



Руководство
администратора

«Finist-ВРМ»

Руководство администратора



1 Оглавление

1	Оглавление	2
2	Архитектура решения	3
3	Установка и настройка решения	4
3.1	Инсталляция системы.....	4
3.2	Состав поставки	4
3.3	Настройка сервера базы данных	4
3.4	Настройка сервера приложений	4
3.5	Запуск нескольких служб сервера приложений на одном сервере	7
3.6	Настройка клиента Finist.Report.UserArm	9
3.7	Перезапуск сервиса.....	10
3.8	Проверка работоспособности системы.....	10
4	Установка и настройка обновлений.....	12
4.1	Первоначальная настройка и регистрация сервиса обновлений	12
4.1.1	Состав поставки.....	12
4.1.2	Настройка сервера обновлений.....	12
4.2	Регулярная установка обновлений	13
4.2.1	Состав поставки.....	13
4.2.2	Установка обновлений.....	14
4.2.3	Проливка скриптов.....	16
4.2.4	Генерация версии	18
5	Функциональность системы	19
5.1	Администрирование системы.....	19
5.2	Сервисные функции	20
5.3	Настройки.....	22

2 Архитектура решения

Настоящий документ предназначен для отражения списка функциональности доступной администратору компоненты «Finist-BPM» (далее – Система).

Система представляет собой набор компонент, объединенных в трехзвенную модель, частями которой являются:

- Клиент - автоматизированное рабочее место пользователя системы;
- Сервер приложений - служба Linux, в которой заложена большая часть реализации обработки запросов пользователей, а так же связь с сервером хранения Баз Данных.
- Сервер базы данных - обеспечивает хранение данных.

В простейшей конфигурации сервер приложений может быть совмещен с сервером базы данных на одной виртуальной или физической машине, к которой возможно подключение по сети одного или нескольких терминалов.

С точки зрения безопасности, надежности, масштабирования системы рекомендуется разделить сервер приложений и сервер хранения баз данных.

Система так же поддерживает установку нескольких серверов приложений для распределения нагрузки между пользователями системы.

Для работы службы должны быть установлены:

- **.NET Framework 4.8** - для работы службы приложений;
- **PostgreSQL**.

3 Установка и настройка решения

3.1 Инсталляция системы

Для повышения безопасности системы следует создать системную учетную запись пользователя Linux – *fr_user@domain* (далее *fr_user*).

В системе Linux пользователь должен иметь права на чтение/запись ко всем используемым Системой каталогам (включая каталоги, в которых установлен сервер приложений для корректного обновления службы), а также доступ к базе данных с правами owner.

3.2 Состав поставки

Поставка представляет собой архив «EventManager.zip», содержащий файлы и приложения для инсталляции решения.

Структура архива:

- Каталог «Server» – сервер приложений;
- Каталог «UserArm» – клиентское приложение;
- Файл с расширением.bak – база данных.

3.3 Настройка сервера базы данных

На подготовленный сервер базы данных развернуть буферную базу данных, входящую в поставляемый дистрибутив.

3.4 Настройка сервера приложений

1. Установка сервера приложений осуществляется в выбранный каталог (далее - каталог установки C:\Finist\Server\) методом копирования файлов из каталога «EventManager\Server» поставки (3.2 Состав поставки).
2. Регистрация сервиса производится выполнением команды в командной строке с правами администратора:

```
sc create EventManager.Server binPath="C:\Finist\Server\EventManager.Service.exe"  
DisplayName="EventManager.Server" start=auto
```

3. В конфигурационном файле EventManager.Service.exe.config в секции <GEst.Server> прописать параметр <ConnectionString>:

```
“server= «Имя сервера БД»; database= EventManager; Integrated Security=true; packet size=4096; connection reset=false; connection lifetime=500; min pool size=1; max pool size=50”
```

В параметре **Integrated Security**=TRUE мы указываем аутентификацию пользователя системной записью Linux, что позволит не хранить в открытом доступе пароль доступа к SQL Server.

Для повышения уровня надежности системы во время выполнения ресурсозатратных хранимых процедур на базе данных (например, в процессе импорта данных или произведения построения потоков или расчетов на большом объеме данных) рекомендуется в секцию <**CommandTimeouts**> добавить строку:

```
<Long value="7200" /> , где 7200 – время таймаута в секундах.
```

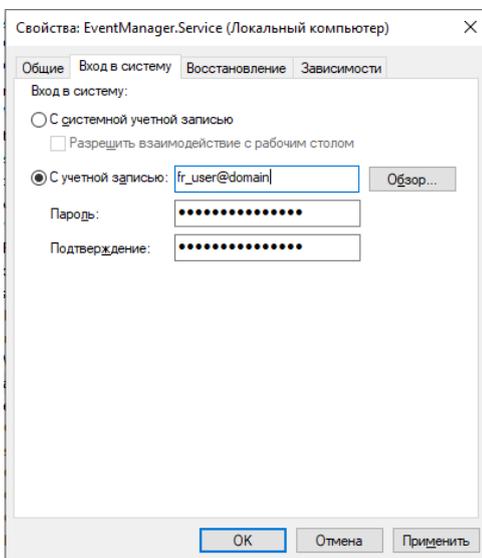
Если «Long value» в указанной секции отсутствует, то его необходимо добавить вручную.

4. Необходимо зарегистрировать WCF.

