

# Ядро и формы обработки 1С

Последние изменения: 2024-03-26

## Ядро

Ядро обработки представляет собой программный интерфейс для взаимодействия между товарно-учетными системами на базе 1С и программами «Магазин 15»/ «Склад 15» на платформе Mobile SMARTS.

Ядро полностью отвязано (абстрагировано, независимо) от конкретной конфигурации 1С. Для интеграции с конкретной конфигурацией предназначена отдельно поставляемая интеграционная обработка, содержащая в себе запросы для [выгрузки справочников](#) и обработчики [загрузки документов](#) из Mobile SMARTS в 1С.

Предполагается, что вносить изменения в ядро могут ТОЛЬКО разработчики «Клеверенс». Для интеграции ядра с конфигурацией 1С нужно вносить изменения в [интеграционную обработку](#). Поэтому всегда используется одна и та же версия ядра, а разница лишь в интеграционных обработках и настройках Mobile SMARTS. Аналогично у платформы и конфигураций «1С: Предприятия» — платформа 1С предоставляет интерфейс для разработки конфигураций и закрыта от изменений разработчиками.

Весь функционал ядра разбит на глобальное ядро/ контекст (исполняется на сервере 1С) и локальное ядро/ контекст (исполняется на тонком клиенте):

- **Глобальное ядро (контекст).** Это модуль обработки, реквизиты и табличные части обработки. Здесь хранится основная часть ядра – механизмы работы с [базой Mobile SMARTS](#), механизмы работы с настройками, универсальные и вспомогательные механизмы работы с данными. При работе в онлайн-режиме здесь же (в реквизитах) хранятся текущие настройки системы.
- **Локальное ядро (контекст).** Специально созданная форма обработки, в которой хранится мини-копия ядра – экспортные методы и функции, отвечающие ТОЛЬКО за работу с компонентой (обращение к методам [COM-объекта Mobile SMARTS](#)).

Для каждого из контекстов существует кэш — структуры данных, которые при запуске ядро получает из баз 1С и Mobile SMARTS, необходимые для дальнейшей работы ядра.

Локальное ядро необходимо только в случаях, когда сервер Mobile SMARTS (COM-компонента) установлен на локальной машине (там же, откуда запускается 1С). В иных случаях (файловая база 1С, сервер Mobile SMARTS установлен на сервере 1С) — используется глобальное ядро.

Техническая реализация данного механизма проста — в реквизитах объекта и в КАЖДОЙ форме создаются обязательные реквизиты «ЯдроОбщее» и «ЯдроКомпоненты» произвольного типа. Данные реквизиты содержат «транзитные» ссылки на используемый контекст.

- Реквизит «ЯдроОбщее» всегда содержит в себе значение «ОбработкаОбъект» для доступа к глобальному контексту.
- Реквизит «ЯдроКомпоненты» содержит в себе либо значение «ОбработкаОбъект» (в случае, если компонента установлена на сервере 1С), либо ссылку на форму с мини-копией ядра (в случае использования клиент-серверного варианта работы и компоненты, установленной на локальной машине).

Для исключения «перекрестного» обращения к контекстам в связи с идентичными именами реквизитов, содержащих в себе ссылки на контексты, жестко регламентируется обращение к ним:

- Из модуля объекта обращение к ядру производится только через предикат «ЭтотОбъект.»: «ЭтотОбъект.ЯдроОбщее» или «ЭтотОбъект.ЯдроКомпоненты».

- Из всех остальных модулей форм обращение производится только через предикат «ЭтаФорма.»: «ЭтаФорма.ЯдроОбщее» или «ЭтаФорма.ЯдроКомпоненты».

Процедуры и функции мини-ядра клиентского контекста полностью идентичны аналогичным процедурам и функциям серверного контекста (можно их просто копипастить). Разница будет только в их вызове, например «ЭтотОбъект.ЯдроОбщее.ПолучитьНастройку(ИмяНастройки)» или «ЭтаФорма.ЯдроКомпоненты.СохранитьДокументНаСервереSMARTS()».

Все процессы передачи данных между глобальным контекстом и остальными субъектами (формами и даже локальным контекстом, вне зависимости от месторасположения последнего) производятся строго через механизм временного хранилища 1С.

**Порядок инициализации контекстов:**

1. При запуске системы из модуля объекта инициализируется глобальный контекст «ЭтотОбъект.ЯдроОбщее = ЭтотОбъект»

2. Считывается настройка «Место установки сервера SMARTS».

Если место установки компоненты – на сервере, тогда инициализируется локальный контекст «ЭтотОбъект.ЯдроКомпоненты = ЭтотОбъект».

Если место установки компоненты – на клиенте, тогда локальный контекст не инициализируется, и будет инициализирован только при открытии главной формы обработки.

3. Если локальный контекст инициализирован – в кэш глобального контекста считываются все настройки из базы Mobile SMARTS («ЯдроОбщее.Настройки = ЯдроКомпоненты.ИнициализироватьНастройки»).

4. В случае работы в онлайн-режиме инициализация на этом заканчивается, ибо онлайн-режим может работать только в случае, когда компонента установлена на сервере (либо использование файлового варианта). В противном случае, при батч-режиме или при режиме автообмена – инициализация продолжается.

5. При открытии главной формы обработки мы повторно инициализируем глобальный контекст, но уже применительно к реквизитам формы – «ЭтаФорма.ЯдроОбщее = ЭтотОбъект».

6. Считывается настройка «Место установки сервера Mobile SMARTS».

Если место установки компоненты – на сервере, тогда инициализируется локальный контекст «ЭтаФорма.ЯдроКомпоненты = ЭтотОбъект».

Если место установки компоненты – на клиенте, тогда локальный контекст инициализируется как «ЭтаФорма.ЯдроКомпоненты = ПолучитьФорму(«ФормаЛокальногоКонтекста»)».

Реквизиты форм «ЯдроОбщее» и «ЯдроКомпоненты» являются наследуемыми, и должны всегда передаваться в подчиненную форму при ее открытии (например, при открытии формы обмена документами из главной формы обработки).

Локальная обработка не имеет доступа к глобальной, соответственно, все необходимые данные настроек (из глобальной обработки) для работы методов локального контекста должны передаваться непосредственно при их вызове.

## Формы

Формы не являются непосредственной частью ядра. Они являются пользовательским интерфейсом для использования 3 главных функций ядра: работа с настройками, обмен справочниками, обмен документами.

Формы делятся на:

- **ключевые формы** — главная форма, форма обмена документами, форма списка отборов справочников, локальный контекст
- **формы списков** (для различных таблиц, т.е. сериализуемых настроек) — узлы, бп, отборы документов, глобальные параметры, произвольные коды, произвольные отборы документов. Содержимое этих форм хранится в настройках базы MS, далее, при старте обработке, помещается в кэш и хранится там до завершения работы обработки. Эти формы доступны только для просмотра (за исключением формы списка отборов справочников) + опционально можно экспортировать/импортировать содержимое список во внешние файлы.
- **формы настроек** (формы карточек) — узла, бп, отбора справочников, отбора документов, правила выгрузки полей, правила загрузки полей, правила выгрузки ТЧ, правила загрузки ТЧ, связи выгрузки ТЧ, произвольный код, произвольный отбор документов). При открытии в данные формы передаётся содержимое текущей строки того списка, из которого она была открыта. Форма позволяет редактировать данные и сохранять их.
- **формы просмотра (ReadOnly)** — документа MS и шапки/строки документа MS
- **служебные формы** — выбор строки из таблицы, загрузки новых ШК, настройки подключения, выбор документа для загрузки в 1С, выбор пользователя или группы, редактирования текста запроса

Формы обработки НЕ ДОЛЖНЫ содержать никаких методов работы с объектными данными 1С или Mobile SMARTS. Вся работа с данными 1С происходит в глобальной обработке, а с данными Mobile SMARTS – в локальной обработке. Все промежуточные методы подготовки, обработки и прочего с полученными данными – находятся только в глобальной обработке.

Не нашли что искали?



