и непомеченный) удалятся.

- Если при просмотре удаляемых объектов убрать пометку с подчиненного, оставив основной, то удаление не произойдет и будет ошибка из предыдущего пункта.

Удаление элемента справочника в случае наличия ссылки на него в табличной части.

- При установке пометки на удаление, в табличной части другого справочника элемент не помечается на удаление. В списке элементов будет отражен один элемент, однако при попытке удаления будет выдано предупреждение о наличии ссылок и элемент не будет удален и можно будет посмотреть список элементов, в которых есть ссылки.
- При удалении через "Удалить" (Shift-Del) элемент будет удален, но в табличной части другого справочника в строке со ссылкой будет надпись "Объект не найден".

Код: 000000001

Наименование: Рулетка

Добавить ↑ ↓ Еще ▾

N	Единица измерения	Значение
1	<Объект не найден> (28:80с...	5
2	Секунда	4

- Для исправления предыдущей ситуации можно использовать Администрирование - Тестирование и исправление информационной базы

Тестирование и исправление информационной базы

Проверки и режимы :

- ☐ Реиндексация таблиц информационной базы
- ☒ Проверка логической целостности информационной базы
- ☒ Проверка ссылочной целостности информационной базы
- ☐ Пересчет итогов
- ☐ Сжатие таблиц информационной базы
- ☐ Реструктуризация таблиц информационной базы

☒ Только тестирование
☐ Тестирование и исправление

При наличии ссылок на несуществующие объекты :
☐ Создавать объекты
☐ Очищать ссылки

При частичной потере данных объектов :
☐ Создавать объекты
☐ Удалять объект

Выполнить
Закрыть
Справка

В служебных сообщениях будет выведен список элементов с битыми ссылками.

Служебные сообщения

- Тестирование начато
- Проверка логической целостности. Справочник.ХарактеристикаТовара.ТабличнаяЧасть.Характеристики.Реквизит.ЕдиницаИзмерения Рулетка.НомерСтр. Объект, на который ссылается значение, отсутствует.
- Тестирование закончено

Это считается медленной процедурой и требует монопольного доступа, можно использовать [внешнюю обработку](#)

Типы данных встроенных классов

Раздел относится к классам Справочник, Документ, ПланВидовХарактеристик, ПланСчетов и ПланВидовРасчета. Идея общая для всех классов, рассмотрен пример класса Справочник. При создании объекта появляются следующие типы (так их называют в документации) - СправочникМенеджер, СправочникСсылка, СправочникОбъект, СправочникВыборка. Эти слова используются часто. Но есть тма вопросов, основной из которых - нахрена было сделано данное разделение терминологий. Ответ: понять и простить.

Названия типов не связаны с их использованием в коде.

СправочникМенеджер

Название типа	Отображение в коде	Назначение
СправочникиМенеджер	Справочники	Содержит все СправочникМенеджер и метод ТипВсеСсылки(), позволяющий получить ссылки на типы всех справочников для последующего сравнения с типом переменной.
СправочникМенеджер	Справочники.ИмяСправочника	Это конкретный элемент СправочникиМенеджера. Понять и простить. <div><pre>текноменкл = Справочники.Номенклатура; //текноменкл это справочникменеджер.</pre></div> <p>Общие действия, относящиеся к конкретному справочнику, а не к его конкретным объектам. Например, методы позволяют создать новый объект или найти объект по коду.</p>

СправочникСсылка

Хранит ссылку, идентифицирующую запись в таблице справочника. Внутренний идентификатор, хранящийся в реквизите Ссылка таблицы справочника. Через

СправочникСсылка можно получить данные всего объекта через методы. При программной работе объект СправочникСсылка получают при поиске элемента справочника. Используется только для чтения данных. При получении ссылки на одну и ту же запись справочника разными способами, ссылки будут одинаковые (в отличие от объекта). Получение СправочникОбъект из ссылки:

```
ЗаписьНоменклатураСсылка = Справочники.Номенклатура.НайтиПоНаименованию("Ручка");  
ЗаписьНоменклатураОбъект = ЗаписьНоменклатураСсылка.ПолучитьОбъект();  
// Здесь идут какие-то изменения  
ЗаписьНоменклатураОбъект.Записать();
```

При первом обращении к свойству происходит запрос к БД и данные кэшируются. При обращении или раз в 20 секунд происходит проверка неизменности данных. При внешнем изменении данных, свойство, к которому происходит обращение, обновляется. Объекты в кэше очищаются через 20 минут.

СправочникОбъект.

Используется для создания, изменения и удаления объекта, для отображения и редактирования всех данных элемента справочника в форме элемента. Так как содержит весь объект, тяжелее ссылки. Очень похоже на передачу данных по ссылке и по значению. При каждом запросе на получение объекта создается новый временный объект.

Объект может быть создан с помощью менеджера справочника. В этом случае создается новый объект, которого еще нет в базе данных. Если его записать, то появится новый объект в базе данных. Объект может быть получен из ссылки. В этом случае из базы данных считывается существующий объект, на который указывает ссылка. Объект также может быть получен из выборки (СправочникВыборка). В этом случае данные объекта заполняются данными, полученными в процессе считывания выборки.

СправочникОбъект оптимизирует запись изменений в базу данных, сохраняя только изменения. Обеспечивается оптимистическая блокировка - запись заблокируется в случае изменения в базе данных после считывания. Это обеспечивает логическую целостность изменения объектов.

Существует пессимистическая блокировка, запрещающая изменения до снятия блокировки. Включается методом Заблокировать(). В основном он предназначен для блокировки объектов, редактируемых в форме. Расширение формы элемента справочника автоматически включает блокировку, чтобы пользователь был уверен что, начав редактировать объект, он сможет его записать.