Регистрация производится выполнением команды в командной строке с правами администратора:

```
netsh http add urlacl url=http://+:8810/GestBPMWebService/ user=все
```

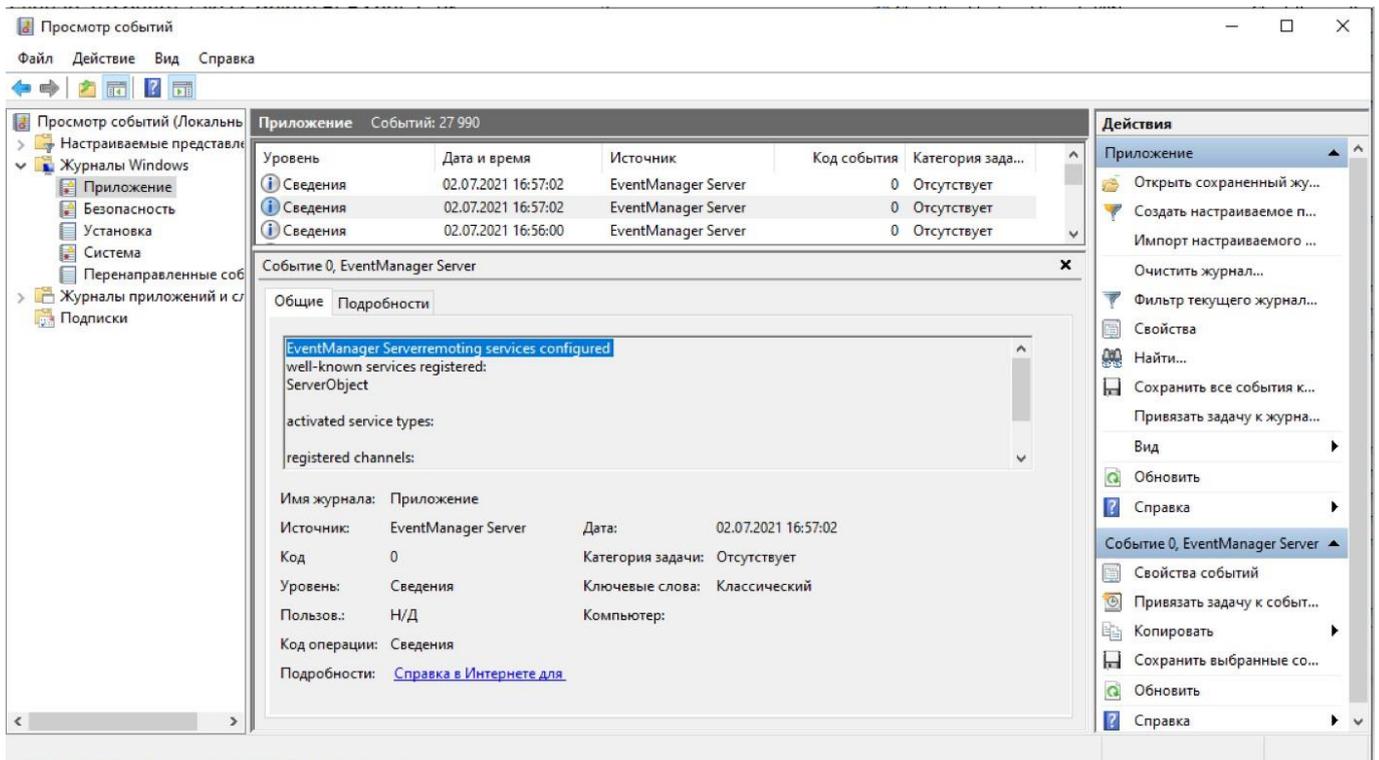
5. После настроек конфигурации следует произвести старт службы "EventManager.Service" из приложения «Службы» Linux от имени пользователя, заведенного в пункте 3.1 Инсталляция системы. Для этого необходимо перейти в свойства службы – Вход в систему и указать логин и пароль.



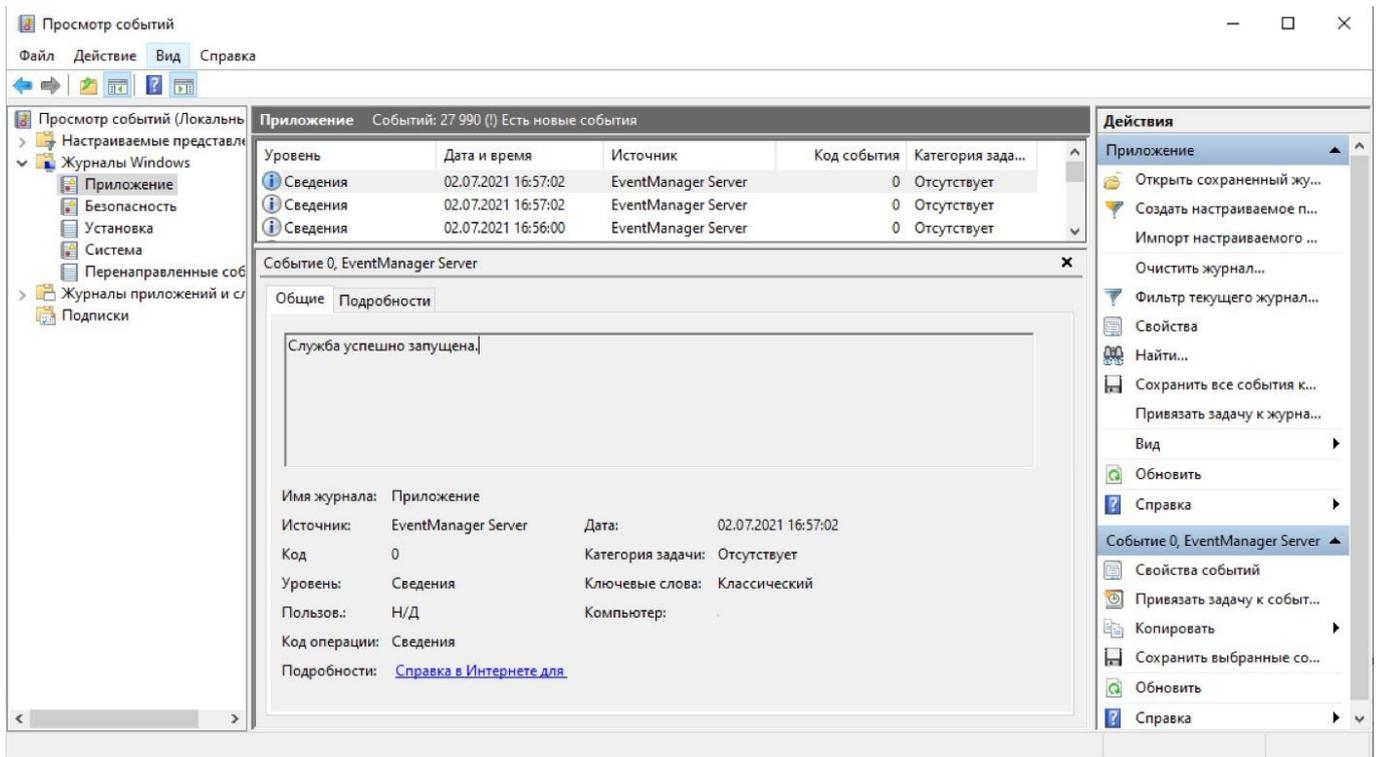
Далее необходимо убедиться в корректности работы службы. Для этого требуется перейти в «Панель управления/Администрирование/Просмотр событий».

