

# **Geocad System Enterprise Edition**

## **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**ОБЕСПЕЧЕНИЕ РАБОТОСПОСОБНОСТИ  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.  
ТЕХНИЧЕСКАЯ ПОДДЕРЖКА.**

**Документация актуальна для программных продуктов:**

**Сервер приложения Geocad System Enterprise Edition**

**АИПС "Адресный реестр"**

**АИПС "Имущественный комплекс"**

**АИПС "Обеспечение градостроительной деятельности"**

**КАИС "Территориальный кадастр предприятия"**

**ПРЕДУПРЕЖДЕНИЕ!**

В содержание данного документа могут быть внесены изменения без предварительного уведомления и ООО «ГЕОКАД плюс» не берет на этот счет никаких обязательств. Описанное здесь программное обеспечение поставляется по лицензионному соглашению или по соглашению о нераспространении. Это программное обеспечение может быть использовано или скопировано лишь в строгом соответствии с условиями этого соглашения. Копирование этого программного обеспечения на какой-либо носитель информации, если на это нет специального разрешения по лицензионному соглашению или соглашению о нераспространении, является противозаконным действием. Никакая часть настоящего руководства ни в каких целях не может быть воспроизведена или передана в какой-либо форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на то нет письменного разрешения ООО «ГЕОКАД плюс».

# Оглавление

<b>Глава I: Введение</b>	<b>6</b>
О документации.....	6
Концепция построения системы, терминология.....	6
О сервере приложения.....	8
О пользователях.....	10
<b>Глава II: Установка программы</b>	<b>12</b>
Введение.....	12
Установка сервера в Windows.....	12
Установка сервера в Linux.....	12
Настройка сервера после установки.....	13
Введение .....	13
Приложение Конфигуратор .....	13
Введение .....	13
Параметры Java-машины.....	14
Параметры серверной части системы.....	14
Настройка лицензии.....	15
Параметры сервисов.....	16
Параметры ведения логов (журналирования).....	17
Параметры HTTP-подключения и сервлетов.....	18
Редактор сертификатов.....	20
Настройка пароля администратора сервера приложений.....	20
Управление контейнером бинов .....	21
Подключение метамоделей .....	21
Введение .....	21
Подключение нескольких метамоделей.....	21
Инициализация структур метамоделей .....	22
<b>Установка клиент-приложений в Windows.....</b>	<b>22</b>
Состав приложений .....	22
Общие сведения .....	22
Установка приложений .....	23
Технические подробности .....	23
Запуск приложений .....	24
<b>Установка обновлений.....</b>	<b>24</b>
Обновление программного обеспечения .....	24
Поставка и порядок установки.....	24
Обновление клиентов .....	25
Обновление структур метамоделей.....	25
<b>Обновление метамоделей .....</b>	<b>26</b>
Общие сведения.....	26
Порядок обновления.....	26
Полное обновление.....	27
<b>Глава III: Конструктор</b>	<b>30</b>
Введение.....	30
Назначение программы .....	30
Окно программы .....	30
Управление сервером.....	30

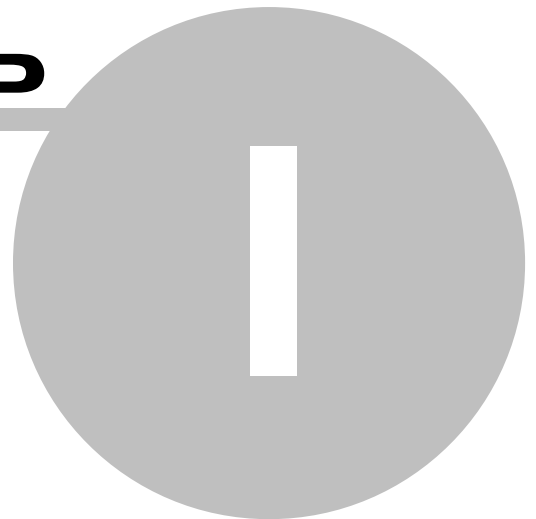
---

Введение .....	30
Память .....	30
Процессор .....	32
Параметры/логи .....	33
Пользователи .....	34
Коннекции .....	36
Обслуживание .....	37
<b>Диагностика различных неполадок.....</b>	<b>38</b>
Периодическое подтормаживание при обращении к серверу .....	38
Неполадки, связанные с файловым хранилищем .....	38
<b>Глава IV: Обеспечение жизненного цикла</b>	<b>41</b>
Устранение неисправностей.....	41
Совершенствование ПО.....	41
Квалификация персонала.....	42
<b>Глава V: Техническая поддержка</b>	<b>44</b>
Лицензионное соглашение.....	44
Поддержка программных продуктов.....	44
<b>Индекс</b>	<b>46</b>

# Введение

**Часть**

---



## Глава I: Введение

### О документации

Данное руководство предназначено для пользователей автоматизированной информационной пространственной системы Geocad System Enterprise Edition (далее по тексту GSEE) и не содержит описания режимов работы и установок программных оболочек (Microsoft Windows, Oracle и других), совместно и под управлением которых работает система. Для получения подробной информации по данным вопросам обращайтесь к соответствующей документации.

Мы предполагаем, что Вы знаете, как работать в среде Windows. Если такие понятия, как "диалоговые окна", "пиктограммы", "полосы прокрутки" и т.п., вызывают у Вас затруднения, мы рекомендуем Вам обратиться к руководству пользователя Microsoft Windows.

Комплект документации GSEE состоит из руководства пользователя и электронных справочников.

Встроенные справочники используются для оперативной помощи пользователю непосредственно в процессе работы с отдельными модулями. Так как все справочники модулей системы используют принцип контекстного поиска, то при их вызове Вы получаете быстрый доступ к информации о текущем режиме работы.

### Концепция построения системы, терминология

В этой главе приведены основные определения, термины и понятия, необходимые пользователю для успешного освоения системных решений GSEE.

GSEE представляет собой распределённую трехуровневую систему, состоящую из СУБД, сервера приложений и клиентских приложений. Код сервера и клиент-приложений выполняется в среде Java, благодаря этому GSEE является платформенно-независимым комплексом и может одинаково работать в разных аппаратных и программных платформах.

Распределение компонентов системы по трем уровням выглядит так:

#### Данные:

- банки данных;
- система управления банками (Oracle, MySQL и др. на выбор заказчика);
- метамодель;

#### Сервер:

- сервер приложений JBoss (открытая архитектура, один из лидеров отрасли);
- серверные компоненты GSEE (библиотеки функций для JBoss, реализующие функционал системы);
- конфигурационные файлы (доступные администратору низкоуровневые настройки);

#### Клиенты:

- приложение для администрирования Constructor;
- клиент-приложение для работы с семантической информацией Forms Pro;
- клиент-приложение для работы с графической информацией Maps Pro;
- клиент-приложение для построения пользовательских отчетов Logic Reports;
- веб-приложение для работы с семантической информацией Web Info Forms;
- веб-приложение для работы с графической информацией Web Maps;
- веб-приложение для работы с семантической и графической информацией Web Lite.

---

Рассмотрим и определим отдельные части системы подробнее.

**Банк данных** - это совокупность пользовательской информации об объектах учётной территории. Банк традиционно делится на две части: данные, непосредственно описывающие

объекты конкретной территории, и данные, не зависящие от объектов, но используемые при их описании (их принято называть **классификаторами** либо **общим банком**).

Банки данных создаются административными приложениями и обслуживаются операторами в процессе эксплуатации информационной системы. Классификаторы обычно поставляются в комплекте с информационной системой, заполненными в соответствии с законодательством и задачами этой системы. Изменение и дополнение классификаторов операторами допускается, но не рекомендуется, так как в этом случае возникнут проблемы при обновлении программы на более новую версию.

Каждый банк данных является конечной реализацией заранее подготовленной **информационной модели (метамодели)**. Такая модель является эталоном представления данных некой информационной концепции (например, Реестр имущества) и содержит описание структур (баз данных, таблиц, полей, связей, объектов, форм, отчетов и т.д.), реализующих эту концепцию. В общем случае один банк может реализовывать и несколько метамodelей, в том числе и пересекающихся по своим сущностям. Обычно метамodelи так же поставляются в комплекте системы, разработанные под требования заказчика и законодательства, однако могут создаваться, изменяться и дополняться и на месте пользователем-администратором.

Для удобного оперирования данными пользователям предоставляется набор **клиент-приложений**, содержащих все необходимые интерфейсные компоненты и инструменты: формы представления и редактирования данных, вызов процедур обработки, работа с графикой и отчетами. Существуют также реализации приложений для работы в среде Web, с меньшей функциональностью, но более простые в установке и освоении.

Клиент-приложения не работают с данными напрямую. Для управления банками данных (во всех аспектах этого понятия) служит **сервер приложений**. Сервер контролирует непосредственно базу данных, обращаясь к ней по мере необходимости, он же выполняет процедуры, генерирует отчеты, управляет метамodelью и системой безопасности. Клиент-приложение в процессе работы обращается к нему за данными для отображения, а так же за объектами метамodelи, требующимися для формирования этого отображения, и отправляет обратно внесенные изменения.

Примечание: хотя формально веб-приложения являются клиентскими, основная их программная часть исполняется на стороне сервера, в специальном компоненте (веб-контейнере). На клиентской стороне (в веб-браузере) отображается только визуальная часть.

Таким образом, вся нагрузка по обработке централизованно сосредоточена на сервере, разгружая и упрощая работу приложений. Кроме того, сервер приложений абстрагирует физическое хранилище банка данных, позволяя одинаково работать с разнородными объектами. Совокупность отлаженных системных соглашений и абстрагирование через метамodelь позволяет адаптировать программный комплекс GSEE для решения разнообразных задач без внесения модификаций в программную часть как сервера, так и клиент-приложений.

Отдельно стоит выделить административные приложения, сгруппированные под общим именем "**Конструктор**". Конструктор предоставляет инструментарий для модификации метамodelи и структур банков данных, раздачи прав пользователям, управления сервером и др. С Конструктором, как правило, работает пользователь-администратор, прошедший специальный курс обучения и имеющий повышенные привилегии в системе безопасности.

---

Рассмотрим далее другие термины, с которыми сталкивается пользователь в процессе работы.

**Таблица (данных)**: буквально таблица (в терминах СУБД), находящаяся в банке данных. Помимо информационных, таблица может иметь ряд служебных полей, связанных с функционированием системы. Таблицы могут ссылаться друг на друга, образуя полноценную реляционную базу данных.

**Семантический объект** (обычно просто **объект**, **факт**): одна запись из таблицы данных, дополненная данными из связанных таблиц (т.н. расшифровки). Пользователь клиент-приложений, как правило, работает именно с объектами.

Объекты естественным образом объединены по **типам объектов**. Описание типа в метамodelи содержит указание на главную таблицу, выбираемые поля, запрос на выборку, понятные имена полей с комментариями, формы и т.п. Обычно один тип объекта строится над одной таблицей, но бывают и исключения, когда для одной таблицы описаны несколько типов (т.н. альтернативные объекты,

например, Зарегистрированные и Новые участки).

Типы объектов объединяются по каким-либо признакам в группы, называемые **аспектами**. Например, объекты недвижимости (участки, здания), природные объекты окружающей среды (водные поверхности, лесные массивы), другие искусственные сооружения (железные дороги, автомагистрали). Аспекты служат исключительно для визуальной группировки и не играют никакой другой роли.

Для многих типов объектов помимо семантической информации нужна еще и графическая (пространственные данные). Набор координат, объединенных в контуры разного типа (полигоны, полилинии и т.п.) называется **графическим объектом**. Графический объект обычно привязан к семантическому, они взаимно друг друга дополняют, хотя в общем случае это не обязательно.

Типу семантических объектов соответствует тип графических, называемый **слоем**. Название "слой" отражает способ отображения графических объектов - слоями, одни над другими. В пределах слоя объекты обычно пространственно не пересекаются. Слоев обычно меньше, чем типов семантических объектов, т.к. не всем типам нужна графика.

Набор типов объектов и слоев называется **моделью**. Комбинируя типы в зависимости от решаемой задачи можно на одних и тех же объектах (и банках данных) формировать разные модели; несколько "совместимых" моделей в таких случаях обычно размещаются в одной метамодели, что сокращает количество проектируемых суцностей.

Помимо непосредственного ведения банка данных, одним из результатов работы операторов являются **отчеты**. Сервер GSEE по запросу пользователя может генерировать несколько типов отчетов: в собственной системе Logic Reports, в формат RTF (Rich Text Format), в формат XLS (Microsoft Excel). Результат передается в клиент-приложение, откуда может быть сохранен в файл или напечатан. Каждый тип объектов может иметь свой набор отчетов, некоторые отчеты могут использоваться с несколькими (или даже со всеми) типами.

Для массовой обработки данных (автоматизированное редактирование, импорт-экспорт и т.п.) применяются **процедуры**. Процедуры так же выполняются на сервере по запросу пользователя и так же могут отличаться для разных типов объектов.

## О сервере приложения

Geocad Systems Enterprise Edition (GSEE) – сервер приложений, является ядром технологии GSEE и основной, неотъемлемой частью АИПС с использованием одного или нескольких клиентских приложений, по разным направлениям деятельности и работы с пространственными объектами. Кроме того GSEE - сервер приложений может использоваться как система для самостоятельной разработки клиентских приложений по технологии Geocad Systems Enterprise Edition (GSEE).

Технология GSEE представляет собой распределённую, многоуровневую систему, состоящую из набора серверных компонент и клиентских приложений. Система позволяет работать удалённо посредством сетей Intranet/Internet, имеет возможность установки одного корпоративного сервера на несколько организаций или на несколько управленческих структур, действующих внутри одной организации и работать на одном сервере с различными ГИС-моделями и различными банками данных. Возможность создания корпоративных и общедоступных интернет-порталов и др.

Основным назначением автоматизированных информационных пространственных систем разработанных по технологии GSEE является, совершенствование информационной поддержки принятия управленческих решений, реализация задач управления на локальных территориях, направленных на создание условий для формирования, развития и эффективного использования социально - экономического и ресурсного потенциала, на основе унификации и интеграции информационных ресурсов, используемых во всех областях жизнедеятельности на территории, и в первую очередь, направленных на повышение ее управляемости и привлекательности, эффективного использования земельно - имущественного комплекса, повышение качества услуг и жизни граждан.

В результате внедрения данной технологии достигается оперативный доступ и обеспечение достоверной атрибутивно - пространственной информацией широкого круга управленческо - хозяйствующих субъектов, публичных организаций и граждан, для их полноценного участия в реализации управленческих и гражданских полномочий, на основе современных подходов к использованию информационно - аналитических систем.



