

# **Руководство администратора SAUMI 4.11 (предварительное издание)**

**ООО «БФТ»**

---

# **Руководство администратора : SAUMI 4.11 (предварительное издание)**

ООО «БФТ»

Дата публикации 18.11.2021

Авторские права © 1999 – 2021 ООО «БФТ»

Настоящее руководство подготовлено с применением технологии *DocBook*

Все права на данное издание защищены Законодательством РФ. Разрешается копирование и воспроизведение в любом виде, в том числе электронном, исключительно в целях служебного использования. Запрещается передача данного издания третьим лицам. Все товарные знаки или зарегистрированные товарные знаки, упоминаемые в настоящем издании, являются собственностью соответствующих правообладателей.

---

# Содержание

Предисловие .....	9
1. «Диспетчер пользователей» .....	13
1.1. Ведение пользователей SAUMI .....	13
1.1.1. Создание новой учетной записи пользователя .....	14
1.1.2. Редактирование учетной записи пользователя .....	15
1.1.3. Удаление учетной записи пользователя .....	15
1.1.4. Смена статуса учетной записи пользователя .....	15
1.2. Полномочия учетной записи пользователя .....	16
1.2.1. Предоставление и изъятие полномочий учетной записи пользователя .....	16
1.2.2. Создание новой роли .....	19
1.2.3. Удаление роли с префиксом SM_ .....	20
1.3. Управление доступом к информационным объектам SAUMI .....	21
1.3.1. Предоставление полномочий доступа .....	23
1.3.2. Разграничение прав доступа к финансовым обязательствам по их типам .....	27
1.4. Регистрация базовых собственников .....	30
1.4.1. Регистрация нового базового собственника .....	30
1.4.2. Предоставление полномочий агентов распорядителя базового собственника .....	31
2. «Менеджер обновлений» .....	37
2.1. Порядок установки обновления .....	37
2.2. Компиляция объектов с ошибками .....	39
2.3. Переиндексация поиска .....	39
2.4. Обновление отчетов, печатных форм, форм сбора отчетных данных и модуля SAUMI-Pascal .....	39
2.4.1. Обновление отчетов (настроек поиска) .....	40
2.4.2. Обновление печатных форм .....	40
2.4.3. Обновление форм сбора отчетных данных в электронном виде .....	40
2.4.4. Обновление процедур SAUMI-Pascal .....	41
3. Управление репозитарием файлов SAUMI .....	43
3.1. Механизм локального обновления SAUMI .....	43
3.1.1. Управление локальным обновлением .....	44
3.2. Вызов утилиты управления репозитарием .....	45
3.3. Копирование файла из репозитария на пользовательский компьютер .....	46
3.4. Добавление нового файла в репозитарий .....	46
3.5. Задание настроек выгрузки файла репозитария .....	48
3.6. Изменение имеющегося в репозитарии файла .....	50
3.7. Изменение настроек выгрузки хранимого в репозитарии файла .....	51
3.8. Добавление группы файлов в репозитарий .....	52
3.9. Удаление файла из репозитария .....	56
4. «Диспетчер настроек» .....	57
4.1. Маски ввода кадастровых номеров .....	58
4.2. Параметры обмена данными отчетности .....	58
4.2.1. Формирование и настройка макета диска сбора данных .....	58
4.2.2. Местоположение формируемого набора файлов отчетных данных .....	59
5. Файлы настроек главного исполняемого модуля SAUMI .....	61
5.1. Файл saumi.ini .....	61
5.1.1. Настройки считывателя штрих-кода – секция [BarCode] .....	62
5.1.2. Настройки протокола WebDAV – секция [DSW_WebDAV] .....	62
5.2. Файл saumi_user_params.ini .....	64
5.2.1. Параметры последнего соединения с БД SAUMI – секция [Common] .....	64
5.2.2. Коды горячих клавиш Главной панели SAUMI – секция [HotKeys] .....	64
5.2.3. Параметры поведения Главной панели SAUMI – секция [MainPanel] .....	65

5.2.4. Параметры оптимизации производительности клиентской части SAUMI – секция [Performance] .....	66
5.3. Общие секции для файлов saumi.ini и saumi_user_params.ini .....	66
5.3.1. Параметры отображения окон данных – секция [ExplorerWindow] .....	66
5.3.2. Журналирование программных событий – секция [Logging] .....	67
6. Аудит изменения БД SAUMI .....	69
6.1. Настройка аудита .....	69
6.2. Данные аудирования .....	72
6.3. Отчеты, представляющие и сопровождающие аудит .....	73
6.4. Архивирование данных аудита .....	75
7. «Монитор подключений» .....	77
8. «Менеджер реквизитов» .....	79
8.1. «Менеджер реквизитов» – беглый взгляд .....	80
8.2. Термины и определения .....	81
8.3. Порядок создания отображаемого дополнительного реквизита .....	82
8.4. Создание поля, представляющего дополнительный реквизит .....	83
8.5. Формирование описания отображаемого дополнительного реквизита .....	86
8.5.1. Формирование описания в «Менеджере реквизитов» .....	87
8.5.2. Указание вкладки – ключ TAB .....	90
8.5.3. Поле ввода – ключ FIELD .....	91
8.5.4. Маска ввода для полей типа CHAR, VARCHAR, VARCHAR2 .....	92
8.5.5. Отрисовка значения реквизита – ключ RFIELD .....	94
8.5.6. Ссылка на экземпляр сущности – ключ LFIELD .....	95
8.5.7. Элемент управления кнопка – ключ BUTTON .....	97
8.5.8. Вывод надписи – ключ LABEL .....	97
8.5.9. Прорисовка рамки – ключ FRAME .....	98
8.5.10. Прорисовка линии – ключ LINE .....	98
8.5.11. Условие отображения реквизита на форме – ключ OPTION .....	98
8.6. Дополнительные настройки реквизитов .....	103
8.6.1. Настройка флажка .....	103
8.6.2. Ограничение ссылочного поля .....	104
8.6.3. Допустимость отсутствия значения и значение по умолчанию .....	104
8.6.4. Сохранение истории изменений значения реквизита .....	105
8.6.5. Учет реквизита в условии поиска .....	106
8.6.6. Создание списка значений реквизита .....	106
8.6.7. Подстановка синонимов в список значений реквизита .....	107
8.7. Восстановление подсистемы поиска после работы в «Менеджере реквизитов» .....	108
8.8. Перечень таблиц, принимающих отображаемые дополнительные реквизиты .....	108
8.9. Ограничения имен полей, представляющих дополнительные реквизиты .....	110
9. «Менеджер отчетов» .....	111
9.1. Составление SQL-запроса .....	111
9.2. Разложение SQL-запроса на составляющие .....	112
9.3. Добавление активности (Свойства) PRIMARY .....	114
9.4. Ссылка на связанные данные .....	114
9.5. Видимость столбцов .....	115
9.6. Дополнительная специальная информация по столбцам .....	115
9.7. Флаг суммируемости .....	116
9.8. Специальные колонки (псевдостолбцы) .....	116
9.8.1. Описание общих секций FROM и WHERE .....	116
9.8.2. PRIMARY – Главный ключ .....	116
9.8.3. RESTRICTION – Ограничение .....	116
9.8.4. ORDER BY – Принудительная сортировка .....	117
9.8.5. DICT – Группировочное поле аналитического отчета .....	117
9.8.6. DATE – Группировочное поле даты аналитического отчета .....	117

---

9.8.7. INDICATOR – Выделение строк цветом и стилем .....	118
10. Параметрические SQL-запросы .....	119
10.1. Введение параметра в SQL-запрос .....	119
10.2. Примеры использования .....	120
11. «Менеджер форм» .....	121
11.1. Создание новой формы .....	122
11.1.1. Тип печатной формы .....	122
11.1.2. Быстрый вызов .....	123
11.1.3. Доступны для роли .....	123
11.1.4. Отладочный набор элементов .....	123
11.2. Дизайнер отчетов .....	123
11.2.1. Построение простейшего автономного отчета .....	125
11.2.2. Привязка списка к данным в окне .....	126
11.2.3. Оформление печатной формы .....	129
11.3. Формирование текстов документов .....	130
11.4. Экранные формы .....	131
11.5. Произвольные диалоговые окна .....	132
11.6. Ограничения поиска связанных данных .....	133
12. «Менеджер справочников» .....	135
12.1. Навигация в перечне справочников .....	135
12.2. Создание нового справочника .....	137
12.3. Изменение названия справочника .....	139
12.4. Добавление поля в справочник .....	139
12.5. Добавление записей в справочнике, редактирование записей .....	141
12.6. Назначение (изменение) дополнительного отображаемого столбца .....	142
12.7. Преобразование линейного справочника в древовидный .....	143
12.8. Назначение принудительной загрузки отдельных справочников пользовательскими рабочими станциями .....	144
12.9. Назначение прав доступа к отдельным справочникам .....	145
12.10. Объединение элементов-дубликатов .....	145
12.11. Автоматическая замена КБК в закрытых договорах .....	146
12.12. Службная таблица DICTIONARIES – перечень справочников .....	149
13. Редактор SAUMI-Pascal .....	151
13.1. Вызов редактора SAUMI-Pascal .....	151
13.2. Строка меню и панель инструментов .....	154
13.2.1. Меню Файл .....	154
13.2.2. Меню Правка .....	155
13.2.3. Меню Поиск .....	156
14. Работа с расчетными коэффициентами .....	159
14.1. «Менеджер коэффициентов» .....	160
14.1.1. Создание нового коэффициента .....	160
14.1.2. Удаление коэффициента .....	161
14.1.3. Изменение описания коэффициента .....	162
14.1.4. Добавление периода .....	162
14.1.5. Удаление периода коэффициента .....	164
14.1.6. Редактирование значений коэффициента .....	164
14.1.7. Сортировка расчетных коэффициентов .....	167
14.2. Использование коэффициентов при автоматическом расчете .....	169
14.2.1. SAUMI-Pascal .....	169
14.2.2. .NET .....	170
14.2.3. Ошибки .....	170
15. «Политика операций» .....	173
15.1. Вкладка «Параметры» .....	173
15.1.1. Параметр «Непрерывный» .....	174

---

15.1.2. Параметр «Реестр» .....	174
15.1.3. Параметр «Площади» .....	174
15.1.4. Параметр «Обязательства» .....	175
15.1.5. Параметр «Авторасчет» .....	175
15.1.6. Параметр «Ценные бумаги» .....	175
15.1.7. Параметр «Назначения» .....	175
15.1.8. Параметр «Состояние» .....	175
15.1.9. Параметр «Доля права» .....	176
15.1.10. Параметр «Один период» .....	176
15.1.11. Параметр «Один субъект» .....	176
15.2. Вкладка «По умолчанию» .....	176
15.2.1. Параметр «Обозначение субъекта» .....	177
15.2.2. Параметр «Субъект по умолч.» .....	177
15.2.3. Параметр «Эксклюзивный» .....	177
15.2.4. Параметр «Скрыть» .....	177
15.2.5. Параметр «Документ» .....	177
15.2.6. Параметр «Основание» .....	178
15.2.7. Параметр «Группа меню» .....	178
15.3. Вкладка «Ограничения» .....	178
15.4. Вкладка «Применимость» .....	180
15.5. Вкладка «Применимость в структуре» .....	181
16. Администрирование работы с документами .....	183
16.1. Механизм поддержки автоматической нумерации документов .....	183
16.2. Настройки сохранения файлов документов .....	184
16.3. WebDAV-сервер хранилища документов SAUMI .....	185
16.3.1. Общие положения .....	185
16.3.2. Тип блокировки .....	185
16.3.3. Настройка типа блокировки .....	186
16.3.4. Включение/отключение работы с WebDAV .....	186
16.3.5. Ассоциированные приложения .....	186
16.3.6. Сетевые настройки .....	187
16.3.7. Протоколирование WebDAV .....	188
16.3.8. Проблема WebDAV-клиента Excel 2003 с кириллическими именами .....	188
16.3.9. Генерация имен файлов документов WebDAV-сервером SAUMI .....	189
16.3.10. Отключение протокола WebDAV для некоторых типов файлов .....	190
16.3.11. Активация новых настроек WebDAV .....	191
17. Настройка шаблонов документов .....	193
17.1. Назначение шаблона виду документа .....	195
17.2. Удаление шаблона .....	195
17.3. Порядок работы с шаблонами MS Word .....	196
17.3.1. Назначение шаблона MS Word .....	196
17.3.2. Создание шаблона MS Word .....	198
17.3.3. Переменная CONTEXT .....	199
17.3.4. Примеры кода SAUMI-Pascal для полей слияния .....	200
17.3.5. Особенности назначения шаблонов MS Word .....	203
17.4. Порядок работы с HTML-шаблонами .....	203
17.4.1. Структура HTML-шаблона .....	204
17.4.2. Назначение HTML-шаблона .....	204
17.4.3. Создание пустого шаблона .....	205
17.4.4. Загрузка HTML-шаблона из spz-файла .....	206
17.4.5. Редактирование назначенного HTML-шаблона .....	206
17.4.6. Интерфейс создания / редактирования шаблона .....	206
17.4.7. Настройка конфигурации шаблона .....	209
17.4.8. Принципы программирования шаблона .....	211

17.4.9. Формирование динамических фрагментов текста .....	212
17.4.10. Запросы к базе данных .....	214
17.4.11. Использование списков .....	215
17.4.12. Использование стилей .....	217
17.4.13. Обработка штрих-кодов .....	218
17.4.14. Отладка HTML-шаблона при помощи Microsoft Visual Studio .....	219
18. Считыватель штрих-кодов .....	223
18.1. Оборудование для работы с штрих-кодами .....	223
18.1.1. Печать изображения штрих-кода .....	223
18.1.2. Считывание штрих-кода .....	224
18.2. Кодирование информации в штрих-коде .....	224
18.3. Настройка SAUMI для работы со сканером штрих-кодов .....	225
18.4. Настройка SAUMI на произвольную обработку считанного штрих-кода .....	225
19. Сбор отчетных данных в электронном виде .....	227
19.1. Организационная схема применения DataStretch .....	227
19.2. Задание нового отчетного периода .....	228
19.3. Формирование набора форм .....	230
19.4. Новые и измененные формы .....	232
19.5. Активирование нового отчетного периода .....	236
19.6. Удаление отчетного периода .....	236
19.7. Набор файлов отчетности .....	237
19.8. Формат mro-файла .....	237
19.8.1. Секция #OPTIONS .....	237
19.8.2. Секция #DATAOPT .....	238
19.8.3. Секция #SAVEDATA .....	242
19.8.4. Секция #EXEC_SAVE .....	243
19.8.5. Секция #EXEC_LOAD .....	244
19.8.6. Секция #LISTS .....	247
19.8.7. Порядок обработки принимаемых отчетных данных .....	248
20. Настройка расчета доходов будущих периодов .....	251
21. Дополнения (плагины) – общие сведения .....	255
22. Модуль выгрузки данных в соответствии с приказом Минфина РФ No 47н .....	259
22.1. Необходимые элементы и структуры данных .....	260
22.2. Необходимое информационное обеспечение .....	260
22.3. Работа плагина .....	262
A. Перечень таблиц .....	265
B. Перечень серверных методов .....	269
C. Роли (привилегии) .....	271

---

# Предисловие

Настоящее руководство содержит сведения, требуемые администратору программного комплекса SAUMI для выполнения различных настроечных действий и поддержания комплекса в рабочем состоянии.

После развертывания программного комплекса SAUMI на технических средствах наипервейшей задачей администратора становится предоставление возможности авторизации, а также полномочий на совершение различных действий персоналу Организации, привлекаемому к работе с комплексом. Создание учетных записей пользователей, а также методы разграничения прав доступа к различным элементам информации рассматривается в Главе 1 «*Диспетчер пользователей*».

Программный комплекс SAUMI является развивающимся программным продуктом. На сайте <http://support.saumi.ru> периодически публикуются сведения об обновлении комплекса до новых версий. Порядок выполнения обновлений версий приводится в Главе 2 «*Менеджер обновлений*».

Глава 3 «*Управление репозитарием файлов SAUMI*» раскрывает механизм обновления файлов и папок клиентской части программного комплекса SAUMI на пользовательских компьютерах посредством централизованного репозитария файлов комплекса. Приводятся сведения о внесении изменений в репозиторий файлов вручную, что может представлять интерес при выполнении собственных расширений возможностей комплекса, а также при получении индивидуальных изменений/добавлений некоторых файлов, полученных из группы сопровождения SAUMI.

Ряд настроек комплекса представляет собой набор некоторых константных значений, оказывающих глобальное влияние на различные аспекты работы комплекса однообразно на всех пользовательских компьютерах. Такие глобальные параметры задаются посредством утилиты «*Диспетчер настроек*», рассматриваемой в Главе 4.

Некоторые настроечные параметры могут задаваться индивидуально в привязке к пользовательским компьютерам и отдельным пользователям. Такие параметры указываются в файлах настроек главного исполняемого модуля SAUMI, а их перечень и назначение приводятся в Главе 5 «*Файлы настроек главного исполняемого модуля SAUMI*».

Сбору информации о произведенных пользователями изменениях данных, содержащихся в БД SAUMI, а также обеспечению визуального представления детальной истории зафиксированных изменений посвящена Глава 6 «*Аудит изменения БД SAUMI*».

Для выполнения некоторых операций требуется отключение всех пользователей, кроме привилегированных. Для определения всех текущих подключений к БД SAUMI служит утилита «*Монитор подключений*», представленная в Главе 7.

Атрибутивный состав информационной модели, реализованной в БД SAUMI, может быть недостаточным для потребностей конкретного Комитета. Техника дополнительных реквизитов, расширяющих информационную модель учета, представлена в Главе 8 «*Менеджер реквизитов*».

Значимым повседневным действием пользователей SAUMI является поиск информации, соответствующей определенным критериям, с последующей визуализацией результатов поиска. Специальным образом скомпонованные метаданные, определяющие критерий поиска, состав выборки, а также параметры ее отображения называются отчетом. Результат вызова отчета (исполнения отчета) отображается в виде соответствующего окна данных. Техника построения отчетов описывается в Главе 9 «*Менеджер отчетов*».

Расширение техники построения запросов в отчетах приводится в Главе 10 «*Параметрические SQL-запросы*».



Генерация текста документа по шаблонному образцу, в котором предусмотрены позиции для вставки контекстно согласованной информации из БД SAUMI, позволяет формировать информационное содержимое документов учетной деятельности пользователей SAUMI. Одним из инструментов, реализующих генерацию документов по шаблонам, является встроенный в SAUMI компонент стороннего разработчика Fast Report, являющийся ядром утилиты, представленной в Главе 11 «*Менеджер форм*».

Под справочником понимается упорядоченный набор уникальных элементов, содержащий краткие и точные сведения, объединённые единой тематикой, причем все элементы одного справочника имеют общий атрибутивный состав. Организация данных с применением справочников повышает точность представления информации, равно как и скорость её обработки, а также минимизирует объём ввода ошибочной информации в БД SAUMI. Ведение многих справочников производится посредством пользовательского интерфейса главного исполняемого модуля SAUMI, однако, специальная утилита «Менеджер справочников» из комплекта администратора предоставляет возможность полного контроля нормативно-справочной информации ПК SAUMI. Указанная утилита рассматривается в Главе 12 «*Менеджер справочников*».

В программный комплекс SAUMI встроен интерпретатор языка программирования, по синтаксису сходного с языком Borland Pascal, который позволяет производить особо тонкую алгоритмическую настройку поведения программного комплекса написанием соответствующих обработчиков – процедур и функций SAUMI-Pascal. Операциям по добавлению, изменению, сохранению и восстановлению программного кода SAUMI-Pascal посвящена Глава 13 «*Редактор SAUMI-Pascal*».

Автоматизация различных расчетов во многих случаях требует привязки к элементам некоторых справочников или их сочетаний значений числовых коэффициентов, представленных в формулах утвержденных методик таких расчетов. Для значений расчетных коэффициентов в SAUMI поддерживается временная версияность. Ведение значений коэффициентов производится с помощью утилиты «Менеджер коэффициентов», приемы работы с которой, а также программное использование коэффициентов представлены в Главе 14 «*Работа с расчетными коэффициентами*».

Настройка разрешений и запретов на оформление различных видов правоотношений (политик учета операций с имуществом), включая их взаимосвязи, производится посредством специальной утилиты, описанию которой посвящена Глава 15 «*Политика операций*».

Глава 16 «*Администрирование работы с документами*» посвящена механизму поддержки автоматической нумерации документов, настройкам сохранения файлов содержимого документов, а также WebDAV-сервера хранилища документов SAUMI, позволяющего работать с содержимым документов нескольким пользователям одновременно.

Помимо указанного выше «Менеджера форм» в SAUMI имеются дополнительные инструменты генерации текста документов на основе MS Word шаблонов, а также HTML-шаблонов, порядок создания и применения которых описывается в Главе 17 «*Настройка шаблонов документов*».

Глава 18 «*Считыватель штрих-кодов*» посвящена применению оборудования для работы с штрих-кодами, кодированию информации штрих-кодами, а также обработке считанного штрих-кода.

Настроечные и организационные действия по сбору и передаче в электронном виде отчетных данных, касающихся имущества, находящегося в хозяйственном ведении или оперативном управлении подотчетных организаций представлены в Главе 19 «*Сбор отчетных данных в электронном виде*».

Настройке отчетности об ожидаемых доходах, производимой в соответствии с положениями федерального стандарта бухгалтерского учета для организаций государственного сектора «Аренда», утвержденного приказом Минфина России от 31 декабря 2016 г. N 258н, посвящена Глава 20 «*Настройка расчета доходов будущих периодов*».

В SAUMI имеется возможность запуска из меню «Дополнения» Главной панели SAUMI автономных программ – исполняемых файлов. В Главе 21 «Дополнения (плагины) – общие сведения» приводятся соглашения, которым должна следовать такая автономная программа.

Глава 22 «Модуль выгрузки данных в соответствии с приказом Минфина РФ No 47н» посвящена применению функциональности, реализованной в виде дополнения, и выполняющей выгрузку данных о земельных участках в соответствии с Приказом Министерства финансов Российской Федерации от 23.03.2006 No 47н «Об утверждении формы „Сведения о земельных участках, расположенных в пределах муниципального образования“ и Рекомендаций по заполнению» и Приказом Федеральной Налоговой Службы от 28 июля 2006 г. No САЭ-3-13/485@ «Об утверждении требований к структуре и содержанию файлов сведений о земельных участках, расположенных в пределах муниципального образования».

В Приложении А «Перечень таблиц» содержится перечень таблиц БД SAUMI, сгруппированных по тематическим разделам.

В Приложении В «Перечень серверных методов» приводится перечень серверных методов (PL/SQL процедур и функций).

В Приложении С «Роли (привилегии)» содержится список ролей Oracle, используемых в SAUMI.

---

# Глава 1. «Диспетчер пользователей»

## Содержание

1.1. Ведение пользователей SAUMI .....	13
1.1.1. Создание новой учетной записи пользователя .....	14
1.1.2. Редактирование учетной записи пользователя .....	15
1.1.3. Удаление учетной записи пользователя .....	15
1.1.4. Смена статуса учетной записи пользователя .....	15
1.2. Полномочия учетной записи пользователя .....	16
1.2.1. Предоставление и изъятие полномочий учетной записи пользователя .....	16
1.2.2. Создание новой роли .....	19
1.2.3. Удаление роли с префиксом SM_ .....	20
1.3. Управление доступом к информационным объектам SAUMI .....	21
1.3.1. Предоставление полномочий доступа .....	23
1.3.2. Разграничение прав доступа к финансовым обязательствам по их типам .....	27
1.4. Регистрация базовых собственников .....	30
1.4.1. Регистрация нового базового собственника .....	30
1.4.2. Предоставление полномочий агентов распорядителя базового собственника .....	31

Утилита «Диспетчер пользователей» предназначена для ведения пользователей SAUMI и их полномочий по выполнению тех или иных функций, а также предоставляет возможность управления правами доступа на уровне таблиц БД SAUMI. Кроме того, с помощью данной утилиты регистрируются базовые собственники и назначаются полномочия агентов их распорядителей (см. «Распорядители, агенты распорядителей, базовые собственники» в главе «Основные понятия SAUMI» руководства пользователя).

## 1.1. Ведение пользователей SAUMI

Перечень учетных записей пользователей содержится на вкладке **Пользователи**. В нижней части окна утилиты на этой вкладке слева отображаются предоставленные пользователю права (роли) на выполнение действий в SAUMI, а в правой нижней части окна перечень всех доступных прав, представленных ролями БД SAUMI.

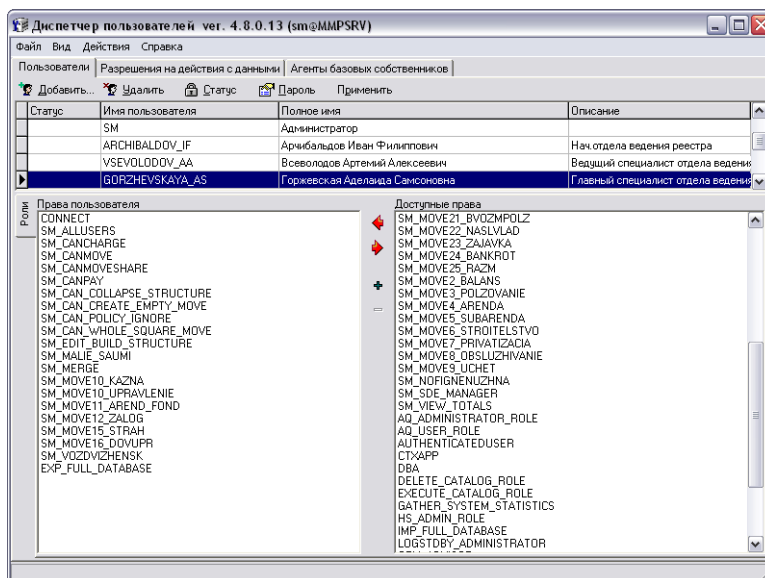


Рисунок 1.1.

### 1.1.1. Создание новой учетной записи пользователя

Для создания новой учетной записи нужно либо нажать кнопку **Добавить** (рис. 1.1) на инструментальной панели, либо нажатием правой кнопки мыши вызвать контекстное меню, в котором выбрать пункт «Добавить пользователя». В результате открывается окно свойств пользователя, которое нужно заполнить соответствующим образом.

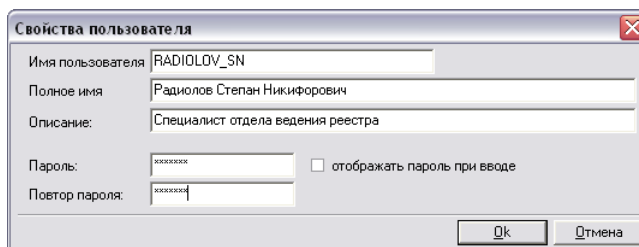


Рисунок 1.2.

В поле *Имя пользователя* вводится последовательность символов, состоящая из букв латинского алфавита, символа подчеркивания '\_' и цифр. Первым символом должна быть буква. Полное имя должно содержать без сокращений фамилию, имя и отчество регистрируемого пользователя. Пароль может содержать любые символы латиницы, цифры и символ подчеркивания. По завершении ввода следует нажать кнопку **Ok**.

### Замечание

После создания учетной записи пользователя следует довести до него (конкретного лица) его имя пользователя и пароль. Исходный пароль пользователю необходимо сменить после первого входа в SAUMI (см. «Смена пароля пользователя» в главе «Начало работы» руководства пользователя).

### 1.1.2. Редактирование учетной записи пользователя

Чтобы изменить учетные данные пользователя SAUMI, нужно либо осуществить двойной щелчок по соответствующей строке в таблице учетных записей (рис. 1.1), либо выделить эту учетную запись и, вызвав правой кнопкой мыши контекстное меню, выбрать в нем пункт «Свойства...». В результате откроется окно свойств пользователя (рис. 1.2), в котором нужно ввести необходимые изменения. По завершении ввода следует нажать кнопку **Ок**.

#### Замечание

Поле *Имя пользователя* редактированию не подлежит.

Также открывается окно свойств выделенного пользователя при нажатии кнопки **Пароль** (рис. 1.1) на инструментальной панели. При этом фокус ввода устанавливается в поле *Пароль*.

### 1.1.3. Удаление учетной записи пользователя

Чтобы удалить имеющуюся учетную запись пользователя SAUMI, ее следует выделить в перечне, а затем либо нажать кнопку **Удалить** (рис. 1.1) на инструментальной панели, либо правой кнопкой мыши вызвать контекстное меню, в котором выбрать пункт «Удалить пользователя». В результате открывается диалоговое окно подтверждения удаления.

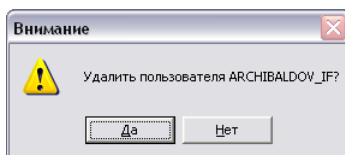


Рисунок 1.3.

В случае подтверждения нажатием на кнопку «Да» выделенная учетная запись пользователя удаляется.

#### Замечание

Учетную запись SM удалить нельзя.

### 1.1.4. Смена статуса учетной записи пользователя

Учетная запись может иметь два значения статуса: «Блокирован» и «Разблокирован» (статус по умолчанию). При блокировании учетной записи вход в SAUMI под этой учетной записью невозможен. Блокированная учетная запись отмечена соответствующей надписью в столбце «Статус» таблицы перечня учетных записей.

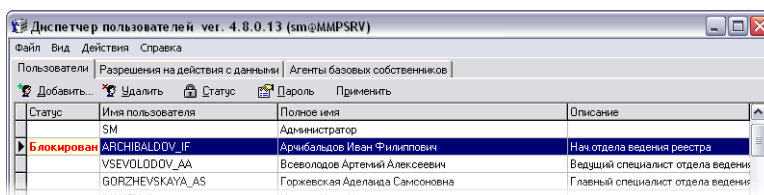


Рисунок 1.4.

Чтобы изменить статус учетной записи пользователя SAUMI, ее нужно выделить в перечне и нажать кнопку **Статус** (рис. 1.4)

## Замечание

Учетная запись SM блокированию не подлежит.

## 1.2. Полномочия учетной записи пользователя

Полномочия выделенной учетной записи пользователя отображаются в нижней части вкладки **Пользователи** (рис. 1.1). Слева отображается список назначенных прав для текущего (указанного в списке) пользователя, справа – список всех возможных доступных прав, которые могут быть назначены данному пользователю.

Отображаемые права – это роли Oracle. Среди них есть, как системные роли Oracle, так и специфичные для SAUMI. Описание системных ролей можно найти в документации по Oracle, а описание специфичных – в Приложении С. Следует отметить, что все специфичные для SAUMI права названы с применением префикса «SM\_».


Оба перечня полномочий, как левый, так и правый (рис. 1.1), упорядочены в следующем порядке:

- роль CONNECT;
- все роли с префиксом SM\_ в алфавитном порядке;
- все остальные роли в алфавитном порядке.

Каждая роль может присутствовать только на одной панели ролей: либо предоставленных выделенному пользователю (левая панель), либо доступных для предоставления (правая панель).

Минимальный набор полномочий пользователю предоставляет роль SM\_ALLUSERS, которая назначается вновь созданной учетной записи пользователя вместе с ролью CONNECT, позволяющей осуществлять соединение с базой данных. Указанные две роли обязательны для входа в SAUMI.

### 1.2.1. Предоставление и изъятие полномочий учетной записи пользователя

В случае необходимости предоставить учетной записи пользователя дополнительные роли нужно выделить эти роли в правом перечне ролей и нажать кнопку  – «Назначить права».

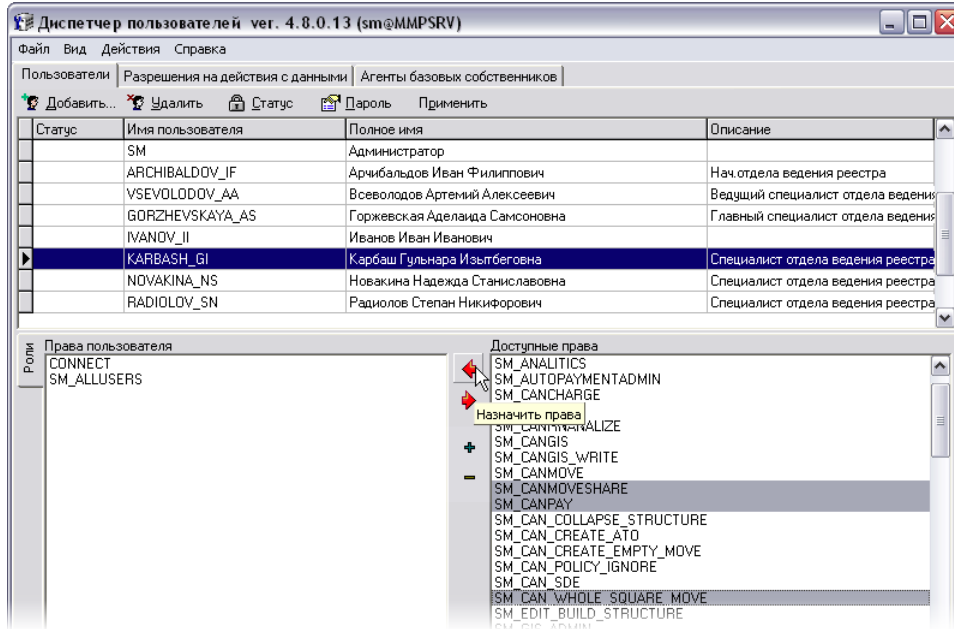


Рисунок 1.5.

В результате учетная запись пользователя получает дополнительные роли, которые переносятся из правого списка ролей в левый.

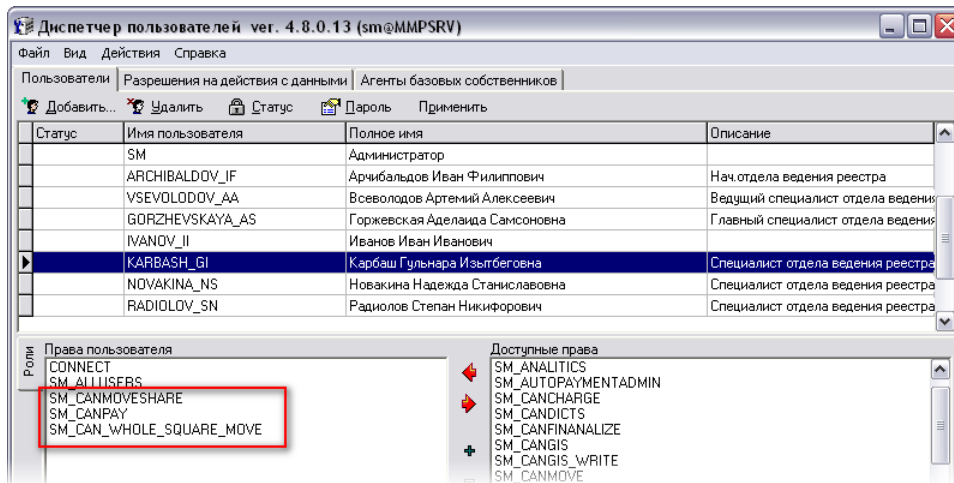



Рисунок 1.6.

Чтобы изъять некоторые роли из учетной записи пользователя, нужно выделить эти роли в левом перечне предоставленных пользователю прав и нажать кнопку  – «Лишить права».

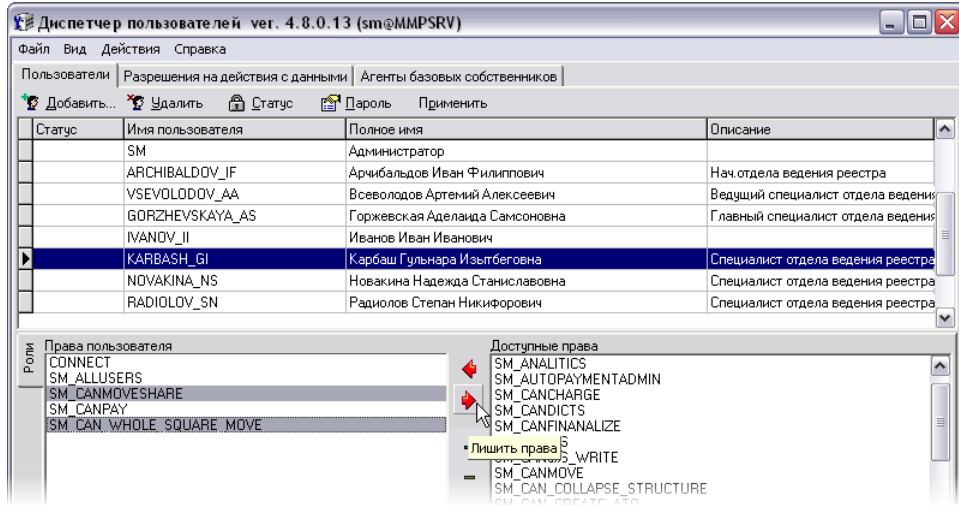


Рисунок 1.7.

В результате изъятые роли исчезают из перечня прав, предоставленных учетной записи пользователя.

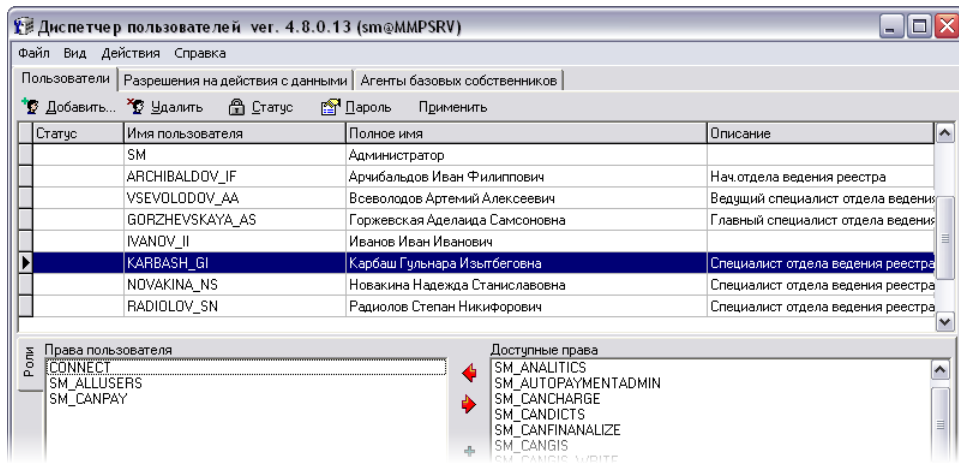


Рисунок 1.8.

Полномочия пользователю по работе с БД можно предоставить копированием списка ролей, предоставленных некоторому другому пользователю. Для этого в контекстном меню выделенного пользователя нужно выбрать пункт «Скопировать права другого пользователя»,



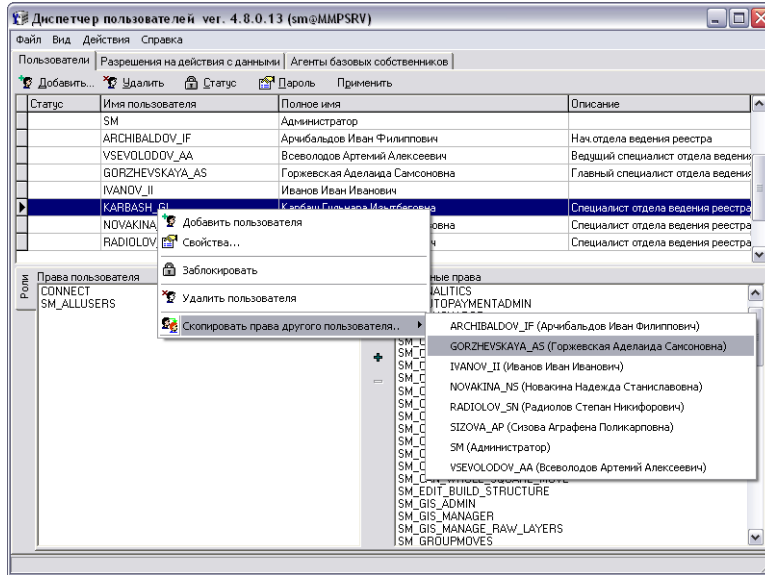


Рисунок 1.9.

что позволяет указать учетную запись другого пользователя, чьи привилегии работы с базой данных копируются выделенному пользователю.

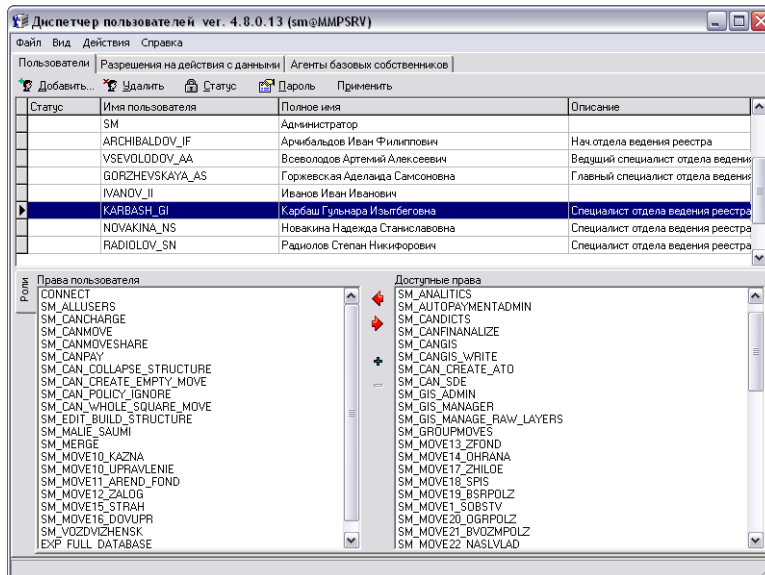



Рисунок 1.10.

## 1.2.2. Создание новой роли

В ряде случаев требуется создать роль, отсутствующую в правом перечне доступных ролей. Чтобы создать новую роль нужно нажать кнопку  – «Создать новую роль». В результате открывается окно ввода, в котором нужно ввести имя новой роли с использованием латинских букв, цифр и символа подчеркивания. Первый символ должен быть буквой. Имя роли должно быть уникальным.

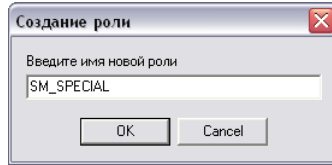



Рисунок 1.11.

По завершении ввода следует нажать кнопку **ОК**. Новая роль появится в перечне доступных ролей и ее можно назначать пользователям, а также использовать для других целей.

## Замечание

Настоятельно рекомендуется для управления полномочиями пользователей SAUMI создавать роли с префиксом `SM_` в их наименованиях.

### 1.2.3. Удаление роли с префиксом `SM_`

Роли с префиксом `SM_` рассматриваются, как роли, управляющие доступом к объектам схемы `SM` базы данных SAUMI. Такие роли могут быть удалены из базы данных с помощью «Диспетчера пользователей». Чтобы удалить роль с префиксом `SM_`, необходимо на вкладке «Пользователи» эту роль выделить в нижней части вкладки на той панели, где она присутствует для текущего пользователя и нажать кнопку  – «Удалить выделенную роль»,

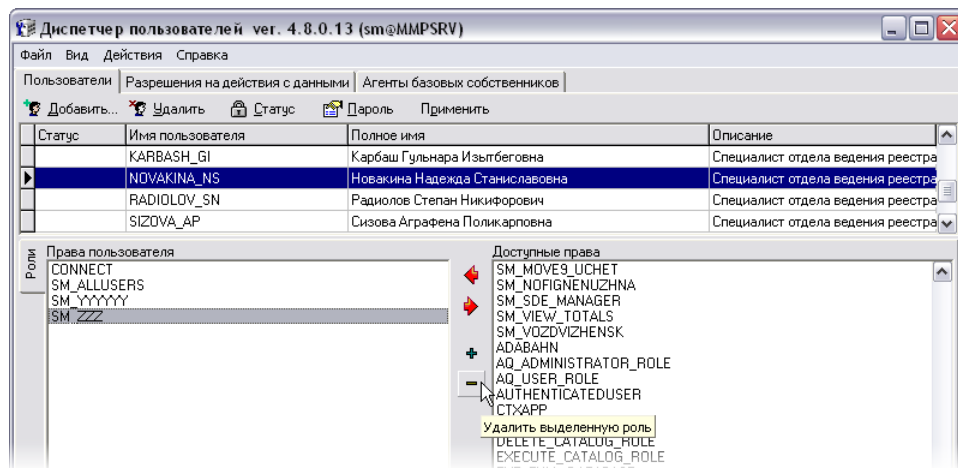


Рисунок 1.12.

которая становится активной после выделения роли с префиксом `SM_` на левой или правой панелях.

После нажатия указанной кнопки выводится предупреждающее окно.

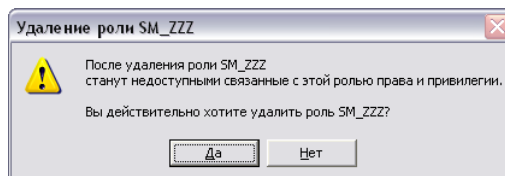


Рисунок 1.13.

Нажатие кнопки «Да» подтверждает удаление и роль удаляется.

### 1.3. Управление доступом к информационным объектам SAUMI

На вкладке «Разрешения на действия с данными» может производиться более тонкая настройка прав доступа ролей и пользователей к некоторым объектам (таблицам) схемы SM, содержащим элементы персональных данных.

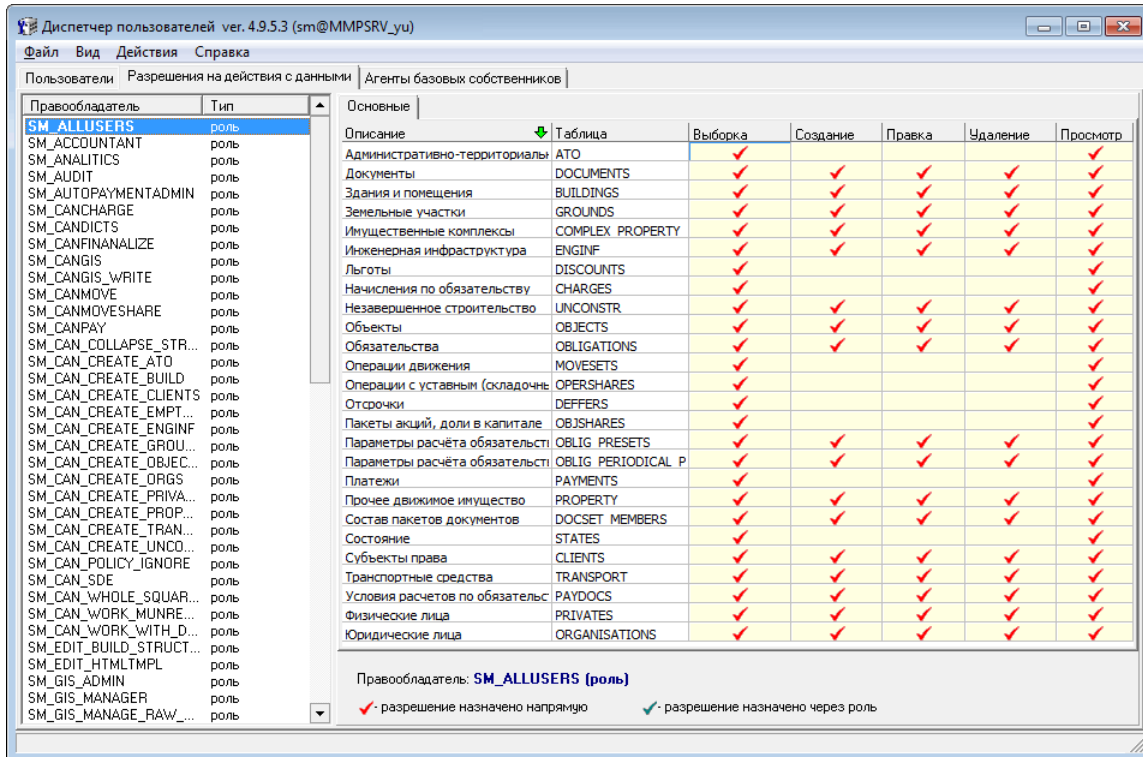


Рисунок 1.14.

Левая панель вкладки содержит перечень правообладателей – ролей с префиксом SM\_ и пользователей. Имена ролей представлены черным шрифтом, причем роль SM\_ALLUSERS выделяется еще и полужирным начертанием, в то время как имена пользователей представлены синим шрифтом, а полужирным начертанием выделен пользователь SM.

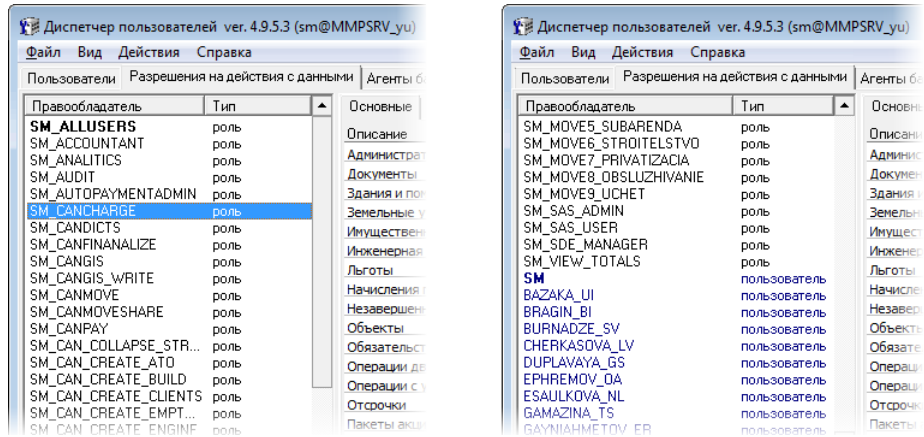


Рисунок 1.15.

Правая панель вкладки «Разрешения на действия с данными» содержит перечень таблиц, зарегистрированных в таблице FM\_TABLES, а также отметки о предоставлении прав доступа к этим таблицам правообладателю, выделенному на левой панели. Отметка о предоставлении какого-либо права доступа представлена значком флажка , который выставляется или снимается двойным щелчком мыши в соответствующей ячейке таблицы.

### Замечание

В перечне таблиц отображаются те таблицы, регистрирующие записи которых в FM\_TABLES имеют значение 'Y' в поле GRANTABLE\_BY\_USERMAN – «Отображать в списке таблиц Менеджера пользователей для управления правами работы с данными этой сущности».

Права доступа «Выборка», «Создание», «Правка», «Удаление» соответствуют традиционным правам реляционных баз данных на выполнение SQL-операторов SELECT, INSERT, UPDATE, DELETE и предоставляются на уровне базы данных.

### Замечание

Следует избегать непредоставления права доступа «Выборка», так как при этом приложение SAUMI не может осуществлять выборку данных из такой таблицы, что зачастую нарушает логику работы программы. Для закрытия видимости данных таблицы следует использовать непредоставление права доступа «Просмотр».

Право доступа «Просмотр» реализуется программно модулями SAUMI и его наличие обеспечивает возможность открывать окно свойств, представляющее информацию какой-либо записи данной таблицы. Исключением здесь следует считать таблицу DOCSET\_MEMBERS (состав пакетов документов), не имеющей собственного окна свойств, тем не менее, составы пакетов документов отображаются на отдельных вкладках «Документы» в окнах свойств различных сущностей SAUMI.

В случае непредоставления права «Просмотр» пользователю SAUMI в его сеансе работы для некоторой таблицы окно свойств, соответствующее данной таблице (кроме DOCSET\_MEMBERS), не открывается и по этой причине права доступа «Создание» и «Правка» по отношению к этой таблице не реализуются. В случае непредоставления права «Просмотр» по отношению к таблице DOCSET\_MEMBERS на вкладках «Документы» окон свойств различных сущностей SAUMI только отображается надпись «Недостаточно прав на просмотр», а контекстное меню пакета документов недоступно, как и копирование перетаскиванием на вкладку.

Предоставление пользователю для некоторой таблицы права «Создание» при наличии прав «Выборка» и «Просмотр» не позволяет создавать новые записи в этой таблице посредством пользовательского интерфейса SAUMI.

Таблица 1.1. Эффективные сочетания прав доступа

Выборка	Создание	Правка	Удаление	Просмотр
✓	✓	✓	✓	✓
✓	✓	✓		✓
✓		✓	✓	✓
✓			✓	✓
✓				✓
✓			✓	

Предоставление права создания новых записей (операция INSERT) в какой-либо из представленных в перечне таблиц автоматически предоставляет право выборки (операция SELECT) в соответствующем генераторе последовательностей, посредством которого генерируется регистрационный код (ID) новой записи этой таблицы.

### 1.3.1. Предоставление полномочий доступа

Изначально наибольший объем полномочий предоставляет роль SM\_ALLUSERS, в частности разрешая выборку и просмотр во всех таблицах, предназначенных для управления доступом к их информационным элементам (рис. 1.14).

Так как роль SM\_ALLUSERS назначается всем пользователям SAUMI, то этот же объем полномочий изначально предоставляется всем пользователям. В случае, когда пользователю SAUMI предоставлен минимальный объем полномочий, а именно, – роли CONNECT и SM\_ALLUSERS – в этом случае таблица прав такого пользователя повторяет таблицу прав роли SM\_ALLUSERS.

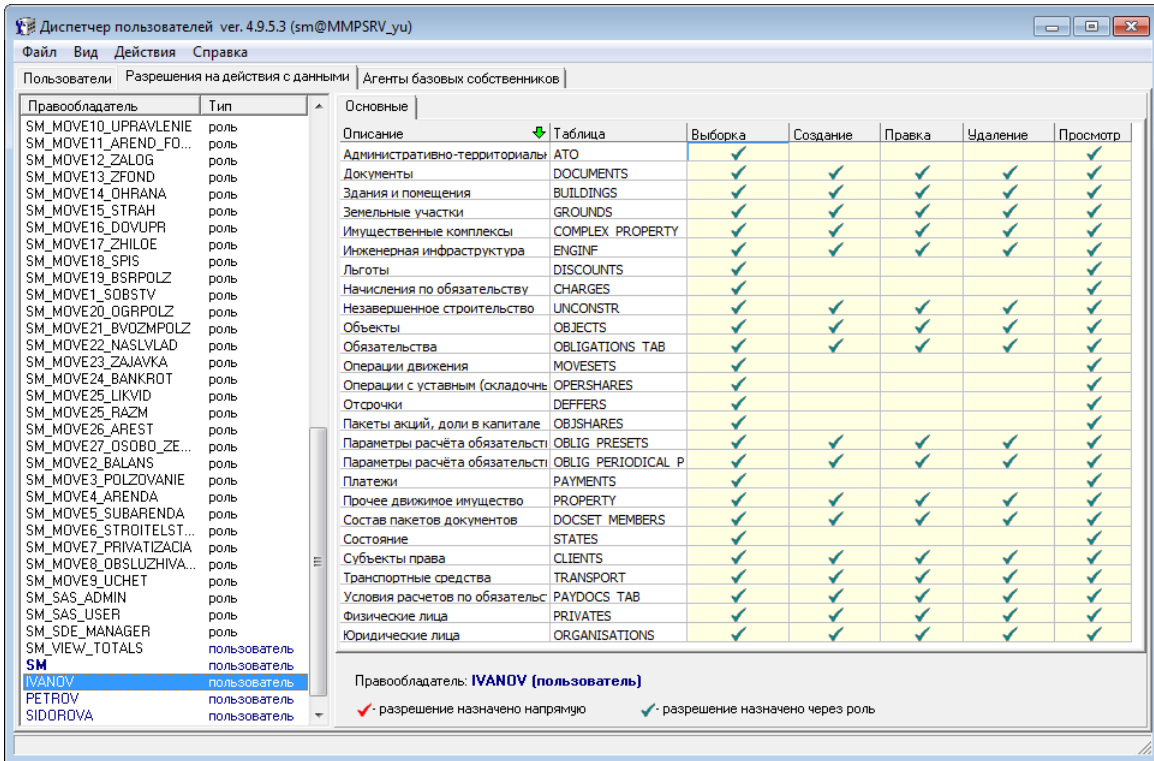


Рисунок 1.16.

Однако, изначальные полномочия роли SM\_ALLUSERS не являются исчерпывающими и повсеместными. Как видно на рис. 1.14 в таблице предоставленных прав имеются пробелы и, в частности в строке, соответствующей таблице платежей PAYMENTS, отсутствуют права на создание, правку и удаление записей.

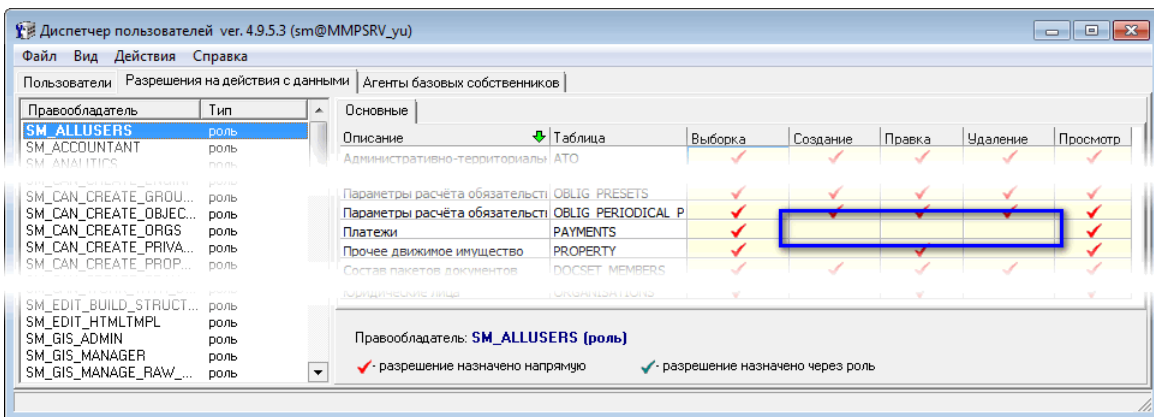


Рисунок 1.17.

Тем самым, при минимальном объеме полномочий пользователь SAUMI не может вводить, редактировать и удалять платежи, хотя окно свойств платежа такому пользователю доступно для просмотра.

С другой стороны недостающие привилегии на ввод, редактирование и удаление платежей предоставляет роль SM\_CANPAY.

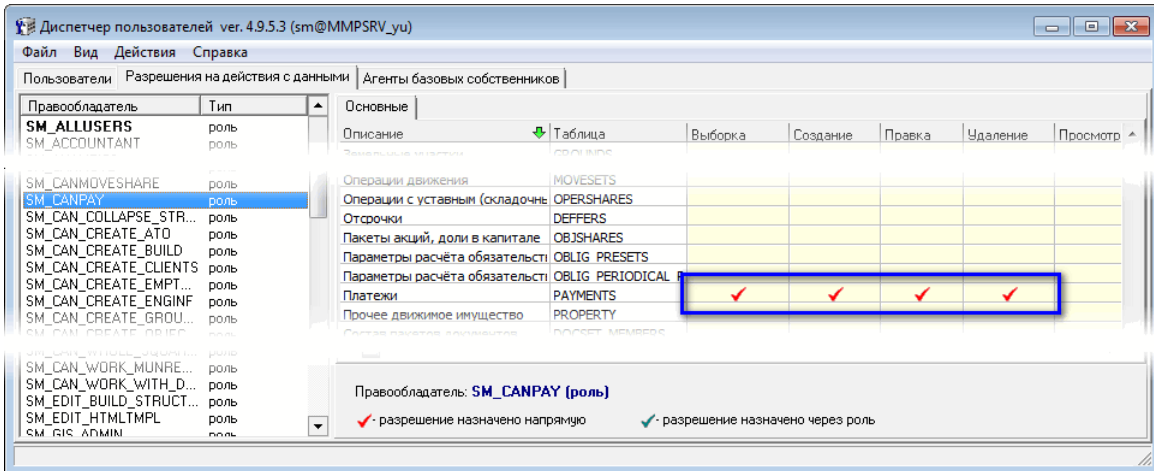


Рисунок 1.18.

Таким образом, назначение пользователю SAUMI роли SM\_CANPAY предоставляет ему возможности полного контроля над информацией окна платежа.

Чтобы сделать окна свойств платежей недоступными обычным пользователям следует для роли SM\_ALLUSERS снять флажок разрешения права «Просмотр» для таблицы PAYMENTS.

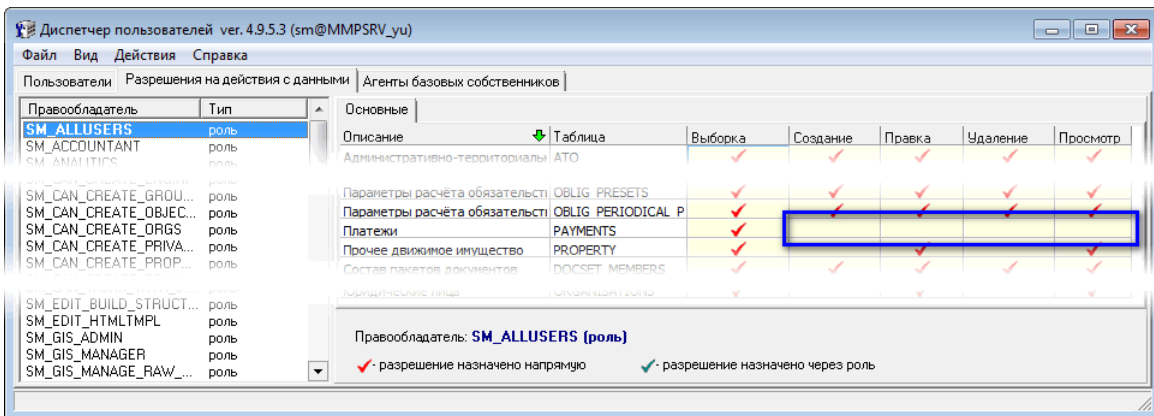


Рисунок 1.19.

Для выделения некоторого круга пользователей, имеющих полный набор полномочий доступа к данным окна свойств платежей можно с одной стороны предоставлять требуемые права пользователям индивидуально напрямую,

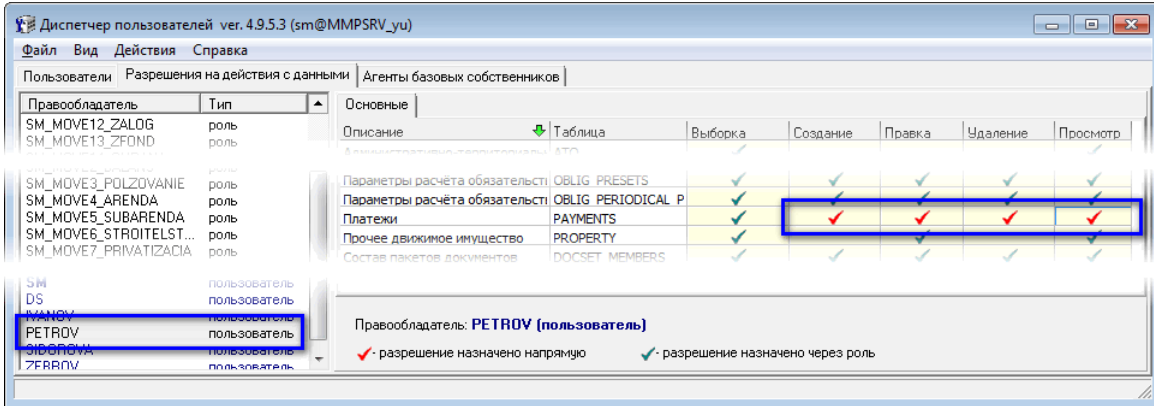


Рисунок 1.20.

а с другой стороны эти права могут быть все предоставлены роли SM\_CANPAY,

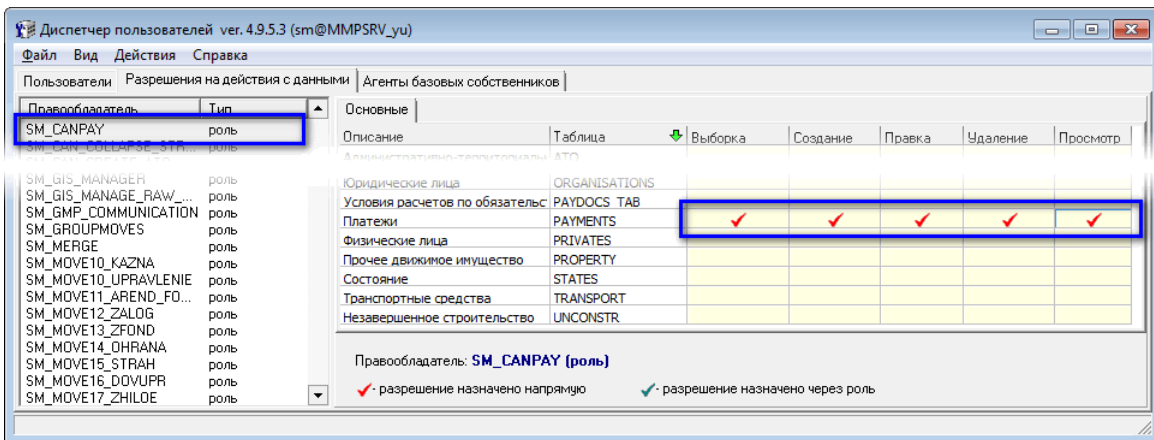


Рисунок 1.21.

а пользователям, в чьи обязанности входит работа с платежами, может быть назначена роль SM\_CANPAY.

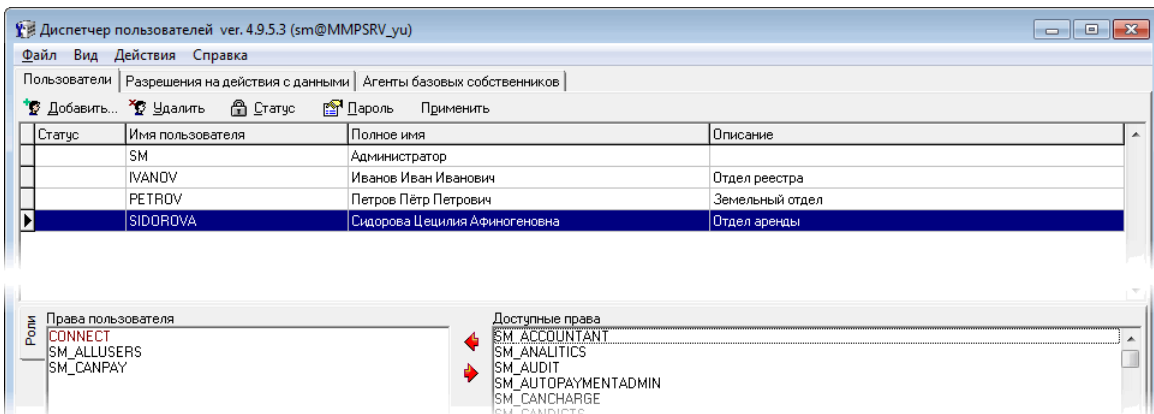


Рисунок 1.22.



Общая схема разграничения прав доступа к некоторой таблице, зарегистрированной в FM\_TABLES, проиллюстрированная выше примером, состоит в следующем.

1. В роли SM\_ALLUSERS для этой таблицы следует убрать те права, которые будут предоставляться особым способом: либо напрямую нужным пользователям, либо посредством дополнительной роли (ролей).
2. В случае решения предоставлять пользователям требуемые для их работы в SAUMI права доступа посредством дополнительной роли или нескольких ролей (например, роль, позволяющая создавать и редактировать записи таблицы и роль, позволяющая просматривать и удалять записи этой же таблицы) следует:
  - создать новую или выбрать среди имеющихся такую роль (роли);
  - предоставить созданной или выбранной роли (ролям) требуемые права доступа;
  - предоставить нужному кругу пользователей роль (роли), наделенную необходимыми в работе правами доступа.

### 1.3.2. Разграничение прав доступа к финансовым обязательствам по их типам

Может быть установлено разграничение прав доступа различных пользователей к финансовым обязательствам в зависимости от их типов. Тип обязательства определяется в окне «Условия по обязательству» значением реквизита «Назначение платежа», который принимает значения из справочника «Типы обязательств» (таблица SM.OBLIGATIONTYPE).

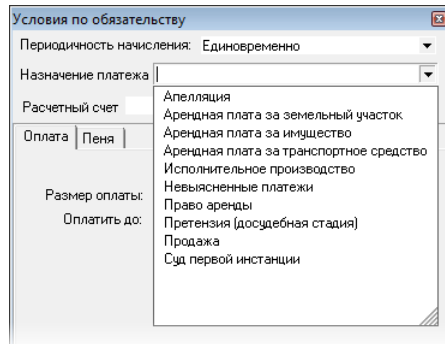


Рисунок 1.23.

Правообладателю (пользователю или роли) могут быть предоставлены права на выполнение действий, обозначаемые как «Создание/Изменение», «Удаление» и «Просмотр».

Право на действие «Просмотр» для обязательства определенного типа включает в себя возможность просмотра условий (параметров) обязательства такого типа, просмотра свода расчетов по такому обязательству.

Право на действие «Удаление» позволяет пользователю удалить в окне операции движения на вкладке «Обязательства» обязательство того типа, на которое данное право предоставлено.

Право на действие «Создание/Изменение» позволяет в окне операции движения на вкладке «Обязательства» добавить обязательство того типа, на которое предоставлено указанное право, редактировать его свойства (параметры).

Для назначения прав доступа к обязательствам определенного типа необходимо перейти на вкладку «Разрешения на действия с данными» и выделить одну из ячеек столбцов с желтым фоном в строке «Обязательства – OBLIGATIONS\_TAB».

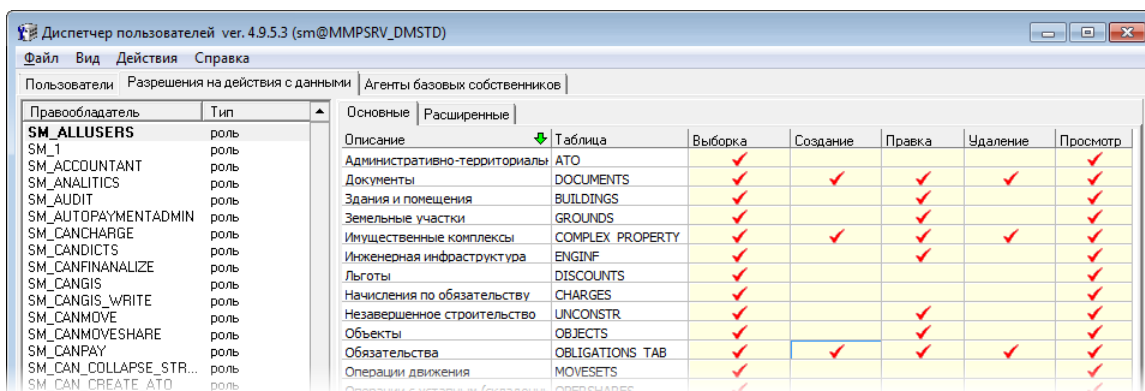


Рисунок 1.24.

В результате отображается дополнительная вкладка «Расширенные», которую следует выбрать. На указанной вкладке отображается таблица, строки которой соответствуют типам обязательств, а среди столбцов присутствуют столбцы «Создание/Изменение», «Удаление» и «Просмотр», соответствующие действиям, права на выполнение которых предоставляются правообладателю. Предоставленное право указывается в ячейке таблицы выставленным флажком ✓. Выставление флажка, как и его снятие, осуществляется двойным щелчком мыши в ячейке таблицы. Представленная таблица отображает полномочия по разным типам обязательств одного правообладателя (роли или пользователя), выделенного на левой панели, а также отмеченного наименованием в нижней части правой панели окна утилиты «Диспетчер пользователей».

## Замечание

Роль SM\_ALLUSERS, предоставляемая всем пользователям, сразу после установки SAUMI не предоставляет никаких полномочий по работе с финансовыми обязательствами любых типов. Чтобы вновь создаваемый пользователь получал некоторый минимальный объем доступа к обязательствам каких-либо типов следует соответствующие полномочия предоставить указанной роли в соответствии с выше приведенными положениями.

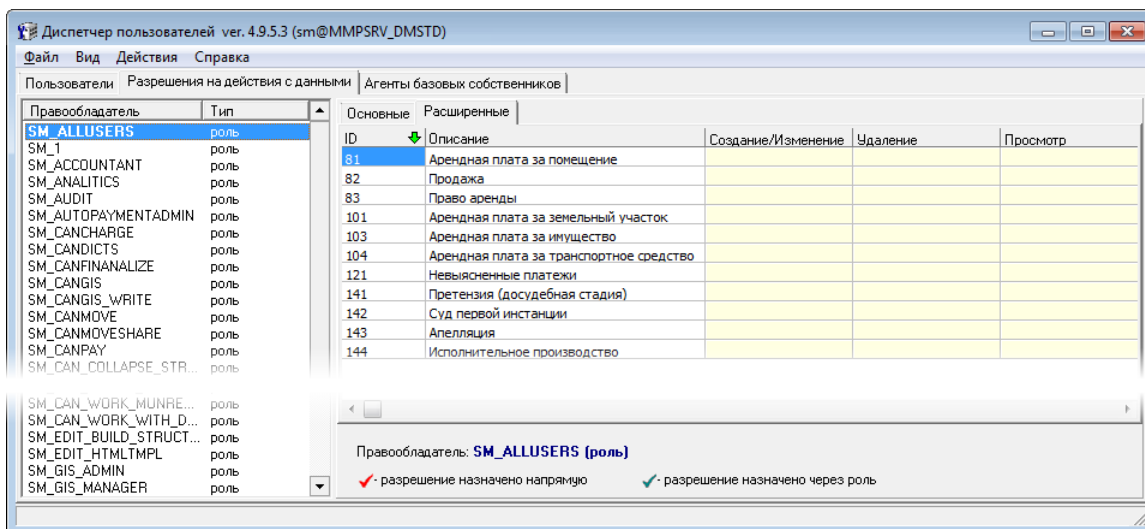


Рисунок 1.25.

Очевидной потребностью разграничить полномочия различных пользователей в доступе к обязательствам различных типов является обособление работы с техническими обязательствами, сопровождающими претензионно-исковую работу (см. главу «Поддержка претензионно-исковой работы в SAUMI» руководства пользователя). В указанном случае пользователю напрямую могут быть предоставлены полномочия выполнения действий по типам обязательств в объеме, как на ниже следующем рисунке.

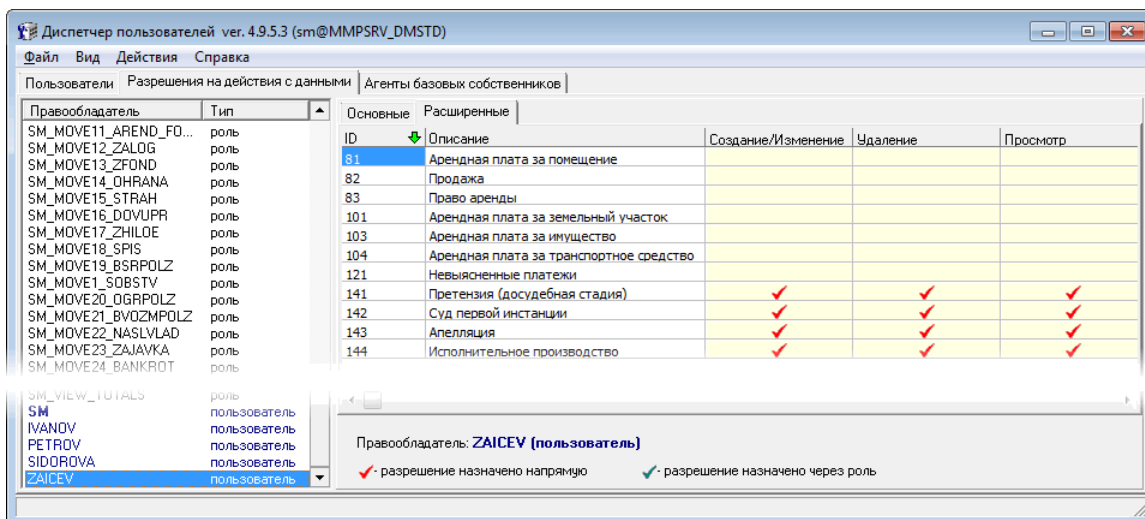


Рисунок 1.26.

## Замечание

Выше описанная схема разграничения прав эффективно действует при выгрузке начислений в ГИС ГМП, т.е. пользователь может произвести выгрузку начислений (см. раздел «Выгрузка начислений в ГИС ГМП» главы «Применение менеджера финансовых поступлений» руководства пользователя) только по обязательствам тех типов, для которых ему предоставлено право выполнять действия, обозначаемые как «Создание/Изменение».

Также при вызове отчетов «Начисления для ГИС ГМП», «Сводный по начислениям ГИС ГМП» и «Сводный по начислениям ГИС ГМП (для тестирования)» из группы отчетов «Взаимодействие с ГИС ГМП» (см. подраздел «Подготавливаемые для ГИС ГМП сведения и выявляемая при выгрузке неполнота данных» главы «Применение менеджера финансовых поступлений» руководства пользователя) отображаются сведения, соответствующие тем обязательствам, на типы которых пользователю предоставлено право выполнять действия «Создание/Изменение».

## 1.4. Регистрация базовых собственников

Программный комплекс SAUMI, развернутый из комплекта поставки, обычно уже содержит зарегистрированных базовых собственников, учет правоотношений по поводу имущества которых занимается Комитет – приобретатель и эксплуатант SAUMI. Также в этом случае уже созданы и надлежащим образом назначены роли агентов базовых собственников, а вся настройка базовых собственников сводится в настройке полномочий пользователей SAUMI выступать агентами распорядителя того или иного базового собственника (см. 1.4.2). Более того, если базовый собственник один, то все пользователи данной установки SAUMI рассматриваются как агенты распорядителя этого единственного базового собственника и никакие действия, описываемые в данном разделе, выполнять не требуется.

Однако возможна ситуация, когда по истечении времени Комитет, используя SAUMI, принимает к учету имущественные правоотношения нового базового собственника. В таком случае необходимо нового базового собственника зарегистрировать и назначить пользователям SAUMI полномочия агентов распорядителя этого базового собственника.

Указанные действия выполняются на вкладке **Агенты базовых собственников** утилиты «Диспетчер пользователей».

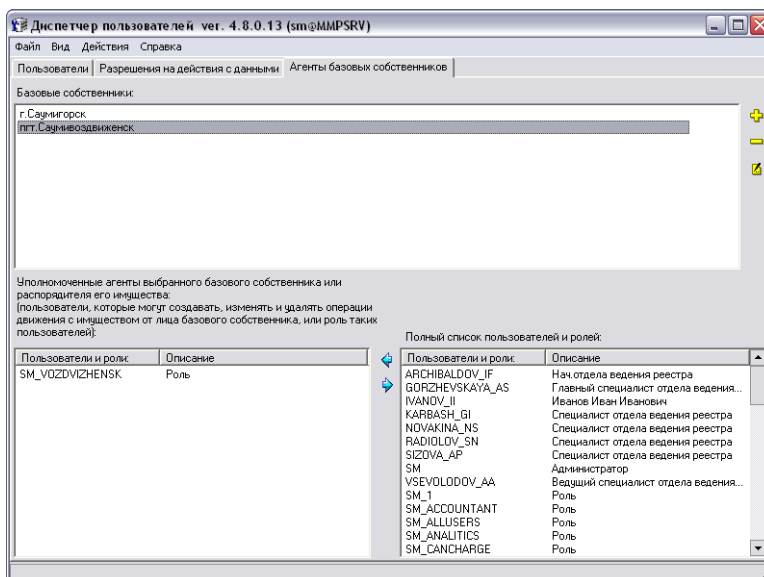



Рисунок 1.27.

### 1.4.1. Регистрация нового базового собственника

Предваряя регистрацию нового базового собственника, его нужно создать как субъект права в БД SAUMI (см. главу «Регистрация объектов и субъектов права в SAUMI» в руководстве пользователя).

Чтобы зарегистрировать субъекта права как базового собственника, необходимо на вкладке **Агенты базовых собственников** нажать кнопку .

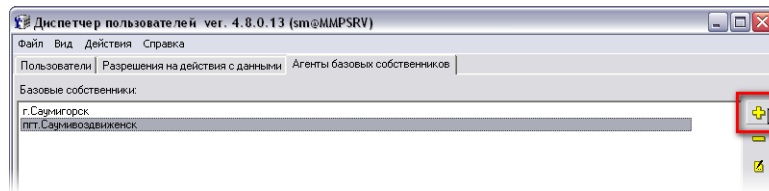


Рисунок 1.28.

В результате откроется окно, в котором новый базовый собственник выбирается из списка всех зарегистрированных субъектов права. После осуществления выбора нужно нажать кнопку **Ок**.

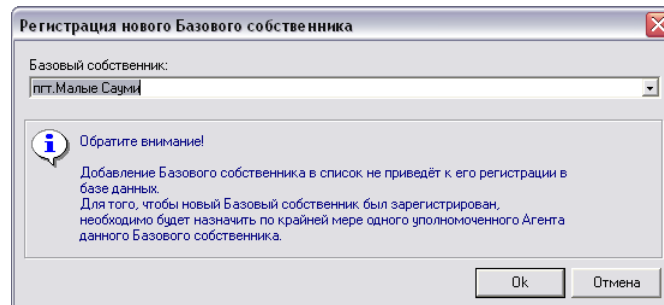


Рисунок 1.29.

В результате выбранный субъект права отобразится в перечне базовых собственников.

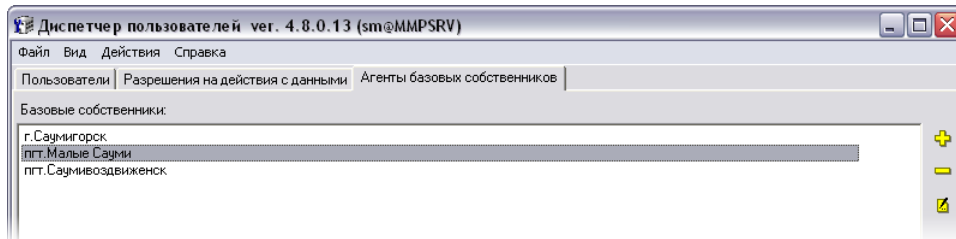


Рисунок 1.30.

Однако, без назначения базовому собственнику уполномоченных агентов, которые будут выполнять учет правоотношений по поводу его имущества, новый базовый собственник не будет сохранен в базе данных. Речь идет о предоставлении соответствующих полномочий пользователям SAUMI.


## 1.4.2. Предоставление полномочий агентов распорядителя базового собственника

В SAUMI реализовано три способа предоставления пользователям полномочий агентов распорядителя базового собственника:

- прямая ассоциация пользователя с базовым собственником;
- косвенная ассоциация пользователей с базовым собственником через предоставленную роль уполномоченного агента;
- комбинация первых двух методов.

### 1.4.2.1. Прямая ассоциация пользователей с базовым собственником

Прямая ассоциация пользователей с базовым собственником осуществляется созданием для базового собственника списка пользователей, уполномоченных выступать агентами его распорядителя.

В нижней части вкладки **Агенты базовых собственников** справа содержится объединенный список регистрационных имен пользователей и ролей. Чтобы установить ассоциацию некоторого пользователя SAUMI с базовым собственником, необходимо выделить этого собственника в перечне базовых собственников, затем выделить в находящемся в нижней правой части окна объединенном списке регистрационное имя нужного пользователя и нажать кнопку .

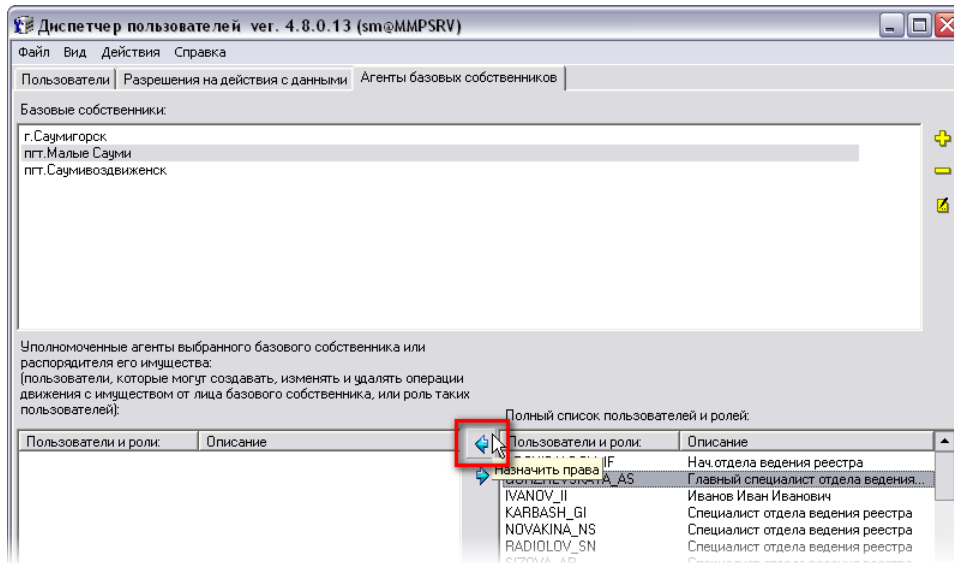


Рисунок 1.31.

