

**ПО DOCUMINO.AРХИВ**

**Инструкция по установке экземпляра программного обеспечения**

Листов 27

2025

## Содержание

1. ТЕРМИНЫ, СОКРАЩЕНИЯ И ОПРЕДЕЛЕНИЯ.....	3
2. ВВЕДЕНИЕ.....	4
2.1. Назначение документа.....	4
2.2. Краткое описание возможностей ПО Documino.Архив.....	4
2.3. Требование к эксплуатирующему персоналу.....	4
3. ОПИСАНИЕ УСТАНОВОК.....	5
3.1 Подготовка операционных систем.....	5
3.1.1 Сервер приложения.....	5
3.1.2 Сервер содержания.....	7
3.1.3 Сервер СУБД.....	9
3.2. Установка и настройка сервера СУБД.....	12
3.3. Установка основных компонент ПО Documino.Архив.....	15
3.3.1 Установка сервера содержимого.....	15
3.3.2 Установка сервера полнотекстового поиска.....	16
3.3.3 Установка сервера извлечения текста.....	17
3.3.4 Установка сервера кэширования.....	18
3.4 Установка экземпляра ПО Documino.Архив. ....	20
3.4.1 Состав архива установки. ....	20
3.4.2 Установка архива экземпляра программного обеспечения.....	23
3.4.3 Подготовка конфигурационных файлов установки. ....	23
3.4.4 Подготовка конфигурации сервера приложения.....	25
3.4.5 Установка/обновление экземпляра программного обеспечения. ....	27

## 1. ТЕРМИНЫ, СОКРАЩЕНИЯ И ОПРЕДЕЛЕНИЯ

В настоящем документе применены следующие сокращения и термины с соответствующими определениями:

Термин/сокращение	Определения
БД	База данных
ПО	Программное обеспечение
СУБД	Система управления базами данных
XQL (eXtended Query Language)	Текстовый язык доступа к данным ПО Documino.Архив. Представляет собой SQL-подобный язык, посредством которого клиентский код взаимодействует с платформой
ПО Documino.Архив	Программное обеспечение Documino.Архив, разработанная компанией «АйДи – Технологии управления»

## 2. ВВЕДЕНИЕ

### 2.1. Назначение документа

Данный документ содержит последовательность действий и указания по установке программной ПО Documino.Архив.

### 2.2. Краткое описание возможностей ПО Documino.Архив

ПО Documino.Архив обладает следующими основными возможностями:

- свой язык доступа к данным – XQL, выражения которого конвертируются в SQL запросы целевой СУБД;
- встроенная модель безопасности – поддержка пользователей, групп, ролей и прав доступа;
- версионность;
- хранение и доступ к контенту;
- возможность реализации на языке Java плагинов любой сложности, сделав их частью Documino.Архив, расширяющие его функциональность.
- поддержка различных схем аутентификации.

### 2.3. Требование к эксплуатирующему персоналу

Для установки, настройки и обновления платформы и всех ее компонент эксплуатирующий персонал (системный инженер) должен обладать опытом:

- работы с ОС Linux на уровне системного инженера;
- установки и настройки СУБД PostgreSQL версии 9.6 и выше;
- установки и настройки сервера приложений Apache Tomcat версии 8 и выше;
- установки и настройки сервера Elasticsearch версии 5 и выше;
- установки и настройки сервера Nginx версии 1.10 и выше;
- установки и настройки сервера Redis версии 3.2 и выше;
- установки и настройки сервера JackRabbit версии 2.14 и выше;
- установки и настройки Java-приложений.

### 3. ОПИСАНИЕ УСТАНОВОК

Для установки платформы необходимо скачать и настроить следующие виды ПО:

1. СУБД PostgreSQL  
<https://www.postgresql.org/download/>
2. Сервер хранения контента  
<https://archive.apache.org/dist/jackrabbit/2.14.2/jackrabbit-standalone-2.14.2.jar>
3. Сервер полнотекстового индексирования и поиска  
<https://www.elastic.co/downloads/past-releases>
4. Сервер извлечения содержимого  
<http://apache-mirror.rbc.ru/pub/apache/tika/tika-server-1.16.jar>
5. Сервер кэширования  
<https://redis.io/download>
6. Сервер приложений  
<https://tomcat.apache.org/download-80.cgi>
7. Oracle Java  
<http://www.oracle.com/technetwork/java/javase/downloads/index.htm>
8. Прокси-сервер  
<http://nginx.org/>
9. В состав дистрибутивов входит установочный архив экземпляра программного обеспечения **documino: documino\_install.rar**  
Ссылка для скачивания дистрибутива предоставляется по запросу на электронную почту: [mail@id-mt.ru](mailto:mail@id-mt.ru).