Кроме того целями внедрения являются:

- сбор, систематизация и интеграция накопленных описательных и пространственных данных по унифицированным правилам, единым (утвержденным) форматам;
- формирование единой политики управления распределенными потоками информации;
- своевременное предоставление (получение) информации об объектах хозяйственной инфраструктуры, расположенных на территории, обеспечение ее полноты, достоверности, актуальности и необходимой конфиденциальности;
- обеспечение возможности совместного использования разнородных данных, получаемых из различных источников, об объектах управления, расположенных на территории;
- уточнение и дополнение информации об объектах инфраструктуры, расположенных на территории, на основе космических и аэрофотоснимков и ГЛОНАСС/GPS технологий;
- разработка типовых решений в рамках данной технологии с целью их последующего тиражирования;
- создание территориальной информационной основы для последующего решения практических задач и взаимодействия структурных подразделений, специалистов Администраций и других заинтересованных лиц.

Данная технология полностью адаптируется и интегрируется с базами данных, созданных на основе различных программных продуктов, в том числе, предыдущего поколения Geocad Systems, вместе с тем обладает рядом существенных преимуществ, в частности это:

- Возможность работы в онлайн режиме по сети Internet без установки клиентского ПО, только с помощью браузера;
- Возможность работы на любых устройствах, под управлением любых операционных систем и СУБД;
- Веб-портал, возможность публиковать общедоступные данные в сети Интернет;
- Возможность создания корпоративного портала внутри предприятия;
- Возможность просмотра данных с помощью мобильных устройств (iPhone, iPad, мобильные телефоны);
- Возможность подключения предприятий и организаций удаленно, либо по корпоративной сети к единому банку с ограничением (либо без) на просматриваемую информацию;
- Развитая ролевая система безопасности с ограничением доступа на просмотр, создание, редактирование, экспорт (импорт) данных;
- Возможность работы в любой системе координат, с динамическим пересчетом в нужную, просмотр объектов имеющих разную систему координат совместно;
- Возможность работы с трехмерными данными, как в Google Earth (поверхности, объекты, модели);
- Масштабируемость и распределённость (в случае увеличения нагрузки на сервер можно устанавливать несколько серверов которые будут работать совместно);
- Возможность одновременной работы с различными базами данных (Oracle, MSSQL, MySQL, PostgreSQL);
- Кроссплатформенность - использование любой операционной системы, как для сервера, так и для клиента;
- Интеграция с различными ГИС (MapInfo, ArcInfo, Polis, Autocad) не только импорт, но и онлайн - интеграция в единый банк данных;
- Поддержка всемирных стандартов WFS, WMS (дает возможность подключаться к банку данных любой ГИС и самим подключаться к сторонним серверам);
- Возможность интеграции с системой документооборота LotusNotes пространственные ограничения на определенную территорию, ограничение на отдельные записи;
- Автоматизация процедур (запускаются по расписанию и позволяет часть работы выполнять по ночам в автоматическом режиме);
- Полный мониторинг работы с БД и развитая система архивирования;
- Развитая система визуализации слоев, с поддержкой произвольных границ, заливок, спецсимволов,

генерализации;

- Работа с метаданными по ГОСТ и по ISO и развитая система редактирования отчетов;
- Поддержка печати и сканирования штрих - кодов, интеграция с файловым хранилищем;
- Функциональность системного интегратора. Позволяет объединить вокруг себя различное ПО для решения прикладных задач.

## О пользователях

При работе с информационными системами на базе GSEE следует чётко понимать отличия в функциях и обязанностях различных категорий пользователей, определяющихся их правами доступа, как к различным программным модулям системы, так и к прикладным (пользовательским) информационным ресурсам.

Поэтому всех пользователей можно разделить на две основные категории:

- пользователи, имеющие административные права - **Администраторы системы**;
- пользователи, не имеющие административных прав - **Операторы системы**.

**Администратором системы** является привилегированный пользователь GSEE, зарегистрированный во время настройки системы (см. Права доступа). В его функции входят обязанности по настройке и поддержанию работоспособности системы, целостности баз данных, назначению прав доступа к информационным ресурсам, массовый импорт и экспорт информации баз данных, резервное копирование. Кроме того, Администратор должен заботиться об общей организации работ: распределение обязанностей операторов по вводу, редактированию и обработке данных, периодический контроль корректности и полноты введенной информации.

Также, Администратор может вести работы по разработке, созданию и корректировке пользовательских компонент системы, т.е. выполнять функции *Проектировщика системы*.

**Проектировщик системы** - это привилегированный пользователь GSEE, т.е. имеющий административные права (см. Права доступа), в функции которого входят: корректирование (адаптация) пользовательских компонент системы под текущие требования решаемой задачи (изменение и добавление таблиц, полей, процедур вычислений и др.), и разработка новых информационных систем (создание новых пользовательских компонент). Проектировщик системы должен хорошо владеть терминологией и средствами проектирования баз данных в среде СУБД и придерживаться правил и соглашений GSEE.

Все остальные пользователи являются **Операторами системы** и выполняют работы по операционному обслуживанию прикладной информации баз данных: ввод, редактирование, формирование и печать выходных документов. Оператор имеет доступ только к тем данным, которые определены правами, данными ему Администратором системы.

# Установка программы

**Часть**

---



## Глава II: Установка программы

### Введение

Компоненты GSEE поставляются в виде нескольких отдельных инсталляторов. В зависимости от договора и комплекта поставки, исполнимые файлы инсталляторов могут называться по-разному и находиться в разных папках комплекта.

Вместе с инсталляторами поставляются и автоматически устанавливаются соответствующие версии Java Runtime Environment (JRE), Java Advanced Imaging (JAI) и других необходимых для функционирования компонентов.

### Установка сервера в Windows

Инсталлятор сервера для операционных систем семейства Windows поставляется в виде исполнимого файла, содержащего внутри все необходимые компоненты, а так же выполняющего некоторые минимальные настройки.

В комплект инсталлятора входят следующие компоненты:

- Java Runtime Environment, 32-битная (i586) и 64-битная (amd64/x64)
- сервер приложений JBoss, веб-сервер Tomcat;
- ядро системы (основная часть системы, осуществляющая работу с базами данных);
- Web-приложения (GEE Web Forms, GEE Web Maps, GEE Web Light и т.д.);
- компоненты установки клиент-приложений по технологии Java Web Start.

В процессе установки будет предложено указать каталог установки, указать базовые сетевые параметры, а так же выбрать способ запуска - через ярлык или службой. После установки сервер может потребовать конфигурирования служб, портов, метамодели, банков данных. Эти действия должны выполняться администратором и описаны в других разделах документации.

В некоторых поставках инсталлятор сервера может содержать настроенную СУБД, а так же метамодели и банки данных.

### Установка сервера в Linux

Инсталлятор сервера для операционных систем семейства Linux поставляется в виде архива tar.gz, содержащего внутри все необходимые компоненты. Установка предполагает выполнение некоторых минимальных настроек вручную.

В комплект инсталлятора входят следующие компоненты:

- Java Runtime Environment, 32-битная (i586) и 64-битная (amd64/x64)
- сервер приложений JBoss, веб-сервер Tomcat;
- ядро системы (основная часть системы, осуществляющая работу с базами данных);
- Web-приложения (GEE Web Forms, GEE Web Maps, GEE Web Light и т.д.);
- компоненты установки клиент-приложений по технологии Java Web Start.
- Конфигуратор

Путь для распаковки архива выбирается исходя из того, что по умолчанию сервер во время работы создает и обновляет довольно много временных файлов в своих подкаталогах. При необходимости, конечно, подкаталоги могут быть перемещены как при помощи конфигурирования сервера, так и через символические ссылки, но обычно сервер размещают на открытой для записи файловой системе.

В корне каталога присутствует файл setjava.sh. Это простой скрипт, выбирающий среду JRE. По

умолчанию он настроен на 64-битную версию, идущую в комплекте с инсталлятором. В принципе, администратор может выбрать любую другую JRE версии не ниже 1.7, но в таком случае может потребоваться дополнительная настройка (например, установка Java Advanced Imaging).

Запуск сервера выполняется командой `server/bin/run.sh`, остановка - командой `server/bin/shutdown.sh`.

Так же в корне каталога сервера присутствует файл `gsee`. Это небольшой скрипт для установки сервиса через стандартный механизм `init.d`. Если есть необходимость запускать сервер именно как сервис, необходимо поместить скрипт в каталог `/etc/rc.d/init.d`, плюс создать ссылку на него в соответствующем каталоге `/etc/rc.d/rc.N` (где N - уровень запуска, обычно 3 или 5). Конкретные пути и механизм могут варьироваться в зависимости от дистрибутива Linux.

Обычно после распаковки архива все необходимые файлы скриптов автоматически получают установленным флаг запуска. В случае, если этого не случилось, надо расставить вручную флаг для всех упомянутых выше скриптов.

В принципе, сервер не требует запуска с правами `root` и вполне работоспособен и под ограниченным аккаунтом. Из обычных неудобств в таком случае чаще всего встречается невозможность использовать порт 80 для Web (порты с номерами ниже 1024 требуют привилегий `root`).

Если операционная система содержит в себе GUI-среду (X Window), для дальнейшего конфигурирования можно воспользоваться идущим в дистрибутиве Конфигуратором. Для запуска используйте скрипт `config/config.sh`.

## Настройка сервера после установки

### Введение

После установки сервера приложений требуется выполнить несколько шагов по конфигурированию.

1. Настройка сервисов, сетевых подключений, лицензий и сертификатов.
2. Подключение метамодели.
3. Инициализация структур метамодели в СУБД.
4. Загрузка метамодели из комплекта поставки. См. раздел [ЗАГРУЗКА И СОХРАНЕНИЕ МЕТАМОДЕЛИ](#) в документации по Конструктору.
5. Создание описания банка данных и инициализация его структур в СУБД. См. разделы [МАСТЕР СОЗДАНИЯ БАНКА ДАННЫХ](#) и [СОЗДАНИЕ И ПРОВЕРКА СТРУКТУР](#) в документации по Конструктору.
6. Создание пользователей в системе и установка разрешений для них. См. раздел [АДМИНИСТРАТОР БЕЗОПАСНОСТИ](#) в документации по Конструктору.

## Приложение Конфигуратор

### Введение

Серверная часть системы GSEE базируется на сервере приложений JBoss, соответственно, большая часть настроек относится к конфигурированию служб JBoss. В основном сервер приложений настраивается несколькими файлами в формате XML, расположенными в разных местах. Для того, чтобы не править эти файлы «вручную» существует специальная утилита **Конфигуратор**.

Конфигуратор для своей работы, в отличие от всех остальных клиентских приложений, требует не удаленного доступа к серверу, а прямого, на уровне файловой системы. Более того, для корректной работы Конфигуратора сервер должен быть остановлен.

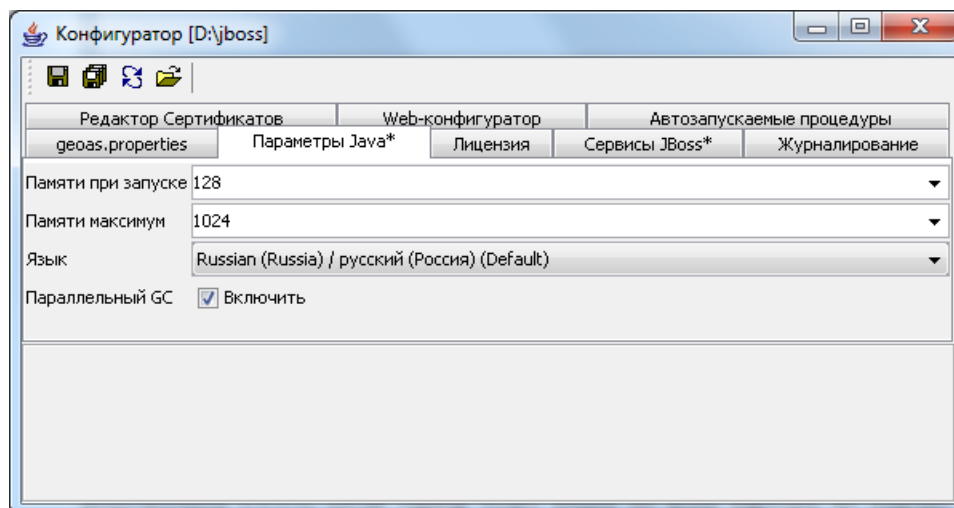
При запуске Конфигуратор требует указать каталог, куда установлен сервер, после чего считывает начальные параметры из файлов.

В случае невозможности запустить Конфигуратор (среда без GUI и без доступа к файлам сервера извне), все настройки могут быть выполнены путём правки файлов в ручную. В каждом разделе далее находятся сведения об этих файлах. Все пути к файлам описаны относительно пути установки сервера JBoss, т.е. обычно это подкаталог server в каталоге установки.

## Параметры Java-машины

Исполняющая среда Java (Java Runtime Environment, Java-машина) имеет ряд настроек, влияющих на ее производительность и поведение. Этим настроек довольно много, часть из них описана в документации к JRE (официальные), часть можно найти в Интернет (неофициальные), Конфигуратор «знает» только самые основные.

Настройки задаются в специальном файле bin\run.vmoptions, по одной в строке. Этот файл используется скриптами запуска (run.bat, run.sh и т.п.) для генерации строки запуска Java-машины.



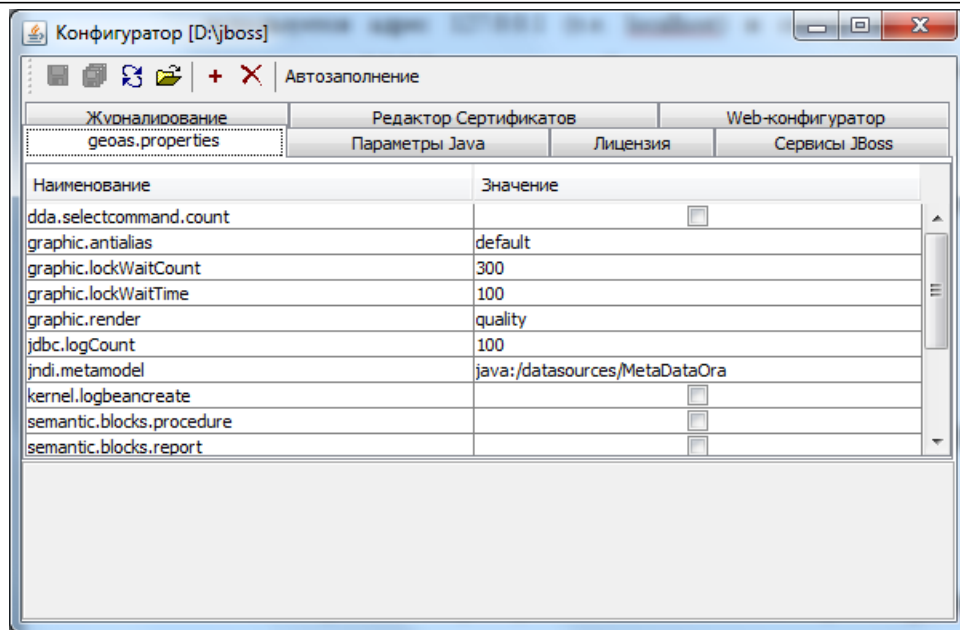
В первую очередь надо обратить внимание на настройки памяти, которую может использовать JRE, стартовое и максимальное значение, 256 и 512 мегабайт по умолчанию. Эти настройки, особенно вторая, должны подбираться в зависимости от нагрузки на сервер.