В результате отмеченное имя пользователя отобразится слева в списке ассоциированных с базовым собственником пользователей и ролей.

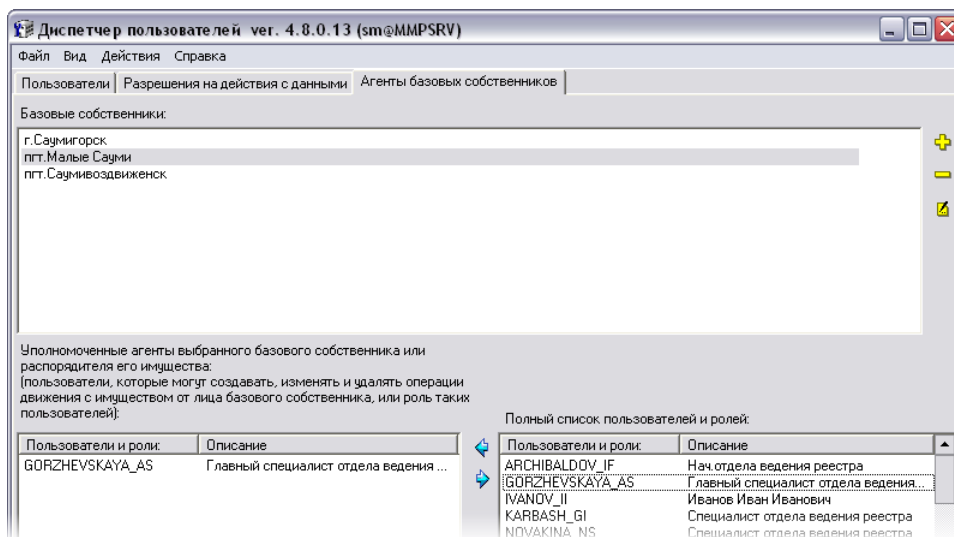


Рисунок 1.32.

Последовательное применение указанных выше действий к учетным именам пользователей, которым требуется предоставить полномочия агентов распорядителя данного базового собственника, позволяет получить полный список ассоциированных с базовым собственником пользователей SAUMI.

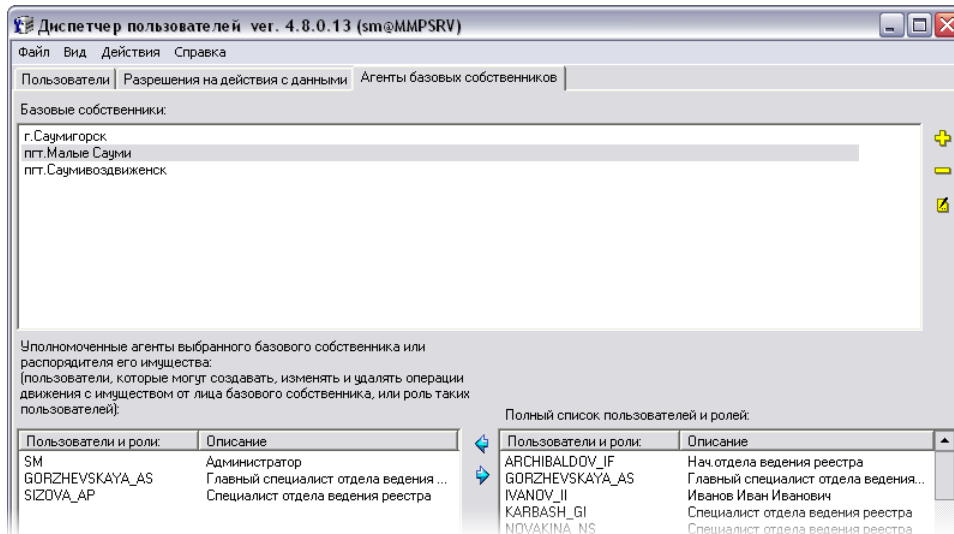



Рисунок 1.33.

## Замечание

Крайне желательно любым из описываемых способов предоставить учетной записи SM (администратор SAUMI) полномочия агента распорядителя всех зарегистрированных в БД базовых собственников.

Кнопка  позволяет удалить отмеченное в списке пользователей и ролей, ассоциированных с базовым собственником, учетное имя пользователя.

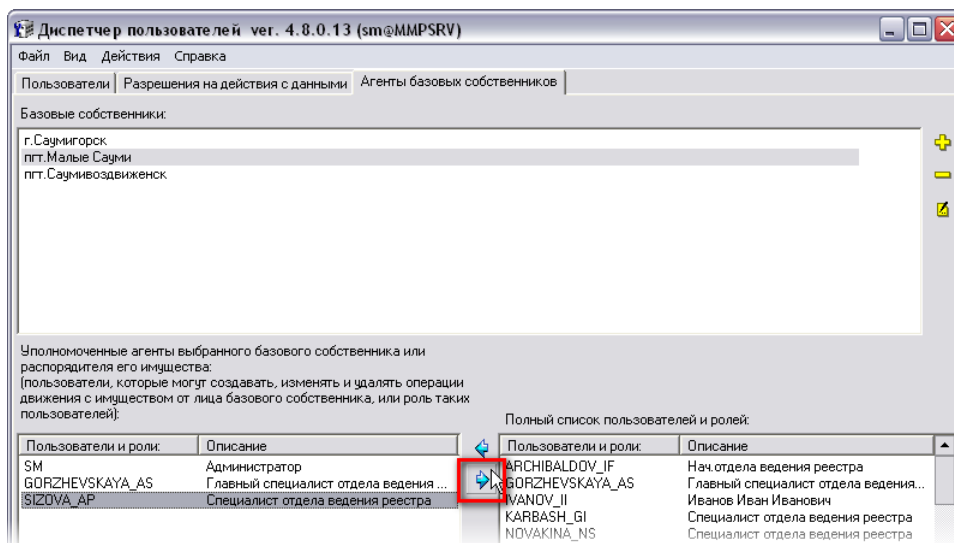


Рисунок 1.34.

### 1.4.2.2. Использование роли уполномоченного агента

Другим способом предоставить пользователю полномочия агента распорядителя базового собственника является назначение учетной записи пользователя соответствующей роли (см. 1.2.1), определяющей такие полномочия.

Роль агента распорядителя базового собственника создается в порядке, описанном в Раздел 1.2.2, «Создание новой роли», на вкладке **Пользователи**.

#### Замечание

Рекомендуется использовать такое построение имени роли: префикс `SM_`, за которым следует набранная латиницей в верхнем регистре строка, созвучная имени субъекта права, являющегося базовым собственником. Например, для административно-территориального образования пгт. Малые Сауми соответствующая роль может быть названа `SM_MALIE_SAUMI`.

Созданная роль по аналогии с учетными записями пользователей ассоциируется (см. выше) с базовым собственником.

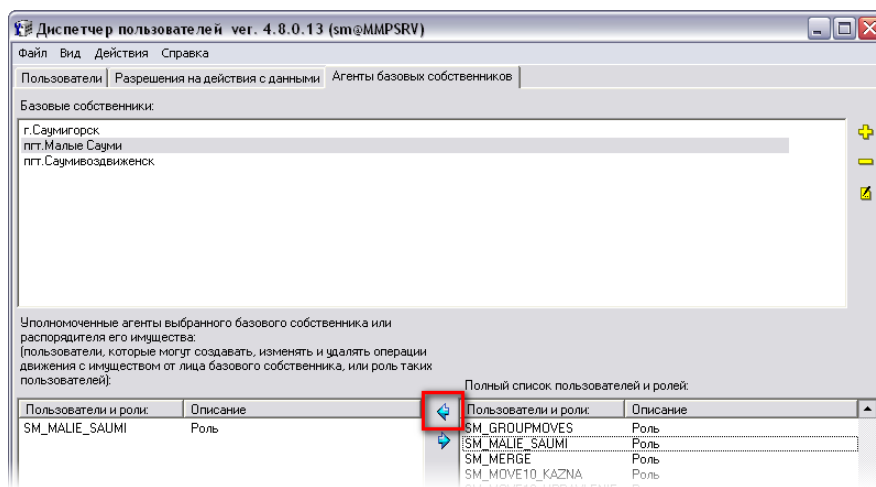


Рисунок 1.35.

После этого предоставление пользователю данной ассоциированной с базовым собственником роли (см. 1.2.1) наделяет пользователя полномочиями агента распорядителя данного собственника.



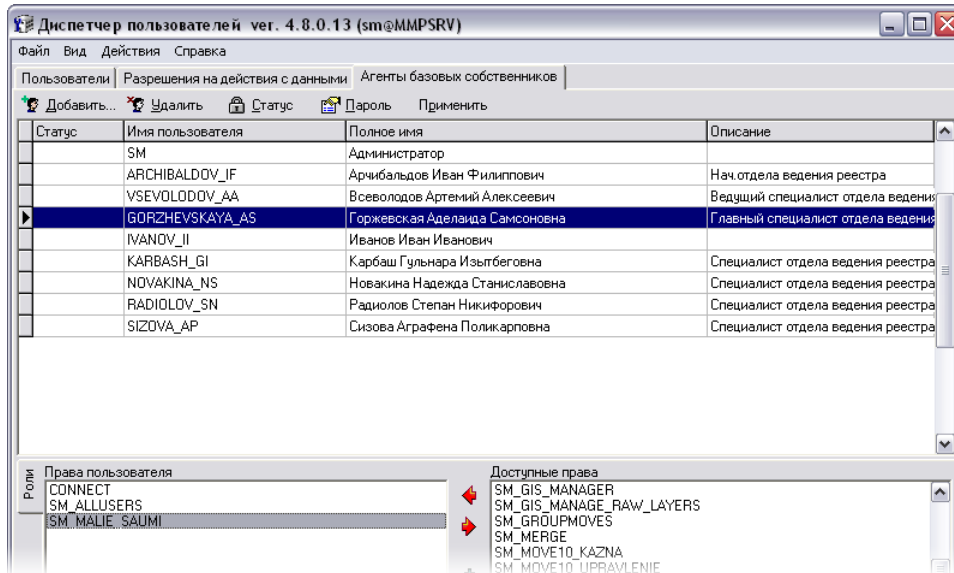


Рисунок 1.36.

Предоставление пользователю нескольких ролей, ассоциированных с разными базовыми собственниками, наделяет пользователя полномочиями агента распорядителя всех этих базовых собственников.

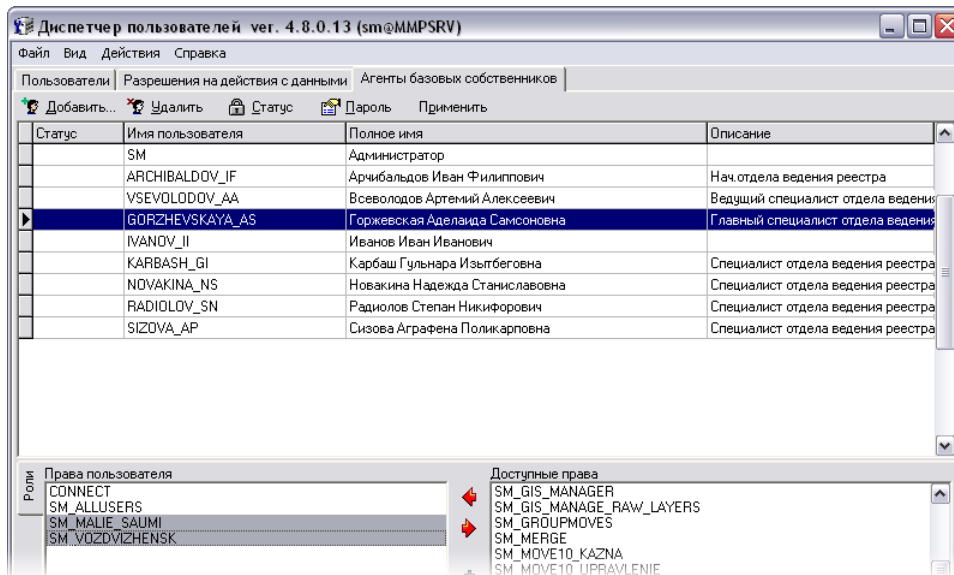



Рисунок 1.37.

## Замечание

При изначальном использовании SAUMI Комитетом в интересах одного базового собственника с этим единственным базовым собственником ассоциируется роль SM\_ALLUSERS. При появлении дополнительных базовых собственников и проведении Комитетом политики строгого разграничения полномочий, когда конкретный пользователь наделяется полномочиями агента распорядителя только одного базового собственника, в этом случае

следует для первого «основного» базового собственника ассоциацию с ролью `SM_ALLUSERS` разорвать, используя кнопку  (рис. 1.35). Затем, следуя выше изложенной схеме, создать новую роль и ассоциировать ее с этим исходным базовым собственником. Полномочия агентов распорядителя исходного базового собственника необходимо предоставить либо прямой ассоциацией пользователей с этим базовым собственником, либо предоставлением им вновь созданной ассоциированной с собственником роли.

---

# Глава 2. «Менеджер обновлений»

## Содержание

2.1. Порядок установки обновления .....	37
2.2. Компиляция объектов с ошибками .....	39
2.3. Переиндексация поиска .....	39
2.4. Обновление отчетов, печатных форм, форм сбора отчетных данных и модуля SAUMI-Pascal .....	39
2.4.1. Обновление отчетов (настроек поиска) .....	40
2.4.2. Обновление печатных форм .....	40
2.4.3. Обновление форм сбора отчетных данных в электронном виде .....	40
2.4.4. Обновление процедур SAUMI-Pascal .....	41

Комплекс SAUMI постоянно развивается, расширяется набор возможностей, часть функциональных возможностей видоизменяется с изменением законодательства. Изменение комплекса может отражаться в изменении файлов, входящих в комплекс и изменении структуры базы данных.

Обновление комплекса производится путем применения пакетов обновления поставляемых в виде `pth`-файлов. Эти файлы содержат информацию о разнице в файлах предыдущей и следующей версии и сценарий обновления структуры базы данных. Как правило, в названии файла указывается номер версии SAUMI, которая подлежит обновлению и, через дефис, номер версии, который будет иметь SAUMI после обновления (номера версий с нулями вместо точек). Например, файл `40903-40904.pth` – это пакет обновления SAUMI с версии 4.9.3 до версии 4.9.4.

Пакеты обновления публикуются на веб-сайте <http://support.saumi.ru> в разделе «Обновления». Скаченные с указанного сайта пакеты обновления устанавливаются с помощью утилиты «Менеджер обновлений», входящей в комплект администратора SAUMI.

## 2.1. Порядок установки обновления

- 1. ОБЯЗАТЕЛЬНО СДЕЛАЙТЕ РЕЗЕРВНУЮ КОПИЮ БАЗЫ ДАННЫХ!** (См. главу «Резервное копирование и восстановление базы данных SAUMI» руководства «Быстрый старт».)
2. Загрузите с сайта пакет обновления.

### Замечание

Рекомендуется создать отдельную папку, куда и поместить файл обновления. Тогда файлы протокола обновления БД будут помещены в эту же папку, что сильно сократит их размер и облегчит чтение, если будут какие-либо сбои. При повторной попытке обновления рекомендуется удалить все файлы с расширением `.log` из этой папки – тогда программа обновления создаст их заново, записав в них протоколы только последней попытки обновления. Файл `db_update.log` содержит протокол обновления структуры БД. Файл `update.log` содержит протокол обновления исполняемых файлов комплекса SAUMI.

3. Запустите утилиту «Менеджер обновлений» из комплекта администратора SAUMI.

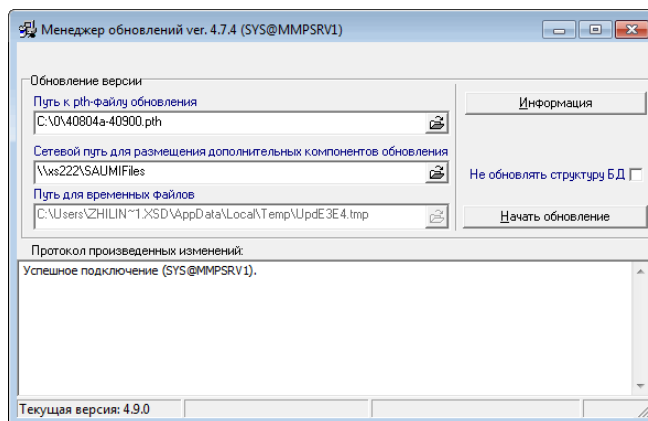


Рисунок 2.1.

4. В поле *Путь к pth-файлу обновления* укажите локальный файл загруженного пакета обновления.

В поле *Сетевой путь для размещения дополнительных компонентов обновления* требуется указать имя реально существующего сетевого ресурса в формате UNC `\\server\netresource\path`, доступного выполняемому сеансу «Менеджера обновлений» для записи. Данный сетевой ресурс используется для копирования на него установочных пакетов дополнительных программных компонент с последующим развертыванием этих компонент на клиентских компьютерах SAUMI, которые должны иметь доступ на чтение этого сетевого ресурса.

## Замечание

Флажок *Не обновлять структуру БД* можно не выставлять. Если данный флажок выставлен, то тогда в БД будут скопированы бинарные файлы и сценарий обновления БД до новой версии, но сам сценарий выполняться не будет. В этом случае, чтобы привести структуру БД к новой версии придётся запустить сценарий целиком или его части вручную в окне «Выполнение PL/SQL-сценария» утилиты «Менеджер реквизитов» из комплекта администратора SAUMI (см. 8.4).

5. Запустите процесс обновления, нажав кнопку **Начать обновление**.
6. Если в процессе выполнения сценария обновления структуры базы данных возникли ошибки, внимательно изучите причины ошибок; если они Вам непонятны – скопируйте фрагмент сценария, вызвавшего ошибки, и само сообщение об ошибках и отошлите Группе сопровождения SAUMI, при этом выполнение сценария следует остановить. Если ошибки Вам понятны, можете самостоятельно принять решение о том как продолжать сценарий. Вы можете прямым способом – вводом SQL команд в окне «Выполнение PL/SQL-сценария» утилиты «Менеджер реквизитов» из комплекта администратора SAUMI (см. 8.4) – внести изменения взамен тех, что должны были выполняться, и игнорировать возникшие ошибки.

## Предупреждение

Если при обновлении возникли ошибки, которые вы самостоятельно не можете разрешить, **НЕМЕДЛЕННО** восстанавливайте содержимое БД из её резервной копии. Не пытайтесь выполнять обновление повторно, не восстановив БД предыдущей версии!

7. Если процесс обновления прошел неудачно – был прерван сценарий или файлы не соответствовали предыдущей версии, необходимо выполнить восстановление базы данных.

8. В случае несоответствия версий файлов, например, их непреднамеренного изменения (вирусами), обратитесь в Группу сопровождения SAUMI для подготовки специального пакета обновления, при котором будет произведена полная замена файлов.
9. Установите отчеты, печатные формы, формы сбора отчетных данных и т.д., если таковые для данного пакета обновления имеются, о чем сообщается в анонсе обновления на сайте <http://support.saumi.ru> в разделе «Обновления». Порядок действий приводится ниже в разделе 2.4.

## 2.2. Компиляция объектов с ошибками

После обновления структуры БД возможно появление объектов с ошибками: в основном это процедуры, функции и представления. Для восстановления нормальной работы БД необходима перекомпиляция данных объектов.

Чтобы выполнить перекомпиляцию объектов с ошибками достаточно запустить утилиту «Менеджер реквизитов» из комплекта администратора SAUMI. Во время запуска эта утилита автоматически проверяет наличие в базе данных объектов с ошибками и производит их перекомпиляцию. В дополнение «Менеджер реквизитов» на панели инструментов содержит кнопку **Компиляция объектов**.

## 2.3. Переиндексация поиска

Поскольку при обновлении могут быть добавлены новые критерии поиска, необходимо произвести его переиндексацию, для того, чтобы изменения вступили в силу. Для этого запустите исполняемый модуль SAUMI, подключившись под пользователем SM. В Главной панели SAUMI выберите пункт «Поиск | Полный поиск». В открывшемся окне нажмите кнопку **Индексировать**. Дождитесь появления сообщения «Критерии поиска проиндексированы».

### Замечание

Переиндексацию поиска необходимо также производить в случае внесения администратором SAUMI каких-либо изменений в «Менеджере отчетов» и «Менеджере реквизитов».

## 2.4. Обновление отчетов, печатных форм, форм сбора отчетных данных и модуля SAUMI-Pascal

Прежде всего, необходимо извлечь файлы из появившихся в папке установки SAUMI (по умолчанию C:\SAUMI) архивов ReportsXXX-XXX.rar (ReportsXXX-XXX.zip) и SPascalXXX-XXX.rar (SPascalXXX-XXX.zip) в папку на жёстком диске. Каждый из извлеченных файлов, соответствует той или иной утилите комплекта администратора SAUMI и располагается в соответствующей папке (папка «Отчеты» – «Менеджер отчетов», папка «Печатные формы» – «Менеджер форм», папка «Формы сбора отчетности» – «Регламент отчетности»). Кроме того, в этих папках могут находиться текстовые файлы с описанием порядка установки, если текстовых файлов нет, то должен использоваться обычный порядок установки.

Обновление осуществляется в следующей последовательности:

- обновление отчетов (настроек поиска);
- обновление печатных форм;
- обновление форм сбора отчетных данных в электронном виде;

- обновление процедур и функций SAUMI-Pascal.

### **2.4.1. Обновление отчетов (настроек поиска)**

1. Запустить "Менеджер отчетов" и подключиться к БД.
2. Экспортировать отчеты, одноименные тем, которые есть в обновлении в файлы. При сохранении рекомендуется дописать к имени файла номер старой версии SAUMI, с которой производилось обновление.
3. Далее.

Если администратор изменял соответствующий отчет (добавлял столбцы или редактировал текст запросов) - необходимо импортировать новый отчет с другим номером (номер может быть любым, кроме тех, которые уже используются «Менеджером отчетов»). Сравнивая, привести старый отчет в соответствие с новым. После чего новый отчет (импортированный с новым номером) удалить.

Если администратор не изменял ни один из обновляемых отчетов, то достаточно нажать кнопку «Импорт», выбрать все файлы отчетов и импортировать их в SAUMI (заменяя существующие отчеты).

4. Запустить SAUMI, подключившись пользователем SM, открыть окно поиска и нажать кнопку «Индексировать».

### **2.4.2. Обновление печатных форм**

1. Запустить «Менеджер форм» и подключиться к БД под учетной записью SM.
2. Сохранить старые печатные формы (то есть те, которые должны быть обновлены). При сохранении рекомендуется дописать к имени файла номер старой версии SAUMI, с которой производилось обновление.
3. Удалить старые печатные формы в соответствующих разделах дерева форм.
4. Загрузить обновленные и новые печатные формы из файлов, содержащихся в архиве обновления форм, в соответствующие формам ветки дерева форм. Номер новой версии SAUMI в названиях загруженных форм следует удалить.

Все операции с формами осуществляются с помощью утилиты «Менеджер форм» из комплекта администратора.

### **2.4.3. Обновление форм сбора отчетных данных в электронном виде**

1. Запустить «Регламент отчетности» и подключится к БД.
2. Сохранить старые формы сбора отчетности (то есть те, которые должны быть обновлены). При сохранении рекомендуется дописать к имени файла номер старой версии (с которой производилось обновление).
3. В списке слева, нажимая «Загрузить форму из файла», импортировать новые формы сбора отчетности.

Новые формы можно будет использовать только в новом периоде. Для использования их в активном периоде пришлось бы проводить всю процедуру сбора отчетности заново, начиная с формирования дискета.

#### 2.4.4. Обновление процедур SAUMI-Pascal

1. Запустить редактор SAUMI-Pascal и подключиться к БД.
2. Сохранить старый модуль `sp_global.pas` (Меню «Файл | Сохранить в файл»). При сохранении рекомендуется дописать к имени файла номер старой версии (с которой производилось обновление).
3. Далее.

Если администратор ничего не изменял - загрузить (Меню «Файл | Открыть из файла» новое содержимое модуля `sp_global.pas` и сохранить в БД (Меню «Файл | Сохранить в базу данных»).

Если администратор вносил изменения - необходимо открыть новый модуль `sp_global.pas` в любом текстовом редакторе и привести старый модуль в соответствие с новым.

---

# Глава 3. Управление репозитарием файлов SAUMI

## Содержание

3.1. Механизм локального обновления SAUMI .....	43
3.1.1. Управление локальным обновлением .....	44
3.2. Вызов утилиты управления репозитарием .....	45
3.3. Копирование файла из репозитария на пользовательский компьютер .....	46
3.4. Добавление нового файла в репозитарий .....	46
3.5. Задание настроек выгрузки файла репозитария .....	48
3.6. Изменение имеющегося в репозитарии файла .....	50
3.7. Изменение настроек выгрузки хранимого в репозитарии файла .....	51
3.8. Добавление группы файлов в репозитарий .....	52
3.9. Удаление файла из репозитария .....	56

## 3.1. Механизм локального обновления SAUMI

Как и любой другой программный комплекс SAUMI после установки на пользовательском компьютере представляет собой набор каталогов и файлов, собранных в одном месте – в каталоге (папке) установки комплекса. По умолчанию – это "C:\SAUMI".

С течением времени состав комплекса претерпевает изменения: выпускаются обновления до новых версий, исправляются выявленные дефекты, реализуется индивидуальная функциональность по заказу. Как результат, в организации – эксплуатанте SAUMI – состав файлов и каталогов комплекса периодически должен изменяться, причем на всех клиентских компьютерах.

Весь набор файлов и подкаталогов, необходимых для функционирования комплекса SAUMI, хранится в БД SAUMI в специальных таблицах: VER\_FILES, VER\_SETFILES, VER\_SETS, VER\_SET\_UPDATES, VER\_UPDATES. Совокупность указанных таблиц называется репозитарием SAUMI.

Исходный репозитарий SAUMI содержится в БД поставки комплекса, и установка, как комплекта администратора, так и комплекта пользователя, производится посредством копирования соответствующих файлов и подкаталогов из репозитария в каталог установки. Само копирование на пользовательский компьютер осуществляется специальной службой обновления SAUMI, устанавливаемой на этом компьютере (см. «Сервис обновления и принудительная выгрузка компонент SAUMI из центрального репозитария» в руководстве «Быстрый старт»).

При вызове пользователем на локальном компьютере основного исполняемого модуля SAUMI `mmr32.exe`, при его загрузке указанная служба обновления получает запрос осуществить проверку соответствия имеющегося на компьютере дерева подкаталогов и файлов SAUMI хранимому в репозитарии. При такой проверке на локальном компьютере отсутствующие подкаталоги создаются, отсутствующие в указанном репозитарием подкаталоге файлы копируются из репозитария, а файлы, находящиеся в нужном подкаталоге, но отличающиеся от их одноименных копий в репозитарии, также копируются из репозитария.

Сравнение файлов на идентичность производится не побайтно, а сравнением их md5-хешей: несовпадение md5-хеша, вычисленного для файла на пользовательском компьютере, с хранимым в



репозитарии md5-хешем одноименного файла репозитария (значение поля MD5HASH в таблице VER\_FILES) означает неидентичность этих файлов и вызывает копирование файла из репозитария на пользовательский компьютер.

Принимая во внимание описываемый механизм обновления, следует иметь ввиду, что после изменения на пользовательском компьютере файлов SAUMI, в частности, локальных настроечных файлов, таких как `gnclient.ini`, `AutoPayment.dll.config` и др., следует сохранить измененные файлы в репозитарии, потому как иначе, первая же перезагрузка SAUMI скопирует из репозитария имеющуюся там копию файла и произведенные локально изменения будут потеряны.

## Замечание

Исключением являются файлы настроек `saumi.ini` и `saumi_user_params.ini` (см. Главу 5), которые не загружаются в репозитарий и не копируются оттуда.

Изменение набора файлов репозитария может быть произведено двумя способами: автоматическим и ручным.

Автоматическое изменение репозитария производится установкой пакетов обновления посредством утилиты из комплекта администратора «Менеджер обновлений» (см. Главу 2).

Для ручного (выборочного) изменения репозитария настоятельно рекомендуется применять утилиту «Управление репозитарием файлов SAUMI» из комплекта администратора, работа с которой описывается в приводимых ниже разделах.

Допустима коррекция репозитария прямым изменением записей таблицы VER\_FILES посредством выполнения соответствующего PL/SQL скрипта, однако при этом следует учитывать следующий момент: для того, чтобы новый (обновленный) файл в репозитарии копировался на клиентские компьютеры, необходимо пересчитать его md5-хеш, хранимый в поле MD5HASH, что и происходит при создании новой записи в этой таблице или при ее изменении, за тем исключением, что замена только тела файла (поле FILE\_BODY типа BLOB) не приводит к пересчету md5-хеша. В последнем случае следует значением поля MD5HASH установить NULL.

## Замечание

Таблицы репозитария VER\_SET\_UPDATES, VER\_UPDATES используются для замены указанной выше службы локального обновления SAUMI.

### 3.1.1. Управление локальным обновлением

В ряде случаев может потребоваться определенное влияние на механизм обновления. Для этого главный исполняемый модуль должен быть запущен из командной строки с соответствующим ключом. Ключ `/updated` подавляет описанный механизм локального обновления.

```
mmp32.exe /updated
```

Описанный выше механизм обновления срабатывает по умолчанию только при запуске главного исполняемого модуля из каталога установки (C:\SAUMI по умолчанию). В ряде случаев может требоваться запуск главного исполняемого модуля из каталога, созданного копированием содержимого каталога установки SAUMI – в этом случае обновление не вызывается. Чтобы задействовать механизм обновления и для этого случая следует запускать главный исполняемый модуль с ключом `/localupdate`.

```
mmp32.exe /localupdate
```

Ключ `/oldupdate` вызывает старый метод обновления, применявшийся в SAUMI до версии 4.6.6 включительно. Сутью этого метода был вызов в фоновом режиме исполнимого файла `cover.exe`, который осуществлял копирование подкаталогов и файлов из репозитария в каталог установки SAUMI. Это решение успешно работает в Windows XP и 2003, но в более поздних версиях Windows вступает в конфликт с устанавливаемым по умолчанию уровнем UAC (User access control), который зачастую запрещает такое копирование, даже под учетной записью, наделенной административными правами.

`mmp32.exe /oldupdate`

## 3.2. Вызов утилиты управления репозитарием

Вызов описываемой утилиты происходит в обычном для администраторских утилит порядке: Пуск | Комплект администратора SAUMI | Управление репозитарием файлов SAUMI.

В открывшемся окне регистрации пользователя следует ввести данные учетной записи привилегированного пользователя SM в БД SAUMI.

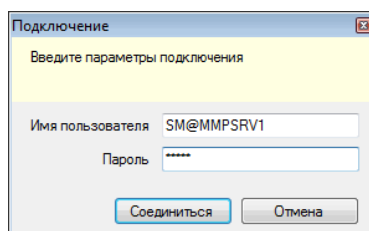


Рисунок 3.1.

В результате откроется окно, в левой панели которого отображается перечень файлов репозитария.

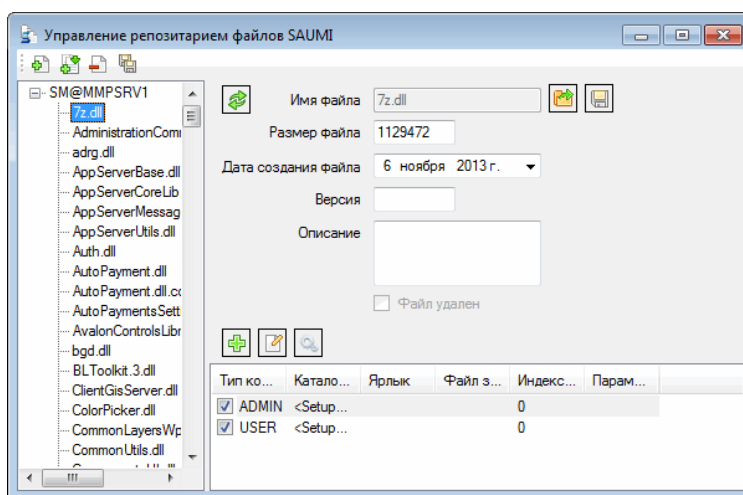



Рисунок 3.2.

Правая панель отображает свойства выделенного файла, учитываемые в репозитарии, часть из которых является резервом будущих изменений и поддерживается для совместимости, не влияя в настоящий момент на реализованный механизм локальных обновлений.

С помощью инструментальных кнопок производятся следующие операции с репозитарием.

1. Копирование файла из репозитария на пользовательский компьютер.
2. Добавление нового файла в репозитарий.
3. Задание настроек выгрузки файла репозитария.
4. Изменение имеющегося в репозитарии файла.
5. Изменение настроек выгрузки файла репозитария.
6. Добавление группы файлов в репозитарий.
7. Удаление файла из репозитария.

### 3.3. Копирование файла из репозитария на пользовательский компьютер

Чтобы скопировать файл из репозитария на пользовательский компьютер нужно выделить этот файл в левой панели описываемой утилиты и нажать кнопку  – «Сохранить файл на диск», находящуюся на правой панели.

В результате откроется стандартное диалоговое окно сохранения файла «Сохранить как».

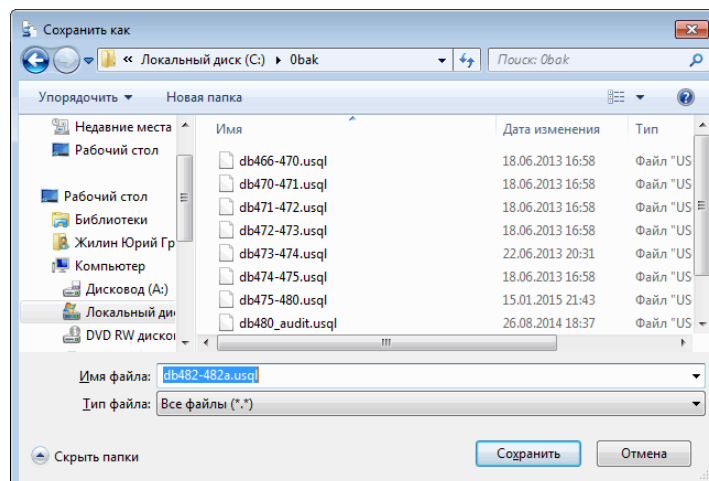



Рисунок 3.3.

Следует выбрать нужный каталог и сохранить там файл, нажав кнопку «Сохранить».

### 3.4. Добавление нового файла в репозитарий

Чтобы добавить новый файл в репозитарий, нужно нажать кнопку  – «Добавить файл» – на панели инструментов утилиты. В результате в нижней части левой панели появится новая позиция для отображения имени нового файла.

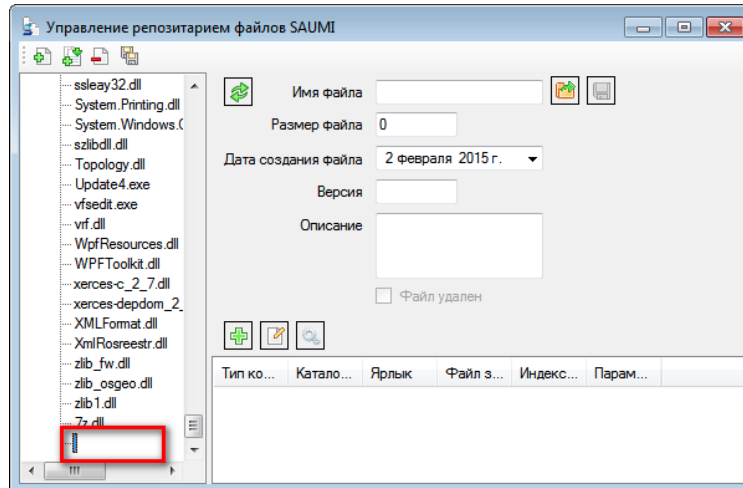



Рисунок 3.4.

Теперь необходимо загрузить в БД сам файл. Для этого нужно нажать кнопку , расположенную на правой панели. В открывшемся стандартном диалоговом окне нужно выбрать необходимый для пополнения репозитария файл и нажать кнопку «Открыть».

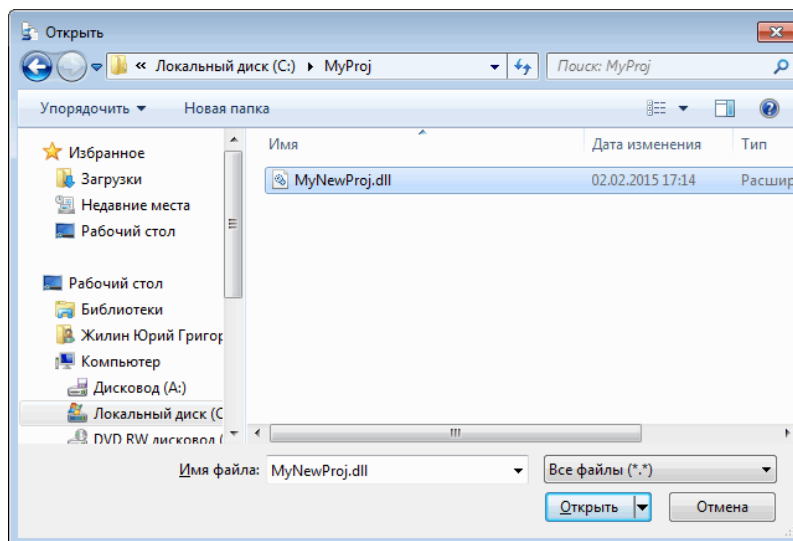


Рисунок 3.5.

Выбранный файл появится в конце списка файлов левой панели. Имени файла предшествует символ «\*», указывающий на тот факт, что с отмеченным файлом в репозитарии намечено произвести изменения либо самого файла, либо связанной с ним информации.

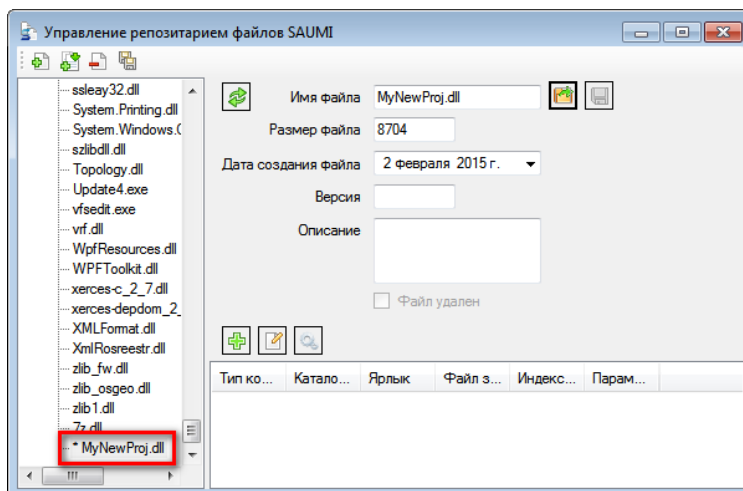



Рисунок 3.6.


В правой панели утилиты в верхней ее части также отображается имя файла и, кроме того, его размер и дата создания.

### Замечание

Выполненные действия по указанной выше схеме еще не производят непосредственную загрузку в репозитарий нового файла, а формируют перечень предполагаемых изменений в репозитарии. Непосредственное выполнение заданных изменений репозитария вызывается нажатием кнопки  – «Сохранить изменения» – на панели инструментов утилиты. Новый файл, предполагаемый к загрузке в репозитарий не имеет никаких настроек выгрузки и поэтому не сможет быть выгружен локальным сервисом обновления SAUMI после сохранения изменений. Следует предварительно задать настройки выгрузки нового файла.

## 3.5. Задание настроек выгрузки файла репозитария

Размещаемый в репозитарии новый файл необходимо снабдить настройками выгрузки – дополнительной информацией, которая позволит располагать новый файл в нужном каталоге на компьютере пользователя при выполнении автоматического локального обновления, а также отображать его в стартовом меню (для исполняемых модулей). Также необходимо определить, относится новый файл к комплекту администратора или к комплекту пользователя.

Чтобы задать новую настройку выгрузки файла, его нужно выделить в списке файлов левой панели репозитария и нажать кнопку  – «Добавить настройку выгрузки».

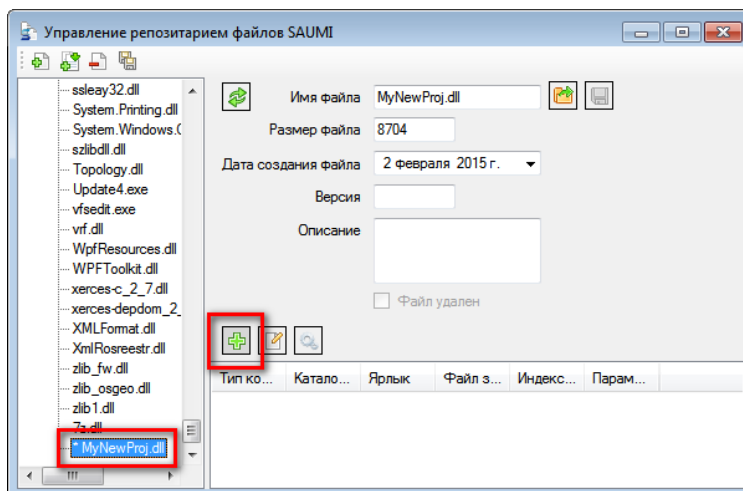


Рисунок 3.7.

В результате открывается окно настройки выгрузки выделенного в левой панели файла.

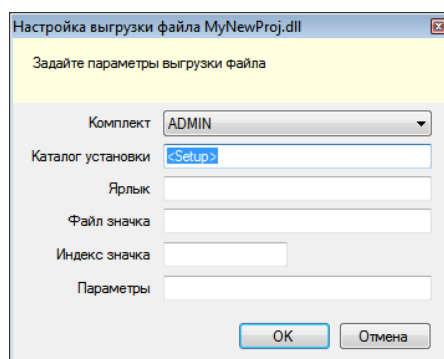


Рисунок 3.8.

Выбираемое из списка значение поля «Комплект» определяет, относится новый файл к комплекту администратора или к комплекту пользователя.

В поле *Каталог установки* по умолчанию подставляется строка *<Setup>*, которая означает подстановку имени каталога установки SAUMI (с: \SAUMI по умолчанию). Каталоги, вложенные в каталог установки SAUMI, задаются добавлением недостающей части названия каталога к префиксу *<Setup>*, например *<Setup>\PaymentFormats*. Каталоги, не вложенные в каталог установки SAUMI, должны указывать абсолютный путь с указанием используемого диска, например *D:\Import1C*.

Непустое поле *Ярлык* определяет представление файла в стартовом меню системы или в его подменю «Все программы». В первом случае значение поля должно начинаться со строки *<Start Menu>*, а во втором – со строки *<Programs>*, включая угловые скобки в обоих случаях. Следом должно идти имя ярлыка, включая вложенные уровни подменю. Например:

- *<Start Menu>\SAUMI 4.9i.lnk;*
- *<Programs>\Комплект администратора SAUMI\Диспетчер настроек.lnk.*

## Замечание

Отображение в стартовом меню рекомендуется применять для исполняемых файлов с расширением `.exe`.

Порождаемый пункт стартового меню или его подменю какого-либо уровня может быть снабжен отображаемым значком, который определяется полями *Файл значка*, *Индекс значка*.

Пустое поле *Файл значка* предполагает, что образ значка содержится в самом файле в соответствии со стандартами Windows, в частности таковыми могут быть исполняемые `.exe` файлы и `.dll` библиотеки. При указании файла-носителя образа значка, отличного от описываемого файла, следует использовать соглашение об обозначении содержащего каталога, описанного для поля *Каталог установки*.


Файл-носитель образа значка может содержать образы нескольких значков. Требуемый значок определяется полем *Индекс значка*, имеющим числовое значение.


Поле *Параметры* следует заполнять для исполняемых файлов. Введенная строка передается при вызове такого исполняемого файла как набор параметров командной строки запуска данного приложения (ключи запуска).

По завершении ввода значений полей окна «Настройка выгрузки файла» следует нажать кнопку **ОК**.


## Замечание

Методологически правильным будет при отнесении нового файла к комплекту пользователя посредством настройки выгрузки также создать аналогичную настройку выгрузки для комплекта администратора, чтобы функциональность комплекта администратора содержала все пользовательские возможности SAUMI.

Описанную процедуру настройки следует применить ко всем назначенным файлу типам комплектов. Отменить внесенные изменения можно нажатием кнопки  на правой панели утилиты.

Произведенные изменения репозитария вступают в силу после нажатия кнопки  - «Сохранить изменения» - на панели инструментов утилиты.

## 3.6. Изменение имеющегося в репозитарии файла

Чтобы заменить содержащийся в репозитарии файл, его нужно выделить в левой панели и нажать кнопку  на правой панели.

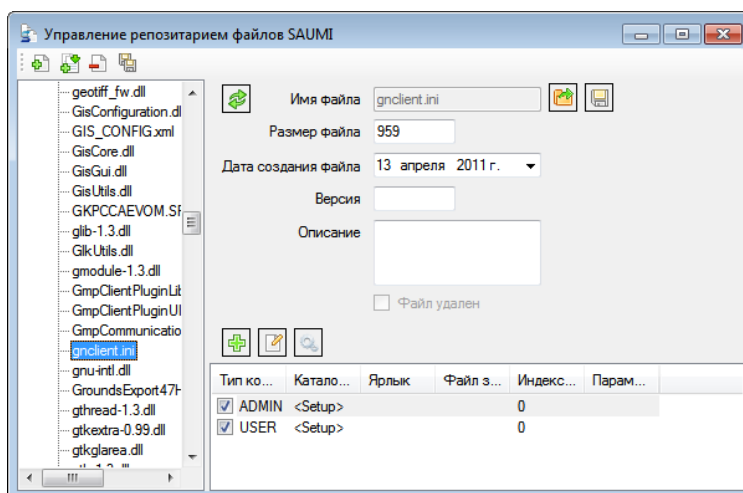


Рисунок 3.9.

В результате откроется стандартное окно выбора файла. Следует выбрать нужный файл и нажать кнопку «Открыть». Содержимое выбранного файла назначено для замены содержимого выделенного в репозитарии файла, о чем свидетельствует символ «\*», примыкающий к имени файла слева.

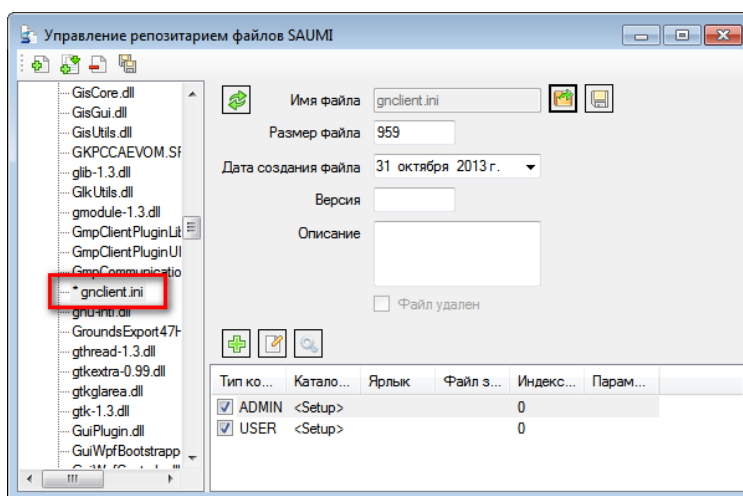



Рисунок 3.10.


Назначенное изменение содержимого файла может быть отменено нажатием кнопки  на правой панели.

Чтобы назначенное изменение вступило в силу, следует нажать кнопку  – «Сохранить изменения» – на панели инструментов утилиты.

### 3.7. Изменение настроек выгрузки хранимого в репозитарии файла

Если хранимый в репозитарии файл был сохранен без назначения ему настроек выгрузки, то задать последние можно в точности так же, как описано в разделе 3.5.



Для файла, сохраненного в репозитарии вместе с настройками выгрузки, можно произвести изменение последних. Для этого нужно выделить в левой панели имя данного файла, на правой панели изменяемую настройку и нажать кнопку  - «Изменить настройку выгрузки» - на дополнительной панели инструментов.

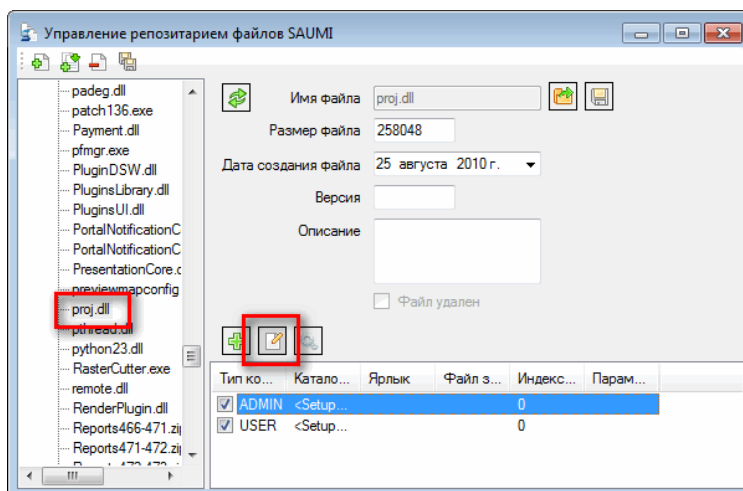



Рисунок 3.11.


В результате открывается окно редактирования настройки выгрузки данного файла, состав и назначение реквизитов которого представлены в разделе 3.5.

Произведенные изменения репозитария вступают в силу после нажатия кнопки  - «Сохранить изменения» - на панели инструментов утилиты.

### Замечание

Следует изменять настройки файлов репозитария из поставки SAUMI, а также из пакетов обновлений, в самых крайних случаях и с особыми предосторожностями. Неадекватное изменение может сделать программный комплекс неработоспособным.

## 3.8. Добавление группы файлов в репозитарий

Возможно одновременное назначение загрузки файлов в репозитарий, что может быть применено при разработке некоторого пользовательского расширения функционала SAUMI, реализованного несколькими файлами. Для назначения загрузки нескольких файлов в репозитарий следует нажать кнопку  - «Загрузить несколько файлов» - на панели инструментов утилиты.

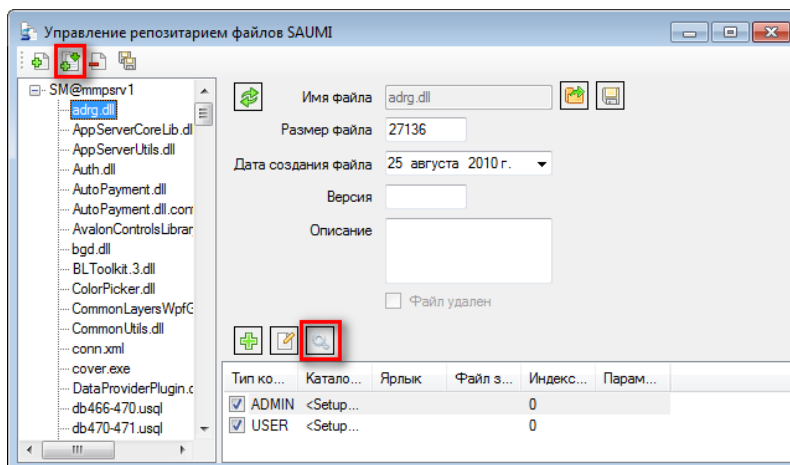


Рисунок 3.12.

В результате открывается стандартный диалог выбора файлов, в котором перечень отбираемых для загрузки в репозиторий SAUMI файлов может формироваться стандартными для Windows средствами работы со списками.

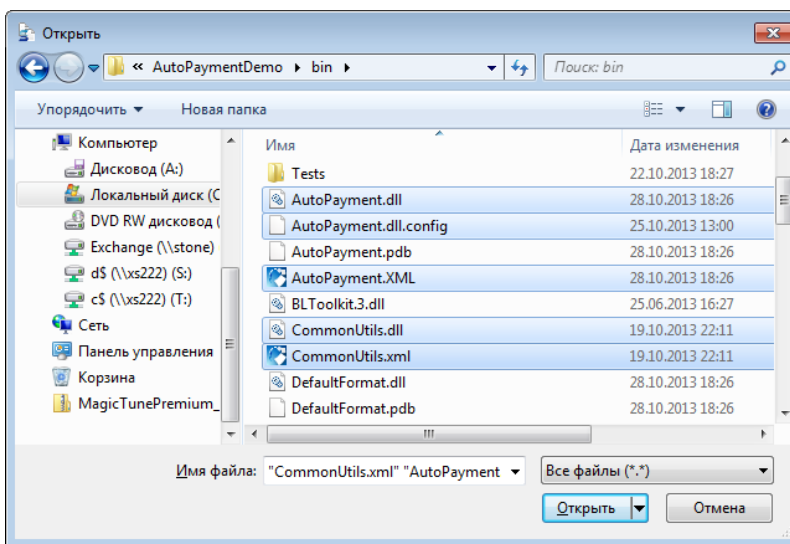


Рисунок 3.13.

В случае, если в репозитории уже есть файл с именем, имеющимся у одного из загружаемых файлов, то выводится диалоговое окно с предупреждающим сообщением.

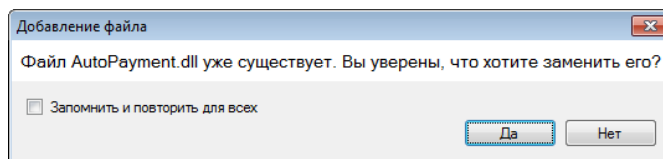




Рисунок 3.14.

Нажатие кнопки «Да» означает замену имеющегося в репозитории файла загружаемым, тогда как нажатие кнопки «Нет» означает запрет на загрузку файла с тем же именем. Флажок «Запомнить и повторить для всех» в выставленном состоянии позволяет перенести сделанный выбор по замене имеющегося файла или, наоборот, о запрете такой замены на все другие файлы, чьи имена представлены как в перечне файлов репозитария, так и в списке выбранных для загрузки файлов.

## Замечание

Так же как и при одиночном добавлении или замене файла репозитария выбранные к загрузке файлы только подготавливаются к их размещению в репозитории, позволяя, однако, производить с ними настроечные действия. Собственно загрузка в репозитарий производится после нажатия кнопки  – «Сохранить изменения».

Добавленные в репозитарий SAUMI файлы не имеют ни одного заданного параметра выгрузки и индивидуальная их настройка может занимать достаточно много времени, притом, что большинство загружаемых одновременно файлов могут иметь общие настройки. Существенно ускорить данную операцию можно копированием уже заданных параметров выгрузки одного файла целой группе файлов. Для этого следует выделить в списке левой панели утилиты файл, параметры которого будут копироваться, и нажать кнопку  – «Копировать настройки выгрузки». В результате открывается диалоговое окно выбора файлов репозитария, которым копируются параметры выгрузки выделенного файла.

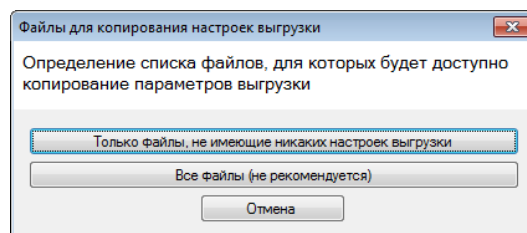


Рисунок 3.15.

Нажатие кнопки «Только файлы, не имеющие никаких настроек выгрузки» ограничивает целевую группу файлов файлами, у которых не задан ни один параметр выгрузки. В частности, такими файлами являются файлы сразу после назначения их для загрузки в репозитарий.

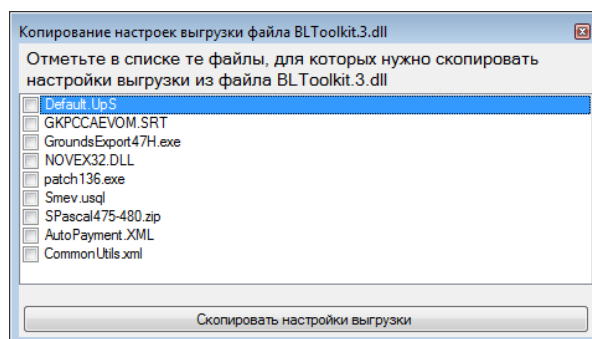


Рисунок 3.16.

Нажатие кнопки «Все файлы (не рекомендуется)» предлагает для копирования им параметров выгрузки выделенного файла все остальные файлы репозитария.

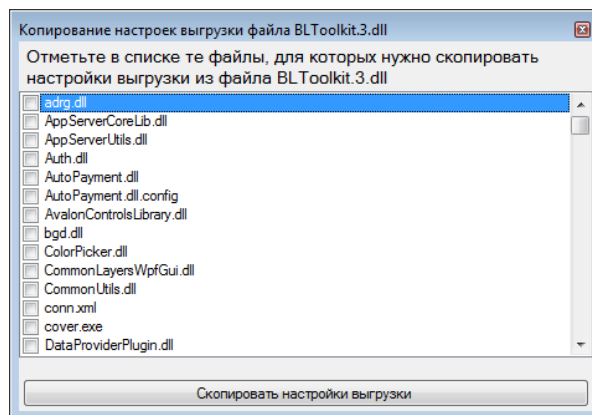


Рисунок 3.17.

Пользоваться данной возможностью нужно с большой осторожностью и осмотрительностью, потому как добавление к имеющимся правильным настройкам выгрузки некоторого файла новых и неправильных может сделать программный комплекс неработоспособным, а поиск такого файла и восстановление его настроек выгрузки может занять значительное время.

### Замечание

Копирование настроек не заменяет уже имеющиеся у файла репозитария настройки, а добавляет новые. Однако применяются именно последние настройки по принадлежности к комплекту (USER или ADMIN).

Список файлов в обоих случаях является потенциальным – рядом с каждым именем файла находится флажок, выставленное состояние которого указывает на вхождение данного файла репозитария в группу принимающих копируемые значения параметров выгрузки. Изменить состояние флажка помимо щелчка мышью в области флажка или имени файла можно также посредством контекстного меню, вызываемого нажатием правой кнопки мыши.

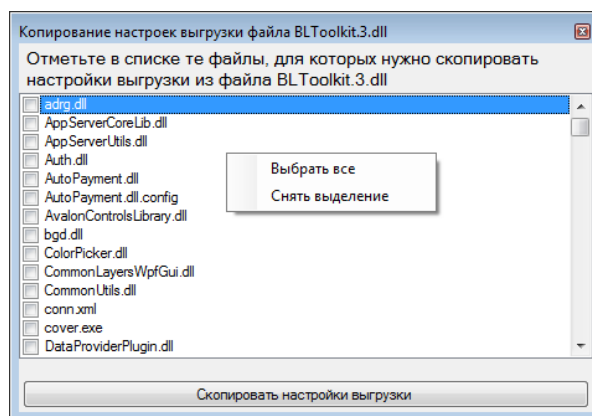


Рисунок 3.18.

Пункт контекстного меню «Выбрать все» выставляет все флажки списка, тогда как пункт «Снять выделение» все флажки переводит в сброшенное состояние.

После выделения всех файлов, требуемых для копирования параметров выгрузки, следует нажать кнопку «Скопировать настройки выгрузки». По завершении копирования настроек выводится соответствующее сообщение.

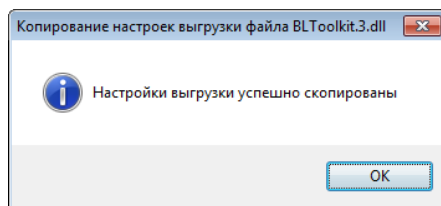




Рисунок 3.19.

Чтобы произведенные изменения репозитария как по составу и содержанию файлов, так и по их настройкам выгрузки, вступили в силу необходимо нажать кнопку  – «Сохранить изменения».

### 3.9. Удаление файла из репозитария

Чтобы удалить файл из репозитария его нужно выделить в списке левой панели и нажать кнопку  панели инструментов утилиты. В результате файл будет помечен на удаление, о чем свидетельствует символ «-», примыкающей слева к имени файла.

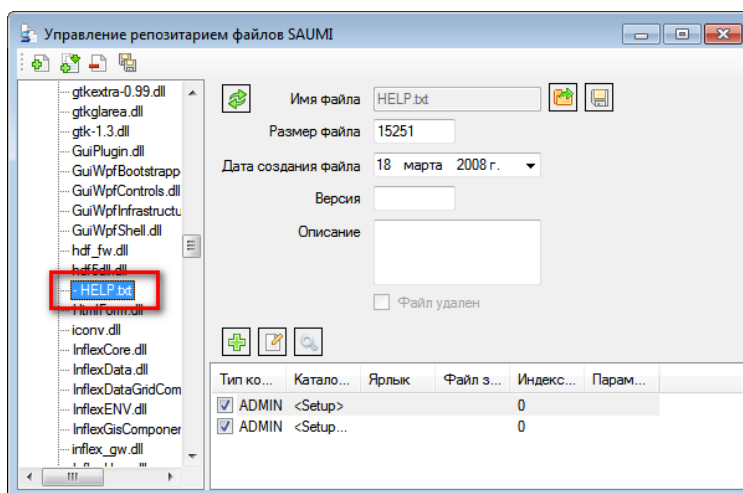



Рисунок 3.20.

Чтобы файл был окончательно удален из репозитария, следует нажать кнопку  панели инструментов.

### Предупреждение

Произведенное удаление является окончательным и удаленный файл может быть восстановлен только добавлением нового файла в репозитарий. Не следует удалять файлы репозитария из поставки SAUMI, а также из пакетов обновлений.

---

# Глава 4. «Диспетчер настроек»

## Содержание

4.1. Маски ввода кадастровых номеров .....	58
4.2. Параметры обмена данными отчетности .....	58
4.2.1. Формирование и настройка макета диска сбора данных .....	58
4.2.2. Местоположение формируемого набора файлов отчетных данных .....	59

В окне утилиты «Диспетчер настроек» из комплекта администратора отображается таблица SM\_SETTINGS. К этой таблице система периодически обращается за жизненно важными данными. Она имеет сходное, с реестром Windows, значение для работы. То есть - это универсальная таблица системных констант, которые могут устанавливаться только администратором системы. В таблице 4.1 приведен список обязательных параметров, которые фигурируют в базовой установке системы.

Таблица 4.1. Базовые параметры

Имя ключа	Описание ключа	Тип данных ключа
ORGNAME	Название организации	STRING
ENDPERIOD	Закрытие периода (проведение платежей)	DATE
DEFDATE	Системная дата по умолчанию (Y/N)	STRING
KUMI_ID	ID записи клиента, определяющего мун. собственность	INTEGER
LOG_CHECK	Флаг включения проверки CANMOVE (Y/N)	STRING
DEF_PAYDOCTYPE	Тип платежного документа по умолчанию (DOCTYPES)	INTEGER
DEF_PAYDOCSUBSCR	Орган подписания платежного документа по умолчанию (DOCSUBSCR)	INTEGER
USE_PARUSGATE	Флаг использования моста Oracle-ПАРУС	STRING
MANAGER_R	Полное название должности руководителя (для печатных документов)	STRING
MANAGER	Имя руководителя (для печатных документов)	STRING
REP_HEADER	Шапка для выписки из реестра (в качестве разделителя " ")	STRING

## Предупреждение

Список представлен не полностью. Помимо этих параметров есть параметры настройки финансовых обязательств и другие параметры, которые не менее важны. Этот список не является конечным и может быть расширен для собственных нужд, например, для использования в триггерах и хранимых процедурах.

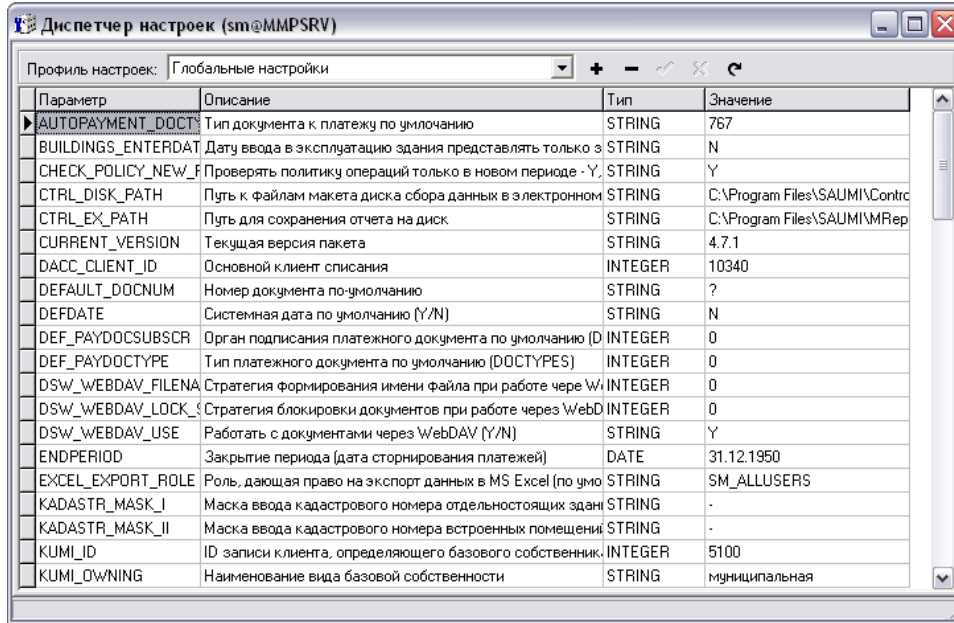


Рисунок 4.1.

## 4.1. Маски ввода кадастровых номеров

Объекты типа «Здания и помещения» имеют в окне свойств поле ввода кадастрового номера на вкладке «Общее», для которого может быть применена маска ввода, причем для отдельно стоящих зданий и встроенных помещений могут применяться разные маски. О задании масок ввода и их интерпретации см. 8.5.4.

Маски ввода кадастровых номеров задаются соответствующими параметрами в «Диспетчере настроек» SAUMI.

Наименование параметра	Назначение параметра
KADASTR_MASK_I	Маска ввода кадастрового номера отдельно стоящих зданий
KADASTR_MASK_II	Маска ввода кадастрового номера встроенных помещений

Приведенные в таблице параметры имеют тип STRING. Если какая-либо из указанных масок ввода не используется, то соответствующий параметр должен иметь значение "-" (символ минус).

## 4.2. Параметры обмена данными отчетности

Для успешного выполнения пользователями SAUMI обмена данными отчетности с подотчетными организациями (см. «Прием отчетных данных юридических лиц» в главе «Регистрация объектов и субъектов права в SAUMI» руководства пользователя) администратором должны быть выполнены ряд настроечных действий, включая задание значений некоторых параметров.

### 4.2.1. Формирование и настройка макета диска сбора данных

Одна часть формируемого для передачи в подотчетную организацию набора файлов содержит массив данных, выбираемых из БД SAUMI, тогда как другие файлы копируются в набор из так

называемого макета диска сбора данных. Последний есть также коллекция файлов, располагаемая в специально созданной папке, причем при каждом создании набора файлов сбора отчетности любой подотчетной организации все файлы из папки макета копируются в набор.

Папка, содержащая макет диска сбора данных, задается параметром CTRL\_DISK\_PATH, имеющим тип STRING. Сразу после развертывания базы данных SAUMI этот параметр указывает на папку c:\SAUMI\ControlDisk.

В папке установки SAUMI на пользовательском компьютере имеется вложенная папка ControlDisk, и, если у всех пользователей, которым вменяется обеспечение обмена отчетными данными с подотчетными организациями, клиентская часть SAUMI развернута в папке с одним и тем же именем, например c:\SAUMI, а также, если число таких пользователей невелико, то макет диска сбора данных можно однообразно сформировать (см. ниже) на компьютерах всех этих пользователей в локальной папке, задаваемой параметром CTRL\_DISK\_PATH. Однако гораздо более практичным с точки зрения внесения изменений в макет будет сформировать его, как разделяемый сетевой ресурс. В последнем случае значение параметра CTRL\_DISK\_PATH должно следовать UNC-нотации и иметь вид

```
\\сетевое_имя_компьютера\сетевое_имя_ресурса\путь_к_макету
```

где часть \путь\_к\_макету может и отсутствовать, если папка, содержащая макет, сама является сетевым ресурсом, сетевое имя которого подставлено в значение параметра CTRL\_DISK\_PATH.

Папку, содержащую макет диска сбора данных, нужно наполнить файлами. Там должны быть файлы setup.exe, HELP.txt, содержащиеся в папке ControlDisk, вложенной в папку установки SAUMI. Кроме того, комплект администратора в папке установки SAUMI содержит архивный файл Reports\_DS.rar, который нужно развернуть и все имеющиеся в архиве файлы также скопировать в папку макета.

## 4.2.2. Местоположение формируемого набора файлов отчетных данных

По умолчанию предполагается, что набор файлов для передачи в подотчетную организацию формируется на дискете, которой в силу принятых в Windows соглашений соответствует диск A:, причем используется корневой каталог. Тем самым, местоположением по умолчанию для набора файлов отчетных данных считается папка A:\.

В руководстве пользователя (см. «Прием отчетных данных юридических лиц» в главе «Регистрация объектов и субъектов права в SAUMI») приводится ряд аргументов для изменения местоположения формируемого набора файлов. Такое изменение задается параметрами USE\_EX\_PATH и CTRL\_EX\_PATH, имеющими тип STRING.

Параметр USE\_EX\_PATH может принимать два значения: 'Y' и 'N' (по умолчанию - 'N'). Значение 'Y' указывает на замену местоположения формируемых наборов файлов с папки A:\ на другую папку, полный путь к которой представлен конкатенацией нескольких строк

```
<значение параметра CTRL_EX_PATH> || '\' || <регистрационный номер (ID) юр.лица> ||  
'\' || <дата в формате ДД.ММ.ГГГГ>
```

где ID юридического лица представлен строкой цифр, а дата задает окончание текущего отчетного периода (см. «Прием отчетных данных юридических лиц» в главе «Регистрация объектов и субъектов права в SAUMI» руководства пользователя).

Задавая значение параметра CTRL\_EX\_PATH, как корневую папку обмена, следует обратить внимание, что эта папка должна существовать и быть доступна для создания и записи в ней файлов всем пользователям, осуществляющим прием отчетных данных из подотчетных организаций. С



точки зрения администратора самым удобным будет назначение в качестве корневой папки обмена сетевого ресурса, доступного с указанными выше полномочиями осуществляющим обмен пользователям. В этом случае, также как и для параметра CTRL\_DISK\_PATH, необходимо придерживаться UNC-нотации задания имен сетевых ресурсов.

---

# Глава 5. Файлы настроек главного исполняемого модуля SAUMI

## Содержание

5.1. Файл saumi.ini .....	61
5.1.1. Настройки считывателя штрих-кода – секция [BarCode] .....	62
5.1.2. Настройки протокола WebDAV – секция [DSW_WebDAV] .....	62
5.2. Файл saumi_user_params.ini .....	64
5.2.1. Параметры последнего соединения с БД SAUMI – секция [Common] .....	64
5.2.2. Коды горячих клавиш Главной панели SAUMI – секция [HotKeys] .....	64
5.2.3. Параметры поведения Главной панели SAUMI – секция [MainPanel] .....	65
5.2.4. Параметры оптимизации производительности клиентской части SAUMI – секция [Performance] .....	66
5.3. Общие секции для файлов saumi.ini и saumi_user_params.ini .....	66
5.3.1. Параметры отображения окон данных – секция [ExplorerWindow] .....	66
5.3.2. Журналирование программных событий – секция [Logging] .....	67

На рабочих станциях применяются индивидуальные настройки главного исполняемого модуля SAUMI. Параметры этих настроек располагаются в двух конфигурационных файлах:

- saumi.ini – содержит общие для рабочей станции настройки;
- saumi\_user\_params.ini – содержит настройки, соответствующие конкретному пользователю системы Windows.

Описываемые конфигурационные файлы являются простыми текстовыми файлами, построчно содержащими параметры. Каждый параметр имеет вид

```
param_name=param_value
```

где param\_name является именем (идентификатором) параметра, а param\_value строкой символов, задающей значение параметра.

Параметры конфигурационных файлов сгруппированы в именованные секции, каждая такая секция начинается со строки, содержащей имя секции, заключенное в квадратные скобки.

Файлы saumi.ini и saumi\_user\_params.ini могут содержать одинаковый набор секций, однако содержимое секций [Common], [HotKeys], [MainPanel] и [Performance] будет считываться только из файла saumi\_user\_params.ini, одноименные секции файла saumi.ini будут игнорироваться. Напротив, настройки секций [BarCode] и [DSW\_WebDAV] считываются из файла saumi.ini, одноименные секции файла saumi\_user\_params.ini даже при их наличии будут проигнорированы.

## 5.1. Файл saumi.ini

Файл saumi.ini размещается в папке установки SAUMI. Файл перезаписывается каждый раз при выходе из SAUMI. Если в момент запуска SAUMI этот файл отсутствует – он создается с настройками параметров по умолчанию.

В файле saumi.ini учитываются следующие секции:

```
[BarCode]
```

[DSW\_WebDAV]

### 5.1.1. Настройки считывателя штрих-кода – секция [BarCode]

В этой секции описываются настройки считывателя штрих-кода.

**Параметр** Enabled

Включает и выключает драйвер сканера штрих-кода.

Допустимые значения:

True, False.

**Примеры:**

Enabled=True – драйвер включен;

Enabled=False – драйвер выключен.

**Параметр** CommPort

Определяет, к какому COM-порту подключен сканер штрих-кода.

Допустимые значения:

COM1, COM2, COM3 и COM4.

**Параметр** BaudRate

Определяет скорость работы порта.

Допустимые значения:

2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200.

**Параметр** DataBits

Определяет разрядность работы порта (количество бит).

Допустимые значения:

7 или 8.

Значения по умолчанию:

```
[BarCode]
Enabled=False
CommPort=COM2
BaudRate=9600
DataBits=8
```

### 5.1.2. Настройки протокола WebDAV – секция [DSW\_WebDAV]

В этой секции описываются настройки протокола WebDAV на локальной машине.

**Параметр** Use

Определяет локальное отключение протокола WebDAV.

Допустимые значения:

0, 1

**Примеры:**

Use=0 – отключение протокола WebDAV;

Use=1 – протокол WebDAV включен.

**Периодическая группа параметров:** Ext\_N, App\_N, Params\_N, где N=1,2,3 и т.д., задает ассоциацию типа файла, определяемого его расширением, и приложения, которое вызывается для его обработки. Params\_N задает параметры командной строки при вызове N-го

ассоциированного приложения, фрагмент строки '{0}' в Params\_N указывает подстановку имени открываемого файла в команду запуска приложения.

**Пример:**

```
Ext_1=txt
App_1=C:\Program Files\Microsoft Office\OFFICE11\WINWORD.EXE
Params_1={0}
```

**Параметры** MinPort, MaxPort

Определяют диапазон номеров портов протокола TCP/IP, по которым производится взаимодействие между главным исполняемым файлом SAUMI, выступающим сервером WebDAV, и офисными приложениями, являющимися WebDAV-клиентами.

Следует использовать достаточно большие номера портов, не используемые другими приложениями. Перечень занятых в системе TCP/IP портов можно узнать из вывода консольной команды Windows **netstat -a**.

**Пример:**

```
MinPort=36422
MaxPort=36457
```

В случае использования на компьютере сетевого экрана (брандмауэра), отличного от брандмауэра из поставки Windows, следует настроить его так, чтобы он обеспечил взаимодействие главного модуля SAUMI с офисными приложениями в установленном диапазоне портов TCP/IP в рамках узла localhost со стандартным IP-адресом 127.0.0.1.

**Параметр** WriteLog

Отключает/включает журналирование WebDAV протокола.

Допустимые значения:

0, 1.

**Примеры:**

WriteLog=0 – отключение журналирования протокола WebDAV;  
WriteLog=1 – журналирование протокола WebDAV включено.

**Параметр** LogHttpHeaders

Отключает/включает журналирование содержимого заголовков HTTP-сообщений, обрабатываемых WebDAV-сервисом. При отключенном журналировании WebDAV протокола (WriteLog=0), журналирование содержимого заголовков HTTP-сообщений также не выполняется.

Допустимые значения:

0, 1.

**Примеры:**

LogHttpHeaders=0 – отключение журналирования содержимого заголовков HTTP-сообщений;  
LogHttpHeaders=1 – журналирование содержимого заголовков HTTP-сообщений включено.

**Параметр** LogFileName

Задаёт файл журналирования WebDAV протокола.

**Пример:**

```
LogFileName=c:\log_dsw.txt
```

Значения по умолчанию:

[DSW\_WebDAV]

```
Use=1
MinPort=36422
MaxPort=36457
WriteLog=0
LogHttpHeaders=0
LogFileNames=<SAUMI директория>\log_dsw.txt
```

Здесь <SAUMI директория> – директория установки программного комплекса SAUMI.

## 5.2. Файл saumi\_user\_params.ini

Файл saumi\_user\_params.ini размещается в директории, определяемой профилем текущего пользователя системы Windows. В Windows XP – это путь %USERPROFILE%\Local Settings\Application Data\Saumi, а в Windows 7 и последующих версиях Windows – %USERPROFILE%\AppData\Local\Saumi, где USERPROFILE есть переменная среды текущего пользователя Windows.

Так же, как и saumi.ini, этот файл перезаписывается каждый раз при выходе из SAUMI, а если в момент запуска SAUMI этот файл отсутствует – он создаётся с настройками параметров по умолчанию.

В файле saumi\_user\_params.ini учитываются следующие секции:

```
[Common]
[HotKeys]
[MainPanel]
[Performance]
```

### 5.2.1. Параметры последнего соединения с БД SAUMI – секция [Common]

В этой секции описываются параметры последнего соединения с БД SAUMI, произведённого на данной рабочей станции.

При запуске главного исполняемого модуля SAUMI используются значения параметров из этой секции для подстановки в окно подключения.

**Параметр** LastLogin

Формат записи значения:

```
[имя_пользователя]@[название_БД]
```

**Пример:**

```
LastLogin=SM@MMPSRV
```

### 5.2.2. Коды горячих клавиш Главной панели SAUMI – секция [HotKeys]

В этой секции описываются коды горячих клавиш Главной панели SAUMI.

Параметр	Выполняемая функция
hkSAUMI	Открывает меню « SAUMI »
hkCreateNewItem	Открывает меню «Создать»
hkFind	Открывает меню «Поиск»
hkObligFind	Открывает окно «Поиск обязательств» (пункт «Обязательства»)

Параметр	Выполняемая функция
hkReports	Открывает меню «Отчёты»
hkHistory	Открывает меню «История»
hkPlugins	Открывает меню «Дополнения»
hkNewObject	Создать новый Объект
hkNewClient	Создать нового Субъекта права
hkNewDoc	Создать новый Документ
hkShowCase	Отображает/скрывает окно буфера SAUMI
hkFindSAUMIPanel	Заставляет кратковременно мерцать «плавающую» панель SAUMI

#### Формат записи горячих клавиш:

[Ctrl+] [Shift+] [Alt+] [Название алфавитно-цифровой или функциональной клавиши]

#### Значения по умолчанию:

```
[HotKeys]
hkSAUMI=Ctrl+F4
hkCreateNewItem=Ctrl+F6
hkFind=Ctrl+F7
hkObligFind=Ctrl+F8
hkReports=Ctrl+F9
hkHistory=Ctrl+F10
hkPlugins=Ctrl+F11
hkNewObject=Shift+F9
hkNewClient=Shift+F10
hkNewDoc=Shift+F11
hkShowCase=Ctrl+F12
hkFindSAUMIPanel=Shift+F7
```

#### Допустимые значения функциональных клавиш:

```
BkSp   Tab     Esc     Enter
Space  PgUp   PgDn    End
Home   Left   Up       Right
Down   Ins    Del
```

Если клавиатура поддерживает специальные клавиши, то можно указать скан-код этой клавиши, поставив перед ним знак решетки '#'.  
 Например: hkFind=#12574

#### Пример:

```
hkFind=#12574
```

Если значение оставить пустым – горячая клавиша будет отключена для этой функции.

### 5.2.3. Параметры поведения Главной панели SAUMI – секция [MainPanel]

В этой секции описываются параметры поведения Главной панели SAUMI.

#### Параметр RegisterAsAppBar

Определяет подвижность или закреплённость панели SAUMI.

Допустимые значения:

True, False.

#### Примеры:

RegisterAsAppBar=True – панель SAUMI постоянно располагается в самом верху экрана.  
RegisterAsAppBar=False – панель SAUMI становится «плавающей» - т.е. становится возможным её перемещать по экрану и располагать в желаемом месте экрана.

Значение по умолчанию:

```
[MainPanel]
RegisterAsAppBar=True
```

## 5.2.4. Параметры оптимизации производительности клиентской части SAUMI – секция [Performance]

В этой секции описываются параметры, позволяющие оптимизировать производительность клиентской части SAUMI при выполнении различных пользовательских операций.

**Параметр** MAX\_OBJ\_TO\_CLIPBOARD

Ограничивает количество элементов, помещаемых в буфер обмена за одну пользовательскую операцию (копирование перетаскиванием мышью между окнами или в буфер обмена длительного хранения). Значение указанного параметра секции [Performance] файла saumi\_user\_params.ini имеет приоритет перед значением одноименного с ним глобального параметра, задаваемого посредством утилиты «Диспетчер настроек», и имеющего то же назначение.

Допустимые значения:

0, положительные целые числа.

**Примеры:**

MAX\_OBJ\_TO\_CLIPBOARD=100 – одновременно мышью может быть «перетащено» не более 100 элементов.

MAX\_OBJ\_TO\_CLIPBOARD=0 – ограничение на количество элементов, помещаемых в буфер обмена за одну пользовательскую операцию, отменяется.

## 5.3. Общие секции для файлов saumi.ini и saumi\_user\_params.ini

Некоторые секции параметров могут присутствовать как в файле saumi.ini, так и в файле saumi\_user\_params.ini.

Общими для файлов saumi.ini и saumi\_user\_params.ini являются секции:

```
[ExplorerWindow]
[Logging]
```

### 5.3.1. Параметры отображения окон данных – секция [ExplorerWindow]

Параметры этой секции позволяют задать многострочное отображение заголовков и/или ячеек колонок в таблицах окон данных. Значения параметров секции [ExplorerWindow] в файле saumi\_user\_params.ini имеют приоритет над значениями одноименных параметров этой же секции в файле saumi.ini.

**Параметр** ColumnHeaderWordWrap

Включает и выключает режим многострочности для заголовков колонок таблицы окна данных.

Допустимые значения:

0, 1.

**Примеры:**

ColumnHeaderWordWrap=0 – многострочность в заголовках колонок отключена;

ColumnHeaderWordWrap=1 – многострочность в заголовках колонок включена.

**Параметр** CellValueWordWrap

Включает и выключает режим многострочности для ячеек таблицы окна данных.

Допустимые значения:

0, 1.

**Примеры:**

CellValueWordWrap=0 – многострочность в ячейках таблицы отключена;

CellValueWordWrap=1 – многострочность в ячейках таблицы включена.

Значения по умолчанию:

**[ExplorerWindow]**

ColumnHeaderWordWrap=0

CellValueWordWrap=0

Для значений

**[ExplorerWindow]**

ColumnHeaderWordWrap=1

CellValueWordWrap=1

включена многострочность, как в заголовках колонок, так и в ячейках.

## 5.3.2. Журналирование программных событий – секция [Logging]

Являясь сложной инженерной системой, программный комплекс SAUMI обладает определенным объемом скрытых дефектов, которые иногда проявляются в виде случайных и трудно воспроизводимых ошибок при работе пользователей с комплексом.

В новой версии реализована возможность ведения журналирования программных событий на пользовательском компьютере, которая позволяет существенно ускорить обнаружение указанных дефектов.

Журналирование является отключаемым, так как во включенном состоянии оно использует значительные вычислительные ресурсы, как пользовательского компьютера, так и сервера БД SAUMI, и использовать его следует осмотрительно.

Режим журналирования программных событий определяется параметрами секции [Logging], которая может присутствовать как в файле saumi.ini, так и в файле saumi\_user\_params.ini. Значения параметров секции [Logging] в файле saumi\_user\_params.ini имеют приоритет над значениями одноименных параметров этой же секции в файле saumi.ini.

**Параметр** EnableLogging

Включает и выключает режим журналирования программных событий.