#### 3.1 Подготовка операционных систем

##### 3.1.1 Сервер приложения

Выполнить обновление операционной системы до актуальной версии.

Типовая настройка параметров ограничения ресурсов процессам операционной системы задается в конфигурационном файле /etc/security:

apache soft	nofile	65536
apache hard	nofile	65536
apache soft	nproc	65536
apache hard	nproc	65536

Типовая настройка параметра ядра /etc/sysctl.conf, параметры необходимо переопределить с учетом фактических ресурсов:

fs.aio-max-nr = 1048576

```
fs.file-max = 6815744
kernel.shmall = 2097152
kernel.shmmni = 65536
kernel.sem = 250 32000 100 128
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
kernel.msgmni=16384
kernel.msgmax=65536
kernel.msgmnb=65536
net.ipv6.conf.all.disable_ipv6 = 1
kernel.shmmax = 4294967295
net.ipv4.tcp_moderate_rcvbuf = 0
net.ipv4.tcp_window_scaling = 0
net.ipv4.tcp_sack = 0
net.ipv4.icmp_echo_ignore_broadcasts = 0
net.ipv4.icmp_ignore_bogus_error_responses = 0
vm.swappiness = 1
vm.dirty_background_ratio = 3
vm.dirty_ratio = 80
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
vm.min_free_kbytes=100000
vm.admin_reserve_kbytes = 60192
kernel.sysrq = 1
```

Необходимо заранее подготовить раздел для последующей установки серверов приложения:

```
fdisk /dev/hdd
```

В случае использования LVM создать необходимые компоненты:

```
sudo pvcreate /dev/hdd
```

```
sudo vgcreate u01 /dev/hdd
```

```
sudo lvcreate -l+100%FREE -n lv_u01_1 u01
```

```
sudo mkfs.ext4 /dev/u01/lv_u01_1
```

Сделать автоматическое монитрование раздела:

```
/etc/fstab:
```

```
/dev/u01/lv_u01_1          /u01    ext4    defaults,noatime    0
```

0

```
sudo mount /u01
```

Создать необходимых пользователей, директории:

```
sudo groupadd -g 2023 apache
```

```
sudo useradd -u 2022 -g 2023 apache
```

```
sudo mkdir -p /u01/app/tomcat8
```

```
sudo chown -R apache: apache /u01/app
```

Установить java последней версии:

Скачать официальный дистрибуит java.

Выполнить установку:

```
sudo tar -xzf jdk1.8.0_152-linux-x64.tar.gz -C /usr/local/java8/  
alternatives --install /usr/bin/java java
```

```
/usr/java/jdk1.8.0_152/bin/java 1
```

```
sudo alternatives --config java
```

### 3.1.2 Сервер содержания

Выполнить обновление операционной системы до актуальной версии.

Настройка параметров ограничения ресурсов процессов  
/etc/security:

```
jackrabbit soft  nofile      65536
```

```
jackrabbit hard  nofile      65536
```

```
jackrabbit soft  nproc       65536
```

```
jackrabbit hard  nproc       65536
```

Настройка параметра ядра /etc/sysctl.conf:

```
fs.aio-max-nr = 1048576
```

```
fs.file-max = 6815744
kernel.shmall = 2097152
kernel.shmmni = 65536
kernel.sem = 250 32000 100 128
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
kernel.msgmni=16384
kernel.msgmax=65536
kernel.msgmnb=65536
net.ipv6.conf.all.disable_ipv6 = 1
kernel.shmmax = 4294967295
net.ipv4.tcp_moderate_rcvbuf = 0
net.ipv4.tcp_window_scaling = 0
net.ipv4.tcp_sack = 0
net.ipv4.icmp_echo_ignore_broadcasts = 0
net.ipv4.icmp_ignore_bogus_error_responses = 0
vm.swappiness = 1
vm.dirty_background_ratio = 3
vm.dirty_ratio = 80
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
vm.min_free_kbytes=100000
vm.admin_reserve_kbytes = 60192
kernel.sysrq = 1
```