Для многопроцессорных систем с большой нагрузкой может использоваться альтернативный менеджмент памяти, т.н. «параллельный Garbage Collector».

Так же можно задать региональные параметры, отличные от настроек ОС. На практике это обычно используется для получения сообщений от драйверов базы данных на языке, более привычном пользователям.

## Параметры серверной части системы

По адресу server/default/conf/geoas.properties находится текстовый файл, содержащий настройки различных модулей серверной части системы GSEE. На одноименной вкладке Конфигуратора эти настройки представлены в виде таблицы.



Чаще всего администратора могут интересовать следующие параметры:

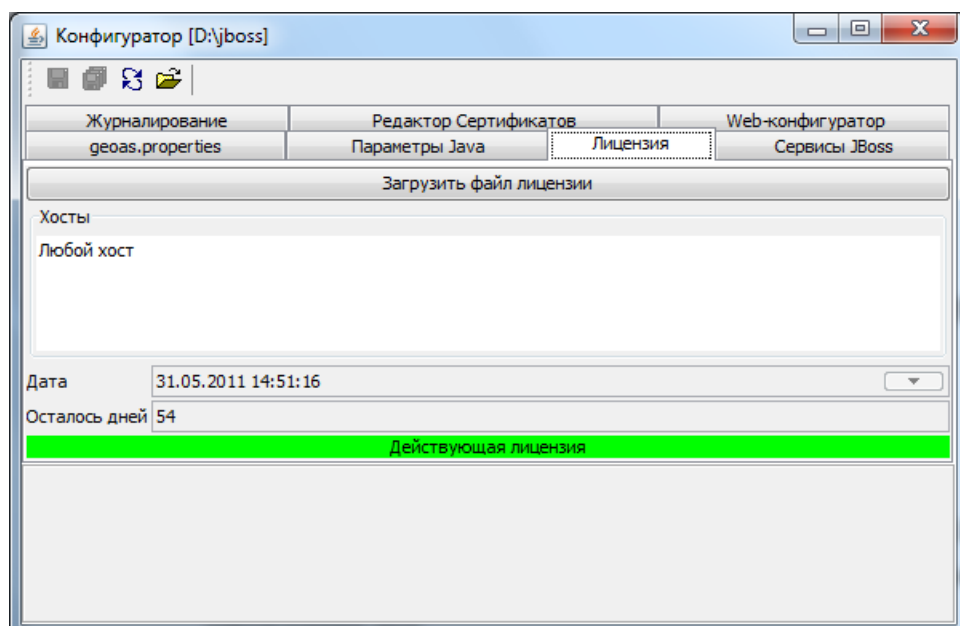
- **dda.selectcommand.count** – разрешать ли производить подсчет выбираемых данных. Если настройка “false”, клиент не сможет узнать, сколько данных будет выбрано по его запросу (например, при открытии формы) до тех пор, пока не выберет все данные. Если настройка “true”, нагрузка на сервер повышается.
- **jdbc.logCount** – включение протоколирования выполнения запросов к СУБД. Протокол ведется в память, настройка указывает, сколько запросов контролировать и помнить. Одновременно включается возможность просмотра текущих выполняемых запросов. См. Управление сервером, вкладка [Коннекции](#).

## Настройка лицензии

Сервер GSEE лицензируется на имя сервера (DNS-имя, по которому пользователи обращаются к нему), и на определенный срок.

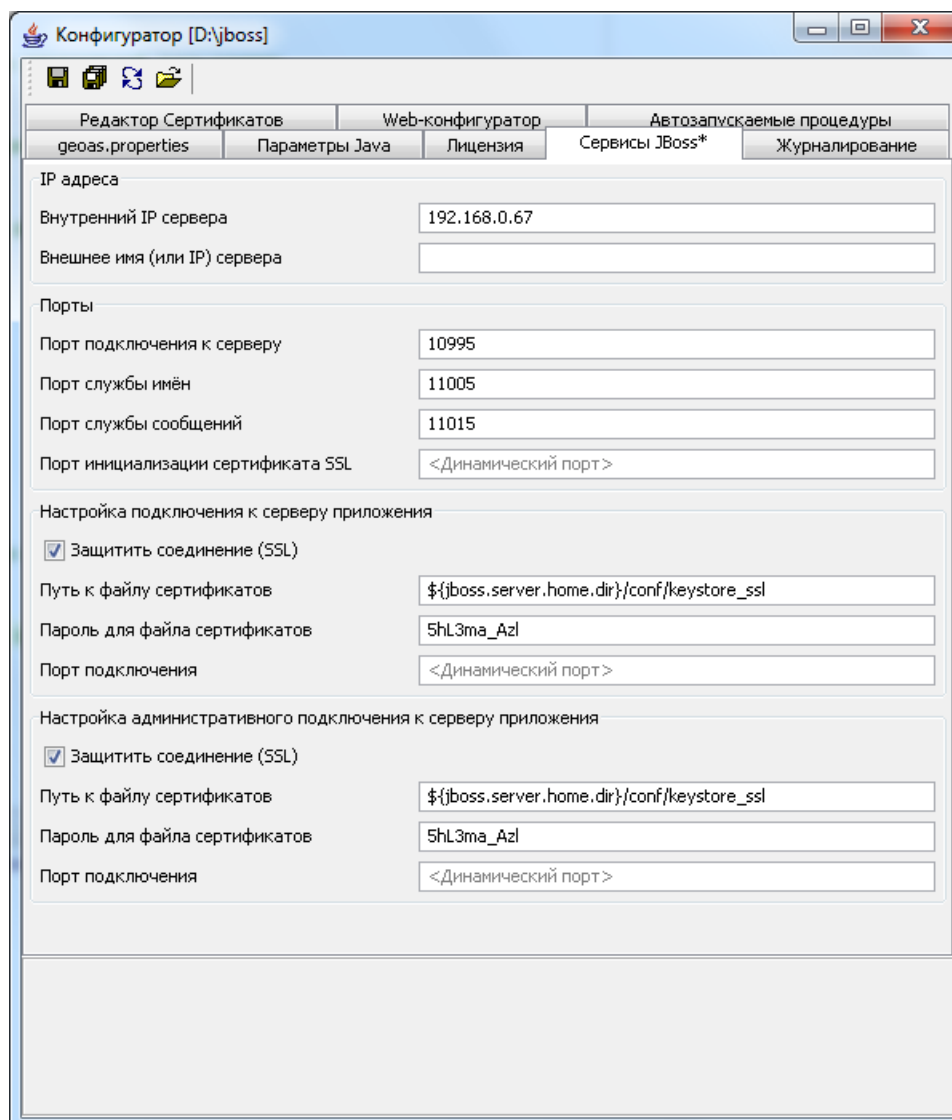
Файл лицензии располагается по пути bin/gsee.lic

На данной вкладке Конструктора производится загрузка файла лицензии на сервер, а так же можно посмотреть состояние загруженной ранее лицензии.



## Параметры сервисов

На данной вкладке настраиваются параметры подключения к различным службам сервера: адреса, порты, защита соединения.



В первую очередь необходимо задать **«внутренний IP»**, т.е. адрес адаптера (сетевой карты), на котором сервер будет открывать порты для подключения. Если этот адрес не задать, будет использоваться 127.0.0.1, т.е. сервер будет доступен только в пределах того же компьютера.

В случае, когда в сети используется преобразование сетевых адресов (NAT, «маскарадинг»), и сервер недоступен для подключения по тому же «внутреннему» адресу, следует задать **«внешнее имя»**, т.е. DNS-имя или IP-адрес, который будут вводить пользователи в диалоге подключения и строке URL браузера. Если используется несколько подсетей, в них всех сервер должен иметь одинаковое имя.

Обратите внимание: при указании в настройках внешнего имени сервер перестанет работать по «внутреннему IP», даже при подключении из той же подсети. Т.е. в таком случае в подсети сервера служба DNS должна быть сконфигурирована так, чтобы сервер был виден с других компьютеров по внешнему имени.

Оба этих параметра сохраняются в файле `run.vmoptions`: внутренний IP в виде строки типа `-Djboss.bind.address=192.168.0.67`, внешнее имя в виде строки типа `-Djava.rmi.server.hostname=gsee_server`. Кроме того, внутренний IP прописывается в файле `shutdown.appoptions` в его единственной строке вида `--server=jnp://192.168.0.67:1099`.



При подключении клиента к серверу для разных нужд используются несколько сетевых портов. Часть из них задается динамически, часть – конфигурируется в конкретные значения. Все их можно менять, подстраиваясь под конкретную конфигурацию сети. Чаще всего отключать динамическую конфигурацию требуется при жестких ограничениях на открытые порты.

**Порт подключения к серверу** клиент-приложения (служба имен JNDI). Используется только для начального подключения. По умолчанию 1099, не может быть динамическим. Если он меняется, пользователь должен указать его явно в диалоге подключения после имени сервера: сервер:порт. Порт конфигурируется в файле `server/default/conf/jboss-service.xml`, для службы `org.jboss.naming.NamingService` (соответствующее место проще всего найти по этому имени), атрибут `Port`.

**Порт службы имен**. Используется при работе для подключения к различным службам по их именам (служба JNDI – Java Naming Directory). По умолчанию динамический. Порт конфигурируется в том же файле при помощи атрибута `RmiPort` (рядом с предыдущим).

**Порт службы сообщений (JMS)**. Через это соединение сервер может асинхронно посылать команды приложению; это используется, например, при работе процедур. По умолчанию динамический. Конфигурируется в файле `server/default/deploy/jms/ui2-service.xml` параметром `ServerBindPort`.

**Порт инициализации сертификата SSL**. Используется в случае защищенного соединения для получения сертификата сервера. По умолчанию динамический. Конфигурируется в файле `server/default/deploy/wc-cert-service.xml`, тем же параметром `serverBindPort`.

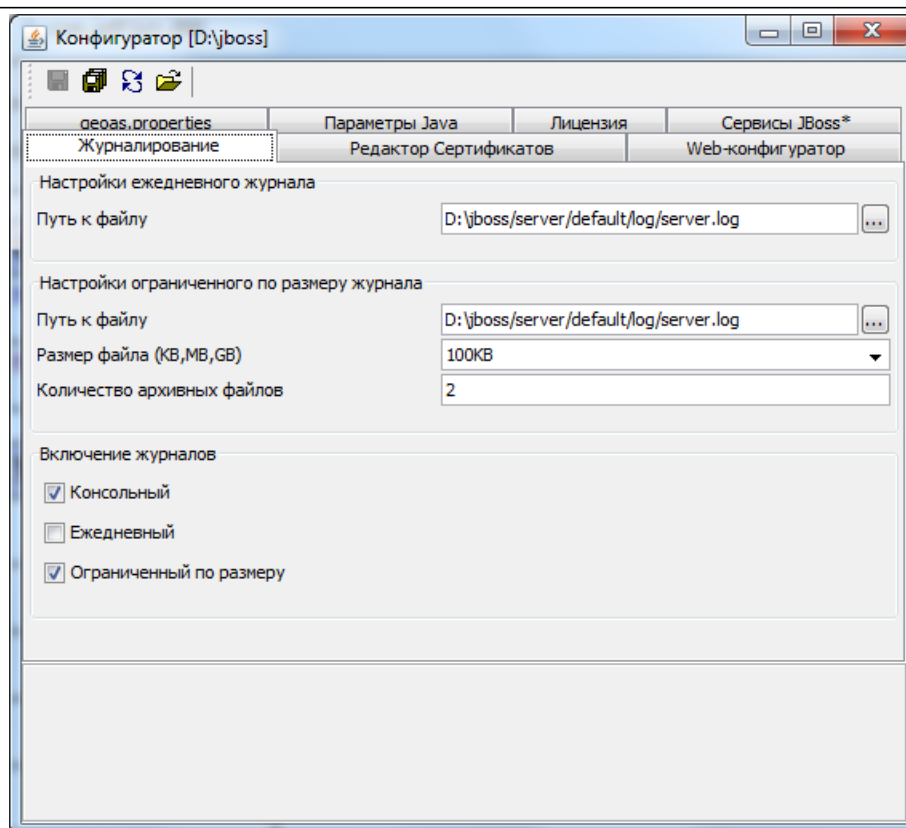
**Порт передачи данных**. Используется для «поточной» передачи данных, в первую очередь при чтении и сохранении файлов у пользователя процедурами. По умолчанию динамический. Конфигурируется в файле `bin/run.vmoptions` строкой `-Dcom.healthmarketscience.rmiio.exporter.port=12345`.

**Порты обмена данными с сервером** (служба удаленных вызовов RMI), отдельно для всех клиент-приложений, и отдельно для Администратора. По умолчанию динамические, зашифрованные протоколом SSL с указанными там же сертификатами. Конфигурируются в файлах `wc-admin-service.xml` (для Администратора сервера приложений) и `wc-kernel-service.xml` (для всех остальных применений), находящихся в каталоге `server/default/deploy` (см. закомментированный параметр `serverBindPort`). Порты в этих файлах могут не совпадать, т.е. для Администратора можно настроить отдельное подключение, например, в целях безопасности.

## **Параметры ведения логов (журналирования)**

Во время работы сервера все его диагностические сообщения (информация, ошибки и т.п.) пишутся в файл протокола. Этот файл может понадобиться администратору или разработчику при настройке сервера, устранении ошибок и т.п.

Для ведения протокола на сервере используется широко известная служба `log4j`, ее настройки хранятся в файле `server/default/conf/jboss-log4j.xml`.