Допустимые значения:

0, 1.

**Примеры:**



EnableLogging=0 – журналирование отключено;

EnableLogging=1 – журналирование включено.

**Параметр** LogConfigFile

Задаёт путь к специальному файлу конфигурации, определяющему все параметры ведения журналирования [необязательный параметр]. Если путь задан относительным, то относительно папки установки SAUMI.

**Пример:**

LogConfigFile=log\log4net.config.xml – путь к конфигурационному файлу относительно папки установки SAUMI.

**Параметр** LogOutputFile=log-file.txt

Задаёт путь к файлу журнала программных событий (данный параметр игнорируется, если задан параметр LogConfigFile) [необязательный параметр]. Если путь задан относительным, то относительно папки установки SAUMI.

**Пример:**

LogOutputFile=log\log-file.txt – путь к файлу относительно папки установки SAUMI.

Значения по умолчанию:

[Logging]

EnableLogging=0

---

# Глава 6. Аудит изменения БД SAUMI

## Содержание

6.1. Настройка аудита .....	69
6.2. Данные аудирования .....	72
6.3. Отчеты, представляющие и сопровождающие аудит .....	73
6.4. Архивирование данных аудита .....	75

Описываемая ниже функциональность предназначена для сбора информации об изменении данных, содержащихся в БД SAUMI, а также для обеспечения визуального представления детальной истории зафиксированных изменений. Под изменениями данных понимаются операции добавления и удаления записей различных таблиц БД, а также изменение значений реквизитов отдельных записей в этих таблицах.

Аудит данных основывается на использовании триггеров БД Oracle, которые срабатывают на события BEFORE INSERT, BEFORE DELETE, BEFORE UPDATE. Генерация следящих за изменениями триггеров происходит автоматически средствами комплекса SAUMI, после определения наборов отслеживаемых таблиц и реквизитов в них, производимых при настройке аудита.

Визуализация собираемых сведений об изменениях осуществляется посредством ряда отчетов.

### ВНИМАНИЕ!!!

Подключение аудита вызывает существенную дополнительную нагрузку на сервер БД SAUMI. Чрезмерное стремление отслеживать все и вся может сделать замедление реакции сервера базы данных неприемлемым. Следует тщательно спланировать аудит и применить его только к действительно важным элементам информации.

## 6.1. Настройка аудита

Для настройки аудита применяются следующие параметры:

*AUDIT* – аудит включен/отключен ('Y'/'N', по умолчанию – 'N');  
*AUDIT\_MOD* – метод определения элементов информации для аудирования ('AUTO' по умолчанию / 'MANUAL');  
*AUDIT\_SAVEMONTH* – см. 6.4.

### Замечание

Значения указанных параметров могут быть изменены посредством утилиты «Диспетчер настроек»(см. главу 4) из комплекта администратора SAUMI.

Изначально аудирование отключено (*AUDIT*='N') и для его включения необходимо с помощью утилиты «Диспетчер настроек» задать параметру *AUDIT* значение 'Y'. При включении аудита все триггеры, которые отслеживают изменения (полный перечень таких триггеров находится в таблице *AUDIT\_TRIGGERS*), переходят из состояния *disable* в состояние *enable*. Отключение аудита переводит указанные триггеры снова в состояние *disable*.

При включении аудита заданием параметру *AUDIT* значения 'Y' осуществляется проверка некоторых объектов в БД, с помощью которых производится аудирование, в частности пакет *AUDIT\_PKG*. Если его нет или он находится в состоянии *INVALID*, – в обоих этих случаях включить аудирование изменения данных невозможно.

Автоматическое определение элементов информации для аудирования (*AUDIT\_MOD='AUTO'*) сводится к указанию таблиц, все поля которых подвергаются аудиту на изменение их значений. Перечень таблиц, по которым производится аудит, указывается в таблице AUDIT\_TABLES. Изначально в перечне аудируемых таблиц указаны следующие таблицы.

	Наименование таблицы	Описание и комментарии
1	ADDRESS	Адреса
2	ATO	Административно-территориальные образования
3	BUILDINGS	Здания и помещения
4	CHARGES	Начисления по обязательству
5	CLIENTS	Субъекты права (аудит создания и удаления ведется по типам)
6	COMPLEX_PROPERTY	Имущественные комплексы
7	DEFFERS	Отсрочки
8	DISCOUNTS	Льготы
9	DOCUMENTS	Документы
10	ENGINF	Инженерная инфраструктура
11	GROUNDS	Земельные участки
12	MOVEITEMS	Движение объектов
13	MOVEPERIODS	Периоды операции движения
14	MOVESETS	Операции движения
15	OBJECTS	Объекты (аудит создания и удаления ведется по типам)
16	OBJSHARES	Пакеты акций, доли в капитале
17	OPERSHARES	Операции с уставным (складочным) капиталом
18	ORGANISATIONS	Юридические лица
19	PAYDOCS	Условия расчетов по обязательству в периоде ОД
20	PAYMENTS	Платежи
21	PRIVATES	Физические лица
22	PROPERTY	Прочее движимое имущество
23	STATES	Состояние
24	TRANSPORT	Транспортные средства
25	UNCONSTR	Незавершенное строительство

Записи таблицы AUDIT\_TABLES имеют реквизиты-флажки:

Аудит создания – поле INS\_AUD (значения 'Y'/'N');

Аудит изменений – поле UPD\_AUD (значения 'Y'/'N');

Аудит удаления – поле DEL\_AUD (значения 'Y'/'N'),

которые определяют виды аудита, применяемые к аудируемой таблице, перечисленной в AUDIT\_TABLES. Значение 'Y' реквизита-флажка означает применение соответствующего аудита к данной таблице, тогда как значение 'N' такой аудит отключает.

Для привлечения к аудиту других таблиц необходимо в AUDIT\_TABLES добавить запись с названием таблицы, по которой требуется проведение аудита и описанными выше реквизитами-флажками указать вид применяемого аудита.

## Замечание

Добавить новую запись в таблицу `AUDIT_TABLES`, а также изменить значения полей уже имеющихся записей можно с помощью утилиты «Менеджер справочников» (см. главу 12) из комплекта администратора SAUMI.

Чтобы осуществленные изменения в таблице `AUDIT_TABLES` по описанной выше схеме имели эффект в механизме аудирования, необходимо вызвать из меню «SAUMI 4.10» подменю «Настройки», а в нем пункт «Актуализировать систему аудирования». В результате автоматически производятся действия, направленные на осуществление соответствия следящих триггеров аудита текущим настройкам.

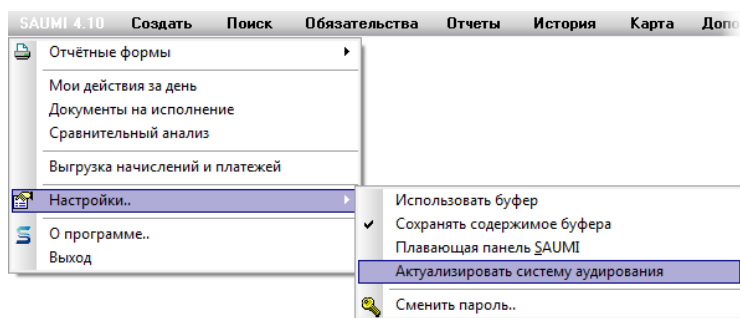


Рисунок 6.1.

## Замечание

Пункт «Актуализировать систему аудирования» отображается в подменю «Настройки», вызываемом из меню «SAUMI 4.10», если пользователю предоставлена роль `SM_AUDIT`. Назначение ролей пользователям производится посредством утилиты «Диспетчер пользователей» (см. главу 1).

Ручное определение элементов информации для аудирования (`AUDIT_MOD='MANUAL'`) позволяет более детально установить перечень реквизитов сущностей, по которым требуется проведение аудита изменения значений этих реквизитов. Когда параметр `AUDIT_MOD` имеет значение 'MANUAL' в «Менеджере реквизитов» для каждой таблицы к описывающим реквизиты полям добавляется поле «Аудит», выставленный флажок в котором подразумевает ведение аудита изменения по данному реквизиту.

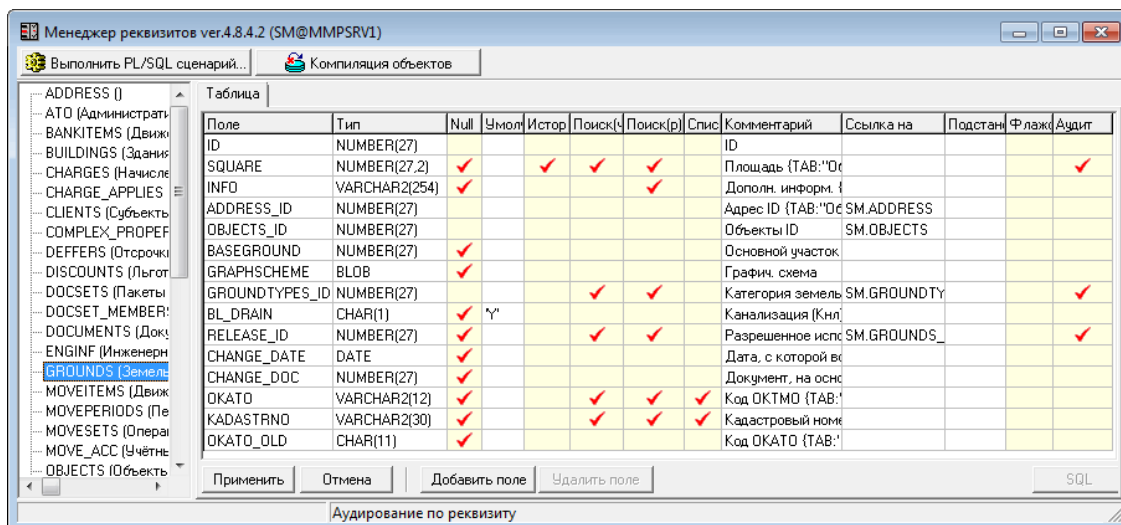


Рисунок 6.2.

Реквизиты таблиц, неотмеченные указанным способом, не подвергаются аудиту изменения при рассматриваемом значении параметра `AUDIT_MOD='MANUAL'`.

### Замечание

В текущей версии поля типа `SDO_GEOMETRY` при любом значении параметра `AUDIT_MOD` аудиту не подлежат.

Аудит удаления и создания новых записей при ручном определении элементов информации для аудирования (`AUDIT_MOD='MANUAL'`) задаются реквизитами-флажками Аудит создания (поле `INS_AUD`) и Аудит удаления (поле `DEL_AUD`) таблицы `AUDIT_TABLES`, как описано выше для автоматического режима (`AUDIT_MOD='AUTO'`), при этом состояние реквизита-флажка Аудит изменений (поле `UPD_AUD`) игнорируется.

При ручном определении элементов информации для аудирования (`AUDIT_MOD='MANUAL'`) ввод в действие выполненных изменений производится вызовом из меню «SAUMI 4.10» подменю «Настройки», а в нем пункта «Актуализировать систему аудирования» (рис. 6.1).

## 6.2. Данные аудирования

Изменения аудируемых элементов информации фиксируются триггерами, что дает возможность получить как старые, так и новые их значения. Каждое изменение данных, так или иначе, привязывается к определенной родовой сущности SAUMI, обеспечивая тем самым логическую целостность информации об этом изменении. Например, при изменении адреса, происходит поиск той сущности, которой он (адрес) принадлежит. Это может быть объект или субъект. Поиск подобных связей производится исключительно на ссылочности между таблицами. Если между таблицами не существует таких связей, изменение данных не сможет быть привязано к сущности.

Существуют четыре родовых вида сущностей, к которым могут быть привязаны изменения данных: объекты, субъекты, документы, операции движения. Этим сущностям соответствуют сопровождающие аудит отчеты.



меню могут быть открыты окна свойств экземпляров родовых сущностей SAUMI, к которым относится изменение (кроме удаленных).

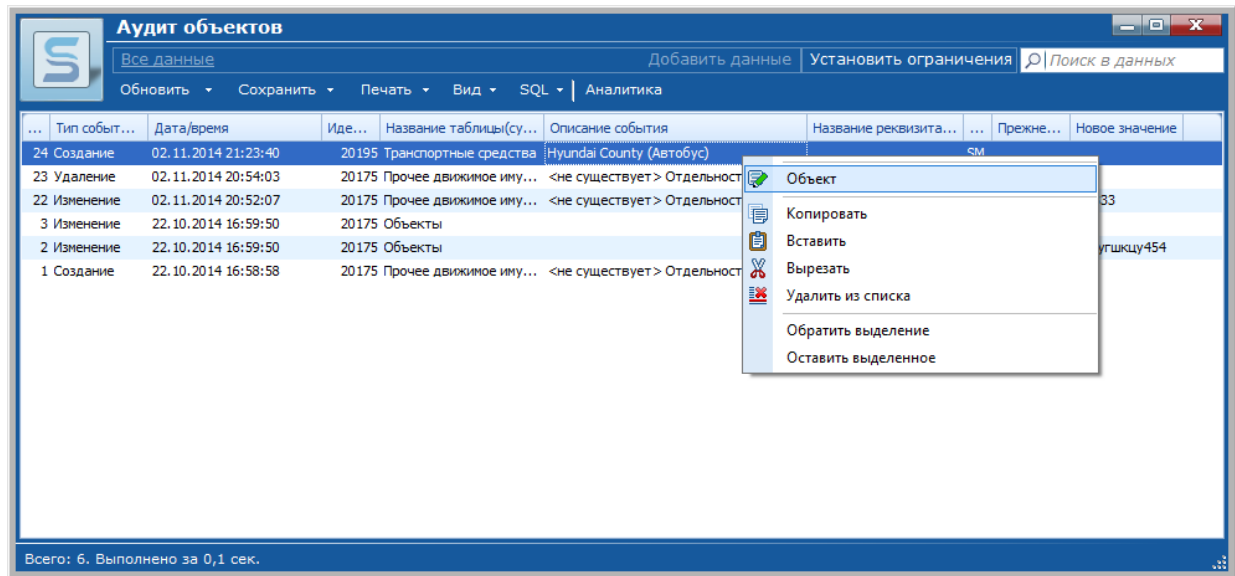


Рисунок 6.5.

## Замечание

Префикс <не существует> в описании отдельного события в некотором отчете по аудиту свидетельствует о том, что соответствующий экземпляр сущности SAUMI удален из БД, а его описание представлено из данных аудита.

Отчет «Системные события аудита» представляет информацию по событиям в настройках самого аудита.

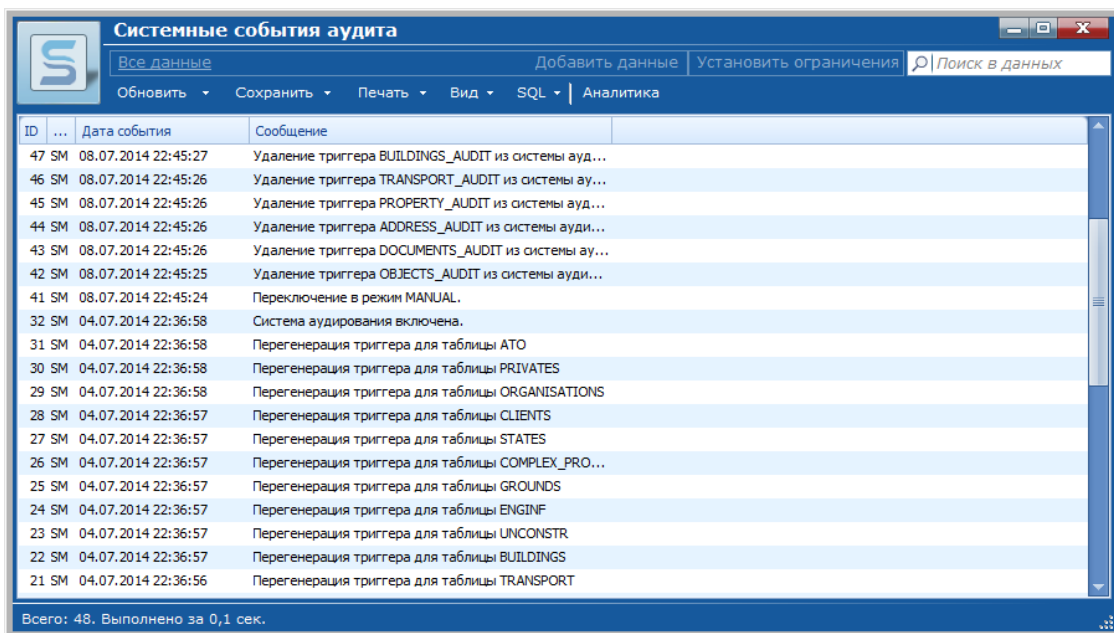


Рисунок 6.6.

## 6.4. Архивирование данных аудита

Аудирование – существенно ресурсоемкий процесс, отнимающий у сервера БД не только процессорное время, но и определенный объем памяти долговременного хранения данных при том, что со временем часть данных аудита теряет свою актуальность, продолжая удерживать ресурсы БД. Для уменьшения объемов хранимой в БД информации аудита предусмотрена возможность по истечении определенного промежутка времени, измеряемого в месяцах, прошедшего с момента создания некоторого элемента информации аудита, скопировать этот элемент во внешний файл определенного формата, а затем удалить из базы данных.

Указанный выше временной интервал, измеряемый в месяцах, задается параметром `AUDIT_SAVEMONTH`, значение которого по умолчанию 999999 означает, что архивирование данных аудита с удалением из БД не осуществляется. Значение параметра может быть изменено посредством утилиты «Диспетчер параметров».

В случае указания реалистичного значения параметра `AUDIT_SAVEMONTH`, определяющего реальное архивирование данных аудита, необходимо зарегистрировать в БД SAUMI в схеме SM директорию, в которой будут создаваться архивные файлы. Для этого необходимо выполнить следующую команду.

```
CREATE OR REPLACE DIRECTORY AUDIT_ARH_PATH AS 'имя_директории';
```

Здесь `AUDIT_ARH_PATH` – уникальное имя объекта базы данных типа DIRECTORY OBJECT, заданное для указания директории сохранения создаваемых архивных файлов аудита, а *имя\_директории* – полное имя директории в файловой системе компьютера, на котором развернут сервер ORACLE, управляющий БД SAUMI. Имя директории в тексте команды должно быть заключено в апострофы. Приведенную выше команду можно выполнить с помощью любого программного средства, интерпретирующего SQL-команды ORACLE, как то: PL/SQL Developer, SQL\*Plus, а также утилиты из комплекта администратора SAUMI «Менеджер реквизитов», в которой вызывается окно выполнения PL/SQL-сценария (см. 8.4).

Пример команды для сервера БД Oracle, работающего под управлением операционной системы Windows.

```
CREATE OR REPLACE DIRECTORY AUDIT_ARH_PATH AS 'C:\SAUMI\AuditData';
```

Пример команды для сервера БД Oracle, работающего под управлением операционной системы UNIX/Linux.

```
CREATE OR REPLACE DIRECTORY AUDIT_ARH_PATH AS 'usr/SAUMI/AuditData';
```

### Замечание

Директория *имя\_директории* не создается при выполнении указанной выше команды и должна быть специально создана администратором в файловой системе серверного компьютера, на котором развернута БД SAUMI. Системная учетная запись, под которой запускается БД SAUMI, должна быть наделена правами создавать и записывать в данной директории файлы.

После задания описанной выше конфигурации ежемесячно 1-го числа сервером Oracle инициируется задание на обработку (job), которое производит выгрузку данных аудита, созданных более `AUDIT_SAVEMONTH` месяцев тому назад в xml-файлы, размещаемые в директории, указываемой объектом `AUDIT_ARH_PATH`. Для данных о системных событиях аудита имена файлов имеют шаблон `yyyy_mm_dd_SM.doc`, а для данных аудита сущностей SAUMI `yyyy_mm_dd_DC.doc`, где `yyyy` – год, `mm` – месяц, `dd` – день системной даты создания указанных файлов. Выгруженная в файлы информация удаляется из базы данных.



## Замечание

В случае отсутствия в БД SAUMI данных аудита, созданных более *AUDIT\_SAVEMONTH* месяцев тому назад выгрузка в xml-файлы не производится, т.е. пустые файлы не создаются.

# Глава 7. «Монитор подключений»

Утилита «Монитор подключений» отображает список, подключенных к Oracle пользователей. Этот список обновляется нажатием кнопки **Обновить**.

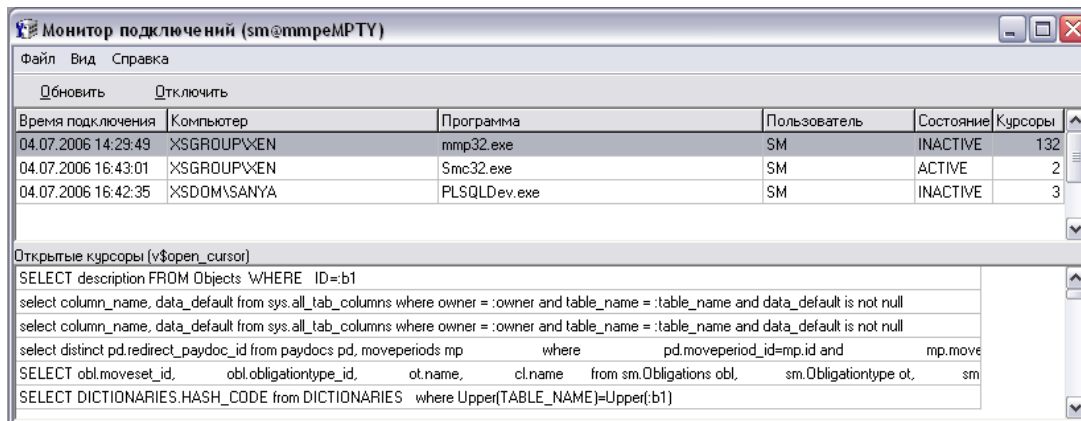


Рисунок 7.1.

---

# Глава 8. «Менеджер реквизитов»

## Содержание

8.1. «Менеджер реквизитов» – беглый взгляд .....	80
8.2. Термины и определения .....	81
8.3. Порядок создания отображаемого дополнительного реквизита .....	82
8.4. Создание поля, представляющего дополнительный реквизит .....	83
8.5. Формирование описания отображаемого дополнительного реквизита .....	86
8.5.1. Формирование описания в «Менеджере реквизитов» .....	87
8.5.2. Указание вкладки – ключ TAB .....	90
8.5.3. Поле ввода – ключ FIELD .....	91
8.5.4. Маска ввода для полей типа CHAR, VARCHAR, VARCHAR2 .....	92
8.5.5. Отрисовка значения реквизита – ключ RFIELD .....	94
8.5.6. Ссылка на экземпляр сущности – ключ LFIELD .....	95
8.5.7. Элемент управления кнопка – ключ BUTTON .....	97
8.5.8. Вывод надписи – ключ LABEL .....	97
8.5.9. Прорисовка рамки – ключ FRAME .....	98
8.5.10. Прорисовка линии – ключ LINE .....	98
8.5.11. Условие отображения реквизита на форме – ключ OPTION .....	98
8.6. Дополнительные настройки реквизитов .....	103
8.6.1. Настройка флажка .....	103
8.6.2. Ограничение ссылочного поля .....	104
8.6.3. Допустимость отсутствия значения и значение по умолчанию .....	104
8.6.4. Сохранение истории изменений значения реквизита .....	105
8.6.5. Учет реквизита в условии поиска .....	106
8.6.6. Создание списка значений реквизита .....	106
8.6.7. Подстановка синонимов в список значений реквизита .....	107
8.7. Восстановление подсистемы поиска после работы в «Менеджере реквизитов» .....	108
8.8. Перечень таблиц, принимающих отображаемые дополнительные реквизиты .....	108
8.9. Ограничения имен полей, представляющих дополнительные реквизиты .....	110

Как показывает практика различных муниципальных и региональных комитетов управления имуществом, по мере освоения SAUMI у пользователей возникает потребность работы с дополнительными реквизитами, которые должны отображаться в окнах свойств сущностей SAUMI и интерактивно изменять свои значения.

Комплект администратора SAUMI содержит служебную программу – утилиту «Менеджер реквизитов», позволяющую создавать дополнительные реквизиты, настраивать их интерактивный ввод и отображение в SAUMI. Кроме того, «Менеджер» позволяет выполнять как отдельные PL/SQL-команды, так и их наборы – PL/SQL-скрипты, которые зачастую присутствуют в пакетах обновлений, а также высылаются службой поддержки SAUMI на места для решения различных задач.

В качестве примера применения «Менеджера реквизитов» рассмотрим добавление для инженерных сооружений такого свойства, как высота, которая может иметь значимость при учете существенно высоких объектов, таких как вышки, мачты, мосты, высоковольтные линии электропередач и др.

Новый реквизит будет создан в таблице, представляющей свойства объектов инженерной инфраструктуры ENGINF. Это будет поле числового типа с тремя значащими цифрами после запятой, позволяющее представлять высоту объектов в метрах с точностью до миллиметра, тип – NUMBER(27,3), с именем HEIGHT.

## 8.1. «Менеджер реквизитов» – беглый взгляд

После вызова под учетной записью SM «Менеджер реквизитов» открывается его окно, в котором нас будет интересовать структура и описание таблицы ENGINF (рис. 8.1).

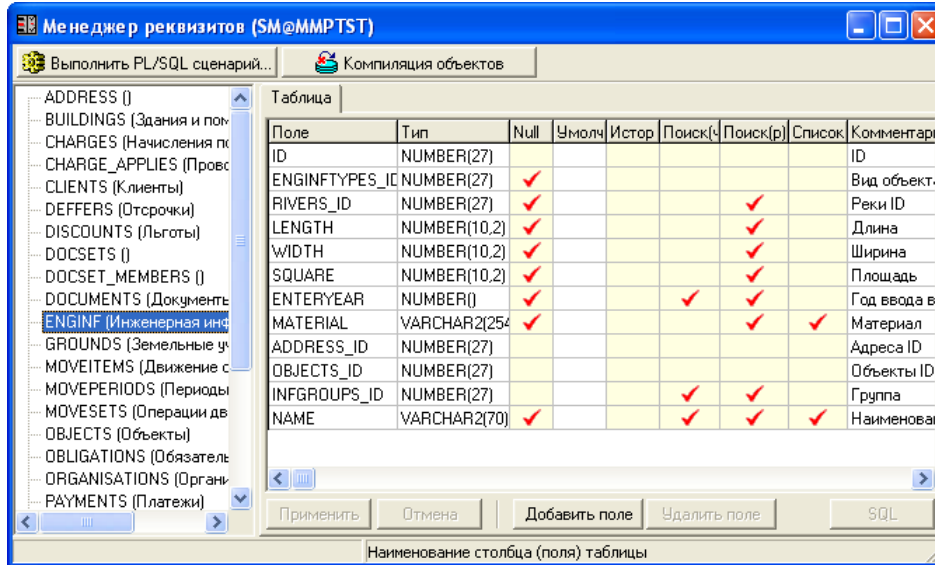


Рисунок 8.1.

Нажатие на кнопку **Добавить поле** (рис. 8.1) открывает диалоговое окно создания нового поля, в котором, для задания, рассматривавшегося во введении, свойства «высота», нужно ввести строку **HEIGHT** (рис. 8.2).

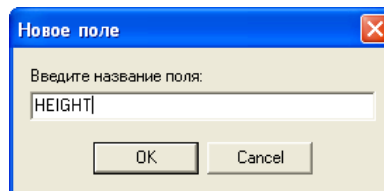


Рисунок 8.2.

После нажатия на кнопку **OK** (рис. 8.2) в правой панели менеджера появится строка свойств нового поля (рис. 8.3), предполагающая ввод типа данных.

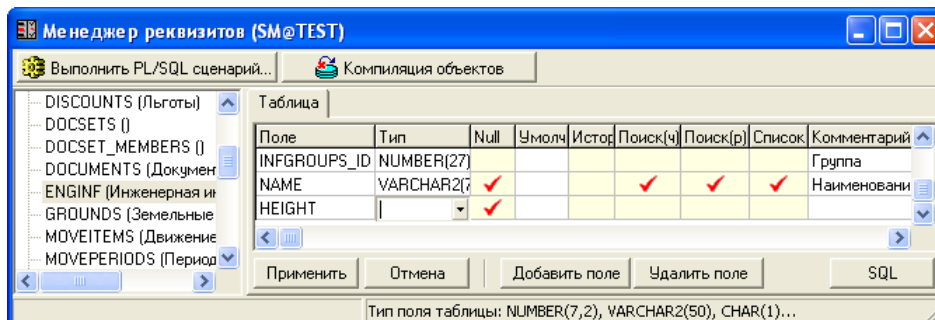


Рисунок 8.3.

Значение типа можно ввести двумя способами: набрать на клавиатуре нужную строку символов (рис. 8.4)

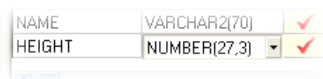


Рисунок 8.4.


или выбрать из открываемого нажатием на кнопку  (рис. 8.3) списка значение NUMBER(27) (рис. 8.5)



Рисунок 8.5.

с последующим изменением точности представления чисел данным полем (рис. 8.6).



Рисунок 8.6.

Как для создания поля, так и для первого знакомства с «Менеджером реквизитов», введенных данных достаточно, и, чтобы проделанная работа не пропала, нужно нажать кнопку **Применить** (рис. 8.3).

Для дальнейшего рассмотрения потребуется уточнить применяемую терминологию.

## 8.2. Термины и определения

### Базовый реквизит

Поле некоторой таблицы базы данных SAUMI, представляющее значение какого-либо свойства моделируемой в базе данных сущности и реализованное разработчиками SAUMI. Т.е. это поля, содержащиеся в поставляемой с SAUMI исходной базе данных, а также поля, появляющиеся в БД после установки пакетов обновлений. Например, для учёта площади земельного участка имеется поле SQUARE в таблице GROUNDS, для ведения государственных номеров транспортных средств поле FEDNO в таблице TRANSPORT и т.д.

### Дополнительный реквизит

Поле таблицы БД, представляющее значение какого-либо свойства учитываемого имущества, добавленное администратором SAUMI в дополнение к базовым реквизитам учета.

### Отображаемый дополнительный реквизит

Дополнительный реквизит учета, снабженный *описанием*, в каком виде этот реквизит должен отображаться в окне свойств сущности, к которой он отнесен.

### Замечание

*Отображаемый дополнительный реквизит учёта средствами SAUMI отображается в экранных формах ввода и его значение может быть задано интерактивно пользователем, тогда как просто*

*дополнительный реквизит, не снабженный таким описанием, свои значения должен получать другим способом, например, с помощью внешних программ.*

<b>Описание отображаемого дополнительного реквизита</b>	Специально оформленный фрагмент комментария к полю таблицы БД, заключенный в фигурные скобки {} и содержащий информацию о том, какие элементы графического пользовательского интерфейса будут применяться при отображении данного поля, каким образом это поле сможет получать свое значение посредством интерактивных действий.
<b>Блок описания</b>	Фрагмент описания дополнительного реквизита, включающий в себя <i>ключевое слово</i> и <i>параметры</i> . Один блок описания задает один определенный <i>элемент ввода-вывода</i> , сопоставляемый отображаемому дополнительному реквизиту.
<b>Ключевое слово (ключ)</b>	Слово из установленного набора слов, определяющих элементы ввода-вывода, сопоставляемые отображаемому дополнительному реквизиту.
<b>Параметры</b>	Фрагменты блока описания отображаемого дополнительного реквизита, определяющие внешний вид, размещение, координаты, размеры и прочие свойства элемента ввода-вывода, задаваемого ключом.
<b>Элемент ввода-вывода</b>	Применяемый элемент графического пользовательского интерфейса, обеспечивающий отрисовку, а также интерактивный ввод значения в соответствующем окне SAUMI.

Детальные аспекты описания отображаемого дополнительного реквизита приводятся в разделе «Формирование описания отображаемого дополнительного реквизита», а для начала уточним порядок его создания.

### **8.3. Порядок создания отображаемого дополнительного реквизита**

Для создания отображаемого дополнительного реквизита нужно выполнить следующие действия:

1. Определить таблицу, полем которой будет представляться значение нового реквизита, определить название и тип этого поля. Названия полей подвержены определенным ограничениям (см. 8.9).
2. Добавить новое поле в выбранную таблицу БД SAUMI.
3. Сформировать описание отображаемого дополнительного реквизита в комментарии к добавленному полю в структуре таблицы.
4. Произвести дополнительную настройку БД, влияющую на вид и поведение нового отображаемого дополнительного реквизита в соответствующем окне SAUMI.

#### **Замечание**

*Формирование описания по излагаемой схеме может быть применено и к базовым реквизитам (действия 3 – 4), оказывая на них такое же воздействие. Более того, многие базовые реквизиты изначально снабжены описаниями, определяющими способ их отображения и ввода в окнах свойств*

*сущностей. Таковыми, в частности, являются все адресные реквизиты объектов, расположение, размеры и порядок следования которых в отображении адреса может быть изменен в соответствии с потребностями на местах.*

Перечисленные действия 2 – 4 могут быть выполнены помимо рассматривавшейся утилиты «Менеджер реквизитов» (было показано исполнение действия 2) посредством таких утилит СУБД Oracle, как SQL\*Plus или PL/SQL Developer. По сути, за каждым приведенным выше действием стоит команда PL/SQL, которую в конечном итоге выполняет применяемое средство.

«Менеджер реквизитов» позволяет маскировать некоторые детали языка манипулирования данными (DDL), облегчая неопытным пользователям решение задачи. Тем не менее, «Менеджер реквизитов» также позволяет выполнять соответствующие DDL-команды, либо набранные в окне специального редактора, либо загруженные из файла с расширением .pdc. В последнем случае, окно редактора команд менеджера не позволяет вносить изменения в текст.

## 8.4. Создание поля, представляющего дополнительный реквизит

Поля, создаваемые для представления дополнительных реквизитов, могут быть следующих типов:

- NUMBER( $n$ [, $m$ ]) для числовых реквизитов; параметр точности  $m$  для целых чисел не применяется (скобки [ ] в нотации указывают на необязательность этого параметра); VARCHAR( $n$ ), VARCHAR2( $n$ ), CHAR( $n$ ) для строковых реквизитов длиной  $n$  символов;
- DATE для представления дат;
- CHAR(1) с ограничением CHECK IN ('Y','N') как булев реквизит;
- NUMBER(27) как поле, ссылающееся на другую таблицу – зарегистрированный в системе справочник; таким, например, является присутствующее в нескольких таблицах БД SAUMI ссылочное поле ADDRESS\_ID, ссылающееся на таблицу адресов ADDRESS.

### Замечание

*Для числовых полей, какова бы ни была запрашиваемая параметром  $n$  длина представления чисел, Oracle в любом случае отводит под такое поле 27 байт. В SAUMI принято соглашение по именованию полей, ссылающихся на другую таблицу, а именно: название такого поля следует составлять из имени таблицы, на которую производится ссылка, и суффикса "\_ID".*

Раздел «Менеджер реквизитов» – беглый взгляд демонстрирует создание нового поля в таблице диалоговыми средствами «Менеджера реквизитов». Тот же результат может быть получен выполнением команды PL/SQL непосредственно в «Менеджере реквизитов».

Рассматривавшееся свойство «высота» для объектов инженерной инфраструктуры задается командой

```
alter table ENGINF add HEIGHT number(27,3);
```

### Замечание

*Завершающий символ «;» - точка с запятой – является неотъемлемой частью команды PL/SQL. Команды PL/SQL могут быть набраны в верхнем, нижнем и смешанном регистрах. Имена полей в БД приводятся к верхнему регистру автоматически.*

Выполнение PL/SQL-команд будет рассмотрено на примере создания в таблице TRANSPORT поля с двумя значениями, которое будет отображаться в виде флажка. Значение 'Y' для этого поля будет интерпретироваться как наличие у транспортного средства газотопливной системы. Команды создания такого поля:

```
alter table TRANSPORT add GASEOUS_FUEL char(1) default 'N';  
comment on column TRANSPORT.GASEOUS_FUEL is  
  'Газотопливная система {ТАВ:"Общее" FIELD:5,260,150,0}';  
alter table TRANSPORT add constraint CK_TRANSPORT_GASEOUS  
  check (GASEOUS_FUEL in ('Y','N'));
```

Чтобы выполнить приведенные выше PL/SQL команды в «Менеджере реквизитов» следует нажать кнопку «Выполнить PL/SQL сценарий...» (рис. 8.1). В результате откроется соответствующее окно (рис. 8.7).

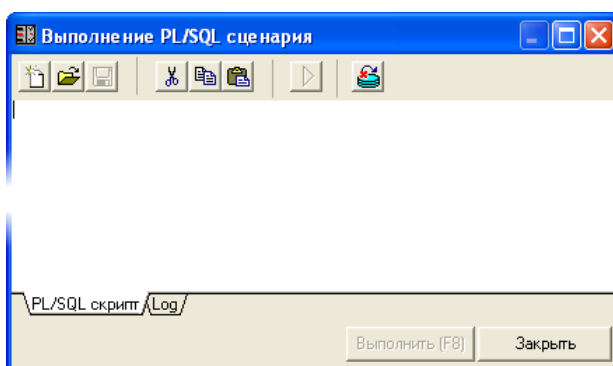


Рисунок 8.7.

На вкладке «PL/SQL скрипт» менеджер позволяет либо ввести текст команд с клавиатуры (скопировать через буфер обмена Clipboard) (рис. 8.8),

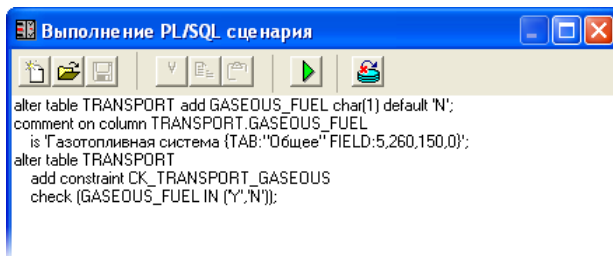



Рисунок 8.8.

либо, нажав кнопку  (рис. 8.7), в стандартном окне выбора файла (рис. 8.9) задать файл с расширением .pdc, содержащий требуемые команды PL/SQL.



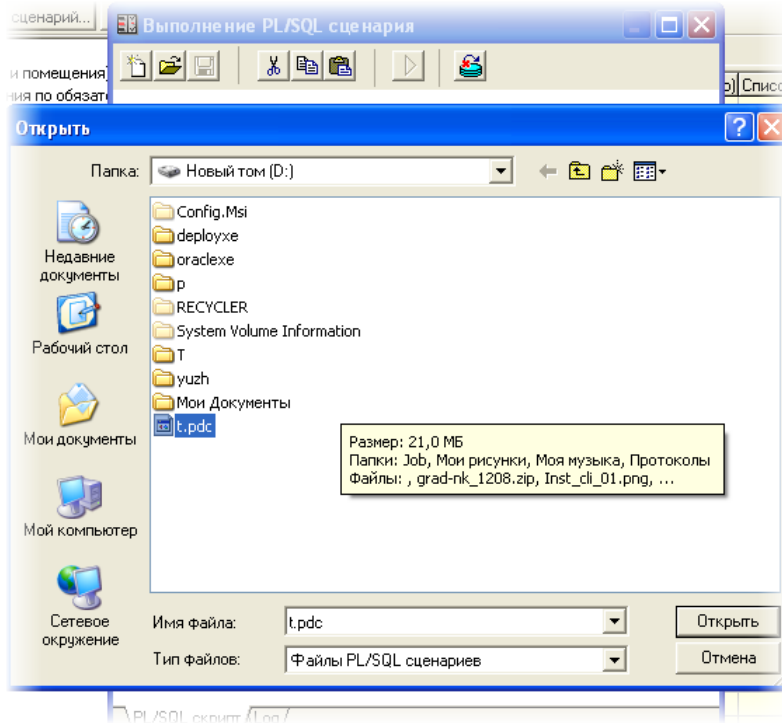



Рисунок 8.9.

Содержимое загруженного .pdc-файла в окне выглядит так же, как набранное с клавиатуры (рис. 8.8), однако изменить команды загруженного текста в этом окне нельзя – их можно только выполнить.

Для выполнения PL/SQL-команд нужно нажать кнопку  (рис. 8.8), в результате чего на вкладке «Log» отобразится протокол исполнения (рис. 8.10).

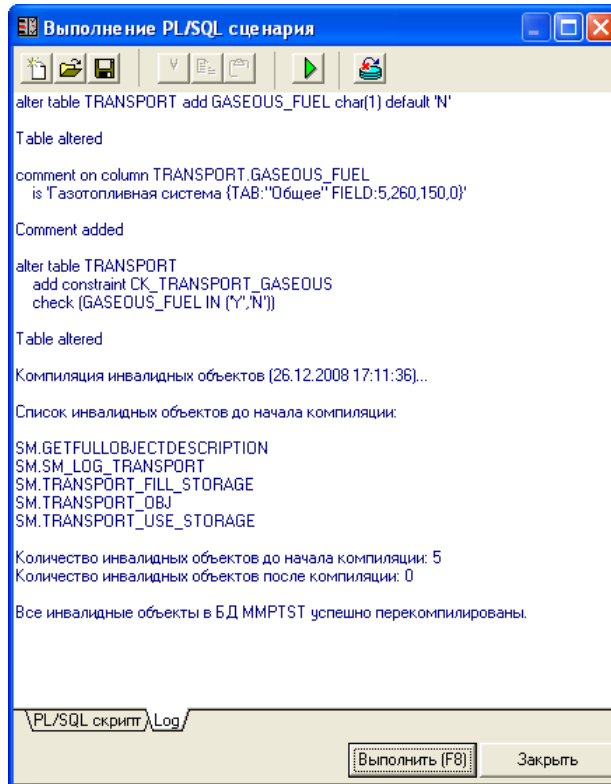


Рисунок 8.10.

Выбор в перечне левой панели менеджера таблицы TRANSPORT позволяет удостовериться в добавлении требовавшегося поля (рис. 8.11), которое по умолчанию будет соответствовать состоянию флажка «сброшен».

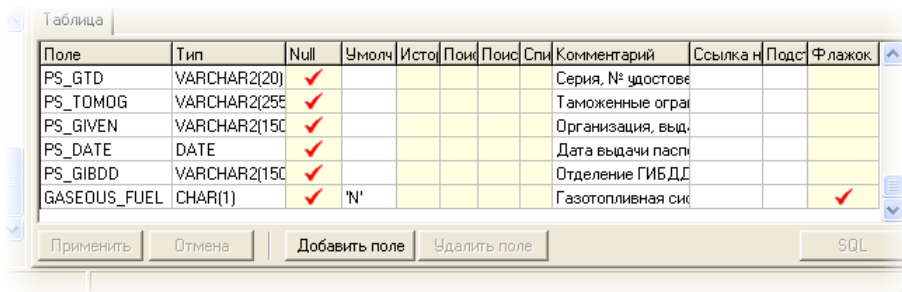


Рисунок 8.11.

На этом краткое наставление по использованию PL/SQL-команд, которые позволяют создавать дополнительные реквизиты, заканчивается.

## 8.5. Формирование описания отображаемого дополнительного реквизита

В общем виде описание отображаемого дополнительного реквизита выглядит следующим образом:

### ТекстКомментария {блок1 блок2 ... блокN}

где **ТекстКомментария** есть пояснение к назначению реквизита, а блоки описания **блок1 блок2 ... блокN** задают отображаемые элементы графического пользовательского интерфейса, связываемые с реквизитом.

### Замечание

*Для элемента управления «флажок», формируемого с помощью ключей FIELD и RFIELD, в отличие от всех остальных случаев, элемент **ТекстКомментария** используется как отображаемое в форме пояснение назначения флажка.*

Блоки, составляющие описание, отделяются один от другого пробелами. Между открывающей фигурной скобкой описания дополнительного реквизита и первым блоком не должно быть пробела, как и между последним блоком и закрывающей фигурной скобкой.

В SAUMI применяются следующие ключевые слова для описания отображаемых дополнительных реквизитов: TAB, FIELD, RFIELD, LFIELD, FRAME, LINE, BUTTON, LABEL, OPTION.

Ключи LABEL, FRAME, LINE, BUTTON могут входить в описание отображаемого дополнительного реквизита многократно. Все остальные ключи могут быть представлены только одним экземпляром. Ключи FIELD, RFIELD, LFIELD взаимно исключают друг друга.

Все слова, относящиеся к ключевым, должны быть набраны в верхнем регистре латинскими буквами. После ключевого слова должно стоять двоеточие. Сразу за двоеточием, без пробела, следует размещать параметры.

Параметры располагается в блоке после двоеточия, следующего за ключевым словом, и отделяются друг от друга запятыми.

В качестве параметров применяются координаты левого верхнего угла элемента ввода-вывода относительно левого верхнего угла доступной для отображения области окна свойств сущности, а также длина элемента ввода-вывода, его высота (всё в пикселях), текст надписи, название процедуры на языке SAUMI-Pascal, которая связывается с описываемым элементом ввода-вывода (например, кнопкой).

Прежде, чем приступить к изложению порядка применения различных ключей, следует отметить, что описание отображаемого дополнительного реквизита воспринимается по приводимым соглашениям и правилам только у строго определенного набора таблиц, представленного в разделе «Перечень таблиц, принимающих отображаемые дополнительные реквизиты».

### 8.5.1. Формирование описания в «Менеджере реквизитов»

«Менеджер реквизитов» предоставляет инструмент, облегчающий формирование описания отображаемого дополнительного реквизита. Созданному полю HEIGHT(см. 8.1) зададим описание реквизита, которое позволит вводить значение этого поля в форме свойств объекта инженерной инфраструктуры.

В окне «Менеджера реквизитов» следует в левой панели выбрать таблицу ENGINF, а в правой панели, в строке, соответствующей созданному полю HEIGHT, выделить ячейку в столбце "Комментарий" (рис. 8.12).

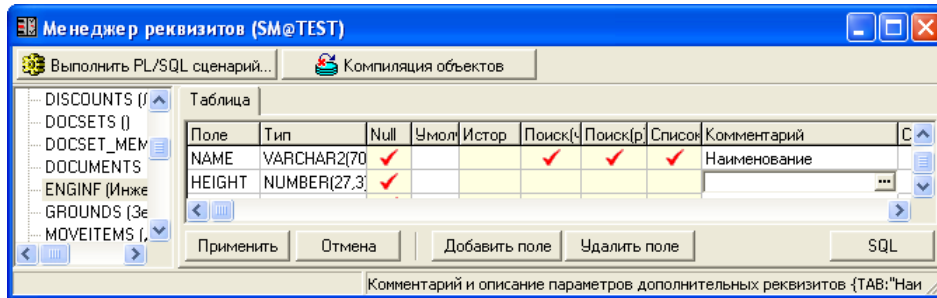



Рисунок 8.12.

После нажатия на появившуюся в выделенной ячейке кнопку  (рис. 8.12), открывается окно ввода комментария поля, в котором нужно предусмотрительно ввести фигурные скобки, а курсор установить между ними (рис. 8.13).

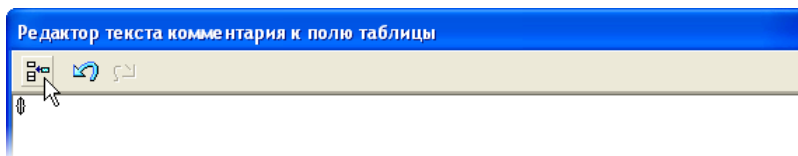



Рисунок 8.13.

В рассматриваемом примере поле HEIGHT предполагается представить как поле ввода на вкладке «Общее», снабженное пояснением: «Высота (м)». Для обеспечения указанного представления потребуются в описании отображаемого дополнительного реквизита применить три ключа: TAB, LABEL и FIELD.

В окне ввода комментария к полю (рис. 8.13) при нажатии на кнопку  откроется перечень применяемых в блоках описания ключей (рис. 8.14).

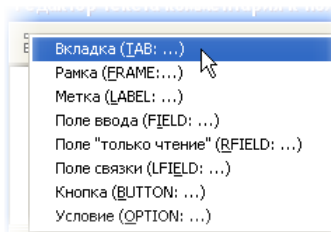


Рисунок 8.14.

После выбора элемента «Вкладка (TAB: ...)» в окне ввода отобразится выбранный ключ с подсказкой TAB: "Название вкладки", где вместо "Название вкладки" нужно ввести "Общее" и установить курсор непосредственно перед закрывающей фигурной скобкой (рис. 8.15).

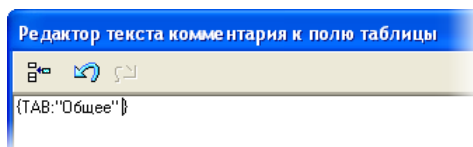


Рисунок 8.15.

Аналогично следует ввести ключи LABEL и FIELD с нужными параметрами (см. 8.5.8, 8.5.3), например, как на рис. 8.16.

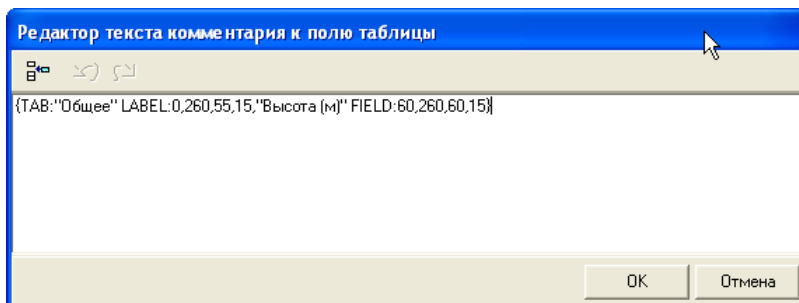


Рисунок 8.16.

Для завершения ввода следует нажать кнопку **ОК** (рис. 8.16), а затем, переместив фокус ввода из редактируемой ячейки *Комментарий*, нажать кнопку **Применить** (рис. 8.12). В результате исходный вид окна свойств объекта инженерной инфраструктуры (рис. 8.17)

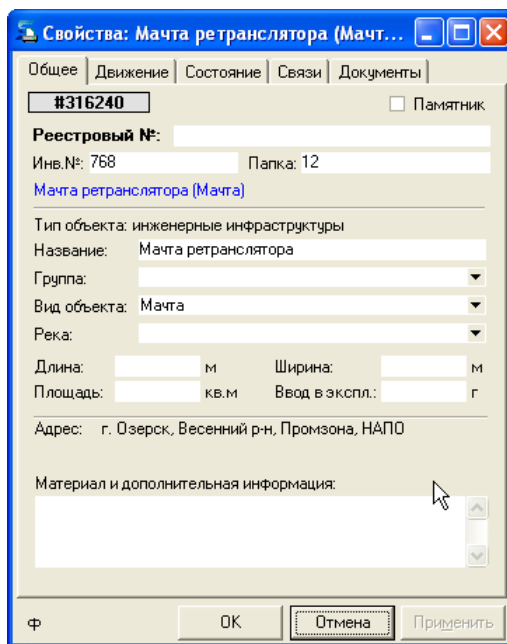


Рисунок 8.17.

изменится в нижней его части (рис. 8.18).

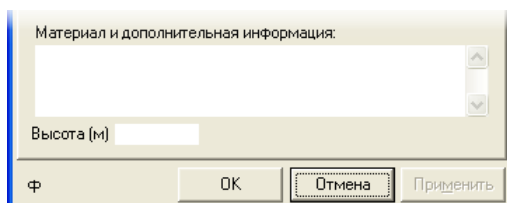


Рисунок 8.18.

Альтернативным способом создания описания отображаемого дополнительного реквизита является выполнение соответствующей SQL-команды (см. 8.4).

```
comment on column ENGINF.HEIGHT  
is '{TAB:"Общее" LABEL:0,260,55,15,"Высота (м)" FIELD:60,260,60,15}';
```

## Замечание

*Подбор координат и размеров отображаемых элементов является итеративным процессом и производится посредством сравнения получаемого изображения с желаемым.*

Далее детально рассматриваются применяемые в блоках описания отображаемых дополнительных реквизитов ключи и их параметры.

## 8.5.2. Указание вкладки – ключ TAB

Блок описания с ключом TAB имеет вид

```
TAB:"Наименование вкладки"
```

и указывает на размещение описываемых управляющих элементов ввода-вывода на вкладке с названием *Наименование вкладки*. Если такая вкладка не существует, то она будет создана.

Блок указания вкладки *должен* быть первым в описании реквизита. Рис. 8.19 иллюстрирует визуальное представление вкладки.

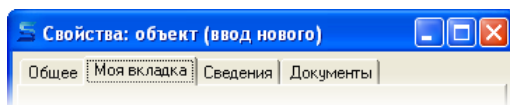


Рисунок 8.19. {TAB:"Моя вкладка"}

Допустимо ключ TAB не указывать. В этом случае при отображении дополнительного реквизита в окне свойств соответствующей сущности применяется текущая вкладка для отображаемых дополнительных реквизитов.

Исходной текущей вкладкой назначается вкладка с именем «Реквизиты», а каждый следующий отображаемый дополнительный реквизит, в описании которого указан ключ TAB, переопределяет текущую вкладку. Очередность обхода реквизитов соответствует нумерации полей во внутреннем представлении таблицы в базе данных, определяемой последовательностью их создания.

## Замечание

*Рекомендуется явно указывать вкладку, на которой будет отображаться дополнительный реквизит.*

Сформулированные выше правила для вкладок имеет ряд исключений.

1. Таблицы OBJECTUSING, MOVEITEMS, MOVESETS не допускают создание новых вкладок, и блок описания с ключом TAB в них применять не следует.
2. Отображаемые дополнительные реквизиты таблицы MOVEPERIODS напротив должны в своем описании явно указывать новую вкладку, отличную от вкладок, созданных разработчиками SAUMI для окна свойств сущности «движение».
3. Исходной текущей вкладкой для таблицы CHARGES является вкладка «Общее».

### 8.5.3. Поле ввода – ключ FIELD

Для осуществления ввода отображаемого дополнительного реквизита применяется блок описания вида

```
FIELD: x, y, w, h[, "маска"]
```

где  $(x, y)$  - пара чисел, задающих относительно окна свойств сущности горизонтальную и вертикальную координаты в пикселях левого верхнего угла отрисовываемого элемента, представляющего реквизит, а параметры  $w$  и  $h$  задают в пикселях ширину этого элемента и его высоту соответственно. Минимальные значения  $x, y$  равны нулю.

Необязательный (на это указывают квадратные скобки в приведенной нотации) параметр *"маска"* задает строку-шаблон ввода значения. Обрамляющие кавычки обязательны при задании этого параметра ключа FIELD. О порядке составления и назначении маски ввода см. 8.5.4.

#### Замечание

Маска ввода применяется только для реквизитов, представленных полями типов CHAR, VARCHAR, VARCHAR2.

Тип и ограничения поля таблицы (см. 8.4), хранящего значение реквизита, определяет применяемый компонент ввода:

- для строкового типа, а также числового типа, не являющегося ссылкой, применяется стандартный компонент ввода (рис. 8.20);

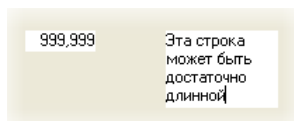


Рисунок 8.20. {FIELD:40,50,50,15} и {FIELD:140,50,80,55}

- для булева реквизита, т.е. для типа CHAR(1) с ограничением CHECK IN ('Y','N'), применяется элемент управления «флажок» (checkbox), выставленное состояние которого связывается SAUMI со значением 'Y', а сброшенное со значением 'N'; текст комментария, предшествующий открывающей фигурной скобке, отображается справа от изображения флажка (рис. 8.21); параметр  $h$  - «высота» - на отображение флажка влияние не оказывает и применяется для совместимости;



Рисунок 8.21. Газотопливная система {FIELD:5,260,150,0}

- для типа DATE применяется компонент выбора даты (datepicker) (рис. 8.22);

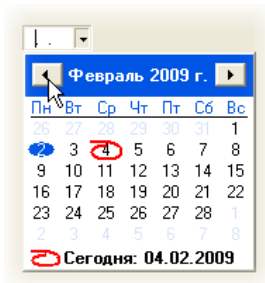


Рисунок 8.22. {FIELD:110,50,50,15}

- для целочисленного типа NUMBER(27), являющегося ссылкой на другую таблицу, применяется компонент работы со справочниками (рис. 8.23). Таблица, на которую направлена ссылка, должна соответствовать определению справочной таблицы, т.е. иметь поля ID, NAME, а также при необходимости и поле ROOT, и её описание должно присутствовать в таблице SM.DICTIONARIES.

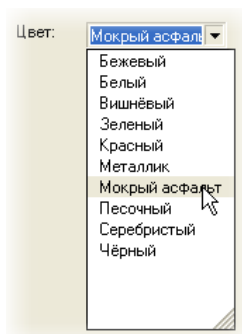


Рисунок 8.23. {FIELD:105,40,100,26 LABEL:55,40,50,12,"Цвет:"}

### 8.5.4. Маска ввода для полей типа CHAR, VARCHAR, VARCHAR2

Маска ввода является строкой символов и состоит из трех частей, разделяемых символом «;».

**Первая часть** является обязательной, она определяет собственно маску ввода. Каждому символу в поле ввода соответствует специальный символ в маске, определяющий, что можно ввести на этом месте.

Таблица 8.1. Описатели маски ввода

Символ маски	Правило восприятия позиции в поле ввода
L	должна быть буква (A-Z, a-z, A-Я, а-я)
I	может быть буква
A	должен быть буквенно-цифровой символ
a	может быть буквенно-цифровой символ
C	должен быть символ ASCII
c	может быть любой символ ASCII
0	должна быть цифра
9	может быть любая цифра



Символ маски	Правило восприятия позиции в поле ввода
#	может быть любая цифра или знаки '+', '-'

В приведенной таблице глагол «должен» означает, что пользователь обязан ввести в маскируемую позицию поля ввода соответствующий символ, тогда как «может» не устанавливает такого долга, а только разрешает ввод соответствующего символа. Например, маска **LL00** для ввода пароля обязывает пользователя ввести пароль из двух букв и двух цифр, а **LL99** – из двух букв и более двух цифр.

Кроме описателей полей ввода, в маске могут быть литералы и другие форматирующие символы.

**Литералами** называются символы, которые появляются при отображении строки ввода, но не могут изменяться пользователем и предназначены только для ее оформления. Перед литералом должен стоять символ «\». Типичный пример – ввод телефонных номеров. Для шестизначного номера с возможностью ввода междугородного кода нужен шаблон `\(9999\)000-000;1;_`. Тогда код города (если он есть) в отформатированном тексте будет заключен в скобки - **(3843)390-227**.

Таблица 8.2. Специальные символы

Символ	Назначение
\	символ, непосредственно предшествующий литералу
:	разделитель для часов, минут и секунд при вводе времени
/	разделитель для дней, месяцев и лет при вводе даты
;	разделитель полей в маске
!	символ определяет подавление пробелов: если он есть в маске (в любом месте), в выходном тексте подавляются пробелы перед текстом; в противном случае - идущие после текста
>	символ, после которого все вводимые буквы преобразуются к верхнему регистру
<	символ, после которого все вводимые буквы преобразуются к нижнему регистру
<>	после этой пары символов преобразование регистров отменяется
-	пустое поле при вводе (отображается как пробел). Курсор не будет останавливаться в этой позиции

**Вторая часть** маски ввода – это символ '0' или '1', определяющий, записываются ли литералы в обработанный текст. При '0' - литералы опускаются. По-умолчанию используется значение '1'.

Например, пользователь ввел цифры **1234567**. Тогда при действующей маске: `999\-99\-00;1;_` значение, которое будет сохранено в базе: **123-45-67**, а для маски: `999\-99\-00;0;_` в базу будет записано значение: **1234567**.

**Третья часть** содержит символ, который будет показываться в полях, предназначенных для ввода. Например, маска ввода номеров машин `л 000 лл;1;х` в поле ввода будет выглядеть как **х ххх хх**. По умолчанию этот символ '\_' (подчеркивание).

Вторая и третья части маски ввода могут отсутствовать. В этом случае будут использованы значения по умолчанию.

#### 8.5.4.1. Пример маски ввода для кадастрового номера

Упрощенный формат кадастрового номера земельного участка выглядит так:

**Нко : Нкр : Нкк : Нзу**

где

**Нко** – номер кадастрового округа;

**нкр** – номер кадастрового района в округе;

**нкк** – номер кадастрового квартала в районе;

**нзу** – номер земельного участка.

Поскольку первые два поля на данной территории, как правило, постоянны, для того чтобы облегчить ввод кадастрового номера, например, в г. Красноярске, можно задать такую маску ввода:

`\2\4\:\5\0\:99999\:9999;1;x`

здесь

**24** – номер кадастрового округа (Красноярский край);

**50** – номер кадастрового района в округе (Красноярск).

Пользователю остается только ввести девять цифр в места, помеченные 'x', вместо того, чтобы вводить шестнадцать символов.

При наборе на клавиатуре, например, следующей комбинации цифр

`014430001`

получим в поле ввода

`24:50:01443:0001`

в базе данных будет сохранено значение

`24:50:01443:0001`

## 8.5.5. Отрисовка значения реквизита – ключ RFIELD

Для отображения значения дополнительного реквизита на форме без возможности его изменить применяется блок описания вида

`RFIELD:x, y, w, h`

название которого происходит от "Read only FIELD" – поле только для чтения.

Параметры, следующие за ключом, имеют тот же смысл, что и у ключа FIELD (см. 8.5.3), определяя область отображения значения реквизита.

### Пример 8.1.

Имеется поле IN\_COURT типа CHAR(1) с ограничением IN\_COURT IN ('Y','N') с комментарием к полю

В суде {RFIELD:10,10,60,15}

В результате в окне свойств отобразится элемент управления "флажок" (checkbox) с надписью справа от него "В суде", а состояние флажка (установлен/сброшен) будет соответствовать текущему значению поля IN\_COURT, изменить которое посредством пользовательского интерфейса будет невозможно (рис. 8.24).



Рисунок 8.24.

## Пример 8.2.

Имеется поле IN\_COURT\_FROM типа DATE с комментарием

```
{LABEL:10,30,40,12,"Передано в суд:" RFIELD:90,30,75,12}
```

Отображаемое значение даты не может быть изменено (рис. 8.25).

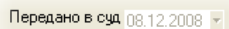


Рисунок 8.25.

## 8.5.6. Ссылка на экземпляр сущности – ключ LFIELD

Как показывает практика, заложенная изначально в БД SAUMI модель учета муниципального имущества может не вполне учитывать возникающие на местах событийно-логические связи между объектами, сущностями такого учета. Иногда может быть удобным проследить дополнительные связи между экземплярами некоторых сущностей.

В описываемой конструкции одна сущность и ее экземпляры называются главными. С помощью ключа LFIELD в описании дополнительного реквизита в окне свойств экземпляра главной сущности отображается строка описания заданного ссылкой экземпляра подчиненной сущности.

Связь устанавливается посредством присвоения дополнительному реквизиту главной сущности, представляющему ссылку, значения первичного ключа (значение поля ID) записи экземпляра подчиненной сущности, которая не обязательно должна отличаться от главной сущности – допустима принадлежность главного и подчиненного экземпляров к одной сущности. Содержательным примером к последнему замечанию может быть создание связей между земельными участками, полученными при делении большого участка (подчиненный экземпляр) на мелкие (главные экземпляры).

В силу вышесказанного, поле отображаемого дополнительного реквизита, устанавливающее описываемую связь, должно быть ссылочного типа (тип данных NUMBER(27)) с ограничением references SOME\_TABLE(ID), где под названием SOME\_TABLE следует понимать имя таблицы, представляющей подчиненную сущность, причем имя таблицы должно быть указано в таблице FM\_TABLES. Ссылочное ограничение устанавливается интерактивно с помощью «Менеджера реквизитов» (см. 8.6.2).

Блок описания с ключом LFIELD имеет вид

```
LFIELD: x, y, w, h
```

где параметры имеют то же назначение, что и у блока описания с ключом FIELD (см. 8.5.3). Определяемая ими область на вкладке формы предназначена для отображения стандартного описания экземпляра сущности, на который производится ссылка.

Изменение значения описываемой ссылки (изначально пустое) производится перетаскиванием мышью в отрисованную область элемента данных соответствующего типа (из окна данных, окна свойств, буфера обмена SAUMI и т.д.).

Щелчок мышью в отрисованной области открывает окно свойств связанного экземпляра сущности. Нажатие правой кнопки мыши в этой же области вызывает контекстное меню с двумя пунктами: «Свойства» и «Очистить», выбор первого из которых открывает окно свойств связанного экземпляра сущности, а выбор второго удаляет ссылку на связанный элемент и очищает область отображения.

Примером уместного использования ключа LFIELD в описании отображаемых дополнительных реквизитов можно считать связь объекта незавершенного строительства с этим же объектом после принятия его в эксплуатацию. С точки зрения SAUMI вторая сущность создается заново и никаких связей с предшествующим состоянием не содержит.

Для исправления такой ситуации, скажем, у объектов инженерной инфраструктуры следует создать в таблице ENGINF поле UNCONSTR\_ID, ссылающееся на таблицу UNCONSTR. Описание отображаемого дополнительного реквизита

```
{ТАВ:"Общее" LABEL:150,260,120,0,"В прошлом объект строительства" LFIELD:10,280,320,30}
```

создает область визуального представления связанного ссылкой объекта незавершенного строительства, изначально пустую (рис. 8.26).

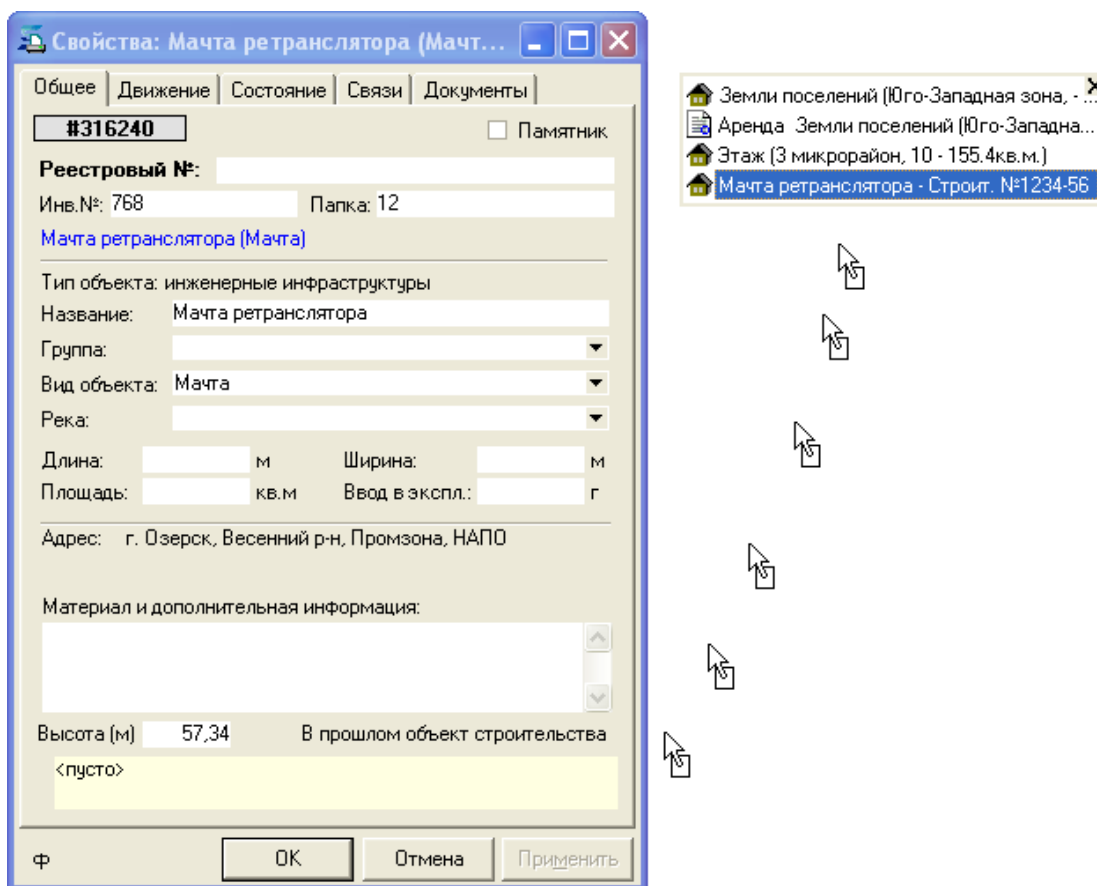


Рисунок 8.26.

Перетаскивание в отрисованную область предварительно сохраненного в буфере обмена SAUMI соответствующего объекта незавершенного строительства устанавливает нужную связь (рис. 8.27).

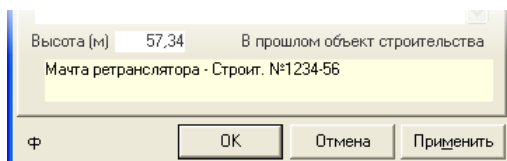


Рисунок 8.27.

## 8.5.7. Элемент управления кнопка – ключ BUTTON

Для добавления в форму ввода возможности интерактивно вызывать выполнение запрограммированной определенным образом функциональности применяется блок описания, имеющий вид

```
BUTTON: x, y, w, h, "Надпись", "SP_Procedure"
```

где числа  $x, y, w, h$  по аналогии с ключом FIELD (см. 8.5.3) определяют прямоугольную область в окне свойств, в пределах которой формируется графический образ элемента управления кнопка. Текст *Надпись* отображается на кнопке, а при ее нажатии вызывается SAUMI-Pascal процедура SP\_Procedure.

SAUMI-Pascal процедура, связанная с BUTTON должна быть описана в SP-редакторе с двумя целочисленными параметрами в следующем виде:

```
Procedure MY_SP_Procedure(Handle:integer; ItemID:integer);
begin
... //некоторые действия по нажатию кнопки
end;
```

Через параметр Handle в процедуру передается дескриптор окна (HWND) той формы, на которой расположена кнопка, а параметр ItemID передает ID экземпляра той сущности, в таблицу которой помещено описание BUTTON: например, если параметр BUTTON описан в комментарии к одному из полей таблицы BUILDINGS, тогда ItemID будет равен ID здания; если в таблице MOVESETS - ItemID будет равен ID операции движения и т.д.

### Замечание

*Сопоставляемая кнопке SAUMI-Pascal процедура должна быть уникальна в пределах всех дополнительных реквизитов, отображаемых на одной вкладке окна свойств сущности. В противном случае в момент открытия окна свойств выводится сообщение об ошибке, а также не отображается кнопка, SAUMI-Pascal процедура которой при компоновке отображаемых элементов вкладки обнаруживается повторно.*

## 8.5.8. Вывод надписи – ключ LABEL

Для вывода текста на форме применяется блок описания вида

```
LABEL: x, y, w, h, "Текст" [, "SP_Procedure"]
```

где числа  $(x, y)$  задают координаты левого верхнего угла прямоугольной области вкладки, где отображается текст *Текст*. Числа  $w, h$  никакого влияния на отображение текста не оказывают и применяются для совместимости. В основном, надписи применяются как пояснения для полей ввода (см. Пример 8.2).

Подстановка необязательного (на это указывают квадратные скобки в приведенной нотации) параметра - вызываемой SAUMI-Pascal процедуры SP\_Procedure - преобразует текст в элемент управления кнопку, отображаемую в виде гиперссылки (рис. 8.28).



Рисунок 8.28. { LABEL:30,40,60,20,"Рассчитать","My\_calc"}

При нажатии на такую кнопку происходит вызов указанной SAUMI-Pascal процедуры.

SAUMI-Pascal процедура, связанная с LABEL следует тем же соглашениям и ограничениям, что и SAUMI-Pascal процедура, связанная с BUTTON (см. 8.5.7).

### 8.5.9. Прорисовка рамки – ключ FRAME

Элемент оформления рамка задается блоком описания вида

FRAME:  $x, y, w, h$

где числа  $x, y, w, h$  по аналогии с ключом FIELD (см. 8.5.3) определяют прямоугольную область в окне данных, по границе которой прорисовывается рамка (рис. 8.29)..

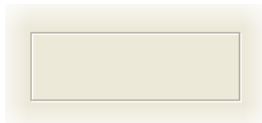


Рисунок 8.29.

### 8.5.10. Прорисовка линии – ключ LINE

Элемент оформления линия задается блоком описания вида

LINE:  $x, y, w, h$

где числа  $(x, y)$  устанавливают горизонтальную и вертикальную координаты начала линии в пикселях.

Прорисовка линии производится от начальной точки либо вниз по вертикали, либо вправо по горизонтали (наклонные линии не допускаются).

Для того чтобы прорисовка производилась по горизонтали, число  $w$  должно превосходить число  $h$ . В таком случае число  $w$  задает длину линии в пикселях.

Линия отображается вертикально, если число  $h$  не меньше, чем  $w$ . При этом число  $h$  задает длину линии в пикселях.

### 8.5.11. Условие отображения реквизита на форме – ключ OPTION

В некоторых случаях может возникнуть необходимость отображать или не отображать реквизит на форме в зависимости от значений некоторых реквизитов. В этом случае применяется блок описания

OPTION: (условие)

который производит визуализацию описываемого отображаемого дополнительного реквизита в зависимости от выполнения условия. Особенно часто такая необходимость возникает при описании дополнительных реквизитов операций движения, когда в операции движения одного вида реквизит используется, а в других не имеет смысла.

В качестве условия используется логическое выражение, составленное из сравнений значений полей таблицы. Допускаются операции сравнения:

$=, <>, >, <, >=, <=$  (по порядку: "равно", "не равно", "больше", "меньше", "не меньше", "не больше").

Поля, участвующие в сравнениях, могут быть только числовых типов, а также даты. Преобразующие функции не применяются. Для полей типа DATE возможно только сравнение двух полей этого

типа. Числовые поля допускают арифметические выражения с применением числовых констант. Допускается объединять несколько сравнений в одно, применяя связующие термы AND (логическое «и» - конъюнкция) и OR (логическое «или» - дизъюнкция), причем связываемые сравнения следует заключить в круглые скобки.

Числовые поля, значения которых не заданы, т.е. имеющие значение NULL, считаются равными нулю. Два поля типа DATE со значением NULL всегда равны. Если не задано в точности одно из них, то они всегда не равны, причем заданное поле всегда больше незаданного. Допускается использовать в арифметических выражениях операций сравнения разность значений двух полей типа DATE. Модуль такой разности равен количеству дней, разделяющих даты, а знак разности «-», если уменьшаемое строго меньше вычитаемого; в противоположном случае знак разности «+».

## Образцы условий

1. Условие однодневности некоторого промежутка времени.

```
OPTION: (DATE_FROM=DATE_TILL)
```

2. Условие превышения высоты некоторого уровня.

```
OPTION: (HIGHT>100)
```

3. Условие превышения расчетной суммы некоторого уровня.

```
OPTION: (PAY_RATE*(DATE_TILL-DATE_FROM+1)>100)
```

4. Проверка условия превышения расчетной суммы некоторого уровня за период времени не менее 30 дней.

```
OPTION: ((DATE_TILL-DATE_FROM>=29)  
AND (PAY_RATE*(DATE_TILL-DATE_FROM+1)>100))
```

### 8.5.11.1. Расширение синтаксиса блока описания OPTION для сущностей движения

Для таблицы MOVEITEMS в условиях блока описания OPTION можно задавать поля таблиц OBJECTS, MOVESETS, MOVEPERIODS, предваряя названия полей этих таблиц префиксами, состоящими из имени таблицы и символа точки.

Например, запись блока описания OPTION для отображаемого дополнительного реквизита таблицы MOVEITEMS в виде

```
OPTION: ((OBJECTS.OBJTYPES_ID<>2) AND (MOVESETS.MOVETYPE_ID=4))
```

означает, что описываемый реквизит будет отображаться только для операции движения "Аренда" (MOVETYPE\_ID=4), и для всех объектов, кроме транспорта (OBJTYPES\_ID<>2).

Для таблицы MOVEPERIODS в условии блока описания OPTION можно задавать поля таблицы MOVESETS.

Например, запись блока описания OPTION для отображаемого дополнительного реквизита таблицы MOVEPERIODS в виде

```
OPTION: (MOVESETS.MOVETYPE_ID<>4)
```

означает, что описываемый реквизит будет отображаться для любых операций движения, кроме "Аренда" (MOVETYPE\_ID<>4).

### 8.5.11.2. Расширение синтаксиса блока описания OPTION для использования площадей

Окно ввода и редактирования использования площадей объекта имущества, имеющего такую характеристику, как площадь («Земельные участки», «Здания и помещения») целиком и полностью реализуется на основе описываемого в настоящей главе механизма дополнительных реквизитов в применении к таблице OBJECTUSING. Это позволяет максимально гибко подстраивать вид этого окна под нужды и потребности пользователей, позволяя вводить новые характеристики использования площадей, непредусмотренные разработчиками.

#### Замечание

Представление ввода и отображения базовых реквизитов таблицы OBJECTUSING, реализуемых полями SQUARE и USINGPURPOSE с помощью механизма описания дополнительных реквизитов может быть изменено, но в любом случае эти реквизиты должны быть представлены для ввода и отображения.

При описании дополнительных реквизитов таблицы OBJECTUSING в секции OPTION можно использовать ссылки на значения полей таблиц OBJECTUSING, OBJECTS, MOVEITEMS, MOVEPERIODS, MOVESETS. При обращении к полям всех перечисленных таблиц, кроме OBJECTUSING, следует перед именем поля выставлять в качестве префикса имя таблицы, отделенное точкой, например: OBJECTS.OBJTYPES\_ID, MOVESETS.MOVETYPE\_ID.

Рассмотрим в качестве примера выделение в арендуемой площади мест общего пользования. Описываемая конструкция позволяет автоматически выделять из общей площади арендуемых объектов площадь, находящуюся в общем пользовании и, по большому счету, являющуюся обременением арендатора, за которое вполне справедливым может быть выведение данной площади из-под начисления арендной платы.

Для достижения описанной цели создается дополнительный реквизит, реализующий элемент управления «флажок» (см. 8.6.1, а также Замечание в 8.5).

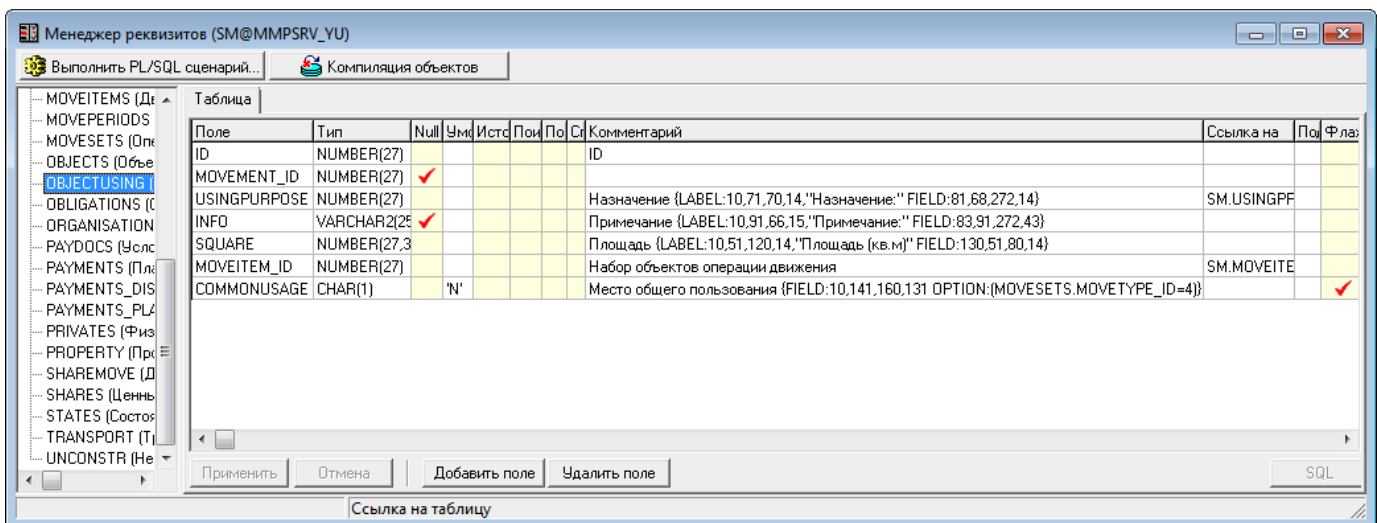


Рисунок 8.30.

Соответствующее поле COMMONUSAGE имеет тип CHAR(1), принимает значение по умолчанию 'N', то есть по умолчанию выделенная площадь не считается местом общего пользования. Создаваемый флажок присутствует только при выделении используемых площадей для операций



движения «Аренда», что обеспечивается ключом OPTION: (MOVESETS.MOVETYPE\_ID=4) в описании этого дополнительного реквизита.

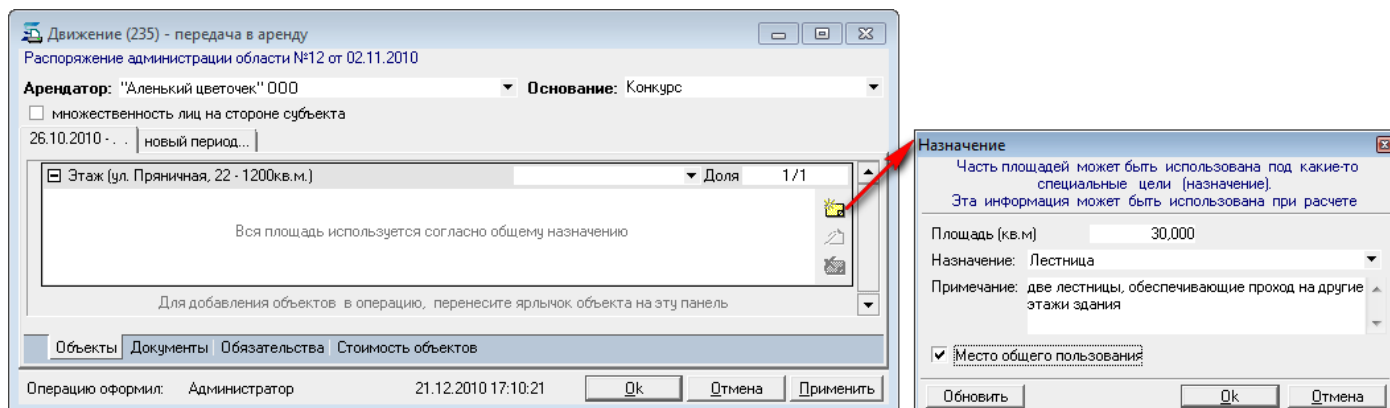


Рисунок 8.31. Выделение используемой площади для ОД «Аренда»

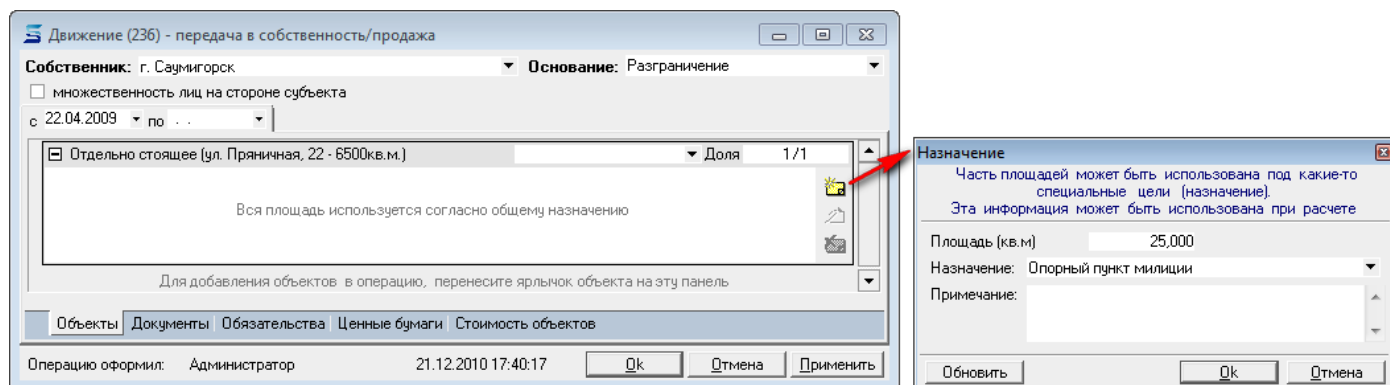


Рисунок 8.32. Выделение используемой площади для ОД «Собственность» – флажок отсутствует

### 8.5.11.3. Использование кратного описания дополнительного отображаемого реквизита

В описании дополнительных реквизитов допускается применение нескольких фрагментов, заключенных в фигурные скобки {}. Указанная возможность позволяет формировать вид формы ввода-вывода в зависимости от выполнения условий ключей OPTION, содержащихся во фрагментах описания дополнительного реквизита, заключенных в фигурные скобки. Ниже приводятся два примера использования кратного описания.