Подготовить каталог и раздел для последующей установки серверов приложения:

```
sudo mkdir /u01
fdisk /dev/hdd
```

В случае использования LVM создать необходимые компоненты:

```
sudo pvcreate /dev/hdd
```



```

sudo vgcreate u01 /dev/hdd
sudo lvcreate -l+100%FREE -n lv_u01_1 u01
sudo mkfs.ext4 /dev/u01/lv_u01_1
/etc/fstab:
/dev/u01/lv_u01_1          /u01          ext4          defaults,noatime
0 0

sudo mount /u01
Создать необходимых пользователей, директории:
sudo groupadd -g 2031 jackrabbit
sudo useradd -u 2031 -g 2031 jackrabbit
sudo mkdir -p /u01/jackrabbit/{bin,config,logs,repository}
sudo chown -R jackrabbit:jackrabbit /u01/jackrabbit

```

Установить java последней версии:

Скачать официальный дистрибуит java.

Выполнить установку:

```

sudo tar -xzvf jdk1.8.0_152-linux-x64.tar.gz -C /usr/local/java8/
alternatives --install /usr/bin/java java
/usr/java/jdk1.8.0_152/bin/java 1
sudo alternatives --config java

```

### 3.1.3 Сервер СУБД

Выполнить обновление операционной системы до актуальной версии.

Настроить параметры ограничения ресурсов процессов  
/etc/security:

```

postgres soft  nofile      65536
postgres hard  nofile      65536
postgres soft  nproc       65536
postgres hard  nproc       65536

```

Настройка параметра ядра /etc/sysctl.conf:

```

fs.aio-max-nr = 1048576
fs.file-max = 6815744

```

```
kernel.shmall = 2097152
kernel.shmmni = 65536
kernel.sem = 250 32000 100 128
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
kernel.msgmni=16384
kernel.msgmax=65536
kernel.msgmnb=65536
net.ipv6.conf.all.disable_ipv6 = 1
kernel.shmmax = 4294967295
net.ipv4.tcp_moderate_rcvbuf = 0
net.ipv4.tcp_window_scaling = 0
net.ipv4.tcp_sack = 0
net.ipv4.icmp_echo_ignore_broadcasts = 0
net.ipv4.icmp_ignore_bogus_error_responses = 0
vm.swappiness = 1
vm.dirty_background_ratio = 3
vm.dirty_ratio = 80
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
vm.min_free_kbytes=100000
vm.admin_reserve_kbytes = 60192
kernel.sysrq = 1
vm.overcommit_memory=2
vm.nr_hugepages=16384
```

Подготовить каталог и раздел для последующей установки сервера СУБД:

```
sudo mkdir /u01
fdisk /dev/hdd
```

В случае использования LVM, создаем необходимые компоненты:

```
sudo pvcreate /dev/hdd
sudo vgcreate u01 /dev/hdd
sudo lvcreate -l+100%FREE -n lv_u01_1 u01
sudo mkfs.ext4 /dev/u01/lv_u01_1
/etc/fstab:
/dev/u01/lv_u01_1          /u01      ext4      defaults,noatime
0 0
sudo mount /u01
```

Создать необходимые пользователей, директории:

```
sudo groupadd -g 36 postgres
sudo useradd -u 36 -g 36 postgres
sudo groupadd -g 1003 redis
sudo useradd -u 1003 -g 1003 redis
sudo groupadd -g 1004 tika
sudo useradd -u 1004 -g 1004 tika
sudo mkdir -p /u01/ pgprod /data
sudo chown postgres.postgres -R /u01/ pgprod
```

Изменить домашний каталог для пользователя postgres -  
/etc/passwd

```
postgres:x:26:26:PostgreSQL Server:/u01/pgprod:/bin/bash
```

Установить java последней версии:

Скачать официальный дистрибуит java.