Задать вопрос в техническую поддержку

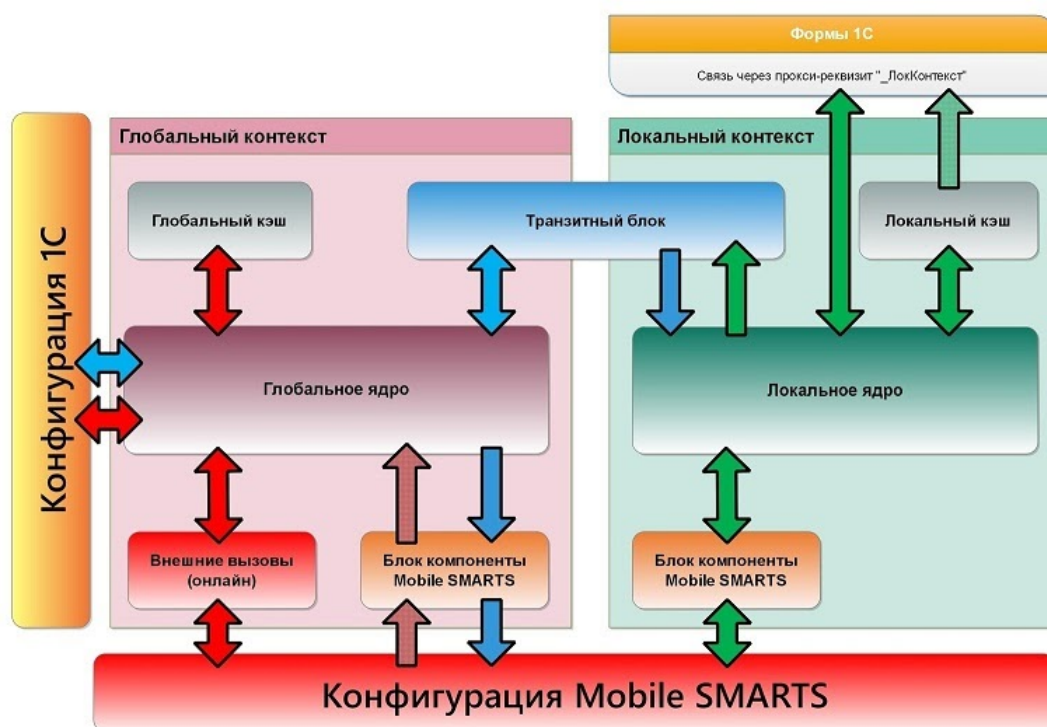
# Функционал ядра обработки 1С

Последние изменения: 2024-03-26

Функционал ядра разбит на 3 крупных блока:

1. Работа с настройками обмена (**настройки бизнес-процессов**, **торговых объектов**, **глобальных параметров**, отборов документов и справочников в разрезе торговых объектов и типов обмена и т.д.).
2. **Выгрузка справочников** в режимах онлайн, офлайн и по регламенту согласно настройкам обмена.
3. **Выгрузка/загрузка документов** из 1С в Mobile SMARTS согласно настройкам обмена.

Все остальные блоки являются вспомогательными для работы с этими 3 блоками.



1. **Транзитный блок** — расположен на клиенте, нужен для передачи управления на сервер (при необходимости), параметры и результат функции передаются через временное хранилище
2. **Блок сохранения/восстановления настроек подключения и параметров текущего узла в хранилище общих настроек (далее - ХОН)** — доступно только на сервере, с клиента доступны через транзитный блок.
3. **Программный интерфейс для работы напрямую с MS (API)** — полностью дублируются на клиенте и на сервере, т.к. **база Mobile SMARTS** может быть зарегистрирована на клиенте либо на сервере 1С. Фактически, это обмен данными с Mobile SMARTS непосредственно через **COM-объекты** и вызовы их методов: терминал коннектор, база, настройки, сторадж коннектор, среда.
  - сохранение/восстановление настроек, удаление настроек, получение настроек по умолчанию;
  - выгрузка справочников (номенклатуры и доп.таблиц);
  - обмен документами - выгрузка, загрузка, удаление, получение списков документов;
  - служебные - получение списка баз, базы по ид, настроек базы, получение текста ошибки, метаданных документов, пользователей, шаблонов **весового товара**.
4. **Клиентские и серверные обёртки для работы с настройками MS:**

- вызывают соответствующие методы API MS (инициализация кэшей, чтение, установка настроек, запись/чтение из кэша, получение настроек по умолчанию);
- вспомогательные механизмы;
- **ТОЛЬКО КЛИЕНТСКИЕ МЕТОДЫ:** сохранения/восстановления из файла таблиц из форм список, либо отдельных элементов таблиц из форм настроек.

#### 5. Клиентские и серверные методы выгрузки справочников и номенклатуры:

- методы для получения данных из 1С есть **ТОЛЬКО НА СЕРВЕРЕ**, т.к. только оттуда можно получить данные из базы, поэтому они вызываются с клиента через транзитный блок;
- также методы для формирования текстов запроса, работы с СКД, постобработка таблиц данных выполняются **ТОЛЬКО НА СЕРВЕРЕ**;
- методы для взаимодействия с API MS одинаковые на клиенте и сервере.

#### 6. Работа с документами 1С — аналогично п.5, получение документов из 1С, запись в 1С, получение списка документов и т.д. выполняется **ТОЛЬКО НА СЕРВЕРЕ**, с клиента вызываются через транзитный блок

#### 7. Внешние вызовы — выполняются только на сервере или внешнем соединении, нужны для работы ТСД в онлайн-режиме. В свою очередь, решают следующие задачи:

- получение актуальных данных справочников из 1С согласно настройкам отборов,
- получение списка документов 1С согласно настройкам отборов,
- выгрузка документа 1С в Mobile SMARTS для его набора на терминале,
- загрузка (завершение) документа из Mobile SMARTS в 1С.

Не нашли что искали?



Задать вопрос в техническую поддержку

# Хранение объектов и переменных на клиенте, на сервере и на форме

Последние изменения: 2024-03-26

## Переменные ядра

Переменные ядра хранят данные, к которым требуется часто обращаться во многих методах ядра (фактически, это кэш). Необходимы для быстрого доступа к часто используемым данным, чтобы не получать их из [баз Mobile SMARTS](#)/ 1С при каждом обращении к ним.

Все данные, которые помещаются в переменные ядра при инициализации обработки (интерактивно или онлайн) откуда-то загружаются — из метаданных базы Mobile SMARTS, настроек базы Mobile SMARTS, из хранилища общих настроек 1С (ХОН), из метаданных конфигурации 1С или создаются программно и заполняются постоянными значениями-константами.

Для локального ядра это:

- клиентские переменные формы «ЛокальныйКонтекст», которые доступны в любых, но только клиентских методах формы
- реквизиты формы «ЛокальныйКонтекст», которые доступны в любых и клиентских, и серверных методах формы.

Имя переменной		
Для чего нужна		
Откуда загружается		
Может ли изменяться		
Клиентские переменные формы «Локальный контекст»		
РасположениеКомпонент		
Тип — Строка, значения — «НаКлиенте» или «НаСервере». Означает, что с клиента 1С или с сервера 1С будет осуществляться взаимодействие с базой Mobile SMARTS. Для обмена через COM обязательно наличие зарегистрированной внешней компоненты на соответствующей стороне (клиент/сервер). Для REST API ничего дополнительно делать не нужно.		
ХОН		
Да, в форме настройки подключения		

## ДанныеТекущегоУзла

Тип — Структура. Содержит в себе

ID узла, наименование (например, «Магазин на Ленина») и ID базы Mobile SMARTS, которая привязана к этому узлу. При этом, к узлу может быть привязана либо **частная база**, либо основная база. Необходимо для случая, когда к текущей базе

1С привязан определенный узел (торговый объект – магазин, склад) и необходимо осуществлять обмен документами и справочниками

не с основной базой Mobile SMARTS, а с частной. В этом случае, в основной базе Mobile SMARTS хранятся только настройки, а справочники и документы – в частной базе.

## Настройки базы Mobile SMARTS

Да, на главной  
форме при  
выборе узла (торгового  
объекта) Категории  
НастроекMS

## КатегорииНастроекMS

Тип – Структура. Содержит метаданные о настройках, сгруппированные по определенным критериям:

- **сериализуемые настройки** – это настройки, которые являются таблицами. Нужны только для того чтобы знать, нужно ли сериализовывать/десериализовать при экспорте/импорте настроек в файл/из файла
- **узловые настройки** – это **настройки, которые можно сохранять в разрезе узлов (магазинов, складов)**, например, настройки обмена справочниками – т.к. для каждого узла может быть необходимость выгружать только «свою» номенклатуру, склады, остатки и т.д.
- **настройки подключения** – нужны, в первую очередь, для **подключения к базе Mobile SMARTS** – ID базы, имя базы, строка подключения, имя ИО, ID текущего узла, данные о конфигурации 1С
- **настройки справочников** – хранит флажки, означающие, какие справочники нужно выгружать – номенклатура, склады, остатки и т.д.
- **настройки номенклатуры** – хранит настройки выгрузки справочника «номенклатура» – использовать шаблоны весового товара, префикс кода весового товара, длина кода весового товара, переписывать номенклатуру при выгрузке и т.д.