Первый пример показывает, как с помощью кратного описания для реквизита земельного участка *Кадастровый №*: производится отключение маски ввода для части земельного участка.

Свойство зарегистрированного в SAUMI участка быть частью некоторого объемлющего земельного участка определяется значением поля BASEGROUND в таблице GROUND. Для части земельного участка это поле содержит идентификатор объемлющего участка (значение поля ID некоторой другой записи таблицы GROUND). Значением поля BASEGROUND для объемлющего участка должно быть NULL. Тем самым нужное описание реквизита *Кадастровый №*: в таблице GROUND может иметь следующий вид.

Кадастровый номер

```
{TAB:"Общее" LABEL:4,30,105,13,"Кадастровый №:" FIELD:110,28,215,15,"\2\4\:\5\0\:99999\:9999;1;X"
OPTION:(BASEGROUND=0)}
{TAB:"Общее" LABEL:4,30,105,13,"Кадастровый №:" FIELD:110,28,215,15 OPTION:(BASEGROUND<>0)}
```

В качестве второго примера рассмотрим упрощенный учет частей земельного участка с применением схемы, при которой выделяемой площади устанавливается условный кадастровый номер, чье отображение и ввод организуется посредством дополнительного реквизита, представленного полем COND\_KAD\_NO таблицы OBJECTUSING, имеющего тип VARCHAR2(20). Условный кадастровый номер благодаря ключу

```
OPTION:(OBJECTS.OBJTYPES_ID=7)
```

в описании дополнительного реквизита будет отображаться, а также будет доступен для ввода, только при назначении использования площадей земельных участков (тип объекта 7).

Во многих случаях при невыполнении условия ключа OPTION в пространстве формы образуется пустое место, так как дополнительный реквизит в нем не отображается. Данный пример проиллюстрирует технику применения кратного описания дополнительных реквизитов с применением ключа OPTION, позволяющую формировать окно ввода и отображения без пропусков оконного пространства вместо отсутствующего по условию дополнительного реквизита. В демонстрационных целях условный кадастровый номер будет для земельных участков располагаться по вертикали между базовыми реквизитами «Назначение» и «Примечание» (рис. 8.32), а для остальных объектов имущества реквизит «Примечание» поднимется вверх.

Для заявленной цели подходят следующие описания для реквизитов «Условный кадастровый номер» и «Примечание», причем для последнего реквизита указаны две последовательности блоков описания, заключенных в фигурные скобки {} и содержащих ключи OPTION, условия которых взаимно исключают друг друга.

**Поле таблицы OBJECTUSING**

COND\_KAD\_NO

INFO

**Описание дополнительного реквизита**

Условный кадастровый номер  
 {LABEL:10,91,120,14,"Кадастровый  
 №" FIELD:130,91,120,14 OPTION:  
 (OBJECTS.OBJTYPES\_ID=7)}  
 Примечание {LABEL:10,91,66,15,"Примечание:"  
 FIELD:83,91,272,43 OPTION:  
 (OBJECTS.OBJTYPES\_ID<>7)}  
 {LABEL:10,111,66,15,"Примечание:" FIELD:83,111,272,43  
 OPTION:(OBJECTS.OBJTYPES\_ID=7)}

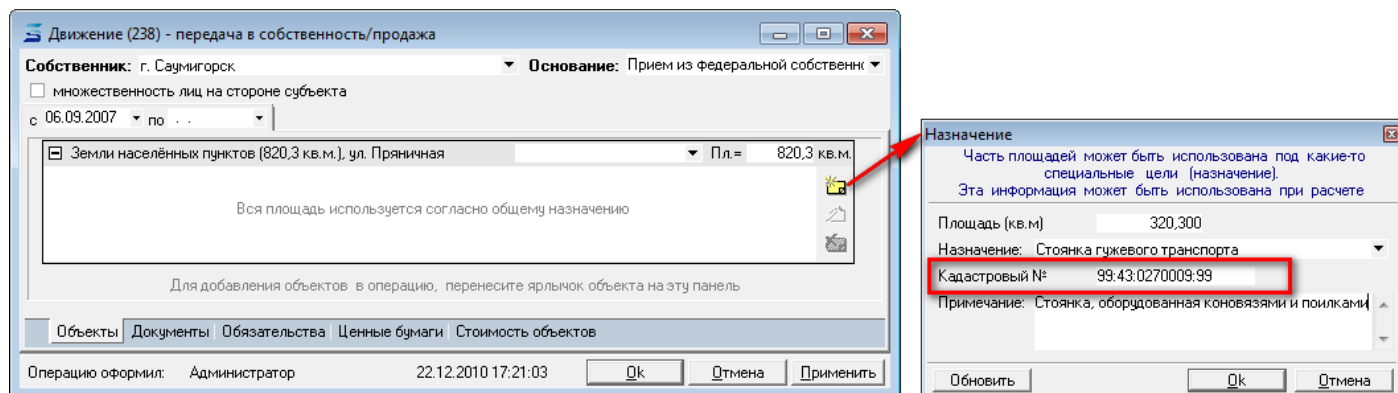


Рисунок 8.33. Дополнительный реквизит между двумя базовыми

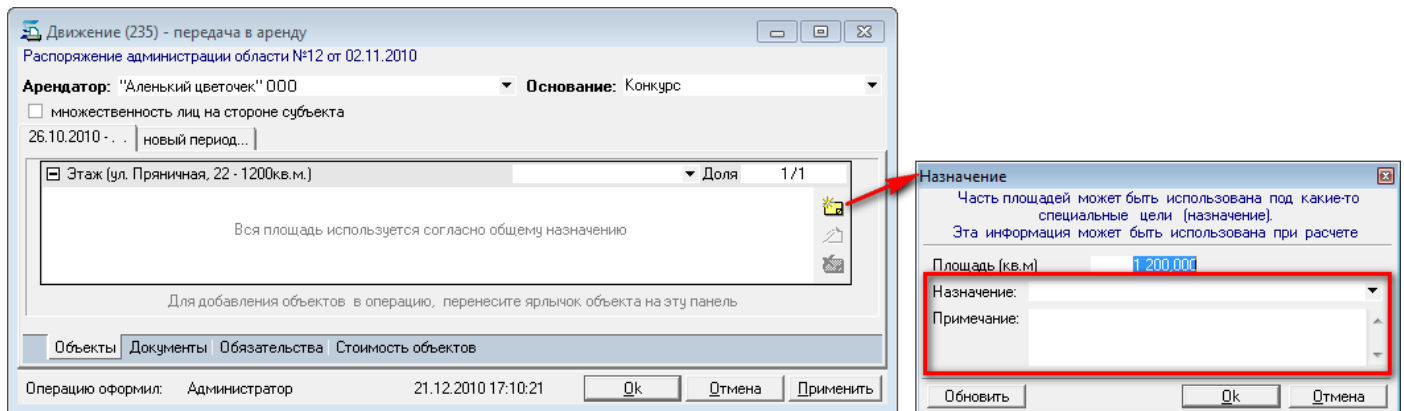


Рисунок 8.34. Дополнительный реквизит из-за условий не отображается, пропуск пространства в форме отсутствует

## 8.6. Дополнительные настройки реквизитов

Для корректного применения в SAUMI нового поля, созданного для представления отображаемого дополнительного реквизита, помимо описания требуется сделать дополнительную подстройку.

Указанное ранее действие 4 (см. 8.3) предполагает установление надлежащих ограничений для реквизитов, отображаемых в формах в виде флажка, а также для реквизитов, представленных полями ссылочного типа.

Вместе с тем, реквизиту, и не только отображаемому, могут быть предписаны следующие свойства: допустимость отсутствия значения, значение по умолчанию, сохранение истории изменения значений реквизита, применение в поиске, создание и поддержка списка всех введенных значений реквизита, замена в предыдущем списке применявшихся значений синонимами.

### 8.6.1. Настройка флажка

Дополнительный реквизит отображается на формах как флажок, если представляющее его поле имеет тип CHAR(1), в описании которого применен ключ FIELD или RFIELD (см. 8.5.3, 8.5.5), и которое может принимать только два значения: 'Y' и 'N' (см. также Замечание в 8.5).

Последнее ограничение задается с помощью «Менеджера реквизитов» следующим образом: нужно двойным щелчком в ячейке на пересечении строки, представляющей поле реквизита (например, поле GASEOUS\_FUEL), и столбца *Флажок* выставить флажок в виде символа ✓ (рис. 8.35).

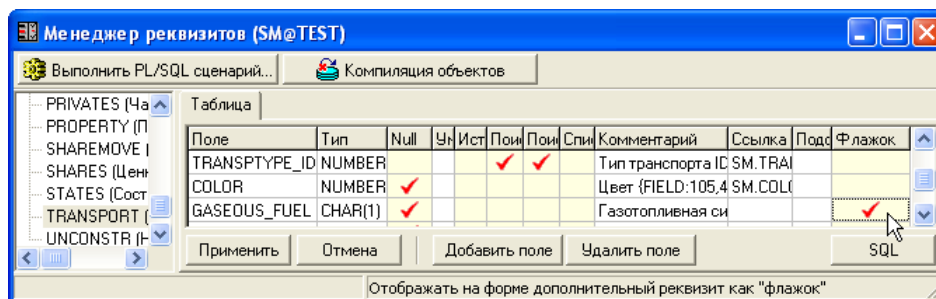


Рисунок 8.35.

Для сохранения введенного изменения следует нажать кнопку **Применить** (рис. 8.35).


SQL-команда, задающая рассматриваемое ограничение имеет вид

```
alter table TRANSPORT add constraint CK_TRANSPORT_GASEOUS
  check (GASEOUS_FUEL in ('Y','N'));
```

где задаваемое имя ограничения CK\_TRANSPORT\_GASEOUS должно быть уникальным в базе данных.

## 8.6.2. Ограничение ссылочного поля

Отображаемый дополнительный реквизит, представленный ссылкой на другую таблицу БД, должен быть снабжен соответствующим ограничением.

Для примера, рассматривавшегося ранее (см. 8.5.6), создадим ссылочное ограничение с помощью «Менеджера реквизитов». Поле UNCONSTR\_ID таблицы ENGINF должно ссылаться на таблицу UNCONSTR. Чтобы наложить такое ограничение нужно в окне «Менеджера реквизитов» активизировать ячейку на пересечении строки поля UNCONSTR\_ID и столбца Ссылка на (рис. 8.36) и выбрать из раскрывшегося нажатием на кнопку  списка элемент SM.UNCONSTR.

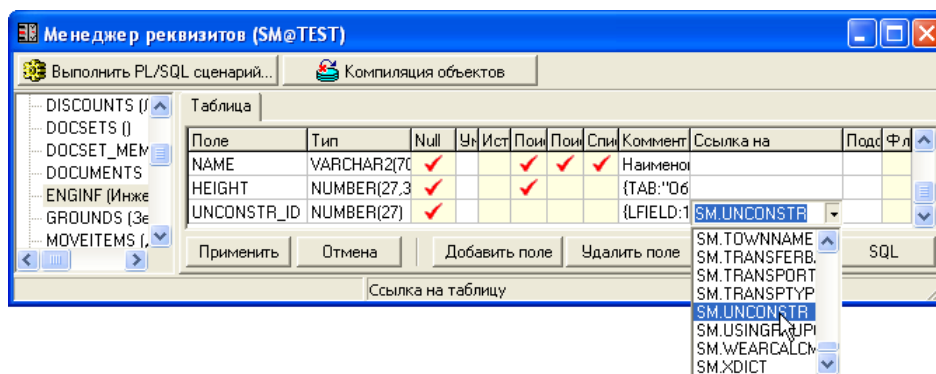


Рисунок 8.36.


Для сохранения введенного изменения следует нажать кнопку **Применить** (рис. 8.36).

## 8.6.3. Допустимость отсутствия значения и значение по умолчанию

Обычной практикой приложений баз данных является допустимость не вводить значения некоторых реквизитов, что облегчает работу пользователя. Отсутствие значения у поля, представляющего реквизит, помечается специальным значением, обозначаемым NULL.

В противоположность такому подходу, некоторые реквизиты должны иметь значение всегда, т.е. пользователь при вводе должен указать такому реквизиту некоторое значение. Тогда при сохранении новой записи в таблице БД осуществляется проверка, на присваивание значения и, в случае его отсутствия, запись не производится, а пользователю указывается на ошибку.

В некоторых случаях, когда отсутствие значения недопустимо, оказывается разумным присваивать реквизиту некоторое значение по умолчанию, которое пользователь может менять в форме ввода.

По умолчанию, если не оговорено обратное, создаваемое новое поле допускает отсутствие значения, т.е. получает значение по умолчанию NULL. Такое поведение поля, представляющего отображаемый дополнительный реквизит, отмечается в окне «Менеджера реквизитов» в столбце Null флажком  (рис. 8.37). Чтобы сделать значение NULL недопустимым, следует осуществить

двойной щелчок мышью в ячейке на пересечении указанного столбца и строки, описывающей нужное поле. Повторный двойной щелчок делает значение NULL снова допустимым.

В столбце *Умолч* можно задать значение по умолчанию. Для примера с газотопливной системой разумным будет предположить, что единица транспорта по умолчанию такой системы не имеет, чему соответствует значение 'N' поля GASEOUS\_FUEL (рис. 8.37).

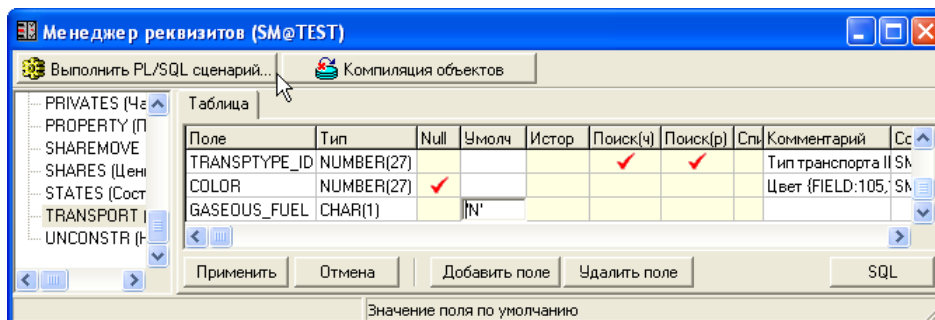


Рисунок 8.37.

Для сохранения введенного изменения следует нажать кнопку **Применить** (рис. 8.37).

#### 8.6.4. Сохранение истории изменений значения реквизита

При учете имущественных отношений ретроспектива во времени некоторых свойств объектов, субъектов, отношений и других сущностей учета может иметь значение. Таковым, например, является площадь земельного участка, которая может меняться при проведении повторных замеров. В результате вносимого изменения предыдущее значение площади участка сохраняется и доступно для просмотра.

Создаваемые дополнительные реквизиты также могут быть наделены свойством сохранять изменения их значений. Для этого в окне «Менеджер реквизитов» в столбце *Истор* нужно пометить ячейку флажком ✓ (рис. 8.38) в строке, представляющей реквизит. Указанный флажок выставляется (снимается) двойным щелчком мыши.

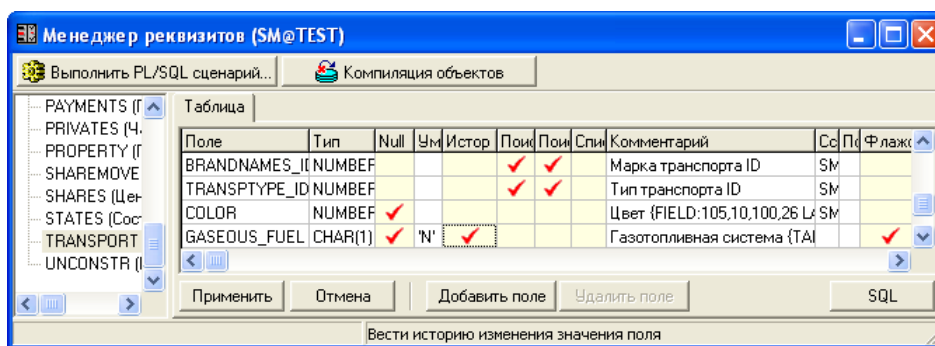


Рисунок 8.38.

Для сохранения введенного изменения следует нажать кнопку **Применить** (рис. 8.38).

Дополнительный реквизит, наделенный свойством сохранять историю изменений его значения, требует указания документа - основания производимых изменений. Для оборудования транспортного средства газотопливной системой таким документом может быть акт приемки ремонтных работ.

В случае появления у экземпляра сущности истории изменений значения реквизита, предыдущие значения отображаются на вкладке **История** окна ее свойств.

### 8.6.5. Учет реквизита в условии поиска

Реквизиты могут участвовать в задании условий как обычного (часто применяемого), так и расширенного поиска в БД SAUMI. Чтобы наделить некоторый реквизит таким свойством нужно в окне «Менеджера реквизитов» в столбце *Поиск (ч)* для обычного (частого) поиска, а в столбце *Поиск (р)* для расширенного поиска пометить соответствующие ячейки флажком  (рис. 8.39) в строке, представляющей отображаемый дополнительный реквизит. Указанный флажок выставляется (снимается) двойным щелчком мыши.

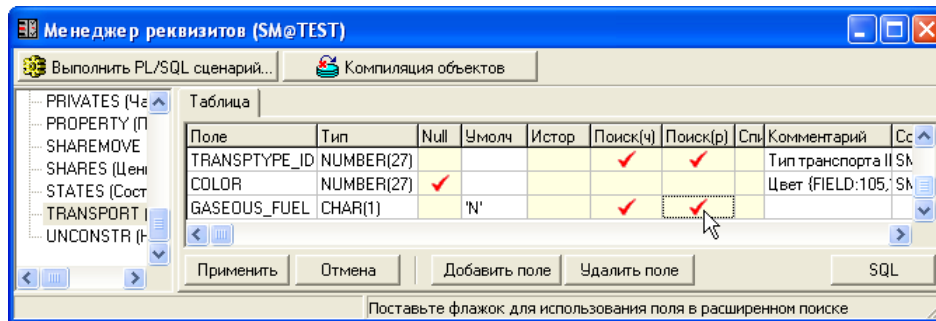


Рисунок 8.39.

Для сохранения введенного изменения следует нажать кнопку **Применить** (рис. 8.39).

### 8.6.6. Создание списка значений реквизита

При формировании условий поиска с участием реквизитов, представленных полями типов CHAR(n), VARCHAR2(n), VARCHAR(n), удобно использовать значения, выбираемые из списка (см. подраздел «Ограничения текстовых реквизитов» главы «Поиск данных в SAUMI» руководства пользователя).

Чтобы сформировать для реквизита такой список нужно в окне «Менеджера реквизитов» в столбце *Список* пометить ячейку флажком  (рис. 8.40) в строке, представляющей реквизит. Указанный флажок выставляется (снимается) двойным щелчком мыши.

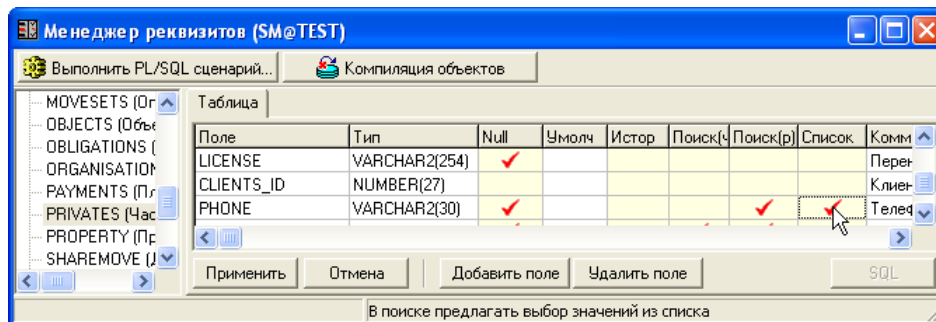


Рисунок 8.40.

Для сохранения введенного изменения следует нажать кнопку **Применить** (рис. 8.40).

## 8.6.7. Подстановка синонимов в список значений реквизита

Некоторые реквизиты, могут иметь значения, трудные для визуального восприятия. Созданный для поиска (см. 8.6.6) список значений такого реквизита будет также труден для восприятия пользователем, формирующим условия запроса. В некоторых случаях неудобочитаемые значения могут быть заменены в выстраиваемом списке значений на информативные синонимы.

### Замечание

*Описываемая возможность применима не столько к отображаемым дополнительным реквизитам, которые принимают значения в диалоге с пользователем, а скорее к некоторым базовым, а также дополнительным реквизитам, не отображаемым в диалоговых формах ввода-вывода. Несмотря на последнее обстоятельство, такие реквизиты могут участвовать в составлении условия запроса.*

Заменяющий синоним является значением, возвращаемым некоторой функцией БД Oracle, как встроенной, так и определенной пользователем.

В качестве примера рассмотрим поле CHARGE\_TYPE ( вид начисления) таблицы CHARGES - начисления по обязательству. Указанное поле получает значение программным способом и может иметь значения: 'FINE' и 'OBLIGATION'. В качестве синонимов приведенных значений для списка будут использоваться соответственно строки 'Пеня' и 'Обязательство'.

Чтобы осуществить указанную замену, нужно в окне «Менеджера реквизитов» выбрать таблицу CHARGES в левой панели, а в правой активизировать ячейку столбца Подстановка для списка в строке, соответствующей полю CHARGE\_TYPE (рис. 8.41).

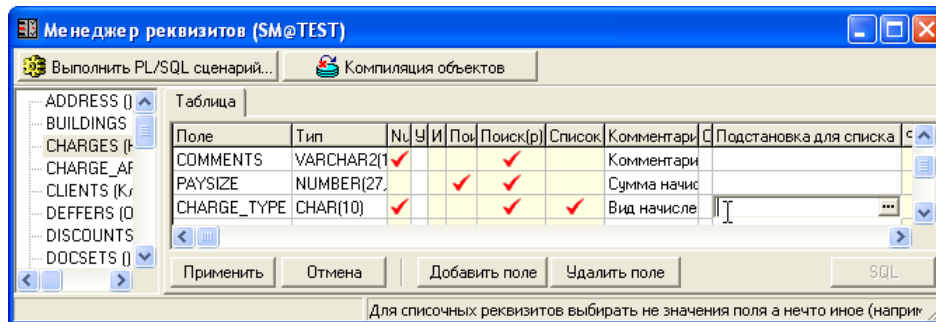



Рисунок 8.41.

После нажатия на появившуюся в выделенной ячейке кнопку  (рис. 8.41), открывается окно ввода (рис. 8.42), в котором нужно задать вызов функции БД Oracle, возвращающей нужные значения (рис. 8.42).

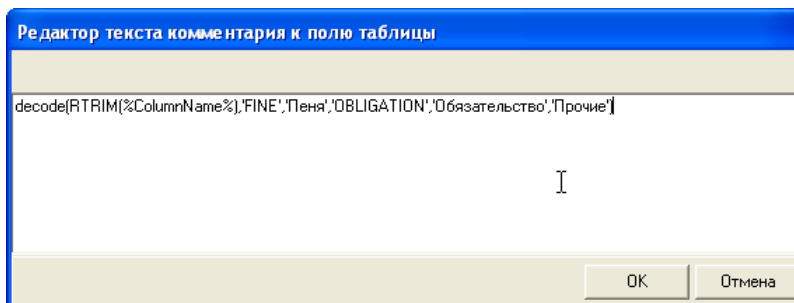


Рисунок 8.42.

Для завершения ввода следует нажать кнопку **ОК** (рис. 8.42), а затем, переместив фокус ввода из редактируемой ячейки, нажать кнопку **Применить** (рис. 8.41).

## 8.7. Восстановление подсистемы поиска после работы в «Менеджере реквизитов»

Изменения, вносимые в структуру БД посредством «Менеджера реквизитов» или другой утилиты, нарушают целостность механизмов поиска SAUMI и без надлежащей корректировки приводят к неправильным результатам поиска.

Для восстановления подсистемы поиска необходимо войти в пользовательское приложение SAUMI под учетной записью SM и открыть окно поиска (рис. 8.43), выбрав элемент «**Полный поиск**» в меню «**Поиск**».

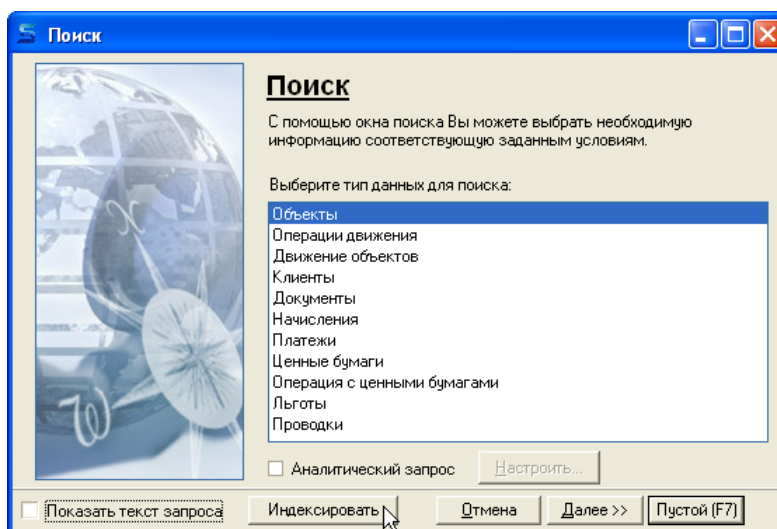


Рисунок 8.43.

В окне «Поиск» (рис. 8.43) нужно нажать кнопку **Индексировать**, в результате чего будет произведены необходимые действия по восстановлению механизма поиска SAUMI.

### Замечание

*Администратору предлагается придерживаться следующей практики: после каждого нажатия кнопки «Применить» в «Менеджере реквизитов» производить выше описанное восстановление подсистемы поиска.*

## 8.8. Перечень таблиц, принимающих отображаемые дополнительные реквизиты

Описание отображаемого дополнительного реквизита воспринимается по приводимым соглашениям и правилам только у строго определенного набора таблиц, сгруппированного по видам сущностей, перечень которых имеет ниже следующий вид.

### Объекты:

- OBJECTS – общие для всех видов объектов реквизиты;



- TRANSPORT – реквизиты транспортных средств;
- ENGINF – реквизиты объектов инженерных инфраструктур;
- BUILDINGS – реквизиты зданий (не помещений!);
- UNCONSTR – реквизиты объектов незавершенного строительства;
- PROPERTY – реквизиты прочих объектов имущества;
- GROUNDS – реквизиты земельных участков;
- таблицы новых видов объектов, созданных пользователями комплекса SAUMI;
- STATES – реквизиты состояний объектов;
- OBJECTUSING – реквизиты использования отдельных площадей земельных участков, а также зданий и помещений.

**Субъекты права:**

- CLIENTS – общие для всех типов субъектов права реквизиты;
- PRIVATES – реквизиты физических лиц;
- ORGANISATIONS – реквизиты юридических лиц.

**Ценные бумаги:**

- SHARES – реквизиты ценных бумаг;
- SHAREMOVE – реквизиты операций с ценными бумагами.

**Документы:**

- DOCUMENTS – реквизиты учета единичных документов.

**Операции движения:**

- MOVESETS – общие реквизиты движения объектов в рамках одного договора;
- MOVEPERIODS – реквизиты в рамках одного договора одного периода, в течение которого условия операции движения остаются неизменными;
- MOVEITEMS – реквизиты одного объекта в течении одного периода, указанного в таблице MOVEPERIODS.

**Финансовый учет:**

- CHARGES – реквизиты начислений;
- PAYMENTS – реквизиты учета платежей;
- DISCOUNTS – реквизиты учета льгот по платежам;
- DEFFERS – реквизиты отсрочек по платежам.

**Адресная информация:**

- ADDRESS – реквизиты учета адресов.

### Замечание

*Все приведенные выше таблицы перечислены в таблице FM\_TABLES, но не каждая таблица, указанная в FM\_TABLES, принимает отображаемые дополнительные реквизиты.*

## 8.9. Ограничения имен полей, представляющих дополнительные реквизиты

Создаваемые дополнительные реквизиты, помимо использования их в формах ввода, в большинстве случаев применяются в различных расчетах, обеспечиваемых программным кодом на языке SAUMI-Pascal. Имена полей, задействованных в программах SAUMI-Pascal, используются как идентификаторы и, тем самым, подвергаются ряду ограничений.

В названиях полей не допускается использование символов кириллицы.

### Внимание

*Особо следует внимательно использовать символ 'С' – это может быть только латинская 'Цэ' – кириллическая 'Эс' НЕДОПУСТИМА!*

Следует избегать дублирования названия полей. Например, очень распространено название поля NAME. Лучше модифицировать такое имя, например, как GR\_NAME для таблицы GROUNDS.

Имена полей не должны совпадать со служебными словами SAUMI-Pascal, представленными в ниже следующем списке.

AND	ELSE	NOT	SHL
ARRAY	END	OF	SHR
AS	EXCEPT	OR	THEN
BEGIN	EXIT	OVERRIDE	TO
BREAK	EXTERNAL	PRIVATE	TRY
CASE	FINALLY	PROCEDURE	TYPE
CLASS	FOR	PROGRAM	UNIT
CONST	FORWARD	PROPERTY	UNTIL
CONSTRUCTOR	FUNCTION	PROTECTED	USES
CONTINUE	IF	PUBLIC	VAR
DESTRUCTOR	IN	PUBLISHED	VIRTUAL
DIV	INHERITED	RECORD	WHILE
DO	IS	REPEAT	WITH
DOWNT0	MOD	SET	XOR

# Глава 9. «Менеджер отчетов»

## Содержание

9.1. Составление SQL-запроса .....	111
9.2. Разложение SQL-запроса на составляющие .....	112
9.3. Добавление активности (Свойства) PRIMARY .....	114
9.4. Ссылка на связанные данные .....	114
9.5. Видимость столбцов .....	115
9.6. Дополнительная специальная информация по столбцам .....	115
9.7. Флаг суммируемости .....	116
9.8. Специальные колонки (псевдостолбцы) .....	116
9.8.1. Описание общих секций FROM и WHERE .....	116
9.8.2. PRIMARY – Главный ключ .....	116
9.8.3. RESTRICTION – Ограничение .....	116
9.8.4. ORDER BY – Принудительная сортировка .....	117
9.8.5. DICT – Группировочное поле аналитического отчета .....	117
9.8.6. DATE – Группировочное поле даты аналитического отчета .....	117
9.8.7. INDICATOR – Выделение строк цветом и стилем .....	118

Отчет – это результат выполнения запроса пользователя по выборке из БД SAUMI на основе некоторых условий определенного массива информации. Результат выборки отображается в окне данных.

Реестр	Описание	Назначение	Собственник (те...	Балансодержатель (т...
14507	Земли населённых пунктов (1711,59 кв.м.) кад. №:42:30:01 13 013:0135...			
14508	Земли населённых пунктов (1018,9 кв.м.) кад. №:42:30:01 13 013:0136,...			
14509	Земли населённых пунктов (738,99 кв.м.) кад. №:42:30:01 13 013:0137,...			
14510	Земли населённых пунктов (972,27 кв.м.) кад. №:42:30:01 13 013:0138,...			
14511	Земли населённых пунктов (964,47 кв.м.) кад. №:42:30:01 13 013:0139,...			
10320	ДИЗО г. Саунигорска Орган самоуправления		МО "Город Сауниг...	
10521	ОФК по МО "Город Саунигорск" Некоммерческая организация			
13804	Отдельстоящее (ул.Кирова, 71)	Административное	МО "Город Сауниг...	Администрация МО "Го.
15628	помещение 6 (пр-кт.Металлургов, 34 - 213,19кв.м.)		МО "Город Сауниг...	
15630	Цокольный этаж (пр-кт.Металлургов, 29 - 791,39кв.м.)		МО "Город Сауниг...	
15632	помещение 1 (пр-кт.Металлургов, 29 - 395,69кв.м.)		МО "Город Сауниг...	

**Итоговые суммы**  
Остаточная стоимость: 60 642 193.71 руб.  
Первонач (восст.) стоимость: 119 321 042.00 руб.

Всего: 8339. Выполнено за 5,9 сек.

Рисунок 9.1. Отчет «Объекты»

Имеющиеся в поставке SAUMI отчеты можно модифицировать, а также создавать свои собственные отчеты. Модификация имеющихся и создание новых отчетов производится посредством утилиты из комплекта администратора SAUMI «Менеджер отчетов».

## 9.1. Составление SQL-запроса

Основой отчета является SQL-запрос к БД SAUMI. В качестве примера можно рассмотреть запрос данных по движению объектов.

```
select
```

```

sm.getobjectdescription(objects.id),
  MoveType.Name,
  Movements.PostingDate,
  Clients.Name
From
  sm.Movements,
  sm.Objects,
  sm.MoveType,
  sm.Clients
where
  Objects.ID=Movements.Objects_ID and
  MoveType.ID=Movements.MoveType_ID and
  Clients.ID=Movements.CLIENTSTO_ID
    
```

Результат приведенного запроса является списком операций движения и имеет следующий вид.

Отдельностоящее (Кирова, 71)	Балансодержание	02-05-1999	Профи ООО
Отдельностоящее (Кирова, 72)	Балансодержание	12-05-1999	Литейщик
Отдельностоящее (Кирова, 73)	Балансодержание	17-05-1999	УралСталь
Отдельностоящее (Кирова, 74)	Балансодержание	22-05-1999	Премьер
Встроенное (Кирова, 71)	Аренда	03-06-1999	Кузин А.Н.

В ряде случаев полученный список может устраивать пользователя, но найдется достаточно и других случаев, когда пользователю потребуются дополнительные столбцы, которые не представлены в этом запросе или, напротив, некоторые столбцы могут оказаться лишними. То есть, необходима технология, которая позволит формировать SQL запрос в зависимости от потребностей пользователя. Для этого служит Менеджер отчетов.

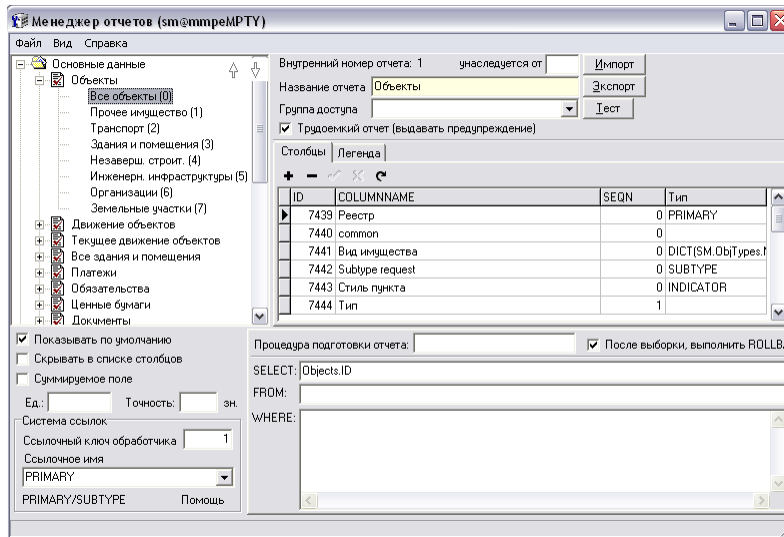


Рисунок 9.2. Менеджер отчетов

## 9.2. Разложение SQL-запроса на составляющие

Основу отчёта составляют столбцы таблицы отчёта, представляющие значения некоторых характеристик запрашиваемых информационных объектов. Состав столбцов отчёта определяется SQL-предложением, которое состоит обычно из трех секций: SELECT, FROM и WHERE. В секции SELECT перечисляются поля таблиц, данные из которых представляются в списке. Секция FROM содержит перечень таблиц и представлений, задействованных в запросе. Секция WHERE содержит ограничения выборки и описание связей между таблицами.

Присутствие в отчёте каждого столбца отражается в указанных выше трех секциях (рис. 9.2). Обязательно добавляется текст в секцию SELECT, отражающий собственно поле выборки. В поле FROM добавляется название таблицы, если оно там еще не было указано. В поле WHERE в случае необходимости добавляются ограничения и связи между таблицами.

Допустим, на примере предыдущего запроса требуется получить список с единственным столбцом – датой начала движения объектов. Такой запрос будет выглядеть следующим образом:

```
select
  Movements.PostingDate
from
  sm.Movements
```

Так как этот запрос описывает только один столбец, раскладывать его на составляющие фактически не требуется. Содержание каждой из секций представляется в виде соответствующей строки и в дополнение вводится название столбца.

```
Название: Дата начала движения
SELECT:   Movements.PostingDate
FROM:     sm.Movements
WHERE:    <не указано>
```

Далее опишем столбец списка, содержащий названия объектов. Для этого потребуется добавить в секцию SELECT поле Objects.Description, в котором хранится описание объектов. В секцию FROM таблицу объектов sm.Objects и в секции WHERE связать таблицу sm.Objects с представлением sm.Movements: Objects.ID=Movements.Objects\_ID. Теперь представим все это в виде структуры описывающей столбец.

```
Название: Объект
SELECT:   Objects.Description
FROM:     sm.Objects
WHERE:    Objects.ID=Movements.Objects_ID
```

Подобным образом можно описать все варианты столбцов, которые могут понадобиться пользователям. Ядро Inflex обращается к этим структурам для того, чтобы составить текст запроса, собрав воедино содержание каждой из трех секций. При этом чтобы все повторения в секции FROM автоматически удалялись, необходимо в качестве разделителя использовать символ '|' вместо символа запятой ', '.

Далее, для полноты понимая, приводятся структуры, описывающие остальные столбцы вышеприведенного запроса.

```
Название: Вид движения
SELECT:   MoveType.Name
FROM:     sm.Movements|sm.MoveType
WHERE:    MoveType.ID=Movements.MoveType_ID
```

```
Название: Адресат
SELECT:   Clients.Name
FROM:     sm.Movements|sm.Clients
WHERE:    Clients.ID=Movements.Clients_ID
```

Некоторые запросы могут содержать общие для всех отображаемых столбцов секции. Так, например, из вышеприведенного запроса общими для всех столбцов является запись sm.Movements секции FROM. Для таких случаев вводятся псевдостолбцы, описывающие общие секции FROM и WHERE.

```
Название: не важно
SELECT:   <не указано>
FROM:     sm.Movements
WHERE:    <не указано>
```

Именно тот факт, что секция SELECT остается незаполненной, имеет значение для ядра, чтобы понять, что это – псевдостолбец, описывающий общие секции.

### 9.3. Добавление активности (Свойства) PRIMARY

Разложение запроса позволяет предоставить набор столбцов для самых разных потребностей. Также существенной является возможность обращаться к данным, представляющим одну строку отчёта, то есть простая операция пользовательского интерфейса, например – двойной щелчок мыши на строке списка, должна отобразить окно свойств соответствующего экземпляра информационной сущности.

Для реализации представленного перехода от списка к окну свойств необходимо добавить в запрос поле, в котором будет содержаться число-идентификатор экземпляра данных, соответствующего этой строке. Так как любой запрос раскладывается на столбцы, то и это поле с идентификатором необходимо представить в виде столбца. Ядро узнает о том, что именно это поле несет информацию об идентификаторе экземпляра по специальному ссылочному имени « PRIMARY », которое необходимо указать для этого столбца.

```

Название: Идентификатор
SELECT:   Movements.ID
FROM:     sm.Movements
WHERE:    <не указано>
Ссылочное имя: PRIMARY
Ссылочный ключ обработчика: 25

```

Как видно в представленном примере, в описании столбцов добавлены два новых параметра – ссылочное имя и ссылочный ключ обработчика. Именно информация из этих двух полей, описывающих столбец, используется ядром, чтобы понять, что строка списка соответствует экземпляру данных с идентификатором равным значению этого столбца. Число 25 соответствует типу «Движение объекта».

Теперь при двойном нажатии пользователя на строчке в списке данных, будет открыто окно свойств.

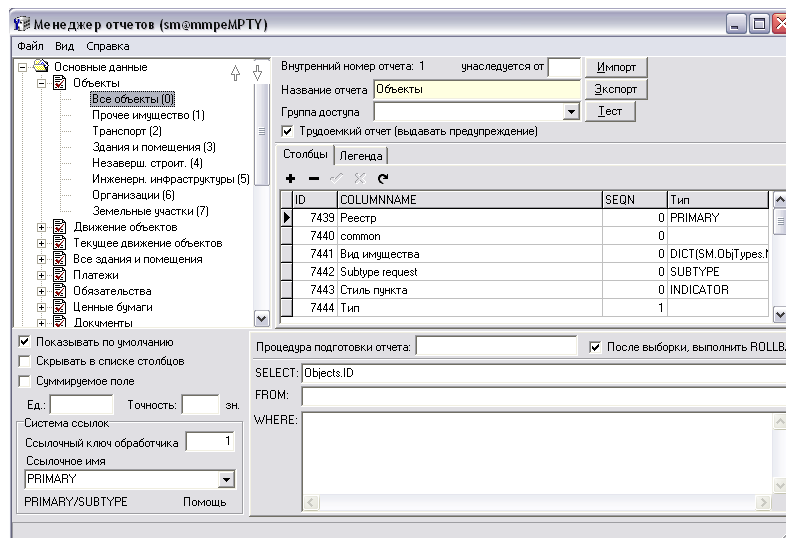


Рисунок 9.3. Добавление активности

### 9.4. Ссылка на связанные данные

Ядро Inflex дает возможность через контекстное меню, вызванное нажатием правой кнопкой мыши на строке списка, перейти к окнам свойств других данных, которые однозначно связаны с выбранной

строкой. Например, каждому экземпляру данных «движение объекта» соответствует один объект – тот, с которым собственно это движение оформляется.

Для предоставления этой возможности ядру необходимо предоставить идентификатор экземпляра данных и указать тип данных. Идентификатор связанного экземпляра данных также оформляется в виде отдельного столбца.

```

Название: Объект
SELECT:   Movements.Object_ID
FROM:     sm.Movements
WHERE:    <не указано>
Ссылочное имя: Свойства объекта
Ссылочный ключ обработчика: 1
    
```

Фраза «Свойства объекта» появится в контекстном меню, а ссылочный ключ обработчика 1 указывает на то, что в этом столбце идентификатор экземпляров данных типа «Объекты».

Описанный столбец играет очень важную роль в работе ядра, потому как он несет в себе информацию об отношениях между разными экземплярами данных.

## 9.5. Видимость столбцов

При разложении запроса, реквизит «Показывать по умолчанию» определяет отображение текущего столбца по умолчанию.

Реквизит имеет силу до тех пор, пока пользователь не определит собственную установку видимости по умолчанию.

Информация о пользовательском выборе по умолчанию сохраняется в его личной схеме.

## 9.6. Дополнительная специальная информация по столбцам

Указывая в ссылочном имени специальные слова для каждого столбца, может быть установлена специальная роль.

Таблица 9.1. Специальные ссылочные имена и их роль

PRIMARY	Значение столбца равно идентификатору экземпляра данных, тип которого указывается в ссылочном ключе обработчика.
SUBTYPE	Значение столбца указывает на номер подтипа, который используется для того, чтобы иметь возможность представлять в списке столбцы специфичные для конкретного подтипа.
INDICATOR	Значение столбца влияет на стиль отображения строки в списке – цвет и стиль шрифта.
RESTRICTION	Специальный столбец (псевдостолбец), поле SELECT не заполняется. Самым важным является поле WHERE. Этот столбец не отображается, он играет ограничивающее действие на выборку. Оформляется этот псевдостолбец для того, чтобы пользователь мог самостоятельно включать и отключать это ограничение.
ORDER BY	Указав это ссылочное имя для столбца, тем самым, Вы установите сортировку будущего списка по умолчанию – по этому столбцу.
DICT	Ссылочное имя – указание ядру Inflex, что этот столбец может использоваться в качестве признака группировки аналитического отчета.
DATE	То же, что и DICT, но только для столбцов, в которых используется дата.

Так как часть столбцов вообще не имеет смысла для отображения пользователю, для них устанавливается признак « **Скрывать в списке столбцов** ».

Из числа столбцов доступных для отображения, устанавливается признак « **Показывать по умолчанию** ». Когда пользователь впервые откроет список, отображены будут только те столбцы, у которых установлен этот признак. В дальнейшем пользователь может включить и отключить любой столбец, а также сохранить для себя собственные настройки отображения столбцов по умолчанию.

## 9.7. Флаг суммируемости

Для того чтобы ядро SAUMI могло определить возможность использования столбца при аналитических расчетах, необходимо поставить флажок “Суммируемое поле” на всех столбцах значения, в которых имеет смысл суммировать. К суммируемым полям относятся площадь, суммы платежей, начислений. Также рекомендуется указывать единицы измерения.

## 9.8. Специальные колонки (псевдостолбцы)

### 9.8.1. Описание общих секций FROM и WHERE

Для того чтобы в каждом столбце не указывать используемые для этого столбца таблицы, можно описать в отдельном псевдостолбце секции FROM и WHERE, но при этом не описывайте секцию SELECT. Этот столбец будет использован исключительно для того, чтобы при выборе пользователем любого набора столбцов, в этих секциях были перечисленные таблицы и условия.

### 9.8.2. PRIMARY – Главный ключ

Как уже было описано выше, это ссылочное имя присваивается полю таблицы, играющему роль **главного ключа**. По этому полю устанавливается однозначное соответствие данных по отношению друг к другу. Как следствие, главный ключ не может дублироваться и не может иметь пустого значения NULL.

### 9.8.3. RESTRICTION – Ограничение

Данное ссылочное имя устанавливается для ограничения выборки данных особым образом.

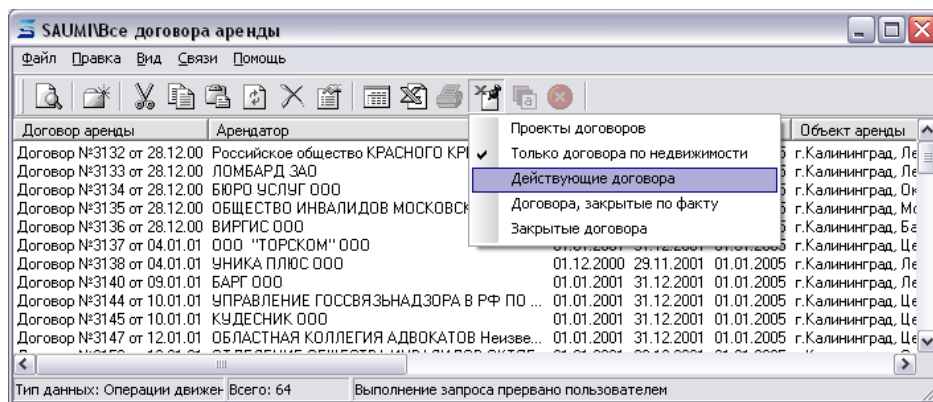


Рисунок 9.4. Ограничения запроса

Например, можно добавить ограничение для показа только действующих договоров. Для этого добавьте столбец с названием «Действующие договора». Укажите ссылочное имя «RESTRICTION», в поле WHERE укажите

```
sm.moveperiods.info is null and sm.documents.date_end>sysdate
```



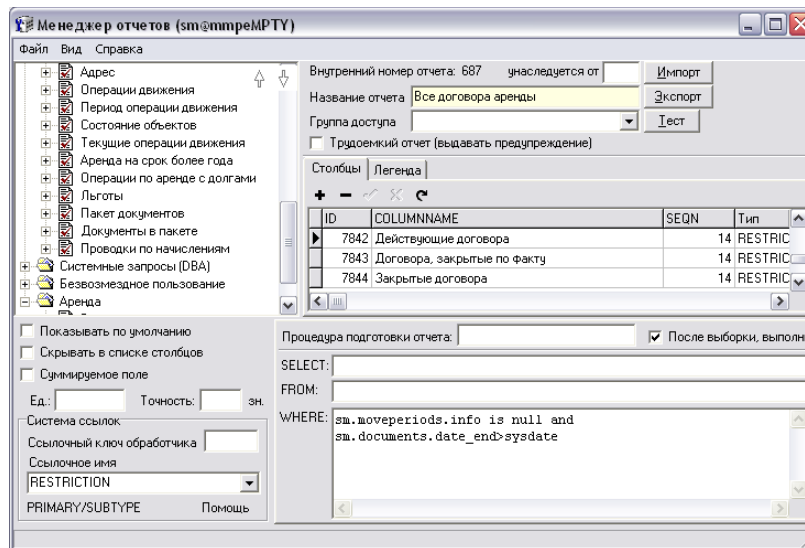


Рисунок 9.5. Добавление ограничения

Флажок «Показывать по умолчанию» нужно снять, иначе пользователи увидят сразу же ограниченный вариант запроса и могут не обратить внимание на то, что там скрыты законченные операции, а как следствие могут быть допущены ошибки в принятии решений.

В дальнейшем этим ограничением можно воспользоваться из окна данных, в панели инструментов устанавливая требуемые ограничения.

После включения ограничения, в списке не будут отображены те операции движения, дата окончания которых уже миновала.

Очень удобно использовать ограничения с параметрическими запросами. Таким образом, можно предоставить пользователю возможность описывать конкретные ограничения.

### 9.8.4. ORDER BY – Принудительная сортировка

Если указать ссылочное имя ORDER BY, то списки данных будут по умолчанию отсортированы по этому полю.

### 9.8.5. DICT – Группировочное поле аналитического отчета

Ссылочным именем DICT описываются группировочные поля аналитических отчетов. Поле может иметь два формата написания - одним словом DICT или DICT(Таблица.Поле).

Одним словом описываются группировочные поля, значения которых имеют вид, готовый для вывода в таблицу, например, название организации.

В случае же, если значение является ссылочным, то необходимо в скобках после слова DICT указать имя таблицы и поле, из которого будет извлекаться значение для вывода на экран. При этом таблица должна иметь главный ключ в поле ID.

### 9.8.6. DATE – Группировочное поле даты аналитического отчета

Особый вид группировочного поля – хронологические значения. Хронологический анализ очень удобен для анализа различных данных изменяющихся во времени. Если установлено ссылочное имя DATE, у пользователя будет возможность выбрать различные интервалы для анализа.

### 9.8.7. INDICATOR – Выделение строк цветом и стилем

Для удобства работы конечного пользователя имеется возможность выделения отдельных строк результата, полученного в окне данных, путем установки определенного цвета и изменение стиля начертания.

При установке ссылочного имени специального значения «INDICATOR» этот столбец рассматривается, как несущий информацию о стиле отображения строки данных на экране.

Пример. Установили ссылочное имя «INDICATOR», а в поле SELECT ввели численное значение «12». В результате все строки при отображении этого отчета будут синими.

Этот псевдо-столбец называется индикатором.

Значение псевдо-столбца имеет побитовую значимость, т.е. каждый бит имеет определенный эффект на стиль отображения строки:

Значимыми являются нижние 8 бит, при этом первые четыре бита устанавливают цвет строки, а верхние четыре - стиль

128	64	32	16	8	4	2	1
К	Ч	Абв	Ж	С3	С2	С1	С0

Таблица 9.2. Кодировка цвета (С3,С2,С1,С0)

Сх	Цвет	Сх	Цвет
0	Черный	8	Светло-серый
1	Бордовый	9	Красный
2	Зеленый	10	Ярко-зеленый
3	Коричневый	11	Желтый
4	Темно-синий	12	Синий
5	Фиолетовый	13	Розовый
6	Сине-зеленый	14	Голубой
7	Темно-серый	15	Белый

Вы можете использовать несколько столбцов-индикаторов, при этом будет происходить битовое наложение (или). Разумеется, только один из них должен отвечать за цвет.

Удобно использовать запросы DECODE для установки стиля по реквизитам.

Пример, в котором будут выделены красным цветом нежелательные субъекты права.

```
SELECT: DECODE (Clients.Unpreferred, 'Y', 9, 0)
```

Пример, в котором будут зачеркнуты ликвидированные субъекты права.

```
SELECT: DECODE (Clients.Destroyed, 'Y', 32, 0)
```

Возможно использование сложения в одном столбце.

```
SELECT: DECODE (Clients.Destroyed, 'Y', 32, 0) + DECODE (Clients.Unpreferred, 'Y', 9, 0)
```

Причем это более предпочтительно, так как снизит нагрузку на сеть (будет передаваться только одно число, вместо двух).

---

# Глава 10. Параметрические SQL-запросы

## Содержание

10.1. Введение параметра в SQL-запрос .....	119
10.2. Примеры использования .....	120

Параметрические SQL-запросы предназначены для того, чтобы однажды подготовленный запрос мог быть многократно применен с возможностью изменения некоторых параметров его выполнения.

Это может быть полезно в случаях, когда требуется подготовить отчет по данным на какой-то конкретный момент. Параметрический SQL-запрос может предоставить возможность пользователю самому указать, на какой момент требуется запрошенный отчет.

Параметрические SQL-запросы обрабатываются ядром SAUMI во время формирования запроса для отображения в окне данных.

Технология может быть использована при описании запросов в «Менеджере отчетов».

## 10.1. Введение параметра в SQL-запрос

В любом месте SQL фрагмент следующего вида будет рассмотрен как параметр.

```
(1)
...
{[VarName:] VarType = [(Default)] [Description [(Item1|Value1|Item2|Value2|...)]]}
...
или
(2)
...
{VarName}
...
```

Здесь в первом (1) случае, используем при однократном или первом применении параметра:

**VarName** – название параметра, необязательное поле, используется в случае необходимости подстановки в несколько мест;

**VarType** – тип параметра:

**DATE** дата (по умолчанию ставится текущая дата),  
**INTEGER** целое число (по умолчанию 0),  
**NUMBER** любое число (по умолчанию 0),  
**STRING** строка (по умолчанию пустая строка),  
**LIST** список значений;

**Default** – значение по умолчанию, без каких-либо кавычек и апострофов, нельзя применять вычисляемые значения;

**Description** – описание параметра, которое будет выведено в диалоге пользователю;

**Item1 | Value1 | Item2 | Value2 | ...** - элементы списка значений для типа параметра LIST. Последовательно перечисляются предлагаемые пользователю для выбора элементы списка (Item1, Item2, ...) и соответствующие этим элементам значения (Value1, Value2, ...). Каждый пункт значения оформляется так, как он должен быть подставлен в текст запроса, т.е. строки и даты должны быть выделены апострофами. Элементы списка и значения разделяются вертикальной чертой "|".

При ошибках в описании параметров SQL запроса будет выдана ошибка «Ошибка обработки параметрического запроса».

В случае если необходимо в запросе использовать символ открытой фигурной скобки "{" в других целях (не для параметров), то поставьте перед фигурной скобкой символ "\" (обратный слэш).

Второй случай, помеченный на рассматриваемом листинге как (2), используется для подстановки значения ранее определенного параметра.

## 10.2. Примеры использования

### Пример 10.1. Запрос вывода суммы двух чисел

```
select
  {NUMBER = Первое слагаемое} +
  {NUMBER = второе слагаемое}
from
  dual;
```

### Пример 10.2. Запрос вывода суммы двух чисел и произведения

```
select
  {x:NUMBER = Первое слагаемое} +
  {y:NUMBER = второе слагаемое},
  {x}*{y}
from
  dual;
```

### Пример 10.3. Запрос вывода суммы или произведения двух чисел по выбору пользователя

```
select
  {NUMBER = Первое слагаемое}
  {LIST = Действие (Сумма|+|Произведение|*)}
  {NUMBER = второе слагаемое},
from
  dual;
```

### Пример 10.4. Запросить балансовую и остаточную стоимость имущества (ID=23745) в день по выбору пользователя

```
select
  sm.GetObjectCost(23745, '{D:DATE = Дата расчета стоимости}','R'),
  sm.GetObjectCost(23745, '{D}','B')
from
  dual;
```

# Глава 11. «Менеджер форм»

## Содержание

11.1. Создание новой формы .....	122
11.1.1. Тип печатной формы .....	122
11.1.2. Быстрый вызов .....	123
11.1.3. Доступны для роли .....	123
11.1.4. Отладочный набор элементов .....	123
11.2. Дизайнер отчетов .....	123
11.2.1. Построение простейшего автономного отчета .....	125
11.2.2. Привязка списка к данным в окне .....	126
11.2.3. Оформление печатной формы .....	129
11.3. Формирование текстов документов .....	130
11.4. Экранные формы .....	131
11.5. Произвольные диалоговые окна .....	132
11.6. Ограничения поиска связанных данных .....	133

Настройка SAUMI в большинстве случаев сводится к формированию особого вида печатных и экранных форм. К печатным формам относятся все виды документов, которые так или иначе выводятся в конечном счете на печать. Экранные формы – настраиваемые формы диалоговых окон, отображаемых пользователю наряду с «родными» окнами SAUMI для выполнения специфичных задач.

Программа «Менеджер форм» построена на основе дерева отчетов, формируемого в «Менеджере отчетов».

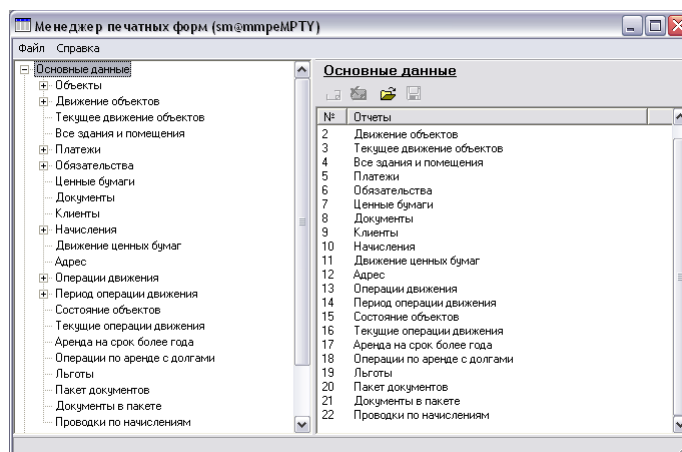


Рисунок 11.1. Менеджер форм

Для каждого отчета в дереве могут быть сопоставлены:

- формы для оформления печатных списков и выписок;
- экранные формы для произвольных диалоговых окон;
- формы для автоматического формирования текстов документов;
- экранные формы для ограничения поиска связанных данных.

## 11.1. Создание новой формы

Для описания новой формы любого вида используется встроенный редактор отчетов. Для того чтобы сформировать новую форму в дереве отчетов укажите отчет, для которого планируется описать форму, в правой части окна будет приведен список форм уже сопоставленных для этого отчета.

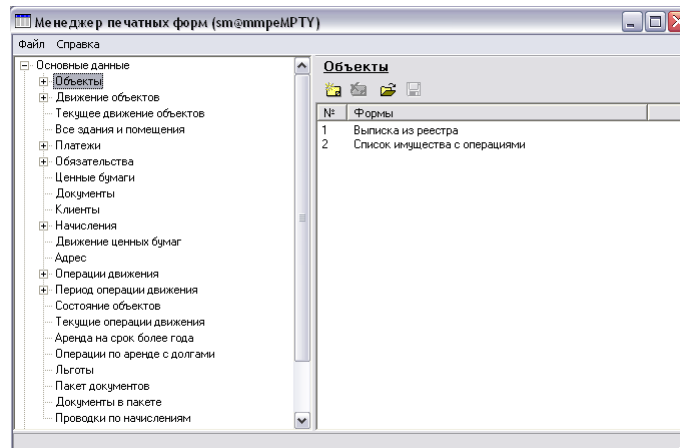


Рисунок 11.2. Указание необходимой ветки

Для добавления новой формы нажмите кнопку создания формы. Вместо списка отчетов будет отображена вкладка описания формы. В этой вкладке необходимо указать название формы. Это название в дальнейшем будет отображаться в различных меню.

Служебное имя указывает на тип формы. Использование служебного имени будет подробно описано ниже. В случае печатных и экранных форм для произвольных диалоговых окон указывать служебное имя не надо.

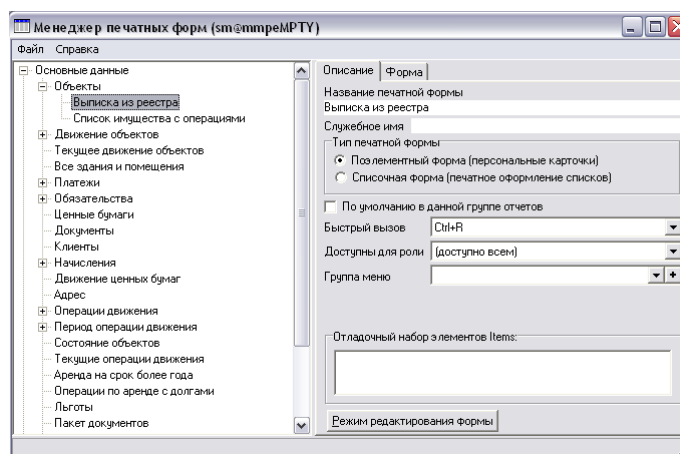


Рисунок 11.3. Создание новой формы

### 11.1.1. Тип печатной формы

Тип печатной формы указывает на то, каким образом пользователю необходимо будет указывать пункты в окне данных, которые следует отразить в печатной форме.

Возможны два типа форм - поэлементная форма и списочная формы.

*Поэлементная форма* используется для оформления различного рода выписок, справок, бланков писем и т.д. Список доступных поэлементных форм можно получить из контекстного меню, выделив один или несколько элементов окна данных, для которых требуется распечатать форму.

*Списочная форма* используется для оформления печати списков, реестров. Списочные формы доступны по нажатию на кнопку «Печать» панели инструментов. При этом в форму будут включены те элементы, которые выделены. Если в окне данных нет выделенных элементов или выделен только один, то в форму будут включены все данные из текущего списка.

### **11.1.2. Быстрый вызов**

Для вызова формы нажатием клавиш необходимо выбрать, на какое сочетание будет настроена реакция системы. Это сочетание клавиш распространяется на все окна SAUMI, из которых доступна выбранная печатная форма.

### **11.1.3. Доступны для роли**

Для ограничения пользования печатной формой можно установить перечень ролей, владельцам которых будет доступен ее вызов. Возможно, окажется, что та группа пользователей, которой требуется обеспечить доступ к печатной форме, не имеет общей роли. В таком случае новую роль можно создать в «Менеджере пользователей» и назначить эту роль той группе пользователей, которой необходимо обеспечить доступ.

Следует иметь в виду, что установка какой-либо конкретной роли, делает возможным доступ к печатной форме исключительно владельцам этой роли.

### **11.1.4. Отладочный набор элементов**

При работе в режиме дизайнера удобно отлаживать вид формы на основе реальных данных, которые можно получить из окна данных. Основой для отражения в форме данных являются главные ключи в соответствии с тем, как они настроены в «Менеджере отчетов». В списке отладочного набора элементов указывается произвольное количество реальных главных ключей. Элементы указывается в виде целых чисел по одному на строке. Количество элементов не ограничено.

## **11.2. Дизайнер отчетов**

После того, как описание формы заполнено, можно переключить «Менеджер форм» в режим редактирования формы нажатием соответствующей кнопки «Режим редактирования формы». После этого появляется окно дизайнера отчетов.

В SAUMI используется генератор отчетов FastReport 2.4. Полная документация по использованию возможностей этого генератора отчетов доступна непосредственно из дизайнера отчетов, меню «Помощь».

Процесс получения готового отчета включает в себя несколько этапов:

- получение базовых данных;
- запрос у пользователя параметров для формирования отчета;
- обработка формы с учетом данных и параметров.

Базовые данные представляют собой совокупность данных, на основе которых строится отчет. Далее осуществляется запрос параметров у пользователя (например, диапазон дат, по которому необходимо вывести данные). Некоторые отчеты обходятся без этого этапа, либо используют фиксированную установку параметров (без запроса их значений в диалоге).

Форма отчета представляет собой набор элементов, описывающих то, как должен выглядеть готовый отчет. Для группировки элементов по их расположению в готовом отчете применяются бэнды (англ. band – полоска). Бэнды разделяются на два вида: служебные (заголовок отчета, страницы и пр.) и бэнды, образующие многострочную часть отчета (далее – дата-бэнды). Дата-бэнды подключаются к источникам данных, и их содержимое выводится столько раз, сколько имеется строк данных в источнике.

Для построения формы отчета используется визуальная среда разработки – дизайнер. В FastReport дизайнер – это средство разработки отчетов, сочетающее простоту и удобство использования. Интерфейс дизайнера выполнен на современном уровне с использованием панелей инструментов (toolbars), расположение которых можно изменять по своему вкусу. Для удобства манипуляции свойствами объектов отчета, используется инспектор объектов, аналогичный инспектору объектов языков программирования Delphi, C++ Buildier, J++ Builder, Visual Basic.

Под обработкой понимается обработка входных данных, модификация формы отчета или отдельных ее компонентов в процессе построения отчета. Простейший пример такой обработки – вывод отрицательных сумм красным цветом. Более сложный пример обработки – печать суммы, которая подсчитывается в подвале группы, в ее заголовке.

Реализовать подобную обработку можно, прибегнув к написанию обработчиков событий на встроенном языке – упрощенному аналогу Pascal. Скрипты, написанные на этом языке, по сути дела, являются обработчиками событий, которые вызываются перед прорисовкой объектов. Это дает возможность выполнять довольно сложную обработку информации.

Возможности встроенного языка FastReport довольно широки. Из скрипта доступны все свойства и методы объектов отчета, а также переменные, поля таблиц БД. В скрипте можно создавать переменные и массивы, которые будут доступны во всем отчете. О возможностях встроенного языка говорит тот факт, что такая довольно сложная задача, как печать сумм группы в ее заголовке (сама сумма считается в подвале группы) средствами языка FastReport делается элементарно.

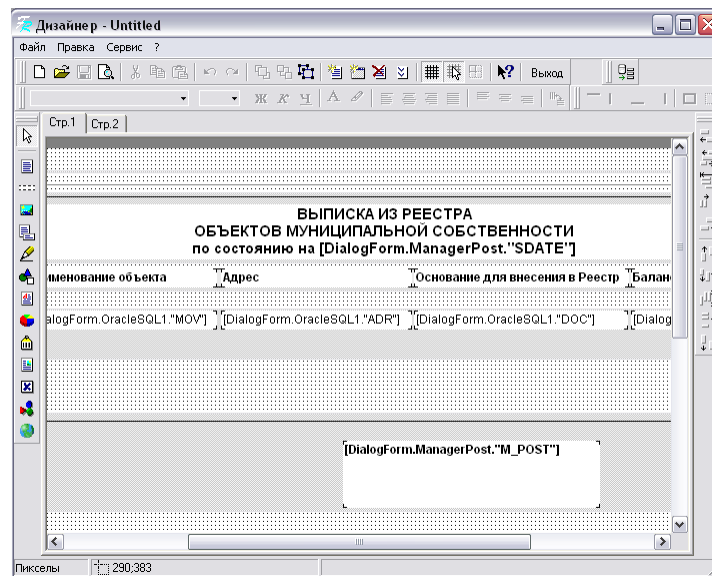


Рисунок 11.4. Редактор Fast Report



Готовый отчет представляет собой продукт деятельности ядра FastReport – то, что мы видим при нажатии кнопки «Предварительный просмотр». В отличие от многих генераторов отчетов, которые хранят содержимое страниц отчета в виде метафайла (т.е. изображения в формате EMF), в FastReport готовый отчет представлен набором объектов, описывающих содержимое каждой страницы отчета. Это позволяет модифицировать готовый отчет, загружая нужную страницу в дизайнер. Кроме того, можно описывать реакцию на щелчок мыши на нужном объекте в режиме предварительного просмотра отчета.

## 11.2.1. Построение простейшего автономного отчета

Для того чтобы овладеть принципами построения отчетов рекомендуется начать с простого примера. В этом примере строится отчет - список зарегистрированных транспортных средств.

Так как в основе отчета лежат данные, необходимо определиться, как их получить и с какой детальностью. Допустим, требуется получить список транспортных средств, в котором будет указан государственный номер, номер двигателя и шасси.

То есть, необходим список транспортных средств следующего вида.

ЗИЛ 130	431410	T078AC	723504	2596322
ЗИЛ 130	431610	Y187BH	382678	2884355
ГАЗ 66		E786AH	1841505	0169165
ЗИЛ 130	433362	A810EM	225559	3441180
ЗИЛ	431410	O236EM	683844	3057034

Последовательность шагов создания формы.

1. В «Менеджере форм» укажите в дереве отчетов, в ветке «Основные данные» отчет «Объекты». В правой части окна нажмите кнопку создания новой формы. Укажите название печатной формы – «Реестр транспортных средств».
2. Задайте тип формы: поэлементная или списочная. В данном случае форма будет списочной.
3. Затем нажмите «Режим редактирования формы». Создайте дополнительную форму диалога через меню «Правка» (либо щелкнув правой кнопкой мыши по наименованию имеющиеся страницы) и поместите на эту форму компонент «Запрос» из панели компонентов в левой части окна «Дизайнера».
4. Выделите вставленный компонент.
5. В инспекторе объектов отображаются свойства выделенного объекта. Дважды нажмите на поле ввода значения свойства SQL. В появившемся окне введите текст запроса.

```
select
  Objects.Description,
  Transport.FedNo,
  Transport.MotorNo,
  Transport.ChassisNo
from sm.Objects, sm.Transport
where Objects.ID=Transport.Objects_ID
```

6. Переключитесь на стр.1 для оформления внешнего вида отчета.
7. Выберите в панели инструментов слева от формы инструмент «Вставить секцию» и поместите ее на страницу. Будет запрошен тип секции – укажите «Master Data».
8. Далее следует указать источник данных `DialogForm.Query1`, то есть тот, где был указан запрос транспортных средств.

9. Включите окно «Вставка поля БД» из меню «Сервис | Панели инструментов | Вставка поля БД».

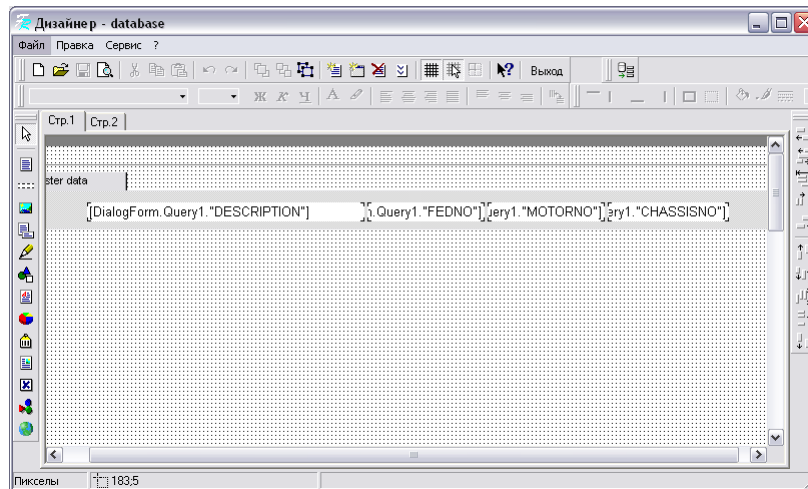


Рисунок 11.5. Размещение полей

10. В окне «Вставка поля БД» выберите источник DialogForm.Query1 и перетащите поочередно все поля (DESCRIPTION, FEDNO, MOTORNO, CHASSISNO) в секцию «Master data».

11. Простейший отчет готов. Его можно просмотреть, нажав кнопку предварительного просмотра.

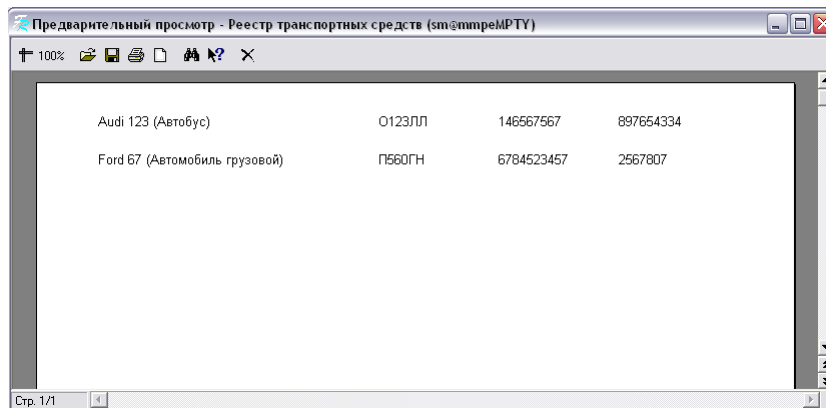


Рисунок 11.6. Предварительный просмотр формы

Полученный отчет является автономным, то есть не зависящим от данных отображаемых в окне данных SAUMI. Подобные отчеты есть смысл делать в соответствующем специальном разделе «Автономные отчеты» «Менеджера форм». Список автономных отчетов доступен пользователям через меню панели SAUMI. Автономные отчеты отличаются от других отчетов еще и тем, что поля, выводимые пользователю, следует помещать в секцию «Master Data». В остальных отчетах для этого используется секция «Detail Data».

### 11.2.2. Привязка списка к данным в окне

Полученный простейший отчет может быть достаточно легко преобразован до вида, когда его содержание будет напрямую зависеть от того, какие данные отображены в окне данных SAUMI. Для этого требуется лишь модифицировать текст SQL запроса так, чтобы ему передавалась информация о том, какие данные следует запрашивать.

```
select
  Objects.Description,
  Transport.FedNo,
  Transport.MotorNo,
  Transport.ChassisNo
from sm.Objects, sm.Transport
where
  Objects.ID=Transport.Objects_ID
and Objects.ID=:ITEM_ID
```

То есть к тексту оригинального запроса добавлена только одна строка (последняя). В этой строке устанавливается ограничение на запрашиваемые строки. В данном случае фактически будет запрошена одна строка со значение ID равным переменной ITEM\_ID.

Далее на форму отчета следует поместить новую секцию «Detail Data» и перетащить туда все поля (DESCRIPTION, FEDNO, MOTORNO, CHASSISNO). В дальнейшем все данные для вывода следует помещать в секции «Detail Data» (кроме автономных отчетов), а секцию «Master Data» следует использовать для связи с переменной ITEM\_ID.

ITEM\_ID – особая переменная, которой поочередно присваивается значение главного ключа списка элементов окна данных. То есть, если Вы поместили отчет в ветку операции движения, то ITEM\_ID будет соответствовать MOVESETS.ID и т.д.

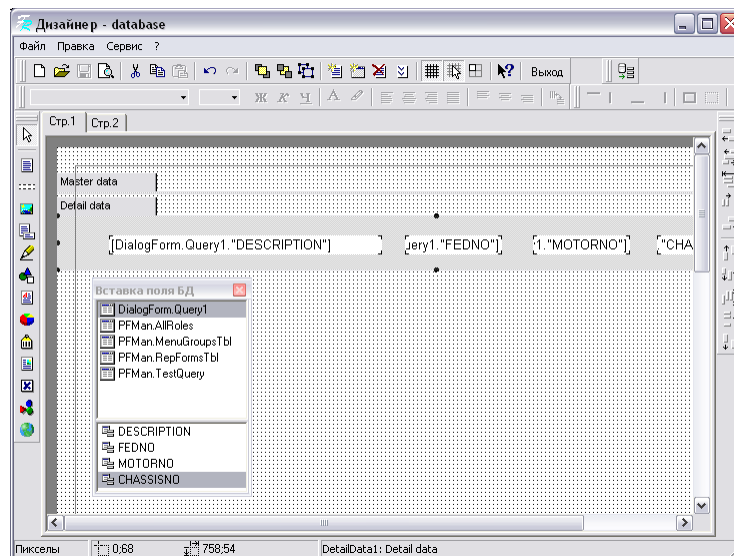


Рисунок 11.7. Добавление новой секции

Теперь переменную следует привязать. Для этого в окне редактирования текста запроса следует нажать значок «Edit variables». Если Вы уже добавили ITEM\_ID в запрос, то достаточно нажать на кнопку «Scan SQL» для инициализации переменной.

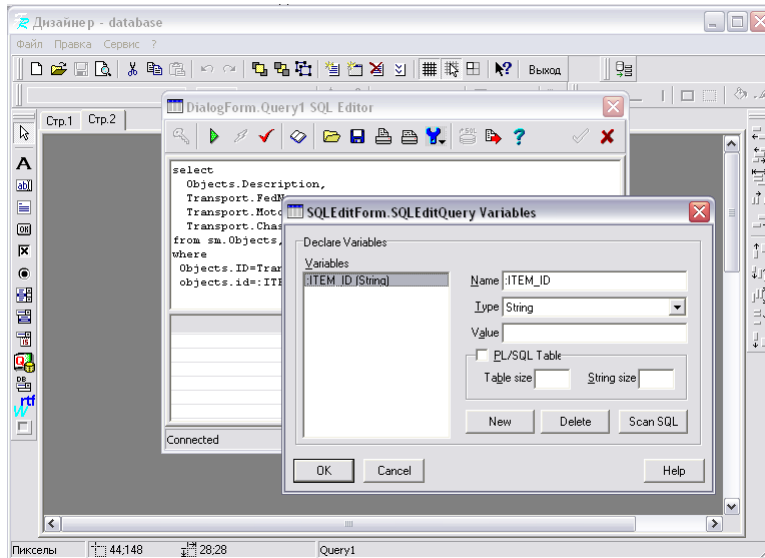


Рисунок 11.8. Привязка переменной ITEM\_ID

Далее необходимо в качестве источника данных секции «Master data» указать специальный источник **Items**. Для этого следует дважды щелкнуть мышкой по свободной части секции, тогда появится окно выбора источника данных для секции. Окно выбора источника данных можно вызвать и через контекстное меню секции, нажав на пункт меню «Редактор».

Для «Detail Data» источником данных будет являться ваш запрос, в данном примере DialogForm.Query1.

Сохраните полученный отчет кнопкой панели инструментов «Сохранить».

Для того чтобы проверить работу полученного отчета, откройте SAUMI, из окна поиска найдите интересующие транспортные средства.

На панели инструментов нажмите на кнопку «Печать». Откроется список возможных печатных форм для этого списка объектов.

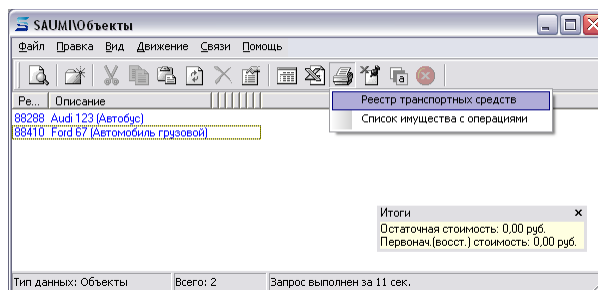


Рисунок 11.9. Вызов получившейся формы

Выберите созданную печатную форму «Реестр транспортных средств». На экране появится окно предварительного просмотра, где в подготовленной печатной форме будут отображены выбранные данные.

### 11.2.3. Оформление печатной формы

Чтобы придать отчету полноценный вид документа необходимо оформить к нему шапку и «подвал». Шапка документа оформляется с помощью секции «Report title». То есть все, что будет помещено в эту секцию, будет распечатано перед списком данных из секции «Detail data». «Подвал» документа оформляется в секции «Report summary». Также можно специально оформить шапку и «подвал» каждой страницы документа с помощью секций «Page header» и «Page footer».

Если необходимо для каждой строки данных вывести дополнительную «подчиненную» информацию, например, для каждого транспортного средства необходимо отобразить список операций движения зарегистрированных с этим транспортом, то можно воспользоваться секцией «Subdetail data». Источником данных для этой секции будет запрос, помещенный на второй странице. Причем этот запрос должен быть связан с первым запросом полем, на основании которого определяется подчиненность данных. Таковым полем является ITEM\_ID.

Текст запроса для примера будет выглядеть следующим образом.

```
select * from sm.Movements where Objects_ID=:ITEM_ID
```

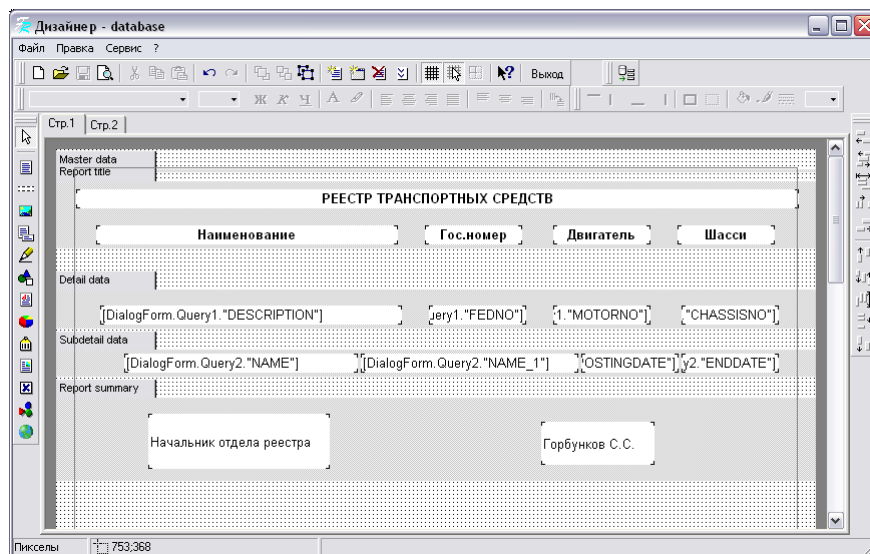


Рисунок 11.10. Добавление новых секций

Чтобы не выбирать все поля, нужно ограничить выборку.

```
select
  MoveType.Name,
  Clients.Name,
  Movements.PostingDate,
  Movements.EndDate
from
  sm.Movements,
  sm.Clients,
  sm.MoveType
where
  Movements.ClientsTo_ID=Clients.ID and
  MoveType.ID=Movements.MoveType_ID and
  Objects_ID=:ITEM_ID
```

Создайте теперь на стр.1 секцию «Subdetail data» и поместите туда поля из запроса DialogForm.Query2.

Теперь Вы получили отчет с более глубокой информацией по отношению к той, которую видит пользователь в окне данных SAUMI.

Наименование	Гос.номер	Двигатель	Шасси
Audi 123 (Автобус)	О123ЛЛ	146567567	897654334
Собственность	КМИ г.Кемерово Орган	01.01.2006	0
Балансодержание	"Новость" МУП	01.01.2006	0
Ford 67 (Автомобиль грузовой)	П560ГН	6784523457	2567807
Собственность	КМИ г.Кемерово Орган	01.01.2006	0
Безвозмездное пользование	ОРГПРОЕКТ МУП	01.01.2006	0
Начальник отдела реестра		Горбунков С.С.	

Рисунок 11.11. Внешний вид формы

### 11.3. Формирование текстов документов

В руководстве пользователя описан метод создания документов на основе шаблона Microsoft Word. Этот способ прост, но обладает намного меньшими возможностями по сравнению с формированием документов с помощью генератора отчетов.

Способ основан на том, что текст документа готовится как типичная печатная форма. При этом есть возможность содержание документа оформить с использованием всех возможностей генератора отчетов.

Для того чтобы эта печатная форма использовалась в качестве основы для документа, необходимо создать печатную форму в ветке «Период операции движения» и в описании формы указать служебное имя **DOCFORM**.

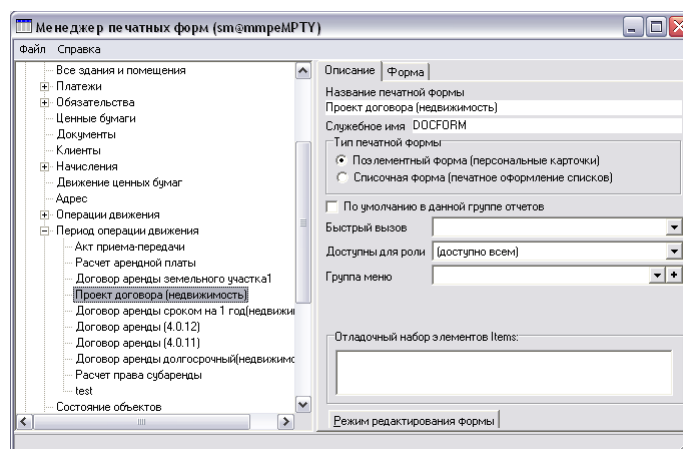


Рисунок 11.12. Формирование текста документа

Эта форма в дальнейшем будет доступна из окна свойств операции движения, со вкладки «Документы». Вызывается такая форма из контекстного меню, активизируемого щелчком правой кнопки мыши на строке описания документа.

После того, как печатная форма сформирована, она отображается в окне приложения FastReport, где она может быть выведена на принтер, а после закрытия окна FastReport форма после подтверждения запроса на сохранение сохраняется в базе данных как значение реквизита *Текст документа* выделенного в пакете документа.

## 11.4. Экранные формы

Использование экранных форм - диалоговых форм, которые могут полностью настраиваться администратором, может иметь несколько применений.

Основная цель экранных форм - настройка автономных отчетов - запрос параметров у пользователя (например, диапазон дат, по которому необходимо вывести данные).

