



Акционерное общество «Флекс Софтваре Системс»  
127055 Россия, г. Москва, ул. Новолесная 2, офис 3  
Тел.: +7 (495) 788 - 03 - 25  
e-mail: [info@flexsoft.com](mailto:info@flexsoft.com)  
[www.flexsoft.com](http://www.flexsoft.com)

---

# **Схема развертывания программного обеспечения FXL 3.0**

## Содержание

1. КОМПОНЕНТЫ СИСТЕМНОГО ПО, НЕОБХОДИМЫЕ ДЛЯ РАБОТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ FXL 3.0 .....	3
2. УСТАНОВКА И НАСТРОЙКА КОМПОНЕНТОВ СИСТЕМНОГО ПО .....	4
2.1 Установка Операционной системы .....	4
2.1 Установка JAVA virtual machine.....	5
2.2 Установка и конфигурирование Yandex Database .....	6
2.3 Установка и конфигурирование DataGrid.....	7
2.4 Установка и конфигурирование среды исполнения FXL 3.0.....	9
3. Конфигурирование и запуск Rest-API программного обеспечения FXL 3.0.....	10
4. Служебные скрипты .....	11
3. ПОРЯДОК ЗАПУСКА И ОТСЛЕЖИВАНИЯ УСПЕШНОСТИ ЗАПУСКА СЕРВИСОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ FXL 3.0 .....	14
3.1. Запуск YDB.....	14
3.2. Запуск DataGrid .....	15
3.3. Приложение и REST-сервис программного обеспечения FXL 3.0 .....	16
3.4. Отслеживания успешности запуска сервисов программного обеспечения FXL 3.0.....	17
4. ОПИСАНИЕ СЕРВИСОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ FXL 3.0 В SYSTEMCTL .....	18
5. УПРАВЛЕНИЕ СЕРВИСАМИ ПРОИЗВОДИТСЯ С ПОМОЩЬЮ УТИЛИТЫ SYSTEMCTL. КРАТКАЯ СПРАВКА ПО УТИЛИТЕ SYSTEMCTL .....	21
6. ЛОГИРОВАНИЕ СЕРВИСОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ FXL 3.0.....	23

## 1. Компоненты системного ПО, необходимые для работы программного обеспечения FXL 3.0

AstraLinux 1.8 редакция «Орел». Сайт производителя [Российская операционная система](#). Запись в реестре отечественного ПО [Операционная система специального назначения «Astra Linux Special Edition»](#)

1.2. JAVA : Axiom JDK pro-11.0.25-full. Сайт производителя [Среда разработки и исполнения Axiom JDK](#). Запись в реестре отечественного ПО - [Среда разработки и исполнения Java "Axiom JDK Pro"](#)

1.3. Среда исполнения приложения FXL 3.0 [Platform V UI Kit - решение для построения интерфейсов в собственном корпоративном стиле | СберТех](#). Запись в реестре отечественного ПО [Platform V UI Kits](#)

1.4. DataGrid: версия SE-17.6.0-beta1. Сайт производителя [Apache Ignite - вычисления в оперативной памяти Platform V DataGrid | СберТех](#). Запись в реестре отечественного ПО [Platform V DataGrid](#)

1.5. PangolinDB - Сайт производителя [Реляционная СУБД для крупного бизнеса от СберТеха](#). Запись в реестре отечественного ПО [Platform V Pangolin DB](#). [Yandex Database \(YDB\) — Локальная база данных Яндекса — On-Premise решение для бизнеса](#)

1.6. Служебные скрипты

## 2. Установка и настройка компонентов системного ПО

### 2.1 Установка Операционной системы

Операционная система AstraLinux 1.8 редакция «Орел». Установка выполняется согласно документации вендора [Серверная Астра | Astra Linux Server](#)

## 2.1 Установка JAVA virtual machine

Установлена в /opt/java/jdk-pro-11.0.25-full согласно документации [Руководство по установке Axiom JDK 11.0.5+11](#)

## 2.2 Установка и конфигурирование Yandex Database

Установка Yandex Database (далее – YDB) выполняется согласно документации вендора [Корпоративная СУБД Яндекса | YDB](#)

YDB должна быть настроена в одноузловом режиме с хранением БД на «сыром» разделе (/dev/disk/by-partlabel/ydb\_FXL\_disk\_01)

