



**АРГУС**  
QWFM CC

**Инструкция для администратора  
«АРГУС Workforce Management  
for Call Centre»**

**Оглавление**

Лист регистрации изменений .....	4
1. Установка PostgreSQL .....	5
1.1 Подготовка системы .....	5
1.1.1 Установка пакетов БД .....	5
1.1.2 Перенос директории данных postgres в раздел дискового пространства argus6	
1.2 Создание БД WFM CC .....	7
1.2.1 Установка утилиты управления БД .....	7
1.2.2 Создание системного пользователя БД и продуктивной БД Argus .....	7
1.2.3 Создание БД WFM CC .....	8
2. Обновление БД WFM CC .....	9
3. Установка СП WFM CC .....	10
3.1 Создание пользователя argus .....	10
3.2 Установка Java-машины .....	10
3.3 Данные часового пояса .....	11
3.4 Локаль и кодировка операционной системы .....	12
3.5 Синхронизация времени .....	12
3.6 Максимальное число открытых файлов и сокетов .....	12
3.7 Максимальное число запущенных процессов .....	13
3.8 Дополнительные рекомендуемые настройки ОС .....	13
3.9 Имя хоста .....	14
3.10 Создание структуры каталогов для артефактов сервера приложений .....	14
3.11 Создание файла конфигурации для установки дистрибутива основного сервера приложений .....	15
3.12 Установка СП WFM CC .....	17
4. Обновление СП WFM CC .....	18
5. Настройка e-mail оповещений .....	19
5.1 Предварительные настройки .....	19
5.2 Настройка файла конфигурации .....	19
5.3 Перезапуск сервера приложений .....	20
6. Логирование .....	20
6.1 Логирование СП WFMCC (WildFly) .....	20
6.1.1 Access.log .....	20
6.1.2 Server.log .....	21

6.1.3 Webservices.log .....	21
6.1.4 Bugreports .....	21
6.2 Логирование БД (PostgreSQL) .....	21
6.3 Логирование СП интеграции (Tomcat) .....	22

**Лист регистрации изменений**

Версия	Дата	Исполнитель	Изменения
1.	10.09.2019	Графов М.С.	Базовая версия документа.
2	01.10.2019	Графов М.С.	Добавлена глава 5.Настройка e-mail оповещений
3.	14.10.2019	Графов М.С.	Добавлена глава 6.Логирование
4.	08.11.2019	Графов М.С.	Добавлена глава 7. Необходимые регулярные процедуры

В состав КТС входят:

- СП WFMCC (WildFly)
- БД (PostgreSQL)
- СП интеграции (Tomcat)

## 1. Установка PostgreSQL

Все команды требуют прав супер-пользователя.

### 1.1 Подготовка системы

На всех хостах отключить **selinux** (требуется перезапуск хоста) и **firewall**:

```
/etc/sysconfig/selinux  
SELINUX=disabled
```

```
service firewalld stop  
systemctl disable firewalld
```

#### 1.1.1 Установка пакетов БД

Команды выполняются под УЗ **postgres**, для установки использовать postgresql не ниже 10 версии:

```
yum install postgresql10-server postgresql10 postgresql10-contrib  
user: postgres  
.bash_profile  
export PGDATA=/var/lib/pgsql/10/data  
export PATH=/usr/pgsql-10/bin:$PATH  
export LD_LIBRARY_PATH=/usr/pgsql-10/lib/  
export PGPASSFILE=/var/lib/pgsql/.pgpass
```

В **pg\_hba.conf** необходимо указать следующие параметры:

/var/lib/pgsql/10/data/**pg\_hba.conf**

```
host all postgres <ip-address> /<netmask> trust  
host all argus_sys <ip-address> /<netmask> trust
```

В **postgresql.conf** необходимо указать следующие параметры:

/var/lib/pgsql/10/data/**postgresql.conf**

```
listen_addresses = '*'  
port = 5432  
wal_level = replica  
  
archive_mode = on  
archive_command = 'cp %p /var/lib/pgsql/10.0/archive/%f'  
max_wal_senders = 4  
wal_keep_segments = 100  
synchronous_commit = local  
synchronous_standby_names = '*'
```

```
hot_standby = on
default_text_search_config = 'pg_catalog.russian'

max_connections = 230
shared_buffers = 768MB
effective_cache_size = 2304MB
maintenance_work_mem = 192MB
checkpoint_completion_target = 0.7
wal_buffers = 16MB
default_statistics_target = 100
random_page_cost = 4
effective_io_concurrency = 2
work_mem = 1139kB
min_wal_size = 1GB
max_wal_size = 2GB
max_worker_processes = 5
max_parallel_workers_per_gather = 3
```

Инициализация БД:

```
# /usr/pgsql-10/bin/postgresql-10-setup initdb
```

### 1.1.2 Перенос директории данных postgres в раздел дискового пространства argus

Останавливаем сервер postgres:

```
su - postgres  
pg_ctl stop (из-под пользователя postgres)
```

Либо:

```
systemctl stop postgresql-10.0 (если сервер был запущен из-под root)
```

Делаем бэкап директории **pgsql**. Можно как средствами самого postgres (**pg\_dump**, **pg\_dumpall** или **pg\_basebackup**), так и более простым способом:

```
cd /var/lib/pgsql/10.0  
tar -zcvf /argus/pgdatabackup.tar.gz data
```

Меняем настройки директории нахождения данных БД в файлах конфигурации postgres, то есть везде где присутствует **/var/lib** меняем значение на **/argus** особенно важен файл: - **.bash\_profile** в **/var/lib/pgsql**