Выполнить установку:

```
sudo tar -xzf jdk1.8.0_152-linux-x64.tar.gz -C /usr/local/java8/
alternatives --install /usr/bin/java java
/usr/java/jdk1.8.0_152/bin/java 1
sudo alternatives --config java
```

### 3.2. Установка и настройка сервера СУБД

Установка СУБД производится согласно инструкциям к выбранной СУБД.

Настроим репозиторий для установки СУБД postgresql и его компонентов:

```
rpm -Uvh http://yum.postgresql.org/9.6/redhat/rhel-7-x86_64/pgdg-redhat96-9.6-3.noarch.rpm
```

Установить основные компоненты СУБД:

```
postgresql-server postgresql-contrib pgadmin powa pgpool-2  
pg_stat_kcache96 pg_top96
```

Создать файл .bash\_profile от пользователя postgres, с указываем в нем переменные окружения:

```
[ -f /etc/profile ] && source /etc/profile  
PGDATA=/u01/pgtest/data  
export PGDATA  
# If you want to customize your settings,  
# Use the file below. This is not overridden  
# by the RPMS.  
[ -f /u01/pgtest/.pgsql_profile ] && source /u01/pgprod/.pgsql_profile  
export PATH=$PATH:/usr/pgsql-9.6/bin:/usr/pgpool-9.6
```

Инициализировать кластер БД от пользователя postgres

```
/usr/pgsql-9.6/bin/initdb -D /u01/pgprod/data
```

Запустить СУБД:

```
/usr/pgsql-9.6/bin/pg_ctl -D /u01/pgprod/data -l logfile start
```

Выполнить базовые настройки для возможности подключения к СУБД удаленно:

```
vim /u01/pgprod/data/postgresql.conf
```

```
listen_addresses = '*'
```

```
vim /u01/pgtest/data/pg_hba.conf
```

```
host    all             postgres    10.0.0.0/8          md5
```

Создать скрипт запуска останова СУБД для systemd:

```
sudo touch /etc/systemd/system/postgresql.service
```

```
[Unit]
```

```
Description=PostgreSQL database server
```

```
After=network.target
```

```
[Service]
```

```
Type=forking
```

```
User=postgres
```

```
Group=postgres
```

```
# Where to send early-startup messages from the server (before the logging
```

```
# options of postgresql.conf take effect)
```

```
# This is normally controlled by the global default set by systemd
```

```
# StandardOutput=syslog
```

```
# Disable OOM kill on the postmaster
```

```
OOMScoreAdjust=-1000
```

```
# ... but allow it still to be effective for child processes
```

```
# (note that these settings are ignored by Postgres releases before 9.5)
```

```
Environment=PG_OOM_ADJUST_FILE=/proc/self/oom_score_adj
```

```
Environment=PG_OOM_ADJUST_VALUE=0
```

```
# Maximum number of seconds pg_ctl will wait for postgres to start.
```

Note that

```
# PGSTARTTIMEOUT should be less than TimeoutSec value.
```

```
Environment=PGSTARTTIMEOUT=270
```

```
Environment=PGDATA=/u01/ pgprod /data
```

```
ExecStart=/usr/pgsql-9.6/bin/pg_ctl start -D ${PGDATA} -s -w -t  
${PGSTARTTIMEOUT}
```

```
ExecStop=/usr/pgsql-9.6/bin/pg_ctl stop -D ${PGDATA} -s -m fast
```

```
ExecReload=/usr/pgsql-9.6/bin/pg_ctl reload -D ${PGDATA} -s
```

```
# Give a reasonable amount of time for the server to start up/shut  
down.
```

```
# Ideally, the timeout for starting PostgreSQL server should be  
handled more
```

```
# nicely by pg_ctl in ExecStart, so keep its timeout smaller than this  
value.
```

```
TimeoutSec=300
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Инициализировать систему инициализации system:

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable postgresql.service
```

```
sudo systemctl start postgresql.service
```

Установить пароль на пользователя postgres в БД:

```
psql
```

```
\password postgres
```