## Программно

Нет

## НастройкиИнициализированы

Тип – Булево. Означает, что настройки успешно загружены из базы Mobile SMARTS. Если флаг = Ложь, значит, необходимо перезагрузить настройки из базы

## Программно

Да, если потеряна связь  
с сервером MS



<b>ПараметрыПодключенияMS</b>
Тип – Структура. Хранит настройки подключения к базе Mobile SMARTS — ID базы, Имя базы, ID приложения, Строка подключения, QR, Логин, Токен, имя ИО, путь ИО, тип ИО, тип установленного продукта, уровень приложения
ХОН
Да, в форме настройки подключения
<b>СтруктураМетаданных</b>
Тип – Структура. Хранит метаданные документов 1С, метаданные документов MS, пользователи MS, устройства MS, таблицы MS, UniПолейНоменклатуры
Метаданные базы 1С, метаданные базы MS, Настройки базы MS
Нет
<b>СОМЧастнойБазыMS</b>
Тип – Структура. Хранит <b>СОМ-объекты</b> для работы с частной базой – TerminalConnector и StorageConnector. Заполняется, только если подключение к базе Mobile SMARTS осуществляется через СОМ. Для <b>REST API</b> не заполняется.
Программно
Да, если изменим базу, к которой будем подключаться для обмена
<b>СОМВременнойБазыMS</b>
Аналогично СОМЧастнойБазыMS, но используется только в форме подключения для проверки соединения, чтобы не затирать соединение с существующими базами
Программно
Да, т.к. служебная переменная
<b>СОМБазыMS</b>
Аналогично СОМЧастнойБазыMS, но используется для взаимодействия с основной базой MS
Программно
Да, если изменим базу, к которой будем подключаться для обмена
<b>ТекущиеНастройкиMS</b>
Тип – Структура. Содержит все настройки текущей базы MS в виде пар Ключ-Значение, например, ВыгружатьНоменклатуруСразу = Истина и т.д.
Настройки базы MS
Да, из всех форм, где могут редактироваться настройки
<b>ПутьКОбработке</b>
НЕ ИСПОЛЬЗУЕТСЯ!!! Путь хранится в переменной ядра ПараметрыПодключенияMS. ПутьИнтеграционнойОбработки
—
—



<b>КОМсоздан</b>
Тип – Булево. Означает, что успешно созданы СОМ-объекты для работы с базами MS – TerminalConnector и StorageConnector и помещены в структуры СОМЧастнойБазыMS/ СОМБазыMS
Программно
Да, если потеряна связь с сервером MS
<b>ПраваПользователя</b>
Тип – Строка. Хранит роль пользователя в 1С – «Администратор», «Пользователь», влияет на отображение некоторых кнопок на главной форме – кнопки настройки, кнопки выбора ИО
<b>Метаданные базы 1С</b>
Нет
<b>ФормаИнтеграционнойОбработки</b>
Тип – ФормаКлиентскогоПриложения. Используется для вызова клиентского обработчика «После открытия формы документа 1С»
Программно
Да, в форме настройки ИО
<b>Реквизиты формы «Локальный контекст»</b>
<b>_ИнтеграционнаяОбработка</b> <b>Адрес</b>
Тип – Строка. Хранит адрес <a href="#">интеграционной обработки</a> во временном хранилище. Впоследствии, интеграционная обработка извлекается из временного хранилища на стороне сервера и из нее вызываются экспортные методы – обработчики загрузки документа и т.д.
Программно
Да, в форме настройки ИО

**АдресаПеременных**

Тип – Структура. Хранит в себе адреса во временном хранилище других переменных ядра. При изменении значений в самих переменных ядра параллельно изменяются эти же значения и во временном хранилище. Эту структуру можно передавать в глобальное ядро и оттуда иметь доступ практически ко всем переменным локального ядра. Состав структуры, жирным и подчеркнутым выделены переменные, хранящие адреса аналогичных переменных ядра:

- **ДанныеТекущегоУзла**
- **ИдентификаторФормы** – служебный УИД для хранения данных во временном хранилище, пока открыта обработка
- **ИнтеграционнаяОбработка** - см. **ИнтеграционнаяОбработка**
- **Адрес**
- **ИнформацияСовместимости** – версия ИО, поддерживаемые конфигурации 1С, поддерживаемые версии конфигураций 1С, используется для сообщения на главной форме о успешном/неуспешном подключении ИО и её совместимости с текущей базой 1С
- **ИспользоватьREST\_API** – флаг, означает возможность использования REST API
- **НастройкиИнициализированы**
- **ПараметрыПодключенияMS**
- **РасположениеКомпоненты**
- **СтруктураМетаданных**
- **СтруктураПодключения**
- **БазыMS** – аналог **ПараметрыПодключенияMS**
- **ТекущиеНастройкиMS**
- **ТипИнтерфейса**

Программно,  
т.к. в нее просто сохраняются  
другие, уже инициализи-  
рованные, переменные ядра

—

**ЕстьЧастнаяБаза**

Тип – Булево. Означает, что к текущей базе 1С привязан какой-либо узел и этот узел будет использовать частную базу для обмена документами и справочниками

Программно

Да, на главной форме при выборе узла (торгового объекта)

**ИмяИнтеграционнойОбра-  
ботки**

Тип – Строка, СправочникСсылка  
Для внешней ИО – путь к файлу внешней обработки.  
Для встроенной в конфигурацию 1С ИО – имя метаданных встроенной обработки.  
Для ИО из справочника «Дополнительные отчеты и обработки» – ссылку на элемент справочника

ХОН

Да, в форме настройки ИО

<b>ИнтеграционнаяОбработка</b> <b>Подключена</b>
Тип – Булево. Означает, что интеграционная обработка успешно подключена. Флаг взводится при инициализации ядра
Программно
Да, если потеряна связь с сервером MS
<b>ПолноеИмяОбъекта</b>
Тип – Строка. Полное имя обработки-ядра, например, «ВнешняяОбработка.КлеверенсТСД_ОсновнаяОбработка»
Программно
Нет
<b>ТипОбработки</b>
Тип – Строка. Определяет расположение ИО - «ФайлНаДиске», «Встроенная» или «СправочникСсылка»
Настройки базы MS
Да, в форме настройки ИО

Для глобального ядра это:

- реквизиты обработки, доступны из любого метода модуля объекта;
  - ЭтоWebСервис – флаг, означающий работу ядра из [веб-сервиса](#);
  - СОМЧастнойБазыMS, СОМВременнойБазыMS, СОМБазыMS – аналоги одноименных переменных локального ядра.
- серверные переменные модуля объекта, доступны из любого метода модуля объекта;
  - КэшАдресаПеременных – копия переменной АдресаПеременных, но используется только для онлайн-режима, когда между вызовами ядра нужно сохранять значения переменных ядра.
- переменная локального ядра АдресаПеременных, передающаяся в методы глобального ядра в качестве параметра, посредством которой есть доступ к большинству переменных локального ядра.

## Что хранится в хранилище общих настроек 1С (ХОН)

### Принцип сохранения/ загрузки данных в ХОН

Если у пользователя 1С есть право доступа «Администрирование данных», то сохраняется и индивидуальная настройка по имени пользователя, и общая (глобальная) настройка для всей базы 1С, с именем пользователя, равным пустой строке «».

Иначе сохраняется только индивидуальная настройка по имени пользователя.

Аналогичный принцип при загрузке данных из ХОН, но можно при загрузке указывать, загружать только

пользовательскую настройку или дополнительно еще и общую настройку

Ключ
Значение
<b>Cleverence_ПараметрыПодключенияMS</b>
Структура, содержащая Ид базы, имя базы, Ид приложения, строку подключения, путь и имя ИО, версию приложения и т.д.
<b>Cleverence_РасположениеКомпоненты</b>
Расположение внешней <b>COM-компоненты</b> для работы из 1С с базой MS: на клиенте 1С или на сервере 1С (файлы Cleverence.MobileSMARTS.ComConnector.dll, Cleverence.Infrastructure.dll, Cleverence.Warehouse.TerminalConnector.dll). При этом, сама база MS может быть расположена где угодно, главное, чтобы к ней был доступ
<b>Cleverence_ДатаАктуальностиКэша</b>
Дата и время, когда в последний раз из MS (файл customsettings.xml) в кэш были считаны настройки базы. Используется только для веб-сервиса либо внешнего соединения. Если прошло более 15 минут после последнего получения настроек, то нужно перечитать настройки из MS, т.к. за это время настройки могли поменять интерактивно из 1С или из панели MS
<b>Cleverence_СтруктураМетаданныхДокументы1С</b>
Метаданные документов 1С – имена документов, их реквизиты, табличные части, реквизиты табличных частей
<b>Cleverence_СтруктураНастроек</b>
Все настройки базы MS, такие как: <b>настройки отборов документов и справочников</b> , <b>бизнес-процессы</b> , произвольные коды и т.д.
<b>Cleverence_СтруктураНастроек_ + ID_ТекущегоУзла</b>
Все настройки базы MS, но уже для конкретного <b>торгового объекта</b>
<b>Cleverence_СтруктураМетаданныхДокументыMS</b>
Метаданные документов MS – имена документов, их реквизиты, табличные части, реквизиты табличных частей
<b>Cleverence_ПользователиMS</b>
<b>Пользователи и группы пользователей</b> , которые заведены в подключенной базе MS
<b>Cleverence_ТаблицыMS</b>
<b>Таблицы MS</b> (имена таблиц, имена полей таблиц) таких как: «Склады», «Остатки», «Цены» и т.д.
<b>Cleverence_УстройстваMS</b>
Ид и имена устройств, подключенных к базе
<b>Cleverence_ИмяИнтеграционнойОбработки</b>
Имя ИО, например, «ИнтеграционнаяОбработка_УТ_11_4»
<b>Cleverence_РежимВыбораБазы</b>
По строке подключения ИЛИ из списка
<b>Cleverence_ТипИнтерфейса</b>
REST_API ИЛИ COM

**Cleverence\_REST\_API\_ДатаАктуальностиСоединения**

Дата и время последнего подключения по REST API к базе MS. Используется только для веб-сервиса либо внешнего соединения

**Cleverence\_ДатаАктуальностиНастроек\_ + ПостфиксБазы**

Дата, когда были сохранены/изменены настройки пользователем.