Заменить в нём необходимо:

```
PGDATA=/var/lib/pgsql/10.0/data на /argus/pgsql/10.0/data
```

После развертывания базы PostgreSQL по умолчанию (**yum install...**) необходимо перенести директорию данных и архив из каталога root (по умолчанию **/var/lib/pgsql/10.0**) в предназначенный для директорий postgres каталог **/argus/pgsql/10.0**, предварительно создав эту директорию. Сама директория и файлы в ней должны принадлежать пользователю **postgres** группы postgres.

```
mkdir -p /argus/pgsql/10.0  
chown -R postgres:postgres /argus/pgsql
```

Запускаем БД и проверяем data\_directory postgres (выполняется под УЗ postgres):

```
pg_ctl start psql  
SHOW data_directory;
```

Если все верно, ответ должен быть таким:

```
data_directory  
-----  
/argus/pgsql/10.0/data
```

**Теперь можно удалить /var/lib/pgsql/10.0**

## 1.2 Создание БД WFM CC

### 1.2.1 Установка утилиты управления БД.

1. Создать директории /home/postgres/dbctl и /home/postgres/dbctl/dumpdir.

```
# mkdir -p /home/postgres/dbctl/dumpdir
```

2. Распаковать в директорию /home/postgres/dbctl утилиту управления базой данных Argus.

```
# unzip dbctl.zip -d /home/postgres/dbctl
```

3. Установить пользователя postgres владельцем утилиты и сделать скрипт dbctl.sh исполняемым

```
# chmod +x /home/postgres/dbctl/dbctl.sh  
# chown -R postgres /home/postgres/dbctl
```

### 1.2.2 Создание системного пользователя БД и продуктивной БД Argus

1. Перейти в директорию /home/postgres/dbctl

```
# cd /home/postgres/dbctl
```

2. Прочитать инструкцию по использованию утилиты dbctl. Для этого необходимо запустить ее без параметров

```
# ./dbctl.sh  
Usage: ./dbctl.sh [list;dump;clone;create;restore;drop] params  
command:  
list - получить текущий список баз  
inituser password - создать системного пользователя argus_sys с указанным  
паролем password  
initdb databasename - создать чистую базу данных databasename без схем  
данных argus box  
initext databasename - активировать на базе данных databasename расширения  
PostgreSQL (должно выполняться после обновления чистой БД)  
dump databasename [filename] - снять дамп с базы databasename в файл
```

```
/home/postgres/dbctl/dumpdir/filename  
clone from_databasename to_databasename - клонировать базу "from" в "to" +  
будет снят дамп /home/postgres/dbctl/dumpdir/from_databasename_datetime.sql  
create databasename filename - создать базу databasename из файла  
/my/filename или /home/postgres/dbctl/dumpdir/filename  
restore databasename filename - восстановить базу databasename из файла  
/my/filename или /home/postgres/dbctl/dumpdir/filename  
drop databasename - удалить базу databasename
```

Из доступных операций нам будут интересны следующие: `inituser <password>`, `initdb <databasename>`, `initext <databasename>`

3. Создать системного пользователя `argus_sys` (имя системного пользователя не может быть изменено). В указанном ниже примере создания используется пароль `"th3yKn0w!"`. Вместо него нужно использовать любой другой пароль, в соответствии с принятыми в компании правилами безопасности. Вводимый пароль используется только в момент создания нового пользователя и нигде не сохраняется. После создания системного пользователя будет возможно авторизоваться в БД от имени `argus_sys` используя указанный при создании пользователя пароль. Важно, пароль необходимо указывать без кавычек и одним словом! Если в результате работы инструкции в консоль вывелось `"done"`, то инструкция выполнена успешно. Ниже приведен пример вызова инструкции и вывода результатов ее выполнения.

```
# ./dbctl.sh inituser th3yKn0w!  
Init argus_sys user  
create user  
CREATE ROLE  
done
```

### 1.2.3 Создание БД WFM CC

1. Создать базу данных. В приведенном примере в качестве имени БД указано `"argus"`. После создания БД будет доступна по указанному имени. В случае успешного выполнения инструкция выведет в консоль `"done"`

```
# ./dbctl.sh initdb argus  
Init DB  
databasename=argus  
CREATE DATABASE  
ALTER DATABASE  
GRANT  
CREATE SCHEMA  
done
```

2. Проверить успешность создания базы данных. Если предыдущие операции были выполнены успешно, то в консоль будет выведено название БД `"argus"`. В противном случае (ошибки при создании БД) консоль останется пустой.

```
# ./dbctl.sh list | grep argus  
argus
```

3. Создать схему `'dbm'`, для этого выполнить команду:



```
CREATE SCHEMA dbm;
```

4. Добавить в extensions расширение btree\_gist, для этого выполнить команду:

```
CREATE EXTENSION bs_tree;
```

## 2. Обновление БД WFM CC

Сделать резервную копию БД.

Распаковать архив с артефактами БД в любое удобное место.

Отредактировать параметры пакета обновления

```
# nano dbmaintain.properties
```

Необходимо указать параметры подключения к БД:

- a. **database.url** - указывается строка `jdbc:postgresql://<ip>:\:<port>/<dbname>`, где `<ip>` - хост базы данных; `<port>` - порт, по которому БД прослушивает входящие соединения, (по умолчанию 5432); `<dbname>` - имя базы данных, измените название БД с `testdb` на нужное (`prod`). Пример корректно указанного параметра

```
database.url=jdbc:postgresql://192.168.100.88\:5432/prod
```

- b. **databases.names** - необходимо указать имя базы данных, которую необходимо обновить. Важно, должно соответствовать имени БД из строки подключения. Например, `prod`.