В окне «Журналы Linux/Приложения» успешный запуск определяется двумя событиями (Источник = EventManager.Server):

1. EventManager.Server remotng services configured



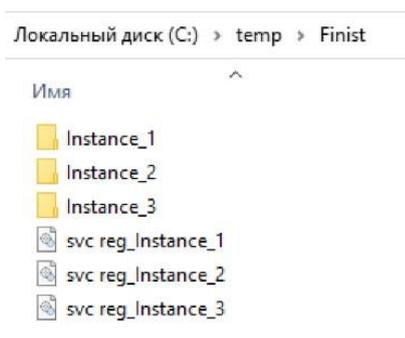
2. Служба успешно запущена.



3.5 Запуск нескольких служб сервера приложений на одном сервере

В случае, если на сервере требуется распределить серверы, направив их на разные базы следует произвести следующие действия:

1. Копируем папку с сервером (службой) на необходимое количество экземпляров:



2. В каждом файле, содержащем конфигурацию сервера EventManager.Service.exe.config изменить значения портов.

В Instance_1:

```
<HostAddress value="tcp://{0}:32801/ServerObject" />
<channel ref="tcp" port="32801" tokenImpersonationLevel="Identification" secure="true"
```

```
protectionLevel="None">
```

В Instance_2:

```
<HostAddress value="tcp://{0}:32802/ServerObject" />  
<channel ref="tcp" port="32802" tokenImpersonationLevel="Identification" secure="true"  
protectionLevel="None">
```

В Instance_3:

```
<HostAddress value="tcp://{0}:32803/ServerObject" />  
<channel ref="tcp" port="32803" tokenImpersonationLevel="Identification" secure="true"  
protectionLevel="None">
```

3. Далее необходимо зарегистрировать сервис для каждого экземпляра посредством выполнения команды в командной строке с правами администратора.

В Instance_1:

```
sc create EventManager.Server_1 binPath= "C:\temp\Finist\Instance_1\  
EventManager.Service.exe" DisplayName= " EventManager.Server (Instance 1)" type= own  
start= auto
```

В Instance_2:

```
sc create EventManager.Server_2 binPath= "C:\temp\Finist\Instance_2\  
EventManager.Service.exe" DisplayName= " EventManager.Server (Instance 2)" type= own  
start= auto
```

В Instance_3:

```
sc create EventManager.Server_3 binPath= "C:\temp\Finist\Instance_3\  
EventManager.Service.exe" DisplayName= " EventManager.Server (Instance 3)" type= own  
start= auto
```

Обязательно должны быть доступны порты на сервере, если порты заняты необходимо указать другие доступные. После всех настроек и проверок следует произвести старт служб из приложения «Службы» Linux от имени пользователя, заведенного в пункте 3.1 Инсталляция системы и проверить корректность запуска через «Просмотр событий» (3.4 Настройка сервера приложений пункт 5).

4. Настроить клиентские приложения на соединения с соответствующими службами, путем указания порта соединения со службой в конфигурационном файле клиентского приложения.

3.6 Настройка клиента **Finist.Report.UserArm**

1. Установка клиентского приложения производится методом копирования файлов из каталога «EventManager\UserArm» поставляемых наборов компонент Системы в удобную для использования папку на рабочей станции Пользователя.

2. Настройка подключения клиентской части к службе сервера приложений производится путем указания в файле *EventManager.UserArm.exe.config* в секции <GEst.Client> параметра "HostName" ip адреса или доменного имени сервера, где развернут сервер приложений.

Если для установки использовался путь по умолчанию – пример расположения конфигурационных файлов:

- Для клиента:

C:\Finist\UserArm\EventManager.UserArm.exe.config

- Секция для проверки:

```
<GEst.Client>
```

```
  <HostName value="имя или ip сервера приложений" />
```

```
  ...
```

```
</GEst.Client>
```

3. Также на сервере приложений и в клиентской части должны быть открыты порты, указанные в конфигурации сервера и клиента в секциях <GEst.Common> в параметре <HostAddress>.

```
<GEst.Common>
```

```
  <HostAddress value="tcp://{0}:32801/ServerObject" />
```

```
</GEst.Common>
```

Если для установки использовался путь по умолчанию – пример расположения конфигурационных файлов:

- Для сервера:

C:\Finist\Service\EventManager.Service.exe.config

- Для клиента:

C:\Finist\UserArm\EventManager.UserArm.exe.config

4. Для входа в систему по умолчанию настроена учетная запись администратора, логин «admin», пароль «sig».

3.7 Перезапуск сервиса.

Перезапуск сервиса производится штатными средствами Linux – программа «Службы».

Для перезапуска сервиса необходимо:

1. Остановить сервис.
2. Запустить сервис.