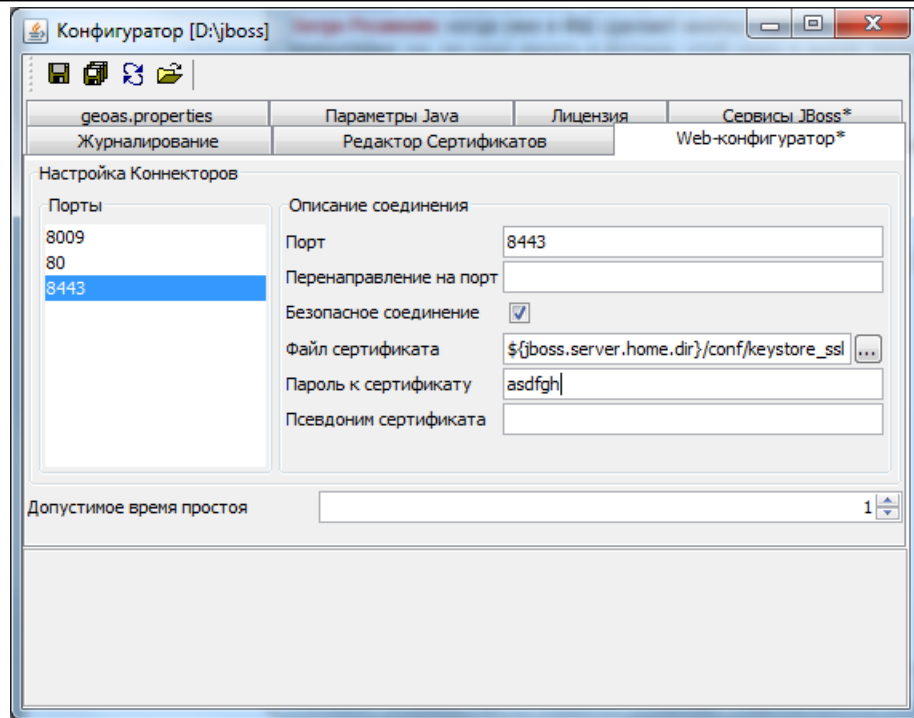
В службе журнала для вывода сообщений куда-либо используются т.н. appender-ы. Три таких объекта специально созданы и могут быть настроены Конфигуратором: консольный (вывод в консоль сервера), ежедневный (по одному файлу лога в день) и ограниченный по размеру.

Особенность последнего из трёх упомянутых appender-ов заключается в том, что он может удалять старые файлы сам, по мере появления новых. Настраивается это соответствующими параметрами в той же вкладке Конфигуратора. К сожалению, «ежедневный» лишен такой возможности и при его использовании надо следить за накоплением файлов логов.

## Параметры HTTP-подключения и сервлетов

Эти настройки делятся на две части: настройка модуля Web-сервера и настройка работающих в нем сервлетов.

Сервис Web-сервера (Apache Tomcat) конфигурируется файлом `server/default/deploy/jboss-web.deployer/server.xml`. Нас в этом файле в первую очередь интересуют настройки коннекторов, т.е. модулей, принимающих входящие запросы.



Обычно настроено два коннектора: открытое соединение по протоколу HTTP, и защищенное – по HTTPS. Отдельная настройка сервлета (см. ниже) может потребовать принудительного переключения с открытого соединения на защищенное.

Каждый коннектор, в первую очередь, определяется портом, который он слушает (соответственно, порт не может быть динамическим).

Для подключения Web-клиентов по протоколу HTTP создан коннектор с портом по умолчанию 80 (стандартный HTTP). Если он меняется, пользователь должен явно его указывать в URL сервера: `http://сервер:порт/`.

Для защищенного соединения используется второй коннектор с портом по умолчанию 443 (стандартный HTTPS). В принципе, пользователь может сразу соединяться по этому протоколу, указав в начальном URL сервера протокол `https://`. Но чаще всего используется (и по умолчанию так и настроено) перенаправление с HTTP-коннектора, для чего в нем настроен соответствующий параметр (Перенаправление на порт).

Для того, чтобы HTTPS функционировал, в коннекторе требуется указать файл сертификата и пароль к нему.

Внутри каталога `server/default/deploy/jboss-web.deployer/` размещено большое количество других настроечных файлов, описанных в документации на Tomcat. В Конфигуратор вынесена только настройка тайм-аута сессии (в минутах). При эксплуатации веб-интерфейса в качестве справочной системы может иметь смысл задать меньшее значение.

Сервлеты, непосредственно отдающие данные пользователю, располагаются в специальных подкаталогах каталога `server/default/deploy/`. В имени такого подкаталога добавлено окончание `.WAR` (Web Archive). Если сервлет требует безопасного подключения, то в нем в файле `WEB-INF/web.xml` должна быть указана настройка

```
<user-data-constraint>
```

```
  <transport-guarantee>CONFIDENTIAL</transport-guarantee>
```

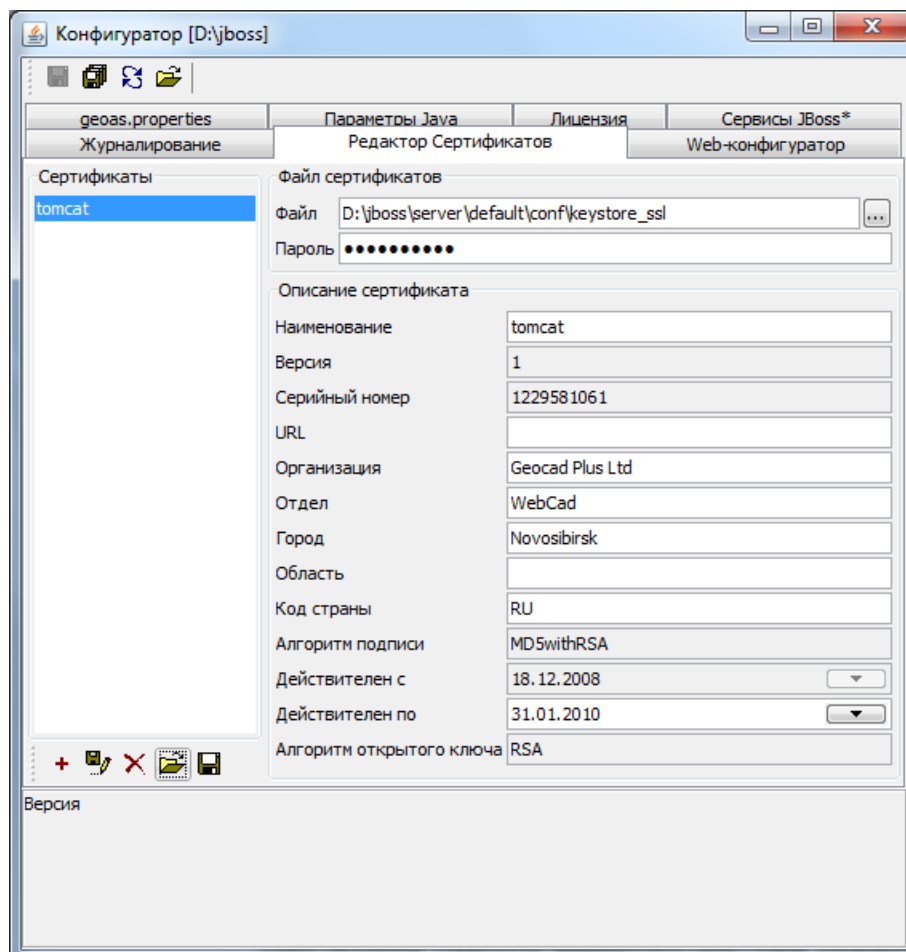
```
</user-data-constraint>
```

Для отключения можно вместо `CONFIDENTIAL` написать `NONE`, тогда при подключении по открытому соединению не будет происходить переключение на безопасное.

## Редактор сертификатов

Для использования шифрования SSL, как для RMI-подключения, так и для веба, требуются сертификаты. По умолчанию в комплекте инсталлятора идет файл сертификата, выданный разработчиками самими себе. Такой сертификат хотя и позволяет осуществлять полноценное шифрование данных, не признается браузерами в качестве надежного и имеет ограниченный срок действия.

Рекомендуется при запуске GSEE в промышленную эксплуатацию получить сертификат из надежных источников, или хотя бы, пользуясь соответствующим функционалом Конфигуратора, выдать его на организацию, эксплуатирующую сервер.



## Настройка пароля администратора сервера приложений

Администратор сервера приложений – это специальное отдельное приложение, служащее для минимальной начальной настройки сервера, достаточной для последующего запуска Конструктора (Конструктор не может подключиться к серверу при отсутствии корректных структур базы данных метамодели (БДММ) т.к. использует систему безопасности, основанную на этих структурах). Под этой настройкой понимается создание и обновление структур БДММ.

Администратор сервера использует более простую систему безопасности, основанную не на базах данных, а на простых текстовых файлах. В файле `server/default/conf/props/users.properties` находится имя и пароль пользователя, имеющего право запустить приложение. После установки там записан `super` и `super` соответственно, и эти значения можно изменить в любом текстовом редакторе. Более того, после создания структур этот файл вообще можно удалить, отключив возможность запустить Администратор.

Все изменения вступают в силу после перезапуска сервера.

## Управление контейнером бинов

Не вдаваясь в технические подробности, можно сказать, что бин – это объект, выполняющий на сервере запросы клиента (получение данных, генерация отчетов, выполнение процедур и т.п.). Для каждого клиента в момент работы создаются несколько разных бинов, которые содержатся в контейнерах, рассортированные по типу. Если клиент долгое время не обращается к серверу (например, пользователь отошел на обед), его бины могут быть переведены в пассивное состояние, т. е. выгружены из памяти на диск, а по прошествии большого времени – вообще удалены.

Контейнеры бинов настраиваются в файле `server/default/deploy/ejb3-interceptors-aop.xml`. На данный момент используются только конфигурации **“Stateful Bean”** и производная от нее **“LongPassivation Stateful Bean”**. Причем все, что имеет смысл настраивать, это конфигурация кэша: время пассивизации и удаления бина.

Найти конфигурацию проще всего по имени, записанному в атрибуте тега `<domain>`. Кэш контейнера описывается в строке, начинающейся с `@org.jboss.annotation.ejb.cache.simple.CacheConfig`, в частности это параметры:

- **`idleTimeoutSeconds`** – время неактивного состояния бина, после которого он может быть пассивизирован, в секундах; имеет смысл сокращать при большой нагрузке на сервер;
- **`removalTimeoutSeconds`** – время неактивного состояния бина, после которого он может быть удален, в секундах; это защита от неправильно работающих клиент-приложений, разрывов сети и т.п. – подчистка ресурсов;

Для кэша типа `LongPassivation` параметр `idleTimeoutSeconds` установлен в значение, большее, чем `removalTimeoutSeconds`, т.е. в нем не производится пассивизации, только удаление. Категорически не рекомендуется сменять такое поведение кэша в этой конфигурации, т.к. такой контейнер используется для бинов, которые по той или иной причине невозможно перевести в пассивное состояние.

Все изменения вступают в силу после перезапуска сервера.

## Подключение метамодели

### Введение

В настоящее время ММ в GSEE может храниться только в реляционной СУБД с подключением по JDBC. База данных метамодели (БДММ) состоит из порядка 40 таблиц, без ее настройки невозможно функционирование сервера.

Для подключения к СУБД используется стандартный механизм управления источниками данных GSEE, с возможностью отслеживать состояние подключения, выполняемые запросы и т.д. Есть возможность использовать несколько метамodelей одновременно.

### Подключение нескольких метамodelей

Конфигурация метамodelей находится в файле `server\default\conf\metamodels.xml`. Его формат на примере:

```
<?xml version="1.0" encoding="windows-1251"?>
<METAMODELS>
  <METAMODEL>
    <DBMS>MySQL</DBMS> <!-- Имя БД -->
    <NAME>mysql</NAME> <!-- Уникальное техническое имя-->
    <DESCRIPTION>MySQL model</DESCRIPTION>
    <URL>jdbc:mysql://localhost:33306/gee_meta</URL>
    <SCHEMA>gee_meta</SCHEMA>
    <USER>root</USER> <!-- имя -->
    <PASSWORD>root</PASSWORD> <!-- пароль -->
  </METAMODEL>
  <METAMODEL>
    <DBMS>Oracle</DBMS>
    <NAME>oracle</NAME>
    <DESCRIPTION>Oracle model</DESCRIPTION>
    <URL>jdbc:oracle:thin:tomsk_meta/tomskmeta@geocad_oracle:1521:sati</URL>
    <SCHEMA>gee_meta</SCHEMA>
  </METAMODEL>
</METAMODELS>
```

```
</METAMODEL>  
</METAMODELS>
```

Внутри элемента NAME записывается т.н. техническое имя, используемое программно для обращения к модели, а так же для запоминания клиент-приложением модели, к которой выполнялось подключение последний раз. В элементе DESCRIPTION записывается название модели, которое увидит пользователь при подключении. Возможно указание нескольких описаний для разных локалей. Логин и пароль могут быть заданы в URL.

## Инициализация структур метамодели

При инициализации структур метамодели происходит создание пустых таблиц в базе данных и заполнение начальной правовой информации – создание пользователя с административными правами. Структуры будут созданы в базе данных, на которую настроен сервер приложений.

Поскольку информация о правах так же содержится в метамодели, данный процесс не может быть проведен из Конструктора, т.к. он использует ее для авторизации. Поэтому в состав пакета управления входит приложение Администратор сервера, использующее другую систему авторизации, не применимую для иных задач, но достаточную для инициализации метамодели.

После установки сервера логин и пароль для Администратора по умолчанию устанавливаются в super/super.

Перед инициализацией структур внимательно посмотрите параметры подключения внизу окна, убедитесь, что это именно та БД, где будет находиться новая метамодель. Для запуска инициализации нажмите кнопку «Создать метамодель». Потребуется ввести имя и пароль пользователя с административными правами, который будет создан после инициализации структур. Этот пользователь после будет использоваться для работы в Конструкторе.

Инициализация структур происходит в течение несколько секунд.

## Установка клиент-приложений в Windows

### Состав приложений

Приложения можно разделить на две группы - пользовательские и административные.

В комплект пользовательских приложений входят:

- GSEE InfoForms Pro (клиент для доступа и редактирования семантической информации);
- GSEE Maps Pro (просмотр и редактирование метрической информации в виде карт);
- GSEE Logic Reports (система генерации и печати отчетов).

Основным приложением является InfoForms Pro, остальные запускаются из него. Таким образом инсталлятор создает только один ярлык — для запуска InfoForms Pro.

В комплект административных приложений входят:

- GSEE Конфигуратор (настройка служб, портов, подключений сервера)
- GSEE Администратор (подготовка работе метамодели и некоторые другие операции);
- GSEE Конструктор (управление сервером и метамоделью).

Эти приложения полностью независимы друг от друга, и для каждого из них создается свой ярлык.

### Общие сведения

Все клиент-приложения, за исключением Конфигуратора, устанавливаются и обновляются с использованием технологии Java WebStart и Java Network Launching Protocol. По этой технологии самый свежий, актуальный и соответствующий версии сервера программный код приложений хранится на сервере и автоматически скачивается и обновляется при запуске приложения пользователем.

Инсталлятор приложений устанавливает исполнимую среду Java Runtime Environment,

включающую в себя модуль WebStart, и создает ярлыки для запуска приложений. Собственно код приложения скачивается с сервера при первом запуске. В дальнейшем при установке обновлений на сервер, одновременно обновляется и репозиторий кода клиент-приложений. При очередном запуске на пользовательском компьютере модуль Java WebStart автоматически, без участия пользователя докачивает изменившиеся модули.

Эта технология гарантирует, что пользователь работает с самой последней версией приложения, доступной для данного сервера и соответствующей его функционалу.

## Установка приложений

Для установки пользовательских и административных приложений GSEE в операционной среде MS Windows используется единый инсталлятор. Он разворачивает в указанной папке среду исполнения Java Runtime Environment, создает несколько небольших исполняемых файлов для запуска модуля WebStart, и ярлыки для них в стандартном меню Пуск.