Обновляется в методах КомЯдро\_ЗаписатьНастройкиSMARTS.

Проверка даты осуществляется в онлайн-вызовах со стороны MS, если настройки изменились, то происходит переинициализация настроек

Не нашли что искали?
















Задать вопрос в техническую поддержку

# Описание хранения настроек в базе Mobile SMARTS и их конвертация

Последние изменения: 2024-03-26

В каждом шаблоне **базы Mobile SMARTS** (например «Клеверенс Магазин 15 Полный.mstmpl») в папке «\Обработки 1С\Настройки по умолчанию» располагаются файлы с настройками по умолчанию (они же — типовые настройки) для разных конфигураций 1С:

Клеверенс Магазин 15 Полный > Обработки 1С > Настройки по-умолчанию			
Имя	Дата изменения	Тип	Размер
 astorVashMagazin	15.04.2020 15:49	Файл "XML"	834 КБ
 DalionTrend	15.04.2020 15:49	Файл "XML"	724 КБ
 DalionUmPro	15.04.2020 15:49	Файл "XML"	853 КБ
 roznica22	15.04.2020 15:49	Файл "XML"	1 347 КБ
 roznica23	15.04.2020 15:49	Файл "XML"	1 370 КБ
 shtrihm7	15.04.2020 15:49	Файл "XML"	614 КБ
 shtrihm52	15.04.2020 15:49	Файл "XML"	774 КБ
 unf16	15.04.2020 15:49	Файл "XML"	1 183 КБ
 unfukr16	15.04.2020 15:49	Файл "XML"	1 161 КБ
 ut103	15.04.2020 15:49	Файл "XML"	582 КБ
 ut110	15.04.2020 15:49	Файл "XML"	1 633 КБ
 ut112	15.04.2020 15:49	Файл "XML"	1 704 КБ
 ut114	15.04.2020 15:49	Файл "XML"	1 726 КБ

Настройка "Склад 15, Полный"

Настройка подключения 1С

Выберите конфигурацию и базу 1С, с которой будет происходить работа в Mobile SMARTS!

Конфигурация 1С: 1С: Управление торговлей 11.4 (поддержка он-лайн)

Выбранная конфигурация, с которой будет происходить дальнейшая работа. От выбранной конфигурации будет зависеть по каким настройкам будет работать выбранная база и какую интеграционную обработку использовать.

База данных 1С: Имя базы: ut114demo  
Версия 1С: 8.3.16.1063  
Папка/Адрес: Srvr="VM-Server2019-PSG-1C";Ref="ut114demo";

Изменить...

Выберите базу 1С, с которой будет происходить работа. База должна соответствовать выбранной конфигурации.

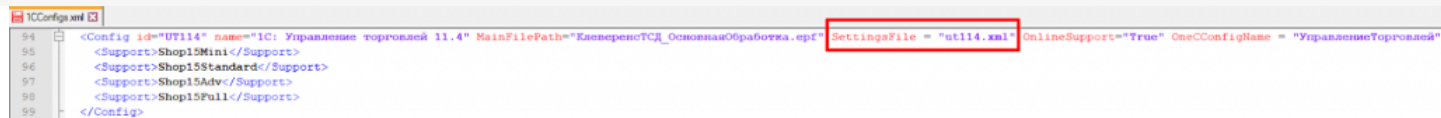
Помощь

Назад

Далее

Прервать

В мастере создания новой базы из шаблона мы выбираем, с какой **конфигурацией 1С** будет обмениваться данными новая база Mobile SMARTS. Соответствие конфигураций 1С и файлов настроек хранится в файле шаблона «1CConfigs.xml»:



После распаковки данных из шаблона в папку базы Mobile SMARTS копируется папка «Обработки 1С» с файлами настроек для всех конфигураций 1С, копируется файл «1CConfigs.xml», а также копируется файл, соответствующий выбранной конфигурации 1С, и сохраняется с именем «default\_customsettings.xml». Например, мы выбрали конфигурацию «Управление торговлей 11.4», поэтому в файле «default\_customsettings.xml» хранятся данные из файла «ut114.xml» (на скриншоте можно увидеть, что размеры файлов совпадают с точностью до байта).

Имя	Тип	Размер	Дата	Атрибуты
astorVashMagazin	xml	853 670	15.04.2020 15:49	—
DaltonTrend	xml	740 403	15.04.2020 15:49	—
DaltonUmPro	xml	873 109	15.04.2020 15:49	—
roznica22	xml	1 379 191	15.04.2020 15:49	—
roznica23	xml	1 402 577	15.04.2020 15:49	—
shtrihm52	xml	792 145	15.04.2020 15:49	—
shtrihm7	xml	628 145	15.04.2020 15:49	—
unf16	xml	1 210 831	15.04.2020 15:49	—
unfukr16	xml	1 188 187	15.04.2020 15:49	—
ut103	xml	595 005	15.04.2020 15:49	—
ut110	xml	1 671 722	15.04.2020 15:49	—
ut112	xml	1 744 076	15.04.2020 15:49	—
ut114	xml	1 766 599	15.04.2020 15:49	—

Имя	Тип	Размер	Дата	Атрибу
Backup	Папка		16.04.2020 16:52	—
Client Installer	Папка		15.04.2020 19:15	—
Connections	Папка		30.03.2020 09:00	—
Control Panel	Папка		11.04.2020 12:51	—
Documents	Папка		30.03.2020 09:00	—
DOSTUP	Папка		11.04.2020 15:59	—
Label Templates	Папка		30.03.2020 09:00	—
Licenses	Папка		30.03.2020 09:01	—
Logs	Папка		15.04.2020 19:15	—
Server	Папка		30.03.2020 10:58	—
XlsCnv	Папка		30.03.2020 09:00	—
Обработки 1С	Папка		30.03.2020 09:00	—
Cleverence.InstallTemplate.Common	dll	750 224	30.03.2020 09:00	—
appres	resources	254 825	30.03.2020 09:00	—
Cleverence.Warehouse.AuthorizationTokens	smc	934	11.04.2020 15:50	—
1CConfigs	xml	24 736	30.03.2020 09:00	—
AppDescription	xml	1 369	30.03.2020 09:00	—
customsettings	xml	5 819	11.04.2020 15:59	—
default_customsettings	xml	1 766 599	30.03.2020 09:00	—
OnRemoteInstall	xml	259	30.03.2020 09:00	—
OnUpdate	xml	973	30.03.2020 09:00	—
settings	xml	682	11.04.2020 15:59	—

В файле «default\_customsettings.xml» хранятся все настройки базы, которые будут использованы по умолчанию. Так как это типовые настройки, то их нельзя изменять программно в коде 1С.

Имя	Тип	Размер	Дата	Атрибу
Backup	Папка		16.04.2020 16:52	—
Client Installer	Папка		15.04.2020 19:15	—
Connections	Папка		30.03.2020 09:00	—
Control Panel	Папка		11.04.2020 12:51	—
Documents	Папка		30.03.2020 09:00	—
DOSTUP	Папка		11.04.2020 15:59	—
Label Templates	Папка		30.03.2020 09:00	—
Licenses	Папка		30.03.2020 09:01	—
Logs	Папка		15.04.2020 19:15	—
Server	Папка		30.03.2020 10:58	—
XlsCnv	Папка		30.03.2020 09:00	—
Обработки 1С	Папка		30.03.2020 09:00	—
Cleverence.InstallTemplate.Common	dll	750 224	30.03.2020 09:00	—
appres	resources	254 825	30.03.2020 09:00	—
Cleverence.Warehouse.AuthorizationTokens	smc	934	11.04.2020 15:50	—
1CConfigs	xml	24 736	30.03.2020 09:00	—
AppDescription	xml	1 369	30.03.2020 09:00	—
customsettings	xml	5 819	11.04.2020 15:59	—
default_customsettings	xml	1 766 599	30.03.2020 09:00	—
OnRemoteInstall	xml	259	30.03.2020 09:00	—
OnUpdate	xml	973	30.03.2020 09:00	—
settings	xml	682	11.04.2020 15:59	—