Создать БД eehd:

```
CREATE DATABASE eeHD;
```

Создать роли и привилегии:

```
CREATE ROLE eeHD LOGIN PASSWORD 'password';  
grant all on database eeHD to eeHD;  
CREATE ROLE devel LOGIN PASSWORD "password";  
GRANT CONNECT ON DATABASE eeHD to eeHD;  
GRANT USAGE ON SCHEMA public TO eeHD;  
GRANT SELECT ON ALL TABLES IN SCHEMA public TO eeHD;  
GRANT DELETE ON ALL TABLES IN SCHEMA public TO eeHD;
```

Отдельные роли на таблицы:

```
psql eeHD  
grant select,insert,update,delete on all tables in schema public to  
eeHD;
```

Для новых таблиц, чтобы заново не переназначать роли:

```
psql eeHD  
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT,  
INSERT, UPDATE, DELETE ON TABLES TO eeHD;
```

### **3.3. Установка основных компонент ПО Documino.Архив**

#### **3.3.1 Установка сервера содержимого**

Скопировать jar файл в ранее созданный каталог для сервера содержимого:

```
/u01/jackrabbit/bin
```

Сформировать строку запуска сервера содержимого:

```
/usr/local/java7/bin/java -Xmx1g -jar  
/home/jackrabbit/bin/jackrabbit-standalone-2.14.2.jar --host IP_хоста  
-r /home/jackrabbit/jackrabbit >> /home/jackrabbit/jackrabbit.log  
2>&1
```

Сделать цель для системы инициализации systemd:

```
sudo touch /etc/systemd/system/jackrabbit.service
```

```
sudo systemctl enable jackrabbit.service
```

```
[Unit]
```

```
Description=jackrabbit
```

```
After=syslog.target
```

```
[Service]
```

```
User=jackrabbit
```

```
ExecStart=/bin/java -Xmx6g -Xms4G -
```

```
Djackrabbit.log=/u01/jackrabbit/logs/jackrabbit.log -
```

```
Djetty.log=/u01/jackrabbit/logs/jetty.log -jar
```

```
/u01/jackrabbit/bin/jackrabbit-standalone-2.16.0.jar --host
```

```
IP_севра --port 8080 -r /u01/jackrabbit/repository >>
```

```
/u01/jackrabbit/logs/jackrabbit.log 2>&1
```

```
SuccessExitStatus=143
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Запуск останов сервера:

```
sudo systemctl start jackrabbit.service
```

```
sudo systemctl stop jackrabbit.service
```

Журнал работы приложения:

```
/u01/jackrabbit/logs/jackrabbit.log
```

### 3.3.2 Установка сервера полнотекстового поиска

Скачать и установить дистрибутив Elasticsearch.

```
sudo rpm -ivh elasticsearch-5.6.4.rpm
```

Установить модуль морфологии:

```
sudo /usr/share/elasticsearch/bin/elasticsearch-plugin install
file:///u01/distr/elasticsearch-analysis-morphology-5.6.4.zip
```

Создать необходимые директории:

```
mkdir -p /u01/elastic/{data,logs}
```

```
chown -R elasticsearch.elasticsearch /u01/elastic
```

Выполнить настройку параметров сервиса:

```
/etc/elasticsearch/elasticsearch.yml
```



```
path.data: /u01/elastic/data
## Path to log files:
path.logs: /u01/elastic/logs
http.cors.enabled: true
http.cors.allow-origin: "*"
network.host: IP
bootstrap.system_call_filter: false
Настроить цель для системы инициализации system;
sudo systemctl daemon-reload
sudo systemctl enable elasticsearch.service
systemctl start elasticsearch
```

### 3.3.3 Установка сервера извлечения текста

Скачать и установить дистрибутив Tika.

Создать структуру каталогов и пользователя для Tika сервера:

```
mkdir -p /u01/tika/{bin,config,logs}
chown -R tika:tika /u01/tika
```

Сформировать строку запуска Tika сервера:

```
/usr/local/java8/bin/java -Djava.awt.headless=true -jar
/u01/tika/bin/tika-server-1.16.jar --port 9900 --host IP_host >
/u01/tika/bin/logs/stdout.log 2>&1 &
```

Создать цель для системы инициализации system:

Создать файл описания сервисов:

/etc/systemd/system/tika.service

содержимое файла:

[Unit]

Description=tika

After=syslog.target

[Service]

User=tika

```
ExecStart=/bin/java -jar /u01/app/tika/bin/tika-server-1.16.jar -  
-port 9900 --host 0.0.0.0 >> /u01/app/tika/logs/tika.log 2>&1  
SuccessExitStatus=143  
[Install]  
WantedBy=multi-user.target
```

### 3.3.4 Установка сервера кэширования

Скачать и установить дистрибутив Redis.