В процессе установки будет предложено ввести адрес (URL и порт протокола HTTP) сервера, откуда будут загружаться и обновляться клиент-приложения. Обычно это тот же сервер, на котором происходит работа с данными. Может быть установлено несколько клиентов для разных серверов; в таком случае допускается установка в одну папку на диске и в одно меню кнопки Пуск.

Немного особняком стоит код приложения Конфигуратор. Поскольку Конфигуратор работает с выключенным сервером, его код невозможно получить по технологии Java WebStart, так что это единственное приложение, чьи файлы устанавливаются непосредственно инсталлятором. По этой же причине, это приложение никогда не обновляется автоматически.

Помимо традиционного способа распространения инсталлятора в комплекте поставки, его всегда можно скачать с работающего сервера по ссылке в браузере: [http://сервер/wc\\_client/](http://сервер/wc_client/), где под «сервер» понимается все тот же адрес сервера, с которым будет позже работать пользователь. По этому адресу находится небольшая веб-страница, на которой помимо актуальных сведений об установке находится ссылка на инсталлятор.

## Технические подробности

Серверная часть системы установки представляет собой специальный сервлет, обрабатывающий запросы модуля Java WebStart. Как и другие сервлеты, он работает под управлением веб-сервера Tomcat и поэтому доступен по порту, на который настроен веб-сервер (протокол JNLP является надстройкой над протоколом HTTP).

Инсталлятор создает в указанной папке для каждого выбранного клиент-приложения пакетный (batch) файл. Например, для InfoForms Pro это будет `infoforms URL.bat`, где URL - адрес сервера, введенный при установке. Содержимое этого файла примерно таково:

```
jre\bin\javaws -Xmx256m "-Dlogfile=infoforms.log" http://url/wc_client/infoforms.jnlp
```

В середине строки задаются параметры запуска Java-машины, в данном примере - количество выделяемой памяти (256 мегабайт) и имя файла отладочного протокола работы. В конце строки указан URL приложения. В данный момент приложения доступны по следующим URL:

GSEE InfoForms Pro: `infoforms.jnlp`

GSEE Constructor: `constructor.jnlp`

GSEE Admin: `admin.jnlp`

В принципе, администратор может сам создать такой файл и настроить запуск, без использования инсталлятора.

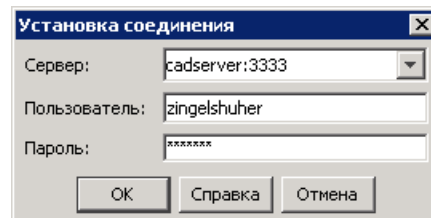
При работе в папке установки будет создана подпапка `cache`, куда помещаются скачанные с сервера программные модули. В случае каких-либо сбоев обновления администратор может удалять эту папку, тогда при следующем запуске приложение полностью скачает себя заново.

## Запуск приложений

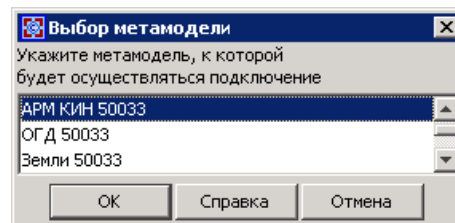
Запуск различных приложений системы осуществляется обычным образом при помощи созданных в процессе установки ярлыков.

В процессе запуска в диалоге подключения к серверу необходимо указать адрес сервера (имя или IP адрес), имя пользователя и пароль. В имени сервера при необходимости надо указать порт. Например: cadserver.mywork.ru или cadserver:3333.

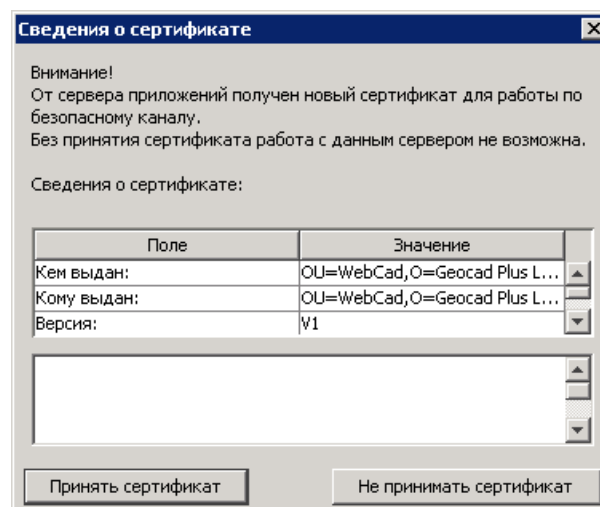
Эти параметры, кроме пароля, автоматически сохраняются после первого запуска и при последующих запусках приложения будут заполнены.



Если на сервере приложения одновременно обрабатывается несколько моделей данных, то будет выдан список, из которого нужно выбрать нужную:



В случае работы с сервером GSEE по защищенному каналу с использованием шифрования SSL при **первом запуске** приложения будет предложено принять и установить сертификат SSL.



## Установка обновлений

### Обновление программного обеспечения

#### Поставка и порядок установки

Обновление ПО предоставляется в виде специальным образом подготовленного приложения, содержащего в себе файлы сервера, требующие обновления. Обновляются как чисто серверные компоненты (библиотеки ядра, процедуры, драйверы СУБД), так и веб-приложения (в т.ч. WebLight и модули WFS/WMS), и автообновляемые по технологии WebStart клиент-приложения.

Обновление ПО распространяется в виде архива, который необходимо полностью распаковать перед запуском. В архиве содержится Java-приложение, которое выполняет все необходимые



действия, и набор файлов для обновления. Архив может быть распакован как на компьютере оператора, так и непосредственно на сервере, если там есть графический интерфейс. Приложение требует прямого доступа к файлам сервера.

В среде Windows запуск производится небольшим приложением `setup.exe`. Оно автоматически находит JRE, которая была установлена либо с клиентскими приложениями, либо с сервером (в зависимости от того, где обновление запускается) и уже в ней запускает собственно обновление.

На сервере в среде Linux обновление следует распаковывать в подкаталог относительно корневого каталога установки, тогда скрипт `setup.sh` сможет автоматически найти ту же JRE, в которой запускается сервер.

В архиве с обновлением может так же располагаться файл `readme.doc`, содержащий список изменений, исправленных ошибок и т.п., которые содержит данное обновление. В случае каких-либо особых ситуаций, требующих ручного вмешательства администратора, разработчики могут добавлять в это описание специальные инструкции, которые нужно будет выполнить до, после или во время обновления.

Так же могут прилагаться инструкции, рекомендуемые сразу после установки обновления проверить тот или иной функционал, чтобы в случае непредвиденных ошибок внести дополнительные изменения оперативно. В любом случае, если обновление выполняется для рабочего сервера, время его лучше согласовать с разработчиками, для скорейшего решения нестандартных ситуаций.

Перед обновлением сервер необходимо остановить обычным образом. В данный момент приложение обновления не проверяет состояние сервера, эта задача возлагается на администратора. Желательно так же создать копию файлов сервера, чтобы при возникновении нестандартных ситуаций быстро отменить изменения и вернуть систему к рабочему виду.

На первом шаге обновлению необходимо указать каталог, куда установлен сервер. При этом выполняются простейшие проверки, подтверждающие правильность указанного каталога, а так же проверяется совпадение версии сервера и обновления. В случае несовпадений выводится предупреждение, однако обновление не запрещается.

Версия сервера – это целое число, увеличивающееся по мере выхода новых версий. Технически – это уникальный номер версии репозитория Subversion разработчиков на момент сборки инсталлятора, поэтому увеличиваться оно может очень нелинейно.

Поскольку обновление выполняет достаточно тривиальные операции, в основном связанные с перезаписью файлов целиком, иногда (по согласованию с разработчиками) можно ставить обновление на версию сервера, для которой оно не рассчитано, в том числе и «в обратную сторону», т.е. откатывать на предыдущую.

На следующем этапе происходит собственно установка обновленных файлов. Обычно это занимает немного времени, после чего сервер готов к запуску.

## **Обновление клиентов**

Если обновление ПО затрагивает клиентскую часть комплекса, то при первом запуске пользователем клиент-приложения на его компьютер автоматически будут загружены все необходимые обновления. Размер скачиваемых данных как правило не превышает нескольких мегабайт, однако может иметь смысл предупредить пользователей о появлении дополнительных окон и сообщений при первом запуске приложения.

Администратору рекомендуется сразу после установки обновления и запуска сервера в работу зайти всеми приложениями (Forms Pro, Конструктор) на сервер. Никаких работ выполнять не обязательно, достаточно факта подключения – оно будет свидетельствовать, что обновление клиентского ПО произошло успешно.

## **Обновление структур метамодели**

Периодически внесение нового функционала требует изменения структур метамодели. Если такая операция требуется, это обязательно будет упомянуто в сопроводительной документации к обновлению. Изменение структур выполняется из приложения Администратор сразу после первого

запуска обновленного сервера. Перезапуск сервера после этой операции не требуется.

Обновление структур может создать новые таблицы в базе данных метамодели, добавить поля к существующим таблицам, переместить данные между ними. В любом случае оно выполняется в автоматическом режиме и редко требует вмешательства оператора. Рекомендуется предварительно сделать копию этой базы данных средствами СУБД.

В коде сервера есть простая защита, не позволяющая ему работать на неподходящей версии: версия метамодели записана в единственной записи таблицы GEE\_VERSION, и проверяется при первом обращении пользователя к серверу. В случае несовпадения об этом будет сказано пользователю, с просьбой обратиться к администратору. Таким образом, в принципе, невозможно забыть или пропустить эту операцию.

Внесение изменений в структуру метамодели выполняется по возможности не деструктивно, т.е. в случае нештатных ситуаций и отката на предыдущую версию серверного ПО, достаточно бывает изменить версию в таблице GEE\_VERSION на предыдущую, чтобы старый сервер не отказался с ней работать. В любом случае, рекомендуется такую ситуацию согласовывать с разработчиками.

## Обновление метамодели

### Общие сведения

Обновление метамодели предоставляется в виде специальным образом сформированного файла в формате XML. Основная операция при обновлении – загрузка этого файла при помощи соответствующих команд Конструктора. В зависимости от объема и содержания обновления может потребоваться выполнение разного объема ручных работ. К каждому обновлению прилагается подробный список операций, которые необходимо выполнить администратору для полного и корректного применения изменений.

До и после выполнения обновления нет необходимости останавливать и перезапускать сервер, необходимо лишь проследить, чтобы в момент обновления к нему не подключались пользователи.

Перед обновлением рекомендуется сделать копию базы метамодели средствами СУБД или Конструктора.

Документация, прилагаемая к обновлению метамодели, должна содержать следующие сведения:

- список новых типов объектов, с указанием наименования каждого объекта, имени его таблицы в банке данных, отметкой, является ли он классификатором, необязательным примечанием;
- список измененных типов объектов, с указанием наименования каждого объекта, имени его таблицы в банке данных, подробным описанием изменений;
- список измененных классификаторов;
- список новых и измененных отчетов и процедур, с примечаниями;
- порядок установки обновления, со ссылками на данный регламент;
- рекомендации по первичной проверке работоспособности после обновления.

### Порядок обновления

Первое, что крайне желательно сделать перед обновлением - сделать бэкап метамодели. Сделать его можно как средствами СУБД, так и с помощью команды Конструктора **Файл - Сохранить метамодель с банками**. Во втором случае будет создан XML-файл, который в случае необходимости откатить изменения можно загрузить обратно командой **Файл - загрузить метамодель**.

Загрузка метамодели выполняется в Конструкторе командой **Файл - Загрузить метамодель**. Потребуется указать XML-файл, переданный разработчиками.

В случае, если в обновлении передавались новые таблицы и/или слои, требуется выполнить обновление описания банков: в Конструкторе командой **Инструменты – Проверка связей таблиц и слоев с банком данных**. В открывшемся диалоге нужно указать банк для проверки и нажать кнопку **Проверить**. По результатам проверки будет показан диалог со списком новых и измененных

объектов. «Новые» – это объекты, которые были добавлены при обновлении метамодели, «Измененные» – которые были перемещены между картриджами. В списке «Лишние» объектов быть не должно, т.к. не описанные в модели (и, следовательно, в эталонном банке) не будут загружены на этапе восстановления описания банков.

Если список найденных изменений соответствует документации на обновление, их следует внести нажатием кнопки **Исправить все**. Несоответствия могут возникнуть, если в предыдущую версию модели или банка были внесены локальные изменения. Если же список содержит неожиданные значения, рекомендуется обратиться к разработчикам.

В общем случае администратор может выполнить всю эту операцию вручную, пользуясь описанием изменений метамодели, полученным от разработчика. По сути надо добавить в соответствующие картриджи банка новые таблицы и слои, а в соответствующие аспекты - новые объекты.

Права на все вновь созданные объекты не раздаются автоматически ни одной роли. Раздача прав на них выполняется штатным образом в Администраторе безопасности.

Обновление описания банков не влияет на структуры банков в СУБД, т.е. если в обновлении передавались новые таблицы и/или слои после обновления описания требует обновить и физические структуры. Для этого служит команда Конструктора **Операции – Создать структуры банка данных** (и другие операцией из той же группы). Кроме того, в случае нетривиальных изменений (перенос полей между таблицами, сложные модификации данных) к обновлению может быть приложено описание действий (текстом или в виде запросов SQL), которые необходимо выполнить администратору банка данными средствами СУБД.

Так же в виде запросов SQL (либо в виде дампа, по договоренности с администратором СУБД) поставляются обновления классификаторов. Поскольку локальная правка классификаторов не предусматривается, администратору высылается полное содержимое не только новых, но и обновляемых таблиц.

## Полное обновление

В некоторых случаях обновление метамодели может поставляться не в виде обновления, а в виде полных структур. Грубо процесс установки такого обновления можно разделить на следующие этапы: сохранение настроек банков, пользователей и прав; сохранение форм WebLight, очистка структур метамодели; закичивание структур, полученных в составе обновления; внесение сохраненных настроек; правка настроек; обновление структур банка данных и раздача прав на новые объекты.

Естественно, перед началом работ желательно сделать полный бэкап метамодели.

Помимо полного бэкапа обязательно понадобится сделать сохранение настроек банков, пользователей и прав командой **Файл – Сохранить банки и права**. Данные сохраняются в XML-файл. Этот файл понадобится далее для восстановления настроек в новой модели.

Если в модели существуют созданные и настроенные формы приложения WebLight, следует сохранить их командой **Файл – Сохранить формы WebLight**. Данные сохраняются еще в один XML-файл.

На этом этап сохранения бэкапов завершается. Следует закрыть Конструктор и запустить приложение Администратор.

В приложении Администратор выполняем команду **Создать структуры метамодели**. Фактически в этот момент происходит такая же инициализация структур, как и при первой установке. На запрос имени и пароля администратора желательно ввести те же данные, что и были ранее.