Для того чтобы убедиться в штатном старте сервиса необходимо в программе «Просмотр событий» открыть папку «Просмотр событий (Локальный)/Журналы Linux/Приложение» и убедиться, что нет ошибок, и есть две записи о событиях:

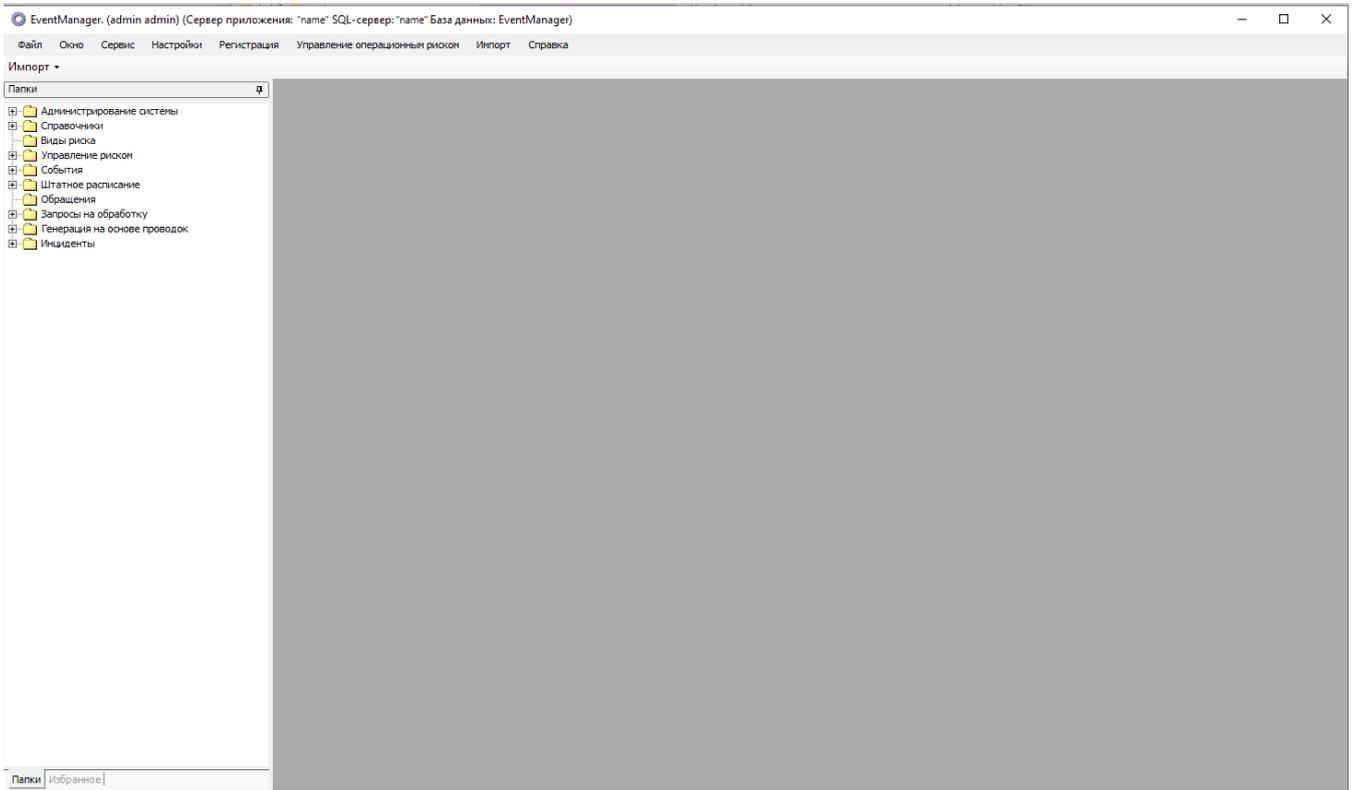
1. EventManager.Server remoting services configured
2. Служба успешно запущена.

(Опционально) Проверить с помощью клиентского приложения работоспособность сервиса выполнением простых действий, таких как открытие списков, фильтрация, открытие форм.

При полной перезагрузке сервера необходимо проверить работоспособность сервиса вышеуказанным способом.

3.8 Проверка работоспособности системы

Для проверки работоспособности Системы необходимо запустить клиентское приложение и убедиться, что запускается главная форма приложения.



4 Установка и настройка обновлений

4.1 Первоначальная настройка и регистрация сервиса обновлений

4.1.1 Состав поставки

Поставка представляет собой каталог «Update», содержащийся в архиве установки решения «EventManager.zip\UserArm» и каталог «UpdateServer», поставляемый в отдельном архиве или в архиве установки решения.

Структура каталога «Update»:

- Приложение «EventManager.UpdateUtility.exe» – утилита обновления клиентского решения.
- Файл «EventManager.UpdateUtility.exe.config» – конфигурационный файл подключения к порту и указания расположения клиентского приложения.
- Динамическая библиотека, состоящая из файлов формата dll.

Структура каталога «UpdateServer»:

- Приложение «EventManager.UpdateServer.exe» – утилита обновления сервера приложений.
- Файл «EventManager.UpdateServer.exe.config» – конфигурационный файл подключения к порту и указания расположения пакета и версии обновления.
- Динамическая библиотека, состоящая из файлов формата dll.

