

Обновление версии структуры настроек Mobile SMARTS

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 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 настом, версия из дефолта
ИзменениеВНастройках = Новый Структура("ВерсияОбработкиНастроек, ВерсияОбработкиКлеверенсТСД"); // + хранит ключи
настроек, которые необходимо записать в настройки MS Если версия из модуля обработки > версия из настом 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 изменен
возвращаются в кэш без сохранения (default_customsettings) (в структуре изменится ВерсияОбработкиКлеверенсТСД) в цикле
вызываем все обработчики обновления ИзмененияВСтруктуре.Вставить?
(ОбновлениеКэша,"ВерсияОбработкиКлеверенсТСД","ВерсияОбработкиНастроек"),МассивПереходныхВерсий[ИндексВерсии].Эт
// последняя версия, на которую удалось перейти без ошибок
```

Изменения ВКэше. Вставить("ВерсияОбработкиНастроек", Изменение В Настройках. ВерсияОбработкиНастроек); Возврат Новый Структура("Изменения ВКэше, Изменение В Настройках", Изменения ВКэше, Изменение В Настройках);

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 изменен
возвращаются в кэш без сохранения (default_customsettings) Возврат Структура.ВерсияОбработкиКлеверенТСД =>
Глядро_ПроверитьОбновлениеТиповыхБизнесПроцессов - конвертация будет выполняться для обратной совместимост
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, АдресаП
... 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 дл
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

ЛокЯдро_ПолучитьНастройкиПоУмолчаниюНаСервереSMARTS, если версия обработки из модуля > версии по умолчанию, 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("ВерсияОбр
// это версия из настом 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 нескольк 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
ВерсияОбработкиНастроек > 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 настро
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 нескольк ключам, см. код в соответствующих формах списков.

Была ли статья полезна?

<input type="radio"/>	Нет
<input type="radio"/>	Да