Конфигурационный файл FXL3-YDB - /opt/ydb/yasubd-24.3.13.12-linux-amd64/config/config\_drive.yaml

Работа с YDB осуществляется по портам:

2136 – GRPC (JDBC)

8765 – веб-консоль мониторинга и выполнения запросов YQL (SQL)

База YDB - /Root/FXL

Первичная БД FXL 3.0 на импортонезависимом технологическом стеке загружается из дампа, предоставленного Флекссофт, загрузка дампа согласно документации [Загрузка из файловой системы | YDB](#). Дамп находится в каталоге /opt/ydb/dumps/FXLTEST2STAGE.ydb.dmp

Запуск и остановка осуществляется через сервис systemctl - ydb-storage.service и ydb-db-FXL.service

Проверка существования топиков осуществляется скриптом check\_topics.sh.

В случае успешной проверки выводится сообщение:

```
Check topic for INT_TR_ENTRIES
Topic for INT_TR_ENTRIES exists
Check topic for MGC_SM_DOCUMENTS
Topic for MGC_SM_DOCUMENTS exists
Check topic for MGC_TR_ENTRIES
Topic for MGC_TR_ENTRIES exists
Check topic for SCS_DM_DOCUMENTS
Topic for SCS_DM_DOCUMENTS exists
4 topics are exists
```

В случае неуспешной проверки, выводится сообщение о недостающих топиках – Not found topic for <имя таблицы> и итоговое сообщение – Not found <количество недостающих топиков> topics. Например:

```
Check topic for INT_TR_ENTRIES
Not found topic for INT_TR_ENTRIES
Check topic for MGC_SM_DOCUMENTS
Topic for MGC_SM_DOCUMENTS exists
Check topic for MGC_TR_ENTRIES
Topic for MGC_TR_ENTRIES exists
Check topic for SCS_DM_DOCUMENTS
Topic for SCS_DM_DOCUMENTS exists
Not found 1 topics
```

**Для проверки топиков обязательно должны быть запущены сервисы ydb-storage.service и ydb-db-FXL.service!!!**

## 2.3 Установка и конфигурирование DataGrid

DataGrid должен быть установлен в /opt/ignite/Ignite-SE-17.6.0-beta1 и настроен в одноузловом in-memory режиме. Персист выполняется в Native storage (п.2.2)

Конфигурационный файл - /opt/ignite/Ignite-SE-17.6.0-beta1/config/node\_config.xml

Настройка соединения с YDB (п.2.2) в случае выбора YDB как persit-store

```
<property name="userAttributes">
  <map>
    <entry key="DATANODE" value="true"/>
    <entry key="CONFIG_NODE" value="true"/>
    <entry key="sync.service.node" value="true"/>
    <entry key="NAME" value="server-node-test1"/>
    <entry key="AUTO_ACTIVATE_NODES_COUNT" value="1"/>
    <entry key="CACHE_STORE_TYPE" value="EXTERNAL"/>
    <entry key="CACHE_STORE_VENDOR" value="YDB"/>
    <entry key="CACHE_STORE_MODE" value="ALL"/>
    <entry key="CACHE_STORE_URL" value="<YDB hostname>:2136/Root/FXL"/>
  </map>
</property>
```

Где:

YDB hostname – сервер YDB

NAME – имя ноды DataGrid (в текущей конфигурации можно изменить)

Настройка порта обслуживания JDBC

```
<property name="clientConnectorConfiguration">
  <bean class="org.apache.ignite.configuration.ClientConnectorConfiguration">
    <property name="thinClientEnabled" value="true"/>
    <property name="port" value="10800"/>
    <property name="thinClientConfiguration">
      <bean class="org.apache.ignite.configuration.ThinClientConfiguration">
        <property name="maxActiveComputeTasksPerConnection" value="1"/>
      </bean>
    </property>
  </bean>
</property>
```

Работа с DataGrid осуществляется по порту 10800 – JDBC

Включение Calcite SQL engine

```
<property name="sqlConfiguration">
  <bean class="org.apache.ignite.configuration.SqlConfiguration">
    <property name="sqlQueryHistorySize" value="100000"/>
    <property name="queryEnginesConfiguration">
      <list>
        <bean
class="org.apache.ignite.indexing.IndexingQueryEngineConfiguration">
          <property name="default" value="false"/>
        </bean>
        <bean
class="org.apache.ignite.calcite.CalciteQueryEngineConfiguration">
          <property name="default" value="true"/>
        </bean>
      </list>
    </property>
  </bean>
</property>
```