4.1.2 Настройка сервера обновлений

1. Установка клиентской части обновления («Update») осуществляется в выбранный каталог (каталог установки C:\Finist\UserArm\Update\) методом копирования файлов из каталога «EventManager\UserArm\Update» поставки (3.1 Инсталляция системы).
2. Установка серверной части обновления («UpdateServer») осуществляется в выбранный каталог (каталог установки C:\Finist\UpdateServer) методом копирования файлов из каталога «EventManager\UpdateServer\» или из отдельно предоставленного архива.
3. Регистрация сервиса производится выполнением команды в командной строке с правами администратора:

```
sc create EventManager.UpdateServer  
binPath="C:\Finist\UpdateServer\EventManager.UpdateServer.exe"  
DisplayName="EventManager.UpdateServer" start=auto
```

4. В конфигурационном файле EventManager\UserArm\Update\EventManager.UpdateUtility.exe.config в секции <Settings> прописать параметр <ServerUri>:

“Ошибка! Недопустимый объект гиперссылки. стол»:«порт»/EventManagerUpdate.rem”

В конфигурационном файле EventManager\UpdateServer\EventManager.Report.UpdateServer.exe.config в секции <channels> прописать параметр <channel>:

ref=“http” port=“«порт»”

На сервере приложений и в клиентской части должны быть открыты порты, указанный в конфигурации сервера.

А также в секции <Settings> прописать параметр <UpdatesFolder>:

Channel ref=“«путь до каталога установки (C:\Finist\Updates)»”

5. После настроек конфигурации следует произвести старт службы "EventManager.UpdateServer" из приложения «Службы» Windows. Далее необходимо убедиться в корректности работы службы (3.4 Настройка сервера приложений пункт 5).

4.2 Регулярная установка обновлений

4.2.1 Состав поставки

Поставка представляет собой каталог Build_1.0.1.0 (нумерация зависит от поставляемого обновления, в последующем просто Build).

Структура каталога «Build»:

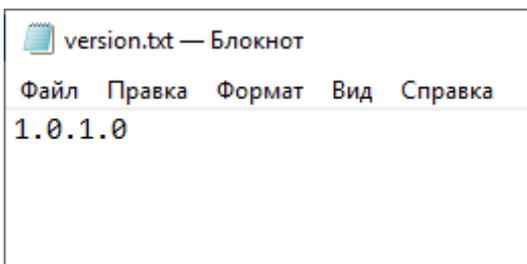
- Каталог «Client» – обновление клиентского решения;
- Каталог «Scripts» – обновление баз данных;
- Каталог «Server» – обновление сервера приложений;
- Каталог «Scenarios» - обновление сценариев приложения;
- Файл Changes.txt – информативный текстовый файл для пользователей решения о поставляемых обновлениях.

4.2.2 Установка обновлений

Рекомендуется перед установкой обновления создать резервные копии баз данных для возможности восстановления решения, в случае некорректного поведения и прочих непредвиденных ситуаций.

1. Необходимо остановить службу приложения «EventManager.Server» и службу обновления «EventManager.UpdateServer» из приложения «Службы» Linux.
2. Далее прогоняем скрипты из каталога «Scripts» в порядке возрастания нумерации. Если скриптов довольно много, то можно воспользоваться деплоером (приложение Deploy – подробнее в 4.2.3 Проливка скриптов).
3. Заменить файлы каталога «Server» (C:\Finist\Server\), где установлено решение, файлами из поставляемого каталога «Server» (каталог Build\Server\).
4. При первичном обновлении необходимо создать каталог «Updates» в корневом каталоге (C:\Finist\). В данной папке создаем каталог с номером текущего обновления, к примеру «1.0.1.0» (в случае выкладки фикса создаем папку с номером следующей версии: «1.0.1.1», далее будет «1.0.1.2» и так до установки следующего обновления: 1.0.2.0).

Выкладываем в эту папку файлы каталога «Client» (каталог Build\Client\). **Важно сохранить структуру данных в том же виде (в основном касается выкладки фиксов)!** Проверяем в текущей папке наличие файла версии – version.txt. В случае отсутствия создаем данный файл и в нем прописываем номер текущего обновления (1.0.1.0). Пример:



В каталоге «Updates» создаем файл versions.xml, в случае его отсутствия. Структура файла должно иметь вид:

```
<?xml version="1.0" encoding="utf-8"?>
<versions>
  <application name="EventManager" handler="FileUpdate">
    <BuildMaker>
      <version number="1.0.1.0">
        <part name="UserARM">
          <itemGEst name="«название файла»" src="1.0.1.0\«путь
            расположения файла»" dst="«путь расположения
```

```
        файла»" />
        ...
        <itemUserARM
        name="EventManager.UserArm.exe"
        src="1.0.1.0\EventManager.UserArm.exe"
        dst="EventManager.UserArm.exe" />
    </part>
    <part name="Server">
        <itemGEst name="«название файла»" />
        ...
        <itemServer name="EventManager.Server.exe"/>
    </part>
</version>
</BuildMaker>
<version number="«номер обновления»">
    <part name="UserARM">
        <item name="«название файла»" src="«номер
        обновления\путь расположения файла»" dst="«путь
        расположения файла»" />
        ...
        <item name="version.txt" src="«номер
        обновления»\version.txt" dst="version.txt" />
    </part>
</version>
</application>
</versions>
```

При последующей выкладке обновления в файл `versions.xml` добавляется блок тегов с описанием файлов, входящих в обновление (от тэга `<version number="*.*.*.*">` до тэга `</version>`). В данном блоке прописывается номер нового обновления, названия и пути файлов, а также файл с новой версией – `version.txt`.

Для упрощения и исключения ошибок, связанные с человеческим фактором, создана автоматическая генерация данного блока (подробнее в 4.2.4 Генерация версии).

5. Далее необходимо запустить службу приложения «`EventManager.Service`» и службу обновления «`EventManager.UpdateServer`» из приложения «Службы» Windows и убедиться в корректности запуска через «Просмотр событий» Windows (3.4 Настройка сервера приложений пункт 5).

6. После успешного запуска служб, необходимо запустить утилиту `EventManager.UpdateUtility.exe`, располагаемую в каталоге «Update» клиентской части (`C:\Finist\UserArm\`). При первичной установке пользовательского (клиентского)

автоматизированного рабочего места достаточно скопировать с работающего клиента или с сервера приложений дистрибутив с клиентским приложением и запускать утилиту обновления (при более старой скопированной версии, первое обновление будет чуть дольше, так как объем закачиваемых файлов окажется больше).