Далее следует закрыть Администратор, заново запустить Конструктор и штатным образом (**Файл - загрузить метамодель**) загрузить файл, полученный от разработчиков.

Далее командой **Файл – Загрузить банки и права** выполняется восстановление описания банков, пользователей и их прав выполняется. Указывается файл, сохраненный ранее. Операция старается по возможности восстановить принадлежность таблиц и слоев ранее созданному банку,

вернуть настройки картриджей, права ролей на эти объекты. Все новые, а так же переименованные объекты найдены не будут. Может понадобится перезапустить Конструктор, чтобы применились измененные права администратора.

При наличии архива форм WebLight восстанавливаем их командой **Файл - Загрузить формы WebLight**. Восстановлены будут только формы для ранее существующих объектов. Для всех новых объектов формы при необходимости нужно создать стандартным образом.

Далее выполняются все те же этапы, что и при обычном, частичном обновлении: обновление описания банка данных, обновление структур, раздача прав на новые объекты, установка новых классификаторов.

**Конструктор**

**Часть**

---



## Глава III: Конструктор

### Введение

#### Назначение программы

Приложение Конструктор предназначено для выполнения большинства работ, связанных с настройкой и обслуживанием сервера приложений GSEE:

- создание и изменение метамодели, включая отчеты и процедуры;
- создание структур банков данных и поддержание их соответствия метамодели;
- управление системой безопасности
- управление ПО сервера, диагностика неисправностей.

Конструктор взаимодействует с сервером через специальные программные интерфейсы и использует его систему безопасности для разграничения доступа, соответственно, требует, чтобы сервер был запущен, а метамодель была настроена. Для начальной настройки и инициализации сервера и метамодели используются отдельные приложения Администратор и Конфигуратор; они работают с конфигурационными файлами сервера напрямую и поэтому не требуют, чтобы он был в рабочем состоянии.

Управление СУБД (включая ведение бэкапа) находится вне рамок приложения.

#### Окно программы

Конструктор использует стандартный интерфейс, в котором большая часть функционала реализуется в различных внутренних окнах. Наиболее часто используется окно редактирования метамодели, с которым, в частности приложение и запускается.

### Управление сервером

#### Введение

Диалог **Управление сервером** предназначен для просмотра и изменения состояния сервера в процессе работы. Он позволяет управлять процессами, подключениями, сессиями пользователей, просматривать состояние памяти, логи работы и т.п.

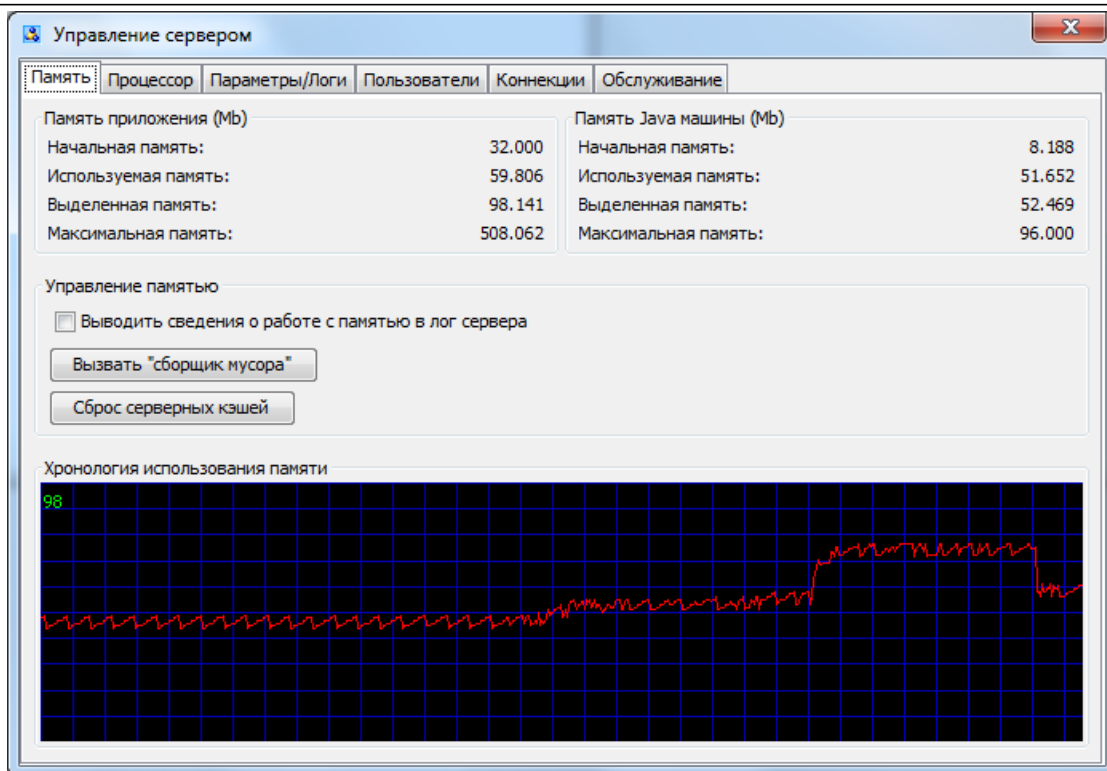
Диалог может быть открыт командой меню **Инструменты - Управление сервером** в **Конструкторе**, а так же командой меню **Сервис - Информация о сервере** в **InfoForms Pro**.

Для работы с этим инструментом требуется наличие у оператора прав читать и изменять состояние сервера.

С другой стороны, набор информации и возможные операции в данном диалоге не зависят от метамодели, в которую выполнялось подключение.

#### Память

Вкладка **Память** предназначена для просмотра и управления состоянием памяти сервера приложений.



В верхней части находятся два блока, отображающих сведения об объеме памяти, занимаемой сервером внутри Java-машины. Память в Java-машине делится на доступную приложению (в данном случае, серверу GSEE, левый блок) и на служебную (правый блок).

Рассмотрим память приложения:

- Начальная - та, которая выделяется Java при запуске сервера. Это число не меняется при работе.
- Используемая - текущий объем памяти, с которой работает ПО сервера.
- Выделенная - память, запрошенная у операционной системы. При запуске сервера она равна начальной. Когда объем используемой памяти доходит до объема выделенной, Java-машина запрашивает у ОС еще.
- Максимальная - предел памяти, который Java может запросить у операционной системы.

Эта память в основном используется сервером для кэширования различных данных с целью уменьшения обращения к СУБД и файлам. Кэш в большинстве случаев является "сбрасываемым", т.е. при необходимости давно неиспользуемые данные могут из него вытесняться, заменяться более актуальными. В кэш помещаются: метамодель, сведения о правах, графические объекты, некоторые параметры банков (например, системы координат). Семантические данные по ряду причин не кэшируются.

Требования к объему памяти постепенно растут при увеличении количества одновременно работающих пользователей. При работе с крупными растрами, при генерации больших отчетов, при работе некоторых процедур требования могут вырасти значительно.

Память Java-машины в общем случае выделяется точно так же, но используется для внутренних целей, в первую очередь для организации потоков выполнения и поэтому её расходование в основном зависит от количества сессий пользователя.

Некоторые [настройки памяти](#) доступны в Конфигураторе.

В нижней части рисуется динамический график использования памяти приложения.

С помощью данных чисел и графика можно анализировать и обнаруживать ситуации с нехваткой памяти. Для тех же целей служат и элементы управления в средней части вкладки.

Следует отметить, что в терминологии Java в "используемую" входит не только память, непосредственно задействованная приложением, но и память, которая уже была им "заброшена", но еще не обработана т.н. "сборщиком мусора" (garbage collector). Периодический вызов сборщика

заметен на графике в виде характерной "пилы". Когда свободной памяти много, сборщик работает понемногу, не обрабатывая всю "зброшенную" память. Когда же используемая память приближается к максимально доступному объему, сборщик, во-первых, начинает работать агрессивней и дольше, а во-вторых начинает зачищать и кэш. При этом падает производительность сервера, пользователями могут ощущаться характерные задержки отклика.

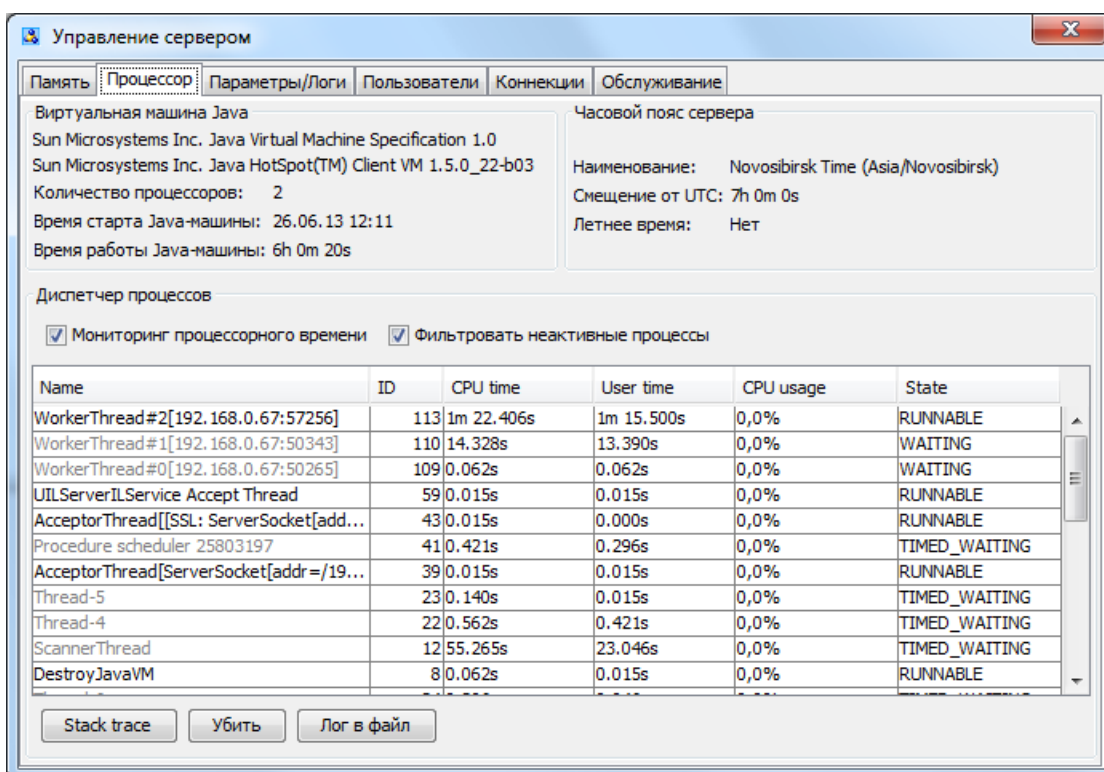
Кнопка **Вызывать "сборщик мусора"** принудительно (насколько это возможно в используемой Java-машине) вызывает очистку неиспользуемой памяти. Обычно по этой команде сборщик выполняет очистку достаточно агрессивно, или даже полностью, но в общем случае результат зависит от нагрузки на сервер, настроек сборщика и версии Java-машины.

Кнопка **Сброс серверных кэшей** вызывает принудительную очистку всех закэшированных объектов. Совместно с принудительным вызовом "сборщика мусора" её можно использовать для определения количества реально неосвобождаемой памяти в текущий момент времени. Кроме того, эта команда позволяет обновить в памяти настройки и кэшируемые данные, которые были изменены напрямую в СУБД в обход сервера, без его перезапуска.

Флажок **"Выводит сведения о работе с памятью в лог"** может пригодиться в случае тонкой настройки работы с памятью. Он включает вывод технической информации в файлы логов сервера.

## Процессор

Вкладка **Процессор** предназначена для просмотра состояния исполнительной части виртуальной машины Java, выполняемых процессов, нагрузки на процессор.



В верхней части вкладки находятся два информационных блока. Левый блок в числе всего прочего отображает наименование реализации Java-машины, что может быть полезно разработчикам при отладке. В правом блоке отображаются сведения о часовом поясе, как он настроен для сервера GSEE.

В нижней части вкладки находится модуль управления процессами.

Внутри Java-машины сервера существует довольно много процессов, как связанных с работой пользователя, так и служебных. Наблюдение за процессами может помочь выявить причины повышенной нагрузки на сервер. В таблице отображаются следующие свойства процессов:

- **Name** - наименование (см. пояснения ниже).



- **ID** - уникальный идентификатор.
- **CPU time** - общее время работы процесса.
- **User time** - полезное время работы процесса, т.е. общее за вычетом служебной нагрузки.
- **CPU usage** - примерный процент использования процессора данным процессом за последнюю секунду.
- **State** - состояние процесса.

По наименованию можно определить назначение процесса. Обычно администраторы могут заинтересовать следующие процессы:

- WorkerThread - рабочий поток, связанный с пользователем. В имени указывается IP-адрес, с которого было выполнено подключение. В общем случае рабочие потоки могут менять пользователя при длительном простое, т.е. в разное время поток с одним ID может использоваться разными пользователями, и у него, в том числе, будет меняться IP-адрес в имени. Однако при нагрузке этого обычно не происходит.
- http=<IP-адрес> - рабочий поток, связанный с веб-пользователем.
- При выполнении процедуры к имени потока в конце добавляется наименование процедуры, например, "WorkerThread#5[10.100.104.15] - Поиск смежников".

Флажок "**Мониторинг процессорного времени**" включает и выключает получение сведений о времени работы процессов. При большом количестве процессов мониторинг сам потребляет определенные ресурсы и при отсутствии необходимости его можно отключить.

Флажок "**Фильтровать неактивные процессы**" убирает из списка процессы с нулевым временем работы. Обычно это довольно большой набор системных процессов Java-машины, совершенно не нужных для наблюдения администратором.