Имя	Тип	Размер	Дата	Атрибу
Backup	Папка		16.04.2020 16:52	—
Client Installer	Папка		15.04.2020 19:15	—
Connections	Папка		30.03.2020 09:00	—
Control Panel	Папка		11.04.2020 12:51	—
Documents	Папка		30.03.2020 09:00	—
DOSTUP	Папка		11.04.2020 15:59	—
Label Templates	Папка		30.03.2020 09:00	—
Licenses	Папка		30.03.2020 09:01	—
Logs	Папка		15.04.2020 19:15	—
Server	Папка		30.03.2020 10:58	—
XlsCnv	Папка		30.03.2020 09:00	—
Обработки 1С	Папка		30.03.2020 09:00	—
Cleverence.InstallTemplate.Common	dll	750 224	30.03.2020 09:00	—
appres	resources	254 825	30.03.2020 09:00	—
Cleverence.Warehouse.AuthorizationTokens	smc	934	11.04.2020 15:50	—
1CConfigs	xml	24 736	30.03.2020 09:00	—
AppDescription	xml	1 369	30.03.2020 09:00	—
customsettings	xml	5 819	11.04.2020 15:59	—
default_customsettings	xml	1 766 599	30.03.2020 09:00	—
OnRemoteInstall	xml	259	30.03.2020 09:00	—
OnUpdate	xml	973	30.03.2020 09:00	—
settings	xml	682	11.04.2020 15:59	—

После первого запуска обработки 1С в базу Mobile SMARTS записывается еще ряд настроек, которые уже считаются не типовыми, а пользовательскими, поэтому они сохраняются в отдельный файл — «customsettings.xml». Часть из них создает мастер первого запуска Mobile SMARTS, другая часть настроек создается из обработки 1С при первом запуске.



<code>&lt;SettingsValueList xmlns:clr="http://schemas.cleverence.ru/clr" capacity="32" count="26"&gt;</code>		
<code>&lt;SettingsValueNode name="НастройкиБизнесПроцессов Пользовательские"&gt;</code>		
<code>&lt;SettingsValueNode name="ВерсияОбработкиНастроек"&gt;</code>		
<code>&lt;SettingsValueNode name="ПроизвольныеКоды Пользовательские"&gt;</code>		
<code>&lt;SettingsValueNode name="ВерсияНастроек"&gt;</code>		
<code>&lt;SettingsValueNode name="ID ТекущегоУзла"&gt;</code>		
<code>&lt;SettingsValueNode name="ПодключеннаяБазаИмяКонфигурации1С"&gt;</code>		заполняются из 1С
<code>&lt;SettingsValueNode name="ПодключеннаяБазаСинонимКонфигурации1С"&gt;</code>		
<code>&lt;SettingsValueNode name="ПодключеннаяБазаСтрокаСоединения"&gt;</code>		
<code>&lt;SettingsValueNode name="ПодключеннаяБазаВерсияПлатформы1С"&gt;</code>		
<code>&lt;SettingsValueNode name="ПодключеннаяБазаВерсияКонфигурации1С"&gt;</code>		
<code>&lt;SettingsValueNode name="База1СПодключенаДата"&gt;</code>		
<code>&lt;SettingsValueNode name="Ид"&gt;</code>		
<code>&lt;SettingsValueNode name="ИдПриложения"&gt;</code>		
<code>&lt;SettingsValueNode name="УровеньПриложения"&gt;</code>		
<code>&lt;SettingsValueNode name="ИдОсновногоПриложения"&gt;</code>		
<code>&lt;SettingsValueNode name="ИмяИнтеграционнойОбработкиКонфигурации1С"&gt;</code>		
<code>&lt;SettingsValueNode name="ПутьИнтеграционнойОбработкиКонфигурации1С"&gt;</code>		
<code>&lt;SettingsValueNode name="ТипОбработки"&gt;</code>		
<code>&lt;SettingsValueNode name="OneCconfigId"&gt;</code>		
<code>&lt;SettingsValueNode name="ExchangeType"&gt;</code>		
<code>&lt;SettingsValueNode name="ПользовательОсновнойБД"&gt;</code>		заполняются мастером первого запуска
<code>&lt;SettingsValueNode name="ПарольПользователяОсновнойБД"&gt;</code>		
<code>&lt;SettingsValueNode name="OneCconfigName"&gt;</code>		
<code>&lt;SettingsValueNode name="OneCconfigVersion"&gt;</code>		
<code>&lt;SettingsValueNode name="OneCProcessingFileName"&gt;</code>		
<code>&lt;SettingsValueNode name="startWizard"&gt;</code>		

Когда мы запрашиваем у сервера Mobile SMARTS какую-то настройку по ее ключу, то сперва данная настройка ищется в пользовательских настройках, и если там не была найдена, то ищется в типовых. Если нигде не найдена, то возвращается значение «Неопределено».

В обработке 1С явно указывается доступный список настроек, которые будут считаны из базы Mobile SMARTS (они же - метаданные настроек MS, структура настроек MS). Делается это в методе `Глядро_ПолучитьПустуюСтруктуруНастроекSMARTS()`, поэтому, если в новом релизе того или иного продукта Mobile SMARTS (например, «Склад 15», «Магазин 15») обновляются метаданные настроек (добавляются новые настройки, удаляются существующие настройки, меняется тип значений существующих настроек), их также нужно явно модифицировать в этом методе для возможности корректной работы с ними в обработке:

```
// Возвращает пустую структуру настроек Mobile Smarts //...
Функция Глядро_ПолучитьПустуюСтруктуруНастроекSMARTS(СериализуемыеНастройки = Неопределено) Экспорт

    СтруктураНастроекSMARTS = Новый Структура;

    СтруктураНастроекSMARTS.Вставить("ИспользоватьШаблоныВесовогоТовара", Ложь);
    СтруктураНастроекSMARTS.Вставить("ПрефиксКодаВесовогоТовара", Ложь);
    СтруктураНастроекSMARTS.Вставить("ДлинаКодаВесовогоТовара", Ложь);
    СтруктураНастроекSMARTS.Вставить("ЧислоЗнаковПослеЗапятойВесТовара", Ложь);
    СтруктураНастроекSMARTS.Вставить("ПрефиксВключенВКодВесовогоТовара", Ложь);
    СтруктураНастроекSMARTS.Вставить("РазмерПорцииНоменклатуры", Ложь);
    СтруктураНастроекSMARTS.Вставить("ВыгружатьНоменклатуруДокументаПередВыгрузкойДокумента", Ложь);
    СтруктураНастроекSMARTS.Вставить("УстановитьПоискПоНаименованиюИАртикулу", Ложь);

    СтруктураНастроекSMARTS.Вставить("ВыгружатьНоменклатуруСразу", Ложь);
    СтруктураНастроекSMARTS.Вставить("ВыгружатьКонтрагентовСразу", Ложь);
    СтруктураНастроекSMARTS.Вставить("ВыгружатьОстаткиСразу", Ложь);
    СтруктураНастроекSMARTS.Вставить("ВыгружатьСерииСразу", Ложь);
    СтруктураНастроекSMARTS.Вставить("ВыгружатьСкладыСразу", Ложь);
    СтруктураНастроекSMARTS.Вставить("ВыгружатьФормыАСразу", Ложь);
```

Также, существует категория сериализуемых настроек, которые не являются значениями примитивного типа («Число», «Строка», «Дата», «Булево», «Ссылка»), и которые необходимо сериализовывать в строку определенного формата для возможности хранения их в текстовом файле и десериализовывать при загрузке из файла. Это могут быть таблицы значений, списки значений и другие составные типы данных. В обработке для целей сериализации/десериализации используются методы `ЗначениеВСтрокуВнутр()` и `ЗначениеИзСтрокиВнутр()`. Для сериализуемых настроек существует метод, в котором описывается состав