7. Импорт сценариев производится непосредственно через само приложение, необходимо открыть каталог «Администрирование системы\Сценарии выполнения процессов». В контекстном меню списка выбрать пункт «Импорт» – в каталоге «Scenarios» выбираем файл сценария и нажимаем «Ок». После импорта всех сценариев из текущего каталога необходимо перезапустить механизм сценариев через пункт меню «Сервис\Перезапустить механизм клиентских сценариев workflow».

Проверить успешность установки текущего обновления можно в приложении через пункт меню «Справка\О программе».

4.2.3 Проливка скриптов

Поставка представляет собой архив «Deploy.zip», содержащий файлы и приложения для проливки скриптов.

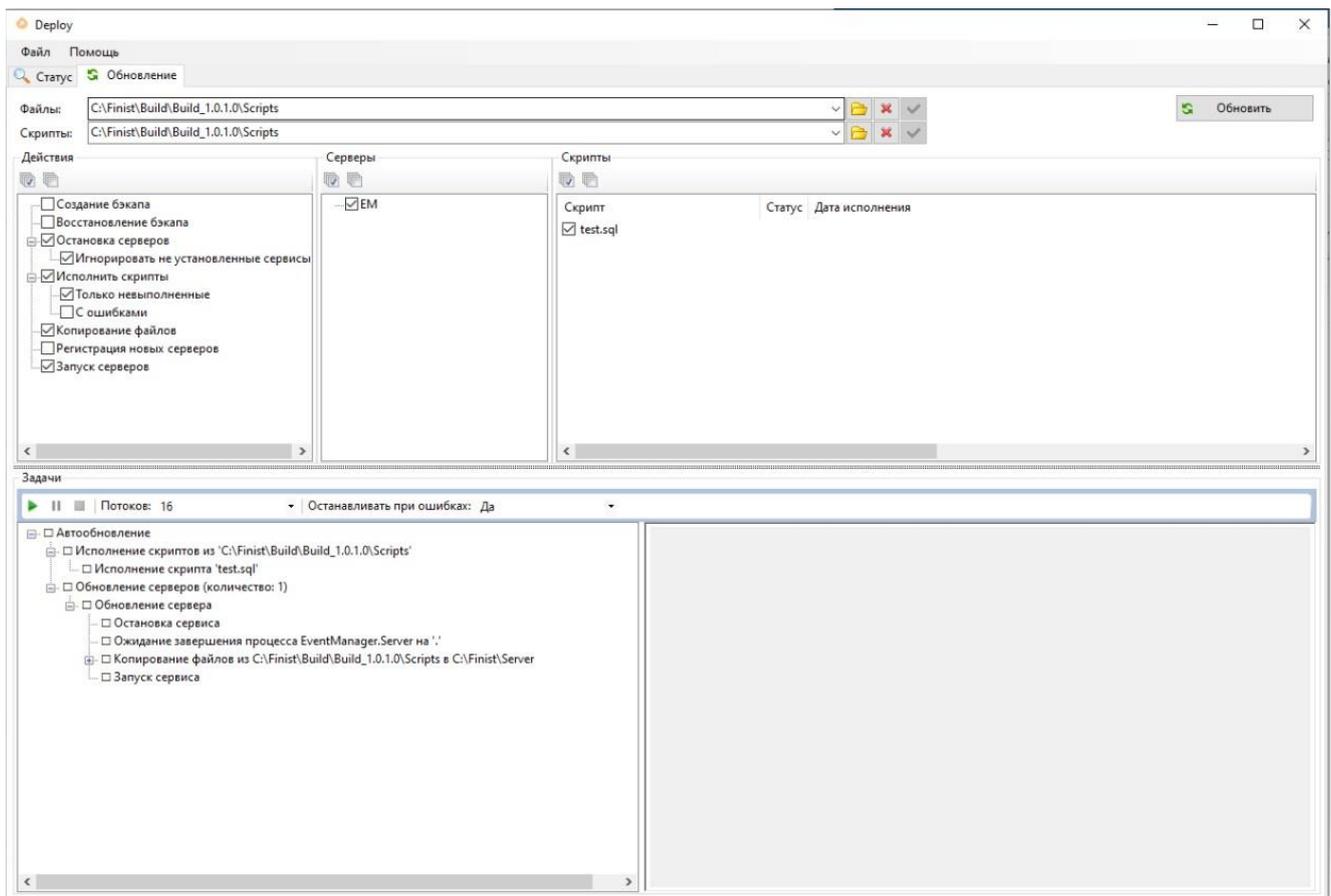
Структура архива:

- Каталог «distrs» – каталог, содержащий файлы формата msi для инсталляции приложения;
- Каталог «logs» – каталог, содержащий логи ошибок приложения;
- Каталог «Resources» – системный каталог, содержащий sql скрипты;
- Файл «Finist.Deploy.Client.exe.config» – конфигурационный файл подключения к базе данных;
- «Finist.Deploy.Client.exe» – файл запуска приложения;
- Файлы формата dll – динамическая библиотека приложения.