Для примера рассмотрим печатную форму реестра транспортных средств, которая позволяет ограничить выборку транспорта исходя из года выпуска.

Возьмем за основу печатную форму, созданную в первом примере. Только теперь создадим данную форму в ветке «Автономные отчеты».

Добавим диалоговое окно, на которое поместим следующие элементы управления: «Label», «Edit» и «Button». Поместим элемент управления «Запрос» аналогично первому примеру. Текст запроса будет следующим.

```
select
  Objects.Description,
  Transport.relyear,
  Transport.FedNo,
  Transport.MotorNo,
  Transport.ChassisNo
from sm.Objects, sm.Transport
where
  Objects.ID=Transport.Objects_ID
```

То есть мы добавили еще одно поле: «Год выпуска». Следует отметить, что текст запроса мы добавили только для удобства размещения полей. В дальнейшем текст будем формировать динамически.

Диалоговое окно на стр.2 выглядит следующим образом.

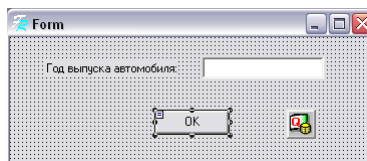


Рисунок 11.13. Пример диалогового окна

На стр.1 отчета поместим секцию «Master Data», укажем для нее источник данных `DialogForm.Query1` и разместим необходимые поля.

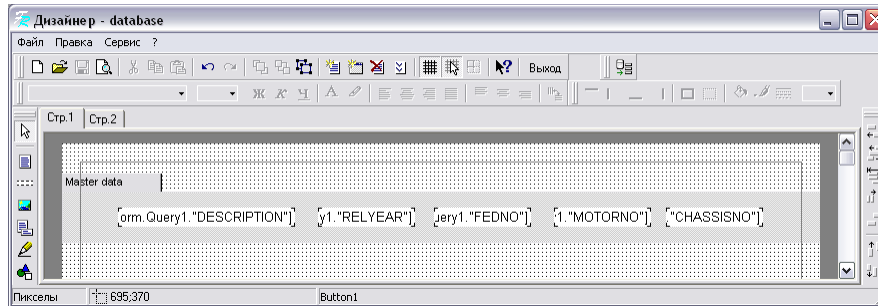


Рисунок 11.14. Пример автономного отчета

Теперь следует очистить текст запроса, так как ранее говорилось, что запрос будет формироваться динамически.

Необходимо написать обработчик события кнопки. В обработчике как раз и будет формироваться запрос исходя из введенного значения года и добавляться активность запросу. Чтобы добавить текст обработчика нужно дважды щелкнуть на кнопке «ОК». Текст обработчика выглядит следующим образом.

```
Begin
//Формируем текст запроса
  Query1.SQL:='select '+
    'Objects.Description, '+
    'Transport.relyear, '+
    'Transport.FedNo, '+
    'Transport.MotorNo, '+
    'Transport.ChassisNo '+
    'from sm.Objects, sm.Transport '+
    'where '+
    'Objects.ID=Transport.Objects_ID '+
    'and Transport.relyear='+Edit1.Text;
//Делаем запрос активным
Query1.Active:=true;
  ModalResult:=mrOk;
End
```

После сохранения запроса его работу можно посмотреть в меню автономных отчетов, то есть после нажатия на панели SAUMI кнопки «SAUMI 4.10».

Подобным образом можно создать отчеты, ограничивающие выборку диапазоном дат.

## 11.5. Произвольные диалоговые окна

В SAUMI применено более универсальное использование экранных форм - использование в качестве инструмента расширения интерфейсных возможностей. Например, можно создать собственные диалоговые окна, которые определенным образом будут обрабатывать данные в БД SAUMI. Например, если систематически проводится перерасчет арендной платы в связи с изменением коэффициентов, можно создать диалоговое окно, из которого можно будет произвести полный перерасчет и сформировать письма-уведомления арендаторам о произведенном перерасчете.

Для формирования экранной формы в дизайнера отчетов, удалите все страницы и создайте форму диалога. Далее Вы можете оформлять окно в соответствии с вашими требованиями.

Работа окна, как правило, завершается нажатием на какую-то кнопку на форме, после чего выполняются определенные действия. Для того чтобы назначить обработку событий нажатия на



выбранную кнопку, дважды щелкните мышкой по ней. В открывшемся окне редактора текста описывается программа на внутреннем языке FastReport.

Хорошим примером для знакомства с принципами составления экранных форм может служить форма «Формирование учётных данных по движению казны», путь вызова которой из меню SAUMI задается следующим образом: **SAUMI 4.10 | Отчётные формы | Учёт имущества казны | Формирование учётных данных по движению казны.**

## 11.6. Ограничения поиска связанных данных

Если воспользоваться возможностью поиска связанных данных из окна данных SAUMI, то при запросе, например, платежей для определенного субъекта права, будут выведены все платежи, произведенные этим субъектом за всю историю с любым назначением. Как правило, такой объем информации не нужен, а требуется получить данные за ограниченный период времени.

Для предоставления пользователю возможности описать это ограничение может быть создана форма ограничения поиска связанных данных. Метод построения этой формы такой же, как и для произвольных диалоговых окон, с тем лишь различием, что требуется в описании формы указать служебное имя `RESTRICT_FORM`. Работа формы должна закончиться вызовом процедуры `ExplorerWindow` с ограничивающим SQL-запросом в качестве параметра. За примером для изучения работы форм ограничения поиска связанных данных обратитесь к форме «Выбор начислений» отчета «Начисления».

---

# Глава 12. «Менеджер справочников»

## Содержание

12.1. Навигация в перечне справочников .....	135
12.2. Создание нового справочника .....	137
12.3. Изменение названия справочника .....	139
12.4. Добавление поля в справочник .....	139
12.5. Добавление записей в справочнике, редактирование записей .....	141
12.6. Назначение (изменение) дополнительного отображаемого столбца .....	142
12.7. Преобразование линейного справочника в древовидный .....	143
12.8. Назначение принудительной загрузки отдельных справочников пользовательскими рабочими станциями .....	144
12.9. Назначение прав доступа к отдельным справочникам .....	145
12.10. Объединение элементов-дубликатов .....	145
12.11. Автоматическая замена КБК в закрытых договорах .....	146
12.12. Служебная таблица DICTIONARIES – перечень справочников .....	149

Используемые в SAUMI справочники играют очень важную роль в систематизации учета, как имущества, так и любых учитываемых сведений, будь то информация о юридических лицах или документах. Хорошо организованные справочники существенно облегчают работу с комплексом за счет того, что пользователям становится понятно, что от них требуется. Также от организации справочников зависит глубина аналитических отчетов, которые в дальнейшем можно получить.

Утилита «Менеджер справочников» из комплекта администратора SAUMI предназначена для централизованного управления справочниками программного комплекса.

## 12.1. Навигация в перечне справочников

В открытом окне «Менеджера справочников» левая панель содержит перечень имеющихся справочников, один из которых выделен и является текущим. Таблица в правой панели отображает значения реквизитов записей текущего справочника.

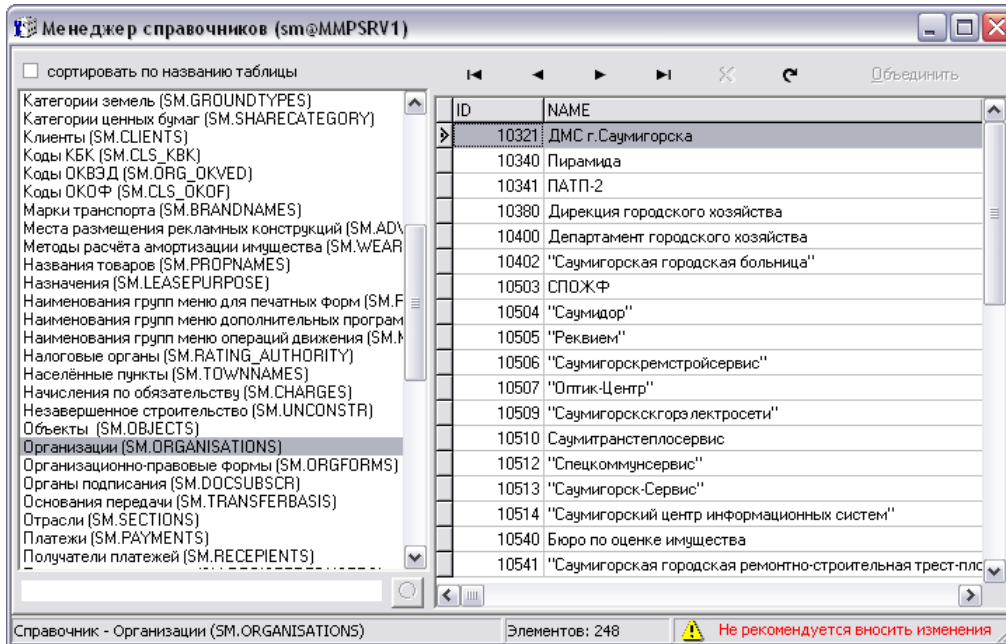


Рисунок 12.1.

Флажок «сортировать по названию таблицы» на панели инструментов задает порядок следования справочников в перечне. Когда этот флажок сброшен, то перечень справочников упорядочен (рис. 12.1) по их русскоязычным названиям (описаниям). Изначально при запуске утилиты флажок сброшен.

При выставлении указанного флажка упорядочение справочников в левой панели производится по названиям таблиц в латинице.

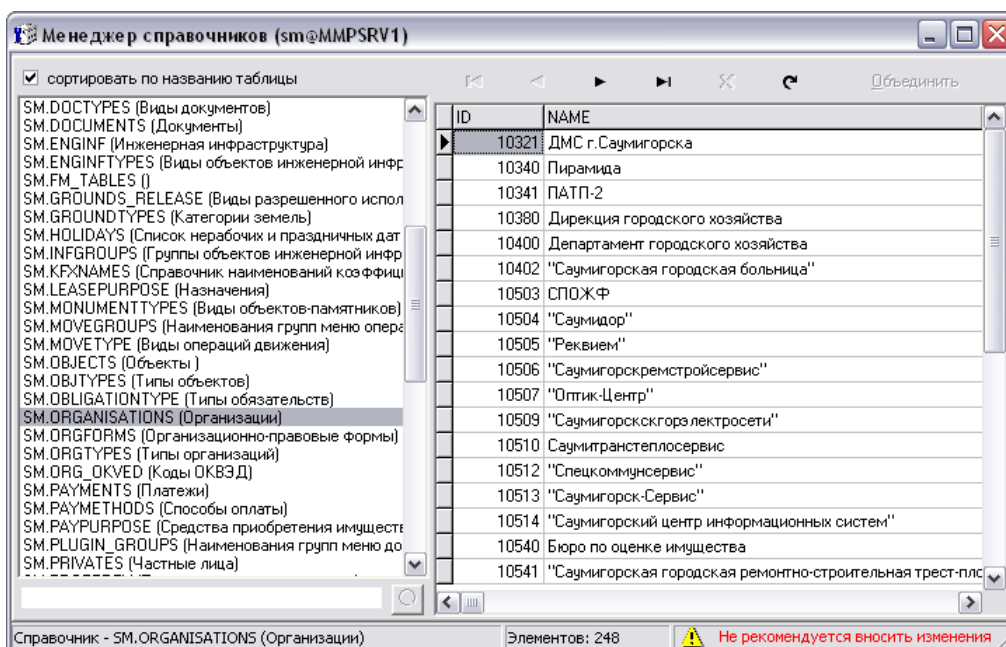



Рисунок 12.2.

В нижней части левой панели окна «Менеджера справочников» располагается строка поиска справочника. Если задать в ней фрагмент описания справочника или часть названия таблицы, а затем нажать кнопку , или клавишу **Enter**, или клавишу **F3**, то будет произведен поиск справочника, и, в случае совпадения введенной строки с любым фрагментом названия таблицы или описания справочника, поиск будет успешным и первая по порядку таблица будет выделена в левой панели утилиты.

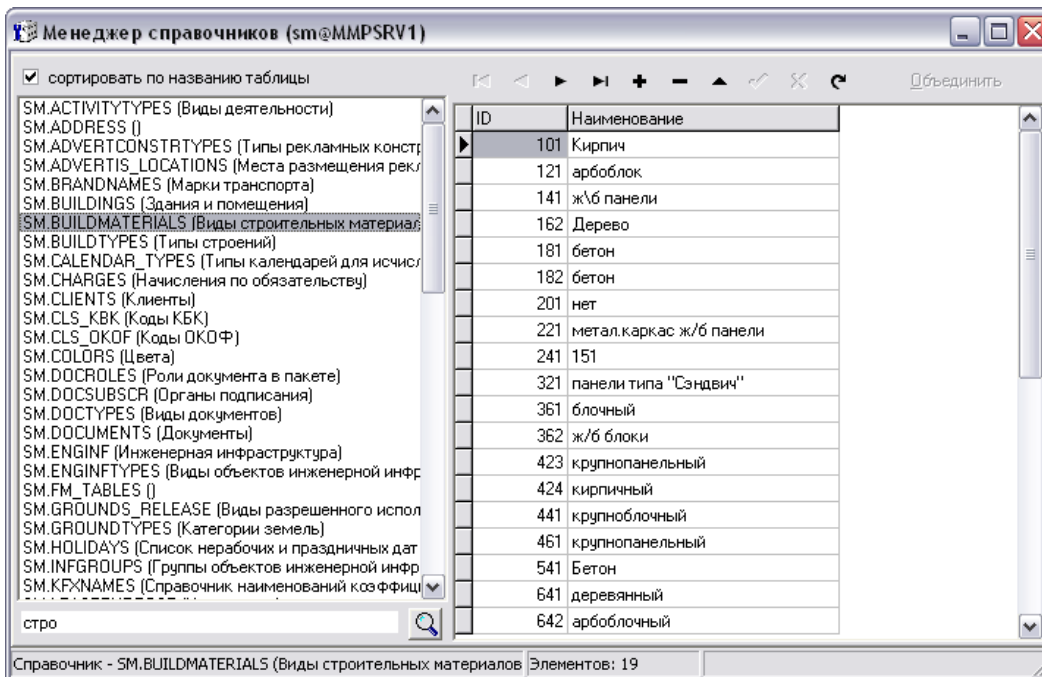



Рисунок 12.3.

Повторное нажатие кнопки , или клавиши **Enter**, или клавиши **F3** без замены строки поиска выделит следующую по порядку таблицу, удовлетворяющую условию, а по достижении последней такой таблицы по порядку будет выделена снова первая, соответствующая заданной строке поиска.

## 12.2. Создание нового справочника

Создание нового справочника производится посредством контекстного меню перечня справочников левой панели «Менеджера справочников», вызываемого нажатием на правую кнопку мыши в указанной панели.

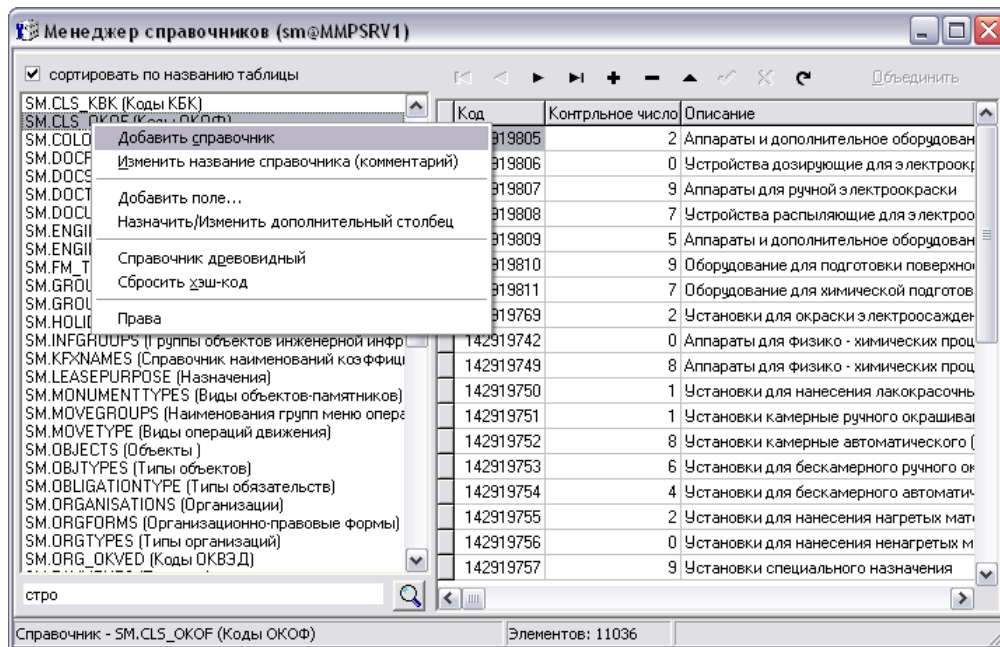


Рисунок 12.4.

В контекстном меню нужно выбрать элемент «Добавить справочник». В результате откроется окно определения нового справочника, в котором нужно задать латинскими символами имя таблицы, а также русскоязычное название справочника (комментарий).

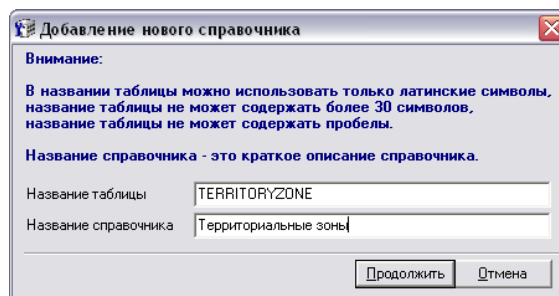


Рисунок 12.5.

Нажатие кнопки **Продолжить** создает новый справочник и вносит его в перечень справочников в левой панели. У нового справочника всегда имеются реквизиты идентификатор и название (поля таблицы соответственно ID и NAME).

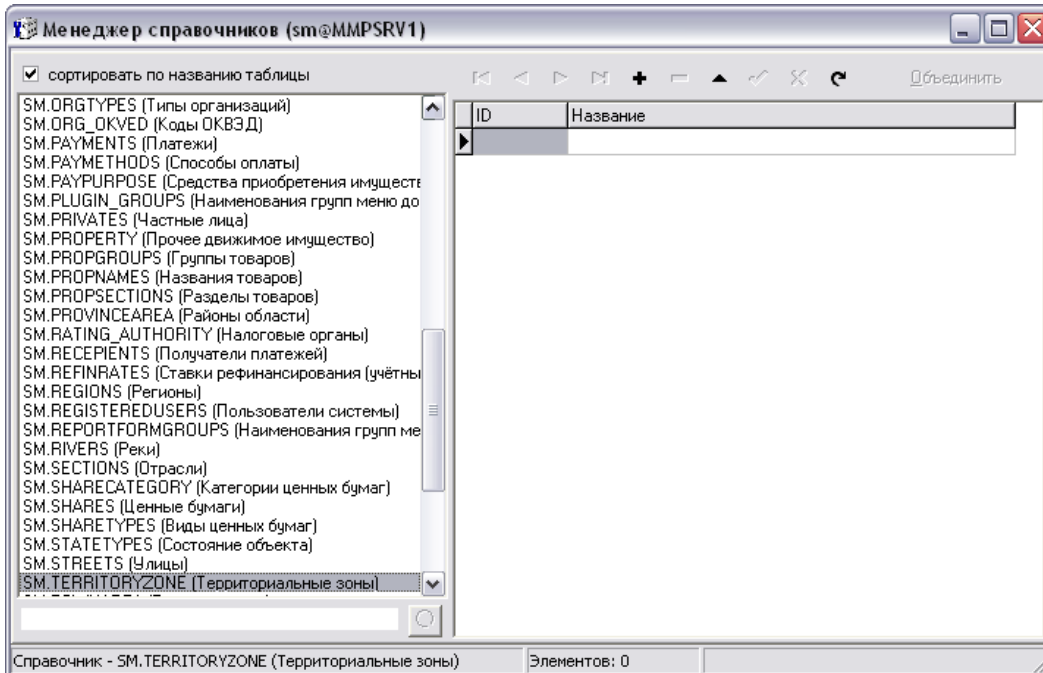


Рисунок 12.6.

### 12.3. Изменение названия справочника

Выбор в контекстном меню (рис. 12.4) элемента «Изменить название справочника (комментарий)» открывает окно редактирования русскоязычного названия (комментария) текущего справочника.

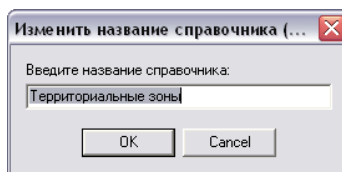


Рисунок 12.7.

#### Замечание

Набранное в латинице название таблицы, представляющей справочник, изменить нельзя – это имя используется как идентификатор в различных конструкциях СУБД Oracle.

### 12.4. Добавление поля в справочник

Чтобы добавить дополнительное поле в справочник, необходимо выделить этот справочник в левой панели «Менеджера» и в контекстном меню (рис. 12.4) выбрать пункт «Добавить поле...». В результате откроется диалоговое окно ввода нового поля.

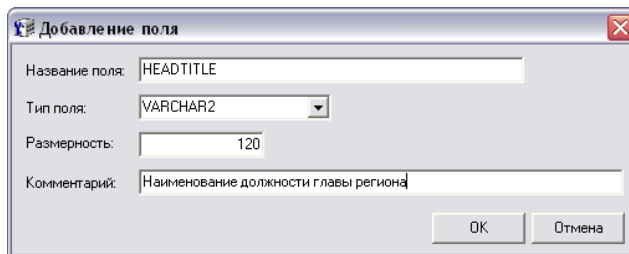


Рисунок 12.8.

Название поля составляется из символов латинского алфавита, цифр и символа подчерка '\_', причем первый символ – не цифра. Кроме того, имя поля не должно совпадать с каким-либо служебным словом SAUMI-Pascal Раздел 8.9, «Ограничения имен полей, представляющих дополнительные реквизиты».

После задания характеристик поля и нажатия на кнопку **ОК**, новое поле появляется в таблице.

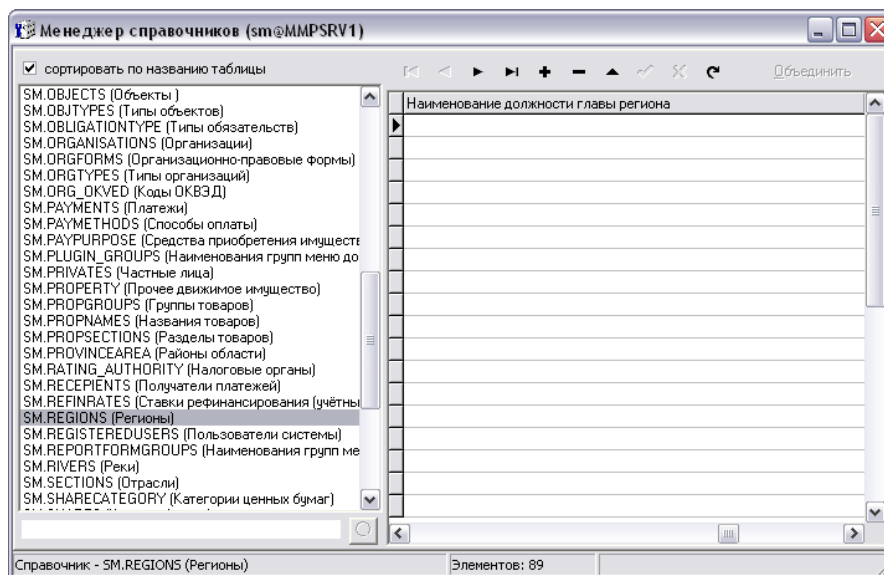


Рисунок 12.9.

## Замечание

Ряд справочников, представленных в перечне левой панели «Менеджера справочников», доступны только для просмотра и защищены от любых изменений. В информационной панели, расположенной в нижней части окна «Менеджера справочников», для таких справочников выводится сообщение (рис. 12.1) «Не рекомендуется вносить изменения в эту таблицу из Менеджера справочников!».

Тем не менее, редактировать защищенные справочники с помощью «Менеджера справочников» можно, но для этого утилита должна быть запущена в особом режиме. Такой режим достигается запуском утилиты из командной строки с модификатором `-all` ключа запуска `/d`.

```
Smc32.exe /d -all
```

Следует пользоваться этой возможностью с очень большой осторожностью, избегая появления несогласованных данных.

## 12.5. Добавление записей в справочнике, редактирование записей

Наиболее правильной является практика редактирования справочников посредством главного исполняемого модуля SAUMI в окне свойств некоторой сущности, содержащей реквизит, ссылающийся на данный справочник. Раскрываемый список имеет контекстное меню, в котором встроены функции добавления новых элементов справочника, а также редактирование имеющихся записей.


### Замечание


Наличие указанной возможности вовсе не означает, что ее должны использовать все пользователи без ограничений – на практике это приводит к появлению дубликатов в справочниках(см. 12.10). Более правильным решением является выполнение следующих трех административных действий.


1. Ограничить права на редактирование справочников для роли SM\_ALLUSERS(см. 12.9).
2. Создать роль (см. 1.2.2), которая бы давала права на редактирование справочника права на редактирование справочника её обладателю(см. 12.9).
3. Выдать созданную роль только квалифицированным специалистам, которые уполномочены вести справочники.

Однако в ряде случаев, в частности при добавлении новых полей в справочник (см. 12.4), редактирование справочника следует осуществлять с помощью «Менеджера справочников».


Табличное представление текущего справочника в правой панели «Менеджера реквизитов» поддерживает функции редактирования. Изменение значений полей производится перемещением фокуса ввода в соответствующую ячейку, как с помощью мыши, так и навигационных клавиш клавиатуры, с последующим вводом требуемого значения.


Сохранение измененной записи происходит либо при выделении в таблице другой записи, либо при нажатии кнопки инструментальной панели  – «Применить».

Пока измененная запись не сохранена, можно произвести отмену изменений всех ее полей нажатием кнопки инструментальной панели  – «Отмена».

Чтобы добавить новую запись нужно либо нажать на кнопку инструментальной панели  – «Вставить», либо выделить самую последнюю строку таблицы и нажать клавишу «Стрелка вниз». В результате в таблице отобразится новая строка, в которой все ячейки пусты, кроме соответствующей полю ID – в ней значение генерируется триггером и менять его не следует, чтобы не нарушать уникальность значения этого поля.

### Замечание

Перемещение на последнюю запись таблицы производится либо выполнением последовательного движения по таблице вниз с помощью мыши или клавиш движения, либо нажатием кнопки инструментальной панели  – «Последняя».

Удаление выделенной записи справочника производится нажатием кнопки инструментальной панели  – «Удалить».



## Замечание

Элементы справочника, на которые уже созданы ссылки из других таблиц, защищены от удаления.

## 12.6. Назначение (изменение) дополнительного отображаемого столбца

Для выполнения выбора элемента справочника средствами пользовательского интерфейса обычно достаточным бывает визуальное отображение списком значений поля NAME элементов справочника. Однако иногда справочники могут содержать дополнительные поля, визуализация значений которых, наряду с наименованием элемента справочника имеет для пользователей определенное информационное значение для осуществления выбора.

Справочная система SAUMI позволяет представлять справочник списком в два столбца, где в первом столбце отображаются наименование элемента (значение поля NAME), а второй столбец – дополнительный отображаемый – содержит значения некоторого заданного поля таблицы справочника.

Назначение (изменение) дополнительного отображаемого столбца текущего справочника производится выбором в контекстном меню (рис. 12.4) элемента «Назначить/Изменить дополнительный столбец».

## Замечание

Если дополнительный отображаемый столбец для справочника уже ранее назначен, то указанный выше пункт контекстного меню отмечен флажком.

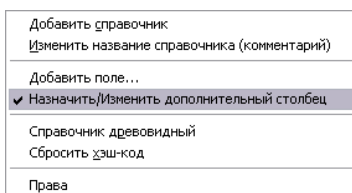


Рисунок 12.10.

Открывающееся в результате окно «Дополнительный столбец при отображении» позволяет задать дополнительный отображаемый столбец по имени поля таблицы справочника, а также ширину в пикселях этого столбца в области отображения.

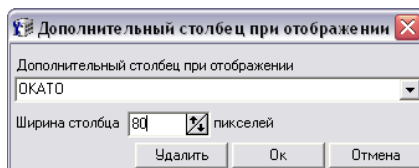


Рисунок 12.11.

К примеру, задание в качестве дополнительного отображаемого столбца ОКАТО в справочнике «Получатели платежей» (таблица RECEPIENTS) приводит к тому, что в окне оформления

обязательства при назначении расчетного счета, осуществляемого выбором значения из этого справочника, помимо наименований получателей отображаются также их коды ОКАТО.

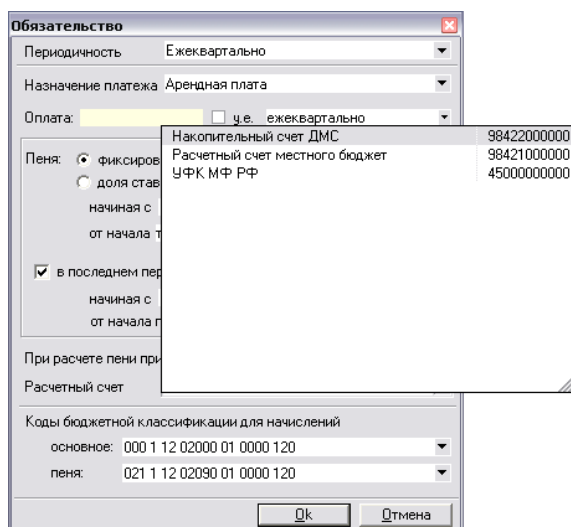


Рисунок 12.12.

## 12.7. Преобразование линейного справочника в древовидный

Древовидная структура справочника позволяет производить в нем более простую навигацию при поиске нужного элемента. При поставке комплекса такую структуру имеют четыре справочника: CLS\_OKOF (справочник кодов ОКОВ), INFGROUPS (справочник групп объектов инженерной инфраструктуры), DOCTYPES (справочник типов документов) и CLS\_KBK (справочник кодов бюджетной классификации).

При наполнении линейного справочника новыми элементами, которые естественным образом группируются, образуя смысловые иерархии, когда несколько элементов по смыслу являются уточненными, более специализированными вариантами некоторого родового элемента – в этом случае может быть целесообразным преобразовать такой справочник в древовидный с помощью «Менеджера справочников».

Чтобы осуществить указанное преобразование, нужно выделить данный справочник как текущий и в контекстном меню (рис. 12.4) выбрать пункт «Справочник древовидный». В результате справочник помечается как древовидный в таблице DICTIONARIES (см. 12.12), а в таблицу преобразовываемого справочника добавляется ссылочное поле ROOT типа NUMBER(27), которое ссылается на запись этого же справочника следующего более высокого уровня иерархии. Корневые элементы справочника должны иметь значение NULL в поле ROOT.

Следуя приведенному правилу можно выстраивать ветви иерархии с помощью «Менеджера справочников», задавая нужные значения в поле ROOT, однако этот способ достаточно трудоемок и подвержен ошибкам. Более приемлемым является перемещение элементов по ветвям с помощью контекстного меню древовидного справочника в главном исполняемом модуле SAUMI, в котором доступен пункт «Перенести в дерево».

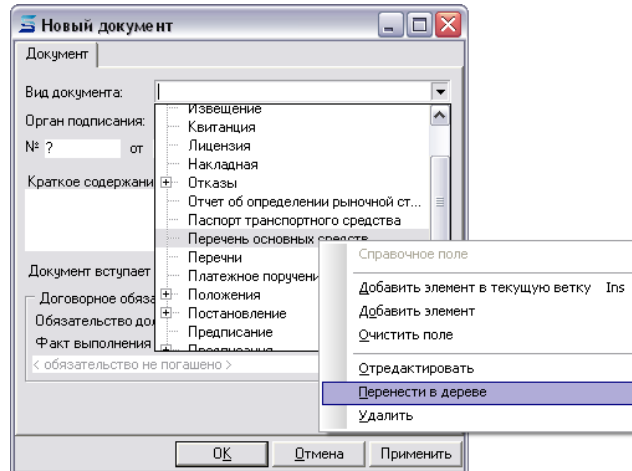


Рисунок 12.13.

Выбор этого пункта контекстного меню текущего элемента открывает диалоговое окно указания примыкающего элемента верхнего уровня, который выбирается из списка двойным щелчком мыши и подтверждается нажатием на кнопку **ОК**.

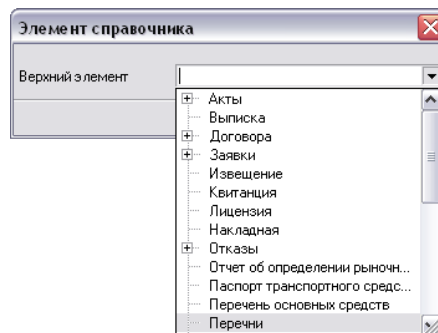


Рисунок 12.14.

## 12.8. Назначение принудительной загрузки отдельных справочников пользовательскими рабочими станциями

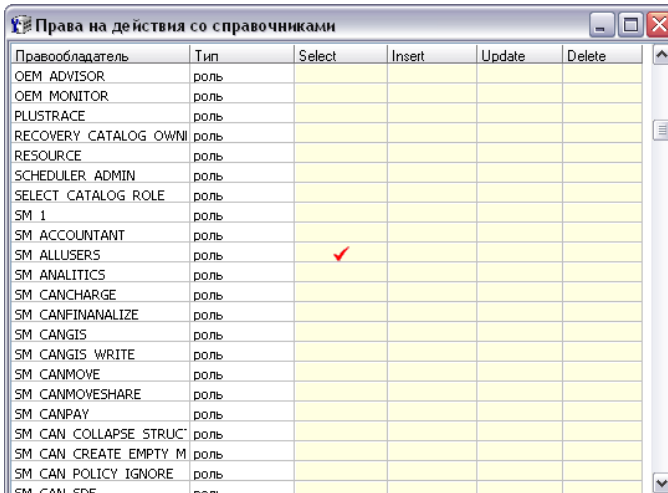
В SAUMI ряд справочников загружаются на локальные рабочие станции во время запуска на них главного исполняемого модуля SAUMI. После сохранения внесенных изменений в базе данных SAUMI локальные копии измененных справочников становятся неактуальными, ставя под сомнение результаты работы пользователей, которые зачастую и не подозревают о произошедших изменениях.

В SAUMI встроен механизм обновления локальных копий справочников, который позволяет автоматически вызывать принудительную повторную загрузку измененных справочников на локальные рабочие станции пользователей. Для активизации этого механизма после внесения в справочник изменений в «Менеджере справочников» администратору следует в контекстном меню (рис. 12.4) этого справочника выбрать пункт «Сбросить хэш-код».

## 12.9. Назначение прав доступа к отдельным справочникам

При вводе в эксплуатацию комплекса следует определиться с разграничением прав доступа к различным справочным таблицам со стороны отдельных пользователей или групп пользователей, объединенных в роли. Такая тактика оберегает справочники от неадекватных изменений и в целом повышает качество хранимой в БД информации. При развертывании комплекса все справочники доступны на чтение из-под роли SM\_ALLUSERS.

Чтобы задать с помощью «Менеджера справочников» права доступа к текущему справочнику, нужно в контекстном меню (рис. 12.4) этого справочника выбрать пункт «Права». В результате откроется окно с таблицей, отображающей права на выполнение операций SELECT (чтение), INSERT (вставка), UPDATE (редактирование), DELETE (удаление) по отношению к записям текущего справочника всех зарегистрированных ролей и пользователей.



Правообладатель	Тип	Select	Insert	Update	Delete
OEM ADVISOR	роль				
OEM MONITOR	роль				
PLUSTRACE	роль				
RECOVERY_CATALOG_OWN	роль				
RESOURCE	роль				
SCHEDULER_ADMIN	роль				
SELECT_CATALOG_ROLE	роль				
SM_1	роль				
SM ACCOUNTANT	роль				
SM ALLUSERS	роль	✓			
SM ANALITICS	роль				
SM CANCHARGE	роль				
SM CANFINANALIZE	роль				
SM CANGIS	роль				
SM CANGIS WRITE	роль				
SM CANMOVE	роль				
SM CANMOVESHARE	роль				
SM CANPAY	роль				
SM CAN COLLAPSE STRUC	роль				
SM CAN CREATE EMPTY M	роль				
SM CAN POLICY IGNORE	роль				
SM CAN SDF	роль				

Рисунок 12.15.


Отметка в виде красного флажка в ячейке таблицы указывает на то, что роли или пользователю, соответствующим строке, разрешено действие, соответствующее заголовку столбца. Разрешение/запрещение действия роли/пользователю осуществляется двойным щелчком мыши в соответствующей ячейке таблицы.

## 12.10. Объединение элементов-дубликатов

Как показывает практика, на начальном этапе освоения комплекса пользователи в силу ряда причин вводят в справочники элементы, которые по смыслу повторяют друг друга, однако воспринимаются компьютерной программой как совершенно разные. Такие элементы называются дубликатами и от них следует избавляться посредством объединения их в один единственный элемент, заменяющий все остальные равнозначные ему дубликаты.

Объединение нескольких элементов справочника производится выделением этих элементов в справочнике щелчком мыши по очередному добавляемому в список элементу с одновременным удержанием нажатой клавиши **Ctrl**.

По завершении формирования списка объединяемых элементов следует навести курсор мыши на элемент этого списка, который будет результатом объединения, и щелкнуть правой кнопкой мыши.

В результате результирующий элемент отмечается значком , примыкающим к элементу слева, и открывается контекстное меню, содержащее пункт «Объединить».

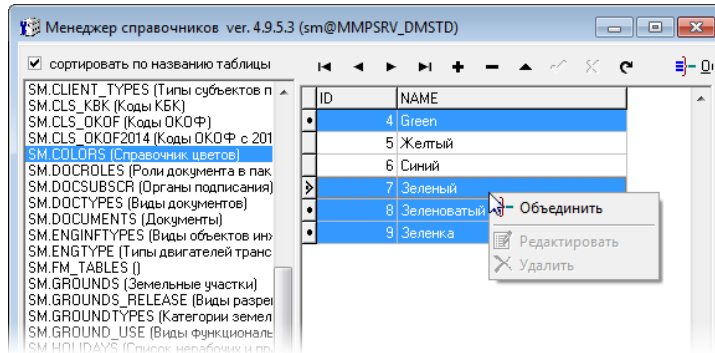


Рисунок 12.16.

Выбор в контекстном меню пункта «Объединить» открывает окно подтверждения,

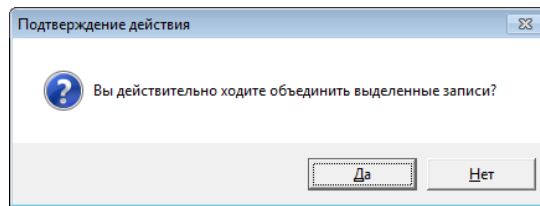


Рисунок 12.17.

после получения которого элементы объединяются в один, отмеченный указанным выше способом.

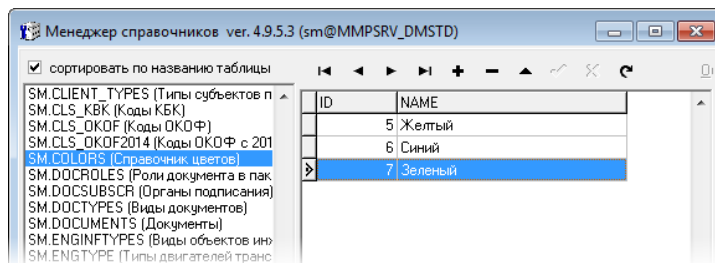


Рисунок 12.18.

### Замечание

Процесс объединения элементов справочника может быть достаточно продолжительным. Следует дождаться окончания объединения выделенных дубликатов.

## 12.11. Автоматическая замена КБК в закрытых договорах

Посредством «Менеджера справочников» может быть реализована возможность автоматически менять КБК пени в закрытых договорах, т.е. когда операция движения уже закрыта, но имеется долг по обязательству этой ОД и на этот долг начисляется пеня. В случае нормативной замены использовавшегося ранее при учете пени КБК на другой может быть задействован соответствующий механизм использования нового КБК в расчетах пени.

Указанная замена производится посредством справочника «Таблица автоматического изменения КБК в начислениях» (таблица CLS\_KBK\_CHANGE), имеющего следующие реквизиты:

- «Дата изменения» – поле CHANGE\_DATE;
- «ID КБК, который следует прекратить использовать в начислениях» – поле FROM\_KBK\_ID;
- «ID КБК, который следует использовать для начислений» – поле TO\_KBK\_ID.

Поля FROM\_KBK\_ID, TO\_KBK\_ID ссылаются на элементы справочника «Коды КБК» (таблица CLS\_KBK).

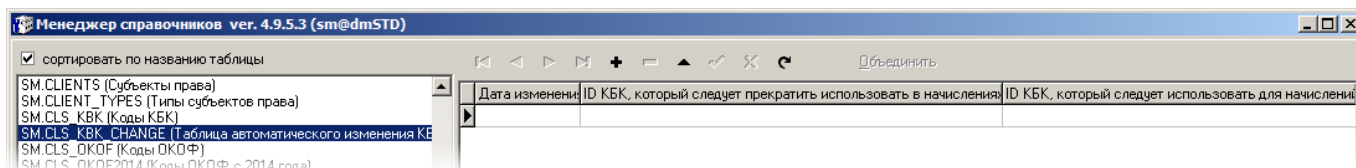


Рисунок 12.19.

Реквизит «Дата изменения» – указывает дату, начиная с которой действующим считается КБК, представленный реквизитом «ID КБК, который следует использовать для начислений», заменяя КБК, представленный реквизитом «ID КБК, который следует прекратить использовать в начислениях».

При создании нового элемента справочника «Таблица автоматического изменения КБК в начислениях» утилитой «Менеджер справочников» идентификаторы элементов из справочника «Коды КБК» могут быть скопированы посредством выделения этих идентификаторов в режиме редактирования во втором запущенном экземпляре этой утилиты и применением сочетания клавиш **Ctrl+C** (команда «Копировать») в окне-источнике, а также последующей команды «Вставить» (сочетание клавиш **Ctrl+V**) в окне-приемнике.

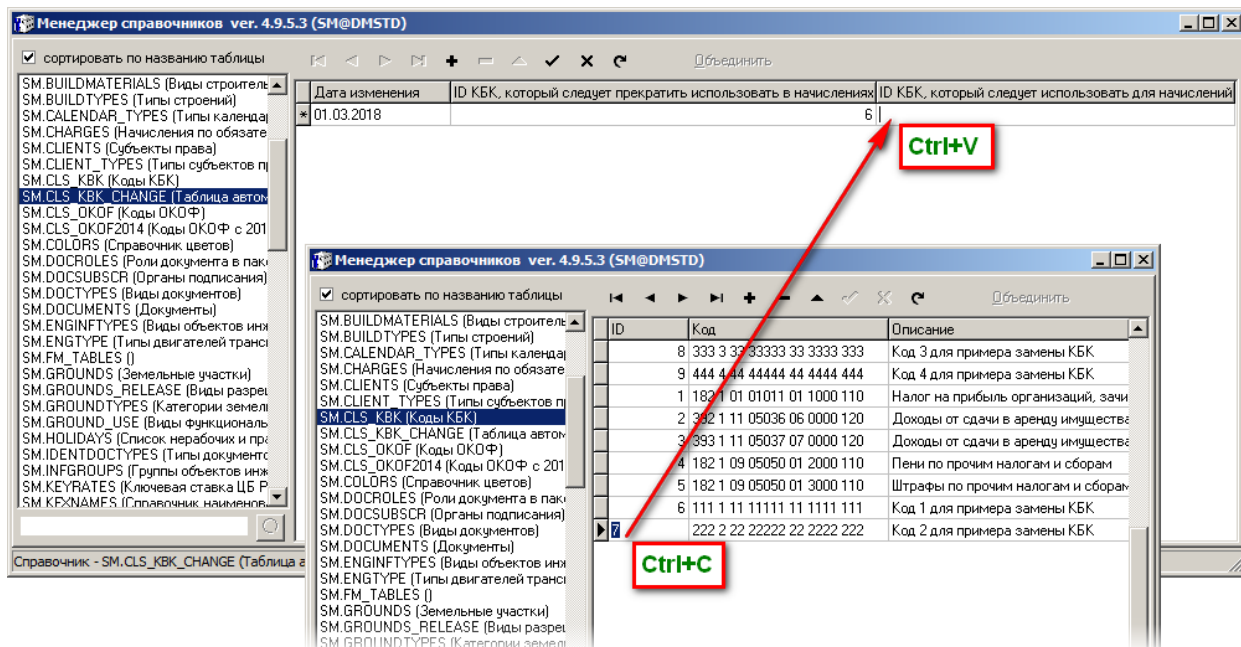


Рисунок 12.20.

Заданные таким способом значения реквизитов «ID КБК, который следует использовать для начислений» и «ID КБК, который следует прекратить использовать в начислениях» гарантируют образование элемента справочника «Таблица автоматического изменения КБК в начислениях» с корректными ссылками на справочник «Коды КБК».

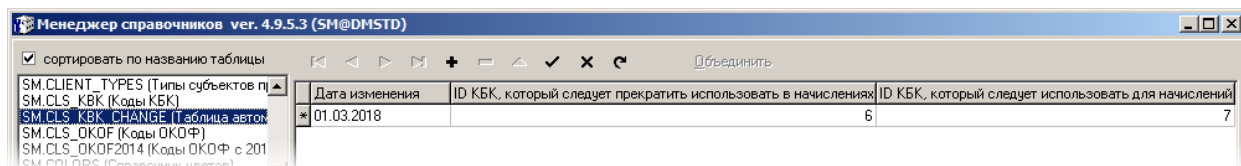


Рисунок 12.21.

С течением времени возможна отмена действия какого-либо ранее введенного на замену КБК и уже его замена, которая также должна быть отражена в данном справочнике соответствующей записью и тогда для конкретной даты расчета эффективный на эту дату КБК определяется возможной последовательностью замен КБК, задаваемой рассматриваемым справочником: КБК1 --- > КБК2 ---> ... ----> КБКn. Представленный на рис. 12.22 набор элементов справочника «Таблица автоматического изменения КБК в начислениях», ссылающихся на элементы справочника «Коды КБК» (рис. 12.20),

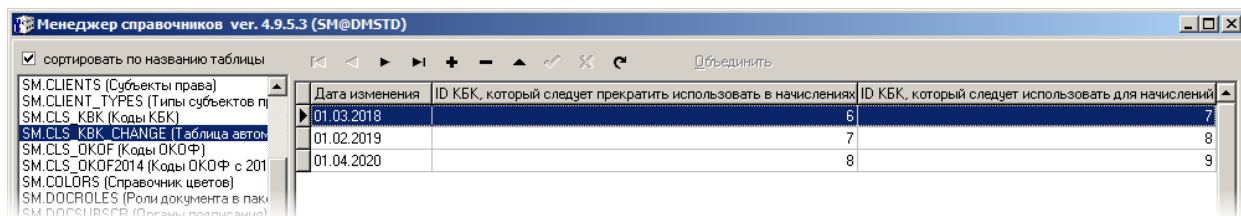


Рисунок 12.22.

иллюстрируют последовательность замен: 111 1 11 11111 11 1111 111 ---> 222 2 22 22222 22 2222 222 ---> 333 3 33 33333 33 3333 333 ---> 444 4 44 44444 44 4444 444 (значения КБК примера некорректны и используются только для наглядности).

При наличии задолженности по обязательству в закрытой ОД во время расчета пени на определенную дату для каждого такого обязательства производится проверка в новом справочнике «Таблица автоматического изменения КБК в начислениях» на наличие записи, у которой значение реквизита «ID КБК, который следует прекратить использовать в начислениях» совпадает с указанным в условиях обязательства КБК, а значение реквизита «Дата изменения» задает дату более раннюю или совпадающую с датой расчета. Обнаружение такого элемента означает необходимость замены КБК, указанного в условиях обязательства, на КБК, эффективный на дату расчета в последовательности замен, задаваемых справочником «Таблица автоматического изменения КБК в начислениях».

Для расширения и интерпретации представленного выше иллюстрационного примера предположим, что некоторая операция движения закрыта датой 31.01.2018, а в ней осталась задолженность, рассчитываемая по которой пеня относилась по условиям обязательства на КБК 111 1 11 11111 11 1111 111. Тогда начисляемая пеня относится с течением времени на следующие КБК, представленные в таблице.

Временной интервал	Применяемый при начислении пени КБК
по 28.02.2018 включительно	111 1 11 11111 11 1111 111

Временной интервал	Применяемый при начислении пени КБК
с 01.03.2018 по 31.01.2019	222 2 22 22222 22 2222 222
с 01.02.2019 по 31.03.2020	333 3 33 33333 33 3333 333
с 01.04.2020	444 4 44 44444 44 4444 444

### Замечание

Представленный выше механизм применим только для закрытых операций движения, т.е. таких, где указана дата окончания последнего периода. В открытых ОД переход на другой КБК осуществляется созданием нового периода, начинающегося датой начала действия этого КБК.

## 12.12. Служебная таблица **DICTIONARIES** – перечень справочников

Все таблицы, используемые в качестве справочников, перечисляются в служебной таблице SM.DICTIONARIES.

Таблица 12.1. Структура таблицы DICTIONARIES

Поле	Тип	Комментарий
TABLE_NAME	VARCHAR2(200)	Название справочной таблицы
HASH_CODE	NUMBER(27)	Хэш-код для кэширования таблицы, заполняется автоматически
IS_STATIC	CHAR(1)	Поле указывает, является ли таблица статичной. Может иметь значения 'Y' и 'N'. В случае если таблица статичная, то она будет кэшироваться на клиентских станциях, и при повторных обращениях при совпадении значений хэш-кодов записи таблицы будут выбираться из локального кэша.
ROOT_FIELD	VARCHAR2(20)	Поле ссылки на уровень выше для древовидных справочников
DISPLAY_COLUMN	VARCHAR2(100)	Название поля справочной таблицы, значения из которой будут отображаться в дополнительном столбце справочника
ID	NUMBER(27)	
IS_SDO	VARCHAR2(1)	
DISPLAY_COLWIDTH	NUMBER(27)	Ширина дополнительной колонки
SEARCH_LINKED	CHAR(1)	Отображать связанные справочники в окне поиска (Y - да, N - нет) - по умолчанию Y



---

# Глава 13. Редактор SAUMI-Pascal

## Содержание

13.1. Вызов редактора SAUMI-Pascal .....	151
13.2. Строка меню и панель инструментов .....	154
13.2.1. Меню Файл .....	154
13.2.2. Меню Правка .....	155
13.2.3. Меню Поиск .....	156

В программный комплекс SAUMI встроен интерпретатор языка, по синтаксису сходного с языком Borland Pascal.

Основная цель введения внутреннего языка – предоставление возможности тонкой алгоритмической настройки поведения программы. Расширение возможностей SAUMI с помощью языка сводится к написанию обработчиков. Обработчики – процедуры и функции SAUMI-Pascal с определенными именами.

Возможности SAUMI-Pascal:

- прямой доступ к БД;
- вызов внешних модулей (DLL-вызовы);
- вывод информации и запрос информации у пользователя.

SAUMI-Pascal включает возможности расширения с построением собственных диалоговых окон. Это позволяет практически неограниченно расширять интерфейс системы, чем и достигается её универсальность.

Так же, с помощью SAUMI-Pascal может быть настроен автоматический расчет арендной платы и других обязательств. При этом **сложность методики расчета арендной платы не ограничена**.

Ввиду большого объема, все материалы о SAUMI-Pascal – описание языка и его возможностей, способы применения и многочисленные примеры – все это представлено в виде html-страниц на сайте <http://support.saumi.ru>, а также на дистрибутивном диске SAUMI в разделе документации.

## 13.1. Вызов редактора SAUMI-Pascal

Текст процедур и функций, реализованный с помощью языка SAUMI-Pascal (код SAUMI-Pascal), хранится в БД SAUMI. Для редактирования кода SAUMI-Pascal применяется специальная утилита комплекта администратора SAUMI – «Редактор SAUMI-Pascal».

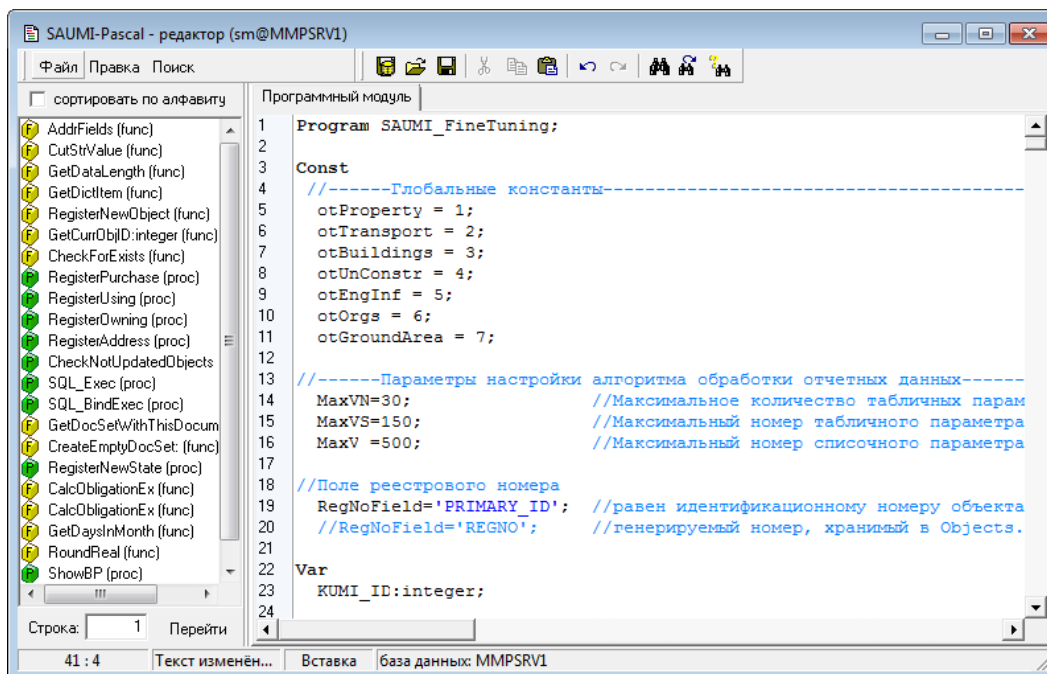


Рисунок 13.1.

Вызов утилиты доступен как из системного меню **Пуск | Все программы | Комплект администратора SAUMI | Редактор SAUMI-Pascal**, так и непосредственно из меню SAUMI «Дополнения», доступного для пользователя SM.

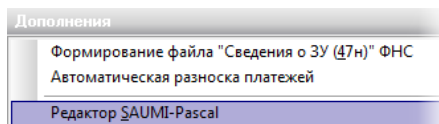


Рисунок 13.2.

На пользовательских компьютерах при работе обычных пользователей SAUMI (не под учетной записью SM) может быть вызван на исполнение ошибочный код SAUMI-Pascal, который при выявлении вызывает прерывание исполняемой операции, сопровождаемое выводом окна с сообщением, описывающим ошибку, а также предложением открыть редактор SAUMI-Pascal.

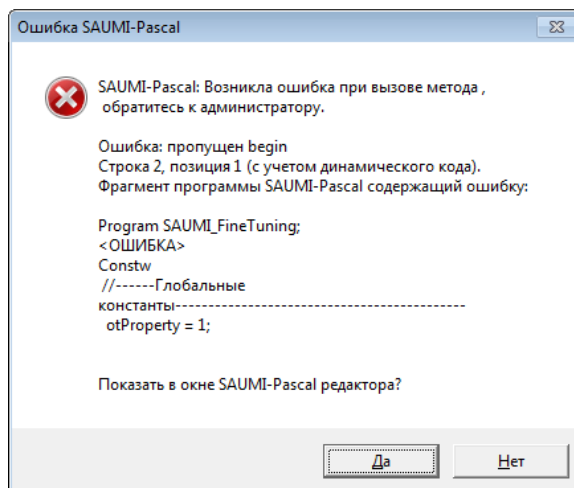


Рисунок 13.3.

После нажатия кнопки **Да** для вызова редактора необходимо ввести пароль пользователя SM (администратора SAUMI).

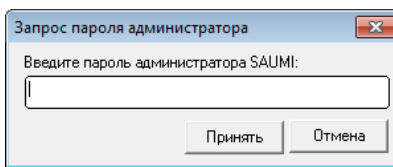


Рисунок 13.4.

Вызов редактора из пользовательской сессии SAUMI позволяет администратору быстрее находить ошибки в программном коде SAUMI-Pascal, используя реальные данные.

### Замечание

Более предпочтительным является вызов редактора непосредственно из SAUMI, так как это позволяет более оперативно проверить выполнение измененного кода.

Окно редактора открывается с загруженной в него текущей версией кода SAUMI-Pascal, хранимого в БД. В окне редактора имеются следующие элементы:

- строка меню;
- панель инструментов;
- панель списка программных единиц;
- средство сортировки списка программных единиц;
- панель редактирования программного кода;
- средство быстрого перехода;
- строка состояния.

## 13.2. Строка меню и панель инструментов

В верхней части окна редактора располагается строка меню и панель инструментов, позволяющие применять традиционные операции редактирования и сохранять результаты их применения.

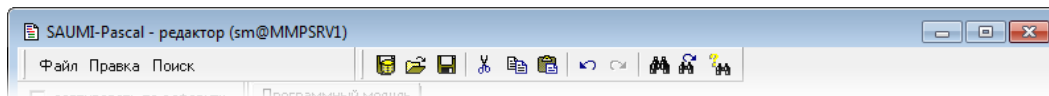


Рисунок 13.5.

### 13.2.1. Меню Файл

Меню «Файл» содержит пункты, управляющие загрузкой кода в редактор и выгрузкой из него, а также пункт завершения работы редактора.

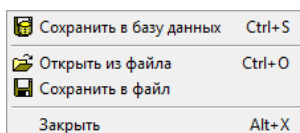



Рисунок 13.6.

Пункт «Сохранить в базу данных» осуществляет выгрузку программного кода SAUMI-Pascal из редактора в БД SAUMI. Собственно применение этого пункта есть конечный положительный результат работы в описываемом редакторе, так как главный исполняемый модуль SAUMI интерпретирует именно загруженный в БД SAUMI программный код SAUMI-Pascal. Для выполнения этой операции также можно использовать сочетание клавиш **Ctrl+S**, а также кнопку  на панели инструментов.

### Замечание

Выгрузка из редактора в БД программного кода полностью уничтожает код SAUMI-Pascal, содержащийся в БД, заменяя его выгружаемым. По этой причине рекомендуется сохранять версию кода SAUMI-Pascal в файл перед каждым существенным его изменением, чтобы иметь возможность восстановить работоспособный код.

Пункт «Сохранить в файл» позволяет сохранять код SAUMI-Pascal, содержащийся в окне редактора, в простой текстовый файл. Открывающееся стандартное диалоговое окно сохранения файла подставляет в качестве имени по умолчанию для файла имя `sp_global.pas`.

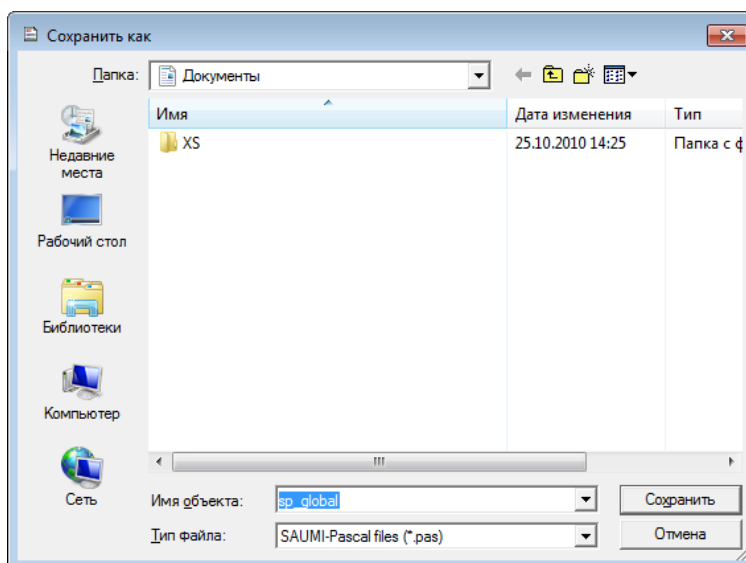




Рисунок 13.7.

Сохранение в файл также может быть вызвано нажатием кнопки  панели инструментов.

### Замечание

В дополнение к предыдущему замечанию для различия версий рекомендуется добавлять к имени файла идентифицирующий версию текст и/или сохранять файлы в папки с информативными именами. Например: `sp_global_20101109.pas`, `sp_global_20101109_рабочий.pas`.

Пункт «Открыть из файла» применяется для загрузки в редактор кода SAUMI-Pascal из текстового файла. Указанный пункт также открывает стандартное диалоговое окно выбора файла. Описываемая операция позволяет загрузить в редактор ранее сохраненный в файл код SAUMI-Pascal для последующей работы с ним, в частности для сохранения в БД SAUMI работоспособной версии кода. Также выполнение описываемой операции вызывается сочетанием клавиш **Ctrl+O** и нажатием кнопки  панели инструментов.

Пункт «Закрыть», равносильный нажатию клавиш **Alt+X**, вызывает закрытие редактора SAUMI-Pascal. В случае если в тексте производились изменения, предлагается сохранить код SAUMI-Pascal в базе данных SAUMI

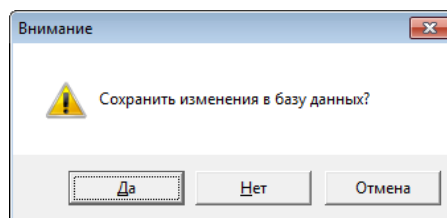


Рисунок 13.8.

### 13.2.2. Меню Правка

Меню «Правка» по составу функциональности аналогично многим приложениям, использующим буфер обмена Windows Clipboard, и позволяет совершать операции по копированию, вырезанию

и вставке блоков текста, а также выполнения отмены предыдущей операции и повторного выполнения отмененной операции.

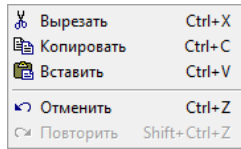


Рисунок 13.9.

Операции, вызываемые выбором пунктов в меню «Правка», также могут быть выполнены посредством нажатия соответствующих кнопок на панели инструментов, а также клавишами, приведенными на рис. 13.9.

### Замечание

Типичной операцией редактирования является вставка в общий массив кода SAUMI-Pascal новой программной единицы (процедуры или функции), полученной, например, из службы поддержки SAUMI. При вставке такой программной единицы следует помнить, что ее размещение по тексту не должно входить в какую-либо уже имеющуюся программную единицу, а также предшествовать основному блоку программы, заключенному между последними операторными скобками `Begin ... End.`

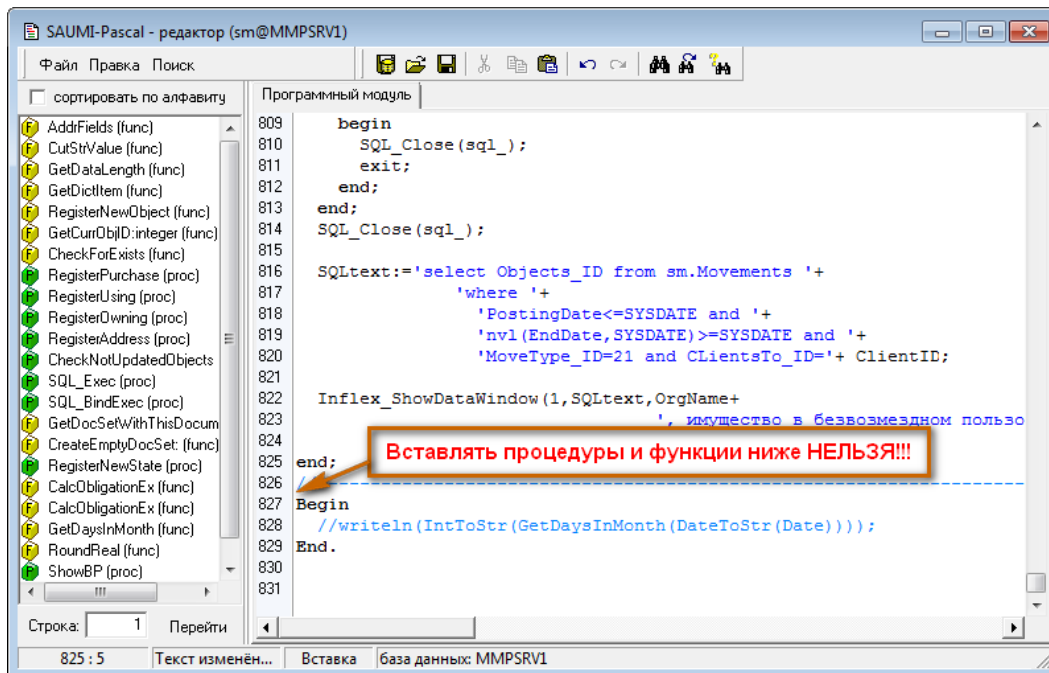


Рисунок 13.10.

### 13.2.3. Меню Поиск

Меню «Поиск» выполняет стандартные для многих приложений Windows функции:

- поиск по заданному фрагменту текста;

- поиск следующего вхождения ранее заданного фрагмента текста;
- поиск и замена вхождения одного фрагмента текста другим заданным фрагментом

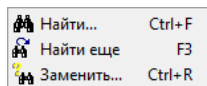


Рисунок 13.11.

Пункты меню «Найти...» и «Заменить...» открывают одно и то же диалоговое окно задания параметров поиска и замены, у которого в секции **Operation** в первом случае выбран зависимый переключатель «Искать», а во втором «Искать и заменять».

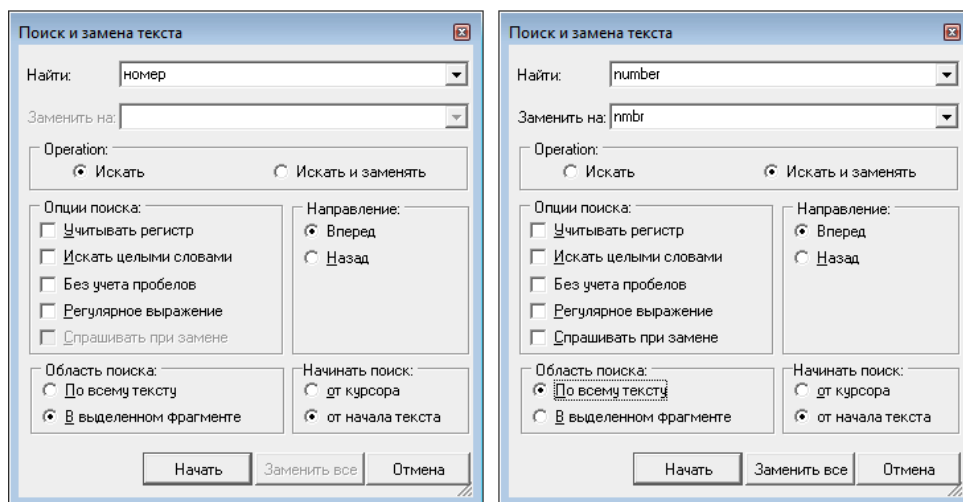


Рисунок 13.12.

Параметры поиска также позволяют менять стартовую позицию, границы и направление поиска, а также ряд различных характеристик (опций) поиска, позволяющих расширить интерпретацию совпадения заданного поискового фрагмента с фрагментом текста кода SAUMI-Pascal.

---

# Глава 14. Работа с расчетными коэффициентами

## Содержание

14.1. «Менеджер коэффициентов» .....	160
14.1.1. Создание нового коэффициента .....	160
14.1.2. Удаление коэффициента .....	161
14.1.3. Изменение описания коэффициента .....	162
14.1.4. Добавление периода .....	162
14.1.5. Удаление периода коэффициента .....	164
14.1.6. Редактирование значений коэффициента .....	164
14.1.7. Сортировка расчетных коэффициентов .....	167
14.2. Использование коэффициентов при автоматическом расчете .....	169
14.2.1. SAUMI-Pascal .....	169
14.2.2. .NET .....	170
14.2.3. Ошибки .....	170

Для унификации и упрощения описания алгоритма автоматического расчета обязательств, была разработана технология работы с коэффициентами.

Главным принципом в данной технологии является – предоставление единого механизма описания коэффициентов с учетом возможности изменения их значений по времени, последующего управления значениями и удобным механизмом обращения к значениям из SAUMI-Pascal и внешних .NET модулей.

Коэффициенты при расчете обязательств могут быть определены совокупностью от одной до трех характеристик, которые также сопоставлены с элементами некоторых справочников. Для таких коэффициентов применяется соответствие каждого сочетания характеристик некоторому значению коэффициента.

### Пример 14.1. Коэффициент, зависимый от одной характеристики

Коэффициент назначения использования объекта. В данном случае в качестве характеристики выступает назначение использования объекта, определяемое в виде справочного реквизита движения объекта. Таким образом, каждому элементу справочника сопоставляется некоторое значение данного коэффициента.

### Пример 14.2. Коэффициент, зависимый от двух характеристик

Коэффициент социальной значимости. Данный коэффициент зависит от двух характеристик: от назначения использования объекта и района города, в связи с тем, что для некоторых районов города, в которых, например, отсутствуют аптеки, необходимо создать привлекательные условия аренды этому виду деятельности. То есть, в данном случае, значение коэффициента определяется двумя характеристиками, каждая из которых является справочным реквизитом.

В некоторых случаях значение коэффициента определяется не некоторым явно определенным справочником, а набором диапазонов значений, например коэффициент, значение которого зависит от высоты потолков. В данном случае все равно целесообразно создать справочник, в



котором ввести два дополнительных поля – нижнюю и верхнюю границы высоты потолка для каждого значения коэффициента.

Разумеется, в этом случае, прежде чем запрашивать значение коэффициента, придется определить, какой элемент справочника применим в конкретном случае.

Использование такого подхода оправдано в связи с тем, что значение коэффициентов могут меняться во времени, и если описывать условия непосредственно в алгоритме, для каждого изменения значения коэффициентов, придется вносить изменения в алгоритм.

К тому же, поскольку значение коэффициентов описывается методиками расчета, утверждаемыми законодательными властями, эти значения могут периодически изменяться.

## 14.1. «Менеджер коэффициентов»

С помощью менеджера коэффициентов можно выполнить следующие действия:

- создание нового коэффициента;
- добавление нового периода коэффициента;
- редактирование значений коэффициента;
- удаление коэффициента.

**Коэффициент** – название переменной, применяемой в расчете обязательства согласно некоторой методике. Значение переменной при расчете определяется в зависимости от прямых и косвенных характеристик сделки – условий использования имущества, характеристик имущества и субъекта. Значение может задаваться с точностью до шести десятичных знаков после запятой.

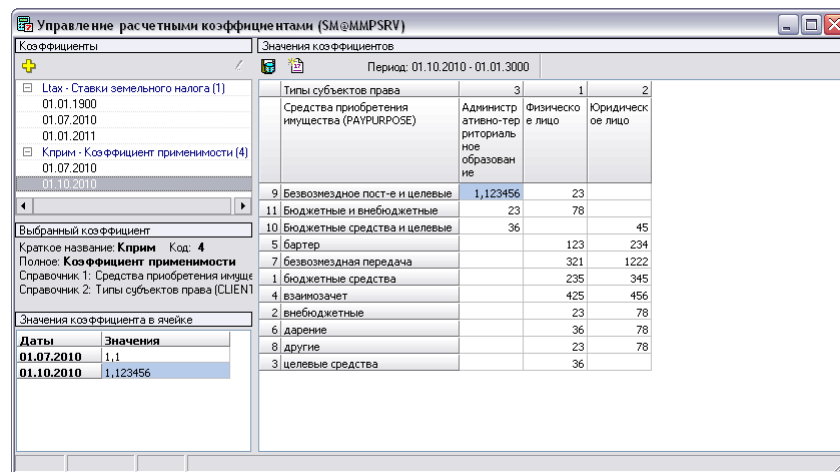



Рисунок 14.1. Вид окна «Менеджера коэффициентов»

### 14.1.1. Создание нового коэффициента

Создание нового коэффициента вызывается либо нажатием кнопки  на панели инструментов, либо выбором пункта «Добавить новый коэффициент» в контекстном меню, вызываемом щелчком правой кнопки в области списка коэффициентов.

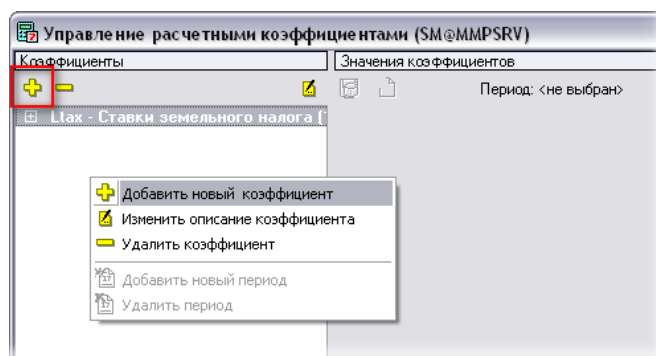


Рисунок 14.2.

При добавлении коэффициента открывается окно настройки коэффициента.

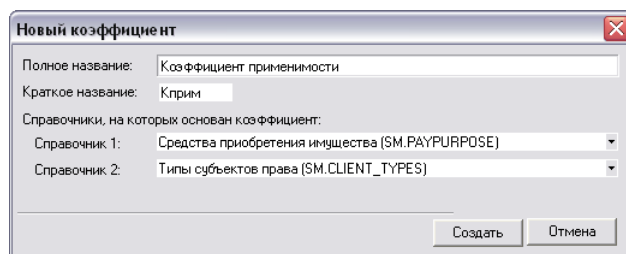


Рисунок 14.3.


**Полное название** – это то название, которое указывает назначение, смысл создаваемого коэффициента.

**Краткое название** – это название, посредством которого к коэффициенту можно обращаться через встроенные механизмы SAUMI-Pascal. Это название должно быть уникальным.

**Справочники 1, 2** – это справочники SAUMI, от которых зависит значение коэффициента. Количество заполненных справочников определяет тип создаваемого коэффициента.

После заполнения всех полей и выбора хотя бы одного справочника в окне создания нового коэффициента нужно нажать кнопку **Создать**.

### 14.1.2. Удаление коэффициента

Удаление коэффициента может быть вызвано либо нажатием кнопки , либо выбором пункта «Удалить коэффициент» в контекстном меню списка коэффициентов; предварительно коэффициент должен быть выделен в списке.

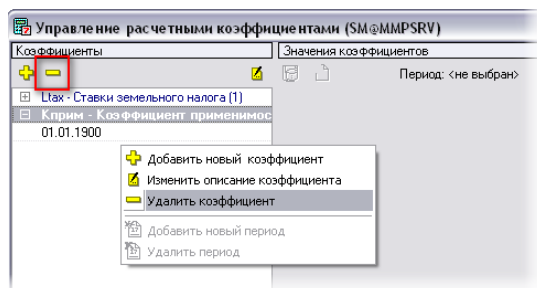


Рисунок 14.4.

Перед удалением выводится предупреждающее сообщение, которое позволяет либо отказать от удаления, либо подтвердить его.

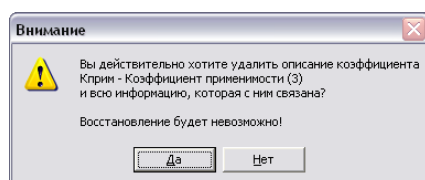



Рисунок 14.5.

### 14.1.3. Изменение описания козэффициента

Изменение описания козэффициента вызывается либо нажатием кнопки , либо выбором пункта «Изменить описание козэффициента» в контекстном меню списка козэффициентов, причем козэффициент должен быть предварительно выделен в списке.

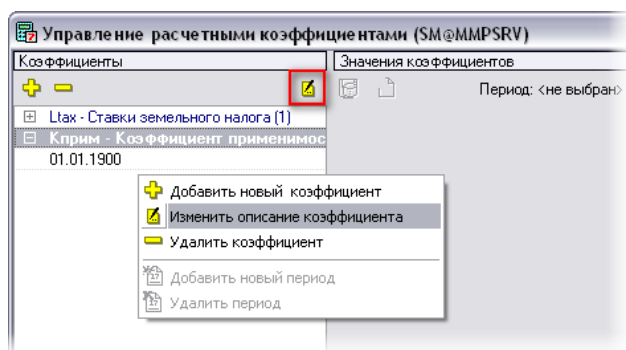


Рисунок 14.6.

В результате открывается окно настройки козэффициента (рис. 14.3), в котором производятся необходимые изменения.

### 14.1.4. Добавление периода

Сразу же после создания козэффициента создается первый период, датой начала действия которого назначается 1-е января 1900 года. Перечень созданных для козэффициента периодов раскрывается

двойным щелчком по строке названия коэффициента. Повторный двойной щелчок по строке названия коэффициента сворачивает раскрытый перечень периодов коэффициента

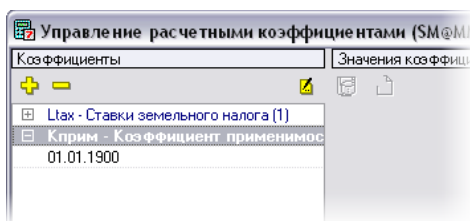



Рисунок 14.7.

Новый период обычно создается, когда необходимо изменить значение коэффициента хотя бы для одного элемента справочника (одномерный коэффициент) или пары элементов (двумерный коэффициент) в силу вступления в действие новых нормативов, на основе которых создан коэффициент. При этом значения коэффициента предыдущих периодов сохраняются и применяются для расчетов, относящихся ко времени, предшествующему началу нового периода коэффициента.

Чтобы создать новый период коэффициента необходимо сначала раскрыть перечень периодов и выделить в нем некоторый период. После этого создание нового периода вызывается либо нажатием активизировавшейся кнопки , либо выбор из контекстного меню элемента «Добавить новый период».

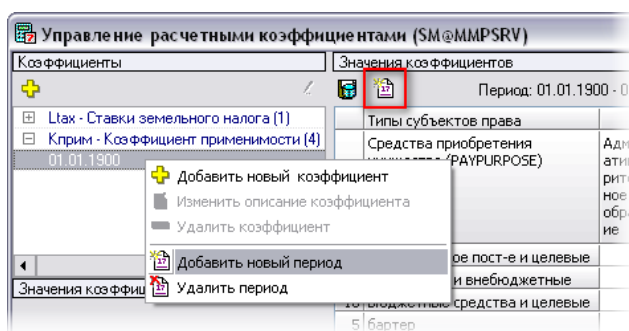


Рисунок 14.8.

В открывающемся следом окне, необходимо задать дату начала создаваемого периода и нажать кнопку «Применить».

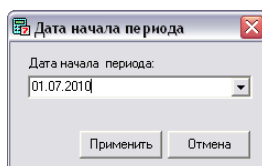


Рисунок 14.9.

В новый период при его создании переносятся значения из периода предыдущего и их нужно актуализировать редактированием.

Виды разрешенного использования земель (GROUNDS_RELEASE)	01.01.1900	01.07.2010
34 Болота	1	1
45 Земельные участки под автостоянками, парковками	2	2
41 Земли запаса в границах населенных пунктов	3	3
42 Земли запаса вне границ населенных пунктов	4	4
30 Земли лесные	5	5
4 Земли под древесно-кустарниковой растительностью, не	6	6
2 Земли под зданиями (строениями), сооружениями	7	7
11 Земли под лесами (в том числе городскими лесами)		
20 Земли под лесом, предоставленные для нужд обороны (не		
13 Земли под обособленными водными объектами, находящимися в		
39 Земли под поверхностными водоемами		

Рисунок 14.10. Дублирование значений из предыдущего периода в новый период

### 14.1.5. Удаление периода коэффициента

Период коэффициента может быть удален. Удаление производится выделением данного периода и выбором в контекстном меню пункта «Удалить период».

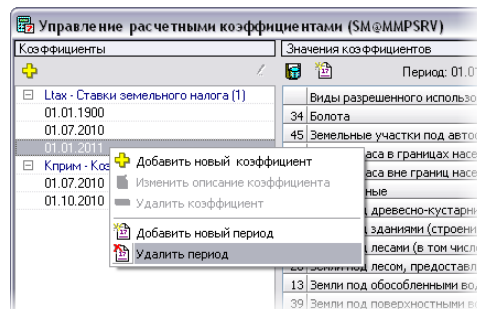


Рисунок 14.11.

В результате выводится окно подтверждения удаления.

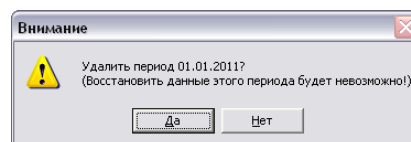


Рисунок 14.12.

### Замечание

Единственный период коэффициента удалять не следует – в противном случае коэффициент без периодов становится совершенно бесполезной конструкцией: к нему нельзя добавить ни один новый период и его можно только удалить.

### 14.1.6. Редактирование значений коэффициента

В древовидном списке коэффициентов и их периодов необходимо выбрать период интересующего коэффициента, при этом в табличной области отображается таблица значений. Вид таблицы

для одномерных и двумерных коэффициентов различается. Для одномерных коэффициентов отображаются в одной таблице разными столбцами значения всех созданных периодов коэффициента.

Управление расчетными коэффициентами (SM@MMPSRV)

Значения коэффициентов

Период: 01.07.2010 - 31.12.2010

Виды разрешенного использования земель (GROUNDS_RELEASE)	01.01.1900	01.07.2010	01.01.2011
34 Болота	1,2	1	1
45 Земельные участки под автостоянками, парковками	2,1	2	2
41 Земли запаса в границах населенных пунктов	3,4	3	3
42 Земли запаса вне границ населенных пунктов	0,4	4	4
30 Земли лесные	0,5	5	5
4 Земли под древесно-кустарниковой растительностью, не	0,6	6	6
2 Земли под зданиями (строениями), сооружениями	1,7	7	7
11 Земли под лесами (в том числе городскими лесами)	2,8	2,5	2,5
20 Земли под лесом, предоставленные для нужд обороны (не	3,1		3,3
13 Земли под обособленными водными объектами, находящимися в			2,2
39 Земли под поверхностными водоемами	1,1		
38 Земли под поверхностными водотоками и водохранилищами на		2,1	
3 Земли под строительством зданий (строений), сооружений	3,2		
44 Земли с неустановленной (неизвестной) категорией			
28 Земли, занятые залежами лечебных грязей			
35 Земли, занятые каменистыми россыпями			
25 Земли, занятые маркированными трассами			
27 Земли, занятые минеральными источниками			
26 Земли, занятые учебно-туристическими тропами			
14 Земли-паша			

Выбранный коэффициент  
 Краткое название: **Ltax** Код: **1**  
 Полное: **Ставки земельного налога**  
 Справочник 1: Виды разрешенного использо  
 Справочник 2:

Рисунок 14.13. Одномерный коэффициент

Двумерный коэффициент представлен таблицей, значение каждой ячейки которой сопоставляется некоторому элементу первого справочника (строка) в описании коэффициента и элементу второго справочника (столбец).

Управление расчетными коэффициентами (SM@MMPSRV)

Значения коэффициентов

Период: 01.10.2010 - 01.01.3000

Типы субъектов права	3	1	2
Средства приобретения имущества (PAYPURPOSE)	Административно-территориальное образование	Физическое лицо	Юридическое лицо
9 Безвозмездное пост-е и целевые		23	
11 Бюджетные и внебюджетные	23		
10 Бюджетные средства и целевые	36		45
5 бартер		23	43
7 безвозмездная передача			
1 бюджетные средства		235	
4 взаимозачет		453	
2 внебюджетные			
6 дарение			
8 другие			
3 целевые средства			

Значения коэффициента в ячейке

Даты	Значения
01.07.2010	234
01.10.2010	

Выбранный коэффициент  
 Краткое название: **Кприм** Код: **4**  
 Полное: **Коэффициент применимости**  
 Справочник 1: Средства приобретения имуще  
 Справочник 2: Типы субъектов права (CLIEN)

Рисунок 14.14. Двумерный коэффициент

Редактировать значения можно непосредственным вводом данных в ячейки таблицы, а также копированием табличных блоков данных, как из программы Excel,

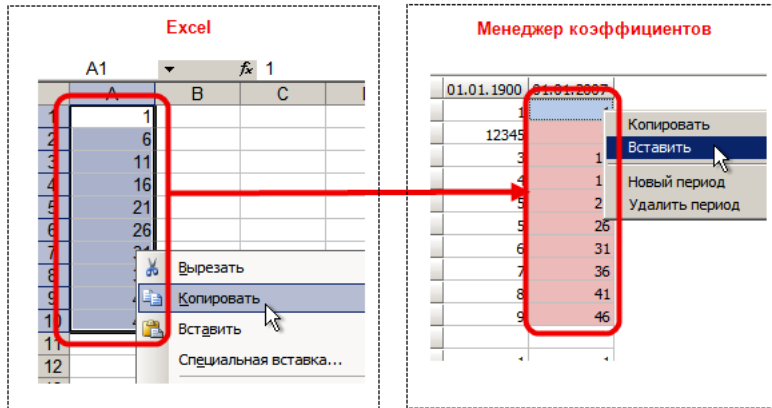


Рисунок 14.15.