сериализуемых настроек — `Глядро_ПолучитьКатегорииНастроекMS()` и метод, который описывает, какие колонки в какой таблице могут существовать — `Глядро_СформироватьМетаданныеНастроекMS()`, например, для таблицы с настройками обмена справочниками могут быть только 2 поля — «ИмяТаблицы» и «НастройкаКомпоновщикаСтрокой». Для таблицы с настройками обмена документами это могут быть поля «ID\_БизнесПроцесса», «Название», «НастройкаКомпоновщикаСтрокой» и т.д. Пример:

```

Функция Глядро_СформироватьМетаданныеНастроекMS()

СтруктураМетаданныхНастроекMS = Новый Структура;

// Обмен справочниками
ШаблонНастроек = Новый Структура;
ШаблонНастроек.Вставить("ИмяТаблицы", Новый ОписаниеТипов("Строка",,,, Новый КвалификаторыСтроки(50)));
ШаблонНастроек.Вставить("НастройкаКомпоновщикаСтрокой", Новый ОписаниеТипов("Строка",,,, Новый КвалификаторыСтроки(0)));
СтруктураМетаданныхНастроекMS.Вставить("НастройкиОбменаСправочников", ШаблонНастроек);

// Обмен документами
ШаблонНастроек = Новый Структура;
ШаблонНастроек.Вставить("ID_БизнесПроцесса", Новый ОписаниеТипов("Строка",,,, Новый КвалификаторыСтроки(40)));
ШаблонНастроек.Вставить("Название", Новый ОписаниеТипов("Строка",,,, Новый КвалификаторыСтроки(100)));
ШаблонНастроек.Вставить("ТипИсходногоДокумента1С", Новый ОписаниеТипов("Строка",,,, Новый КвалификаторыСтроки(200)));
ШаблонНастроек.Вставить("ТипИсходногоДокумента1С_Синоним", Новый ОписаниеТипов("Строка",,,, Новый КвалификаторыСтроки(200)));
ШаблонНастроек.Вставить("НастройкаКомпоновщикаСтрокой", Новый ОписаниеТипов("Строка",,,, Новый КвалификаторыСтроки(0)));
ШаблонНастроек.Вставить("Используется", Новый ОписаниеТипов("Булево"));
ШаблонНастроек.Вставить("ВариантОтбора", Новый ОписаниеТипов("Строка",,,, Новый КвалификаторыСтроки(0)));
ШаблонНастроек.Вставить("ID_ПроизвольногоОтбора", Новый ОписаниеТипов("Строка",,,, Новый КвалификаторыСтроки(0)));
СтруктураМетаданныхНастроекMS.Вставить("НастройкиОтборовУзлов_Онлайн", ШаблонНастроек);
СтруктураМетаданныхНастроекMS.Вставить("НастройкиОтборовУзлов_Интерактивный", ШаблонНастроек);
СтруктураМетаданныхНастроекMS.Вставить("НастройкиОтборовУзлов_Автообмен", ШаблонНастроек);

// Бизнес-процессы
ШаблонНастроек = Новый Структура;
ШаблонНастроек.Вставить("ID_БизнесПроцесса", Новый ОписаниеТипов("Строка",,,, Новый КвалификаторыСтроки(40)));
ШаблонНастроек.Вставить("Используется", Новый ОписаниеТипов("Булево"));
ШаблонНастроек.Вставить("ЭтоТиповойБизнесПроцесс", Новый ОписаниеТипов("Булево"));
ШаблонНастроек.Вставить("Название", Новый ОписаниеТипов("Строка",,,, Новый КвалификаторыСтроки(255)));
ШаблонНастроек.Вставить("ТипИсходногоДокумента1С", Новый ОписаниеТипов("Строка",,,, Новый КвалификаторыСтроки(255)));

```

Для того чтобы в **процессе работы с обработкой** каждый раз не обращаться к базе Mobile SMARTS за получением значения настройки, все настройки при запуске обработки кэшируются в переменную ядра `АдресаПеременных.ТекущиеНастройкиMS`, а для **онлайн-режима** — дополнительно в хранилище общих настроек 1С по ключу `Cleverence_СтруктураНастроек`. Впоследствии, если какая-то настройка изменилась, она сохраняется и в базу Mobile SMARTS и обновляется в кэше, чтобы поддерживать актуальность настроек в кэше. Для онлайн-режима дополнительно существует механизм обновления настроек: если кто-то параллельно работает с базой Mobile SMARTS из 1С, то этот пользователь может изменить настройки, поэтому каждые 900 секунд происходит проверка актуальности настроек в кэше. Если они стали неактуальными, они перечитываются из базы Mobile SMARTS в кэш. Для этого в хранилище общих настроек 1С сохраняются следующие данные:

- `Cleverence_ДатаАктуальностиНастроек` — дата, когда настройки были сохранены/изменены пользователем.
- `Cleverence_ДатаАктуальностиКэша` — дата, когда данные были загружены в кэш. Если дата кэша старше даты настроек, то запускается обновление кэша.

Например, вы поменяли **глобальный параметр** у какого-то узла, от которого зависит отбор документов. При работе в режиме онлайн он никак не обновится в кэше, пока соединение не будет перезапущено. Это и есть проверка на актуальность.

Иначе говоря, `Cleverence_ДатаАктуальностиНастроек` — это дата последней записи настроек в Mobile SMARTS, а `Cleverence_ДатаАктуальностиКэша` — это дата последнего считывания настроек из Mobile SMARTS в кэш. И если `Cleverence_ДатаАктуальностиКэша <= Cleverence_ДатаАктуальностиНастроек`, т.е. если после того как считали настройки, что-то еще изменили, то нужно перечитать настройки.

Во всех формах, в которых возможен просмотр или редактирование настроек, например, **бизнес-процессов**, произвольных кодов, отборов документов и т.д. существует функционал по экспорту/импорту настроек в/из файла. Это может понадобиться для создания резервной копии настроек или для переноса настроек в другую базу Mobile SMARTS.

## Формат хранения настроек бизнес-процессов и произвольных кодов

Бизнес-процессы хранятся в 3х ключах:

- `НастройкиБизнесПроцессов_ПоУмолчанию`.
- `НастройкиБизнесПроцессов_Пользовательские`.
- `НастройкиБизнесПроцессов_ПользовательскиеНастройки`.

`НастройкиБизнесПроцессов_ПоУмолчанию` — это наши типовые настройки, которые недоступны для редактирования. Они разворачиваются вместе с базой Mobile SMARTS и обновляются в файле `default_customsettings.xml`. Пользователь может только поменять некоторые параметры, например, «Использование», «Контроль количества», «Режим записи», «Выбор пользователя» и т.д.

`НастройкиБизнесПроцессов_ПользовательскиеНастройки` — сюда сохраняются те самые параметры, которые разрешено менять пользователю в типовых настройках. Они сохраняются в файл `customsettings.xml`. Если по этому ключу отсутствует настройка, значит, пользователь ничего не менял в типовых бизнес-процессах.

`НастройкиБизнесПроцессов_Пользовательские` — сохраняются бизнес-процессы, которые создал пользователь. Они сохраняются в файл «`customsettings.xml`».

При инициализации настройки бизнес-процессы собираются из этих трех ключей в единую таблицу (метод `Глядро_СобратьОбщуюСтруктуруБизнесПроцессов()`) + проходит обновление настроек, если версия настроек ниже используемой в ядре.

Если пользователю надо что-то поменять в типовом бизнес-процессе, он может скопировать его, поменять что нужно и отключить типовой. Его скопированный бизнес-процесс будет иметь другой ID.

Главный вопрос - как мы обновляем наши типовые БП?

1. Разворачиваем базу Mobile SMARTS с типовыми настройками.
2. Открываем обработку, заходим в [настройки бизнес-процессов](#).
3. В чистой базе будут только типовые, недоступные для редактирования бизнес-процессы. Сохраняем их в файл.
4. Заходим в папку с базой Mobile SMARTS.
5. Удаляем, переименовываем или меняем расширение у файла `default_customsettings.xml`.
6. Перезапускаем обработку, заходим в настройки бизнес-процессов.
7. Видим пустой список, загружаем настройки из файла, в который сохраняли типовые.
8. Настройки загрузились, теперь они доступны как пользовательские.
9. Меняем всё что душе угодно.
10. Открываем файл «`customsettings.xml`», форматируем в нормальный вид. Находим ключ `НастройкиБизнесПроцессов_Пользовательские`.
11. Переименовываем этот ключ в `НастройкиБизнесПроцессов_ПоУмолчанию`.
12. Если менялись произвольные коды, аналогично находим ключ `ПроизвольныеКоды_Пользовательские` и переименовываем в `ПроизвольныеКоды_ПоУмолчанию`.
13. Копируем с заменой эти ключи в нужный файл настроек в зависимости от конфигурации —

«`\Git\2.Git_m15\Общий\Обработки 1С\Настройки по-умолчанию`».

Аналогичная ситуация с произвольными кодами:

1. `ПроизвольныеКоды_ПоУмолчанию` — по этому ключу в файле «`default_customsettings.xml`» хранятся

типовые произвольные коды. Они недоступны для изменения из обработки.

2. **ПроизвольныеКоды\_Пользовательские** — по этому ключу в файле «customsettings.xml» хранятся пользовательские произвольные коды. Они доступны для изменения из обработки.

При инициализации произвольные коды собираются из этих двух ключей в единую таблицу (метод `Глядро_СобратьОбщуюСтруктуруПроизвольныхКодов()`) + проходит обновление настроек, если версия настроек ниже используемой в ядре. При экспорте бизнес-процессов в файл также экспортируются и произвольные коды

В прошлых релизах обработки пользовательские произвольные коды сохранялись в 4 ключа:

- **ПроизвольныеКоды\_Выгрузка\_ШапкаДокумента.**
- **ПроизвольныеКоды\_Выгрузка\_ТабличнаяЧастьДокумента.**
- **ПроизвольныеКоды\_Загрузка\_ШапкаДокумента.**
- **ПроизвольныеКоды\_Загрузка\_ТабличнаяЧастьДокумента**, но потом стали храниться в одном ключе **ПроизвольныеКоды\_Пользовательские**, поэтому, для обратной совместимости с прошлыми релизами обработки, существует метод `Глядро_ОбъединитьПроизвольныеКоды()`, который объединяет все пользовательские произвольные коды в один ключ **ПроизвольныеКоды\_Пользовательские**.