Запустить пакет обновления на выполнение.

```
dbmaintain.sh updateDatabase
```

При успешном выполнении обновления в консоль будет выведена фраза "*The database has been updated successfully*"

```
14:05:22,674 INFO [CommandLine] - Loaded configuration from file
dbmaintain.properties
14:05:22,757 INFO [SimpleDataSource] - Creating data source. Driver:
org.postgresql.Driver, url: jdbc:postgresql://192.168.100.88:5432/prod, user:
argus_sys, password: <not shown>
...
13:56:38,385 INFO [DefaultDbMaintainer] - The database has been updated
successfully.
```

Прим.: в ОС семейства GNU/Linux необходимо исправить файл `dbmaintain.sh`, убрав `dos`-символы перевода строки. Сделать это можно с помощью утилиты `dos2unix`:

```
dos2unix dbmaintain.sh
```

Также может понадобиться добавление драйвера JDBC в переменные окружения пользователя (драйвер находится в каталоге `driver`, который находится в каталоге с артефактами БД):

```
export DBMAINTAIN_JDBC_DRIVER=/home/user/db/target/dbm/driver/postgresql-
9.4.1208.jar
```

### 3. Установка СП WFM CC

#### 3.1 Создание пользователя argus

Выполнить команду `useradd` для создания пользователя `argus`. Данная команда должна выполняться с правами суперпользователя.

```
# useradd argus
```

Установить пароль для пользователя `argus`. Данная команда должна выполняться с правами суперпользователя. После ввода команды `passwd` появится приглашение для ввода пароля созданного ранее пользователя `argus`. Вводимые символы при этом не отображаются. После ввода пароля будет предложено повторить пароль для того, чтобы избежать случайной опечатки при вводе. По завершению ввода пароля второй раз система сообщит об успешности выполнения операции. Пример:

```
# passwd argus
Смена пароля для пользователя argus.
Новый пароль:
Повторите ввод нового пароля:
passwd: все токены проверки подлинности успешно обновлены.
```

Проверить созданного пользователя. Для этого ввести команду:

```
# su - argus
```

Если команда выполняется с правами суперпользователя, то ввод пароля не требуется. Иначе, система предложит ввести пароль для пользователя `argus`. После успешного ввода пароля в консоли отобразится имя пользователя, для которого активен текущий сеанс. Для возврата к предыдущему пользователю используется сочетание клавиш `Ctrl+D`.

#### 3.2 Установка Java-машины

Java версии 8 доступна для скачивания со страницы:

```
http://www.oracle.com/technetwork/java/javase/downloads/index.html
```

Загрузка JDK 8u77 из архива сайта Oracle требует наличия зарегистрированной учетной записи.

Рекомендуется устанавливать Java-машину в каталог `/argus/jdk/jdk1.8.0_77` как JDK (java development kit), а не как JRE (java runtime environment), т.к. JDK предоставляет дополнительные инструменты для диагностики приложения.

Необходимо в файле `.bash_profile` (или `.profile`, в зависимости от того, какой файл используется в ОС) пользователя ОС (запускающего установочный файл СП или сам процесс СП) указать переменные:

- Переменную окружения **JAVA\_HOME**, в которой требуется прописать путь до JDK:

```
export JAVA_HOME=/argus/jdk/jdk1.8.0_77
```

- В переменной окружения **PATH** указать путь до **bin** каталога **JDK**:

```
export PATH=/argus/jdk/jdk1.8.0_77/bin:$PATH
```

Желательно для установки брать последнюю Java-машину, в которой учитывается отсутствие перехода на зимнее время в Российской Федерации.

Или же имеющаяся Java-машина должна быть обновлена с помощью **Java Time Zone Updater**. Дополнительную информацию

см. <http://www.oracle.com/technetwork/java/javase/tzupdater-readme-136440.html>

### 3.3 Данные часового пояса

Перед установкой СП необходимо убедиться, что в JDK установлены актуальные данные часового пояса.

Данные часового пояса проверяются и обновляются утилитой Java Time Zone Updater, которая поставляется в составе дистрибутива СП Аргус.

Необходимо скопировать файл tzupdater.jar из **КаталогУстановкиСП/tools/java** в каталог bin установленного пакета JDK.

Перед запуском TZUpdater необходимо остановить все экземпляры программного обеспечения (СП), которое использует JDK.

Чтобы успешно обновить данные часового пояса, нужно убедиться, что у УЗ (argus) есть достаточно прав для изменений в каталоге **JDK\_HOME/jre/lib**.

Для обновления данных часового пояса нужно перейти в каталог **JDK\_HOME/bin** и выполнить команду:

```
java -jar tzupdater.jar -u
```

Для проверки успешного обновления данных часового пояса необходимо скопировать файл TestWinterTime26102014.class из **КаталогУстановкиСП/tools/java** в каталог **JDK\_HOME/bin**.