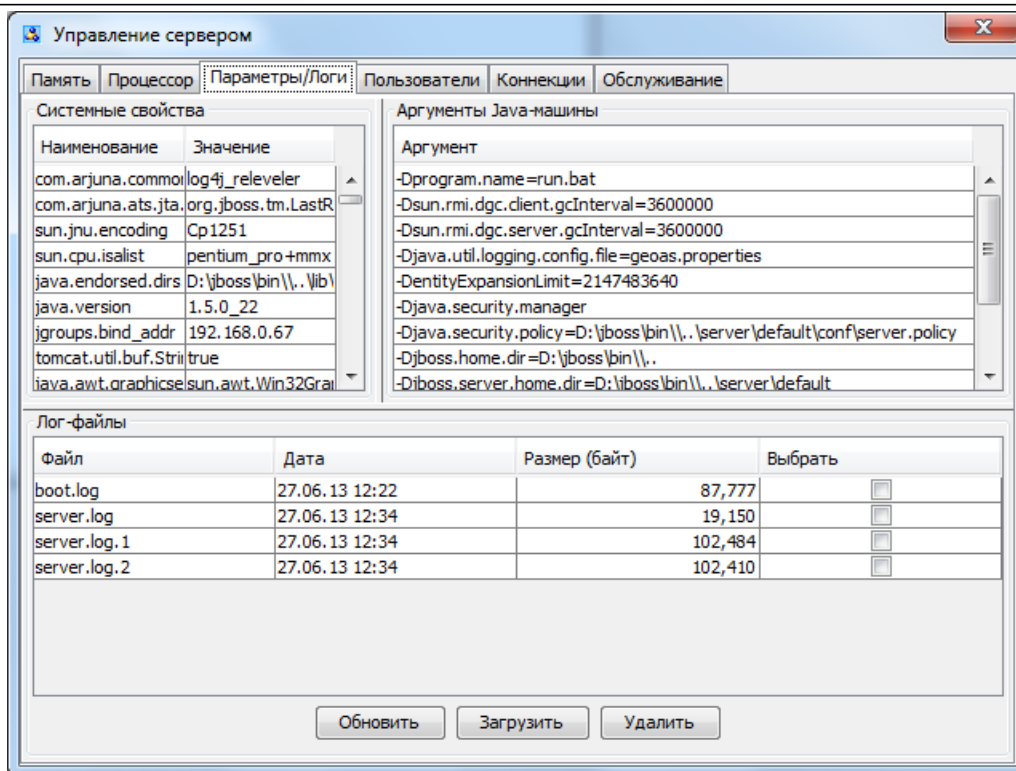
Кнопка **Stack trace** выводит т.е. "трассировку стека вызовов", т.е. информацию о текущей точке выполнения в выбранном потоке. Эта информация может быть полезна разработчикам для отладки.

Кнопка **Убить** аварийно прекращает выполнение выбранного в списке потока, что может быть полезно, если поток "завис" при выполнении.

Кнопка **Лог в файл** формирует специальный текстовый файл с информацией о всех выполняющихся потоках, что может быть полезно разработчикам для отладки.

## Параметры/логи

Вкладка **Параметры/логи** предназначена для просмотра параметров запуска виртуальной машины Java и скачивания логов работы сервера. Сведения с этой вкладки могут использоваться при отладке и поиске ошибок, как разработчиками, так и администратором.



В верхней части вкладки находятся два блока: со списком системных свойств и со списком аргументов запуска Java-машины сервера.

В нижней части находится панель загрузки файлов логов. Она может использоваться для получения файлов логов в случае, если нет прямого доступа к компьютеру с сервером. В списке отображаются все файлы из соответствующего каталога сервера приложений.

При нажатии кнопки **Загрузить** выбранные файлы будут скопированы на компьютер, где запущен Конструктор, в указанный пользователем каталог.

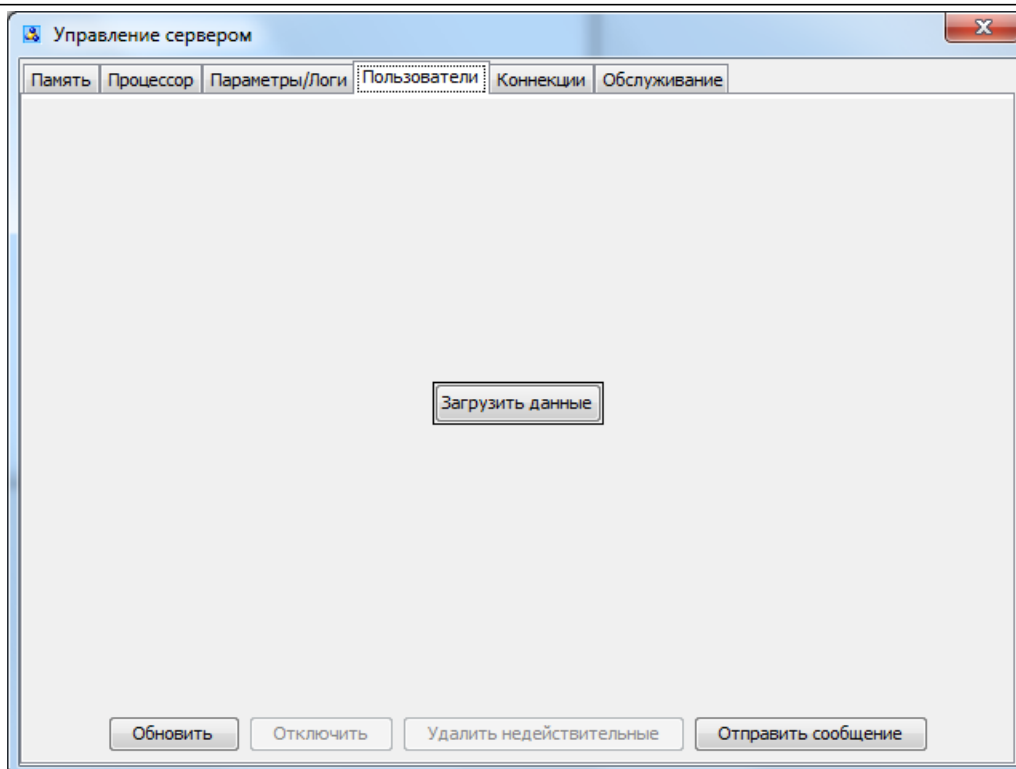
Кнопкой **Удалить** можно удалить выбранные ненужные файлы.

Настройка ведения логов осуществляется в [Конфигураторе](#).

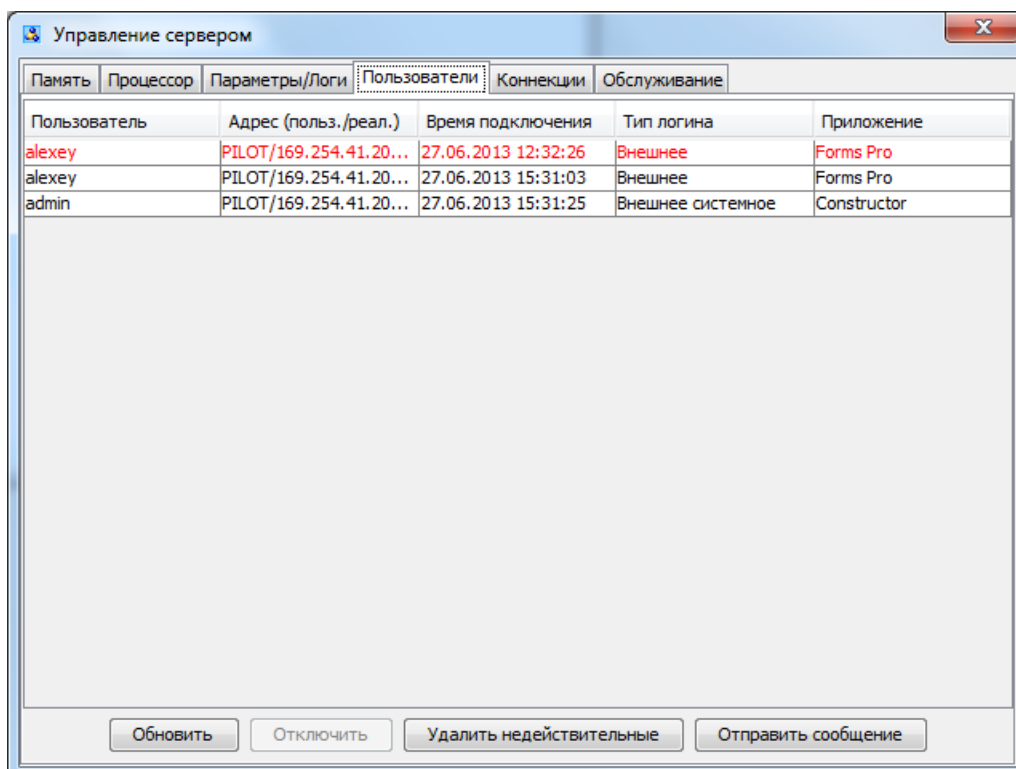
## Пользователи

Вкладка **Пользователи** предназначена для управления пользовательскими сессиями.

Сервер приложений хранит информацию о текущих пользовательских сессиях, однако не проверяет автоматически их работоспособность в период бездействия пользователя. Сессии автоматически закрываются при бездействии, однако обычно тайм-аут устанавливается достаточно большим (10 часов по умолчанию). При получении списка сессий из диалога управления сервером, сервер выполняет проверку соединения с приложениями. Это может занять достаточно длительное время, поэтому сразу при открытии вкладки Пользователи список сессий не открывается автоматически, ожидая нажатия кнопки **Загрузить данные**.



После загрузки на вкладке отображается список сессий с информацией о них.



Сессии, помеченные красным, не прошли проверку, т.е. сетевое соединение с приложением, инициировавшим такую сессию, отсутствует. В редких случаях, когда проверку осуществить не удалось (например, из-за сетевых тайм-аутов, когда за некоторое время не удалось понять, оборвано ли соединение действительно), сессии помечаются фиолетовым.

Для каждой сессии отображается несколько полей:

- **Имя пользователя**, создавшего данную сессию.
- **Два адреса компьютера**: один - как его сообщило приложение, другое - как его видит сам сервер

(могут не совпадать в некоторых конфигурациях сети, например, при использовании NAT)

- **Время подключения**, т.е. создания сессии.
- **Тип логина**. Под внешними подразумеваются соединения из приложений, например, InfoForms Pro или тот же Конструктор, под внутренними - web-сессии, т.к. такая сессия с т.з. функционирования сервера приложений целиком существует внутри его Java-машины.
- **Приложение** - имя приложения, из которого выполнен вход.

Кнопка **Обновить** перечитывает список сессий с сервера. Автоматического, через какой-либо интервал, перечитывания не происходит.

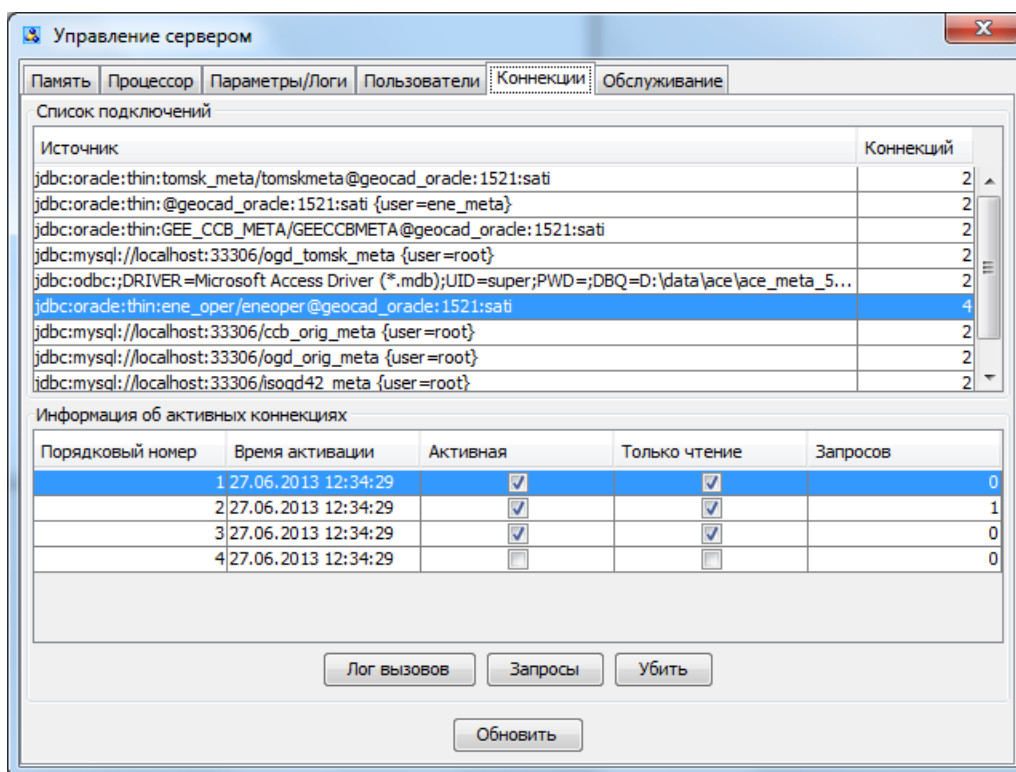
Кнопка **Отключить** принудительно закрывает выбранные сессии. При этом по возможности корректно закрываются имеющиеся соединения с базой данных. Несохранившиеся пользователем данные пропадут. Никаких сообщений об отключении приложение пользователя не получает, сессий закрывается исключительно локально, на стороне сервера.

Кнопка **Удалить недействительные** закрывает разом все сессии, для которых не удалось подтвердить существование соединения с приложением.

Кнопка **Отправить сообщение** позволяет отправить произвольный текст в приложение пользователю.

## Коннекции

Вкладка **Коннекции** предназначена для просмотра состояния соединений (коннекций) с базами данных и выполняющихся запросов.



В верхней части вкладки отображается список описаний подключений. При выборе одного из элементов списка в нижней части вкладки отображаются сведения о коннекциях, взятых по этому описанию.

В GSEE широко используется т.н. пул коннекций, когда коннекции к СУБД не закрываются после завершения использования неким процессом, а передаются для использования в другой процесс. Это позволяет не затрачивать время на инициализацию соединения (которое может быть весьма значительным). Кроме того, при отсутствии транзакций простые операции чтения данных могут вообще выполняться разными процессами одновременно на одной и той же коннекции.

Коннекции закрываются либо самой СУБД (по каким-либо причинам), либо принудительно

командой **Убить** (см. ниже). Самостоятельно сервер коннекции из пула не закрывает.

В списке коннекций отображаются следующие сведения:

- Время последней активации (т.е. взятия из пула).
- Флажок "Активная", указывающий, что данная коннекция в данный момент используется.
- Флажок "Только чтение", указывающий, что эта коннекция используется для чтения данных без транзакций.
- Количество выполняющихся в данный момент запросов на этой коннекции.

Если включено ведение лога запросов (см. [Конфигуратор](#)), то доступны команды Лог вызовов и Запросы.