так и непосредственно из таблицы некоторого периода некоторого коэффициента, включая текущие, посредством контекстного меню (выделение блока производится щелчком мыши в одном из углов выделяемого блока с последующим щелчком мыши с удержанием клавиши **Shift** в противоположном углу).

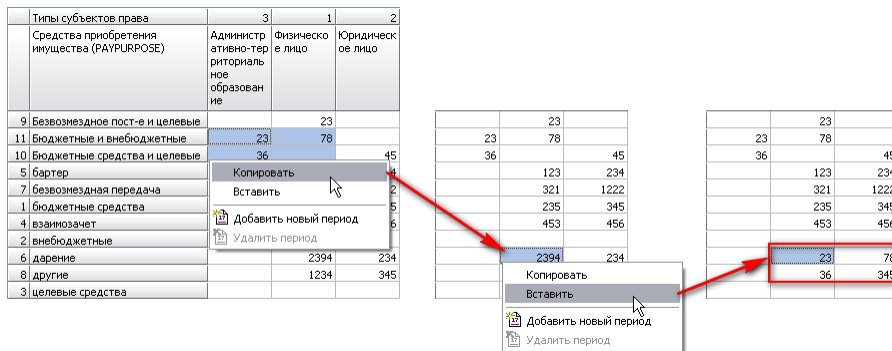


Рисунок 14.16.

При изменении значения двумерного коэффициента в одной из ячеек в панели хронологии измененное значение выделяется красным цветом и надписью «не сохр.». Указанное выделение поддерживается до тех пор, пока изменения не будут сохранены в БД.

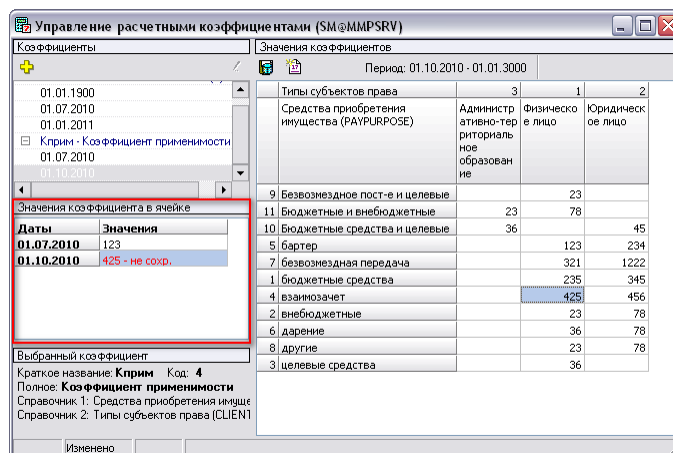


Рисунок 14.17.

Сохранение значений коэффициента в БД SAUMI производится нажатием кнопки , которая становится активной при выборе некоторого периода некоторого коэффициента.

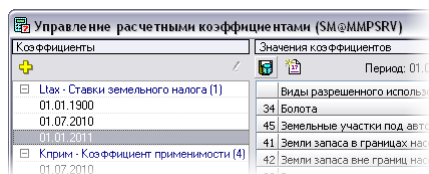


Рисунок 14.18.

При изменении выбора коэффициента или периода выводится сообщение, предлагающее сохранить значения коэффициента.

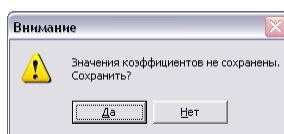


Рисунок 14.19.

### 14.1.7. Сортировка расчетных коэффициентов

Строки отображаемой таблицы коэффициентов могут быть отсортированы по значениям одного из столбцов.

В случае одномерного коэффициента строки могут быть отсортированы либо по значениям идентификаторов элементов справочника, к которому привязан коэффициент, либо по наименованиям этих элементов, либо по значениям коэффициента. Чтобы отсортировать таблицу по значениям одного из указанных столбцов по убыванию необходимо щелкнуть мышью в заголовочной ячейке этого столбца.



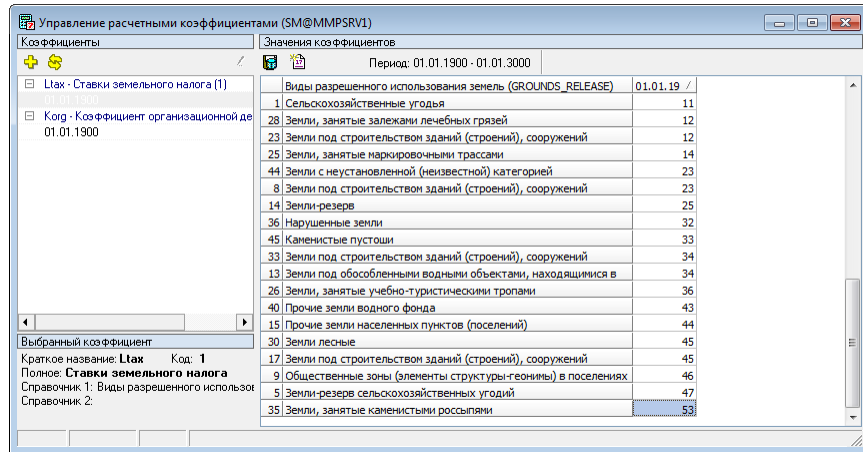


Рисунок 14.20. Сортировка по значению коэффициента

Повторный щелчок в этой же ячейке меняет порядок сортировки на противоположный.

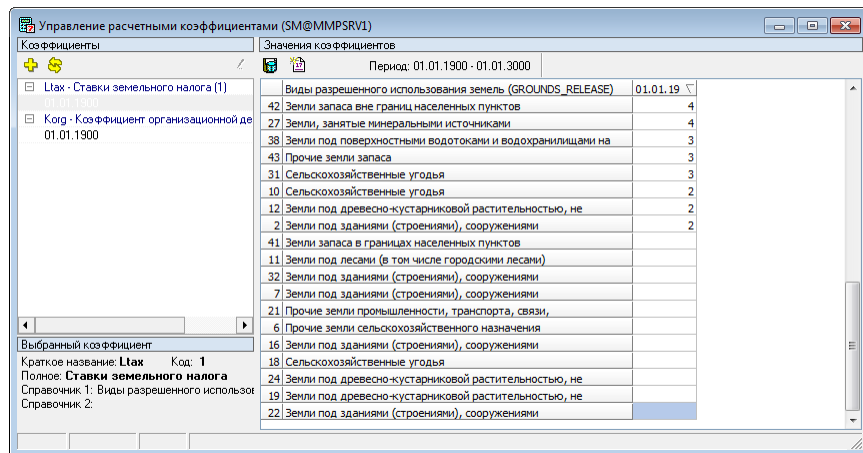


Рисунок 14.21. Убывающая сортировка по значению коэффициента

Таблица, представляющая двумерный расчетный коэффициент также может быть отсортирована либо по значениям идентификаторов элементов справочника, заданного первым в описании коэффициента, либо по наименованиям элементов этого первого справочника, либо по значениям коэффициентов, соответствующих какому-либо одному элементу справочника, заданного в описании коэффициента вторым.

Чтобы отсортировать таблицу по значениям одного из указанных столбцов по убыванию необходимо щелкнуть мышью в ячейке этого столбца, являющейся второй в заголовочной части таблицы.

Управление расчетными коэффициентами (SM@MMPSRV1)

Значения коэффициентов

Период: 01.01.1900 - 01.01.3000

Организационно-правовые	882	681	541	461	881	641
Виды деятельности (ACTIVITYTYPES)	Благотворительный фонд	ГСК /	ГУ	ГУП	Государственная инспекция	ЗАО
55 Муниципальные услуги	9			5		
44 Прокат			4			
42 Ландшафтные работы			7		3	
51 Банковская деятельность	1	1	1	1	1	1
41 Озеленение	2				5	8
57 Журналистика	2			5		
50 Промышленность	3			4		
54 Госуслуги	3		6			7
49 Торговля мелкооптовая	1	3	6	2	5	2
53 Отправление культов	4	4	5			
48 Торговля розничная		5				
58 Природоохранная деятельность		5		5		6
46 Общепит		5	6			
43 Гостиничные услуги		6	3	8		
47 Торговля оптовая		8				

Рисунок 14.22. Сортировка значений коэффициента по возрастанию для организационно-правовой формы ГСК

Повторный щелчок в этой же ячейке меняет порядок сортировки на противоположный.


Управление расчетными коэффициентами (SM@MMPSRV1)

Значения коэффициентов

Период: 01.01.1900 - 01.01.3000

Организационно-правовые	882	681	541	461	881	641
Виды деятельности (ACTIVITYTYPES)	Благотворительный фонд	ГСК /	ГУ	ГУП	Государственная инспекция	ЗАО
52 Благотворительность	5	65				
45 Строительство и ремонт		45				
56 Юридические услуги		32				
47 Торговля оптовая		8				
43 Гостиничные услуги		6	3	8		
46 Общепит		5	6			
58 Природоохранная деятельность		5		5		6
48 Торговля розничная		5				
53 Отправление культов		4	4	5		
49 Торговля мелкооптовая	1	3	6	2	5	2
54 Госуслуги		3	6			7
50 Промышленность		3		4		
57 Журналистика		2		5		
41 Озеленение		2			5	8
51 Банковская деятельность	1	1	1	1	1	1

Рисунок 14.23. Сортировка значений коэффициента по убыванию для организационно-правовой формы ГСК

Нажатие кнопки инструментальной панели  – «Обновить» – загружает снова из базы данных сохраненные ранее значения коэффициентов, отменяя сделанные после сохранения изменения.

## 14.2. Использование коэффициентов при автоматическом расчете

### 14.2.1. SAUMI-Pascal

В SAUMI-Pascal для доступа к значениям коэффициентов имеются три функции `GetFX1`, `GetFX2`, `GetFX3` для нахождения расчетных коэффициентов, зависящих от одного, двух и трех параметров соответственно в заданный период времени.

Перечисленные функции объявлены следующим образом.

```
function GetFX1(KFX_ShortName:string; Dict1_ID:Integer; ForDate:TDateTime):Double;

function GetFX2(KFX_ShortName:string; Dict1_ID:Integer; Dict2_ID:Integer; ForDate:TDateTime):Double;

function GetFX3(KFX_ShortName:string; Dict1_ID:Integer; Dict2_ID:Integer; Dict3_ID:Integer;
                ForDate:TDateTime):Double;
```

Здесь параметру `KFX_ShortName` в качестве значения должно передаваться краткое название используемого коэффициента (см. 14.1.1), а параметрам `Dict1_ID`, `Dict2_ID`, `Dict3_ID` передаются значения идентификаторов конкретных элементов справочников, соответствующих используемому коэффициенту. Параметром `ForDate` задается дата, для которой требуется определить значение коэффициента.

## Замечание

«Менеджер коэффициентов» не поддерживает интерактивный ввод коэффициентов, зависящих от трех параметров, тем не менее, ведение таких коэффициентов возможно непосредственно в таблицах `SM.KFXNAMES` и `SM.KFXVALUES`.

### Пример 14.3. Обращение к функциям

```
var
  KFX1,
  KFX2:real;
  ...
begin
  ...

  KFX1:=GetFX1('ПК2', 88, StrToDate('01.02.2009'));
  ...

  KFX2:=GetFX2('Кприм', 3155, 281, Date);
  ...
end;
```

## 14.2.2. .NET

При программировании в .NET для доступа к значениям коэффициентов нужно использовать библиотеку `Inflex.ENV.dll` и класс `InflexCLR.Environment.SaumiPascalFunc`. В этом классе содержится три метода аналогичных методам доступным из `SAUMI-Pascal`.

```
public static double GetFX1(string KFX_ShortName, int Dict1_ID, DateTime ForDate)

public static double GetFX2(string KFX_ShortName, int Dict1_ID, int Dict2_ID, DateTime ForDate)

public static double GetFX3(string KFX_ShortName, int Dict1_ID, int Dict2_ID, int Dict3_ID,
                             DateTime ForDate)
```

Имена и назначение параметров совпадают с именами параметров одноименных `SAUMI-Pascal` функций (см. подраздел выше).

### Пример 14.4. Обращение к методу

```
double d = SaumiPascalFunc.GetFX2("Кисп", 28, 2721, DateTime.Parse("01.06.2009"));
```

## 14.2.3. Ошибки

При использовании функций доступа к значениям коэффициентов произойдет ошибка, если не найден коэффициент с указанным кратким названием или найдено несколько таких коэффициентов.

Также ошибка произойдет, если будет найдено несколько значений коэффициента. Такое возможно, если для поиска значений коэффициента будет использоваться функция, не соответствующая по параметрам коэффициенту. Например, для коэффициента с двумя параметрами будет использована функция `GetFX1` с одним параметром.

Если не найдено ни одного значения, то ошибка не произойдет и будет возвращено значение 0 (нуль).

---

# Глава 15. «Политика операций»

## Содержание

15.1. Вкладка «Параметры» .....	173
15.1.1. Параметр «Непрерывный» .....	174
15.1.2. Параметр «Реестр» .....	174
15.1.3. Параметр «Площади» .....	174
15.1.4. Параметр «Обязательства» .....	175
15.1.5. Параметр «Авторасчет» .....	175
15.1.6. Параметр «Ценные бумаги» .....	175
15.1.7. Параметр «Назначения» .....	175
15.1.8. Параметр «Состояние» .....	175
15.1.9. Параметр «Доля права» .....	176
15.1.10. Параметр «Один период» .....	176
15.1.11. Параметр «Один субъект» .....	176
15.2. Вкладка «По умолчанию» .....	176
15.2.1. Параметр «Обозначение субъекта» .....	177
15.2.2. Параметр «Субъект по умолч.» .....	177
15.2.3. Параметр «Эксклюзивный» .....	177
15.2.4. Параметр «Скрыть» .....	177
15.2.5. Параметр «Документ» .....	177
15.2.6. Параметр «Основание» .....	178
15.2.7. Параметр «Группа меню» .....	178
15.3. Вкладка «Ограничения» .....	178
15.4. Вкладка «Применимость» .....	180
15.5. Вкладка «Применимость в структуре» .....	181

Несмотря на унифицированный подход, окна операций движения разных видов могут существенно отличаться. Так, например, окно операции движения передачи на баланс имеет намного меньше возможностей (реквизитов), чем тоже окно при передаче объекта в аренду. Эти различия диктуются законодательством, а в базе данных SAUMI эти особенности каждого вида движения настраиваются с помощью специальной программы, входящей в комплект администратора SAUMI, – утилиты «Политика операций».

В «Политике операций» описываются особенности каждого вида движения и допустимости оформления операции движения к данному объекту, при условии уже оформленных на него других операций движения.

## 15.1. Вкладка «Параметры»

Настройка параметров на вкладке «Параметры» непосредственным образом влияет на внешний вид окна операции движения. Настройка большинства параметров производится установкой или снятием флажков.

ID	Название	Непрерыв	Реестр	Площади	Обязатели	Авторасче	Ценные б\п	Назначени	Состояние	Доля прав	Один пери	Один суб
1	Собственность		✓	Исключение	✓		✓	✓	✓	✓		✓
2	Балансодержание			Исключение				✓	✓	✓		✓
3	Пользование			Исключение	✓			✓	✓	✓		✓
4	Аренда			Пересечение	✓			✓	✓	✓		✓
5	Субаренда			Пересечение	✓			✓	✓	✓		✓
6	Строительство			Исключение	✓		✓	✓	✓	✓		✓
7	Приватизация			Исключение	✓			✓	✓	✓		✓
8	Обслуживание			Исключение	✓			✓	✓	✓		✓
9	Учет			Исключение				✓	✓	✓		✓
10	Казна			Исключение				✓	✓	✓	✓	✓
11	Арендный фонд			Пересечение				✓	✓	✓	✓	✓
12	Залог			Исключение	✓			✓	✓	✓		✓
13	Залоговый фонд			Исключение				✓	✓	✓	✓	✓
14	Охрана			Пересечение	✓			✓	✓	✓		✓
15	Страхование			Пересечение	✓			✓	✓	✓		✓
16	Доверительное управление			Пересечение				✓	✓	✓		✓
17	Жилый фонд			Исключение				✓	✓	✓		✓
18	Списание			Исключение				✓	✓	✓	✓	✓
19	Бессрочное пользование			Исключение	✓			✓	✓	✓		✓
20	Сервитут			Исключение	✓			✓	✓	✓		✓
21	Безвозмездное пользование			Исключение				✓	✓	✓		✓
22	Наследуемое владение			Исключение	✓			✓	✓	✓		✓

Рисунок 15.1. Политика операций, вкладка «Параметры»

Каждый столбец соответствует определенным возможностям.

### 15.1.1. Параметр «Непрерывный»

Если установлен этот параметр, то при оформлении операции движения на какой-либо объект система будет контролировать непрерывность сроков действия этих операций. Например, такой контроль уместен для вида движения «Собственность», так как у имущества всегда есть собственник, и, соответственно, при окончании срока действия собственности одного лица незамедлительно вступает в силу право собственности на этот объект у другого лица.

### 15.1.2. Параметр «Реестр»

Данный параметр устанавливается для тех видов движения, оформление операции по которым влияет на формирование реестрового номера. Если параметр установлен, то при оформлении операции будет вызвана функция SAUMI-Pascal OnGenerateRegNo, которая описывается администратором.

#### Замечание

Документация по SAUMI-Pascal, представленная в виде html-страниц на сайте <http://support.saumi.ru>, а также на дистрибутивном диске, содержит пример реализации обработчика OnGenerateRegNo.

### 15.1.3. Параметр «Площади»

Этот параметр определяет политику работы с площадями при оформлении операции движения того же вида на часть объекта. Имеет значение только для оформления операций с объектами типа «Здания и помещения».

Всего определено две политики: «Исключение» и «Пересечение».

Этот параметр имеет действие только в одном случае – при оформлении операции движения на часть тех площадей, на которые в это время уже оформлена другая операция того же вида.

Например, здание находится в муниципальной собственности и при этом необходимо передать одно помещение этого здания другому собственнику. В этом случае требуется политика площадей «Исключение», то есть передаваемая часть площадей исключается из муниципальной собственности с момента оформления собственности на помещение в этом здании.

Или другой случай: на этаже, сданном в аренду одному арендатору, другому арендатору передать часть помещений нельзя. Для реализации этого случая устанавливается политика «Пересечение», а эта политика предполагает, что площади пересекаются и как следствие не допускается оформление второй операции движения этого вида.

Вне зависимости от того, какую Вы выберете политику, система будет правильно рассчитывать используемые площади. Политика «Пересечение» используется, как правило, для того, чтобы не допустить случайной повторной сдачи объектов в аренду, так как это грозит судебными разбирательствами.

#### **15.1.4. Параметр «Обязательства»**

Для тех видов движения, для которых необходим учет финансовых обязательств, необходимо установить этот параметр. При его установке в окне операции движения в каждом периоде появится соответствующая вкладка, в которой можно назначить финансовые обязательства, вносить дополнительные начисления и платежи.

#### **15.1.5. Параметр «Авторасчет»**

Если четко определены правила расчета финансовых обязательств, то их можно описать в функции SAUMI-Pascal CalcObligationEx. Но для того, чтобы эта функция вызывалась из вкладки «Обязательства» окна операции движения, необходимо установить этот параметр.

#### **Замечание**

Документация по SAUMI-Pascal, представленная в виде html-страниц на сайте <http://support.saumi.ru>, а также на дистрибутивном диске, содержит пример реализации обработчика CalcObligationEx.

#### **15.1.6. Параметр «Ценные бумаги»**

Некоторые виды движения допускают движение ценных бумаг, соответствующее оформляемой операции движения. Например, муниципалитет может передать акционерному обществу некоторое имущество в качестве доли в уставном капитале. Это можно определить как продажа акционерному обществу имущества и покупка на эти деньги у этого акционерного общества некоторого пакета акций.

Для того чтобы в окне операции движения появилась возможность оформления движения ценных бумаг собственно и требуется установить параметр «Ценные бумаги».

#### **15.1.7. Параметр «Назначения»**

В случае, когда при оформлении операций движения на объекты требуется учитывать информацию об использовании отдельных площадей этого объекта, необходимо включить этот параметр. Это приводит к появлению соответствующей возможности.

#### **15.1.8. Параметр «Состояние»**

При оформлении операций движения некоторых видов недостаточно информации о самом объекте, также требуются данные о денежном эквиваленте имущества, на которое оформляется

движения. Так, например, при передаче в собственность всегда наряду с перечнем имущества присутствует информация о том, в какую сумму оценивается это имущество.

### **15.1.9. Параметр «Доля права»**

В соответствии с Гражданским кодексом на один и тот же объект могут иметь права несколько лиц, при этом доля права каждого лица явно определяется. Например, право собственности на квартиру может быть трех человек, у одного доля права  $\frac{1}{2}$ , у двух остальных  $\frac{1}{4}$ . Доля права может быть определена для таких прав как собственность и аренда.

Для того чтобы у конкретного вида движения была возможность определять долю права при оформлении на конкретного адресата необходимо выставить флажок «Доля права».

### **15.1.10. Параметр «Один период»**

Если для данного вида движения возможно существование только одного периода, то тогда необходимо выставить флажок «Один период». Это применимо, например, для передачи в арендный и залоговый фонды.

### **15.1.11. Параметр «Один субъект»**

Помимо правоотношений с долевым обязательством законодательство предусматривает правоотношения с солидарным обязательством для нескольких лиц (см. «Правоотношения с солидарным обязательством» в главе «Операции движения» руководства пользователя). Соответствующая такому правоотношению операция движения может учитывать не только адресата ОД, который всегда один, но и дополнительных субъектов права, участвующих в правоотношении солидарно с адресатом ОД.

Чтобы операция движения поддерживала указанную множественность лиц на стороне субъекта, необходимо сбросить флажок «Один субъект». Кроме того, на вкладке «По умолчанию» должен быть сброшен флажок «Эксклюзивный» (см. 15.2.3).

## **15.2. Вкладка «По умолчанию»**

В этой вкладке можно описать, какие поля будут отображаться на форме «Операция движения» при выборе определенного вида операции.



ID	Название	Обозначение субъекта	Субъект по умолч.	Эксклюзивный	Скрыть	Документ	Основание	Группа меню
1	Собственность	Собственник	0					
2	Балансодержание	Балансодержатель						
3	Пользование	Пользователь					-100	
4	Аренда	Арендатор					-100	
5	Субаренда	Субарендатор					-100	
6	Строительство	Строительная организация					-100	
7	Приватизация		0	✓	✓		514	
8	Обслуживание						-100	
9	Учет						-100	
10	Казна		0	✓	✓		-100	
11	Арендный фонд		0	✓	✓	10647	-100	
12	Залог	Залогодержатель					545	
13	Залоговый фонд		0	✓	✓		-100	
14	Охрана	Охранная организация					-100	
15	Страхование	Страховщик					-100	
16	Доверительное управление	Доверительный управляющий					-100	
17	Жилой фонд		0	✓	✓		461	
18	Списание		10340	✓	✓			
19	Бессрочное пользование	Пользователь					-100	
20	Сервитут						522	
21	Безвозмездное пользование	Пользователь					-100	
22	Наследственное владение	Землевладелец					-100	

Рисунок 15.2. Политика операций, вкладка «По умолчанию»

### 15.2.1. Параметр «Обозначение субъекта»

С помощью этого параметра можно установить, как в окне операции движения будет называться адресат операции движения. Если параметр не задан, то используется обозначение по умолчанию «Субъект».

### 15.2.2. Параметр «Субъект по умолч.»

Этот параметр устанавливает значение по умолчанию при оформлении новой операции движения данного вида. В качестве значения параметра устанавливается идентификатор субъекта права из таблицы CLIENTS – значение поля CLIENTS.ID соответствующей записи.

#### Замечание

Значение 0 (нуль) указывает на текущего базового собственника.

### 15.2.3. Параметр «Эксклюзивный»

При выставленном значении этого флажка, пользователи не могут менять установленное параметром «Субъект по умолч.» значение адресата. Имеет место для регламентных видов движения – арендный фонд, жилой фонд, залоговый фонд, т.е. там, где значение этого поля не имеет какой-либо определенной роли.

### 15.2.4. Параметр «Скрыть»

Этот флажок выставляется в дополнение к флажку «Эксклюзивный», т.е. тогда, когда адресата изменять нельзя. В результате поле ввода адресата отсутствует в окне операции движения.

### 15.2.5. Параметр «Документ»

В этот параметр записывается ID из таблицы DOCUMENTS, задающий документ, который по умолчанию будет заноситься в пакет документов операции движения определенного вида.

Например, при передаче какого-либо объекта в арендный фонд в пакете документов уже есть документ «Фиктивный документ о передаче в арендный фонд».

### 15.2.6. Параметр «Основание»

Этот параметр устанавливает значение по умолчанию значение поля «Основание» на форме операции движения. В этот параметр записывается значение ID из таблицы TRANSFERBASIS. Если основание не требуется, то необходимо записать «-100».

### 15.2.7. Параметр «Группа меню»

В этом параметре можно задать название подменю, в которое надо поместить вызов данного вида движения из меню «Действия». Если имеется большое количество видов движения, может быть удобным сгруппировать виды движения по смыслу. В этом случае у пользователей меню действия будет заменено несколькими подменю в соответствии с группами, к которым отнесен тот или иной вид движения.

## 15.3. Вкладка «Ограничения»

Так как SAUMI должна решать не только техническую задачу учета, но и обеспечивать логический контроль корректности вносимых данных, то нужно ориентироваться и в инструменте настройки этого логического контроля.

Разумеется, SAUMI, как и любая реальная информационная система, не в состоянии учесть все юридические тонкости, но есть некоторый набор формальных проверок, выполнив которые можно существенно уменьшить вероятность оформления юридически недопустимых действий.

SAUMI предоставляет возможность настроить проверку корректности оформления одновременных операций движения разного вида. К примеру, недопустимо сдавать в аренду объект, на который не зарегистрирована собственность. Или на объект, не сданный в аренду, нельзя оформить субаренду. Эти и подобные настройки оформляются на вкладке «Ограничения» программы Политика операций.

В заголовках строк и столбцов указаны названия видов движения. На пересечении строки и столбца, т.е. двух видов движения (например, аренда и пользование) установленный цвет ячейки определяет допустимость оформления данной операции движения.

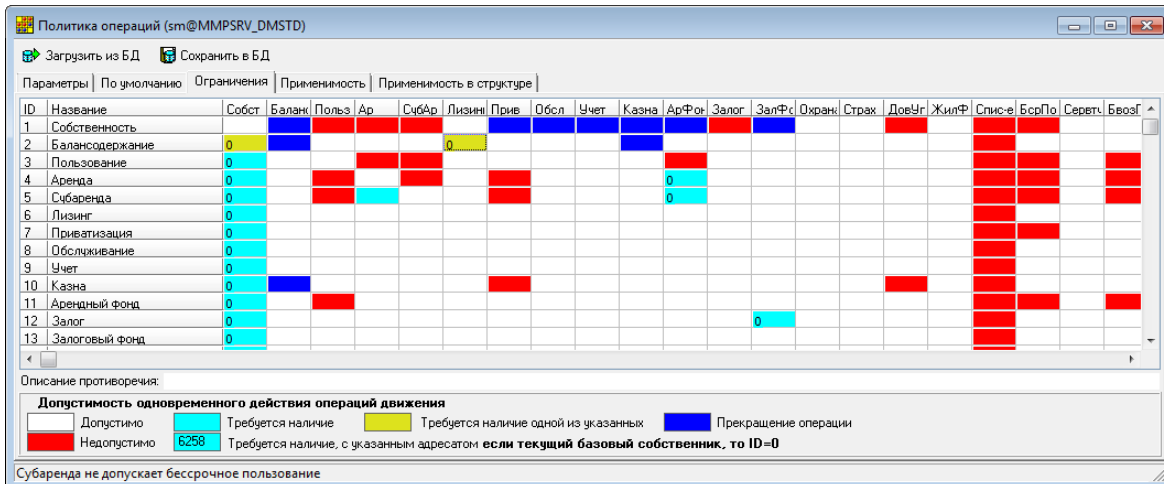


Рисунок 15.3. Политика операций, вкладка «Ограничения»

Рассмотрим на примере поведение системы при оформлении операции движения аренда. Найдем в строках среди всех видов движения – «Аренда». Двигаясь по строке, и анализируя цвет каждой из ячеек на пересечении этой строки со столбцами, можем установить допустимость совершения данной операции движения. Например, установленный красный цвет ячейки на пересечении вышеназванной строки «Аренда» и столбца «Пользование» означает запрет на возможность оформления операции движения «Аренда», если уже оформлена операция движения «Пользование». То же самое означает ячейка красного цвета на пересечении «Аренда» и «Приватизация» – объект не может быть сдан в аренду в тот же период, в течение которого он находится в процессе приватизации.

Однако помимо запрета на совершение тех или иных операций движения «Политика операций» позволяет установить требование обязательности совершения определенной операции движения. Например, в случае аренды, бирюзовый цвет ячейки, находящейся на пересечении строки «Аренда» и столбца «Арендный фонд», говорит о том, что на момент совершения операции движения «Аренда», должна быть обязательно оформлена операция движения «Арендный фонд». А указание в ячейке ID субъекта права накладывает дополнительное ограничение на совершение операции движения по отношению к конкретному субъекту права. Так, в случае аренды, ячейка бирюзового цвета со значением 0 (нуль) на пересечении строки «Аренда» и столбца «Собственность» выступает обязательным требованием принятия сначала объекта в муниципальную (государственную) собственность, так как значение ID 0 (нуль) указывает на текущего базового собственника. Только после такого принятия становится возможным совершать по отношению к объекту операцию движения «Аренда».

Ограничение «Требуется наличие одной из указанных», устанавливаемое закрасиванием соответствующих клеток в желтый цвет, интерпретируется следующим образом: создание для объекта имущества движения с таким ограничением возможно, если уже создана операция движения какого либо типа из данного ограничения. Например, движение «Балансодержание» для объекта может быть создано, если этот объект либо находится в собственности базового собственника, либо в лизинге также у базового собственника.

Кроме вышеназванных ограничений в этой вкладке есть возможность установить автоматическое прекращение одних операций движения при оформлении других. Для этого устанавливается синий цвет ячейки. Например, при выбытии имущества из муниципальной (государственной) собственности (т.е. после оформления операции движения «Собственность») автоматически будут прекращены той же датой операции движения «Обслуживание», «Учет», «Казна», «Арендный фонд», «Залоговый фонд» (рис. 15.3).

В случае нарушения заданных ограничений при оформлении операции движения пользователю выводится протокол выявленных нарушений.

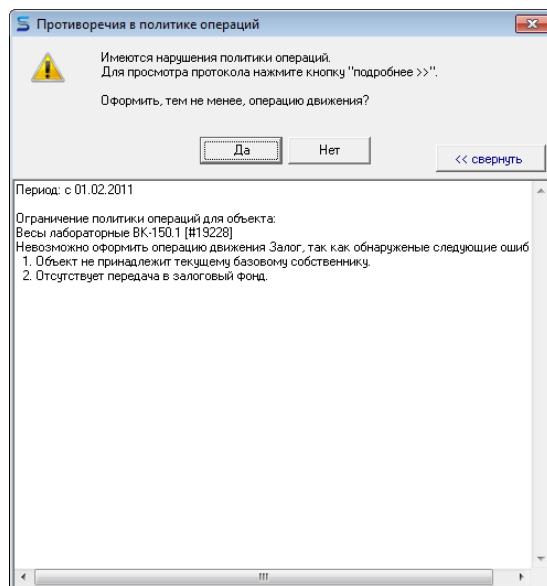


Рисунок 15.4.




## Примечание администратору

Пользователь, наделенный ролью SM\_CAN\_POLICY\_IGNORE, может оформлять операции движения с нарушением заданных ограничений. Следует проявлять осмотрительность в предоставлении данной роли.

## 15.4. Вкладка «Применимость»

На вкладке «Параметры» нет такой настройки видов движения, как применимость определенного вида движения к определенному виду имущества. А иногда необходимость в этом есть. Например, объекты незавершенного строительства ни при каких условиях не могут быть объектом жилого фонда. А «наследуемое владение» имеет отношение только к земельным участкам.

С другой стороны, политика, установленная на вкладке «Ограничения», распространяется на все виды объектов, что тоже может вызывать проблемы с оформлением движения на такой вид имущества, как земельные участки, которые могут сдаваться в аренду, при этом у них не определена форма собственности.

Для того чтобы определить, как следует поступать системе в этих двух случаях, служит вкладка «Применимость». Каждая ячейка стоит на пересечении определенного вида движения и типа имущества. Каждая ячейка может иметь три состояния, которые обозначаются графическими значками  и , а также как пустая ячейка .

ID	Наименование	Прочее движи	Транспортные	Здания и поме	Незавершенс	Инженерная и	Организации	Земельные чл
1	Собственность							
2	Балансодержание						X	
3	Пользование							
4	Аренда							✓
5	Субаренда							✓
6	Строительство	X	X				X	✓
7	Приватизация							✓
8	Обслуживание							
9	Учет							
10	Казна						X	
11	Арендный фонд							
12	Залог							
13	Залоговый фонд							
14	Охрана							
15	Страхование							
16	Доверительное управление							
17	Жилой фонд	X	X		X	X	X	X
18	Списание						X	X
19	Бессрочное пользование	X	X					✓
20	Сервитт							✓
21	Безвозмездное пользование							✓
22	Наследственное владение							✓
23	Заявка							✓
24	Банкротство	X	X	X	X	X		X
25	Ликвидация	X	X	X	X	X		X
26	Арест							

X - операция не применима    ✓ - не проверять политику

Рисунок 15.5. Политика операций, вкладка «Применимость»

Значок означает неприменимость вида движения к виду имущества. Например, операция движения «Жилой фонд» не применима к таким типам объектов, как «Транспортные средства» и «Прочее имущество».

Значок указывает системе не проверять политику, определенную на вкладке «Ограничения» при оформлении вида движения к имуществу выбранного вида. В приведенном на рисунке примере отключена проверка политики операций для земельных участков. Это связано с тем, что на вкладке ограничения установлена в т.ч. необходимость сначала оформлять принятие объекта в муниципальную собственность и лишь потом система дает возможность оформлять все прочие операции движения.

Поскольку в настоящее время разграничение земель еще до конца не прошло, то зачастую необходимо оформить операции движения на земельные участки, которые еще не находятся в муниципальной собственности, т.е. надо отключить проверку на факт оформления операция движения «Собственность» по отношению к таким типам объектов, как «Земельные участки». Как раз это устанавливается значком на вкладке «Применимость».

Пустая ячейка не устанавливает никаких дополнительных правил и ограничений сверх тех, которые установлены на вкладках «Параметры» и «Ограничения».

Изменение значений ячейки производится последовательным осуществлением двойного щелчка в поле ячейки, пока в ней не отобразится нужный знак или же ячейка станет пустой.

## 15.5. Вкладка «Применимость в структуре»

Для объектов, допускающих представление в виде иерархической структуры (типы объектов «Здания и помещения» и «Земельные участки»), на разных уровнях иерархии может требоваться различная применимость операций движения к верхнему уровню иерархий и более нижним. В частности, оформление продажи земельного участка в виде операции движения «Собственность» безусловно допустима, однако, эта же операция не может применяться к части земельного участка, как к объекту земельного права.

Установление различий в применимости операций движения производится на вкладке «Применимость в структуре».

## Замечание

Вкладка «Применимость в структуре» уточняет правила вкладки «Применимость» по отношению к земельным участкам и их частям, а также зданиям и помещениям. Если на вкладке «Применимость» для операции движения по отношению к объектам этих типов указано, что операция не применима или, что проверка применимости не производится, то уточнения на вкладке «Применимость в структуре» не имеют никакого значения.

ID	Операция	Здания и помещения		Земельные участки	
		целое	часть	целое	часть
1	Собственность				X
2	Балансодержание				
3	Пользование				
4	Аренда				
5	Субаренда				
6	Строительство				
7	Приватизация				X
8	Обслуживание				
9	Учет				
10	Казна				
11	Арендный фонд				
12	Залог				
13	Залоговый фонд				
14	Охота				
15	Страхование				
16	Довольствие от владения				
17	Жилый фонд				
18	Списание				
19	Бессрочное пользование				
20	Сервитут				
21	Безвозмездное пользование				
22	Наследственное владение				X
23	Заявка				
24	Банкротство				
25	Ликвидация				
26	Авест				

X - операция не применима

Рисунок 15.6.

Изменение применимости конкретной операции движения к целому или части как зданий и помещений, так и земельных участков осуществляется двойным щелчком в соответствующей ячейке таблицы вкладки «Применимость в структуре». Пустая ячейка указывает на допустимость данной ОД к части или целому объекту рассматриваемого вида имущества, тогда как значок X соответствует неприменимости ОД в данном случае.

---

# Глава 16. Администрирование работы с документами

## Содержание

16.1. Механизм поддержки автоматической нумерации документов .....	183
16.2. Настройки сохранения файлов документов .....	184
16.3. WebDAV-сервер хранилища документов SAUMI .....	185
16.3.1. Общие положения .....	185
16.3.2. Тип блокировки .....	185
16.3.3. Настройка типа блокировки .....	186
16.3.4. Включение/отключение работы с WebDAV .....	186
16.3.5. Ассоциированные приложения .....	186
16.3.6. Сетевые настройки .....	187
16.3.7. Протоколирование WebDAV .....	188
16.3.8. Проблема WebDAV-клиента Excel 2003 с кириллическими именами .....	188
16.3.9. Генерация имен файлов документов WebDAV-сервером SAUMI .....	189
16.3.10. Отключение протокола WebDAV для некоторых типов файлов .....	190
16.3.11. Активация новых настроек WebDAV .....	191

Стандартная автоматическая нумерация может быть подстроена администратором более точно под нужды Комитета. Ниже приводятся сведения о деталях механизма такой нумерации.

Ряд настроек работы с документами обусловлен теми возможностями, которые предоставляет приобретенная лицензия «Расширенные возможности работы с документами». Данная лицензия позволяет хранить в БД SAUMI не только регистрационные реквизиты документа, но и его содержимое в электронном представлении в виде файла определенного формата.

Сохраненные в БД SAUMI файлы документов могут выгружаться на пользовательскую рабочую станцию для их открытия в соответствующих типам файлов приложениях, с помощью которых могут производиться изменения с последующим сохранением измененных файлов документов обратно в базу данных SAUMI Раздел 16.3, «WebDAV-сервер хранилища документов SAUMI», причем имеется возможность коллективной работы с содержимым одного документа.

Указанные выше возможности требуют применения администратором комплекса определенных настроек.

## 16.1. Механизм поддержки автоматической нумерации документов

Весь механизм основан на трех PL/SQL методах:

- QUERYDOCNUM – функция запроса номера документа из банка свободных номеров;
- RETURNDOCNUM – процедура возврата неиспользованного номера в банк свободных номеров;
- DOCUMENT\_POST – триггер на сохранение документа,

дополненных двумя функциями, оперирующими заданными префиксом и/или суффиксом:

- UNWRAP\_DOCNO – функция «очищает» номер документа от префикса и суффикса для данного типа документов для передачи "чистого" номера в процедуру RETURNDOCNUM;
- WRAP\_DOCNO – функция возвращает номер документа, «обрамляя» его префиксом и суффиксом, заданным для данного типа документа,

а также функцией GetRootDocTypeID, которая для заданного типа документа находит корневой тип документа с выставленным флажком «Автоматическая нумерация» (значение 'Y' поля RESERVING в таблице DOCTYPES), если у заданного типа документа этот флажок не выставлен.

Функция QUERYDOCNUM вызывается исполняемым модулем SAUMI в момент, когда пользователь при создании документа выбрал вид документа, для которого установлен признак автоматической нумерации.

Процедура RETURNDOCNUM вызывается для возврата номера документа в банк свободных номеров.

В стандартной реализации методов QUERYDOCNUM, RETURNDOCNUM заложен алгоритм, при котором используются две таблицы:

- DOCNUMS - таблица, в которой хранится номер очередного документа в последовательности;
- RESERVED\_DOC\_NUM - таблица, в которой хранятся зарезервированные под создаваемые документы номера.

Термин «банк свободных номеров» достаточно условен, поскольку этот банк формируется двумя вышеприведенными таблицами.

Функция QUERYDOCNUM работает так, что в случае запроса номера, она сначала делает выборку в таблице RESERVED\_DOC\_NUM на поиск "просроченных" зарезервированных номеров. Если таковых нет, то выдается очередной номер документа из таблицы DOCNUMS, и выданный номер документа записывается в таблицу RESERVED\_DOC\_NUM со сроком резервирования в 24 часа (резервируется).

При сохранении документа срабатывает триггер DOCUMENT\_POST, который удаляет номер сохраняемого документа из числа зарезервированных, и это приводит к тому, что этот номер больше никогда не будет присвоен другому документу.

В случае если происходит возврат номера документа в банк свободных номеров, функция RETURNDOCNUM помечает возвращаемый номер документа в таблице RESERVED\_DOC\_NUM как просроченный, что приводит к тому, что он будет присвоен другому документу функцией QUERYDOCNUM.

## 16.2. Настройки сохранения файлов документов

В SAUMI применяется ограничение на типы сохраняемых в базе данных файлов, представляющих содержимое документов. Список допустимых типов файлов определяется параметром *STORED\_DOC\_EXTENSIONS*, задаваемым с помощью утилиты «Диспетчер настроек» из комплекта администратора SAUMI (см. Главу 4). В поле «Значение» этого параметра указываются разделенные символом точка с запятой ';' расширения имен файлов, соответствующих допустимым для сохранения типам файлов. По умолчанию применяется следующий список допустимых типов файлов.

```
doc;rtf;frp;xls;pdf;txt;html;htm;bmp;jpg;jpeg;gif;tif;tiff;
```

Добавление нового типа сохраняемых в БД SAUMI файлов производится добавлением к имеющемуся списку расширений имен файлов новых типов.



## Замечание

Одному типу файлов могут соответствовать несколько расширений, как это видно из приведенного списка по умолчанию. В частности, графическому формату JPEG соответствуют расширения `jpeg` и `jpg`.

Еще одним ограничением является размер сохраняемого файла. Параметр `STORED_DOC_MAX_SIZE` задает пороговое значение размера файлов в мегабайтах: при размере файла, превышающем пороговое значение, сохранение файла не может быть произведено. Изменение данного порога с помощью утилиты «Диспетчер настроек» следует производить с осторожностью, чтобы избежать переполнения доступного дискового пространства, используемого СУБД Oracle (в частности, Oracle XE 11g2 имеет ограничение по объему базы данных в 11 Гб).

## 16.3. WebDAV-сервер хранилища документов SAUMI

### 16.3.1. Общие положения

Сохраняемые в БД SAUMI документы могут открываться в соответствующих приложениях, где они способны подвергаться изменениям с последующим сохранением снова в БД SAUMI. В случае одновременного открытия одного документа двумя пользователями и внесения изменений в документ каждым из них, если не принять надлежащие меры, то работа одного из пользователей – того, что сохранит документ первым – будет утеряна, причем сам «пострадавший» пользователь об этом даже не будет подозревать. Такая ситуация называется конфликтом версий документа или коллизией.

Избегать указанных потерь произведенных изменений в документах форматов Microsoft Office и OpenOffice.org позволяет реализованный в SAUMI протокол WebDAV, управляющий одновременным доступом к сохраненным в БД документам посредством блокировок и отслеживанием номеров версий документов. Впервые сохраненному документу присваивается номер версии 1 (единица), а всякое повторное его сохранение в БД независимо от того, изменился документ или нет, увеличивает номер версии на единицу. Открывая документ в офисной программе, пользователь также получает актуальный на момент открытия номер версии. Эффективным способом предотвращения потерь изменений является запрет переписывать документ, чей номер версии больше, чем у предлагаемого к сохранению.

### 16.3.2. Тип блокировки

Всего поддерживается четыре варианта работы с блокировками, перечисленные ниже с указанием обозначющего номера.

#### Оптимистичная (0)

Первый открывающий документ пользователь блокирует другим пользователям возможность производить из офисных программ сохранение в БД этого документа, хотя возможность вносить изменения в офисных программах сохраняется (в отличие от варианта 1). При этом произведенные изменения пользователь может сохранить только локально на своем рабочем компьютере.

#### Пессимистичная (1)

Первый открывающий документ пользователь блокирует доступ к нему другим пользователям, которые могут открыть этот документ только для чтения.

#### Отсутствует (2)

Открывая документ одновременно, пользователи вносят изменения, и тот из них, кто первым осуществит сохранение, увеличит номер версии документа, и после этого другие

пользователи сохранить свои изменения уже не смогут, так как их номера версий будут меньше текущей.

### Отсутствует контроль версии (3)

Никакие блокировки не происходят, никакие версии не учитываются, актуальным будет последний сохраненный файл. Данный тип блокировки внешне выглядит как отсутствие какого-либо механизма разрешения коллизий, однако, он позволяет повторно сохранять в SAUMI документы форматов OpenOffice.org, а также Microsoft Excel, тогда как при отключении поддержки протокола WebDAV (см. 16.3.4) сохранять повторно в SAUMI можно только файлы Microsoft Word.

## 16.3.3. Настройка типа блокировки

Для установки типа блокировки необходимо с помощью утилиты «Диспетчер настроек» из комплекта администратора SAUMI (см. Главу 4) изменить значение параметра *DSW\_WEBDAV\_LOCK\_STRATEGY*. Возможные варианты 0, 1, 2, 3, описанные выше.

## 16.3.4. Включение/отключение работы с WebDAV

По умолчанию работа с документами через WebDAV включена.

### 16.3.4.1. Глобальное отключение

Для того чтобы отключить работу с документами через WebDAV для всех пользователей SAUMI на всех машинах, необходимо посредством утилиты «Диспетчер настроек» из комплекта администратора SAUMI (см. Главу 4) задать параметру *DSW\_WEBDAV\_USE* значение 'N' (без апострофов), которое отключает WebDAV.

### 16.3.4.2. Локальное отключение

В файле локальных настроек *saumi.ini* (см. Главу 5) можно настроить локальное отключение WebDAV для конкретной машины. Для этого нужно в секцию *[DSW\_WebDAV]* файла настроек *saumi.ini* добавить параметр *Use=0*. Если секции нет, то ее нужно создать, если параметр *Use* есть и имеет ненулевое значение, то его нужно изменить на ноль.

## Замечание

Файл настроек *saumi.ini* располагается в папке установки SAUMI (по умолчанию – это *c:\SAUMI*).

## 16.3.5. Ассоциированные приложения

### 16.3.5.1. Действие по умолчанию

WebDAV по умолчанию обрабатывает только файлы с расширениями *.rtf*, *.doc*, *.xls*, *.docx*, *.xlsx* (Microsoft Office), а также *.odt*, *.ods* (OpenOffice.org). Для каждого типа файлов ищется ассоциированная программа, если программа не находится, то для обработки документов этого типа пользователь должен будет сам выбрать программу-обработчик, а сервис WebDAV в таком случае будет недоступен, как и возможность сохранить документ в БД непосредственно из выбранного пользователем приложения.

Ассоциация типа документа с программой-обработчиком определяется с возрастанием приоритета:

- реестром Windows;

- общими настройками ассоциаций;
- локальными настройками ассоциаций.

Т.е., если, например, с типом файла `.rtf` в реестре Windows ассоциирована программа `WinWord.exe`, а локальные настройки этот тип файла ассоциируют с `Wordpad.exe`, то будет вызвана именно последняя программа.

### 16.3.5.2. Общие настройки

Ассоциированные с типами документов приложения указываются в таблице `DSW_EXT`. Новая ассоциация добавляется выполнением соответствующего PL/SQL-скрипта, задающего значения полей `dsw_type`, `ext`, `app`, `params` этой таблицы. В поле `dsw_type` следует задавать значение 'WEBDAV', а в поле `ext` расширение, соответствующее типу файла. Поле `app` должно содержать путь к приложению, открывающему файлы данного типа, а поле `params` параметры командной строки запуска данного приложения. Фрагмент строки '{0}' в поле `params` указывает подстановку имени открываемого файла в команду запуска приложения.

```
insert into sm.dsw_ext (dsw_type, ext, app, params)
values ('WEBDAV', 'txt',
'C:\Program Files\Microsoft Office\OFFICE12\WINWORD.EXE', '{0}')
```

Для выполнения приведенной в примере SQL-команды можно использовать утилиту «Менеджер реквизитов» из комплекта администратора SAUMI (см. Главу 8).

### 16.3.5.3. Локальные настройки

В файле настроек `saumi.ini` (см. Главу 5) можно задать локальную ассоциацию приложений. Для этого в секцию `[DSW_WebDAV]` нужно добавить значения `Ext_N`, `App_N`, `Params_N`.

```
Ext_1=txt
App_1=C:\Program Files\Microsoft Office\OFFICE11\WINWORD.EXE
Params_1={0}
```

Локальные настройки ассоциаций перекрывают действие общих настроек.

## 16.3.6. Сетевые настройки

В SAUMI-реализации протокола WebDAV взаимодействие по схеме клиент-сервер происходит исключительно в пределах рабочей станции, где в качестве сервера выступает главный исполняемый модуль SAUMI, а клиентскими приложениями являются офисные приложения. Возможность ведения версий и блокировок, описанных выше, обусловлена общей базой данных, которая позволяет каждому такому локальному серверу иметь информацию об открытии файла из БД SAUMI и его версиях.

Локальное, т.е. в рамках узла `localhost` со стандартным IP-адресом `127.0.0.1`, взаимодействие между WebDAV сервером и клиентами осуществляется с использованием сетевого протокола TCP/IP в диапазоне портов, установленном параметрами `MinPort`, `MaxPort` секции `[DSW_WebDAV]` файла `saumi.ini`. По умолчанию применяется диапазон портов от 36422 до 36457.

```
[DSW_WebDAV]
MinPort=36422
MaxPort=36457
```

Следует использовать достаточно большие номера портов, не используемые другими приложениями. Перечень занятых в системе TCP/IP портов можно узнать из вывода консольной

команды Windows **netstat -a** (для страничного вывода можно воспользоваться конвейером команд **netstat -a | more**).

## Замечание

Обеспечивающий на рабочей станции сетевую безопасность сетевой экран – брандмауэр – может стать причиной невозможности WebDAV-взаимодействия, если установленный диапазон портов *MinPort* – *MaxPort* будет закрыт брандмауэром. Приведенное замечание касается не столько брандмауэра из поставки Windows, не препятствующего локальным взаимодействиям, сколько сетевых экранов других производителей, способных обеспечивать более жесткую политику защиты. Такие брандмауэры следует настроить надлежащим образом, для обеспечения взаимодействия главного модуля SAUMI с офисными приложениями в установленном диапазоне портов TCP/IP.

### 16.3.7. Протоколирование WebDAV

Параметр *WriteLog* секции *[DSW\_WebDAV]* файла настроек *saumi.ini* при нулевом значении (по умолчанию) отключает протоколирование работы протокола WebDAV в SAUMI на локальной машине. Значение *WriteLog=1* вызывает указанное протоколирование в файле с именем, задаваемым параметром *LogFile*. По умолчанию файл протоколирования *log\_dsw.txt*, располагаемый в директории главного исполняемого модуля SAUMI.

```
[DSW_WebDAV]
WriteLog=1
LogFileName=c:\log_dsw.txt
```

### 16.3.8. Проблема WebDAV-клиента Excel 2003 с кириллическими именами

Сотрудники ряда организаций при использовании WebDAV-протокола для доступа к документам, сохраненным в БД SAUMI, столкнулись со следующим некорректным поведением табличного процессора MS Excel 2003. При сохранении истории использованных документов (эта возможность применяется в Excel по умолчанию), срабатывающей при закрытии окна документа, приложение производит аварийное завершение работы, если документ был открыт из БД SAUMI.

Причиной здесь является некорректная обработка программой Excel 2003 URL-адреса файла, генерируемого WebDAV-сервером хранилища документов SAUMI, который для более точной идентификации вставляет в имя файла кириллическое название вида документа и его номер, а затем производит его URL кодирование, при котором все кириллические символы заменяются кодовыми последовательностями вида %hh – здесь hh обозначает двухсимвольный шестнадцатеричный код кириллической буквы.

Простым выходом из описанной ситуации является снятие флажка «Помнить список файлов» в настройках Excel 2003, в результате чего история последних использованных документов перестает сохраняться. Однако такой способ может быть неудобным для ряда пользователей, а также достаточно проблематичным в большой организации.

Решает проблему отказ от использования в генерируемых именах файлов кириллических символов, однако, это снижает идентификационные свойства имен. Другое совсем радикальное решение – отключить WebDAV протокол, и тогда файл будет представлен именем локального файла, а не кодированным URL-адресом интернет-ресурса. Но в этом случае пользователи лишаются механизма блокировок совместного доступа к файлам, ради которого, собственно, WebDAV протокол и был реализован в SAUMI. Более мягким решением представляется отключение WebDAV протокола для конкретных типов файлов, которые либо редко применяются, либо не подвергаются изменениям действиями пользователей, т.е. используются для просмотра.

Администраторам SAUMI предлагается посредством гибких настроек реализации протокола WebDAV использовать оба подхода – альтернативную схему именования файлов и отключение WebDAV для определенных типов файлов.

Реализованная альтернативная схема генерации имен файлов создает имя файла с использованием префикса из ASCII-символов и числового суффикса. Параметры настроек позволяют установить альтернативную схему генерации имен файлов, как глобально на сервере для всех пользователей, так и локально на конкретных пользовательских компьютерах.

Дополнительно имеется возможность указать, какие приложения открывают сохраненные в БД SAUMI файлы без использования протокола WebDAV. Такое указание производится с использованием ассоциаций типов файлов и приложений их обрабатывающих (см. 16.3.5) и может быть сделано как глобально, так и локально.

### 16.3.9. Генерация имен файлов документов WebDAV-сервером SAUMI

Как отмечалось выше, WebDAV-сервер SAUMI реализует две схемы генерации имен выгружаемых из хранилища файлов документов: *содержательную* и *альтернативную*.

Схема генерации **содержательных** имен имеет своей целью более точно идентифицировать документ по имени его файла посредством указания в имени названия вида документа и его учетный номер. Если приложения, с помощью которых пользователи открывают сохраненные в БД SAUMI документы, не испытывают проблем с URL кодированными кириллическими именами, то следует применять именно эту схему, которая используется по умолчанию.

В случаях, аналогичных MS Excel 2003, следует применять **альтернативную** схему генерации имен выгружаемых файлов. Сгенерированное по этой схеме имя состоит из двухсимвольного префикса "ID" и цифрового суффикса, представляющего уникальный номер, генерируемый БД SAUMI.

#### Замечание

В любом случае к имени файла добавляется расширение, соответствующее типу файла.

#### 16.3.9.1. Глобальное задание схемы генерации имен файлов

Для глобального задания схемы генерации имен файлов применяется параметр `DSW_WEBDAV_FILENAME_STRATEGY`, который может быть изменен посредством утилиты «Диспетчер настроек» из комплекта администратора SAUMI (см. Главу 4).

Указанный параметр может принимать два значения 0 и 1:

- значение 0 задает **содержательную** схему имен;
- значение 1 устанавливает **альтернативную** схему.

#### 16.3.9.2. Локальное задание схемы генерации имен файлов

Для локального задания схемы генерации имен файлов, действующей только на компьютере пользователя, применяется параметр `FileNameStrategy`, располагаемый в секции `[DSW_WebDAV]` файла настроек `saumi.ini`.

Новый локальный параметр по аналогии с глобальным параметром `DSW_WEBDAV_FILENAME_STRATEGY` принимает два значения 0 и 1:

- значение 0 задает **содержательную** схему имен;
- значение 1 устанавливает **альтернативную** схему.

Соответствующая запись в файле настроек `saumi.ini` имеет следующий вид для содержательной схемы.

```
[DSW_WebDAV]
FileNameStrategy=0
```

### 16.3.10. Отключение протокола WebDAV для некоторых типов файлов

Некоторые типы файлов, сохраняемых в БД SAUMI, могут применяться только в информационных целях без модификации их пользователями, либо такая модификация может быть строго регламентирована так, что только один ответственный пользователь имеет полномочия заменять содержимое документа. Примером такого подхода могут быть отсканированные образы входящих документов, представление которых в виде электронных графических образов формата `.tiff` готовит и сохраняет в БД SAUMI ответственный секретарь.

В таких случаях нет надобности применять доступ к файлу документа данного типа через протокол WebDAV и с помощью настроек можно указать, что для таких файлов протокол WebDAV при выгрузке из хранилища документов SAUMI не применяется.

Отключающие настройки используют ассоциации типов файлов и приложений, как глобальные, так и локальные, и заключаются в задании вместо пути к исполняемому файлу приложения строкового значения `STOP-SUPPORT` для соответствующего параметра.

Ниже отключение протокола WebDAV для некоторого типа файлов демонстрируется на примере графических файлов формата TIFF, которому соответствуют расширения файлов `.tif` и `.tiff`.

#### 16.3.10.1. Глобальное отключение

Чтобы осуществить глобальное отключение протокола WebDAV для графических файлов формата TIFF, администратору необходимо выполнить следующие команды PL/SQL.

```
insert into sm.dsw_ext (dsw_type, ext, app, params)
  values('WEBDAV','tif','STOP-SUPPORT','{0}');

insert into sm.dsw_ext (dsw_type, ext, app, params)
  values('WEBDAV','tiff','STOP-SUPPORT','{0}');
```

Для выполнения указанных команд можно использовать утилиту «Менеджер реквизитов» из комплекта администратора SAUMI (см. Главу 8).

#### 16.3.10.2. Локальное отключение

Локальное отключение протокола WebDAV для графических файлов формата TIFF, производится добавлением следующих параметров в секции `[DSW_WebDAV]` файла настроек `saumi.ini`.

```
[DSW_WebDAV]
Ext_1=tif
App_1=STOP-SUPPORT
Params_1={0}

Ext_2=tiff
App_1=STOP-SUPPORT
Params_1={0}
```

## Замечание

Приводимые здесь номера групп параметров 1 и 2 используются только для примера и должны быть заменены в случае необходимости на номера, еще не использованные в локальных настройках ассоциаций типов файлов и приложений (см. 16.3.5).

### **16.3.11. Активация новых настроек WebDAV**

Чтобы производимые в соответствии с вышеописанным настройки WebDAV, как общие, так и локальные, вступили в силу, необходимо перезагрузить клиентскую часть SAUMI.

## Замечание

Локальные настройки WebDAV имеют приоритет над глобальными.

---

# Глава 17. Настройка шаблонов документов

## Содержание

17.1. Назначение шаблона виду документа .....	195
17.2. Удаление шаблона .....	195
17.3. Порядок работы с шаблонами MS Word .....	196
17.3.1. Назначение шаблона MS Word .....	196
17.3.2. Создание шаблона MS Word .....	198
17.3.3. Переменная CONTEXT .....	199
17.3.4. Примеры кода SAUMI-Pascal для полей слияния .....	200
17.3.5. Особенности назначения шаблонов MS Word .....	203
17.4. Порядок работы с HTML-шаблонами .....	203
17.4.1. Структура HTML-шаблона .....	204
17.4.2. Назначение HTML-шаблона .....	204
17.4.3. Создание пустого шаблона .....	205
17.4.4. Загрузка HTML-шаблона из spz-файла .....	206
17.4.5. Редактирование назначенного HTML-шаблона .....	206
17.4.6. Интерфейс создания / редактирования шаблона .....	206
17.4.7. Настройка конфигурации шаблона .....	209
17.4.8. Принципы программирования шаблона .....	211
17.4.9. Формирование динамических фрагментов текста .....	212
17.4.10. Запросы к базе данных .....	214
17.4.11. Использование списков .....	215
17.4.12. Использование стилей .....	217
17.4.13. Обработка штрих-кодов .....	218
17.4.14. Отладка HTML-шаблона при помощи Microsoft Visual Studio .....	219

Рассматриваемые в данной главе вопросы предполагают наличие приобретенной лицензии «Расширенные возможности работы с документами», которая позволяет использовать следующие способы формирования содержимого документов, регистрируемых в базе данных SAUMI.

1. Загрузка из существующего файла допустимого формата (см. раздел «Работа с текстом документа в окне свойств документа» главы «Работа с документами в SAUMI» руководства пользователя).
2. Формирование содержимого документа с помощью генератора отчетов FastReport на основе печатной формы, разработанной с помощью утилиты «Менеджер форм» из комплекта администратора SAUMI (см. 11.3).
3. Формирование содержимого документа на основе шаблона MS Word.
4. Формирование содержимого документа с использованием HTML-шаблона.

Доступ к тому или иному способу формирования содержимого документов осуществляется из контекстного меню работы с документами в соответствующих окнах SAUMI. Способу 1 соответствует пункт контекстного меню «Свойства документа», способу 2 – «Сформировать текст из формы», а способам 3 и 4 соответствует пункт «Создать на шаблоне».



## Замечание

Чтобы перечисленные выше возможности были доступны, эксплуатант SAUMI должен иметь лицензию «Расширенные возможности работы с документами». Кроме того, пользователи, использующие учетные записи, отличные от SM, должны быть наделены ролью SM\_CAN\_WORK\_WITH\_DOCBODY, а те из них, кто будет участвовать в создании HTML-шаблонов, должны быть наделены также ролью SM\_EDIT\_HTMLTMPL.

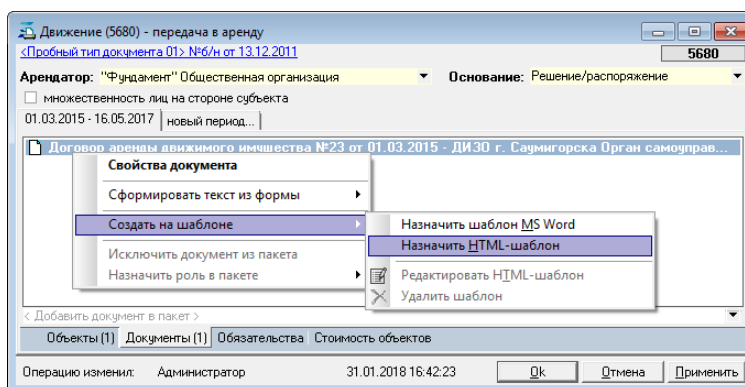


Рисунок 17.1. Контекстное меню работы с документами при запуске SAUMI под учётной записью SM

## Замечание

Пункты контекстного меню, предназначенные для администрирования шаблонов документов, отображаются только при запуске главного исполняемого модуля SAUMI (mp32.exe) под учётной записью SM (администратор БД SAUMI). В контекстном меню обычного пользователя с предоставленной ему ролью SM\_CAN\_WORK\_WITH\_DOCBODY отображается список доступных шаблонов.

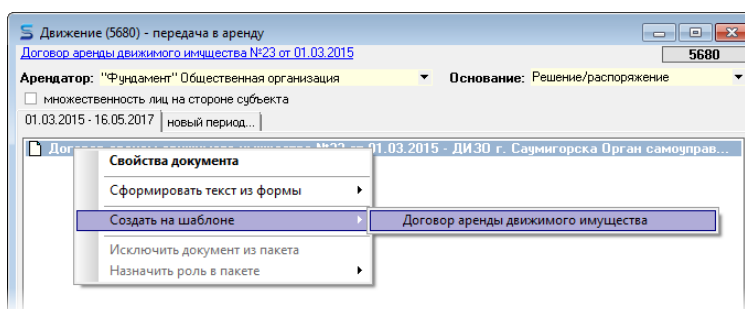


Рисунок 17.2. Контекстное меню работы с документами обычного пользователя с ролью SM\_CAN\_WORK\_WITH\_DOCBODY

## Замечание

Пункты контекстного меню, предназначенные для применения HTML-шаблонов и их администрирования, отображаются, только при наличии в папке установки SAUMI библиотек InflexENV.dll и HtmlForm.dll.

Порядок загрузки содержимого документа из существующего файла допустимого формата и формирование содержимого документа с помощью генератора отчётов FastReport на

основе печатной формы, разработанной в «Менеджере печатных форм» SAUMI описаны в соответствующих разделах Руководств SAUMI.

В данном разделе описывается порядок работы с шаблонами MS Word и HTML-шаблонами. Шаблоны содержат постоянную (статическую) составляющую текста, общую для всех генерируемых документов, а также используют определенные механизмы, обеспечивающие подстановку в нужных местах переменных фрагментов (динамическую часть) текста, как то: номер и дату документа, персоналии, объекты имущества и их описание и т.д.

## 17.1. Назначение шаблона виду документа

Для того чтобы пользователи могли формировать содержимое документов определенного вида, как с помощью шаблона MS Word, так и HTML-шаблона, такой шаблон следует назначить. Назначение шаблонов обоих типов производится администратором под учетной записью SM посредством контекстного меню работы с документами, которое доступно на вкладках "Документы" окон свойств различных сущностей SAUMI.

### Замечание

Меню работы с текстом документа, доступное в окне свойств документа, позволяет назначать только шаблоны MS Word (не HTML-шаблоны).

Предварительно пакет документов уже должен содержать документ того вида, которому назначается соответствующий шаблон. Вызывая правой кнопкой мыши контекстное меню такого документа, необходимо выбрать элемент «Создать на шаблоне», а в открывшемся подменю в зависимости от типа шаблона либо пункт «Назначить шаблон MS Word», либо пункт «Назначить HTML-шаблон» (рис. 17.1). Детализация выполняемых при назначении шаблона действий зависит от его типа и приводится ниже отдельно для шаблонов MS Word и HTML-шаблонов.

Результатом назначения становится ассоциация шаблона с видом документа и типом сущности SAUMI, из окна свойств которой производилось назначение. Т.е. назначение, например, HTML-шаблона «Договор аренды земельного участка» в окне операции движения для вида документа «Договор аренды» никоим образом не вызывает назначения этого же шаблона для документов вида «Договор аренды» в окне свойств юридического лица, если каким-либо образом такой документ попадет в пакет документов организации.

Одному виду документов могут соответствовать несколько шаблонов, причем разных типов. Например, упоминавшемуся выше виду документа «Договор аренды» в контексте операций движения помимо HTML-шаблона «Договор аренды земельного участка» могут быть сопоставлены также шаблоны MS Word «Договор аренды недвижимого имущества» и «Договор аренды транспортного средства».

Количество назначаемых шаблонов определенному виду документов в контекстном меню работы с документами окна свойств определенной сущности SAUMI ничем не ограничено.

## 17.2. Удаление шаблона

Для удаления назначенного виду документов в определенном окне свойств сущности SAUMI шаблона, как MS Word, так и HTML, применяется контекстное меню работы с документами, вызываемое щелчком мыши по строке описания документа данного вида в окне свойств данной сущности. В контекстном меню нужно выбрать пункт «Создать на шаблоне», в результате чего открывается подменю, содержащее элемент «Удалить шаблон», выбор которого выводит перечень шаблонов, назначенных данному виду документа в окне данной сущности SAUMI.

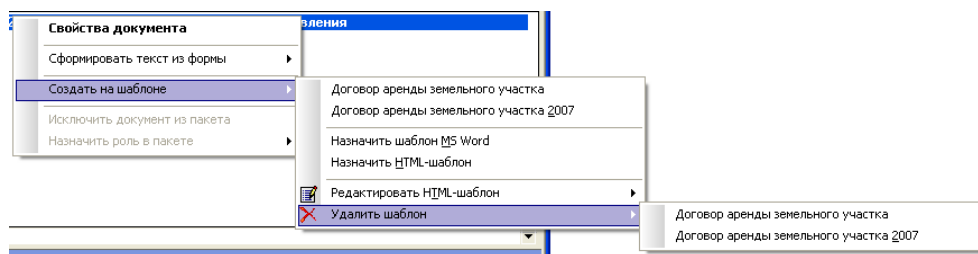


Рисунок 17.3.

Удаленный шаблон перестаёт отображаться в списке доступных шаблонов. При этом содержимое документов, которое было сформировано с использованием этого шаблона, никуда не исчезает, и по-прежнему доступно для редактирования, печати и т.п.

## 17.3. Порядок работы с шаблонами MS Word

Выбор в контекстном меню работы с документами (рис. 17.1) элемента «Создать на шаблоне» вызывает подменю, содержащее в верхней своей части перечень уже назначенных данному виду документов шаблонов MS Word, а в нижней части пункты «Назначить шаблон MS Word», «Удалить шаблон». В целом, описанное подменю позволяет администратору для выбранного документа выполнить следующие действия.

1. Создать текст документа, используя уже назначенные и потому доступные для данного вида документов шаблоны MS Word (см. подразделы «Создание текста документа на шаблоне MS Word» и «Работа с текстом документа в пакете» главы «Работа с документами в SAUMI» руководства пользователя).
2. Назначить шаблон MS Word для данного вида документа.
3. Удалить шаблон для данного вида документа.

Пункты 2 и 3 доступны только администратору SAUMI, который прежде чем назначить шаблон, должен его создать.

### 17.3.1. Назначение шаблона MS Word

Выбор элемента «Назначить шаблон MS Word» в подменю «Создать на шаблоне» контекстного меню работы с документами

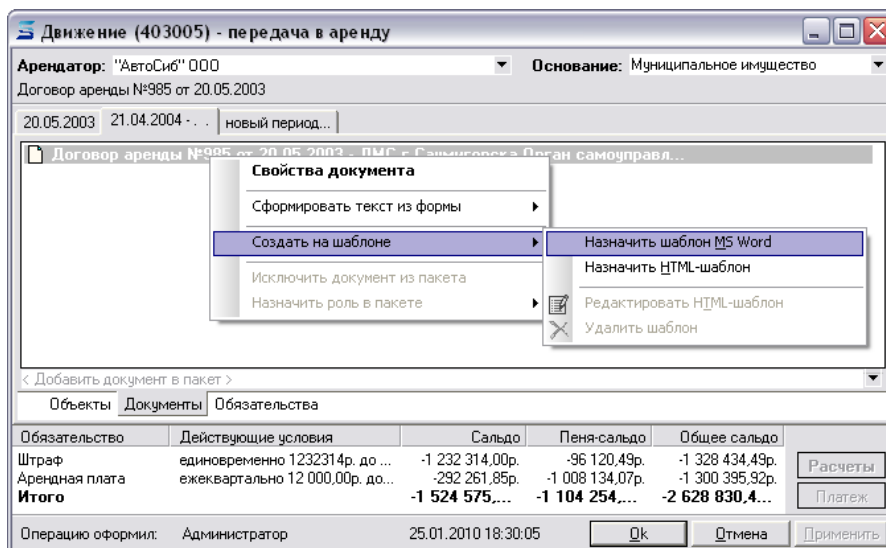


Рисунок 17.4.

открывает окно выбора файла, в котором нужно указать соответствующий файл шаблона MS Word с расширением dot.

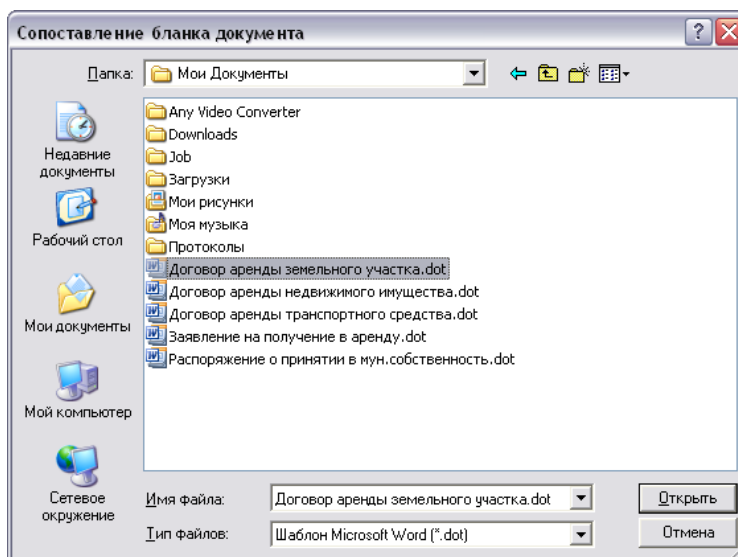


Рисунок 17.5.

В результате произведенного назначения в списке доступных шаблонов для данного вида документов появляется новый элемент.

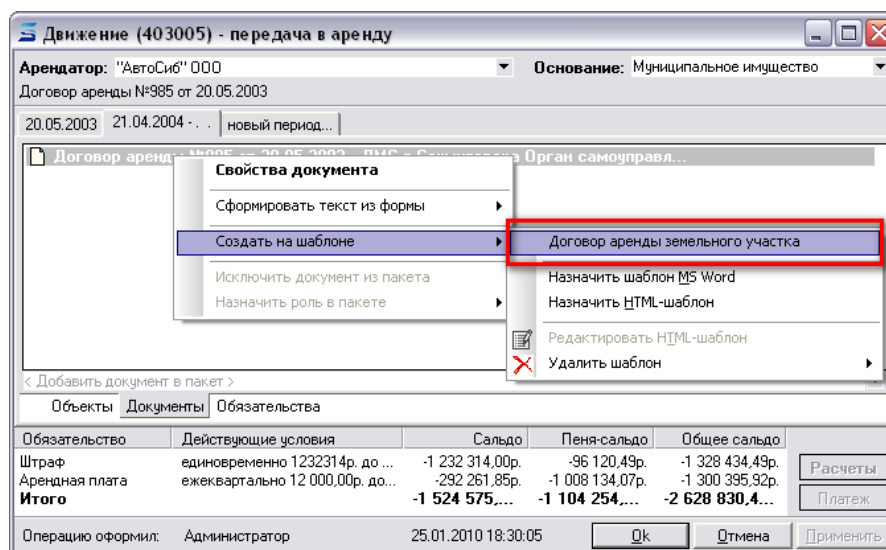


Рисунок 17.6.

## Замечание

Название шаблона MS Word в списке доступных для вида документов шаблонов совпадает с наименованием файла назначенного шаблона и не подлежит редактированию. Следует задавать информативные и однозначные названия файлов шаблонов, чтобы правильно выбрать нужный шаблон из списка, формируемого из имен файлов.

### 17.3.2. Создание шаблона MS Word

Разработка шаблона MS Word производится в следующем порядке.

1. В редакторе MS Word создаётся документ, который предполагается использовать в качестве шаблона, например, для документов типа «Договор Аренды».
2. В созданном документе набирается текст, который будет общим для всех документов «Договор Аренды».
3. В те места, куда должны быть проставлены конкретные данные конкретного документа (например, номер документа, дата документа, наименование субъекта права, сроки начала и окончания договора и т.д.) вставляются поля слияния.
4. Созданный документ сохраняется как шаблон Word, при этом следует задать содержательное имя файла, позволяющее точно идентифицировать назначение созданного шаблона.

## Замечание

Документ, созданный в MS Word 2007 и в более поздних редакциях MS Word, следует сохранять как шаблон Word 97-2003.

Поля слияния определяются следующей последовательностью шагов.

1. Выбрать в меню MS Word «Вставка» элемент «Поле»;
2. В открывшемся окне «Поле» выбрать категорию «Автоматизация»;

3. В списке полей выбрать `DOCVARIABLE`;
4. Нажать кнопку «Коды поля»;
5. Ввести справа от `DOCVARIABLE` код поля, заключенный в кавычки. Для вставки номера документа в качестве кода применяется имя "DOCNO".

`DOCVARIABLE "DOCNO"`

Для вставки даты документа – имя "DOCDATE".

`DOCVARIABLE "DOCDATE"`

В остальных случаях в качестве кода поля используется вызов SAUMI-Pascal функции с переменной `CONTEXT` в качестве аргумента в следующей нотации.

`DOCVARIABLE "SP_FUNC:MyFunc (CONTEXT) "`

Здесь `MyFunc` – название SAUMI-Pascal функции, описанной администратором SAUMI. Функция должна возвращать строковое значение, которое будет подставляться вместо поля. Тип переменной `CONTEXT` зависит от типа сущности, в окне свойств которой производится вызов шаблона. Так для окна операции движения переменная `CONTEXT` имеет тип `TMoveSet`, следовательно, в описанной администратором SAUMI-Pascal функции можно извлечь любые данные типа `TMoveSet` и передать их в документ Word для подстановки. О типе переменной `CONTEXT` в окнах других типов см. 17.3.3.

## Замечание

В коде поля слияния между `DOCVARIABLE` и выражением в кавычках должен быть в точности один пробел. После выражения в кавычках также должен быть один пробел. В противном случае вставка контекста не происходит.

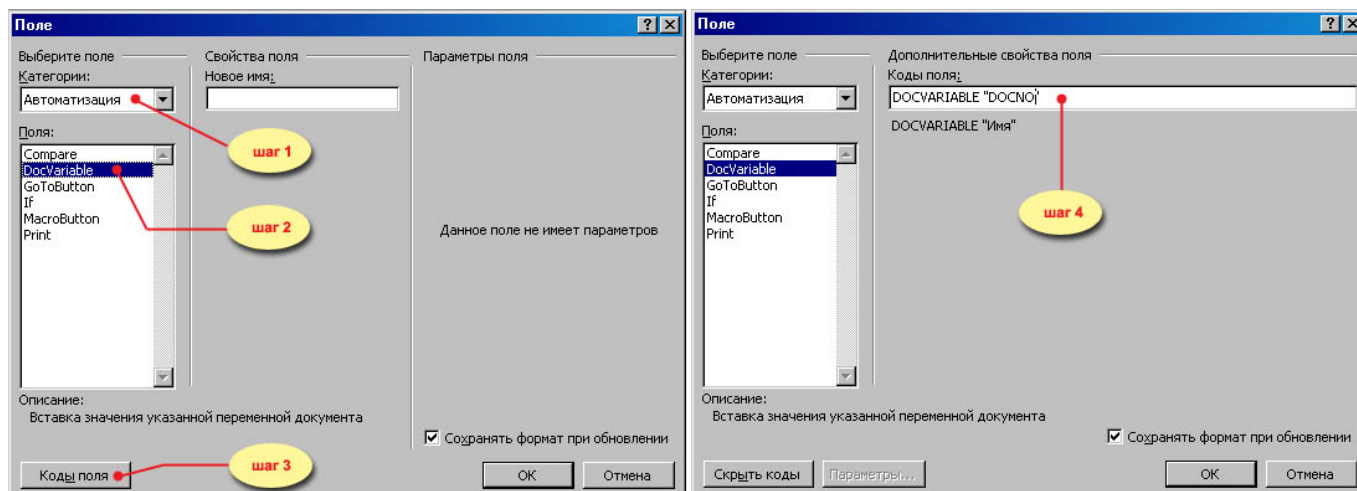


Рисунок 17.7. Вставка поля слияния

### 17.3.3. Переменная `CONTEXT`

Для того чтобы при формировании документа на основе шаблона можно было использовать информацию, которая отображается в соответствующем окне, но может еще отсутствовать в БД (данные кэшированы), введена переменная `CONTEXT`. Тип этой переменной определяется тем, из

какого окна вызывается контекстное меню работы с документами. При вызове меню из окна операции движения переменная `CONTEXT` имеет SAUMI-Pascal тип `TMoveSet`, при вызове меню из окна свойств субъекта права переменная `CONTEXT` имеет SAUMI-Pascal тип `TClient`.

Переменная `CONTEXT` определена также для ряда других окон SAUMI:

- свойств объектов;
- регистрации состояния объектов;
- платежей;
- дополнительных начислений;
- отсрочек;
- льгот;
- операций с уставным (складочным) капиталом (фондом).

В окнах приведенного выше списка переменная `CONTEXT` имеет SAUMI-Pascal тип `Integer`, а в качестве значения ей присваивается идентификатор экземпляра сущности, представленного открытым окном, через контекстное меню которого вызывается создание документа на основе шаблона.

## Замечание

Определение контекста по идентификатору экземпляра сущности осуществляется через запрос к базе данных SAUMI. Это значит, что перед формированием документа на основе шаблона MS Word должно быть произведено сохранение в БД как вновь создаваемого, так и изменяемого экземпляра сущности. Указанное сохранение производится нажатием на кнопку «Применить» или «ОК» в определяющем контекст окне. В противном случае информация будет либо совсем не найдена (для вновь созданного и несохраненного экземпляра сущности), либо будет найдена, но не будет содержать произведенные изменения.

### 17.3.4. Примеры кода SAUMI-Pascal для полей слияния

Описание языка SAUMI-Pascal представлено в виде html страниц на сайте <http://support.saumi.ru>, а также на дистрибутивном диске SAUMI.

#### Пример 17.1. Вставка наименования субъекта права текущей операции

*В некоторое поле слияния документа Word необходимо вставить наименование субъекта права текущей операции движения.*

Следующая SAUMI-Pascal функция решает эту задачу.

```
function GetClientName (MS:TMoveSet):string;  
// в качестве CONTEXT из окна свойств операции движения передаётся TMoveSet  
begin  
    Result:=MS.Client.ClientName;  
end;
```

В создаваемом шаблоне документа в то место, где должно появиться наименование субъекта права по приведенной выше схеме вставляется поле слияния со следующим кодом.

```
DOCVARIABLE "SP_FUNC:GetClientName (CONTEXT) "
```

В момент создания текста документа из окна ОД в данное поле будет произведена подстановка наименования субъекта.

### Пример 17.2. Вставка даты окончания текущей операции движения

*В некоторое поле слияния документа Word требуется вставить дату окончания текущей операции движения.*

Соответствующая SAUMI-Pascal функция.

```
function GetToDate (MS:TMoveSet):string;  
// в качестве CONTEXT из окна свойств операции движения передаётся TMoveSet  
begin  
  if MS.Moveperiods[MS.CurPeriodIndex].DateTo<MS.Moveperiods[MS.CurPeriodIndex].DateFrom then  
    Result:='' // дата окончания договора не определена  
  else  
    Result:=DateToStr (MS.Moveperiods[MS.CurPeriodIndex].DateTo);end;
```

Соответствующий код поля слияния.

```
DOCVARIABLE "SP_FUNC:GetToDate (CONTEXT) "
```

В момент создания текста документа из окна ОД в данное поле будет произведена подстановка даты окончания текущей операции движения.

### Пример 17.3. Вставка зависящей от контекста строки

*В некоторое поле слияния документа Word нужно вставить слово «да», если субъект права является юридическим лицом, и слово «нет» в противном случае. Текст документа будет формироваться из окна свойств клиента.*

Соответствующая SAUMI-Pascal функция.

```
function IsOrganisation (Cl:TClient):string;  
// в качестве CONTEXT из окна свойств субъекта передаётся TClient  
begin  
  if Cl.Organisation then  
    Result:='да'  
  else  
    Result:='нет';  
end;
```

Соответствующий код поля слияния.

```
DOCVARIABLE "SP_FUNC:IsOrganisation (CONTEXT) "
```

В момент создания текста документа из окна свойств субъекта права в данное поле будет вставлено слово «да» (или «нет»).