```
tar xzf redis-4.0.8.tar.gz  
$ cd redis-4.0.8  
$ make  
$make install
```

Сделать цель для системы инициализации system:

```
/etc/systemd/system/redis.service  
Description=Redis In-Memory Data Store  
After=network.target  
[Service]  
Type=forking  
User=redis  
Group=redis  
Environment=statedir=/run/redis  
PermissionsStartOnly=true  
PIDFile=/run/redis/redis.pid  
ExecStartPre=/bin/touch /var/log/redis.log  
ExecStartPre=/bin/chown redis:redis /var/log/redis.log  
ExecStartPre=/bin/mkdir -p ${statedir}  
ExecStartPre=/bin/chown -R redis:redis ${statedir}  
ExecStart=/usr/local/bin/redis-server /etc/redis.conf  
ExecStop=/usr/local/bin/redis-cli -p 10999 shutdown  
ExecReload=/bin/kill -USR2 $MAINPID  
Restart=always  
RestartSec=3  
[Install]
```

WantedBy=multi-user.target

Выполнить базовые настройки сервера Redis:

```
bind 0.0.0.0
protected-mode no
port 10999
tcp-backlog 511
timeout 0
tcp-keepalive 300
daemonize yes
supervised auto
pidfile /run/redis/redis.pid
loglevel notice
logfile ""
databases 16
always-show-logo yes
save 900 1
save 300 10
save 60 10000
stop-writes-on-bgsave-error yes
rdbcompression yes
rdbchecksum yes
dbfilename dump.rdb
dir /home/redis/
slave-serve-stale-data yes
slave-read-only yes
repl-diskless-sync no
repl-diskless-sync-delay 5
repl-disable-tcp-nodelay no
slave-priority 100
lazyfree-lazy-eviction no
lazyfree-lazy-expire no
lazyfree-lazy-server-del no
```

slave-lazy-flush no  
appendonly no  
appendfilename "appendonly.aof"  
appendfsync everysec  
no-appendfsync-on-rewrite no  
auto-aof-rewrite-percentage 100  
auto-aof-rewrite-min-size 64mb  
aof-load-truncated yes  
lua-time-limit 5000  
slowlog-log-slower-than 10000  
slowlog-max-len 128  
latency-monitor-threshold 0  
notify-keyspace-events ""  
hash-max-ziplist-entries 512  
hash-max-ziplist-value 64  
list-max-ziplist-size -2  
list-compress-depth 0  
set-max-intset-entries 512  
zset-max-ziplist-entries 128  
zset-max-ziplist-value 64  
hll-sparse-max-bytes 3000  
activerehashing yes  
client-output-buffer-limit normal 0 0 0  
client-output-buffer-limit slave 256mb 64mb 60  
client-output-buffer-limit pubsub 32mb 8mb 60  
hz 10  
aof-rewrite-incremental-fsync yes

### **3.4 Установка экземпляра ПО Documino.Архив.**

#### **3.4.1 Состав архива установки.**

На рисунке 1. показан состав архива установки ПО.

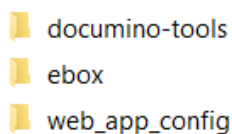


Рисунок 1.

Каталог documino-tools – содержит утилиты для установки и сборки экземпляра программного обеспечения.

Содержимое каталога:

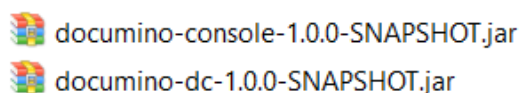


Рисунок 2.

Каталог ebox – содержит основные конфигурационные файлы для обновления/установки экземпляра программного обеспечения.

Так же в нем расположен пакет скриптов xqI для обновления базовых типов и основные плагины и формы для обновления/установки экземпляра программного обеспечения.

Содержимое каталога ebox:

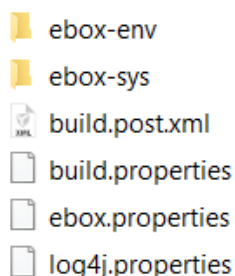


Рисунок 3.

Конфигурационные файлы установки из каталога ebox:

- build.properties – конфигурационный файл содержит параметры/реквизиты доступа к основным подсистемам documino.
- ebox.properties - конфигурационный файл содержит параметры/реквизиты доступа к серверу содержания и серверу СУБД.

- log4j.properties - конфигурационный файл содержит параметры/реквизиты журналирования процесса установки/обновления экземпляра программного обеспечения.
- build.post.xml - описание установки/обновления экземпляра программного обеспечения.

Каталог ebox\ebox-env\install - пакет скриптов xql для обновления базовых типов.

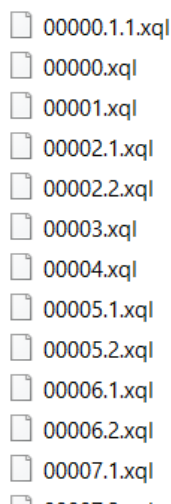


Рисунок 4.

Каталог ebox\ebox-sys - основные плагины и формы для обновления/установки экземпляра программного обеспечения.

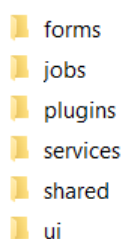


Рисунок 5.

Каталог web\_app\_config - содержит следующие файлы:

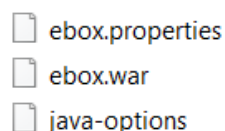


Рисунок 6.

- ebox.properties - конфигурационный файл работы веб приложения на сервере приложения, например на сервере приложения Apache-Tomcat

- `ebox.war` – основной бинарный файл экземпляра программного обеспечения `documino`.
- `java-options` – параметры `java` для запуска экземпляра программного обеспечения для веб-сервера.

### 3.4.2 Установка архива экземпляра программного обеспечения.

В начале необходимо создать структуру каталогов, если еще не созданы, пример:

```
/u01/work.
```

В данный каталог распаковать архив экземпляра программного обеспечения, должно получиться:

```
/u01/work/documino-tools
```

```
/u01/work/ebox
```

```
/u01/work/web_app_config
```

Необходимо выдать соответствующие права для пользователя на каталоги и файлы, от которого будет запускаться процесс установки.

### 3.4.3 Подготовка конфигурационных файлов установки.

Основные конфигурационные файлы установки после пункта 2, будут расположены в каталоге `/u01/work/ebox`.

Состав и описание файлов параметров установки экземпляра программного обеспечения:

#### **build.properties:**

`application.url=http://xxx.xxx.xxx.xxx/`    **#необходимо указать IP адрес сервера приложения.**

**#Redis – указываем IP адрес и порт, где расположен сервер кэширования.**

```
redis.host=xxx.xxx.xxx.xxx
```

```
redis.port=10999
```

**#ldap – указывай адрес и реквизиты LDAP.**

```
ldap.url=ldap://ldap.rosseti.ru:389
```

```
ldap.base.dn=dc=RU  
ldap.connection.dn=Anonymous  
ldap.connection.password=
```

```
#Elastic Search – указываем IP адрес и порт, где расположен сервер  
полнотекстового поиска  
es.host= xxx.xxx.xxx.xxx  
es.port=9200
```

```
#Apache Tika - указываем IP адрес и порт, где расположен сервер  
извлечения текста.  
tika.host= xxx.xxx.xxx.xxx  
tika.port=9900
```

```
#SMTP Server – указываем адрес и порт сервера SMTP.  
smtp.host=localhost  
smtp.port=25
```

### **ebox.properties**

**#указываем реквизиты доступа к серверу СУБД.**

```
db=postgres  
db.host= xxx.xxx.xxx.xxx  
db.port=5432  
db.name=uhk  
db.user=uhk  
db.password=passwd
```

**#указываем IP адрес сервера содержания**