Из каталога JDK\_HOME/bin выполнить команду:

```
java TestWinterTime26102014
```

При успешном обновлении данных часового пояса будет выведена таблица:

```
Local time Instant (hours since 01.01.1970 GMT)
2014-10-26 00:00:00 (+0400) 392852
2014-10-26 01:00:00 (+0400) 392853
2014-10-26 01:00:00 (+0300) 392854
2014-10-26 02:00:00 (+0300) 392855
2014-10-26 03:00:00 (+0300) 392856
```

До обновления данных часового пояса выводится таблица:

```
Local time Instant (hours since 01.01.1970 GMT)
2014-10-26 00:00:00 (+0400) 392852
2014-10-26 01:00:00 (+0400) 392853
2014-10-26 02:00:00 (+0400) 392854
2014-10-26 03:00:00 (+0400) 392855
2014-10-26 04:00:00 (+0400) 392856
```

Дата хоста не имеет значения для этого теста, но на хосте должен быть часовой пояс Europe/Moscow.

Если данные часового пояса не обновились, то следует повторить процедуру обновления.

### 3.4 Локаль и кодировка операционной системы

Операционная система должна поддерживать кодировку UTF-8 и в ней должны быть установлены форматы времени, даты и чисел (locale) в соответствии с используемыми пользователями языковыми стандартами (при установке в Российской Федерации нужна локаль ru\_RU.UTF-8).

В операционных системах Linux проверка локали осуществляется командой locale. Ниже показан вывод команды на правильно настроенном сервере:

```
$ locale
LANG=ru_RU.UTF-8
LC_CTYPE="ru_RU.UTF-8"
LC_NUMERIC="ru_RU.UTF-8"
LC_TIME="ru_RU.UTF-8"
LC_COLLATE="ru_RU.UTF-8"
LC_MONETARY="ru_RU.UTF-8"
LC_MESSAGES="ru_RU.UTF-8"
LC_PAPER="ru_RU.UTF-8"
LC_NAME="ru_RU.UTF-8"
LC_ADDRESS="ru_RU.UTF-8"
LC_TELEPHONE="ru_RU.UTF-8"
LC_MEASUREMENT="ru_RU.UTF-8"
LC_IDENTIFICATION="ru_RU.UTF-8"
LC_ALL=
```

### 3.5 Синхронизация времени

Для корректной работы СП Аргус требуется, чтобы была обеспечена синхронизация времени ОС СП и ОС, на которой установлена БД Аргус.

### 3.6 Максимальное число открытых файлов и сокетов

Необходимо увеличить ограничение на максимальное число открытых файлов и сокетов по сравнению с дефолтным. Сервер удерживает открытыми множество файлов. Сервер приложений открывает сокет, устанавливая исходящие сетевые соединения к базе данных Аргус и к серверам сторонних приложений/баз данных, с которыми поддерживается интеграция. Сервер приложений открывает сокет, принимая входящие соединения от браузеров пользователей. Требуемое максимальное число открытых файлов и сокетов рассчитывается по формуле:

$$\max_{\text{открытых сокетов и файлов}} = \max_{\text{конкурентных пользователей}} * 20$$

*Windows.*

В зависимости от числа пользователей и версии операционной системы в корректировке могут нуждаться следующие настройки TCP/IP в разделе реестра:

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\services\Tcpip\Parameters]:
- MaxFreeTcbs
- MaxHashTableSize
```

Как правило, не нуждаются в корректировке:

```
- TcpTimedWaitDelay
- KeepAliveTime
```

Linux.

Параметры настройки ядра. Посмотреть текущие настройки уровня ОС:

```
cat /proc/sys/fs/file-max
```

Установить значение параметра уровня ОС в файле /etc/sysctl.conf:

```
sysctl -w fs.file-max=102400
```

После изменения параметров – активировать изменения в ядре без перезагрузки хоста:

```
sysctl -p /etc/sysctl.conf
```

Посмотреть настройки для текущего пользователя (УЗ) ОС:

```
ulimit -n
```

Установить значение для УЗ в файле /etc/security/limits.conf:

```
argus soft nofile 100000  
argus hard nofile 100000
```

### 3.7 Максимальное число запущенных процессов

Необходимо увеличить ограничение на максимальное число запущенных процессов по сравнению с дефолтными. При работе сервера без дополнительных http-портов, количество потоков сервера может находиться в пределах 200-1000, а добавление очередного дополнительного http-порта влечет создание пула на 256 потоков. Таким образом, в ОС у пользователя, под которым запускается сервер приложений, следует увеличить максимальное число запущенных процессов (max-user-processes). Значение параметра рассчитывается по формуле:

$$\text{max-user-processes} \geq 1000 + \text{additional-ports-count} * 256$$

где additional-ports-count - количество дополнительных http-портов (argus.io.additional-http-ports)

Посмотреть настройки для текущего пользователя (УЗ) ОС:

```
ulimit -a | grep processes
```

Установить значение для УЗ в файле /etc/security/limits.conf:

```
argus soft nproc 4000  
argus hard nproc 4000
```

### 3.8 Дополнительные рекомендуемые настройки ОС

Linux.

Необходимо увеличить значение **socket send/receive buffer** по сравнению с дефолтными в файле **/etc/sysctl.conf**:

```
default socket receive buffer:    sysctl -w net.core.rmem_default=262144  
default socket send buffer:       sysctl -w net.core.wmem_default=262144  
max socket receive buffer:        sysctl -w net.core.rmem_max=262144  
max socket send buffer size:      sysctl -w net.core.wmem_max=262144
```

После изменения параметров – активировать изменения в ядре без перезагрузки хоста:

```
sysctl -p /etc/sysctl.conf
```

### 3.9 Имя хоста

Имя хоста не должно содержать символ подчеркивания. Проверить имя можно следующей командой:

```
uname -n
```

### 3.10 Создание структуры каталогов для артефактов сервера приложений

Рекомендуемая структура каталогов для размещения ПО Аргус:

/argus	Каталог содержит в себе СП и дополнительное окружение для поддержания его работы. Для размещения рекомендуется выделять отдельный раздел диска.
/argus/distr	Каталог содержит в себе дистрибутивы и пакеты установок СП Аргус и дополнительного ПО. Например: <i>/argus/distr/1450/argus-dist-3.14.0.16954-customer.jar</i> <i>/argus/distr/1450/customer.prod.argus-app-01.properties</i> <i>/argus/distr/jdk_8u77_linux_x64.gz</i>
/argus/jboss_arch	Каталог хранит в себе резервные копии СП (формат копии: <i>ддммгггг/jboss_prod</i> ). Например: <i>/argus/jboss_arch/16012016/jboss_prod</i>
/argus/jboss_prod	Каталог установки СП.
/argus/jdk	Каталог с установленной JDK.
/argus/jdk/jdk1.8.0_77	DK версии 8 обновления 77. Требуется для работы СП Аргус.
/argus/nmon	Каталог, содержащий отчеты nmon о производительности системы (формат: <b>&lt;сетевое_имя_узла&gt;_ггммдд_0101.nmon</b> ) и их архивы (формат: <b>&lt;сетевое_имя_узла&gt;_ггммдд_0101.nmon.gz</b> ).
/argus/scripts	Вспомогательные скрипты.
/argus/tmp	Каталог для временных файлов СП

Назначить пользователя **argus** владельцем директорий, например:

```
# chown -R argus /argus
```

Скопировать дистрибутив сервера приложений в директорию /argus/distr.

### 3.11 Создание файла конфигурации для установки дистрибутива основного сервера приложений

На хосте предварительно заведена УЗ **argus** с правами на каталог **/argus**. Создать в директории **/argus/distr** текстовый файл **wfmcc-main.properties** со следующим содержимым:

```
INSTALL_PATH=/argus/jboss_prod/main-0.6.1

jboss.bind.address=<server_address>
jboss.socket.binding.port-offset=0

argus.app.memory.max-size=0
argus.app.debug-mode.enabled=true

argus.app.admin.user=<admin_username>
argus.app.admin.pass=<admin_password>
argus.java.home.path=/argus/jdk/jdk1.8.0_181
argus.app.security-mode.enabled=false

argus.db.address=<db_address>
argus.db.port=<db_port>
argus.db.name=<db_name>
argus.db.user=<db_user>
argus.db.pass=<pd_pass>

argus.system.ds.max-pool-size=50

argus.mail.enabled=true
argus.mail.smtp.user=<mail_username>
argus.mail.smtp.pass=<mail_userpass>
argus.mail.smtp.port=25
argus.mail.smtp.host=<mail_host>
argus.mail.smtp.ssl.enabled=false
argus.mail.smtp.starttls.enabled=false
```

Назначение основных свойств приведены в таблице ниже

Наименование свойства	Назначение свойства
INSTALL_PATH	Директория, в которую будет установлен дистрибутив сервера приложений, рекомендуется <b>/argus/jboss_prod/</b>
jboss.bind.address	Адрес сервера приложений, например, 192.168.100.133 для запуска слушателя только на этом интерфейсе или 0.0.0.0 для запуска слушателя на всех доступных на хосте интерфейсах, включая localhost
jboss.socket.binding.port-offset	Смещение портсета. По умолчанию смещение равно 0, в результате чего основной сервер приложений доступен по порту 8080, личный кабинет по порту 8190, консоль администратора сервера приложений по порту 9990. Если указать смещение, например, 5, то портсет сместится на это число, т.е. 8080 -> 8085, 8190 -> 8195, 9990 -> 9995. Возможны отрицательные смещения.

Наименование свойства	Назначение свойства
argus.app.memory.max-size	Максимальный размер оперативной памяти, доступный серверу приложений. Рекомендуется указывать такой размер ОП, чтобы сервер приложений не свопился.
argus.app.debug-mode.enabled	Если true, то сервер приложений будет запущен в режиме отладки с расширенным логгированием.
argus.app.admin.user	Имя администратора сервера приложений. При установке дистрибутива будет создана учетная запись администратора для управления сервером приложений из консоли администрирования сервера приложений, доступной по порту 9990. Пример, admin
argus.java.home.path	Путь к установленной JDK
argus.app.security-mode.enabled	Режим безопасности # false, если стандартный режим безопасности, true повышенный режим безопасности (в режиме повышенной безопасности переконфигурирование не доступно, отсутствует каталог .config)
argus.app.admin.pass	Пароль администратора сервера приложений. Пример, th3yKn0w!
argus.db.address	Адрес сервера БД. Например, 192.168.100.180
argus.db.port	Порт, прослушиваемый сервером БД. Для PostgreSQL по умолчанию 5432
argus.db.name	Имя базы данных, созданной ранее на шаге 5.3.4, например, argus
argus.db.user	Пользователь, от имени которого сервер приложений будет взаимодействовать с базой данных. На текущий момент может быть только argus_sys.
argus.db.pass	Пароль пользователя, от имени которого сервер приложений будет взаимодействовать с базой данных. Необходимо указать пароль, используемый для создания системного пользователя БД на шаге 5.3.3
argus.system.ds.max-pool-size	Максимальный размер пула соединений с базой данных. Зависит от настроек БД. Рекомендуется установить значение не менее планируемого количества пользователей + 10, и не более 100. Например, для 10 активных одновременно пользователей рекомендуется указать значение 25.
argus.mail.enabled	Если true, то будет включен модуль рассылки электронных сообщений. Если false, то все остальные настройки argus.mail.* не обрабатываются
argus.mail.smtp.user	Имя пользователя для авторизации на почтовом сервере, например, <a href="mailto:noreply@thecompany.com">noreply@thecompany.com</a>
argus.mail.smtp.pass	Пароль пользователя для авторизации на почтовом сервере, например, th3yKn0w!
argus.mail.smtp.port	Порт почтового сервера, по умолчанию 25
argus.mail.smtp.host	Хост почтового сервера, например, <a href="mailto:mail.thecompany.com">mail.thecompany.com</a>
argus.mail.smtp.ssl.enabled	Если true, то будет использовано ssl шифрование. Зависит от настроек почтового сервера



Наименование свойства	Назначение свойства
argus.mail.smtp.starttls.enabled	Если true, то будет включен STARTTLS. Зависит от настроек почтового сервера

### 3.12 Установка СП WFM CC

Установку СП можно запустить в трех режимах:

- В режиме использования конфигурационного файла

Инсталлятор не будет запрашивать параметры установки, значения параметров будут считаны из конфигурационного файла специфичной конфигурации заказчика (например, **customer.prod.argus-app-01.properties**). Установка в этом режиме возможна только, если конфигурационный файл специфичной конфигурации включен в состав пакета установки.

Перед установкой следует проверить параметры, указанные в файле специфичной конфигурации заказчика. Конфигурационный файл специфичной конфигурации от одного хоста - можно использовать и при установке СП на другой хост, внося необходимые изменения, следуя инструкции, приведенной в файле или учитывая описание параметров. В случае, если два экземпляра СП на хостах: хост1 и хост2 работают с одной и той же БД, рекомендуется во избежание конфликтов указывать различные значения в параметре *argus.db.user*.

Пример запуска инсталлятора:

```
java -jar argus-dist-[Версия].[Номер_сборки]-[Наименование_Заказчика].jar  
-  
options [Наименование_Заказчика].[Продуктив_или_тест].[Наименование_СП].properties
```

- В режиме консольного интерфейса

Инсталлятор будет запрашивать параметры установки в консоли.

Пример запуска инсталлятора:

```
java -jar argus-dist-[Версия].[Номер_сборки]-[Наименование_Заказчика].jar -  
console
```

- В режиме графического интерфейса

Инсталлятор будет запрашивать параметры установки, используя графический оконный интерфейс. На Linux данный режим возможен, только если установлена подсистема графического интерфейса (x-windows).

Пример запуска инсталлятора:

```
java -jar argus-dist-[Версия].[Номер_сборки]-[Наименование_Заказчика].jar
```

#### 4. Обновление СП WFM CC

Загрузите дистрибутив СП, распакуйте и поместите в любое удобное место. Файл дистрибутива имеет название вида *argus-dist-0.1.0.65472-ccwfm.jar*.

Выгрузите дистрибутив на удаленный сервер любым удобным способом (scp/sftp/etc) в каталог */argus/distr/*.

Для удобства можно предварительно удалить старую версию дистрибутива.

#### **Предупредите всех заинтересованных лиц о проведении обновления!**

Подключитесь к серверу, на котором развернут СП по ssh:

```
ssh root@ip_addr
```

где *ip\_addr* - адрес нужного сервера.

Сделайте себя пользователем *argus* командой

```
su - argus
```

Выполните команду для остановки СП:

```
/argus/jboss_prod/bin/runjboss.sh stop
```

Убедитесь, что СП был остановлен, для этого выполните команду:

```
ps aux | grep standalone
```

Вывод команды должен содержать только одну строку.

Выполните бэкап старой версии СП: переименуйте каталог */argus/jboss\_prod*, либо переместите его.

Перейдите в каталог */argus/distr/* и выполните команду для установки СП:

```
java -jar argus-dist- -options prod.properties
```

где *-prod.properties* — файл с предустановками СП (прим.: копирование команды в консоль может привести к невыполнению этой команды, используйте ручной ввод). Передвиньте курсор на конец "argus-dist-", а затем нажмите Tab для автодополнения названия файла дистрибутива.

В случае необходимости, отредактируйте параметры СП в файле */argus/jboss\_prod/standalone/configuration/standalone.xml*, например, адрес и порт БД.

Запуск сервера приложений должен быть выполнен от имени пользователя *argus*. Под пользователем *root* сервер приложений не запустится по соображениям безопасности. Для запуска СП выполните команду:

```
/argus/jboss_prod/bin/runjboss.sh start
```

Убедитесь в успешном запуске СП. Например, выполните в консоли по ssh:

```
tail /argus/jboss_prod/standalone/log/last_boot_errors.log
```

После выполнения команды *tail* в консоль начнет выводиться лог запуска сервера приложений. При успешном запуске в итоге будет выведено

```
12:28:52,068 INFO [LastBootErrorsFinalizer] (EE-ManagedThreadFactory-default-Thread-1) Сервер запущен
```

Либо успешность запуска можно проверить командой:

```
$ cat /box/app/main-0.6.1/standalone/log/last_boot_errors.log
```

При наличии ошибок запуска они будут выведены в консоль. Если в консоль будет выведено только сообщение "Сервер запущен", то запуск прошел успешно.

## 5. Настройка e-mail оповещений

### 5.1 Предварительные настройки

На SMTP-сервере предварительно должен быть создан пользователь, который будет использован для отправки оповещений.

### 5.2 Настройка файла конфигурации

Для настройки e-mail оповещений пользователей системы WFMCC в файле конфигурации «standalone.xml» заполняются следующие параметры:

argus.mail.enabled	Параметр для включения и отключения e-mail оповещений. Установите значение true для включения или false для отключения e-mail оповещений.
argus.mail.smtp.user	Полное имя пользователя для отправки e-mail оповещений, включая логин, @ и домен.
argus.mail.smtp.pass	Пароль пользователя argus.mail.smtp.user. При отсутствии аутентификации (argus.mail.smtp.auth.enabled=false) пароль не используется.
argus.mail.smtp.port	Порт SMTP-сервера.
argus.mail.smtp.host	Хост SMTP-сервера.
argus.mail.smtp.ssl.enabled	Установите значение true при использовании SSL или false при его отсутствии.
argus.mail.smtp.starttls.enabled	Установите значение true при использовании STARTTLS или false при его отсутствии.
argus.mail.smtp.auth.enabled	Установите значение true при наличии аутентификации или false при ее отсутствии.

В случае использования NTLM-аутентификации, блок subsystem должен содержать **username/password**:

```
<subsystem xmlns="urn:jboss:domain:mail:2.0">
  <mail-session name="default" debug="true" jndi-name="java:jboss/mail/Default"
    from="{argus.mail.from}">
    <smtp-server outbound-socket-binding-ref="mail-smtp"
      ssl="{argus.mail.smtp.ssl.enabled}" tls="{argus.mail.smtp.starttls.enabled}"
      username="{argus.mail.smtp.user}" password="{argus.mail.smtp.pass}"/>
    </mail-session>
  </subsystem>
```

### 5.3 Перезапуск сервера приложений

После внесения изменений в файл конфигурации необходимо перезапустить Docker-контейнер сервера приложений, используя команду:

```
docker restart
```

## 6. Логирование

Системные логи всех компонентов КТС см. в каталоге /var/log ..

### 6.1 Логирование СП WFMCC (WildFly)

На СП существует 4 вида логов:

- access.log
- server.log
- webservices.log
- bugreports

Более подробно, чем отличаются данные логи и какая информация в них находится.

#### 6.1.1 Access.log

Логи обращения к СП. Содержат ошибки недоступности сервиса (например, 503 "Service unavailable").

Формат отображения лога задается в настройках СП:

```
<access-log pattern="%t %a &quot;{%i,X-Forwarded-For}&quot; %u [%I] %S %r %s {%r,javax.servlet.error.status_code} %b %D {%i,Referer} &quot;{%i,User-Agent}&quot;" prefix="access."/>
```

Где:

```
%a - Remote IP address
%A - Local IP address
%b - Bytes sent, excluding HTTP headers, or '-' if no bytes were sent
%B - Bytes sent, excluding HTTP headers
%h - Remote host name
%H - Request protocol
%l - Remote logical username from identd (always returns '-')
%m - Request method
%p - Local port
%q - Query string (excluding the '?' character)
%r - First line of the request
%s - HTTP status code of the response
%t - Date and time, in Common Log Format format
%u - Remote user that was authenticated
%U - Requested URL path
%v - Local server name
%D - Time taken to process the request, in millis
%T - Time taken to process the request, in seconds
%I - current Request thread name (can compare later with stacktraces)
```

### 6.1.2 Server.log

Общие логи СП и всех пользователей. Содержат ошибки, информацию о загрузке.

### 6.1.3 Webservices.log

Request-response входящих (inbound) и исходящих (outbound) SOAP-вызовов. В данных логах содержатся интеграционные запросы.

Настройки логирования webservices.log.

В конфигурационном файле standalone.xml для продуктивных систем в качестве параметра указано значение:

```
<periodic-rotating-file-handler name="WEBSERVICES_LOG" autoflush="false">
```

В этом случае запись в webservices.log будет происходить с задержкой только после заполнения соответствующего буфера.

При тестировании можно указать параметр autoflush="true" и получать логирование в webservices.log незамедлительно.

После внесения изменений в standalone.xml необходим перезапуск СП!

### 6.1.4 Bugreports

Детальные сессионные логи (хранятся в виде архивов) каждого пользователя в отдельности. Содержат подробности инцидентов, стек ошибок, sql-запросы. Наиболее полные (по своей информативности) логи.

Формируются двумя способами:

- автоматическая генерация - при ошибках web-services и UI;
- ручная генерация - при нажатии кнопки "создать отчет об ошибке"

## 6.2 Логирование БД (PostgreSQL)

Настройки параметров логирования БД PostgreSQL осуществляется в файле:

```
<каталог_установки_БД>/data/postgresql.conf
```

Примеры:

- log\_duration = on # Включает логирование запросов
- log\_min\_duration\_statement = 100 # Устанавливает минимальное время (в миллисекундах) выполнения запроса, который попадает в лог
- log\_checkpoints = on # в лог будет попадать информация о checkpoint'ах
- log\_connections = on # в лог будет попадать информация о новых подключениях к PostgreSQL
- log\_disconnections = on # в лог будет попадать информация об отключениях от PostgreSQL
- log\_lock\_waits = on # в лог будет попадать информация о длительных ожиданиях сессий в попытке получить блокировку на объекте
- log\_temp\_files = 1024 # в лог будет попадать информация о временных файлах свыше 1024K (1M)
- log\_autovacuum\_min\_duration = 0
- log\_line\_prefix = '%t [%p]: [%l-1] unname=%u,dbname=%d,appname=%a,rhost=%h ' # формат логов

- log\_directory = 'pg\_log' # место хранения лог-файлов БД
- log\_filename = 'postgresql-%Y%m%d.log'
- log\_file\_mode = 0640

### 6.3 Логирование СП интеграции (Tomcat)

Файл конфигурации находится внутри jar-файла СП, логи находятся в