### Пример 17.4. Описание объекта из окна свойств объекта

*В формируемый из окна свойств объекта на основе шаблона MS Word документ требуется вставить описание объекта.*

Соответствующий код поля слияния.

```
DOCVARIABLE "SP_FUNC:GetObjDescr (CONTEXT) "
```



Функция `GetObjDescr` возвращает строку описания объекта по идентификатору объекта. Само описание объекта формирует PL/SQL-функция `GetObjectDescription`.

```
function GetObjDescr(OBJ_ID:integer):string;
var
  sql:integer;
begin
  sql:=SQL_BindOpen('select sm.GetObjectDescription(:ID) descr from dual',[OBJ_ID]);
  if not SQL_Eof(sql) then
  begin
    Result:=SQL_FieldAsString(sql,'descr');
  end;

  SQL_Close(sql);

end;
```

### Пример 17.5. Описание объекта из окна регистрации состояния объекта

*В формируемый из окна состояния объекта на основе шаблона MS Word документ требуется вставить описание объекта.*

Соответствующий код поля слияния.

```
DOCVARIABLE "SP_FUNC:GetObjDescrByStateID(CONTEXT)"
```

Функция `GetObjDescrByStateID` возвращает строку описания объекта по идентификатору записи состояния объекта.

```
function GetObjDescrByStateID(STATE_ID:integer):string;
var
  sql:integer;
begin
  sql:=SQL_BindOpen('select sm.GetObjectDescription(st.Objects_id) descr '+
                    'from sm.States st '+
                    'where st.ID=:STID',[STATE_ID]);
  if not SQL_Eof(sql) then
  begin
    Result:=SQL_FieldAsString(sql,'descr');
  end;

  SQL_Close(sql);

end;
```

### Пример 17.6. Описание состояния объекта из окна регистрации состояния объекта

*В формируемый из окна регистрации состояния объекта на основе шаблона MS Word документ требуется вставить описание состояния объекта.*

Соответствующий код поля слияния.

```
DOCVARIABLE "SP_FUNC:GetStateDescr(CONTEXT)"
```

Функция `GetStateDescr` находит описание состояния объекта в справочной таблице `STATETYPES` по идентификатору записи состояния объекта.

```
function GetStateDescr(STATE_ID:integer):string;
```

```

var
  sql:integer;
begin

  sql:=SQL_BindOpen('select stt.NAME descr '+
                    'from '+
                    ' sm.STATEYPES stt, '+
                    ' sm.STATES st '+
                    'where '+
                    ' st.STATEYPES_ID=stt.ID and '+
                    ' st.ID=:STID',[STATE_ID]);

  if not SQL_Eof(sql) then
  begin
    Result:=SQL_FieldAsString(sql, 'descr');
  end;

  SQL_Close(sql);

end;

```

### 17.3.5. Особенности назначения шаблонов MS Word

При назначении в контекстное меню работы с документами определенного окна шаблонов MS Word, использующих обработку переменной CONTEXT, администратору следует проявлять аккуратность и осмотрительность.

При использовании SAUMI-Pascal обработчиком несоответствующего контекста в лучшем случае порождаемый шаблоном документа будет содержать в полях слияния сообщение об ошибке (при подмене контекста окна операции движения или окна свойств субъекта права). В худшем же случае без всяких предупреждений могут быть сформированы документы с совершенно неправильными данными – такое возможно, когда контекст представлен значением ID экземпляра сущности, отличной от той, для окна свойств которой создан данный SAUMI-Pascal обработчик (например, когда обработчик предполагает в контексте ID платежа, а шаблон назначен в окне дополнительных начислений).

Шаблоны MS Word, не использующие переменную CONTEXT, могут назначаться в контекстное меню работы с документами в любых окнах, поддерживающих такую работу.

## 17.4. Порядок работы с HTML-шаблонами

Основой применения HTML-шаблонов является язык разметки HTML, позволяющий сформировать статическую, постоянную часть текста документа, а также задать его оформление – структуру из заголовков, абзацев, таблиц, маркированных и нумерованных списков, а также других структурных элементов, поддерживаемых в HTML, а кроме того задать визуальное представление текста: шрифты, в которых отображаются элементы текста, и их параметры (размер, наклон, толщина, цвет), отступы, выравнивание и др.

Динамические фрагменты документа формируются посредством кода на языке C# с помощью данных, извлекаемых из БД SAUMI.

Файл документа, формируемого на основе HTML-шаблона, по умолчанию имеет тип .rtf.

### Замечание

Настоящее издание ни в коей мере не является руководством ни по языку HTML, ни по языку C#. Соответствующие материалы об указанных языках широко распространены, как в сети Интернет, так и в многочисленных книжных изданиях.

## 17.4.1. Структура HTML-шаблона

Структурно HTML-шаблон представим в виде набора файлов некоторых типов:

- файл \*.aspx (компилируемая страница ASP – обязательно присутствует в единичном экземпляре);
- файл \*.cs (подключаемый файл с кодом С#, может отсутствовать);
- файл \*.config (файл конфигурации, определяет настройки для компилятора и опции генерации документа – обязателен в единственном числе);
- файлы ресурсов \*.jpg, \*.jpeg, \*.bmp, \*.gif, \*.png;
- файл сборки – динамически подключаемая dll библиотека;
- другие типы файлов, которые могут быть использованы при создании шаблона.

Скомпонованный и отлаженный шаблон сохраняется либо на локальном компьютере в виде одного архивного файла с расширением .spz, содержащего все указанные выше компоненты, либо в БД SAUMI.

Создание, загрузка из файла и редактирование HTML-шаблона, реализующие его назначение виду документа и подстройку под потребности пользователей SAUMI, производятся с помощью модуля «Менеджер HTML-форм».

## 17.4.2. Назначение HTML-шаблона

Назначение HTML-шаблона виду документов в окне свойств субъекта права или операции движения осуществляется выбором в контекстном меню документа данного вида, присутствующего в пакете документов, пункта «Назначить HTML-шаблон».

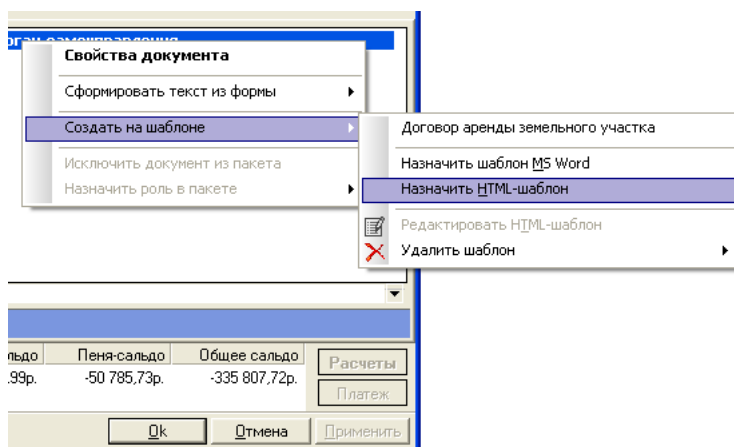


Рисунок 17.8.

В результате открывается окно, в котором следует либо выбрать существующий шаблон (отображаются названия всех шаблонов, и MS Word и HTML, привязанных к данному типу документа), либо задать новое название шаблона. Если будет выбрано название из списка – существующий шаблон будет замещён вновь назначаемым.

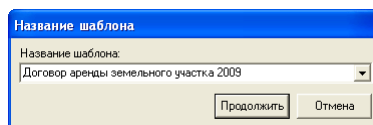



Рисунок 17.9.

После нажатия кнопки «Продолжить», открывается пустое окно редактора HTML-шаблонов «Менеджер HTML-форм», с помощью которого осуществляется формирование, корректировка и отладка HTML-шаблона. Завершается назначение шаблона сохранением его в БД SAUMI, вызываемого нажатием на кнопку , расположенную на инструментальной панели «Менеджера HTML-форм справа».

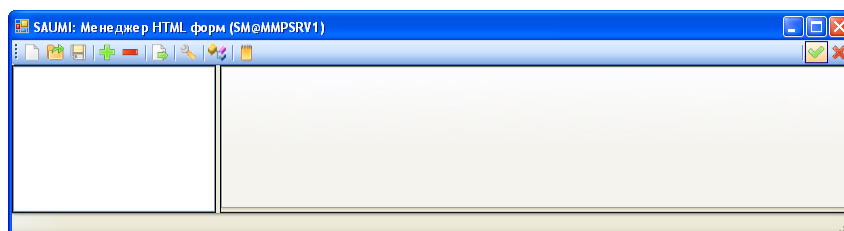



Рисунок 17.10.

Получение конечного HTML-шаблона, способного генерировать текст документа, реализуется двумя способами: созданием шаблона «с нуля» или загрузкой в редактор подготовленного на стороне архивного spz-файла с возможным последующим редактированием компонент шаблона.

Источником HTML-шаблонов являются пакеты обновления SAUMI, которые зачастую содержат spz-файлы новых или модифицированных шаблонов. Также возможно создание нового шаблона на основе уже зарегистрированного в БД – для этого последний сначала должен быть выгружен в архивный spz-файл, а затем загружен в «Менеджер HTML-форм» для редактирования при новом назначении шаблона.

### 17.4.3. Создание пустого шаблона

Создание нового шаблона «с нуля» производится редактированием исходного пустого шаблона, имеющего все необходимые и исходно согласованные компоненты. Чтобы задать пустой шаблон нужно нажать на кнопку  инструментальной панели «Менеджера». Все файлы, входящие в состав нового HTML-шаблона, имеют основное имя default и различаются расширениями.

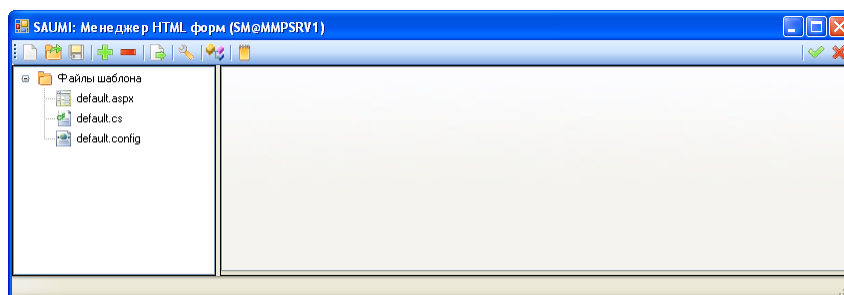



Рисунок 17.11.

## 17.4.4. Загрузка HTML-шаблона из spz-файла

Чтобы загрузить HTML-шаблон из архивного spz-файла нужно нажать на кнопку  инструментальной панели «Менеджера», которая вызывает стандартное окно Windows выбора файла. Указание нужного spz-файла приводит к загрузке всех его компонент в редактор.

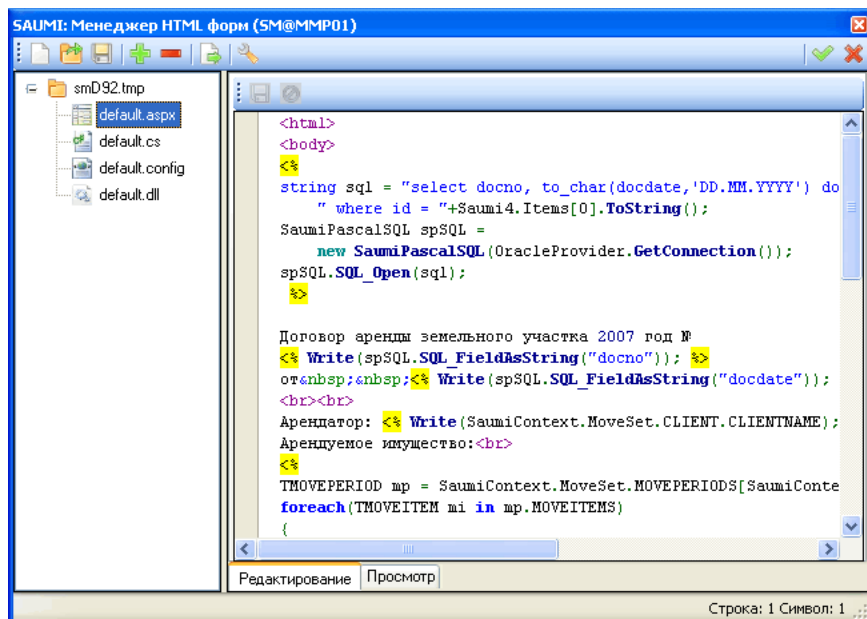


Рисунок 17.12.

## 17.4.5. Редактирование назначенного HTML-шаблона

Также редактор HTML-шаблонов может быть вызван для изменения шаблона, уже назначенного виду документа. Вызов редактора осуществляется из контекстного меню работы с документом выбором последовательно пунктов: «Создать на шаблоне» – «Редактировать HTML-шаблон» – «название шаблона, который следует отредактировать», причем в списке шаблонов отображаются названия только HTML-шаблонов, зарегистрированных в БД для данного вида документов.

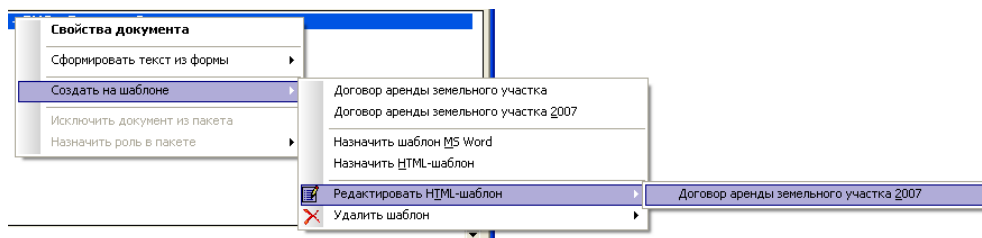


Рисунок 17.13.

Выбор одного из шаблонов в списке приводит к загрузке всех его компонент из БД в редактор.

## 17.4.6. Интерфейс создания / редактирования шаблона

Основная рабочая область менеджера HTML форм делится на две области:

- список файлов проекта;
- область редактирования/просмотра.

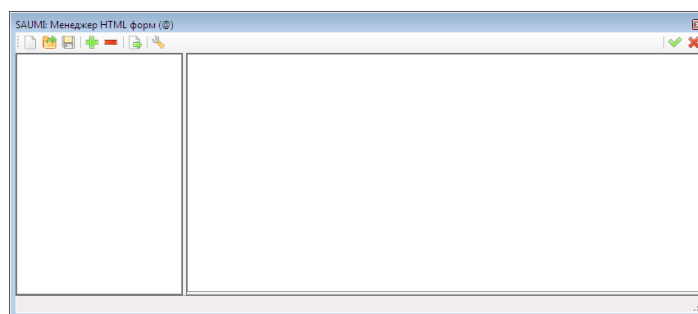


Рисунок 17.14. Окно редактора шаблона

В области файлов проекта представлены имена всех файлов, задействованных в создании шаблона: ASP страница, C# файл, файл конфигурации и др. При выделении любого типа файла в списке файлов шаблона для пользователя становятся доступны элементы управления, соответствующие редактированию/просмотру шаблона или его ресурсов, расположенные на правой рабочей области.

Для `aspx`-файла доступны редактирование и просмотр.

Для `config`- файла доступно редактирование.

Для `cs`-файла доступно редактирование.

Для всех остальных типов файлов (ресурсов шаблона) только просмотр.

Выполнение двойного щелчка по имени файла в списке файлов шаблона (левая область) открывает выделенный файл внешним приложением, ассоциированным с типом расширения выделенного файла.

Рассмотрим элементы управления для редактирования и просмотра `aspx`-страницы. При выборе `aspx`-файла его содержимое доступно для редактирования на вкладке «Редактирование» окна «Менеджера HTML-форм».

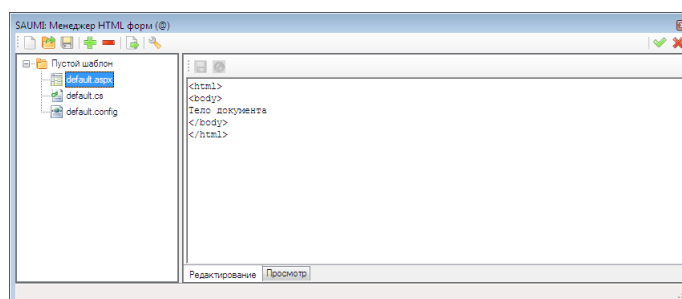


Рисунок 17.15. Редактор `aspx`-страницы

Результаты генерации документа доступны для просмотра на вкладке «Просмотр»,

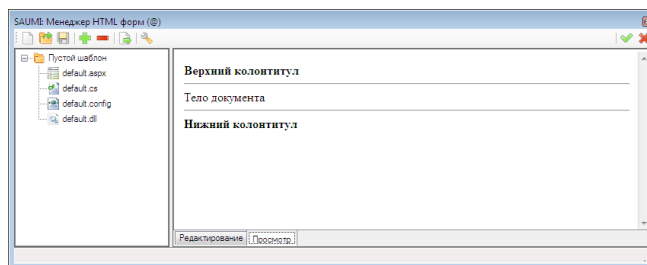


Рисунок 17.16. Просмотр результата

а при ошибках компиляции на вкладке «Просмотр» выводится информация об этих ошибках.

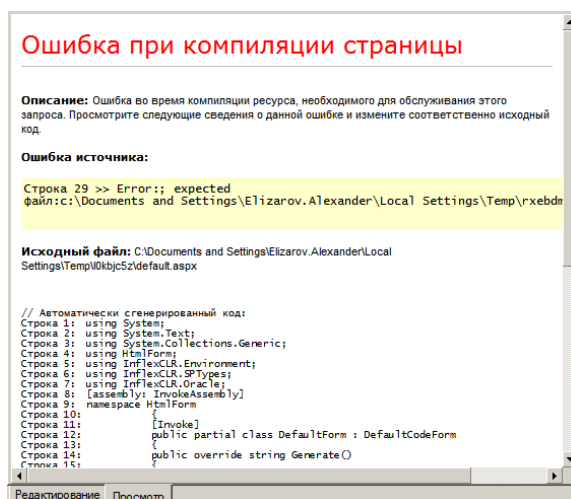










Рисунок 17.17. Отчет об ошибках компиляции

Для управления процессом создания/редактирования шаблона пользователю на инструментальной панели доступны следующие кнопки.

-  Создает шаблон по умолчанию, добавляет в список файлов шаблона следующие файлы: default.aspx, default.cs, default.config. Данный шаблон может служить отправной точкой для дальнейшей разработки сложных шаблонов.
-  Загружает шаблон из архивного spz-файла.
-  Сохраняет шаблон в архивный spz-файл на компьютере пользователя SM без закрытия «Менеджера HTML-форм».
-  Добавляет файл в список файлов шаблона.
-  Удаляет выделенный файл из списка файлов шаблона, если нет выделенного файла, то предлагается удалить все файлы из списка файлов шаблона.
-  Открывает внешнее приложение, ассоциированное для редактирования документов RTF, и загружает в него сгенерированный RTF документ.
-  Открывает окно для создания/редактирования отладочной информации.
-  Сохраняет шаблон в БД SAUMI и закрывает окно «Менеджера HTML-форм» (успешное завершение).

✗ Закрывает окно «Менеджера HTML-форм» без сохранения выполненных изменений.

## Замечание

Файлы ресурсов предпочтительно именовать, используя латинский алфавит. Это условие диктуется наличием различных кодировок используемых html.

### 17.4.7. Настройка конфигурации шаблона

Связь и взаимодействие между имеющимися в HTML-шаблоне страницей ASP, кодом C# и средой MS .NET Framework осуществляется при помощи файла конфигурации с расширением `.config`. Настройка конфигурации шаблона осуществляется визуальными средствами «Менеджера HTML форм» при выборе файла конфигурации в отображаемом на левой панели списке файлов шаблона.

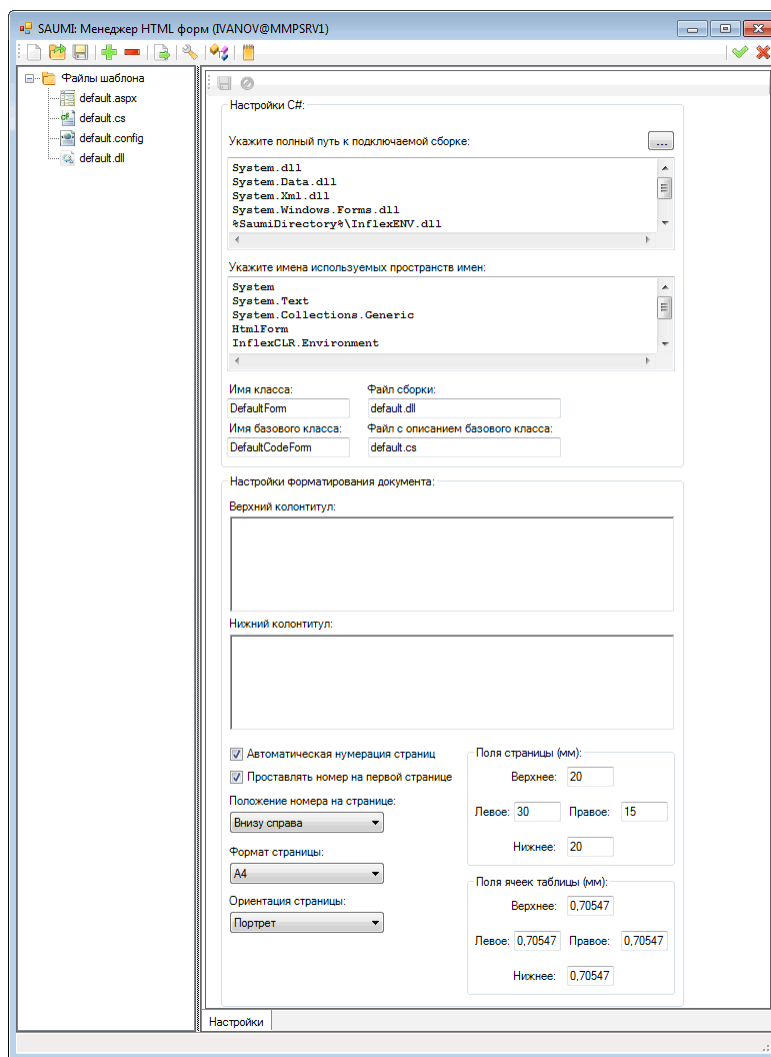


Рисунок 17.18.

На рис. 17.18 представлены параметры по умолчанию, заданные в файле `default.config` только что созданного пустого шаблона. Эти же параметры применяются при генерации документов на основе данного шаблона, если файл конфигурации в шаблоне отсутствует (удален).



Конфигурация определяет параметры компиляции ASP страницы и параметры форматирования документа, располагаемых на правой панели «Менеджера» в секциях «Настройки C#» и «Настройки форматирования документа» соответственно.

Настройки C# по умолчанию имеет следующий вид.

Подключаемые сборки (подключаемые динамические библиотеки .NET).

- System.dll
- System.Data.dll
- System.Xml.dll
- System.Windows.Forms.dll
- %SaumiDirectory%\InflexENV.dll
- %SaumiDirectory%\SPTypes.dll
- System.Data.OracleClient.dll.

### Замечание

Для библиотек из поставки SAUMI в указании пути следует применять префикс %SaumiDirectory%, который указывает каталог установки SAUMI. Библиотеки без указания пути считаются принадлежащими среде MS .NET Framework. Для других библиотек следует указывать абсолютный путь.

Подключаемые пространства имен.

- System
- System.Text
- System.Collections.Generic
- HtmlForm
- HtmlForm.PlugIn
- InflexCLR.Environment
- InflexCLR.SPTypes
- InflexCLR.Oracle

Имя класса `DefaultForm` – это имя присваивается динамически создаваемому классу.

Файл сборки `default.dll` – это имя присваивается динамически подключаемой библиотеке итоговой сборки шаблона, содержащей динамически создаваемый класс.

Имя базового класса `DefaultCodeForm` – указывает базовый класс для динамически создаваемого класса.

Файл с описанием базового класса `default.cs` – содержит декларацию базового класса.

### Замечание

В шаблоне могут находиться и другие `cs`-файлы, но только код `cs`-файла, заданного вышеприведенным параметром, компилируется и исполняется.

Параметры настроек форматирования документа «Верхний колонтитул» и «Нижний колонтитул» могут содержать только статический HTML-код.

## 17.4.8. Принципы программирования шаблона

Основой шаблона является ASP-файл с расширением `.aspx`. Этот файл может содержать, как стандартные теги HTML, так и скриптовые вставки на языке C#, заключенные в специальные скобки `<% %>` (стандарт ASP.NET). Теги HTML организуют структуру документа (заголовки, абзацы, таблицы и т.д.), наполненную постоянным текстом, а скриптовые вставки обеспечивают подстановку в нужных местах переменных фрагментов текста.

### Замечание

«Менеджер HTML-форм» не поддерживает теги и директивы ASP.NET.

Фрагмент ASP-файла шаблона, формирующего текст договора аренды, иллюстрирует подстановку номера в заголовок документа. Полностью указанный шаблон может быть загружен в «Менеджер HTML-форм» (см. 17.4.4) из файла `LeaseXS.spz`, который доступен по ссылке [http://support.saumi.ru/file\\_manager/getFile/?id=LeaseXS.zip](http://support.saumi.ru/file_manager/getFile/?id=LeaseXS.zip), а также содержится в каталоге примеров на дистрибутивном диске SAUMI.

```
<html>
<body>
<%
string sql = "select docno, to_char(docdate,'DD.MM.YYYY') docdate from sm.documents " +
"where id = "+Saumi4.Items[0].ToString();
SaumiPascalSQL spSQL = new SaumiPascalSQL(OracleProvider.GetConnection());
spSQL.SQL_Open(sql);
%>

<h1 align="center">Договор <% Write (" № " + spSQL.SQL_FieldAsString("docno")); %><br>
аренды земельного участка</h1>
... ..
</body>
</html>
```

Код C# из вставок `<% %>` формирует тело динамически создаваемого класса с именем, задаваемым параметром конфигурации «Имя класса». Имя базового класса для динамически создаваемого определяется параметром «Имя базового класса», причем декларация этого класса задается в `.cs`-файле шаблона, указываемом параметром «Файл с описанием базового класса».

Базовый класс должен быть наследником класса `AbstractForm`, реализованного в сборке `HtmlForm.dll`. В целях расширения функциональности такое наследование может быть непрямым, а от другого наследника `AbstractForm`, реализованного в некоторой подключенной с помощью файла конфигурации сборке или в `.cs`-файле шаблона, указываемом параметром «Файл с описанием базового класса». Внутри вставок `<% %>`. доступны типы данных сборок, подключенных при помощи файла конфигурации.

В результате компиляции ASP страницы создается сборка, содержащая класс, генерирующий HTML код из исходной ASP страницы, и выполняющая вставки C# кода из той же страницы. Например, ASP страница

```
<html><body>
<p>Тестовая страница</p>
<% int a = 10; this.Write("Переменная a=" + a.ToString()); %>
</body></html>
```

компилируется в класс C#

```
public class DefaultForm : DefaultCodeForm
{
    public override string Generate()
    {
        this.Write("<html><body><p>Тестовая страница</p>");
        int a=10;
        this.Write("Переменная a=" + a.ToString());
        this.Write("</body></html>");
        return base.Generate();
    }
}
```

Специфицирование текущего экземпляра динамически создаваемого класса производится посредством указателя `this`, а его базового класса – указателем `base`.

Вывод строк в HTML-поток осуществляется при помощи функции `AbstractForm.Write(string Message)`. Аналог этой функции – функция `AbstractForm.WriteLine(string Message)` – осуществляет вывод в HTML-поток строки `Message`, добавляя к ней в конце HTML-тег `<br>`, который в генерируемом тексте вызывает переход на новую строку.

Функция `Generate()` класса `DefaultCodeForm` возвращает имя файла, содержащего сгенерированный из переданных в HTML-поток данных документ, который по умолчанию имеет формат RTF.

Результат генерации файла документа может быть переопределен изменением значений свойств `Convert`, `FileName`, `NeedSave`, объявленных в классе `AbstractForm`, а также перегрузкой виртуального метода `BeforeConvert` этого класса.

Свойство `Convert`, имеющее тип `bool` и значение по умолчанию `true`, определяет, нужно ли производить конвертацию полученных данных HTML-потока в формат RTF; присвоение данному свойству значения `false` отменяет такое преобразование.

Свойство строкового типа `FileName` определяет имя результирующего файла документа. Значение этого свойства задает главный исполняемый модуль SAUMI при вызове шаблона и оно может быть изменено встроенным в шаблон C#-кодом. В БД SAUMI сохраняется содержимое файла (если он существует) с именем, указанным данным свойством класса `AbstractForm`.

Свойство `NeedSave`, также имеющее тип `bool` и значение по умолчанию `true`, определяет, нужно ли производить сохранение полученных данных HTML-потока в файл, заданный свойством `FileName`.

Метод `string BeforeConvert(string htmlStr)` класса `AbstractForm` вызывается после того, как HTML-текст полностью сгенерирован по шаблону, но еще не был конвертирован в RTF формат. Метод может быть перегружен в cs-файле шаблона для реализации некоторого преобразования HTML-текста, представленного параметром `htmlStr`, возвращая строку, содержащую результат преобразования.

Для прерывания генерации документа (к примеру, в случае выяснения отсутствия определенных данных в БД SAUMI), определены два метода.

```
void Terminate();
void TerminateAlert(string Message);
```

Метод `Terminate` просто прерывает генерацию без всяких сообщений, тогда как `TerminateAlert` еще и отображает модальное окно (системный метод `MessageBox`) с текстом сообщения `Message`. В обоих случаях результирующий файл не создается и содержимое документа не сохраняется в БД SAUMI.

## 17.4.9. Формирование динамических фрагментов текста

Переменные составляющие текста документа, генерируемого на основе HTML-шаблона, формируются, как из данных, содержащихся в БД SAUMI, так и еще не сохраненных в БД, но

имеющихся в структурах памяти работающего локально главного модуля SAUMI. Перед началом генерации такого документа SAUMI формирует контекст вызова dll-библиотеки шаблона. Доступ к значению контекста можно получить через свойства статического класса `SaumiContext`.

Указанный класс содержит следующие свойства, передающие контекст генерации документа. Все типы приводимых свойств аналогичны соответствующим predefined типам данным SAUMI-Pascal.

Название свойства	Тип свойства	Комментарий
Address	TADDRESS	Адрес
Client	TCLIENT	Субъект права
Obligation	TOBLIGATION	Обязательство
Movement	TMOVEMENT	Старый вид представления для движений
MoveSet	TMOVESET	Движение
Property	TPROPERTY	Свойства объекта

Свойства статического класса `SaumiContext` получают значения в зависимости от вида сущности SAUMI, из окна свойств которой производится вызов генерации документа на основе HTML-шаблона. Так, при вызове из окна операции движения определено свойство `MoveSet`, тогда как все остальные свойства класса `SaumiContext` имеют значение `null`.

## Замечание

При использовании predefined типов данных SAUMI-Pascal названия всех свойств класса, подобных SAUMI-Pascal типам, нужно писать большими буквами. Например, имя субъекта из контекста окна свойств субъекта права имеет полный идентификатор `SaumiContext.Client.CLIENTNAME`, а имя субъекта операции движения в контексте окна операции движения определяется полным идентификатором `SaumiContext.MoveSet.CLIENT.CLIENTNAME`.

Следующий код представляет пример обращения к контексту вызова.

```
<html>
<body>
<h1>Заголовок документа</h1>
<p>Содержание документа</p>
<%
// Здесь обращение к контексту вызова
if (SaumiContext.MoveSet!=null)
{
this.WriteLine("ИНН субъекта права операции движения: " + SaumiContext.MoveSet.CLIENT.INN);
}
else
{
%>
<h2>документ может быть сформирован только для операции движения</h2>
<%
// Запись сделанная выше эквивалентна
// this.Write("<h2>документ может быть сформирован только для операции движения</h2>");
}
%>
</body>
</html>
```

Дополнительно к описанному выше контексту перед началом формирования шаблона в переменную `Saumi4.Items[0]` помещается идентификатор экземпляра сущности, для которой происходит формирование документа.

```
<html>
<body>
<h1>Заголовок документа</h1>
<p>Содержание документа</p>
<%
// Здесь обращение к ИД
this.WriteLine("ИД = " + Saumi4.Items[0].ToString());
%>
</body>
</html>
```

## 17.4.10. Запросы к базе данных

При формировании динамического содержимого документа можно осуществлять прямой доступ к базе данных. Для этого можно воспользоваться классами `OracleProvider` и `SaumiPascalSQL`. Методы этих классов дублируют методы прямого доступа к базе данных из SAUMI-Pascal.

Пример использования `OracleProvider`.

```
<html>
<body>
<h1>Заголовок документа</h1>
<p>Содержание документа</p>
<%
// Здесь обращение к OracleProvider
int hSQL = OracleProvider.SQL_Open("select * from sm.objects where rownum<=10");
while(!OracleProvider.SQL_EOF(hSQL))
{
this.WriteLine(OracleProvider.SQL_FieldAsString(hSQL, "ID"));
OracleProvider.SQL_Next(hSQL);
}
OracleProvider.SQL_Close(hSQL);
%>
</body>
</html>
```

Пример использования `SaumiPascalSQL`.

```
<html>
<body>
<h1>Заголовок документа</h1>
<p>Содержание документа</p>
<%
// Здесь обращение к SaumiPascalSQL
SaumiPascalSQL spSQL = new SaumiPascalSQL(OracleProvider.GetConnection());
spSQL.SQL_Open("select * from sm.objects where rownum<=10");
while(!spSQL.SQL_EOF())
{
this.WriteLine(spSQL.SQL_FieldAsString("ID"));
spSQL.SQL_Next();
}
spSQL.SQL_Close();
%>
</body>
</html>
```

Для работы с `select`-запросами удобно воспользоваться специальным классом `QuerySQL`.

```
<html>
<body>
<h1>Заголовок документа</h1>
<p>Содержание документа</p>
<%
// Здесь обращение к QuerySQL
```

```
foreach(Row r in QuerySQL.Query("select * from sm.objects where rownum<=10"))
{
    this.WriteLine(r.FieldAsString("ID"));
}
%>
</body>
</html>
```

Так как языком программирования динамического содержания документа является полноценный C#, то для обращения к базе данных можно также воспользоваться стандартными средствами доступа к базам данных или средствами доступа, предоставленными сторонними разработчиками. Для получения текущего соединения необходимо воспользоваться функцией `OracleProvider.GetConnection()`.

## 17.4.11. Использование списков

Язык HTML не поддерживает многоуровневые нумерованные списки, которые являются значимой составляющей структуры текста управленческого (делового) документа. Чтобы обойти это ограничение, класс `AbstractForm` содержит ряд методов, позволяющих генерировать связанные многоуровневые списки, подставлять номера в текст в виде ссылок на элементы списка, а также создавать списки, маркированные задаваемым символом.

```
// Создать список и сделать его текущим
void BeginList();
void BeginList(ListSymbolType Type);
void BeginList(ListSymbolType Type, String CustomSymbol);

// Закончить текущий список; если он был вложен в другой список, то последний продолжить
void EndList();

// Начать следующий уровень и установить его текущим
void NextLevel();
// Закончить текущий уровень
void EndLevel();

// Создать новый пункт в текущем уровне
void Next();
void Next(String Key); // Позволяет создавать пункт, на который можно сослаться, при этом Key должен быть
                       // уникальным в рамках генерации одного документа

// Для работы со ссылками на пункты списка нужно использовать следующие функции:
String Link(String Key); // Возвращает номер (маркер) пункта с ключом Key
bool Exists(String Key); // Возвращает true, если есть пункт с ключом Key
```

### Замечание

Метод `NextLevel()` применим только к спискам, маркированным арабскими цифрами. Иными словами, многоуровневая маркировка только для нумерованных списков.

Тип маркировки или нумерации задается в функции `BeginList` переменной типа перечисления `ListSymbolType`.

```
public enum ListSymbolType {
    ARAB_DIGIT, // Нумерация арабскими цифрами
    CUSTOM_SYMBOL, // Маркировка произвольным символом
    RUSSIAN_LOWER, // Маркировка русским алфавитом в нижнем регистре
    RUSSIAN_UPPER, // Маркировка русским алфавитом в верхнем регистре,
    LATIN_LOWER, // Маркировка латинским алфавитом в нижнем регистре
    LATIN_UPPER, // Маркировка латинским алфавитом в верхнем регистре
    ROMAN_DIGIT_LOWER, // Маркировка римскими цифрами в нижнем регистре (i, ii, iii, iv, v и т.д.)
    ROMAN_DIGIT_UPPER // Маркировка римскими цифрами в верхнем регистре (I, II, III, IV, V и т.д.)
}
```

}

### Пример построения списков.

```
<html>
<body>
<style>p {font-size: 14pt;}</style>
<p>Начальный текст</p>
<%
// Создадим список, нумерованный арабскими цифрами
this.BeginList();
%>
<p><%this.Next();%>Это пункт 1.</p>
<p><%this.Next("key1");%>Это пункт 2.</p>
<p><%this.Next("key2");%>Это пункт 3.
Использована ссылка на пункт 2 : <%this.Write(this.Link("key1"));%></p>
<%this.NextLevel();%>
<p><%this.Next("key3");%>Это подпункт 3.1.
Использована ссылка на пункт 3 : <%this.Write(this.Link("key2"));%></p>
<p><%this.Next();%>Это подпункт 3.2.
Использована ссылка на подпункт 3.1 : <%this.Write(this.Link("key3"));%></p>
<%this.EndLevel();%>
<p><%this.Next("key4");%>Это пункт 4.</p>
<%
// Создадим список, нумерованный кириллическими буквами со скобкой
this.BeginList(ListSymbolType.RUSSIAN_LOWER, "");
%>
<p><%this.Next();%>Это элемент а) перечисления пункта 4.</p>
<p><%this.Next("key_be");%>Это элемент б) перечисления пункта 4.</p>
<p><%this.Next();%>Это элемент в) перечисления пункта 4.</p>
<%EndList();%>
<p><%this.Next("key5");%>Это пункт 5.</p>
<%
// Создадим маркированный список с маркером в виде средней длины тире - униккод 0x2013
this.BeginList(ListSymbolType.CUSTOM_SYMBOL, ('\u2013').ToString());
%>
<p><%this.Next();%>Это элемент перечисления пункта 5.</p>
<p><%this.Next();%>Это элемент перечисления пункта 5.
Использована ссылка на перечисление <%this.Write(this.Link("key_be"));%>
пункта <%this.Write(this.Link("key4"));%>
</p>
<%EndList();%>
<p><%this.Next();%>Это пункт 6.</p>
<%EndList();%>
<p>Оконечный текст</p>
</body>
</html>
```

### Генерируемый текст.

Начальный текст

1. Это пункт 1.
2. Это пункт 2.
3. Это пункт 3. Использована ссылка на пункт 2 : 2.
- 3.1. Это подпункт 3.1. Использована ссылка на пункт 3 : 3.
- 3.2. Это подпункт 3.2. Использована ссылка на подпункт 3.1 : 3.1.
4. Это пункт 4.
- а) Это элемент а) перечисления пункта 4.

- б) Это элемент б) перечисления пункта 4.
  - в) Это элемент в) перечисления пункта 4.
  - 5. Это пункт 5.
  - Это элемент перечисления пункта 5.
  - Это элемент перечисления пункта 5. Использована ссылка на перечисление б) пункта 4.
  - 6. Это пункт 6.
- Оконечный текст

## 17.4.12. Использование стилей

Допускается в HTML-шаблон вставлять элементы стилей (CSS). Рекомендуется использовать линейные (атрибут `style` отдельного тега) и вложенные стили (тег `<style>` с правилами стилей) в `aspx`-файле. Использование отдельного файла стилей хотя и допускается посредством добавления его в список файлов шаблона и указания на него ссылки из `aspx`-файла, однако средства его редактирования в редакторе HTML-шаблона отсутствуют, и поэтому следует использовать уже отлаженный отдельно `css`-файл.

### Замечание

Все файлы, включенные в HTML-шаблон, во время формирования текста документа распаковываются из БД SAUMI в одну временную папку, а потому при подключении `css`-файла с помощью тега `<link>` необходимо в атрибуте `href` указывать только имя этого файла без уточнения пути.

```
<link rel="stylesheet" type="text/css" href="common.css">
```

Также в атрибутах ссылок не следует изменять путь и для других файлов, включенных в шаблон.

Тег `<hr>` не поддерживает стили, но может использовать следующие атрибуты: `size` – высота в мм, `width` – ширина в мм, `align` – выравнивание.

Для свойств стилей `margin-top` и `margin-bottom` единицей измерения являются пункты, определяющие размеры шрифтов в MS Word (1 пункт ~ 0,3528 мм).

При установке межстрочного интервала с помощью свойства стиля `line-height` для корректной трансляции `html`-файла в формат `rtf` следует указывать единицу размера интервала `em` (`em` – текущий размер шрифта). В частности, полуторный межстрочный интервал задается в описании следующим свойством: `line-height:1.5em`.

При описании правил стилей не следует в одном правиле указывать более одного селектора. Например, вместо одного правила

```
h1, h2, h3 {text-align:right}
```

следует задать три правила

```
h1 {text-align:right}
h2 {text-align:right}
h3 {text-align:right}
```

При преобразовании HTML содержимого в RTF учитываются только следующие свойства стилей:



- color
- background-color
- font
- font-weight
- font-family
- font-size
- text-align
- text-indent
- font-style
- margin-top
- margin-bottom
- line-height
- top
- left
- width
- text-decoration
- vertical-align
- page-break-before
- page-break-after
- list-style-type

### 17.4.13. Обработка штрих-кодов

Вместе с SAUMI поставляется библиотека `iTextSharp` <http://itextsharp.sourceforge.net/>. Менеджер HTML-форм может работать с этой библиотекой. Одной из возможностей, реализованной в `iTextSharp`, является поддержка штрих-кодов. Для обработки штрих-кодов в проекте нужно добавить `iTextSharp`, как подключаемую сборку, и написать код, подобный ниже следующему.

```
// Генерируем изображение штрих кода и сохраняем его на диск
BarcodeCodabar codabar = new BarcodeCodabar();
// 1234567 - кодб а/а - стартовый и стоповый символ
codabar.Code = "a1234567a";
codabar.CreateDrawingImage(System.Drawing.Color.Black, System.Drawing.Color.White)
.Save("C:\\1234567.gif", System.Drawing.Imaging.ImageFormat.Gif);
```

Пример реализации HTML-шаблона, обрабатывающего штрих-коды, может быть загружен в «Менеджер HTML-форм» (см. 17.4.4) из файла `codabar.spz`, который доступен по ссылке [http://support.saumi.ru/file\\_manager/getFile/?id=codabar.zip](http://support.saumi.ru/file_manager/getFile/?id=codabar.zip), а также содержится в каталоге примеров на дистрибутивном диске SAUMI.

## 17.4.14. Отладка HTML-шаблона при помощи Microsoft Visual Studio

Для отладки кода HTML-шаблона необходимо использовать сторонний отладчик. Ниже описывается способ отладки при помощи интегрированной среды разработки Microsoft Visual Studio на примере простого модельного кода, представленного на рисунке в окне «Менеджера HTML-форм».

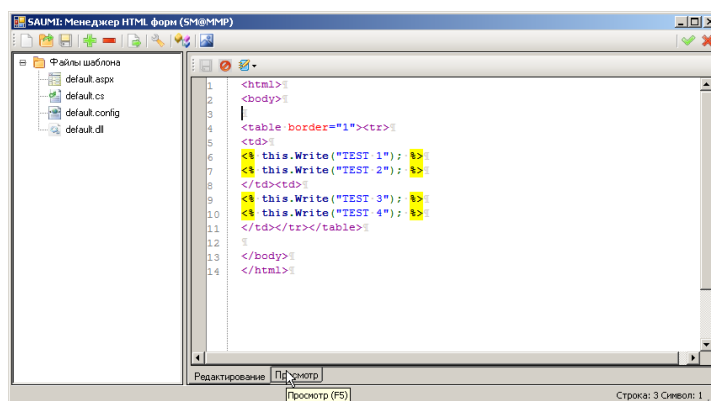


Рисунок 17.19.

Параллельно необходимо запустить Microsoft Visual Studio и в меню **Tools** выбрать пункт «Attach to Process ...».

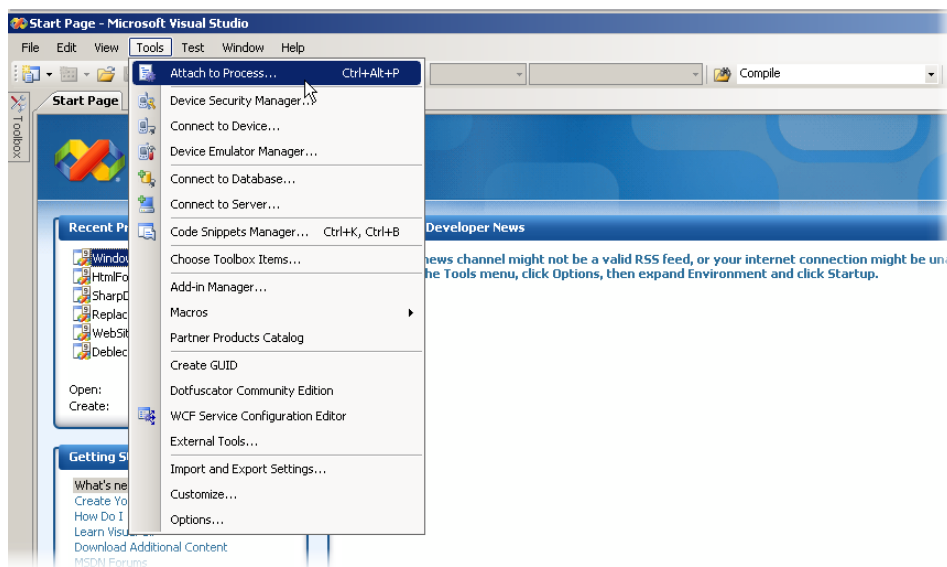


Рисунок 17.20.

В открывшемся окне со списком всех запущенных процессов, которые можно отлаживать данным отладчиком, следует выбрать процесс SAUMI – mmp32.exe.

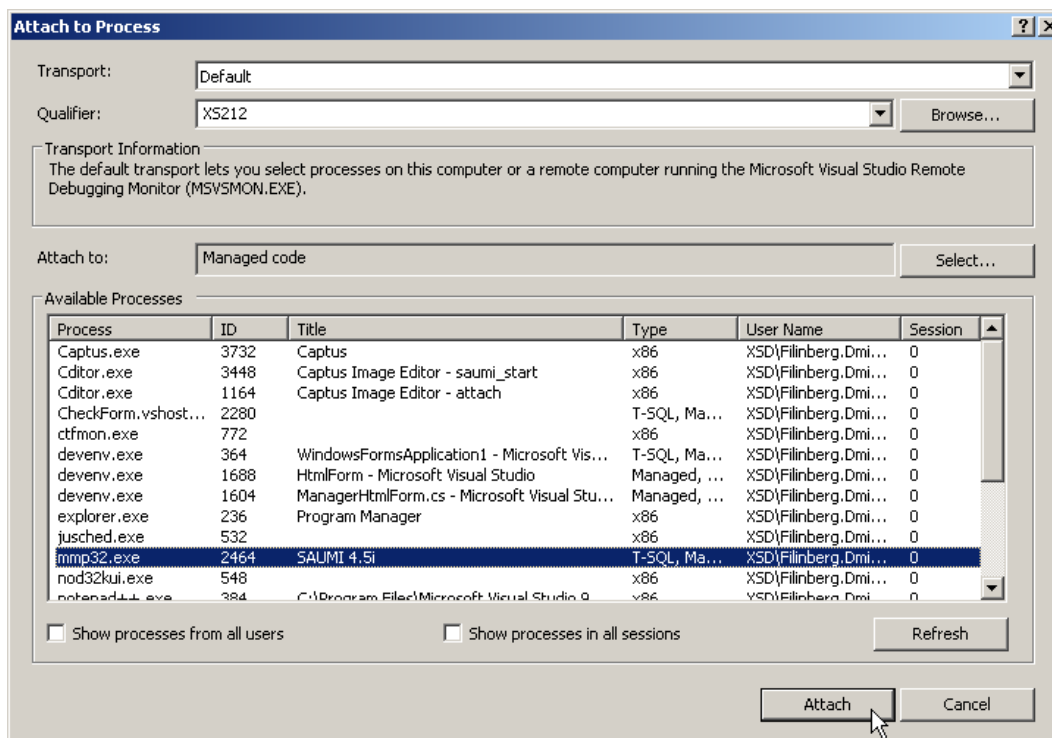


Рисунок 17.21.

После нажатия кнопки **Attach** или двойного щелчка по строке `mmp32.exe` в списке процессов (рис. 17.21) отладчик подключается к процессу SAUMI.

Далее следует переключиться в «Менеджер HTML форм», выбрать для редактирования `aspx`-файл и на правой панели редактора выбрать вкладку **Просмотр** или нажать кнопку **F5** (рис. 17.19). При этом начнется генерация документа, и отладчик перейдет в самое начало функции генерации документа.

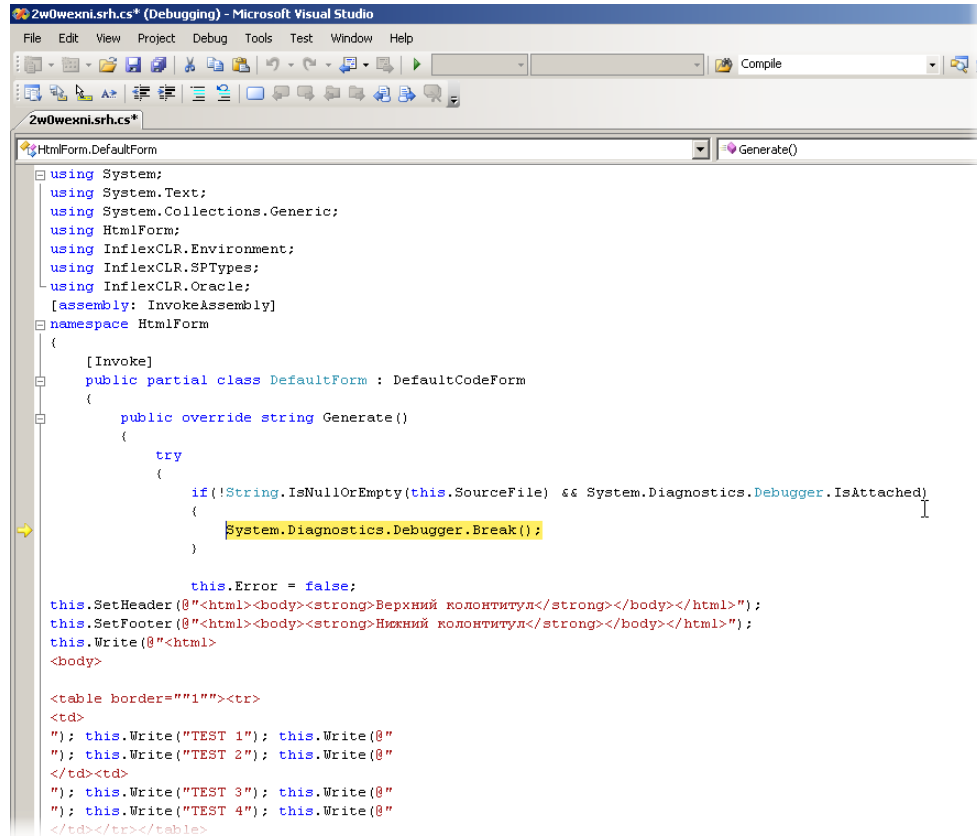


Рисунок 17.22. Переход отладчика к коду класса, сгенерированного на основе ASP-кода

Теперь при помощи отладчика можно просматривать значения локальных переменных, значения объектов и значения каждого поля доступного объекта. Для переключения между режимами дизассемблированного кода и исходного хода в контекстном меню (в окне редактирования) нужно выбрать **Go To Disassembly** (переключится на IL код) или **Go To Source Code** (переключится на отладку C# кода).

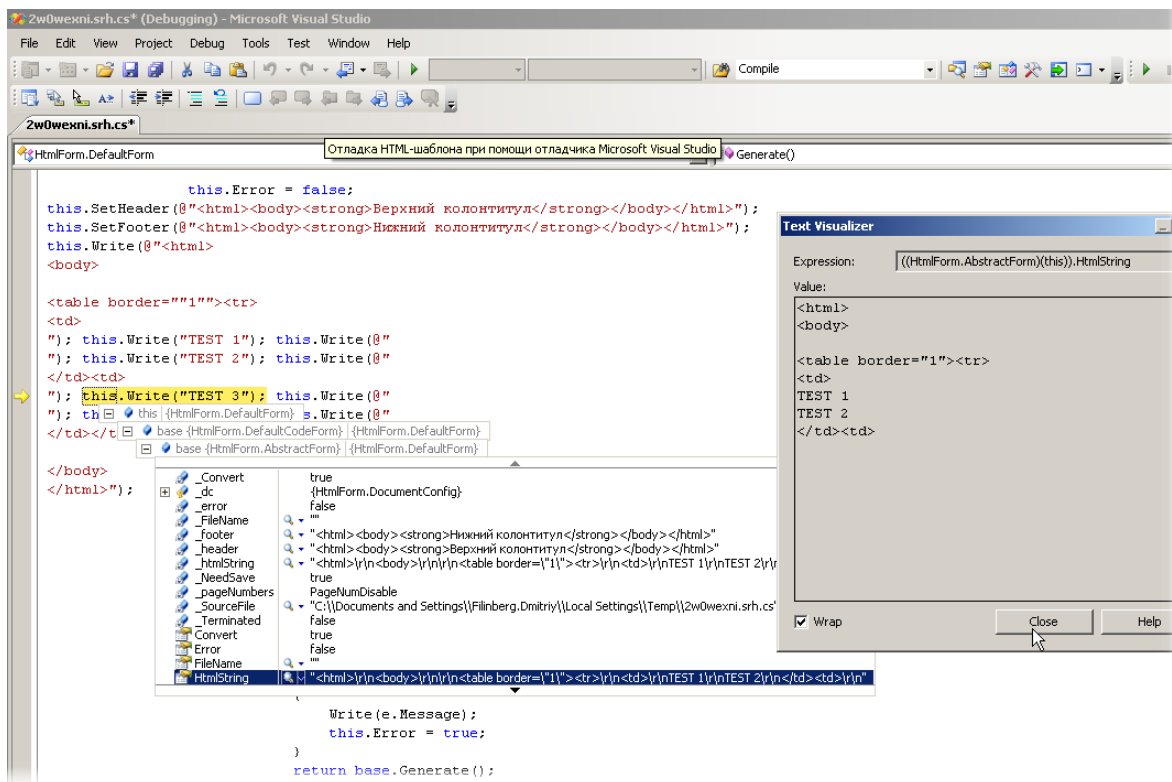


Рисунок 17.23. Процесс отладки

Горячие клавиши отладчика Microsoft Visual Studio:

- F5 – продолжить выполнения без остановки;
- F9 – установить точку остановки отладчика, в текущей строке (где находится каретка);
- F10 – выполнить очередной шаг;
- F11 – выполнить очередной шаг и войти внутрь тела метода.

Более полное описание отладчика Microsoft Visual Studio можно найти по ссылке <http://msdn.microsoft.com/>.

### Замечание

Отладчик Microsoft Visual Studio поддерживает технологию, которая позволяет во время отладки изменять код и переносить текущий контекст выполнения вперед или назад по исходному коду, однако не стоит забывать, что для отладки доступен исходный файл, который динамически генерируется приложением «Менеджер HTML-форм», таким образом, изменения в отлаживаемом коде никак не отразятся на исходном ASP-коде, представленном в окне «Менеджера HTML-форм».

---

# Глава 18. Считыватель штрих-кодов

## Содержание

18.1. Оборудование для работы с штрих-кодами .....	223
18.1.1. Печать изображения штрих-кода .....	223
18.1.2. Считывание штрих-кода .....	224
18.2. Кодирование информации в штрих-коде .....	224
18.3. Настройка SAUMI для работы со сканером штрих-кодов .....	225
18.4. Настройка SAUMI на произвольную обработку считанного штрих-кода .....	225

Применение штрих-кодов в SAUMI позволяет существенно сократить время поиска информации. По документу, который пользователь держит в руках, с нанесенным на него штрих-кодом, система однозначно определяет, что это за документ и отображает соответствующую этому документу информацию на экране.

Так, например, подготовленные квитанции для оплаты по договорам аренды работников соответствующего отдела обрабатываются на порядки быстрее, чем поиск в системе соответствующего договора, по неоднозначным данным содержащимся в квитанции. Плательщики не всегда точно указывают в квитанции, за что производится оплата – бывают ошибки в указании адресов, номеров договоров и т.д. В результате есть вероятность, что оператор отдела аренды, ответственный за внесение информации по платежам, отнесет платеж не к тому обязательству. То есть использование штрих-кодов увеличивает скорость обработки документов и существенно снижает вероятность ошибок.

## 18.1. Оборудование для работы с штрих-кодами

Весь процесс использования штрих-кодов разбивается на два этапа, на каждом из которых применяется соответствующее оборудование. Первый этап – печать изображения штрих-кода на договорах, квитанциях, самоклеющихся этикетках. Второй этап – считывания изображения, его декодирование и передача в компьютер на обработку программе.

### 18.1.1. Печать изображения штрих-кода

Так как рисунок штрих-кода – это всего лишь графическая картинка, то он может быть распечатан любыми существующими средствами печати, позволяющими работать с выбранным разрешением. То есть для печати может быть использован обычный лазерный или струйный принтер, для крупных этикеток допускается использование даже матричных принтеров.

В случаях необходимости печати большого числа самоклеющихся этикеток, которые предполагается использовать на учитываемом имуществе вместо нанесения инвентарных номеров краской, имеет смысл использовать специальные принтеры для печати на этикетках. Их достоинство – очень высокая скорость печати. Эти принтеры бывают нескольких типов, различающихся принципом печати и многими другими характеристиками.

Следует особо отметить, какой тип не рекомендуется применять в области учета имущества. К этому типу относятся термопринтеры. Дело в том, что их принцип основан на изменении цвета специальной бумаги при нагреве. Это очень удобно для розничной торговли, так как не требуется краска. Отрицательная сторона этого способа в том, что специальная бумага для этих принтеров чернеет со временем. В результате если на термопринтере распечатать наклейки с реестровыми

и инвентарными номерами в виде штрих-кодов, то через несколько лет их изображение будет невозможно считать.

В качестве компромисса рекомендуется использовать листы самоклеющейся бумаги формата А4, на которой можно печатать с помощью лазерного принтера. Бумага должна быть тонкой и с очень клейким слоем.

### **18.1.2. Считывание штрих-кода**

Существует множество различных устройств, предназначенных для считывания изображения штрих-кода. Различаются они максимальной шириной изображения, расстояния с которого возможно считывание, поддержкой различного набора стандартов штрих-кода, интеллектуальности, методом подключения к компьютеру, надежности, дизайна, и пр. Все они называются также сканерами штрих-кода. При этом нет разницы, каким способом было распечатано изображения штрих-кода.

SAUMI не предъявляет особых требований к считывателям штрих-кода. В основу были приняты общие стандарты обмена данных между считывателем и компьютером через интерфейс RS232 (COM-порт).

Комплекс SAUMI совместим со сканерами CIPHER 1067/1090/1021+ и аналогичными методом передачи информации через RS232 интерфейс.

Некоторые сканеры штрих-кода предоставляют возможность даже при отключении от компьютера считывать штрих-коды и запоминать их, а после подключения передать их в компьютер. Это может быть удобно, если для работы с документами требуется сходить в архив, с каждого документа считать код, вернуться к компьютеру и запросить список этих документов через SAUMI.

Существует особый вид считывателей штрих-кодов, способных работать без подключения к компьютеру, относящиеся к особо интеллектуальным. Называются они портативными терминалами считывания штрих-кодов. Их использование целесообразно при инвентаризации имущества. Портативные терминалы позволяют не только считать штрих-код и в дальнейшем передать компьютеру, но и сравнивать списки запомненных кодов, которые загружаются перед инвентаризацией из SAUMI, отображать на собственном экране наименование имущества, которое еще не освидетельствовано. Объем внутренней памяти этих устройств позволяет запомнить списки имущества нескольких организаций.

Инвентаризация, полностью проводимая с помощью портативных терминалов считывания штрих-кодов, позволяет исключить фальсификацию данных инвентаризации имущества.

## **18.2. Кодирование информации в штрих-коде**

В комплексе SAUMI заложены определенные алгоритмы обработки информации закодированной в штрих-коде. Знание этих алгоритмов позволяет администратору комплекса самостоятельно определять поведение системы в зависимости от информации в штрих-коде.

В качестве основной кодировки, используемой в SAUMI, является стандарт штрих-кодирования Codabar. Это один из наиболее устойчивых и безопасных кодов.

Любой штрих-код, считанный SAUMI, должен иметь определенный формат: Первые две цифры определяют тип данных в соответствии с «Менеджером отчетов» (номер отчета), затем следует цифра, определяющая основное действие, далее идет символ минус "-", после чего указывается идентификационный номер данных.



Рисунок 18.1.

В некоторых случаях необходимо хранить с кодом дополнительную информацию. В этих случаях её дописывают к коду через знак '/', например: 013-45323/58224.

Предопределенные типы данных SAUMI для использования штрих-кодах:

- 01 – объекты;
- 02 – документы;
- 03 – субъекты права;
- 04 – ценные бумаги;
- 25 – операции движения;
- 05 – обязательства;
- 61 – платежи;
- 08 – операции с ценными бумагами;
- 06 – состояние объекта.

Предопределенные типы данных SAUMI, допустимые для использования в штрих-кодах:

- 2 – удалить данные;
- 3 – отобразить окно свойств для соответствующих данных;
- 4 – внести информацию подчиненную идентификационному коду.

Дополнительная информация используется определяемыми обработчиками. Например, при регистрации платежа код 054-333333/500000 будет означать внесение платежа в размере 5000,00 рублей на обязательство № 333333.

### 18.3. Настройка SAUMI для работы со сканером штрих-кодов

Как уже было сказано выше, SAUMI поддерживает только сканеры, передающие информацию через RS232 интерфейс, т.е. подключаемые к COM-порту. Настройка SAUMI для работы со сканером штрих-кодов производится, как правило, администратором комплекса, хотя эти действия может производить любой опытный пользователь.

Для того чтобы активизировать работу со сканером штрих-кода необходимо в папке установки SAUMI (по умолчанию c:\SAUMI) найти файл SAUMI.INI. Этот файл содержит параметры работы SAUMI на клиентской станции. В разделе BarCode, обозначенном квадратными скобками, определяются четыре параметра, значения которых влияют на поддержку сканера штри-кодов. Описание указанных параметров приводится в главе 5 настоящего руководства.

### 18.4. Настройка SAUMI на произвольную обработку считанного штрих-кода

Начиная с версии SAUMI 4.3.1 появилась возможность обрабатывать полученную после считывания штрих-кода информацию произвольным образом. Для того чтобы такая обработка была возможна,



Администратором SAUMI должен быть описан обработчик события считывания штрих-кода `OnBarcodeEvent`. Начиная с версии 4.3.4 имеется возможность в этом обработчике изменить содержимое закодированной информации и передать новое значение переменной `Barcode` для обработки его по умолчанию SAUMI. Для этого, после внесения изменений в `Barcode` параметру `Handled` надо присвоить значение `false`.

Помимо этого, в версии 4.3.4, расширен формат кодируемой информации для штрих-кодов, используемых при регистрации платежа (на бланках квитанции на оплату, например). Теперь для регистрации платежа, штрих-код может задаваться в следующем формате.

```
054-<номер обязательства (ID из таблицы OBLIGATIONS)>/  
<Сумма платежа, умноженная на 100>/  
<ID из таблицы CLS_KBK (ID записи из справочника Кодов бюджетной классификации)>
```

Пример: **054-1256/1893200/4** – здесь

**054** – зарегистрировать платёж по обязательству ( **05** – тип данных "обязательство", **4** – внести информацию);

**1256** – номер обязательства;

**1893200** – сумма платежа, умноженная на 100, т.е. платёж на сумму 18932,00 руб.;

**4** – ID записи из справочника КБК, к которому следует отнести данный платёж.

Допустимо закодировать и в прежнем формате: **054-1256/1893200** , где

**054** - зарегистрировать платёж по обязательству;

**1256** – номер обязательства;

**1893200** – сумма платежа, умноженная на 100, т.е. платёж на сумму 18932,00 руб.

Поскольку ID КБК отсутствует, данный платёж не будет отнесён к какому-либо КБК, и оператору надо будет сделать это вручную в окне регистрации платежа.

---

# Глава 19. Сбор отчетных данных в электронном виде

## Содержание

19.1. Организационная схема применения DataStretch .....	227
19.2. Задание нового отчетного периода .....	228
19.3. Формирование набора форм .....	230
19.4. Новые и измененные формы .....	232
19.5. Активирование нового отчетного периода .....	236
19.6. Удаление отчетного периода .....	236
19.7. Набор файлов отчетности .....	237
19.8. Формат mro-файла .....	237
19.8.1. Секция #OPTIONS .....	237
19.8.2. Секция #DATAOPT .....	238
19.8.3. Секция #SAVEDATA .....	242
19.8.4. Секция #EXEC_SAVE .....	243
19.8.5. Секция #EXEC_LOAD .....	244
19.8.6. Секция #LISTS .....	247
19.8.7. Порядок обработки принимаемых отчетных данных .....	248

Порядок отчетности перед комитетом по управлению муниципальным (государственным) имуществом подотчетных организаций определяется различными нормативно-правовыми актами. Комплекс SAUMI предоставляет программное средство сбора и передачи отчетности в электронном виде, позволяющее учитывать в БД SAUMI отчетные данные, касающиеся имущества, находящегося в хозяйственном ведении или оперативном управлении подотчетных организаций. Указанное средство именуется DataStretch.

## 19.1. Организационная схема применения DataStretch

На основании локальных распорядительных документов администратором SAUMI задается, а затем и активируется новый отчетный период. Во временной промежуток между заданием нового периода и его активированием производится вся подготовительная работа, в результате которой в соответствии с нормативами актуализируются формы отчетности для различных видов организаций, справочники, передаваемые в эти организации, а также перечень самих отчитывающихся организаций. После активирования нового периода изменить перечень и сами формы отчетности, справочники и перечень организаций нельзя до объявления нового отчетного периода.

Для каждой подотчетной организации сотрудниками Комитета (пользователями SAUMI) на мобильном носителе (дискета, flash-диск) на основе имеющихся в БД SAUMI данных осуществляется подготовка массива информации, позволяющего с помощью программного модуля DataStretch вводить отчетные данные в соответствующие формы ввода.

Носитель передается в подотчетную организацию, ответственные работники которой запускают содержащийся на переданном носителе установочную программу, развертывающую DataStretch и сопутствующие данные на локальном компьютере. Далее осуществляется ввод данных отчетной организации в представленных формах ввода DataStretch. Результат такого ввода сохраняется на мобильном носителе и передается обратно в Комитет.

Пользователи SAUMI вызывают обработку данных с мобильных носителей, полученных из отчитывающихся организаций, в БД SAUMI. При обработке информации об имуществе, записанной на сменном носителе, автоматически проверяется ее соответствие и согласованность с содержащимися в БД данными. При отсутствии выявленных ошибок новые данные записываются в БД SAUMI. При выявлении несоответствий отчет не принимается, о чем выводится соответствующее сообщение, и представляется протокол выявленных несоответствий.

## Замечание

С развитием Интернета и подключением к нему Комитетов и отчитывающихся организаций все более широкое применение получает передача данных отчетности посредством электронной почты.

## 19.2. Задание нового отчетного периода

Перед заданием нового отчетного периода администратору необходимо иметь регистрационный номер документа, являющегося основанием нового отчетного периода. Указанный номер послужит также основанием регистрации новых еще незарегистрированных объектов имущества и новых операций с этими объектами, информация о которых будет содержаться в подготовленной организацией отчетности. Все операции в SAUMI производятся на основе документов.

Самой благоприятной является ситуация, когда руководство муниципалитета, управляя процессом отчетности в подотчетных организациях, издает соответствующий распорядительный документ на каждый новый отчетный период. Такой документ следует зарегистрировать в SAUMI (см. раздел «Регистрация документов в SAUMI» в главе «Работа с документами в SAUMI» руководства пользователя) и использовать его регистрационный номер (идентификатор в базе данных SAUMI) для указанных выше целей. Регистрационный номер отображается в правом верхнем углу окна свойств документа.

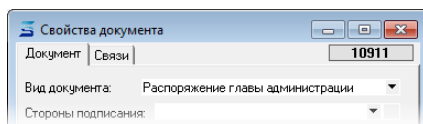


Рисунок 19.1.

В случае, когда реальные распорядительные документы, регулирующие отчетность, не принимаются, нужно воспользоваться регистрационным номером некоторого фиктивного документа, аналогичного «Неизвестному документу» (см. соответствующий раздел в главе «Работа с документами в SAUMI» руководства пользователя). Для такого документа может быть создан тип «Реестр», или «Обработка отчетности», или иной, указывающий на факт регистрации операций на основании отчетных данных и являющийся подтипом «Фиктивных документов».

Новый отчетный период задается посредством утилиты «Регламент отчетности» из комплекта администратора SAUMI. Сразу после развертывания комплекта SAUMI ни один отчетный период еще не задан и не отображается в левой панели, тогда как в правой находится список зарегистрированных на текущий момент в системе отчетных форм.

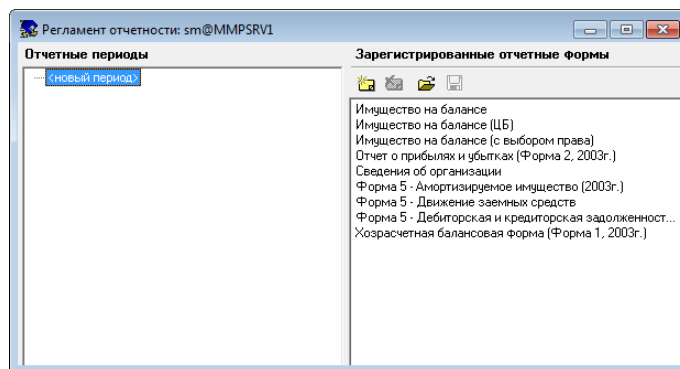


Рисунок 19.2.

Для задания нового периода отчетности следует осуществить двойной щелчок мышью по элементу <новый период> в левой панели окна. В результате откроется окно ввода параметров нового отчетного периода, в котором нужно в полях «Отчетная дата» и «Регистрационный номер документа» указать дату последнего дня отчетного периода и выше обсуждавшийся регистрационный номер документа (идентификатор документа в БД SAUMI) соответственно. По завершении ввода следует нажать кнопку «Ок».

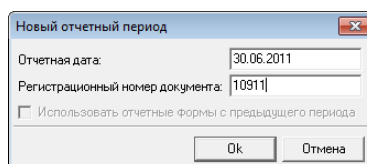


Рисунок 19.3.

В результате будет создан новый отчетный период, отображаемый в левой панели.

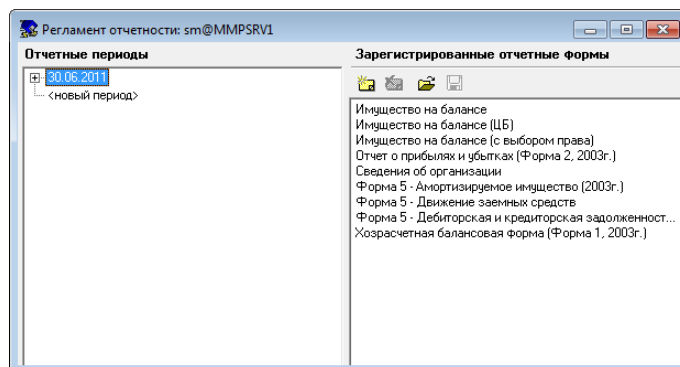



Рисунок 19.4.

## Замечание

Флажок «Использовать отчетные формы с предыдущего периода» в окне создания нового отчетного периода становится доступным при создании следующих за самым первым периодом и позволяет для формирования перечня отчетов использовать перечень

предыдущего периода. В случае сброса этого флажка перечень отчетов формируется «с чистого листа», как в самом первом периоде.

### 19.3. Формирование набора форм

Отчетный период и связанные с ним элементы представляются как ветвь дерева отчетов. При нажатии на кнопку  слева от индикации нового периода эта ветвь раскрывается, представляя вложенный перечень всех типов муниципальных организаций, из которых возможен прием отчетных данных с помощью SAUMI.

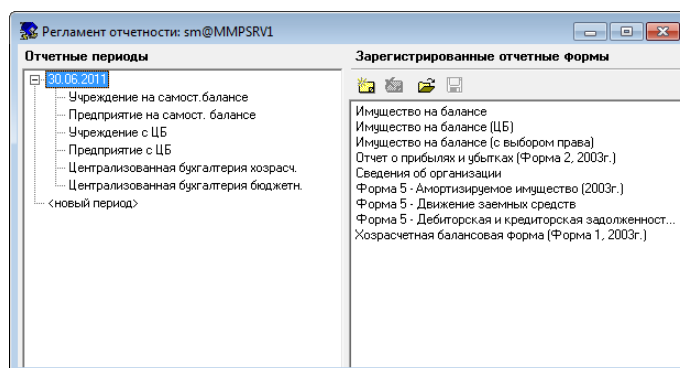


Рисунок 19.5.

При создании самого первого после развертывания SAMI отчетного периода ни одному типу организаций не назначены отчетные формы и их следует назначить, скомпоновав соответствующий набор форм. Рекомендуются следующие наборы форм для различных типов юридических лиц.

Типы юридических лиц	Рекомендуемые формы
«Учреждение на самост. балансе»	«Сведения об организации»
	«Имущество на балансе»
«Предприятие на самост. балансе»	«Сведения об организации»
	«Имущество на балансе»
	«Хозрасчетная балансовая форма»
	«Отчет о прибылях и убытках»
«Централизованная бухгалтерия бюджетн.»	«Сведения об организации»
	«Имущество на балансе (ЦБ)»

#### Замечание

Формы отчетов зависят от типа организации, это значит, что перед принятием отчетности необходимо удостовериться, что у всех отчитывающихся организаций был правильно установлен тип юридического лица, иначе может получиться так, что детскому саду придется заполнять отчет о прибылях и убытках.

Так как за имущество учреждений отчитывается соответствующая централизованная бухгалтерия (ЦБ), то прием отчетности от них необязателен, хотя возможен, включением отчета «Сведения об организации». Важно, чтобы при сборе отчетных данных из ЦБ перед активированием отчетного периода были зарегистрированы *все* учреждения, обслуживаемые централизованными бухгалтериями, иначе бухгалтеры ЦБ просто не смогут соотнести имущество с конкретным учреждением.

Для добавления в набор формы, следует выделить ее мышью в правой панели и, удерживая левую кнопку мыши нажатой, подвести курсор к типу организации на левой панели, а затем отпустить левую кнопку мыши (техника перетаскивания).

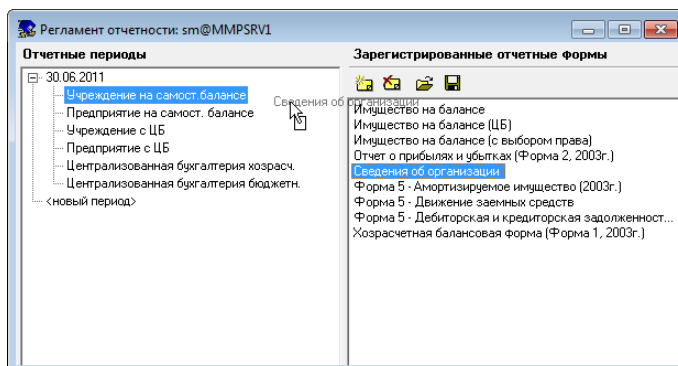


Рисунок 19.6.

В результате в следующем уровне иерархического представления элементов, составляющих отчетный период, отобразится назначенная типу юридического лица форма.

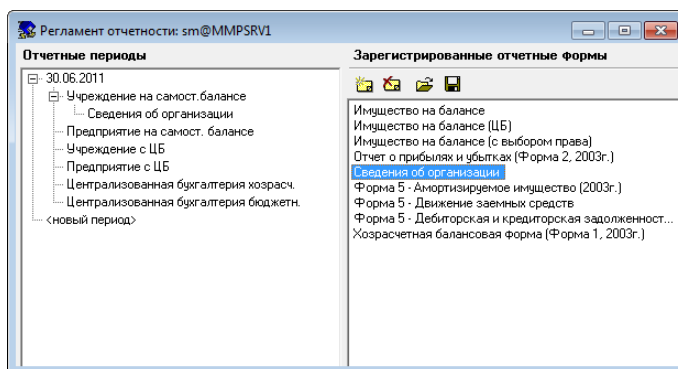


Рисунок 19.7.

Форму «Сведения об организации» следует ставить в наборе первой по порядку. Для изменений порядка форм в наборе используется перетаскивание мышью.

Для удаления из созданного набора нежелательной формы следует выделить эту форму в наборе

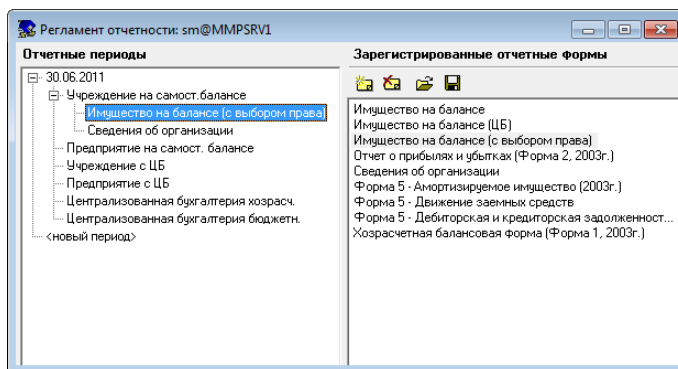


Рисунок 19.8.

и либо нажать клавишу **Del**, либо вызвать правой кнопкой мыши контекстное меню, в котором выбрать пункт «Исключить форму из отчета». После подтверждения удаления форма удаляется из набора.

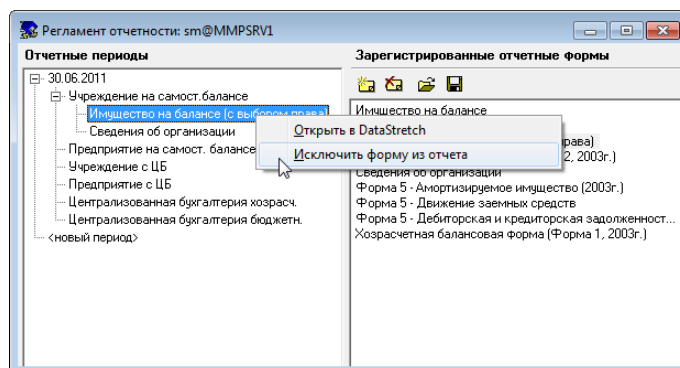


Рисунок 19.9.

Вид получаемого отчета (вид формы ввода) может быть оценен с помощью вызова программы сбора отчетных данных DataStretch (с ее помощью реальные данные вводятся в отчитывающихся организациях). Указанный вызов осуществляется из контекстного меню формы выбором пункта «Открыть в DataStretch» (рис. 19.9).

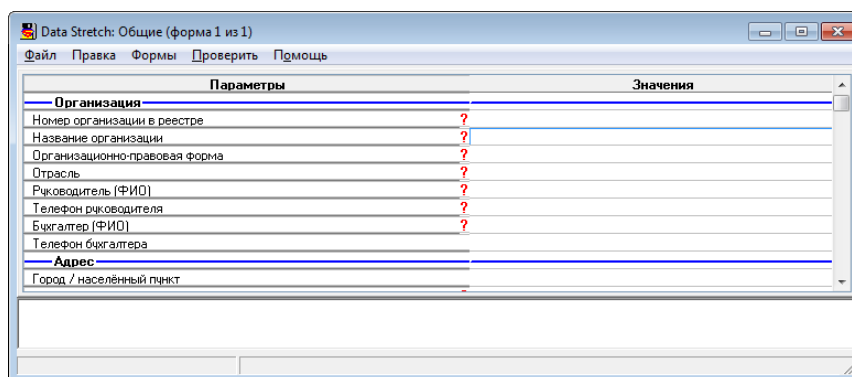


Рисунок 19.10.


## 19.4. Новые и измененные формы

Набор отчетных форм и их информационный состав определяется действующим законодательством и в идеале должен быть постоянным от одного отчетного периода к другому. Однако с течением времени законодательство меняется, что может потребовать изменений как в перечне отчетных форм, так и в их составе и в порядке заполнения.

Еще одной причиной изменения отчетных форм может быть обновление версии SAUMI, вызывающее изменения в структуре базы данных SAUMI, когда меняются реквизиты, используемые при обработке отчетных данных. Такие изменения происходят сравнительно редко и сопровождаются соответствующими пояснениями к анонсам обновлений новых версий SAUMI. В указанном случае может потребоваться изменение описания ряда используемых форм. Во всех остальных случаях потребности изменений в используемых отчетных формах менять их описание нельзя, а необходимо создать и описать новые формы.

В случае создания новой формы, заменяющей некоторую ранее использовавшуюся, при компоновке нового отчетного периода старую форму следует удалить из набора форм, а новую, соответственно, туда вставить (см. 19.3).

Помимо обновлений до новых версий одним из источников новых и измененных форм может быть Группа сопровождения SAUMI. После обновлений версий описания новых форм располагаются во внутренних разделах с именами Электронная отчетность или Отчетность архивов Reportsyyy-xxx.zip, располагаемых в папке установки SAUMI. Указанные разделы могут содержать полное описание формы в виде mro-файла (см. 19.8) или код обрабатывающих процедур на языке SAUMI-Pascal, предназначенный для замены процедур, использовавшихся ранее.

Для регистрации новой готовой формы, представленной mro-файлом, необходимо нажать на панели инструментов утилиты «Регламент отчетности» кнопку  – «Загрузить форму из файла». В результате открывается стандартное окно выбора файлов, в котором нужно указать необходимый mro-файл и нажать кнопку «Открыть».

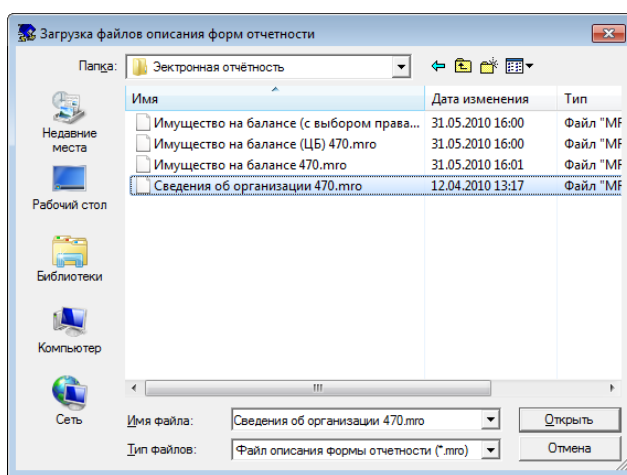


Рисунок 19.11.

В результате новая форма отображается на правой панели зарегистрированных в SAUMI форм отчетности.

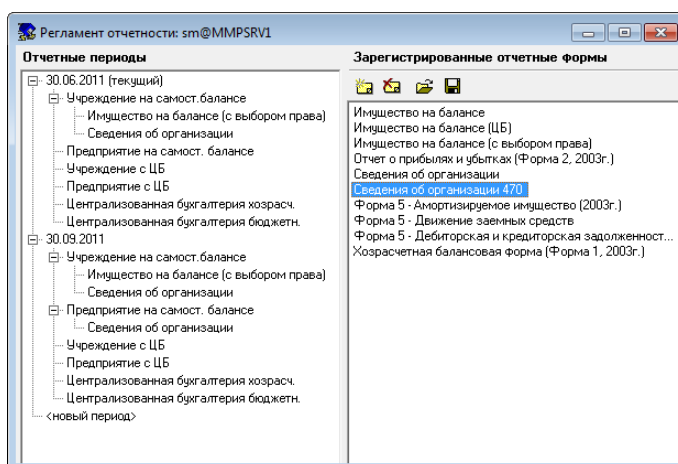


Рисунок 19.12.



Для изменения какой либо секции описания формы необходимо выделить на правой панели изменяемую форму и, нажав правую кнопку мыши вызвать контекстное меню, в котором выбрать пункт «Редактировать».

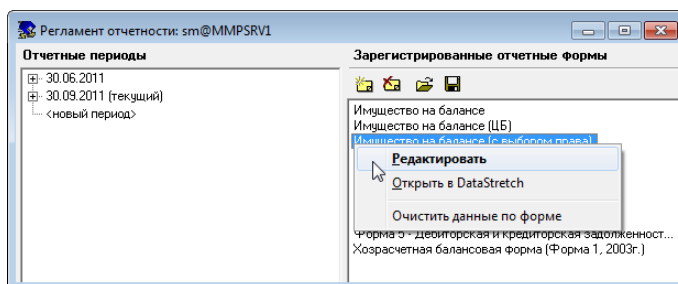


Рисунок 19.13.

В результате открывается редактор отчетных форм, в котором следует выбрать вкладку, соответствующую изменяемой секции. О соответствии секций описания формы вкладкам редактора форм (см. 19.8).