Изначально необходимо в конфигурационном файле Finist.Deploy.Client.exe.config прописать параметр <appSettings>:

```
<add key="ConnectionString" value="server=«Сервер подключения»; database=«База данных»; integrated security=false – в случае идентификации по логину и паролю и true – в случае windows аутентификации; user=«Логин»; password= «Пароль» (для windows аутентификации логин и пароль не указывается); persist security info=True; packet size=4096; connection reset=false; connection lifetime=500; min pool size=1; max pool size=50" />
```

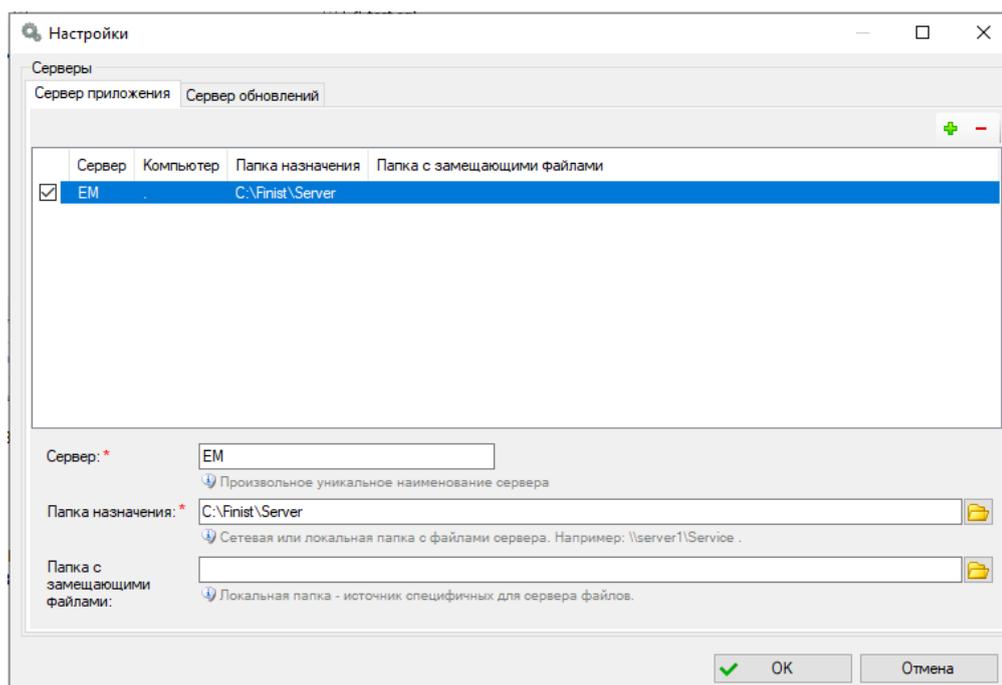
Интерфейс приложения имеет следующую структуру:



На вкладке «Обновление» необходимо указать путь откуда будет происходить копирование файлов и проливка скриптов – поля «Файлы» и «Скрипты».

В блоке «Действия» проставляем галочки в тех пунктах, которые хотим исполнить.

Куда будет происходить копирование файлов и на какой сервер будет происходить проливка скриптов настраивается в настройках, открыть которые можно через пункт меню «Файл\Настройки».



В поле «Сервер» указываем наименование сервера куда необходимо пролить скрипты, папка назначения – путь куда будет происходить копирование файлов.

В блоке «Скрипты» мы можем увидеть все скрипты и их статус. Если скрипт уже был пролит ранее (к примеру в последнем выложенном фиксе), то он будет иметь статус «Выполнен» и будет проставлена дата исполнения.

После того как все настройки выполнены запускаем проливку скриптов –  (зеленая стрелочка) в блоке «Задачи». По выполнению каждого скрипта будет происходить их «окраска» в зеленый цвет, означающее успешное исполнение.

4.2.4 Генерация версии

Для автоматической генерации версии в файл `versions.xml` (C:\Finist\Updates) необходимо распаковать архив «Generate versions.zip» в каталог «Updates».

Структура архива:

- Каталог «_ps» – каталог сценариев;
- `generate_versions.lnk` – ярлык генерации;
- `generate_versions.ps1` – сценарий генерации.

После импорта файлов каталога «Client» (каталог `Build\Client\`) в каталог с номером обновления (к примеру, 1.0.1.0 в каталоге `C:\Finist\Updates\`) и указания версии в файле `version.txt` необходимо запустить генерацию блока текущего обновления путем нажатия на `generate_versions.lnk`. После успешной отработки сценария необходимо проверить отработку генерации, для этого открываем файл `versions.xml` (C:\Finist\Updates) и проверяем что появился блок от тэга `<version number="*.*.*.*">` до тэга `</version>` в конце файла с текущим номером обновления и обновляемыми файлами.

5 Функциональность системы

В разделе приведен перечень функций доступных администратору системы.

Доступность функций администратора определяется ролью «Администратор»:

- Все доступные функции администратора находятся в списке «Администрирование системы» в левой панели системы «Папки»;
- Дополнительно на главной панели системы открывается доступ к сервисным функциям настройкам системы «Сервис» и «Настройки»;
- Дополнительно для администратора системы открывается доступ к списку «Справочники» в левой панели системы «Папки» для Наполнения, Изменений и просмотра данных основных справочников системы.

5.1 Администрирование системы

В разделе приведен основной перечень и краткое описание функций доступных администратору Системы.

№ п/п	Наименование	Краткое описание
1.	Задания выполняемые по расписанию	Создание, Изменение, Просмотр операций для выполнения по заданному расписанию
2.	Значок	Создание, Изменение, Просмотр значков системы для использования в рамках стилизации элементов интерфейса системы (кнопки, команды)
3.	Индексы базы данных	Просмотр индексов базы данных из интерфейса системы.
4.	Клиентские модули	Создание, Изменение, Просмотр клиентских модулей системы для ограничения доступа и организации ролевой модели на более крупном уровне. Клиентские модули представляет собой обособленный блок функциональности.
5.	Команды	Создание, Изменение, Просмотр команд для запуска процессов, сценариев и иной функциональности системы. * Может быть представлена в виде кнопок в интерфейсе системы или в виде логики запуска процессов и сценариев по заданному сигналу системы
6.	Конфигурационные параметры	Создание, Изменение, Просмотр конфигурационных параметров для указания значимых как системных параметров, так и бизнес параметров, которые устанавливаются администраторами системы и используется в случае определения глобальных настроек системы или точечных локальных настроек системы для использования сценариях процессах системы.
7.	Логические условия	Создание, Изменение, Просмотр логических проверочных или разрешающих условий для валидации данных объектов сущности в момент сохранения объекта и/или при использовании команды
8.	Настройки импорта	Создание, Изменение, Просмотр сопоставлений файлов и атрибутов сущностей системы для последующей разовой загрузки и/или регулярной загрузки в комбинации с функционалом планировщика заданий