Не нашли что искали?



Задать вопрос в техническую поддержку

# Обновление версии структуры настроек Mobile SMARTS

Последние изменения: 2024-03-26

Подробно про механизм хранения настроек в базе Mobile SMARTS и их конвертацию можно прочитать [здесь](#).

В типовых настройках (файл «default\_customsettings.xml») хранится отдельная служебная настройка по имени «ВерсияНастроек\_ПоУмолчанию» (ее значение представляет собой строку, например, «1.0.1.2»). При выходе нового релиза возможна ситуация, когда изменится структура настроек (добавляются новые настройки, удаляются существующие настройки, меняется тип значений существующих настроек). В этом случае версия настроек по умолчанию будет иметь другое значение, например, «1.0.1.3». Поэтому, чтобы обеспечить работоспособность уже существующих у пользователя настроек в новом релизе, предусмотрен механизм обновления структуры настроек.

В некоторых случаях, при обновлении своей текущей [базы Mobile SMARTS](#) до нового релиза, пользователь только копирует новые файлы обработки-ядра и [интеграционной обработки](#), а файл «default\_customsettings.xml» не копирует, он остается тем же, каким он был при первом разворачивании текущей базы Mobile SMARTS из шаблона. Поэтому в самой обработке-ядре существует переменная, отвечающая за версию настроек – ВерсияОбработки, например, «1.3.0.1», она получается из метода СведенияОВнешнейОбработке().ВерсияСтруктурыНастроек.

В кэше хранятся:

- ВерсияОбработки — версия из модуля обработки;
- ВерсияОбработкиКлеверенсТСД, она же ВерсияНастроек\_ПоУмолчанию — версия обработки по умолчанию из «default\_customsettings.xml»;
- ВерсияОбработкиНастроек — версия из «customsettings.xml»;
- ВерсияНастроек — из «customsettings.xml», нужна для хранения версии типовых бизнес-процессов.

## Обновление настроек при инициализации ядра

Краткий стек вызовов при инициализации ядра:

```
ГЛядро_ПолучитьДанныеДляИнициализации => ГЛядро_ВыполнитьОбновлениеСтруктурыНастроек =>
ГЛядро_ВыполнитьОбработчикОбновления => ГЛядро_ОбновлениеНастроекДоВерсии_1_0_1_1
..... => ГЛядро_ОбновлениеНастроекДоВерсии_x_x_x_x =>
ГЛядро_ПроверитьОбновлениеТиповыхБизнесПроцессов - конвертация будет выполняться для обратной
совместимости, независимо от версии настроек => ГЛядро_ОбновитьСтруктуруНастроекБизнесПроцессов
=> ГЛядро_КонвертироватьВыгрузкуПолей - вызывается 2 раза для шапки и для ТЧ      =>
ГЛядро_КонвертироватьЗагрузкуПолей - вызывается 2 раза для шапки и для ТЧ
```

Подробный стек вызовов при инициализации ядра:

```
=> ГЛядро_ПолучитьДанныеДляИнициализации() пСтруктураНастроек = Новый
Структура("ВерсияОбработкиНастроек,ВерсияОбработкиКлеверенсТСД",Неопределено,Неопределено);
пСтруктураНастроек =
ГЛядро_ПолучитьНастройкиНаСервереSMARTS(пСтруктураНастроек,,,АдресаПеременных); //
пСтруктураНастроек.ВерсияОбработкиНастроек = 1 003 000 001 (берется из customsettings.xml) //
пСтруктураНастроек.ВерсияОбработкиКлеверенсТСД = Ложь (откуда? из default? в customsettings такого ключа
нет) => ГЛядро_ВыполнитьОбновлениеСтруктурыНастроек() ВерсияНастроек_ПоУмолчанию =
пСтруктураНастроек.ВерсияОбработкиКлеверенсТСД // ВерсияНастроек_ПоУмолчанию = 0
ВерсияОбработки = СведенияОВнешнейОбработке().ВерсияСтруктурыНастроек; // ВерсияОбработки = 1 003 000
001 (берется из модуля обработки) ВерсияНастроек = пСтруктураНастроек.ВерсияОбработкиНастроек; //
```



ВерсияНастроек = 1 003 000 001

## 1. Если КлючНастройки = Неопределено, Тогда // т.е. обновляем все настройки ИзмененияВКэше = Новый

Структура("ВерсияОбработки,ВерсияОбработкиНастроек,ВерсияОбработкиКлеверенсТСД",ВерсияОбработки,ВерсияНастроек,ВерсияНастроек\_ПоУмолчанию); // версия из модуля обработки, версия из кастом, версия из дефолта ИзменениеВНастройках = Новый Структура("ВерсияОбработкиНастроек,ВерсияОбработкиКлеверенсТСД"); // + хранит ключи настроек, которые необходимо записать в настройки MS Если версия из модуля обработки > версия из кастом сеттингс, тогда =>

Глядро\_ВыполнитьОбработчикОбновления() // Обновление настроек, которые после реструктуризации отправятся в базу SMARTS и сохранится версия настроек (customsettings) (в структуре изменится только ИзменениеВНастройках.ВерсияОбработкиНастроек)

Если версия из модуля обработки > Версия по умолчанию (из default?) тогда =>

Глядро\_ВыполнитьОбработчикОбновления() // Обновление настроек, которые сохранить нет возможности, а все изменения возвращаются в кэш без сохранения (default\_customsettings) (в структуре изменится ВерсияОбработкиКлеверенсТСД) в цикле вызываем все обработчики обновления ИзмененияВСтруктуре.Вставить(?)

(ОбновлениеКэша,"ВерсияОбработкиКлеверенсТСД","ВерсияОбработкиНастроек"),МассивПереходныхВерсий[Индекс] // последняя версия, на которую удалось перейти без ошибок

ИзмененияВКэше.Вставить("ВерсияОбработкиНастроек",ИзменениеВНастройках.ВерсияОбработкиНастроек); Возврат Новый

Структура("ИзмененияВКэше,ИзменениеВНастройках",ИзмененияВКэше,ИзменениеВНастройках);

## 2. Реструктуризация по ключу настройки.

Если версия из модуля обработки > версия настроек по умолчанию, тогда

Глядро\_ВыполнитьОбработчикОбновления() // Обновление настроек, которые сохранить нет возможности, а все изменения возвращаются в кэш без сохранения (default\_customsettings) Возврат

Структура.ВерсияОбработкиКлеверенсТСД => Глядро\_ПроверитьОбновлениеТиповыхБизнесПроцессов - конвертация будет выполняться для обратной совместимости, независимо от версии настроек

СтруктураНастроек = Новый

Структура("ВерсияНастроек,ВерсияНастроек\_ПоУмолчанию",Неопределено,Неопределено);

СтруктураНастроек =

Глядро\_ПолучитьНастройкиНаСервереSMARTS(СтруктураНастроек,,,АдресаПеременных); ...

преобразование типовых бизнес-процессов из таблицы значений в массив структур и сохранение, но только в кэш ... Глядро\_СохранитьНастройкуВБазеSMARTS("ВерсияНастроек",

СтруктураНастроек.ВерсияНастроек\_ПоУмолчанию,,АдресаПеременных);

Глядро\_ЗаписатьВКэшНастройкуSMARTS("ВерсияНастроек",СтруктураНастроек.ВерсияНастроек\_ПоУмолчанию,АдресаПеременных);

... перенос пользовательских бизнес-процессов из ключа "НастройкиБизнесПроцессов" в ключ "НастройкиБизнесПроцессов\_Пользовательские" ... =>

Глядро\_ОбновитьСтруктуруНастроекБизнесПроцессов - изменение, добавление, удаление колонок из таблицы бизнес-процессов и т.д. => Глядро\_КонвертироватьВыгрузкуПолей - вызывается 2 раза для шапки и для ТЧ => Глядро\_КонвертироватьЗагрузкуПолей - вызывается 2 раза для шапки и для ТЧ ... преобразование типовых произвольных кодов из таблицы значений в массив структур и сохранение, но только в кэш ... ... преобразование пользовательских произвольных кодов и сохранение в настройки и в кэш ...