```
<каталог_установки_СП>/integration/log
```

и

```
<каталог_установки_СП>/sync/log
```

Более подробную информацию по логированию можно найти тут - <https://docs.spring.io/spring-boot/docs/current/reference/html/boot-features-logging.html>.

## 7. Необходимые регулярные процедуры по сопровождению системы WFM CC

### 7.1 Архивация и удаление журналов операций

При работе ПО Аргус – в лог-файлах формируется большое количество отладочной информации.

Лог файлы необходимы для проведения контраварийных работ, поэтому должны храниться на хосте какое-то время (как правило это 3-5 дней).

Необходимо следить за местом на дисковом пространстве, чтобы формируемые лог-файлы не заполняли его полностью (что грозит остановкой сервиса), а также своевременно удалять лог-файлы по истечению оговоренного срока хранения.

Для экономии места на диске лог-файлы рекомендуется архивировать.

Логи складываются по пути:

```
<Каталог_установки_СП>/standalone/log/
```

Удалять и сжимать лог-файлы можно скриптами, прописав их в планировщик задач, например в **cron** (OS Linux).

Пример скрипта архивации и удаления логов:

```
#!/bin/bash
#Скрипт архивации логов. Хранит архивы N дней
export PATH=/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin
export argus_logs=/Data/jboss_prod/standalone/log
export logfile=/Data/scripts/arch_logs.log
export bugreports=/Data/jboss_prod/standalone/log/bugreports
date_now=$(date +%Y\-%m\-%d);
echo
'=====
>> $logfile
echo "" >> $logfile
echo "Started at `date`" >> $logfile
echo " >> $logfile
ls -al $argus_logs >> $logfile
echo " >> $logfile

#Переменные
OLD_LOGS=$(date -d "-1 day" +"%Y-%m-%d" )
```

```
echo $OLD_LOGS
BUGREPORTS=$(date -d "-1 day" +"%Y.%m.%d")
echo $BUGREPORTS
OLD_ARCH_LOGS=$(date -d "-30 day" +"%Y-%m-%d")
echo $OLD_ARCH_LOGS
OLD_ARCH_BUGREPORTS=$(date -d "-30 day" +"%Y.%m.%d")
echo $OLD_ARCH_BUGREPORTS

#Создадим архивы старше 1 дня
find /Data/jboss_prod/standalone/log -name "*$OLD_LOGS*" | xargs tar -cvzf
/Data/jboss_logs_arch/$OLD_LOGS.logs.tar.gz
find /Data/jboss_prod/standalone/log/bugreports -name "*$BUGREPORTS*" | xargs tar
-cvzf /Data/jboss_logs_arch/$BUGREPORTS.logs.bugreports.tar.gz

#Удалим все файлы старше 1 дня из каталога логов СП
find /Data/jboss_prod/standalone/log -name "*$OLD_LOGS*" ! -name "gcstats.log*" -
exec rm -rf {} \;
find /Data/jboss_prod/standalone/log/bugreports -name "*$BUGREPORTS*" -exec rm -
rf {} \;

#Удаление архивов, которые лежат дольше, чем кол-во дней хранения логов
find /Data/jboss_logs_arch -mtime +31 -type f -exec rm -rf {} \;

#find /Data/jboss_logs_arch -name "*$OLD_ARCH_LOGS*" -exec rm -rf {} \;

#find /Data/jboss_logs_arch -name "*$OLD_ARCH_BUGREPORTS*" -exec rm -rf {} \;
echo "Finished at `date`" >> $logfile
echo
'=====
>> $logfile
```

## 7.2 Настройка очистки архивов БД

Для очистки архива от старых записей пользуемся встроенным инструментом pg\_archivecleanup из-под пользователя postgres.

В директории /argus/pgsql/10/data создаём скрипт archive\_cleanup.sh с правами на исполнение владельцем postgres.

Такого содержания:

```
#!/bin/bash

ARCHIVEDIR='/argus/pgsql/10/archive'
CHKPOINT=$(find $ARCHIVEDIR -type f -not -name "*.*" -printf '%f\n' | sort -r |
head -1)
cd $ARCHIVEDIR
/usr/pgsql-10/bin/pg_archivecleanup -d $ARCHIVEDIR $CHKPOINT
2>>/argus/pgsql/10/data/cleanup.log
```

Далее необходимо скрипту выдать гранты и права:

```
chown postgres:postgres /argus/pgsql/10/data/archive_cleanup.sh
chmod 774 /argus/pgsql/10/data/archive_cleanup.sh
```

Скрипт находит последний созданный WAL в папке с архивами, подставляет его в команду postgres pg\_archivecleanup, которая удаляет все архивы до указанного в команде.

Весь вывод pg\_archivecleanup идёт в /argus/pgsql/10/data/cleanup.log.

Выполнение скрипта запланировано в cron с частотой раз в час:

```
crontab -e
```

Вносим:

```
0 * * * * /argus/pgsql/10/data/archive_cleanup.sh  
:wq
```

Поскольку primary и standby могут меняться местами, скрипт очистки и планирования в cron устанавливается на каждом из хостов с БД.