		* Механизм для импорта данных из внешних файловых источников
9.	Отчеты	Создание, Изменение, Просмотр шаблонов отчетов для агрегации и вывода данных в файлах формата XML, EXCEL, PDF Генерация операционных отчетов на базе созданных шаблонов * Требуется установка плагина для Microsoft Excel для упрощенной настройки шаблонов.
10.	Пользователи	Создание, Изменение, Просмотр пользователей системы с возможностью создания нескольких профилей системы, настройка параметров аутентификации и настройка ролей пользователя для ограничения доступа к данным и элементам системы.
11.	Программные объекты базы данных	Просмотр основных объектов базы данных из интерфейса системы. * Триггеры, Процедуры, Функции
12.	Процессы	Создание, Изменение, Просмотр процессов, которые представляют собой дополнительные действия в системе, которые могут быть использованы в сценариях. Предназначены для расширения базовой функциональности Системы.
13.	Роли	Создание, Изменение, Просмотр ролей в Системе для организации ролевой модели доступа к данным элементами системы
14.	Списки объектов	Создание, Изменение, Просмотр списков для вывода списочного вида данных с возможностью указания необходимых условий для ограничения выборки и для ограничения доступа определенных ролей и пользователей. *Также имеет другие параметры для более детальной настройки
15.	Стили списков	Создание, Изменение, Просмотр стилей для цветовой индикации записей в списках данных по заданным условиям
16.	Сущности	Создание, Изменение, Просмотр сущностей (бизнес-объектов) и их атрибутивного состава для организации хранения данных в Системе. * Позволяет также создавать представления, вычисляемые атрибуты и настраивать сущности на основе таблиц сторонней базы данных.
17.	Сценарии выполнения процессов	Создание, Изменение, Просмотр сценариев системы для автоматизации бизнес-логики и бизнес-правил. * Инструмент для создания бизнес-правил для сверки, обработки, наполнения данных
18.	Формы	Создание, Изменение, Просмотр интерфейсных форм для
19.	Шаблоны документов	Создание, Изменение, Просмотр шаблонов документов для агрегации и вывода данных в файлах формата WORD, PDF
20.	Экземпляры сценариев	Функциональность для Просмотра и Отладки сценариев * При каждом запуске сценария создается экземпляр сценария, в котором записывается полный процесс работы сценария
21.	Элементы управления	Просмотр элементов управления для применения и/или расширения дополнительных свойств элементов интерфейсной формы: поля, списки, группы полей на форме и т.д.

5.2 Сервисные функции

В разделе приведен основной перечень и краткое описание сервисных функций.

Наименование	Краткое описание
Просмотр произвольного списка	Возможность просмотра данных любой сущности системы, к которой определен доступ на уровне ролевой модели

Вкл/Выкл сервисное меню	Возможность просмотра дополнительной информации по интерфейсным формам системы: Сущность объекта, KernelID объекта
Просмотр ошибок	Просмотр лога ошибок сохраненных в корневой директории, где размещен сервер приложений
Перезапуск системы безопасности	Перезапуск системы безопасности для применения изменений внесенных в элементы ролевой модели: Изменение прав доступа, Создание/Изменение ролей, Создание/Изменение клиентских модулей.
Монитор серверных процессов	Возможность просмотра активных серверных процессов и подробной информации по ним: Приоритет, Процесс, Состояние, Объект, Этап, Время старта, CPU, Пользователь и т.д.
Перезапуск холдера обработчика отчетов	Перезапуск механизма отчетности для применения изменений внесенных в существующие отчетность и шаблоны документов и инициализация(регистрации) новых отчетов и документов
Просмотр запланированных задач	Просмотр активных запланированных заданий в планировщике заданий
Управление сценариями	Перезапуск механизма серверных сценариев для применения изменений внесенных в существующие серверные сценарии и инициализация(регистрации) новых серверных сценариев
Зарегистрировать серверные процессы	Перезапуск механизма процессов для применения изменений внесенных в существующие процессы и инициализация(регистрации) новых процессов
Зарегистрировать формы данных	Перезапуск механизма интерфейсных форм для применения изменений внесенных в существующие интерфейсные формы и инициализации(регистрации) новых интерфейсных форм
Перегрузить метаописания объектов на всех клиентских приложениях	Перезапуск механизма работы с сущностями для применения(загрузки метаописаний) изменений внесенных в существующие сущности и регистрации новых сущностей
Перегрузить холдеры-бизнес-объектов	Перезапуск отдельных механизмов для работы с прочими элементами системы для применения изменений внесенных и новых: <ul style="list-style-type: none"> • График рабочего времени • Значки • Команды интерфейса • Конфигурационные параметры системы • Логические условия • Настройки импорта • Описания процессов • Порядковые номера • Стили списков • Сущности • Схемы десериализации • Схемы сериализации Формы ввода данных
Перезапустить механизм клиентских сценариев	<ul style="list-style-type: none"> • Перезапуск механизма клиентских сценариев для применения изменений внесенных в существующие клиентские сценарии и инициализации(регистрации) новых клиентских сценариев
Подключенные пользователи	Возможность просмотра списка подключенных/активных пользователей системы
Поиск в объектах системы	Возможность поиска по ключевым словам во всех объектах системы, в том числе системных

--	--

5.3 Настройки

В разделе приведен основной перечень и краткое описание функций, доступных на главной панели в меню «Настройки».

Наименование	Краткое описание
Управление логами	Возможность просмотра и настройки локальных и серверных логов