Предоставляется доступ через свойства к данным объекта. Значения СправочникОбъект заполняются при считывании существующего объекта значениями из базы данных. Проверка факта изменения в СправочникСсылка и отсутствия изменений в одном из объектов:

&НаСервереБезКонтекста

Процедура ПроверитьАктуальностьНаСервере()

```
□ЕдиницаСсылка = Справочники.ЕдиницыИзмерения.НайтиПоНаименованию("Метр");
```

```
□ЕдиницаОбъект1 = ЕдиницаСсылка.ПолучитьОбъект();
```

```
□ЕдиницаОбъект2 = ЕдиницаСсылка.ПолучитьОбъект();
```

```
□ЕдиницаОбъект2.Наименование = "Метр2";
```

```
□ЕдиницаОбъект2.Записать();
```

```
□Сообщить("В ссылке:" + ЕдиницаСсылка.Наименование);
```

```
□Сообщить("В объекте 1:" + ЕдиницаОбъект1.Наименование);
```

```
□Сообщить("В объекте 2:" + ЕдиницаОбъект2.Наименование);
```

```
  //ЕдиницаОбъект1.Наименование = "Метр3";
```

```
□//ЕдиницаОбъект1.Записать();
```

КонецПроцедуры

&НаКлиенте

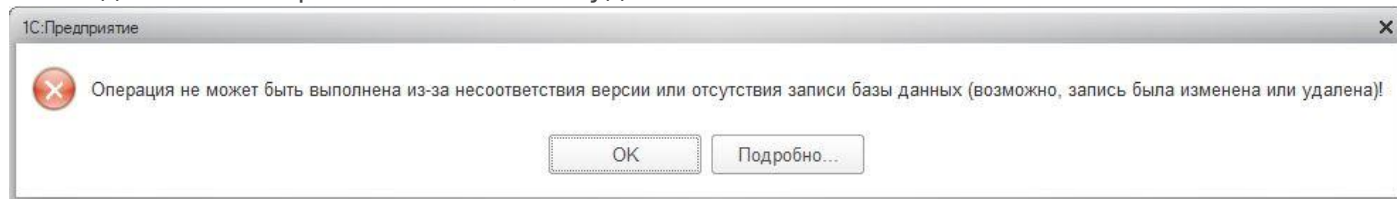
Процедура ПроверитьАктуальность(Команда)

```
□ПроверитьАктуальностьНаСервере();
```

КонецПроцедуры

—	В ссылке:Метр2
—	В объекте 1:Метр
—	В объекте 2:Метр2

Наименование в ЕдиницаОбъект1 действительно не изменилось. Если раскомментировать команды записи первого объекта, то будет ошибка:



Похоже, что обновление СправочникОбъект возможно только через повторное получение, поэтому в продуктовом решении минимум необходимо добавить обработку ошибок и, возможно, запись в цикле до успешного сохранения + понимание что же изменилось, почему, и нужно ли в принципе обновлять объект.

Получение ссылки из объекта:

```
СсылкаНаОбъект = МойОбъект.Ссылка;
```

Поскольку реквизит ссылка - это параметр в базе данных, то до записи нового объекта в ней ничего нет. Проверка:

&НаКлиенте

Процедура ДобавитьЕдиницуИзмерения(Команда)

□СоздатьЕдиницуИзмерения(Объект.НоваяЕдиницаИзмерения);

КонецПроцедуры

&НаСервереБезКонтекста

Процедура СоздатьЕдиницуИзмерения(НазваниеЕдиницы)

□НовыйОбъект = Справочники.ЕдиницыИзмерения.СоздатьЭлемент();

□НовыйОбъект.Наименование = НазваниеЕдиницы;

□Сообщить("До записи:" + НовыйОбъект.Ссылка);

□НовыйОбъект.Записать();

□Сообщить("После записи:" + НовыйОбъект.Ссылка);

КонецПроцедуры

Сообщения:

— До записи:

— После записи:Метр

СправочникВыборка

Предназначен для обхода всех элементов справочника. Считанные элементы могут быть получены в качестве значений типа СправочникОбъект, при этом они не будут повторно считываться из базы данных. Выборка считывает данные объектов целиком (все поля и все табличные части).

&НаСервереБезКонтекста

Процедура ВывестиВсеЕдиницыНаСервере()

□Выборка = Справочники.ЕдиницыИзмерения.Выбрать();

□Пока Выборка.Следующий() Цикл

□□Сообщить(Выборка.Наименование);

□КонецЦикла;

КонецПроцедуры

&НаКлиенте

Процедура ВывестиВсеЕдиницы(Команда)

□ВывестиВсеЕдиницыНаСервере();

КонецПроцедуры

СправочникСписок

[Доп. информация](#)

Предназначен для динамического просмотра данных справочника в элементе управления ТабличноеПоле (свойство Данные). Возможность считывать данные порциями. В том случае, если ни один из индексов таблицы не соответствует заданному пользователем упорядочиванию, механизм динамического просмотра не будет задействован. Может быть настроен состав считываемых полей (колонок), отбор и сортировка. Список осуществляет считывание данных порциями в процессе навигации пользователем в табличном поле. Обращение к реквизитам на чтение аналогично СправочникСсылка.

Регламентные задания

Разобраться, [интересная статья](#)

Программное создание конфигурации.

Хер. Внешний обработчик может только уведомить об отсутствии чего-либо в конфигурации, создать кодом не получится.

Тестирование

Через жопу. [Статья о тестировании](#) Используется уже другой (!) язык. Им видимо двух не хватало...

Git

Еще нужно разобраться, но вроде есть.

Revision #15

Created 8 January 2025 17:20:57 by Admin

Updated 14 February 2025 07:16:24 by Admin