```
content.url=jcr://http:// xxx.xxx.xxx.xxx:8080/server
```

```
documino.trace=true  
cache.feature.lifetime=600  
cache.type.lifetime=600
```



**log4j.properties**

log4j.rootLogger=DEBUG, CONSOLE, FILE

#----- CONSOLE -----

log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender

log4j.appender.CONSOLE.threshold=DEBUG

log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout

log4j.appender.CONSOLE.layout.ConversionPattern=[%t] %d{dd.MM.yyyy  
HH:mm:ss,SSS} %p %c{1} - %m%n

#----- FILE -----

log4j.appender.FILE=org.apache.log4j.RollingFileAppender

log4j.appender.FILE.File=/u01/work/logs/trace.log # указываем  
расположение файла журнала, необходимо создать каталог  
/u01/work/logs

log4j.appender.FILE.MaxFileSize=50MB

log4j.appender.FILE.layout=org.apache.log4j.PatternLayout

log4j.appender.FILE.layout.ConversionPattern=[%t] %d{dd.MM.yyyy  
HH:mm:ss,SSS} %p %c{1} - %m%n

**build.post.xml**

Оставляем без изменений.

**3.4.4 Подготовка конфигурации сервера приложения.**

В данном примере в качестве сервера приложения используется сервер Apache-Tomcat.

Каждый экземпляр сервера приложения Apache-Tomcat будет расположен по пути:

/u01/app/tomcat8/instance1-n, где n = 0,1,2 ... и т.д.

Необходимо бинарный файл экземпляра программного обеспечения /u01/work/web\_app\_config/ebox.war поместить по следующему пути сервера приложения: /u01/app/tomcat8/instance1-0/webapps

/u01/work/web\_app\_config/ebox.properties – конфигурационный файл работы веб приложения, необходимо скопировать по следующему пути сервера приложения:

/u01/app/tomcat8/instance1-0/conf

Состав и описание файла параметров запуска экземпляра программного обеспечения сервера приложения - ebox.properties:

В файле указываем реквизиты подключения к серверу СУБД и серверу содержания:

db=postgres

db.host=xxx.xxx.xxx.xxx # необходимо указать IP адреса сервера СУБД.

db.port=5432

db.name=uhk

db.user=uhk

db.password=password

#uidgen.url=mrmi4:// xxx.xxx.xxx.xxx:6989

#необходимо указать IP адрес сервера содержания.

content.url=jcr://http://xxx.xxx.xxx.xxx:8080/server

В сценарий запуска сервера приложения /u01/app/tomcat8/instance1-0/bin/catalina.sh добавить параметры запуска экземпляра программного обеспечения:

export JAVA\_OPTS="-Debox.properties=/u01/app/tomcat8/instance1-0/conf/ebox.properties -

Dlog4j.configuration=file:///u01/app/tomcat8/instance1-0/conf/log4j.properties -Dlog4j.debug -

Dsun.zip.disableMemoryMapping=true -Xmx3G -Xms3g -

XX:+AlwaysPreTouch -XX:+UseConcMarkSweepGC -

XX:+CMSParallelRemarkEnabled -XX:+ParallelRefProcEnabled -

```
XX:+CMSClassUnloadingEnabled -XX:CMSInitiatingOccupancyFraction=80 -
XX:+UseCMSInitiatingOccupancyOnly -XX:+UseParNewGC -
XX:NewSize=512m -XX:MaxNewSize=512m -
Djava.library.path=$LD_LIBRARY_PATH -XX:hashCode=5 -
Dapplication.preferencesrepository.password=dmc_wdk_preferences_owner
-Djava.library.path=/usr/local/apr/lib"
JPDA_ADDRESS=0.0.0.0:8000
CATALINA_PID=/u01/app/tomcat8/instance1-0/tomcat.pid
```

### **3.4.5 Установка/обновление экземпляра программного обеспечения.**

Установку/обновление экземпляра программного обеспечения можно выполнить, используя следующий сценарий:

```
java -Dbuild.properties.file=/u01/work/ebox/build.properties -
Dlog4j.configuration=file:///u01/work/ebox/log4j.properties -
Ddocumino.properties=/u01/work/ebox/ebox.properties -jar
/u01/work/documino-tools/documino-dc-1.0.0-SNAPSHOT.jar -f
/u01/work/ebox/build.post.xml
```

Все пути к конфигурационным и бинарным файлам указаны исходя из структуры каталогов, созданной в данной инструкции по установке - /u01/work.