## Обновление настроек при получении настроек по умолчанию

ЛокГлядро\_ПолучитьНастройкиПоУмолчаниюНаСервереSMARTS, если версия обработки из модуля > версии по умолчанию, то вызываем: => Глядро\_ВыполнитьОбновлениеНастроек - для каждого ключа из структуры настроек и записывает новое значение настройки по ключу, вызывает: =>

Глядро\_ВыполнитьОбработчикОбновления - вызывает по порядку все обработчики настроек вида

Глядро\_ОбновлениеНастроекДоВерсии\_1\_0\_1\_1

## Действия при экспорте настройки в файл

Вызывается при экспорте настройки в файл в формах списков «Узлы», «ГлобальныеПараметры», «БизнесПроцессы», «ОтборыСправочников», «ПроизвольныеКоды», «ПроизвольныеОтборыДокументов».

```
СтруктураНастроек = Новый Структура; СтруктураНастроек.Вставить("ТаблицаУзлов",
ПолучитьСтруктуруУзловДляВыгрузки(Истина));
СтруктураНастроек.Вставить("ВерсияОбработкиНастроек", _ЛокКонтекст.ЛокЯдро_ПолучитьИзКэшаНастройкуSMARTS
// это версия из кастом сеттингс, сохраняемая в файл, которая, впоследствии, может стать устаревшей,
неактуальной _ЛокКонтекст.ЛокЯдро_СохранитьНастройкиВФайл(СтруктураНастроек, "НастройкиУзлов");
```

Для бизнес-процессов и произвольных кодов делается несколько иные преобразования, т.к. эти настройки хранятся по нескольким ключам, см. код в соответствующих формах списков.

## Действия при импорте настройки из файла

```
Вызывается при импорте из файла настроек в формах списков «Узлы», «ГлобальныеПараметры»,
«БизнесПроцессы», «ОтборыСправочников», «ПроизвольныеКоды», «ПроизвольныеОтборыДокументов»
СтруктураНастроек = _ЛокКонтекст.ЛокЯдро_ПолучитьНастройкиИзФайла("НастройкиУзлов"); Если
СтруктураНастроек <> Неопределено Тогда //Выполним обновление - Начало ВерсияОбработки
= _ЛокКонтекст.ЛокЯдро_ПолучитьИзКэшаНастройкуSMARTS("ВерсияОбработки"); // а это текущая версия
настроек, из модуля обработки Если СтруктураНастроек.Свойство("ВерсияОбработкиНастроек") Тогда
ВерсияОбработкиНастроек = СтруктураНастроек.ВерсияОбработкиНастроек; Иначе
ВерсияОбработкиНастроек = 0; КонецЕсли; Если ВерсияОбработки > ВерсияОбработкиНастроек
Тогда // если текущая версия выше, чем версия из файла, тогда выполняем обработчики обновления
СтруктураВызова = Новый Структура; СтруктураВызова.Вставить("ТипМетода", "Функция");
СтруктураВызова.Вставить("ИмяМетода", "ВыполнитьОбновлениеНастроек"); // алгоритм такой же как при
получении настроек по умолчанию СтруктураВызова.Вставить("ПараметрыМетода", Новый
Структура("ВерсияНастроек, СтруктураНастроек", ВерсияОбработкиНастроек, СтруктураНастроек));
СтруктураНастроек =
_ЛокКонтекст.Транзит_ВызватьГлобальныйМетодНаКлиенте(СтруктураВызова, _ЛокКонтекст.АдресаПеременных);
КонецЕсли; //Выполним обновление - Конец
```

Для бизнес-процессов и произвольных кодов делается несколько иные преобразования, т.к. эти настройки хранятся по нескольким ключам, см. код в соответствующих формах списков.

Не нашли что искали?



Задать вопрос в техническую поддержку



# Алгоритм работы с основной базой и базой для обмена на примере обработки продукта Mobile SMARTS

Последние изменения: 2024-03-26

## Общее правило:

База для обмена используется в форме обмена документами в процессе загрузки/выгрузки документов и в формах выгрузки справочников и номенклатуры для выгрузки/очистки справочников, во всех остальных случаях используется основная база.

База для обмена определяется при открытии обработки или изменении текущего торгового объекта на главной форме по следующему алгоритму:

1. Если текущий торговый объект не выбран, то базой для обмена считается основная база.
2. Если текущий торговый объект выбран, то:
  - Если у него установлен переключатель, «Использовать основное подключение», то базой для обмена считается основная база.
  - Если у него установлен переключатель, «Использовать частную базу», то базой для обмена считается выбранная частная база.

## Форма списка бизнес-процессов

- сохранение таблицы настроек бизнес-процессов
  - выгрузка доп.таблицы «Бизнес-процессы»
  - получение таблицы настроек БП и ТЧ с настройками
  - получение списка типов документов MS
  - восстановление настроек БП «по умолчанию»
- 
- в основную базу
  - во все базы: основная + все частные
  - из основной базы
  - из основной базы
  - из основной базы

## Форма настройки бизнес-процесса

- получение списка пользователей
- 
- из основной базы

## Форма списка настроек обмена документами

- сохранение и восстановление ТЧ с настройками
- восстановление настроек «по умолчанию»

- в основной базе
- из основной базы

#### Форма списка настроек обмена справочниками

- сохранение и восстановление ТЧ с настройками
- в основной базе

#### Форма списка торговых объектов

- сохранение и восстановление ТЧ с настройками
- выгрузка доп.таблицы «Узлы»
- в основной базе
- во все базы: основная + все частные

#### Форма настройки выгрузки номенклатуры

- настройки компоновки загружаются/сохраняются
- номенклатура выгружается
- очистка номенклатуры
- из основной базы
- в базу для обмена
- в базе для обмена

#### Форма настройки выгрузки справочников

- настройки компоновки загружаются/сохраняются
- справочник выгружается
- очистка справочника
- из основной базы
- в базу для обмена
- в базе для обмена

#### Форма настройки полей обмена

- сохранение и восстановление ТЧ с настройками
- в основной базе

#### Форма настройки произвольного кода

- сохранение и восстановление ТЧ с настройками
- в основной базе

#### Форма выбора табличной части

- получение списка доп.таблиц документа MS
- получение списка полей ТЧ документа MS
- из основной базы
- в основной базе

#### Форма настройки торгового объекта

- сохранение и восстановление ТЧ с настройками
- получение дерева пользователей
- выгрузка доп.таблицы «Узлы»
- в основной базе
- из основной базы
- во все базы: основная + все частные

#### Форма выбора параметра торгового объекта

- восстановление ТЧ с настройками
- в основной базе

#### Форма выбора торгового объекта

- восстановление ТЧ с настройками
- в основной базе

#### Форма выбора произвольного кода

- сохранение и восстановление ТЧ с настройками

- в основной базе

## Форма просмотра документа Mobile SMARTS

- получение данных документа MS
- получение данных шапки документа MS

- из базы для обмена
- из базы для обмена

## Форма обмена документами

При открытии:

- получение ТЧ с настройками
- метод «СформироватьДеревоПользователейMS»
- команда «ЗаполнитьСписокДокументовMS» получает шапки документов
- команда «УдалитьДокументMS»
- команда «ОткрытьФормуДокументаMS» получает список колонок ТЧ «СтрокиПлан» и «СтрокиФакт»

- из основной базы
- из базы для обмена
- из базы для обмена
- из базы для обмена
- из базы для обмена

Выгрузка документов на ТСД:

- таблицу настроек БП получаем
- выгрузка документа идет

- из основной базы
- в базу для обмена

Загрузка документов с ТСД:

- таблицу настроек БП получаем
- при поиске подходящего БП получаем данные документа MS и шапку документа MS
- удаление документа после загрузки
- получение списка доп.таблиц, получение данных из доп.таблиц
- получение данных документа MS для загрузки в 1С
- получение данных документа MS для создания новых штрихкодов

- из основной базы
- из базы для обмена
- из базы для обмена
- из базы для обмена
- из базы для обмена
- из базы для обмена

## Форма выбора бизнес-процесса

При открытии:

- получение списка типов документов MS
- из базы для обмена

## Форма главная

- при выборе и очистке текущего узла получаем базу для обмена: «Объект.IDБазыMSДляОбмена = ПолучитьIDБазыMSДляОбмена();»
- при открытии формы и после удачного подключения базы MS в методе «СоздатьОбъектДрайвера»:

- получаем параметры подключения (ид, имя, строка подключения, имя приложения, ид приложения)
- восстанавливаем настройки в методе «ВосстановитьНастройки» (здесь восстанавливаются флажки выгрузки справочников и параметры весового товара)
- восстановление ТЧ с настройками

- из основной базы
- из основной базы
- из основной базы

получаем базу для обмена: «Объект.IDБазыMSДляОбмена = ПолучитьIDБазыMSДляОбмена();»

- при закрытии формы или при скрывании группы настройки справочников (если есть изменения и нажали «не сохранять изменения»), в методе «СохранитьНастройкиЗавершение» восстанавливаем настройки
  - «СохранитьНастройкиВыгрузкиСправочниковMS» - сохраняет флажки и параметры весового товара
- 
- из основной базы
  - в основную базу

---

Не нашли что искали?



[Задать вопрос в техническую поддержку](#)