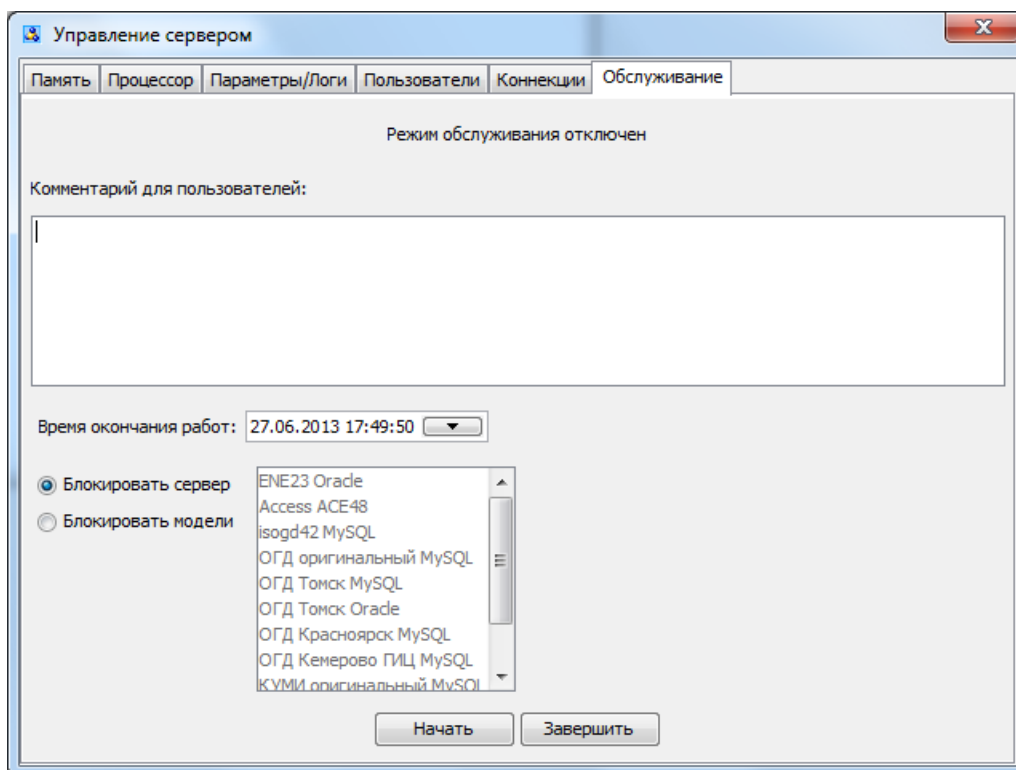
Кнопка **Лог вызовов** открывает новое окно с текстом, в котором подробно описаны все происходившие на данной коннекции операции с БД, с описанием передаваемых и возвращаемых данных, со стеком вызовов и т.п. технической информацией, которая может пригодиться разработчикам при поиске ошибок.

Кнопка **Запросы** показывает текст всех выполняющихся в данный момент запросов на данной коннекции, с информацией о сессиях пользователей, этим запрос выполняющих.

Кнопка **Убить** закрывает выбранную коннекцию и удаляет ее из пула.

## Обслуживание

Вкладка **Обслуживание** предназначена для перевода сервера в режим обслуживания, когда к нему не могут подключиться пользователи.



Режим обслуживания предназначен, в основном, для обновления метамодели или для больших изменений в правах. Обычно после перевода сервера в этот режим администратор заставляет выйти оставшихся пользователей, либо принудительно закрывает их сессии. До вывода сервера из режима обслуживания (либо до его перезагрузки) пользователи больше не могут войти в систему. Исключение составляет лишь само приложение Конструктор.

При попытке входа пользователю будет показано сообщение, заданное в этой вкладке, а так же сообщено о времени окончания работ. Строго говоря, указанное время не обязывает завершить обслуживание, оно используется лишь в сообщении пользователю.

Есть возможность блокировать как сервер целиком, так и отдельные метамодели (если их несколько).

## Диагностика различных неполадок

### Периодическое подтормаживание при обращении к серверу

#### Жалоба

Пример жалобы от пользователей: все операции выполняются быстро, но стоит только несколько секунд не попользоваться приложением, как очередное обращение к серверу начинает выполняться с заметной задержкой, после чего опять все работает быстро. Под обращением к серверу обычно подразумевается открытие формы выбора объекта, открытие формы с данными, получение очередной порции данных при прокрутке в таблице до конца, изменение местоположения и масштаба в графике.

При этом на сервере включена защита соединения по SSL.

#### Возможная причина

Для справки: клиент-приложение держит некоторое количество TCP/IP подключений к серверу в пуле, однако в целях экономии ресурсов при неиспользовании закрывает эти подключения по таймауту.

При создании нового соединения с защитой SSL происходит процедура инициации (т.н. SSL handshake), в процессе которой в т.ч. выполняется "reverse DNS lookup", т.е. определение имен хостов по их IP. Достаточно часто встречается ситуация, когда это определение работает медленно. Характерный признак: на клиентском компьютере при выполнении команды "ping -a IP" перед началом пингования происходит заметная задержка, в то время как команда "ping хост" выполняется сразу (подразумевается указание IP и хоста сервера).

#### Способы исправления

Единственным полноценным решением является правильное конфигурирование службы DNS в сети и на клиенте.

Временным решением может быть:

- отключение SSL на сервере;
- увеличение значения параметра `invokerDestructionDelay` в конфигурационном файле `ws-kernel-service.xml` (помогает частично, при большой нагрузке на сервер может привести к отказам).

## Неполадки, связанные с файловым хранилищем

#### Жалобы

- Долгое начальное открытие формы данных, при этом операции с файловым хранилищем оказываются заблокированными, в то время как все необходимые права и настройки заданы.
- Операции с файловым хранилищем разрешены, но при чтении и записи файлов происходят ошибки класса "Файл не найден", "Запись запрещена" и т.п.

#### Возможные причины

Для справки: при инициализации формы данных происходит считывание прав и различных настроек, в том числе – настроек файлового хранилища. При этом обязательно проверяется физическое наличие каталога для хранения файлов.

При ошибочном указании каталога, либо при неправильно настроенном доступе к нему, а так же в зависимости от операционной системы, могут случаться следующие проблемы:

- Операционная система сообщает об отсутствии такого каталога, но происходящие при этом проверки выполняются достаточно долго. Очень характерно в том случае, когда указан сетевой каталог, иногда задержки достигают десятка секунд.
- Операционная система сообщает о наличии каталога, но не дает читать его содержимое и создавать в нем новые файлы и подкаталоги. Обычно бывает при неправильной раздаче прав,

---

особенно характерно так же при сетевом каталоге.

### **Способы исправления**

Желательно проверять наличие каталога, указанного в параметрах банка XFiles\_Root и XFiles\_Trash, а так же прав на него, причем с правами пользователя операционной системы, под которым работает сервер приложений. Характерная ошибка администратора: только проверка наличия каталога, причем под аккаунтом администратора (суперпользователя). Следует не забывать, что аккаунты служб (сервисов, "демонов") зачастую ограничены в правах доступа, особенно на сетевые каталоги.

# Обеспечение жизненного цикла

**Часть**

---

**IV**



## Глава IV: Обеспечение жизненного цикла

### Устранение неисправностей

При возникновении какого-либо сбоя в работе сервера приложения рекомендуется воспользоваться следующим сценарием:

- Получить лог файлы сервера приложения. Если приложение Конструктор удастся запустить и подключиться к серверу, то получить лог файлы можно способом, описанным в главе [Параметры/Логи](#). При невозможности запуска Конструктора лог файлы можно найти вручную на компьютере с установленным сервером приложения. К ним относятся (путь указан относительно папки установки сервера приложения):
  - \bin\run.log
  - \server\default\log\boot.log
  - \server\default\log\server.log
- Проанализировать сообщения об ошибках, выведенных в лог файл. В случае успешного анализа исправить причину возникновения ошибки и при необходимости остановить и запустить заново сервер приложения. В качестве примеров исправления ошибок можно указать:
  - Сбой в работе СУБД, на которой хранится описание метамодели и информационный банк данных, отсутствие необходимых прав доступа к объектам СУБД.
  - Сбой в работе локальной сети, неверная настройка параметров сети или прав доступа к необходимым ресурсам.
  - Проблемы с оборудованием, в т.ч. отсутствие свободного дискового пространства.
- При невозможности самостоятельно понять причину сбоя необходимо обратиться за [технической поддержкой](#) в ООО "ГЕОКАД плюс". В письме необходимо обязательно приложить лог файлы, по возможности описать события, предшествующие ошибке.

При возникновении сбоя в работе клиент-приложения нужно использовать аналогичный сценарий, но анализировать надо в первую очередь лог файлы на компьютере пользователя. Если в этих файлах не будет обнаружено нужного описания ошибки, то следует переключиться на лог сервера.

Папка для вывода лог файлов на пользовательском компьютере указывается при установке клиент-приложения. По умолчанию это папка установки программы. К пользовательским лог файлам относятся:

- infoforms.log - лог файл приложений Forms Pro, Maps Pro и Logic Reports.
- constructor.log - лог файл приложения Конструктор.
- admin.log - лог файл приложения Администратор.

Помимо указанных файлов в папке можно найти файлы с аналогичными именами с добавлением даты, например, "infoforms-2016-06-27.log". Эти файлы содержат логи работы приложения в предыдущие дни.

### Совершенствование ПО

При работе с программой для повышения ее эффективности и актуальности может потребоваться модификация, вызванная следующими причинами:

- Изменение методической части программы в связи с изменением Российского и местного законодательства.
- Изменение существующих или добавление новых отчетных форм.
- Изменение существующих или добавление новых процедур обработки данных.
- Добавление новых функциональных возможностей.

Для внесения изменений в программу необходимо официальное обращение (письмо) в адрес руководства ООО "ГЕОКАД плюс" с указанием причин изменения программы, которое должно включать:

- Наименование и реквизиты закона, ведомственной инструкции или иного документа, на основании которого необходимо внести изменения в программу. Если документ отсутствует в публичном доступе в печати или сети Интернет, то необходимо приложить этот документ к обращению.
- Образцы печатных форм с пояснениями к правилам их заполнения, включая ссылки на нормативную документацию или законодательство.
- Алгоритмы и правила обработки данных специализированными процедурами. Форматы обменных файлов (название, ссылки на имеющуюся документацию).
- Описание новых функциональных возможностей, примеры их использования.

## Квалификация персонала

Для квалифицированного обслуживания сервера приложения GSEE необходимо, чтобы сотрудник (сотрудники) обладал следующими знаниями:

- Установка, настройка и администрирование сервера приложения:
  - Администрирование сервера: выделение вычислительных и дисковых ресурсов, подключение к сети и т.п. базовые вещи.
  - Настройка сетевой конфигурации, особенно в случае доступа из многих подсетей, использования NAT и проху.
  - Управление пользователями и группами доступа Active Directory / LDAP в случае их использования для авторизованного доступа в рамках приложения.
  - Администрирование СУБД, которая используется для хранения информационных банков и метамодели: управление базами данных, табличными и дисковыми пространствами, раздача прав доступа, создание и восстановление резервных копий, в идеале - анализ производительности и причин сбоев.
- Для самостоятельной доработки приложения под конечные нужды организации:
  - Понимание принципов устройства реляционных баз данных - требуется не только для проектирования, но и для тонкой раздачи прав доступа.
  - Проектирование РБД (таблицы, связи, SQL, отчёты), базовые навыки программирования на С-подобных языках.
- Для оказания помощи операторам системы:
  - Работа с растовыми изображениями: оптимизация, конвертация, понимание цветовых схем, принципов сжатия.
  - Работа с обменными файлами при импорте-экспорте: представление о внутреннем устройстве, ограничениях конкретных форматов, умение преобразовывать, знакомство со сторонними продуктами - источниками данных.

# Техническая поддержка

**Часть**

---



## Глава V: Техническая поддержка

### Лицензионное соглашение

**ВАЖНОЕ ЗАМЕЧАНИЕ:** Приведенная ниже информация является дополнением к договору на поставку программного обеспечения, но ни в каком случае не заменяет его.

Данное программное обеспечение в целом и все его компоненты в отдельности являются коммерческим продуктом. Все авторские права принадлежат фирме ООО "ГЕОКАД плюс".

Вы можете использовать данное программное обеспечение только таким способом, как это оговорено в договоре на поставку. Вы не можете передавать данное программное обеспечение или его компоненты, включая документацию, или каким либо другим способом давать доступ к ним третьим лицам, если это не оговорено в договоре на поставку. **ПОВТОРНОЕ РАСПРОСТРАНЕНИЕ И АРЕНДА ЗАПРЕЩЕНЫ.**

Вы не имеете права модифицировать, переводить, предпринимать обратное конструирование, перекомпилировать, пытаться раскрыть исходный код данного программного обеспечения за исключением способов, описанных в документации. Также Вы не можете удалять или изменять торговые марки, логотипы, значки авторского права или любые другие обозначения владельца в программном обеспечении, или в копиях, сделанных с данного программного обеспечения.

Программный продукт поставляется в форме "как есть", в соответствии с принципом, общепринятым в международной компьютерной практике. Это означает, что за проблемы, возникающие в процессе эксплуатации или инсталляции программ, разработчики ответственности не несут. Разработчики прилагают все усилия для того, чтобы данные проблемы никогда не возникали. Авторы не несут никакой ответственности за Вашу неспособность выполнить Ваши обязательства, или за какие - либо потери информации, прибыли и другие последствия Вашей деятельности по использованию программных средств, вследствие Вашей квалификации.

### Поддержка программных продуктов

Если у Вас появились вопросы по использованию программных продуктов ООО "ГЕОКАД плюс", сначала сделайте следующее.

1. Обратитесь к документации и другим печатным материалам, поставляемым в комплекте с данной программой.
2. Посмотрите, сможет ли помочь Справка программы. В любой момент Вы можете воспользоваться оперативной Справкой.

Если в Вашей организации есть сотрудник, на которого возложена обязанность по сопровождению программного продукта GSEE (системный администратор или методист), попробуйте получить помощь у него.

Если Вы не смогли найти ответ, по вопросам получения поддержки программного продукта Вас могут проконсультировать дилер фирмы, у которого Вы приобрели данный программный продукт, или непосредственно в ООО "ГЕОКАД плюс".

Обращаясь за консультацией, Вам нужно будет предоставить следующие сведения:

1. Номер используемой версии программного продукта. Для этого выполните команду **О программе** меню **Справка**, и на закладке **Информация** найдите параметры **Версия** и **Дата сборки**.
2. Тип аппаратуры, с которой Вы работаете, включая сетевую аппаратуру, если таковая используется.
3. Тип используемой операционной системы.
4. Точный текст сообщения, которое появилось на экране.
5. Описание того, что случилось, и что Вы делали в тот момент, когда возникла проблема.
6. Описание того, как Вы пытались разрешить ее.

Если у Вас на компьютере установлена электронная почта, то удобнее всего послать письмо в службу технической поддержки. Для этого выполните команду **Письмо в службу тех. поддержки** в меню **Справка**. При отправке письма таким способом будет автоматически собрана часть вышеперечисленных сведений.

Адрес ООО "ГЕОКАД плюс"

630034

ул. Троллейная, д. 35, ООО "ГЕОКАД плюс"

г. Новосибирск, Россия

тел. 1: (383) 352-13-33

тел. 2: (383) 352-14-04

тел. 3: (383) 352-15-50

E-mail: [techsup@geocad.ru](mailto:techsup@geocad.ru)

Internet: <http://www.geocad.ru/>

---

# Индекс

## - А -

Администратор системы 10

## - З -

Запуск приложения 24

## - Л -

Лицензионное соглашение 44

## - О -

О документации 6

Оператор системы 10

## - П -

Поддержка программных продуктов 44

Проектировщик системы 10

## - У -

Установка клиент-приложений

    Запуск приложения 24

Установка сервера приложения 12