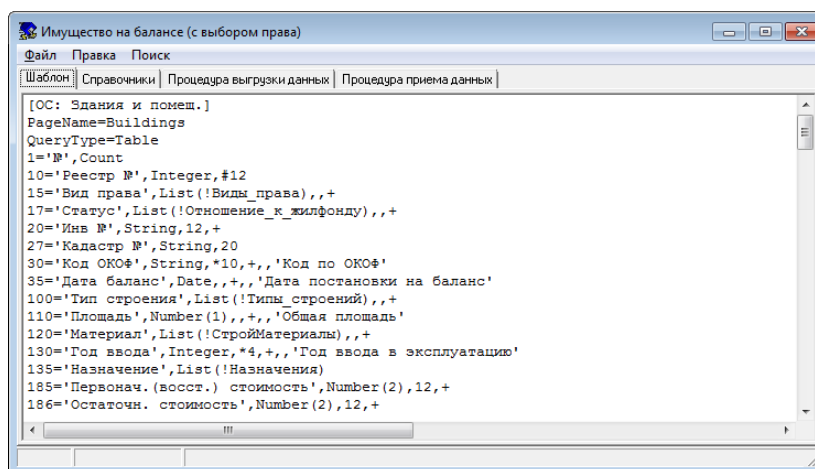


Рисунок 19.14.

Полностью заменить описание формы на другое, представленное в виде mro-файла, можно из меню «Файл» редактора форм. В этом меню нужно выбрать пункт «Открыть».

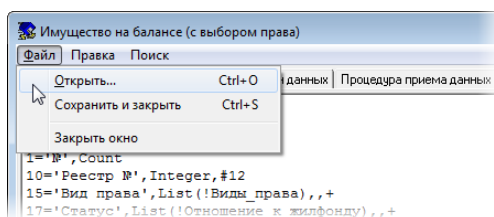


Рисунок 19.15.

в результате чего открывается окно (рис. 19.11) выбора mro-файла.

Для сохранения изменений в форме необходимо в меню «Файл» редактора выбрать пункт «Сохранить и закрыть». Сохранение изменений требует подтверждения в виде соответствующего диалогового окна.

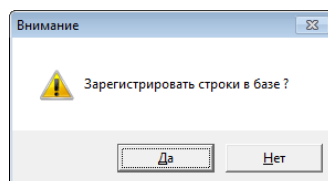



Рисунок 19.16.

## Замечание

Перед внесением любых изменений в описание формы крайне желательно сохранить имевшееся описание, чтобы иметь возможность откатить изменения и восстановить работоспособный вариант формы. Описание формы сохраняется в виде `xml`-файла. Чтобы осуществить такое сохранение необходимо выделить на правой панели сохраняемую форму и нажать кнопку .

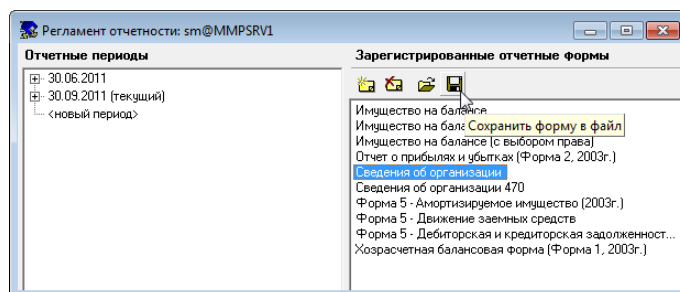


Рисунок 19.17.

В результате открывается стандартное окно сохранения файла. Имя формы можно расширить для смыслового увязывания сохраненного файла с определенным событием или иными обстоятельствами.

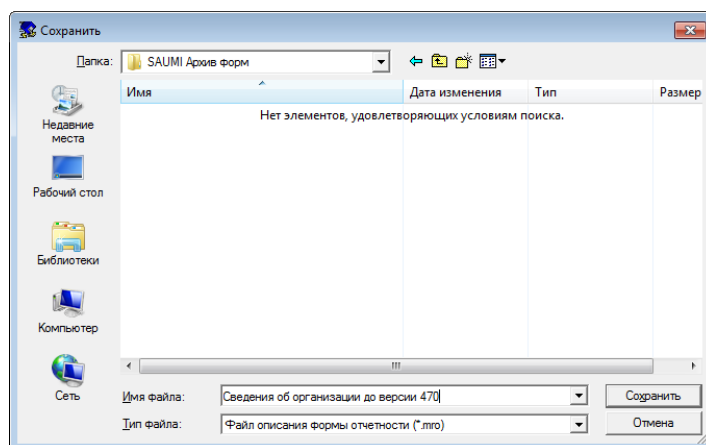


Рисунок 19.18.

## 19.5. Активирование нового отчетного периода

Активирование нового отчетного периода производится выделением на левой панели этого периода и вызова контекстного меню нажатием правой кнопки мыши. В контекстном меню следует выбрать пункт «Активировать период».

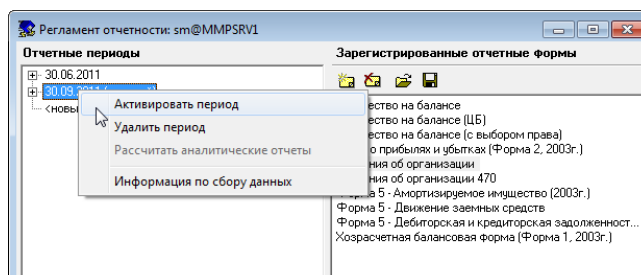


Рисунок 19.19.

При активировании периода происходит компоновка отчетных форм для каждого типа организаций в отчетном периоде.

Компоновка – формирование файлов `dataopt.ini`, `lists.ini`, стыковка алгоритмов выгрузки и загрузки данных и организация структуры хранения табличных данных для организаций каждого типа организаций.

Файл `dataopt.ini` – описание вида отчетных форм, отображаемых в программе ввода отчетных данных `DataStretch`, составляется из секций `#DATAOPT` (см. 19.8.2) описаний всех отчетных форм, сопоставленных типу юридического лица.

Файл `lists.ini` – описание и содержимое справочников, использованных в форме, формируется из справочников БД SAUMI на основе описаний всех отчетных форм, сопоставленных типу юридического лица.

Состав справочников зависит от типа юридического лица и формируется именно при активировании периода, причем набор справочников определяется из описаний всех форм, соответствующих типу юридического лица. Для исключения коллизий несоответствия разных версий форм и справочников, после активирования периода повторная компоновка отчетных форм в данном периоде становится недоступной. Также недоступна замена форм в предыдущих периодах, для сохранения соответствия принятых в те периоды отчетов отчетным формам тех же периодов.

В случае крайней необходимости исправления текущего периода, можно активировать период предыдущий и потом изменить набор форм и изменить данные в справочниках SAUMI. Однако всегда следует учитывать последствия такого шага, выражающегося в несоответствии уже переданных в отчетные организации наборов файлов отчетности набору форм и справочной информации.

## 19.6. Удаление отчетного периода

В ряде случаев возможно возникновение потребности в удалении созданного отчетного периода. Для реализации такой возможности в контекстном меню отчетного периода присутствует пункт «Удалить период» (рис. 19.19).

## Замечание

Отчетный период может быть удален, только если не было произведено ни одной приемки отчетных данных за этот период.

## 19.7. Набор файлов отчетности

Формируемый для передачи в подотчетную организацию (см. раздел «Прием отчетных данных юридических лиц» главы «Регистрация объектов и субъектов права в SAUMI» руководства пользователя) набор файлов состоит из двух частей: из файлов, полученных выборкой данных из базы данных SAUMI и из файлов, образующих так называемый макет диска. О настройке макета диска и пути сохранения формируемого набора см. 4.2.

Файлов, содержащих информацию из базы данных SAUMI, в точности три: это упоминавшиеся в предыдущем разделе файлы `dataopt.ini` и `lists.ini`, а также файл `SaveData.dat`, содержащий данные, изначально представляемые в формах ввода с помощью программы `DataStretch`. Вводимые с помощью `DataStretch` данные сохраняются в этом же файле, а обработка принимаемых отчетных данных в Комитете сводится к обработке файла `SaveData.dat` (см. 19.8.7).

## 19.8. Формат mro-файла

Выгруженное, как указано выше (см. Замечание в 19.4), описание отчетной формы представляет собой текстовый файл с расширением `.mro` и в кодировке `win1251`. Файл разбит на секции, содержание которых и определяет вид разделов формы, их состав и результат обработки принимаемых отчетных данных. Файл может редактироваться с помощью любого доступного текстового процессора (например, с помощью приложения Windows «Блокнот»).

Могут применяться следующие секции:

**#OPTIONS** – обязательная – в этой секции хранится название и тип формы;  
**#DATAOPT** – обязательная – содержит шаблон и описание форматов ввода значений формы; копируется в файл `dataopt.ini`;  
**#SAVEDATA** – используется в формах типа TABLE (см. 19.8.1) (см. Раздел 19.8.1, «Секция #OPTIONS») для формирования макета таблицы с указанием колонок ввода и колонок поясняющих надписей; копируется в файл `SaveData.dat`;  
**#EXEC\_SAVE** – содержит SAUMI-Pascal процедуру подготовки, осуществляющую подготовку начальных данных (фрагмент файла `SaveData.dat`), используемых при заполнении формы;  
**#EXEC\_LOAD** – содержит SAUMI-Pascal процедуру загрузки введенных в отчитывающейся организации данных (фрагмент вернувшегося файла `SaveData.dat`) с помощью данной формы;  
**#LISTS** – описание используемых в форме справочников.

### 19.8.1. Секция #OPTIONS

Секция содержит название и тип формы. Секция #OPTIONS является обязательной.

Формат:

```
#OPTIONS
FormName=<Название отчетной формы>
FormType=<тип отчетной формы>
```

Название отчетной формы – строка <Название отчетной формы> – отображается в списке форм в программе «Регламент отчетности». Для различения одноименных форм, имеющих отличия, в

частности, в силу изменений законодательства или по иным причинам целесообразно указывать в название первый период, в котором форма была использована.

Тип отчетной формы <тип отчетной формы> – может иметь одно из четырех значений: PROPERTY, PARAMS, TABLE, OTHER.

**PROPERTY** – форма для обработки данных об имуществе, обрабатывается с помощью программного кода на SAUMI-Pascal, содержащегося в секции #EXEC\_LOAD.

**PARAMS** – параметрическая форма отчетности с расположением одного параметра в строке, обрабатывается автоматически без использования кода SAUMI-Pascal секции #EXEC\_LOAD.

**TABLE** – параметрическая форма отчетности с несколькими параметрами, представленными колонками таблицы, обрабатывается автоматически, секция #EXEC\_LOAD не требуется.

**OTHER** – параметрическая форма с произвольным алгоритмом обработки, обрабатывается программным кодом из секции #EXEC\_LOAD.

## Замечание

Формы различаются по двум признакам, во-первых, по способу представления вводимых полей – параметрическая или табличная; во-вторых, по способу обработки получаемых из организации отчетных данных: либо встроенными в SAUMI алгоритмами обработки параметрических и табличных форм, либо посредством кода на SAUMI-Pascal, который позволяет организовать ввод и изменение записей различных таблиц БД SAUMI.

Для типа формы FormType=TABLE может указываться ключ Table\_Fields, который содержит список заголовков колонок таблицы, отделяемых один от другого символом точка с запятой «;». Например:

```
Table_Fields=На начало периода;Поступило;Выбыло;На конец периода
```

Однако, в текущей реализации программы DataStretch данный ключ ни на что не влияет, а заголовки колонок для формы типа FormType=TABLE опеределются списком параметров в секции #DATAOPT.

## 19.8.2. Секция #DATAOPT

Секция #DATAOPT описывает шаблон формы ввода для программы DataStretch. Обязательная секция.

Формат:

```
#DATAOPT
[Название раздела]
PageName=<Уникальный идентификатор раздела> – строка символов, используемая для идентификации раздела
в алгоритмах обработки.
QueryType=Parameters/Table – метод сбора данных: Parameters – ввод данных в виде "параметр-значение",
Table – ввод данных в табличном виде, где параметр – колонка таблицы.
ParamName=<Название колонки наименований параметров> (QueryType=Parameters), по умолчанию – Параметры
CodeName=<Название колонки кодов> (QueryType=Parameters & ShowCodes=yes), по умолчанию – Код
ValueName=<Название колонки значений> (QueryType=Parameters), по умолчанию – Значения
ValuesWidth=<Ширина колонки значений> (QueryType=Parameters), по умолчанию вычисляется автоматически
ShowCodes=yes/no – показывать коды параметров (QueryType=Parameters), по умолчанию – yes
CanAdd=yes/no – возможность добавления строк (QueryType=Table), по умолчанию – yes
```

Comment=Текст комментария к разделу, не выводится в форме.

<Описание параметра 1>  
 <Описание параметра 2>  
 ... ..  
 <Описание последнего параметра>

Описание одного параметра имеет следующий вид.

123='Название параметра',Тип\_данных,(#)(\*)Длина,Обязательный(+/-),Выражение\_для\_проверки,Комментарий

Здесь уникальный в описываемом разделе формы числовой номер до знака равенства (123 в приведенном выше шаблоне) является идентификатором данного параметра и может использоваться в SAUMI-Pascal процедурах обработки отчетных данных, а также входит в Выражение\_для\_проверки параметров описываемого раздела.

Тип\_данныхString – строковый тип

Integer – целочисленный тип (символы 0..9)

Number(n) – числовой тип (символы 0..9 и разделитель разрядов), n-число знаков после запятой

List(...) – выбор значения из списка

List(Пункт1|Пункт2|Пункт3|...) – используется фиксированный список значений, представленный не в секции #LISTS, а здесь же – в описании параметра

List(id1!Пункт1|id2!Пункт2|id3!Пункт3|...) – тоже, но в результат вносится значение id для соответствующего элемента списка

List(!Название\_списка) – используется список значений, определяемый отдельно в секции #LISTS(см. 19.8.6)

Фрагмент описания параметра Длина задает длину значений параметра в символах и может сопровождаться префиксом либо в виде символа #, который указывает на то, что значения описываемого параметра только для вывода в форме без возможности их изменить, либо в виде символа \*, который указывает на обязательное совпадение символьной длины значений параметра с указанным числом. Примеры: #13 – длина значений параметра 13, параметр нередактируемый; \*10 – длина значений параметра должна быть в точности 10; 15 – параметр может принимать значения до 15 символов длиной и может редактироваться.

Обязательность ввода параметра определяется фрагментом описания Обязательный, при этом, если параметр обязательно должен быть введен, то необходимо выставить в этой позиции описания символ + (плюс), в противном случае – символ - (минус).

Фрагмент описания Выражение\_для\_проверки задает арифметическое выражение из других параметров, в результате значения описываемого параметра не вводятся, а вычисляется. Параметры, участвующие в записи арифметического выражения, задаются конкатенацией префикса v и идентификационного номера параметра (см. выше). Описание параметра

050='Прибыль (убыток) от продаж (строки (010-020-030-040))',Number(2),#15,, 'V010-V020-V030-V040'

указывает, что параметр с идентификатором 050 вычисляется из параметров с идентификационными номерами 010, 020, 030, 040.

Фрагмент описания параметра Комментарий отображается в нижней панели окна программы DataStretch при переносе на него фокуса ввода. Например, ввод параметра, описанного как

30 ='Руководитель (ФИО)',String,50,+,, 'Фамилию, имя и отчество руководителя указывайте без сокращения'

сопровождается выводом соответствующей подсказки.

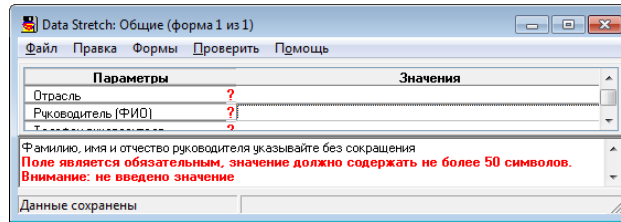


Рисунок 19.20.

Порядок параметров устанавливается очередностью их описания.

Для параметрических форм (QueryType=Parameters) могут применяться два вида горизонтальных разделителей, позволяющих визуально группировать параметры на форме: SeparatorX и LineX, где X номер разделителя в данном разделе формы. Применяется следующая нотация.

Separator1=Текст в строке разделителя Separator1  
Line1=Текст в строке разделителя Line1

Визуально SeparatorX выделяется цветом заливки фона полосы разделителя, отрисовываемой от левого до правого края окна формы, а в LineX выводимы в строке разделителя текст дополняется узкой полосой.

```
#DATAOPT

[Форма 1 - Баланс]
PageName=Balance
QueryType=Parameters
ParamName=Строка баланса
ValueName=Сумма
ShowCodes=yes

Separator1=АКТИВ
Line1=I. ВНЕОБОРОТНЫЕ АКТИВЫ
110='Нематериальные активы',Number(2),15
120='Основные средства',Number(2),15
130='Незавершенное строительство',Number(2),15
... ..
```

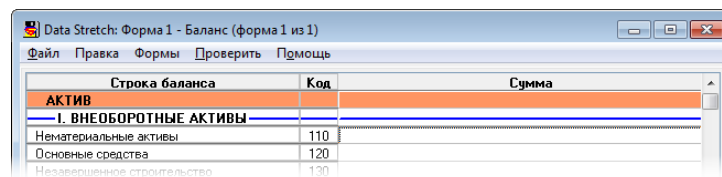


Рисунок 19.21.

Секция #DATAOPT файла описания формы (про-файл) может содержать несколько разделов, названия которых формируют меню выбора форм в программе DataStretch.

```
#OPTIONS

FormName=Имущество на балансе
FormType=PROPERTY

#DATAOPT
```

[ОС: Здания и помещ.]  
 .....  
 [ОС: Транспорт]  
 .....  
 [ОС: Незаверш.строит]  
 .....  
 [ОС: Объекты инж. инфраструктур]  
 .....  
 [ОС: Прочее имущество]  
 .....

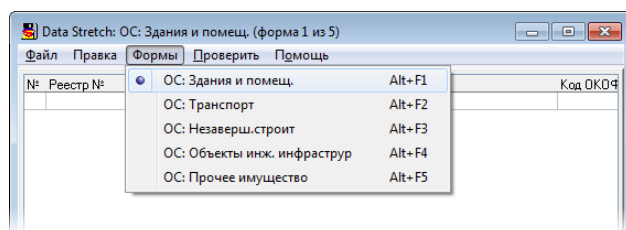


Рисунок 19.22.

Для раздела с табличным представлением параметров (QueryType=Table) названия параметров задают заголовки колонок.

```
#OPTIONS
FormName=Форма 5 - Движение заемных средств
FormType=TABLE

#DATAOPT

[Форма 5 - Движение заемных средств]
PageName=F51
QueryType=Table
CanAdd=no
CanDelete=no
10='Наименование показателя',String,#20
20='Код строки',Integer,#3
30='На начало периода',Number(2),15
40='Получено',Number(2),15
50='Погашено',Number(2),15
60='На конец периода',Number(2),15
```

Наименование показателя	Код строки	На начало периода	Получено	Погашено	На конец периода
Долгосрочные кредиты	110				
-в том числе не погашенные в срок	111				
Долгосрочные займы	120				
-в том числе не погашенные в срок	121				
Краткосрочные кредиты	130				
-в том числе не погашенные в срок	131				
Краткосрочные займы	140				
-в том числе не погашенные в срок	141				
		0	0	0	0

Рисунок 19.23.

В редакторе отчетных форм секции #DATAOPT соответствует вкладка «Шаблон».



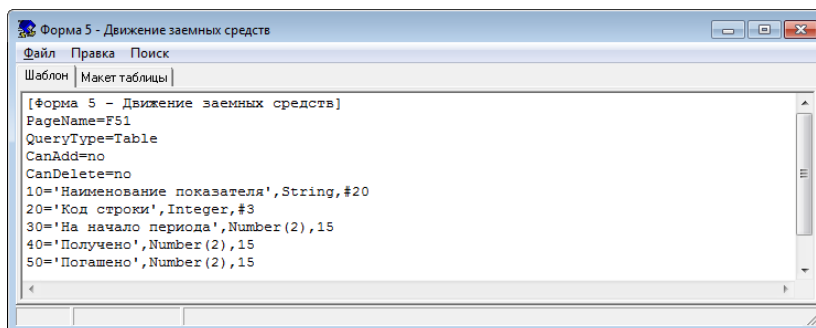


Рисунок 19.24.

## Замечание

Имеются количественные ограничения при описании разделов. Количество описанных табличных параметров не должно превышать 100, а максимальный номер табличного параметра 1500. Максимальный номер параметра в параметрическом разделе 15000.

### 19.8.3. Секция #SAVEDATA

В секции могут содержаться начальные значения параметров, описанных в разделах секции #DATAOPT. Применяется в основном для FormType=TABLE и PARAMS. Необязательная секция.

Формат для раздела с QueryType=Parameters:

```
#SAVEDATA
[Название раздела]
Номер_параметра1=Значение1
Номер_параметра2=Значение2
... ..
```

где Номер\_параметраX – идентификационный номер параметра, заданный при его описании в секции #DATAOPT.

Формат для раздела с QueryType=Table:

```
#SAVEDATA
[Название раздела]
1="Значение11", "Значение12", ..., "Значение1X"
2="Значение21", "Значение22", ..., "Значение2X"
... ..
```

где X – точное количество параметров, описанных в одноименном разделе секции #DATAOPT. В отличие от параметрического раздела начальные значения параметров табличного раздела заключаются в кавычки. Отсутствие начального значения параметра в строке таблицы задается парой кавычек, следующих подряд "" (пустая строка).

Представленная на рис. 19.23 форма «Форма 5 - Движение заемных средств» определяется следующими начальными данными секции #SAVEDATA.

```
#SAVEDATA
[Форма 5 - Движение заемных средств]

1="Долгосрочные кредиты";"110";"";"";"";""
2=" -в том числе не погашенные в срок";"111";"";"";"";""
3="Долгосрочные займы";"120";"";"";"";""
```

```
4=" -в том числе не погашенные в срок";"121";"";"";"";""
5="Краткосрочные кредиты";"130";"";"";"";""
6=" -в том числе не погашенные в срок";"131";"";"";"";""
7="Краткосрочные займы";"140";"";"";"";""
8=" -в том числе не погашенные в срок";"141";"";"";"";""
```

Содержимое секций #SAVEDATA из всех форм набора, соответствующего типу организации, копируется в файл SaveData.dat

В редакторе отчетных форм для FormType=TABLE секции #SAVEDATA соответствует вкладка «Макет таблицы».

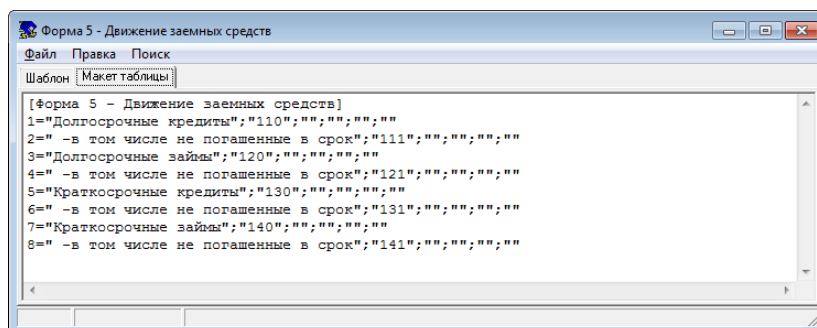


Рисунок 19.25.

Для FormType=PARAMS секция #SAVEDATA может быть отредактирована в предварительно выгруженном mro-файле, после чего в режиме редактирования измененный файл нужно снова загрузить.

#### 19.8.4. Секция #EXEC\_SAVE

Помимо начальных данных, статически описанных в секции #SAVEDATA, для FormType=PROPERTY и FormType=OTHER применяется выгрузка начальных данных, соответствующих конкретному юридическому лицу, в файл SaveData.dat. Такая выгрузка осуществляется с помощью SAUMI-Pascal процедуры, текст которой размещается в секции #EXEC\_SAVE. Описываемая секции не является обязательной, даже для FormType=PROPERTY и FormType=OTHER, т.е. начальная информация для этих видов форм может не предоставляться в отчитывающуюся организацию.

В режиме редактирования секции #EXEC\_SAVE соответствует вкладка «Процедура выгрузки данных».

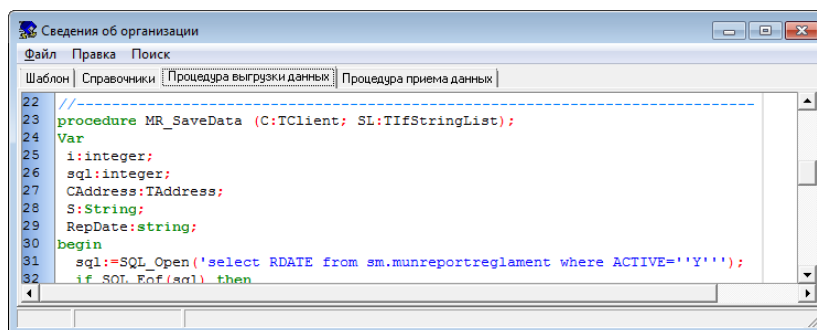


Рисунок 19.26.

Код SAUMI-Pascal начальной выгрузки данных должен содержать процедуру с именем MR\_SaveData и со следующей сигнатурой.

```
procedure MR_SaveData (C:TClient; SL:TifStringList);
```

Процедура MR\_SaveData может вызывать любые встроенные функции и процедуры SAUMI-Pascal, а также другие процедуры и функции на языке SAUMI-Pascal, размещаемые, как в секции #EXEC\_SAVE, так и в поставляемом с SAUMI специальном модуле, называемом sp\_global.pas, доступном для редактирования специальной утилитой комплекта администратора SAUMI – «Редактор SAUMI-Pascal»(см. главу 13).

Секция имеет следующий формат.

```
#EXEC_SAVE
... ..
процедуры и функции, вызываемые из процедуры MR_SaveData
... ..
procedure MR_SaveData (C:TClient; SL:TifStringList);
begin
    код реализации MR_SaveData
end;
```

При запуске процедуры MR\_SaveData в качестве параметра передается структура описания юридического лица, для которого требуется подготовить данные, – параметр с. Результат необходимо формировать в списке строк SL.

Список строк формируется по разделам, объявленным в секции #DATAOPT. Самой первой строкой фрагмента списка, соответствующего некоторому разделу, является заголовок этого раздела, включающий квадратные скобки []. Остальные строки должны соответствовать заданным в секции #DATAOPT описаниям параметров.

Сформированный список SL после завершения работы процедуры MR\_SaveData поэлементно копируется в строки файла SaveData.dat, образуя соответствующий фрагмент начальных данных.

При работе процедуры, как правило, используются SQL-вызовы для запроса текущей информации применимой к данному юридическому лицу, из которой формируется текстовая строка в соответствии с заданными описаниями параметров и видом раздела (QueryType=Parameters или QueryType=Table). Формат создаваемых строк тот же, что и в секции #SAVEDATA (см. 19.8.3).

Для лучшего понимания применимых при программировании процедуры MR\_SaveData техник и сопоставления с получаемыми результатами рекомендуется изучить соответствующий код в отчетных формах, имеющихся в поставке SAUMI, а также и рассмотреть получаемый при этом файл SaveData.dat.

### 19.8.5. Секция #EXEC\_LOAD

Обратной операцией по отношению к выгрузке начальных данных для FormType=PROPERTY или FormType=OTHER является загрузка отчетных данных, полученных из отчитывающейся организации в файле SaveData.dat. Указанная загрузка осуществляется с помощью SAUMI-Pascal процедуры, текст которой размещается в секции #EXEC\_LOAD. Описываемая секция является обязательной для FormType=PROPERTY и FormType=OTHER.

В режиме редактирования секции #EXEC\_LOAD соответствует вкладка «Процедура приема данных».

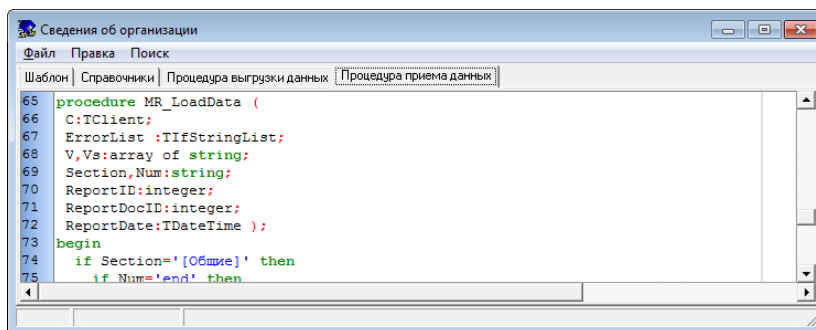


Рисунок 19.27.

Код SAUMI-Pascal начальной выгрузки данных должен содержать процедуру с именем MR\_LoadData и со следующей сигнатурой.

```
procedure MR_LoadData (C:TClient; ErrorList :TIfStringList;
  V,Vs:array of string; Section,Num:string; ReportID:integer;
  ReportDocID:integer; ReportDate:TDateTime );
```

Процедура MR\_LoadData может вызывать любые встроенные функции и процедуры SAUMI-Pascal, а также другие процедуры и функции на языке SAUMI-Pascal, размещаемые, как в секции #EXEC\_LOAD, так и в поставляемом с SAUMI специальном модуле, называемом sp\_global.pas, доступном для редактирования специальной утилитой комплекта администратора SAUMI – «Редактор SAUMI-Pascal»(см. главу 13).

Секция имеет следующий формат.

```
#EXEC_LOAD
... ..
процедуры и функции, вызываемые из процедуры MR_LoadData
... ..
procedure MR_LoadData (C:TClient; ErrorList :TIfStringList;
  V,Vs:array of string; Section,Num:string; ReportID:integer;
  ReportDocID:integer; ReportDate:TDateTime );
begin
  код реализации MR_LoadData
end;
```

Процедура запускается многократно для обработки каждой строки отчета. В качестве параметров передаются:

C – структура описания юридического лица, у которого принимается отчет;

ErrorList – список строк, куда можно записывать информацию о логических ошибках обработки отчета;

V – массив строк формы; применяется для форм типа OTHER;

Vs – массив значений текущей строки; применяется для форм типа PROPERTY;

Section – название раздела;

Num – номер, стоящий в начале строки до знака = (равно), или значение start перед первой строкой раздела, или значение end после последней строки раздела.

ReportID – идентификационный номер, присваиваемый данной приемке отчета в данном отчетном периоде от данной организации; фиксируется в таблице MUNREPORTS;

ReportDocID – регистрационный номер документа для отчетного периода;

ReportDate – дата отчетного периода.

Процедура может добавлять новые записи в таблицах базы данных SAUMI и вносить любые изменения значений в имеющихся записях. Загрузка данных всего отчета (всех разделов файла DataSave.dat) выполняется в пределах одной транзакции. При наличии хотя бы одной строки в списке ErrorList считается, что отчет был обработан с ошибками и транзакция откатывается. Если же строк в списке ErrorList нет, то отчет считается успешно обработанным и транзакция завершается нормально.

Список ErrorList формируется пустым перед приемом отчета и передается как параметр сначала обработчику события начала приема отчетных данных – процедуре OnGetMunReportBegin, затем последовательно с каждым разделом отчета его процедуре обработки и в заключение обработчику события окончания приема отчетных данных – процедуре OnGetMunReportEnd. Об обработчиках событий в SAUMI см. раздел «Расширение возможностей комплекса определением обработчиков событий» в описании языка SAUMI-Pascal, представленного в виде html страниц на сайте <http://support.saumi.ru>, а также на дистрибутивном диске SAUMI. Каждый следующий обработчик разделов дописывает в список ErrorList свои сообщения, и если итоговый список оказывается непустым, то он выводится в соответствующем окне (см. окончание раздела «Прием отчетных данных юридических лиц» главы «Регистрация объектов и субъектов права в SAUMI» руководства пользователя).

Массивы строчного типа V,Vs передают значения параметров, выделенные из строк обрабатываемого раздела для QueryType=Parameters (массив V) и для QueryType=Table (массив Vs). При этом в массиве V для QueryType=Parameters определены элементы, содержащие значения параметров, расположенных в обрабатываемом разделе от его начала до текущей строки, на которой была вызвана процедура MR\_LoadData, а для QueryType=Table массив Vs содержит значения параметров, заданных в текущей строке обрабатываемого раздела. Для нумерации элементов указанных массивов используются идентификационные номера соответствующих параметров, описанных для обрабатываемого раздела в секции #DATAOPT (см. 19.8.2) описания формы.

Для раздела формы (FormType=OTHER) с описанием

```
#DATAOPT

[Общие]
PageName=Common
QueryType=Parameters

Line1=Организация
5 = 'Номер организации в реестре', Integer, #10, +
10 = 'Название организации', String, 50, +
20 = 'Организационно-правовая форма', List (!ОргФормы) , , +
22 = 'Отрасль', List (!Отрасли) , , +
30 = 'Руководитель (ФИО)', String, 50, +
40 = 'Телефон руководителя', String, 12, +
... ..
```

при обработке следующего фрагмента файла DataSave.dat

```
[Общие]
5=10480
10=Дворец культуры
20=МУП
22=Культура
30=Донкихотов Александр Апанасович
40=33-44-55
```

... ..

во время вызова процедуры MR\_LoadData на строке 22=Культура определены значения элементов V[5], V[10], V[20], V[22], тогда как значения всех остальных элементов массива v не определены. Чтобы иметь возможность оперировать значениями всех параметров раздела в коде процедуры MR\_LoadData, достаточно обрабатывающий код вставить внутрь условного оператора

```
if Section='[Общие]' then
  if Num='end' then
    begin
      ... код обработки значений параметров
    end;
```

## Замечание

Проверка имени секции (if Section='[Общие]' в примере выше) всегда необходима для форм с несколькими разделами и в любом случае желательна для акцентирования назначения кода.

Для лучшего понимания применимых при программировании процедуры MR\_LoadData техник рекомендуется изучить соответствующий код в отчетных формах, имеющихся в поставке SAUMI.

## 19.8.6. Секция #LISTS

В секции #LISTS задаются справочники, используемые при вводе значений параметров посредством выбора из списков в различных разделах формы. Секция может отсутствовать, если ни один параметр в секции #DATAOPT не имеет тип данных List(...).

Секция имеет следующий формат.

```
#LISTS
<описание справочника 1>
<описание справочника 2>
... ..
<описание последнего справочника>
```

Описание одного справочника задается следующим образом

```
[НазваниеСправочника]
CanAdd=no/yes - по умолчанию - no
Sorted=yes/no - по умолчанию - yes
Source=ИмяТаблицы1 (ПолеТаблицы)
```

Ключ CanAdd=yes позволяет вводить новые элементы справочников с помощью программы DataStretch, однако элементы справочников могут быть зарегистрированы в SAUMI автоматически только с помощью кода на SAUMI-Pascal, вызываемого с помощью процедуры MR\_LoadData (см. 19.8.5).

Ключ Sorted=no позволяет отказаться от сортировки справочника по значению поля таблицы, указанного в ключе Source.

Ключ Source=ИмяТаблицы1 (ПолеТаблицы) указывает справочную таблицу SAUMI и поле в этой таблице, значения которого образуют список справочника. Обычно используется поле NAME.

В режиме редактирования секции #LISTS соответствует вкладка «Справочники».

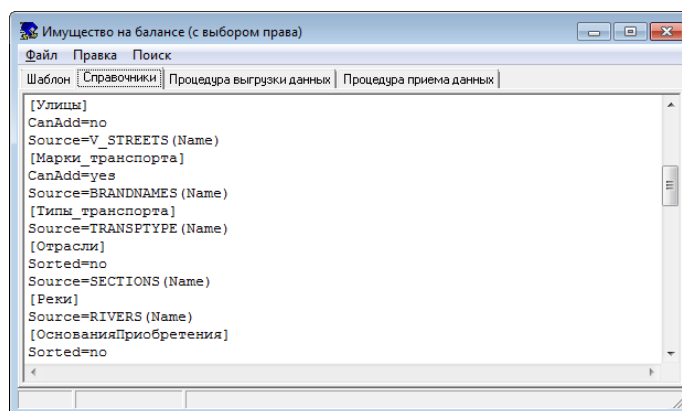


Рисунок 19.28.

### 19.8.7. Порядок обработки принимаемых отчетных данных

Из отчитывающейся организации в Комитет возвращается набор данных, продублированный в двух файлах: 1234567.dat и 1234567.~da, где 1234567 – идентификационный номер юридического лица (значение поля ID в таблице CLIENTS). Оба файла получены в результате дописывания в начало текстового файла SaveData.dat некоторой предупреждающей и идентификационной информации с последующим сжатием в архивный файл. Описание операции приема отчетных данных представлено в последнем подразделе раздела «Прием отчетных данных юридических лиц» главы «Регистрация объектов и субъектов права в SAUMI» руководства пользователя).

При считывании отчетных данных происходит проверка соответствия принимаемого набора данных юридическому лицу, а также активированному отчетному периоду. В случае их соответствия определяется идентификатор производимого приема (передается в частности под именем ReportID в процедуру приема данных MR\_LoadData) и производится последовательная обработка строк, содержащихся в сформированном в отчетной организации файле SaveData.dat.

Файл SaveData.dat состоит из разделов, каждый из которых начинается с названия раздела в квадратных скобках []. Пример состава различных разделов в файле SaveData.dat.

```
[Общие]
5=11451
10=МУП"ДГХ"
20=МУП
22=Жилищно-коммунальное хозяйство
... ..
[Форма 1 - Баланс]
110=0
111=
112=
113=
120=2818
121=
122=2818
... ..
[Форма 2 - Отчет о прибылях и убытках]
010=1693901
020=834219
029=859682
030=1139431
040=
... ..
[OC: Здания и помещ.]
```

```
1="1";"";"10001";"13 0002000";"25.08.1997";"ЖИЛОЙ ДОМ";"644";"ж\б панели";"1978";"Жилой дом";"925347...
2="2";"";"10189";"11 4522341";"30.10.2000";"нежилое строение, склад";"360";"ж\б панели";"2000";"прои...
... ..
[ОС: Транспорт]
1="1";"";"01510";"15 3410112";"25.03.2006";"Автомобиль легковой";"ВАЗ";"211110";"2005";"з555кз98";"2...
2="2";"";"10944";"15 3410195";"30.09.2009";"Автомобиль грузовой";"КАМАЗ";"551111";"1997";"к468ам";"55...
... ..
[ОС: Прочее имущество]
1="1";"";"1";"16 3612050";"30.09.2008";"Прочие основные ср-ва";"Мебель";"кресла";"Кресло офисное";"1...
2="2";"";"2";"16 3612050";"30.09.2008";"Прочие основные ср-ва";"Мебель";"Комплект офисной мебели";"0...
3="3";"";"16";"14 3020040";"01.10.2008";"Прочие основные ср-ва";"вычислительная техника";"Концентрат...
... ..
```

При последовательной обработке строк файла по распознанному названию определяется завершение предыдущего раздела с выполнением завершающих действий, определяется форма, в секции #DATAOPT (см. 19.8.2) которой описан данный раздел, извлекается из БД SAUMI описание параметров и обрабатывающих процедур (см. 19.8.5).

Производится последовательная обработка каждой строки раздела с выделением номера из строки, который для параметрических разделов является идентификатором соответствующего параметра, описанного в секции #DATAOPT (см. 19.8.2).

Выделяется значение определяемого номером параметра для параметрических разделов и значения всех параметров в табличных разделах; значения представлены в виде строк, не включающих кавычки и разделители, применяемых в табличных разделах.

Для разделов, описанных в формах с `FormType=PARAMS` или `FormType=TABLE`, значения параметров записываются в таблицу `MUNREPORTLINES`; для разделов, описанных в формах с `FormType=OTHER` или `FormType=PROPERTY` вызывается SAUMI-Pascal процедура `MR_LoadData`, описанная в соответствующей разделу форме (см. 19.8.5).

Содержимое полученного из организации файла `SaveData.dat` сохраняется в виде большого бинарного объекта (поле `SAVEDATA_FILE`) в таблице `MUNREPORTS`.



---

# Глава 20. Настройка расчета доходов будущих периодов

Отображаемые печатными формами «Ведомость начислений доходов будущих периодов (субъекты)», «Ведомость начислений доходов будущих периодов», «Оборотно-сальдовая ведомость доходов будущих периодов» сведения об ожидаемых доходах (см. главу «Доходы будущих периодов» руководства пользователя) зависят от ряда параметров и иных настроек.

При расчете ожидаемого дохода важными факторами являются срок действия договора и условия по обязательству.

Срок действия договора определяется двумя способами:

- через PL/SQL процедуру `IC_MovesetDocument`. Эта процедура создана для подстройки администратором SAUMI алгоритма определения основного документа операции движения, а так же даты начала и окончания действия договора. По умолчанию производятся null-значения, но процедура содержит закомментированный код, определяющий основной документ посредством вызова функции `sm.getmovesetmaindocid`, а срок действия договора по минимальной и максимальной дате периодов в операции движения;
- если вышеописанная процедура производит null-значения, срок действия договора определяется по основному документу операции движения. Если дата окончания не задана, – в этом случае договор считается бессрочным.

Ожидаемый доход засчитывается по следующему алгоритму: на начало первого периода ОД с условием по обязательству рассчитывается сумма ожидаемого дохода с момента начала периода и до окончания действия договора. При этом выделяются некоторые особенности в расчетах.

*Новые периоды.* Если до окончания договора создается новый период, в этом случае происходит списание оставшейся суммы ожидаемого дохода (сумма по кредиту) и начисление дохода по новым условиям до конца действия договора (сумма по кредиту). По умолчанию, пакет `INCOME_CHARGE_PKG` выдает уже итоговую сумму-дельту. Эта сумма может отражаться по дебету (если изменение привело к увеличению ожидаемого дохода), либо по кредиту (если доход уменьшился).

*Автопродлонгация.* Если срок действия договора истек, а операция движения не закрыта, либо закрыта более поздней датой, то тогда поведение механизма расчета ожидаемого дохода определяет значение параметра `IC_DOC_AUTO_PROLONGATION`:

- значение 0 (нуль) – значение по умолчанию – расчет прекращается;
- значение 1 (единица) – происходит автоматическое начисление ожидаемого дохода по действующим на момент окончания условиям договора на тот же срок;
- значение 2 (два) – происходит периодическое увеличение ожидаемого дохода на сумму периодических начислений;
- значение 3 (три) – происходит продлонгация интервала расчета дохода на количество месяцев, указанных значением параметра `IC_CONTRACT_MONTHS_DEF`.

*Досрочное расторжение.* Если последний период в ОД заканчивается раньше, чем срок действия договора, то остаток ожидаемого дохода списывается.

*Бессрочные договоры.* Расчет дохода по бессрочным договорам зависит от значения параметра `IC_CALC_BY_UNLIM_DOC`. Значение по умолчанию 'N' отменяет такой расчет. При значении 'Y' расчет выполняется и доход определяется исходя из количества месяцев, заданных параметром `IC_CONTRACT_MONTHS_DEF`. При этом с каждым новым платежным периодом срок договора автоматически продлевается до указанного количества месяцев – в результате вместе с автоматическим уменьшением суммы ожидаемого дохода (кредит) происходит такое же автоматическое ее увеличение на ту же самую сумму (дебет). Таким образом, конечная сумма ожидаемого дохода по бессрочным договорам никогда не изменяется.

## Замечание

Требуемые значения параметров задаются с помощью утилиты из комплекта администратора «Диспетчер настроек».

Также существует возможность отключать расчет ожидаемого дохода по типам обязательств (элементы справочника «Типы обязательств» – таблица `OBLIGATIONTYPE`). Для этого реквизиту «Рассчитывать ожидаемый доход» (поле `SM.OBLIGATIONTYPE.ic_calc`) указанного справочника необходимо установить нужное значение ('Y' – расчет ведется, 'N' – расчет не ведется). По умолчанию, расчет ведется для всех обязательств кроме ПИР (технические обязательства, сопровождающие претензионно-исковую работу – см. главу «Поддержка претензионно-исковой работы в SAUMI») руководства пользователя.

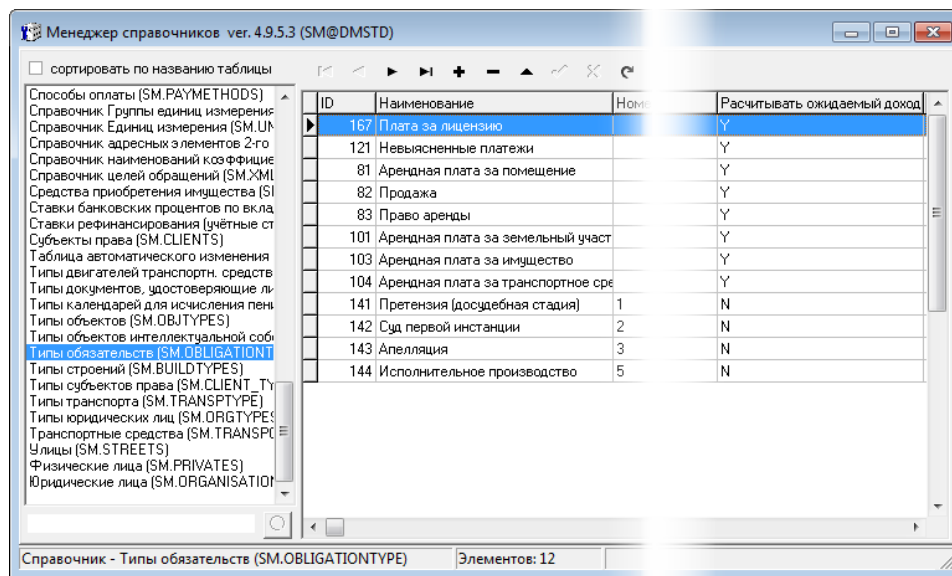


Рисунок 20.1.

К увеличению (дебету) доходов будущих периодов относятся:

- сумма ожидаемого дохода на момент заключения договора,
- автопродлонгация учета дохода будущих периодов (как по истечении договора, так и периодическая),
- увеличение ожидаемого дохода в результате изменений условий по обязательству.

К уменьшению (кредиту) доходов будущих периодов относятся:

- сумма периодических начислений, как произведенных, так и предполагаемых в расчетный период в соответствии с условиями договора;

- списание дохода в результате досрочного расторжения договора;
- уменьшение ожидаемого дохода в результате изменений условий по обязательству.

---

# Глава 21. Дополнения (плагины) – общие сведения

Под **дополнением (плагином)** в SAUMI подразумевается автономная программа (исполняемый файл), которую можно запускать из Главной панели SAUMI из меню «Дополнения». В качестве дополнения (плагины) может выступать любая программа, если для её работы не требуется наличие каких-либо дополнительных файлов (кроме системных библиотек или файлов, которые Windows сможет найти по путям, прописанным в переменных среды). Если для работы программы в качестве дополнения (плагины) требуются дополнительные файлы - они должны находиться в каталоге запуска главного модуля SAUMI – `mmpr32.exe` – на рабочей станции.

Дополнения (плагины) можно объединять в группы.

Если исполняемый файл, используемый в качестве дополнения (плагины), должен подключаться к БД SAUMI, используя параметры подключения основного модуля SAUMI, можно избежать необходимости ввода имени пользователя, пароля и названия базы данных. При запуске в качестве дополнения (плагины), этой программе передаются значения для трех дополнительных параметров запуска:

**/USER**=<имя пользователя, запустившего основной модуль SAUMI>  
**/BASE**=<название базы данных, к которой подключен основной модуль SAUMI>  
**/PASS**=<пароль, введённый пользователем при подключении основного модуля SAUMI к базе данных>

Используя переданные значения, программа-плагин может подключиться к текущей базе данных напрямую, не требуя повторного ввода параметров подключения от пользователя.

Для того, чтобы зарегистрировать программу в качестве плагина, необходимо выполнить ряд шагов.

Если плагин должен входить в какую-то группу, необходимо создать эту группу, если она ещё не создана.

Для создания группы нужно вставить соответствующую запись в таблицу **SM.PLUGIN\_GROUPS**. Эта таблица является справочником и доступна для редактирования из «Менеджера справочников» комплекта администратора SAUMI. Другой способ – прямая вставка соответствующей записи в таблицу **SM.PLUGIN\_GROUPS**:

**ID** – использовать секвенс **SM.SEQPLUGIN\_GROUPS**;  
**NAME** – название группы, отображаемое в меню «Дополнения»;  
**ORD** – порядок следования группы в меню «Дополнения», если null - будет помещаться в конец списка групп в алфавитном порядке по полю **NAME**.

Вставить запись, соответствующую программе, которая должна использоваться как плагин, в таблицу **SM.VER\_FILES** - название файла (**FILENAME**), дата файла (**FILEDATE**), импортировать файл программы в поле **FILEBODY**.

Вставить запись в таблицу **SM.PLUGINS**:

**ID** – использовать секвенс **SM.SEQPLUGINS**;  
**MENU\_CAPTION** – наименование пункта в меню «Действия» Главной панели SAUMI;  
**ORD** – порядок следования дополнения в своей группе в меню «Дополнения»;  
**FILE\_NAME** – название исполняемого файла (БЕЗ РАСШИРЕНИЯ!);  
**FILE\_EXTENTION** – расширение исполняемого файла;

**EXECUTE\_PARAMS** – параметры командной строки, используемой для запуска плагина;  
**EXECUTE\_ROLE** – роль, определяющая возможность запуска плагина из меню «Дополнения» Главной панели;  
**PLUGIN\_GROUPS\_ID** – ссылка на группу плагинов (если данный плагин не входит ни в какую группу - **null** и плагин будет отображаться в меню «Дополнения» после списка групп, в порядке, заданном значением поля **ORD** или (если **ORD=null**) в алфавитном порядке по полю **MENU\_CAPTION**);  
**NEED\_LOGIN** – должен ли основной модуль SAUMI передавать плагину параметры подключения к БД. Значения: 'Y' - передать, 'N' или **null** – не передавать.

После запуска основного модуля SAUMI - mmp32.exe, в Главной панели появится пункт «Дополнения», раскрыв который, можно запустить необходимый плагин на выполнение. Перед запуском плагина, соответствующий ему файл извлекается из таблицы **SM.VER\_FILES**, помещается в каталог, из которого вызывался основной модуль SAUMI и запускается на выполнение.

### Пример 21.1.

*Зарегистрируем в качестве плагина стандартную программу Windows «Калькулятор» (calc.exe). Этот плагин разместим в группе плагинов «Стандартные». Мы хотим, чтобы группа "Стандартные" была первой по порядку в меню «Дополнения».*

1. Создаём группу «Стандартные» в меню «Дополнения» Главной панели SAUMI:

```
insert into sm.plugin_groups (id, name, ord) values (sm.seqplugin_groups.nextval, 'Стандартные', 1);
commit;
```

2. Вставляем запись для программы calc.exe в таблицу **VER\_FILES**:

```
insert into sm.ver_files (filename, filedate) values ('calc.exe','01.01.2007');
commit;
```

Импортируем исполнимый файл программы "Калькулятор" - calc.exe (обычно располагается в папке System32 в каталоге установки Windows) в поле **FILEBODY** таблицы **VER\_FILES** для строки, где **FILENAME='calc.exe'** (например, с помощью программы PL/SQL Developer)

3. Вносим запись в таблицу **PLUGINS**:

```
insert into sm.plugins (id, menu_caption, file_name, file_extention, execute_params,
                      execute_role, plugin_groups_id)
values (sm.seqplugins.nextval, 'Калькулятор', 'calc', 'exe', null, 'SM_ALLUSERS',
       sm.seqplugin_groups.currval);
commit;
```

Если всё сделано правильно, при запуске SAUMI появится меню «Дополнения», в нём - группа «Стандартные», в которой будет пункт «Калькулятор». При выборе этого пункта будет запущен стандартный калькулятор Windows.

### Пример 21.2.

*Зарегистрируем в качестве плагина программу, разработанную администратором SAUMI, которая должна подключаться к БД SAUMI для осуществления каких-то своих функций, причём при запуске следует указать следующий ключ - '/show=N'. Пусть эта программа называется MyProg.exe. Этот плагин разместим вне какой-либо группы плагинов, непосредственно в меню «Дополнения».*

1. Вставляем запись для программы MyProg.exe в таблицу **VER\_FILES**:

```
insert into sm.ver_files (filename, filedate) values ('MyProg.exe','02.02.2007');
```

```
commit;
```

Импортируем исполнимый файл программы `MyProg.exe` в поле **FILEBODY** таблицы **VER\_FILES** для строки, где **FILENAME**='MyProg.exe' (например, с помощью программы PL/SQL Developer)

## 2. Вносим запись в таблицу **PLUGINS**:

```
insert into sm.plugins
      (id,menu_caption,file_name,file_extention,execute_params,execute_role,need_login)
values
      (sm.seqplugins.nextval,'Моя программа','MyProg','exe','/show=N','SM_ALLUSERS','Y');
commit;
```

Если всё сделано правильно, при запуске SAUMI появится меню «Дополнения», в котором будет пункт «Моя программа». При выборе этого пункта будет запущена программа `MyProg.exe` с ключом `/show=N` и будут добавлены следующие параметры: `/USER=<имя пользователя, запустившего основной модуль SAUMI>` `/PASS=<пароль пользователя>` `/BASE=<название БД, к которой произведено подключение основного модуля SAUMI>`.

## Замечание

После запуска любого дополнения (плагина) из Главной панели SAUMI, соответствующий файл извлекается из БД и помещается в каталог запуска SAUMI. Это полноценная программа (exe-файл) и может быть запущена в дальнейшем самостоятельно, вне зависимости от того, запущен главный исполняемый модуль SAUMI (`mmp32.exe`) или нет. Возможность запуска этого модуля из главной панели SAUMI останется в любом случае и в любой момент времени.

---

# Глава 22. Модуль выгрузки данных в соответствии с приказом Минфина РФ No 47н

## Содержание

22.1. Необходимые элементы и структуры данных .....	260
22.2. Необходимое информационное обеспечение .....	260
22.3. Работа плагина .....	262

Описываемый модуль (плагин) осуществляет выгрузку данных о земельных участках в соответствии с Приказом Министерства финансов Российской Федерации от 23.03.2006 No 47н «Об утверждении формы „Сведения о земельных участках, расположенных в пределах муниципального образования“ и Рекомендаций по заполнению» и Приказом Федеральной Налоговой Службы от 28 июля 2006 г. No САЭ-3-13/485@ «Об утверждении требований к структуре и содержанию файлов сведений о земельных участках, расположенных в пределах муниципального образования».

Модуль представляет собой exe-файл (GroundsExport47H.exe), который зарегистрирован в SAUMI как дополнение (плагин) (см. 21). Это означает, что его можно вызвать из Главной панели SAUMI из меню «Дополнения». В результате выполнения GroundsExport47H.exe будут сгенерированы два файла: XML-файл, содержащий собственно информацию о земельных участках и их правообладателях и XSD-файл, содержащий информацию о структуре выше названного XML-файла. В налоговую инспекцию подлежат передаче оба эти файла.

Если данные в электронном виде передаются без использования ЭЦП (электронной цифровой подписи), то, согласно приказу, необходимо передачу оформлять актом приёма-передачи. Акт составляется в двух экземплярах. Дату и номер этого акта, а также дату его подписания необходимо будет занести в соответствующие поля ввода в модуле GroundsExport47H.exe (рис. 22.3, секция «Сопроводительный документ»). Отсюда следует, что перед запуском модуля, формирующего отчётные данные, необходимо иметь на руках этот акт, уже подписанный отчитывающейся стороной.

Отчитывающейся стороной является уполномоченный распорядитель базового собственника – соответствующий Комитет. Отчет формируется для текущего базового собственника. Чтобы сформировать отчет другого административно-территориального образования, нужно надлежащим образом сменить базового собственника (см. соответствующий раздел в главе «Начало работы» руководства пользователя).

## Замечание

В отчет собираются сведения о тех земельных участках, у которых имеется кадастровый номер, которые имеют код ОКТМО, совпадающий с кодом ОКТМО базового собственника, и на которые оформлена одна из операций движения: «Собственность», или «Бессрочное пользование», или «Наследуемое владение».

На основании пункта 12 статьи 396 Налогового кодекса Российской Федерации органы муниципальных образований ежегодно до 1 февраля года, являющегося налоговым периодом, обязаны сообщать в налоговые органы по месту своего нахождения сведения о земельных участках,

признаваемых объектом налогообложения в соответствии со статьей 389 Кодекса, по состоянию на 1 января года, являющегося налоговым периодом.

## 22.1. Необходимые элементы и структуры данных

Для того чтобы формировать сведения о земельных участках в пределах муниципального образования в электронном виде так, как это предусмотрено Приказами, в структуре БД SAUMI определены следующие элементы данных и связи между ними:

- справочник «Налоговые органы» (**RATING\_AUTHORITY**);
- поле «Код ОКТМО» (**OKATO**) в таблице **ORGANISATIONS**;
- записи в таблице **ORGANISATIONS** привязаны к справочнику **RATING\_AUTHORITY** по полю **RATING\_AUTHORITY\_ID**;
- поля «Код ОКТМО» (**OKATO**) и «Кадастровый номер» (**KADASTRNO**) в таблице **GROUND**s;
- справочник «Категория земель» (**GROUNDTYPES**) соответствует приводимому в Приложении 1 к Рекомендациям по заполнению формы «Сведения о земельных участках, расположенных на территории муниципального образования» (Приказ 47н) одноименному справочнику; коды категорий земель, указанные в данном приложении, представлены полем **CODE**;
- в справочнике «Разрешенное использование земель» (**GROUND**s\_RELEASE) имеются реквизиты «Признак - индивидуальное строительство» (**INDIVIDUAL**) и «Признак - жилищное строительство» (**HOUSING**);
- на основе справочника «Разрешенное использование земель» (**GROUND**s\_RELEASE) поддерживается расчетный коэффициент (см. главу 14) с кратким названием  $L_{tax}$  - «Ставки земельного налога»;
- справочник «Регионы» (**REGIONS**) соответствует приводимому в Приложении 2 к Рекомендациям по заполнению формы «Сведения о земельных участках, расположенных на территории муниципального образования» (Приказ 47н) справочнику «Коды субъектов Российской Федерации»; коды субъектов РФ, указанные в данном приложении, представлены полем **CODE**;
- таблица **ADDRESS** привязана к справочнику **REGIONS** по полю **REGIONS\_ID**.

## 22.2. Необходимое информационное обеспечение

Для получения полных и достоверных сведений отчета, предусматриваемого положениями Приказов 47н и САЭ-3-13/485@, в БД SAUMI должна быть внесена перечисляемая ниже информация.

1. У каждого административно-территориального образования должны быть указаны код ОКТМО и уполномоченный распорядитель собственности – соответствующий Комитет. (см. «Свойства субъекта права „Административно-территориальное образование“» в главе «Регистрация объектов и субъектов права в SAUMI» руководства пользователя).
2. В окне свойств каждого Комитета, как юридического лица, на вкладке **Реквизиты** должны быть введены значения реквизитов, копируемые в описываемый отчет, как то:
  - *Руководитель* – ФИО лица, возглавляющего Комитет;



- *Должность* – официальное название должности руководителя Комитета;
- коды *ИНН* и *КПП*, указанные в свидетельстве о постановке на учет в налоговом органе Комитета как юридического лица;
- в адресе Комитета (вкладка **Юридическое лицо** в окне свойств Комитета как юридического лица) должен быть указан регион;
- *Налоговый орган* – наименование налогового органа, выбираемое из справочника «Налоговые органы» (таблица RATING\_AUTHORITY). Контекстное меню справочника позволяет ввести в справочник требуемую запись в случае ее отсутствия. Значения полей требуемой записи справочника «Налоговые органы» указываются в свидетельстве о постановке на учет в налоговом органе.

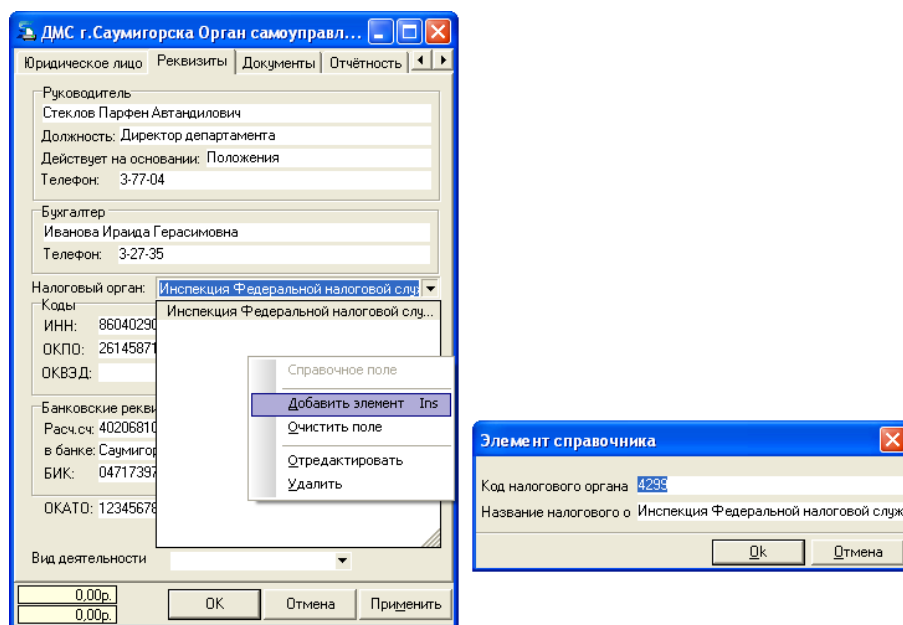


Рисунок 22.1.

3. Для расчетного коэффициента  $L_{tax}$  – «Ставки земельного налога» – необходимо создать период и внести значения ставок земельного налога для различных вариантов разрешенного использования земель в соответствии с действующими нормативами, определяющими величины указанных ставок. Так как сведения для описываемой отчетности должны предоставляться по состоянию на 1 января года, являющегося налоговым периодом, то и период расчетного коэффициента  $L_{tax}$  должен обеспечить его актуальность на эту дату.

### Замечание

Создавать новый период расчетного коэффициента необходимо только в случае, когда в установленном порядке в муниципальном образовании принимаются изменения ставок земельного налога. При создании нового периода в него копируются значения расчетного коэффициента периода предыдущего, что упрощает внесение небольшого числа изменений.

4. Содержание справочников «Категория земель» (**GROUNDTYPES**) и «Регионы» (**REGIONS**) должно соответствовать Приложениям 1 и 2 к Рекомендациям по заполнению формы «Сведения о

земельных участках, расположенных на территории муниципального образования» (Приказ 47н) соответственно.

5. В справочнике «Разрешенное использование земель» (**GROUNDS\_RELEASE**) необходимо правильно выставить значения признаков индивидуального строительства и жилищного строительства для каждого элемента справочника. Для этого следует либо воспользоваться утилитой «Менеджер справочников» из комплекта администратора SAUMI, либо на вкладке **Общее** окна свойств земельного участка для реквизита *Разрешенное использование* раскрыть список выбора значений и с помощью контекстного меню вызвать редактирование элемента данного справочника.
6. Для земельных участков, на которые оформлены ОД «Собственность», «Бессрочное пользование» и «Наследуемые владения», должны быть заданы кадастровый номер, код ОКТМО, категория земель и разрешенное использование.
7. У каждого правообладателя должен быть задан адрес, а в адресе – регион. Кроме того, у юридических лиц необходимы *ИНН* и *КПП*; у физических лиц реквизит «*Ф.И.О.*» должен содержать имя и отчество без сокращений, паспортные данные должны быть представлены в полном объеме.

## 22.3. Работа плагина

После запуска модуля проводятся необходимые проверки на полноту и соответствие информации. При обнаружении ошибок в данных открывается окно, содержащее протокол обнаруженных ошибок.

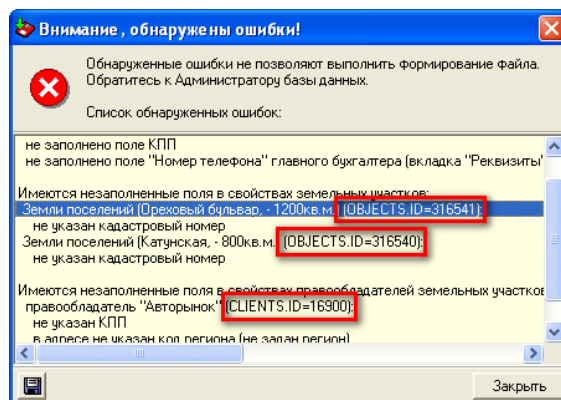



Рисунок 22.2. Протокол ошибок

Протокол содержит перечень выявленных несоответствий, причем все они привязаны к конкретным экземплярам сущностей SAUMI, которые также указаны в протоколе в виде (имя\_таблицы.ID=рег.код), где рег.код является регистрационным кодом экземпляра сущности SAUMI в БД (рис. 22.2). Двойной щелчок по строке протокола, содержащей описанную конструкцию, открывает окно свойств определенного регистрационным кодом экземпляра сущности SAUMI, позволяя осуществить надлежащую корректировку данных. Протокол ошибок может быть сохранен в текстовый файл, нажатием на кнопку .

После исправления ошибок нужно запустить модуль заново. И так до тех пор, пока все ошибки не будут исправлены, и не откроется главное окно модуля.

Рисунок 22.3. Главное окно

Год текущего налогового периода определяется из системной даты и изменению средствами SAUMI не подлежит.

При выставленном флажке «Предоставить сведения за все налоговые периоды» в отчет вносятся все имеющиеся в БД SAUMI сведения о земельных участках и их правообладателях. Такой отчет следует выполнить при первом обращении в налоговый орган на основании Приказа No 47. В последующие налоговые периоды описываемый флажок выставлять не следует, в результате чего в отчете будут только сведения за год, предшествующий году текущего налогового периода.

Значения группы полей, собранных в секции «Руководитель», выбираются из реквизитов соответствующего Комитета, как юридического лица, и могут быть изменены редактированием.

Сопроводительный документ, как отмечено в преамбуле настоящей главы, к моменту формирования отчета должен быть уже подписан и зарегистрирован, его номер и дата подписания вносятся в соответствующие поля секции «Сопроводительный документ».

В секции «Исполнитель» вносится информация о пользователе SAUMI, запустившем описываемый плагин, выбираемая из зарегистрированных в БД сведений о пользователе. В качестве номера телефона подставляется номер телефона главного бухгалтера распорядителя собственности АТО. Эти данные также могут быть изменены редактированием.

После надлежащего заполнения и редактирования всех полей в главном окне нужно нажать кнопку «Начать». В результате будут сформированы два файла (XML и XSD), определенные Приказом Федеральной Налоговой Службы от 28 июля 2006 г. N САЭ-3-13/485@, о чем выводится соответствующее сообщение.

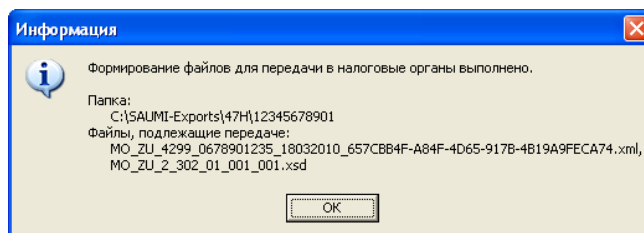


Рисунок 22.4. Главное окно

Файлы размещаются в папке, задаваемой полем «Путь для выгрузки» (по умолчанию `C:\SAUMI-Exports\47Н\код_ОКТМО`). Если в указанной папке до формирования отчета находились какие-то файлы – они удаляются. Передаче в налоговую инспекцию подлежат оба полученных файла и Акт приёма-передачи (сопроводительный документ).

# Приложение А. Перечень таблиц

	Раздел	Название таблицы или представления	Тип	Описание	
1	Объекты	OBJECTS	TABLE	Объекты	
2	Объекты		OBJTYPES	TABLE	Типы объектов
3	Объекты		OBJPHOTOS	TABLE	Фотографии объектов
4	Объекты		MONUMENTTYPES	TABLE	Виды памятников
5	Объекты		CLS_OKOF	TABLE	ОКОФ (классификатор)
6	Объекты	PROPERTY	TABLE	Имущество	
7	Объекты		PROPSECTIONS	TABLE	Разделы товаров
8	Объекты		PROPGROUPS	TABLE	Группы товаров
9	Объекты		PROPNames	TABLE	Названия товаров
10	Объекты	BUILDINGS	TABLE	Здания и помещения	
11	Объекты		BUILDTYPES	TABLE	Тип строения
12	Объекты		BUILDMaterials	TABLE	Вид стройматериала
13	Объекты		EXPLICATIONS	TABLE	Экспликации
14	Объекты	ENGINF	TABLE	Инженерн. инфрастр.	
15	Объекты		ENGINFTYPES	TABLE	Вид объекта
16	Объекты		INFGROUPS	TABLE	Группы объектов инж.
17	Объекты		RIVERS	TABLE	Реки
18	Объекты	GROUNDS	TABLE	Земельные участки	
19	Объекты		GROUNDTYPES	TABLE	Тип зем. участка
20	Объекты	TRANSPORT	TABLE	Транспорт	
21	Объекты		TRANSPType	TABLE	Типы транспорта
22	Объекты		BRANDNames	TABLE	Марки транспорта
23	Объекты	UNCONSTR	TABLE	Незаверш. строит	
24	Адрес	ADDRESS	TABLE	Адреса	
25	Адрес		STREETS	TABLE	Улицы
26	Адрес		TOWNAREA	TABLE	Район города
27	Адрес		PROVINCEAREA	TABLE	Район области
28	Состояние	STATES	TABLE	Состояние	
29	Состояние		STATETYPES	TABLE	Состояние объекта
30	Субъекты права	CLIENTS	TABLE	Субъекты права	
31	Субъекты права		SECTIONS	TABLE	Отрасли
32	Субъекты права	ORGANISATIONS	TABLE	Юридические лица	
33	Субъекты права		ORGFORMS	TABLE	Орг. - правовые формы
34	Субъекты права		ORGTYPES	TABLE	Типы юридических лиц

Перечень таблиц

	Раздел	Название таблицы или представления	Тип	Описание
35	Субъекты права	PRIVATES	TABLE	Физические лица
36	Движение	MOVESETS	TABLE	Операции движения
37	Движение	MOVETYPE	TABLE	Виды движения
38	Движение	TRANSFERBASIS	TABLE	Основания передачи
39	Движение	MOVEPERIODS	TABLE	Периоды операции движения
40	Движение	MOVEITEMS	TABLE	Движение объектов
41	Движение	LEASEPURPOSE	TABLE	Назначения
42	Движение	OBJECTUSING	TABLE	Использование площадей
43	Движение	USINGPRUPOSE	TABLE	Использование площади объекта
44	Движение	POLICYERRORS	TABLE	Описание несоответствий политики операций
45	Движение	MOVEMENTS	VIEW	Движение объектов
46	Финансы	OBLIGATIONS	TABLE	Обязательства
47	Финансы	PAYDOCS	TABLE	Договора по оплате
48	Финансы	OBL_PERIODTYPES	TABLE	Типы периодов для обязательств
49	Финансы	RECEPIENTS	TABLE	Получатели платежей
50	Финансы	OBLIGATIONTYPE	TABLE	Типы обязательств (назначения)
51	Финансы	CHARGES	TABLE	Начисления по обязательству
52	Финансы	PAYMENTS	TABLE	Оплаты
53	Финансы	DISCOUNTS	TABLE	Льготы
54	Финансы	CHARGES_APPLIES	TABLE	Проводки
55	ЦБ	SHARES	TABLE	Ценные бумаги и доли
56	ЦБ	SHARETYPES	TABLE	Виды ЦБ
57	ЦБ	SHARECATEGORY	TABLE	Категория ЦБ
58	ЦБ	SHARESBYORG	VIEW	Ценные бумаги по эмитентам
59	ЦБ	SHAREMOVE	TABLE	Движение ЦБ
60	Документы	DOCUMENTS	TABLE	Документы
61	Документы	DOCTYPES	TABLE	Виды документов
62	Документы	DOC_SUBSCRIBERS	TABLE	Списки сторон в документах
63	Документы	DOCPATTERNS	TABLE	Шаблоны документов Word
64	Документы	DOCPATTERNFIELDS	TABLE	Поля слияния документов Word
65	Документы	DOCNUMS	TABLE	Очередной номер документа
66	Документы	RESERVED_DOC_NUM	TABLE	Зарезервированные номера документов
67	Документы	DOCSETS	TABLE	Пакеты документов
68	Документы	DOCSET_MEMBERS	TABLE	Перечень документов в пакетах документов
69	Документы	DOCROLES	TABLE	Роли документов в пакетах документов
70	Отчетность	MUNREPORTFORMS	TABLE	Формы отчетности
71	Отчетность	MUNREPORTLINECODES	TABLE	Описание строк отчетных форм
72	Отчетность	MUNREPORTCOLCODES	TABLE	Описание граф отчетных форм
73	Отчетность	MUNREPORTREGLAMENT	TABLE	Регламент отчетов

Перечень таблиц

	Раздел	Название таблицы или представления		Тип	Описание
74	Отчетность		MUNREPORTFORMS_REG	TABLE	Привязка отчетных форм к периодам
75	Отчетность		MUNREPORTDATAOPT	TABLE	Файлы шаблонов отчетности
76	Отчетность	MUNREPORTS		TABLE	Данные о принятых отчетах
77	Отчетность		MUNREPORTLINES	TABLE	Строки принятых отчетных форм
78	Отчетность	PAYPURPOSE		TABLE	Средства приобретения имущества
79	Inflex	REPORTSTREE		TABLE	Дерево отчетов и сущностей ядра Inflex
80	Inflex		REPORTSSUBTYPES	TABLE	Описание субсущностей ядра Inflex
81	Inflex		DEFINECOLUMNS	TABLE	Столбцы для формы
82	Inflex		DEFINECOLUMNS_COMPLEX	VIEW	Столбцы для формы (с наследованием колонок от сущности)
83	Inflex		STYLELEGEND	TABLE	Легенда стилей строк окна данных
84	Inflex	REPORTFORMS		TABLE	Формы отчетности
85	Inflex	DICTIONARIES		TABLE	Справочники
86	Inflex	REGISTEREDUSERS		TABLE	Пользователи системы
87	Inflex	FM_TABLES		TABLE	Таблицы настраиваемые Менеджером реквизитов
88	Inflex		FM_TAB_COLUMNS	TABLE	Расширенное описание полей таблиц
89	Inflex		FM_V_TAB_COLUMNS	VIEW	Расширенное описание полей таблиц (удобно для просмотра)
90	Inflex	SM_CHANGES		TABLE	Таблица хранения истории значений
91	Inflex		SM_V_CHANGES	VIEW	История значений (удобно для просмотра)
92	Inflex	SM_SETTINGS		TABLE	Глобальные настройки (Диспетчер настроек)
93	Inflex	SP_BODY		TABLE	Текст модулей SAUMI-Pascal (Редактор SAUMI-Pascal)
94	Inflex	PRESET_COLUMNS		TABLE	Пользовательские настройки видимости колонок окна данных ядра Inflex
95	Inflex	SM_USER_BUFFER		TABLE	Содержание пользовательского буфера длительного хранения
96	Inflex	SM_USER_HISTORY		TABLE	Перечень последних данных, с которыми работал пользователь (история)
97	Inflex	SM_USER_REPORTSTREE		TABLE	Пользовательское меню «Отчеты»
98	Inflex	SM_USERSETTINGS		TABLE	Пользовательские установки (Диспетчер настроек)
99	Inflex	USER_REPORTFORMS		VIEW	Пользовательское дерево отчетов, отфильтрованное по правам
100	Служебные	MONTHS		TABLE	Перечень названий месяцев года
101	Служебные	NUMS		TABLE	Служебная таблица механизмов аналитики ядра Inflex

# Приложение В. Перечень серверных методов

<b>Методы для работы с объектами</b>		
	GETOBJECTDESCRIPTION	Возвращает название объекта
	GETPOSITION	Возвращает имя субъекта права, использующего данный объект (на текущий момент)
	GETPOSITIONCLIENTID	Возвращает ID субъекта права, использующего данный объект (на текущий момент)
	GETPOSITIONCLIENTIDBYDATE	Возвращает ID субъекта права, использующего данный объект на определенный момент
	GETPOSITIONENDDATE	Возвращает дату начала операции по указанному виду движения
	GETPOSITIONSTARTDATE	Возвращает дату окончания операции по указанному виду движения
	GETPOSITIONMOVEID	Возвращает ID текущей операции движения указанного вида
	GETPURPOSE	Возвращает назначение объекта
	GETOBJSTATE	Возвращает состояние объекта
	GETOBJSTATEID	Возвращает ID состояния объекта
	GETOBJECTINDICATOR	Функция вычисления значение индикатора
	GETOBJECTCOST	Функция вычисления значение стоимости объекта на заданный момент времени
	GETFIRSTUSAGEDAY	Функция возвращает дату ввода в эксплуатацию (первый день использования объекта). Анализируется состояние объекта, реквизит «Объект введен в эксплуатацию»
<b>Методы для работы с субъектами права</b>		
	GETCLIENTBALANCE	Возвращает текущее сальдо субъекта права
	GETCLIENTBALANCEBYDATE	Возвращает сальдо субъекта права на определенную дату
<b>Методы для работы с операциями движения</b>		
	GETMOVEBALANCE	Возвращает текущее сальдо операции
	GETMOVEMENTSINDICATOR	Функция вычисления значение индикатора
	GETDOCBYMOVEID	Возвращает ID документа-основания операции
<b>Бухгалтерия</b>		
	CALCFINE	Возвращает сальдо и пеню по обязательству
	FULLRECALCCHARGES	Полный перерасчет и регистрация начислений по обязательству и вызов перерасчет пени
	FULLRECALCFINES	Полный перерасчет пени
	PUTCUSTOMCHARGE	Зарегистрировать произвольное начисление
	DELCUSTOMCHARGE	Удалить произвольное начисление
<b>Документы</b>		
	QUERYDOCNUM	Запрос номера документа для документов с автонумерацией
	RETURNDOCNUM	Возврат номера документа для документов с автонумерацией
	EXPANDADDR	Описание адреса по ID
<b>Сервисные методы</b>		



## Перечень серверных методов

	EXPANDDOC	Описание документа по ID
	DECODEMONTHYEAR	Возвращает название месяца и год по дате
	GETMONTHNAME	Возвращает название месяца по дате
	GETPARAMETER	Значение параметра таблицы глобальных настроек SM_SETTINGS
	GETUSERNAME	Полное имя текущего пользователя
<b>Служебные методы</b>		
	REPORTINHERITED	
<b>Устаревшие методы с версии 2.4i</b>		
	CANMOVE	

## Приложение С. Роли (привилегии)

№	Привилегия	Описание
1	SM_ALLUSERS	Обязательная для всех пользователей SAUMI роль, которая включает в себя минимальный набор прав, необходимый для запуска SAUMI и просмотра информации.
2	SM_ANALITICS	Обладатель данной роли может формировать аналитические отчеты. (Аналитические отчеты, подготовленные администратором и предоставленные для всеобщего пользования через меню «Отчеты» могут выполняться и без наличия этой роли).
3	SM_AUDIT	Элемент «Аудит» отображается в меню «Отчеты».
4	SM_AUTOPAYMENTADMIN	Роль позволяет настраивать автоматическую разность платежей.
5	SM_CANCHARGE	Право внесения дополнительных начислений.
6	SM_CANDICTS	Возможность ведения справочников.
7	SM_CANFINANALIZE	Финансовый анализ.
8	SM_CANGIS	ГИС-возможности.
9	SM_CANGIS_WRITE	Редактирование объектов на карте.
10	SM_CANMOVE	Для оформления операций движения необходимо иметь эту роль. Но ее одну иметь не достаточно, также необходимо иметь роль, соответствующую определенному виду движения SM_MOVEXX_YY, где XX-номер вида движения в таблице MOVETYPE.
11	SM_CANMOVESHARE	Операции с ценными бумагами.
12	SM_CANPAY	Внесение информации о платежах.
13	SM_CAN_COLLAPSE_STRUCTURE	Возможность сворачивания/разворачивания структуры на вкладке «Структура» окна свойств здания.
14	SM_CAN_CREATE_ATO	Позволяет создавать субъекты права типа «АТО».
15	SM_CAN_CREATE_BUILD	Позволяет создавать объекты типа «Здания и помещения».
16	SM_CAN_CREATE_CLIENTS	Позволяет создавать субъекты права типов, дополнительно специфицированных другими ролями SM_CAN_CREATE_*.
17	SM_CAN_CREATE_EMPTY_MOVE	Создание безобъектных операций движения.
18	SM_CAN_CREATE_ENGINF	Позволяет создавать объекты типа «Инженерная инфраструктура»
19	SM_CAN_CREATE_GROUNDS	Позволяет создавать объекты типа «Земельные участки»
20	SM_CAN_CREATE_OBJECTS	Позволяет создавать объекты типов, дополнительно специфицированных другими ролями SM_CAN_CREATE_*.
21	SM_CAN_CREATE_ORGS	Позволяет создавать субъекты права типа «Юридические лица».
22	SM_CAN_CREATE_PRIVATES	Позволяет создавать субъекты права типа «Физические лица».
23	SM_CAN_CREATE_PROPERTY	Позволяет создавать объекты типа «Прочее движимое имущество».
24	SM_CAN_CREATE_TRANSPORT	Позволяет создавать объекты типа «Транспортные средства».
25	SM_CAN_CREATE_UNCONSTR	Позволяет создавать объекты типа «Незавершенное строительство».
26	SM_CAN_POLICY_IGNORE	Разрешено игнорировать политики операций.
27	SM_CAN_SDE	В меню «Дополнения» главной панели SAUMI присутствует SDE.

Роли (привилегии)

№	Привилегия	Описание
28	SM_CAN_WHOLE_SQUARE_MOVE	Позволяет при оформлении ОД из вкладки «Структура» окна свойств объекта, передавать объекты «Зданий и помещений» целиком, не показывая диалоговое окно «Участвует ли в ОД помещение целиком или только его часть».
29	SM_CAN_WORK_MUNREPORT	Показывает вкладку «Отчетность» в окне свойств юридического лица, обеспечивая доступ к электронной отчетности.
30	SM_CAN_WORK_WITH_DOCBODY	Определяет возможность работы с содержимым документов (сканирование, просмотр, удаление/добавление, редактирование и т.д.).
31	SM_EDIT_BUILD_STRUCTURE	Разрешено редактировать структуру зданий.
32	SM_EDIT_HTMLTMPL	Возможность редактирования html-шаблонов для учетных записей, отличных от SM.
33	SM_GIS_ADMIN	Администратор ГИС.
34	SM_GIS_MANAGER	Менеджер ГИС.
35	SM_GIS_MANAGE_RAW_LAYERS	Не используется. Присутствует для совместимости.
36	SM_GMP_COMMUNICATION	Делает доступным меню «ГИС ГМП» в модуле «Менеджер финансовых поступлений».
37	SM_GROUPMOVES	Групповые операции движения.
38	SM_MERGE	Право на объединение записей (реализовано в ядре для субъектов права).
39	SM_MOVE10_KAZNA	Казна.
40	SM_MOVE10_UPRAVLENIE	Управление.
41	SM_MOVE11_AREND_FOND	Арендный фонд.
42	SM_MOVE12_ZALOG	Залог.
43	SM_MOVE13_ZFOND	Залоговый фонд.
44	SM_MOVE14_OHRANA	Охрана.
45	SM_MOVE15_STRAH	Страхование.
46	SM_MOVE16_DOVUPR	Доверительное управление.
47	SM_MOVE17_ZHILOE	Жилой фонд.
48	SM_MOVE18_SPIS	Списание.
49	SM_MOVE19_BSRPOLZ	Бессрочное пользование.
50	SM_MOVE1_SOBSTV	Собственность.
51	SM_MOVE20_OGRPOLZ	Ограниченное пользование (сервитуты).
52	SM_MOVE21_BVOZMPOLZ	Безвозмездное пользование.
53	SM_MOVE22_NASLVLAD	Наследуемое владение.
54	SM_MOVE23_ZAJAVKA	Заявка.
55	SM_MOVE24_BANKROT	Банкротство.
56	SM_MOVE25_LIKVID	Ликвидация.
57	SM_MOVE25_RAZM	Размещение рекламных конструкций.
58	SM_MOVE26_AREST	Арест.
59	SM_MOVE27_OSOBO_ZENNOE	Особо ценное имущество.
60	SM_MOVE2_BALANS	Балансодержание.
61	SM_MOVE3_POLZOVANIE	Пользование.
62	SM_MOVE4_ARENDA	Аренда.

Роли (привилегии)

№	Привилегия	Описание
63	SM_MOVE5_SUBARENDA	Субаренда.
64	SM_MOVE6_STROITELSTVO	Строительство.
65	SM_MOVE7_PRIVATIZACIA	Приватизация.
66	SM_MOVE8_OBSLUZHIVANIE	Обслуживание.
67	SM_MOVE9_UCHET	Учет.
68	SM_SAS_ADMIN	Роль необходима для запуска консоли управления сервисами SAS.
69	SM_SAS_USER	Роль необходима для запуска консоли управления запросами SAS.
70	SM_SDE_MANAGER	Менеджер SDE.
71	SM_VIEW_TOTALS	Просмотр свода расчетов по обязательству.