## Конфигурирование кластера Datagrid

### [Clustering | Ignite Documentation](#)

Простая конфигурация статического (количество нод зафиксировано в конфигурации) кластера:

```
<property name="ipFinder">
  <bean
class="org.apache.ignite.spi.discovery.tcp.ipfinder.vm.TcpDiscoveryVmIpFinder">
    <property name="addresses">
      <list>
        <value>node-1.mydomain:44510..44513</value>
        <value>node-2.mydomain:44510..44513</value>
        <value>node-3.mydomain:44510..44513</value>
      </list>
    </property>
  </bean>
</property>
```

Для каждой ноды необходимо указать. Пример для node-1.mydomain:

```
<bean class="org.apache.ignite.spi.discovery.tcp.TcpDiscoverySpi">
  <property name="localAddress" value="node-1.mydomain"/>
  <property name="localPort" value="44510"/>
</bean>
```

Дополнительные параметры DataGrid-JVM в файле /opt/ignite/ignite-SE-17.6.0-beta1/config/jvm.opts.

Изменяемые параметры DataGrid-JVM:

- Xmx -Xms – параметры размера хипа JVM (настраиваются в зависимости от профиля нагрузки)
- Djava.io.tmpdir – каталог хранения временных файлов
- Dcom.sun.management.jmxremote.port -Dcom.sun.management.jmxremote.rmi.port – порты мониторинга DataGrid-JVM (подробнее в [Monitoring and Management Using JMX Technology - Java SE Monitoring and Management Guide](#))

Запуск и остановка осуществляется через сервис systemctl - ignite.service

## 2.4 Установка и конфигурирование среды исполнения программного обеспечения FXL 3.0

Установка выполняется согласно документации вендора [Руководство по установке - Workflow](#).

Архив для сборки докером - FXL30.tgz

Настройка URL Datagrid в скрипте старта приложения - <путь распакованного FXL30.tgz>/owned/flex-D-01.001.00-7095-distrib/package/docker/sbt-lite-demo/docker-entrypoint.sh

```
#!/bin/sh
```

```
java -Dsbt_lite_demo.ignite.addresses=<datagrid URL>:<port> -  
Dsbt_lite_demo.autoProcessing=true -Dsbt_lite_demo.ignite.schema=TBG \
```

```
    -Dsbt_lite_demo.flexBaseUrl=http://localhost:8080/sbt-lite-  
demo/rest/v1/flexsoft/parcels \
```

```
    -Dsbt_lite_demo.ignite.user="tbg" -Dsbt_lite_demo.ignite.password="tbg" -jar /sbt-  
lite/sbt-lite-demo.jar
```

Сборка приложения в docker-имидж выполняется командой

```
docker build -f <путь распакованного FXL30.tgz>/owned/flex-D-01.001.00-7095-  
distrib/package/docker/sbt-lite-demo/Dockerfile ./
```

Далее, запускаем собранный docker-имдж.

### 3. Конфигурирование и запуск Rest-API программного обеспечения FXL 3.0

Файл сервиса /opt/rest-api/fxl-rest-api-1.0.0.jar

Работа с DataGrid осуществляется по порту 8091.

Настройка выполняется в файле /etc/systemd/system/FXL-rest-api.service

```
Environment="JAVA_HOME=/opt/java/jdk-pro-11.0.25-full"  
Environment="PORT=8091"  
Environment="URL_IGNITE=localhost:10800/TBG"  
Environment="SESSION_USER=USR2"  
ExecStart=/opt/java/jdk-pro-11.0.25-full/bin/java -jar ./fxl-rest-api-1.0.0.jar -  
DIGNITE_ALLOW_DML_INSIDE_TRANSACTION=true
```

Где:

PORT – порт обслуживание запросов

URL\_IGNITE – JDBC URL DataGrid

Запуск и остановка осуществляется через сервис systemctl - FXL-rest-api.service

Лог работы сервиса Rest-API - /var/log/flexsoft/fxl-rest-api.log

## 4. Служебные скрипты

Наименование скрипта	Назначение скрипта
/opt/ignite/Ignite-SE-17.6.0-beta1/check_fxl-ignite_ready.sh	Проверка завершения запуска «DataGrid». Используется в ExecStartPre сервисов FXL-front-app.service FXL-rest-api.service FXL-sync-oracle2ydb.service FXL-sync-ydb2oracle.service
/opt/ignite/Ignite-SE-17.6.0-beta1/run_ignite_script_cli.sh	Выполнение sql-скриптов в «DataGrid»
/opt/ydb/yasubd-24.3.13.12/drop_create_topics.sh	Пересоздание топиков YDB для синхронизации и отправки сообщений в Мультикарту
/opt/ydb/yasubd-24.3.13.12/check_topics.sh	Проверка существования необходимых топиков YDB

Скрипты с правами на исполнение от пользователя root

### Скрипт check\_fxl-ignite\_ready.sh

```
#!/bin/bash

export JAVA_HOME=/opt/java/jdk-pro-11.0.22
export IGNITE_HOME=/opt/ignite/Ignite-SE-17.6.0-beta1
export IGNITE_URL=<ignite hostname>:21801
export PATH=$JAVA_HOME/bin:$PATH

cd $IGNITE_HOME

SQLMARKER=FXL_IGNITE_READY
SQL2CHECK="SELECT concat('$SQLMARKER ',value) as FXLIGNITEREADY FROM sys.DS_ATOMICLONGS t
WHERE t.name = 'CLUSTER_ENV_READY_FLAG_NAME'"
[ ! -z "$1" ] && [ "$1" = "create_indexes" ] && do_create_indexes=yes ||
do_create_indexes=no
steps=40
step=0
delayonattp=10

while [ $step -le $steps ]; do
    is_ignite_ready=`$IGNITE_HOME/bin/sqlline.sh --outputformat=csv --verbose=true -u
"jdbc:ignite:thin://$IGNITE_URL" -n tbg -p tbg --fastConnect=true --silent=true \
-e "$SQL2CHECK" 2> /dev/null | grep "$SQLMARKER" | sed -e "s/'//g" | cut -f 2 -d " "`

    if [ "0$is_ignite_ready" = "01" ]; then
        echo `date +%Y-%m-%d %H:%M:%S` " Ignite ready"
        sleep $delayonattp
        break
    else
        echo `date +%Y-%m-%d %H:%M:%S` " Ignite not ready"
    fi
    step=`expr $step + 1`
    sleep $delayonattp
done
if [ "0$is_ignite_ready" = "01" ]; then
    echo `date +%Y-%m-%d %H:%M:%S` " Ignite successfully started"
    if [ $do_create_indexes = "yes" ]; then
        echo `date +%Y-%m-%d %H:%M:%S` " Delay ${delayonattp}sec"
        echo `date +%Y-%m-%d %H:%M:%S` " Run create indexes script"
        ./run_ignite_script_cli.sh indexes.sql
        echo `date +%Y-%m-%d %H:%M:%S` " Creation indexes script finished"
    fi
    exit 0
else
    echo `date +%Y-%m-%d %H:%M:%S` " Ignite was not started until timeout"
    exit 1
fi
```

### Скрипт run\_ignite\_script\_cli.sh. SQL-скрипт передается первым параметром

```
#!/bin/bash

export JAVA_HOME=/opt/java/jdk-pro-11.0.22

/opt/ignite/Ignite-SE-17.6.0-beta1/bin/sqlline.sh --verbose=true -u
"jdbc:ignite:thin://127.0.0.1:21801" -n tbg -p tbg --fastConnect=true --silent=true --
run=$1 2> /dev/null
```

### Скрипт drop\_create\_topics.sh

```
#!/bin/bash

ydb_grpc_url=localhost:2136
ydbhome=/opt/ydb/yasubd-24.3.13.12
yqlcmd="$ydbhome/bin/ydb -e -vvv --endpoint grpc://$ydb_grpc_url --database /Root/FXL --
user root --no-password "
tables="MGC_SM_DOCUMENTS MGC_TR_ENTRIES SCS_DM_DOCUMENTS"
change_feed_name=CDC_CHANGEFEED0
hours2keep=168

for t in $tables; do
    echo "Drop consumer CDC_${t}_CONSUMER"
    $yqlcmd topic consumer drop --consumer CDC_${t}_CONSUMER TBG/${t}/$change_feed_name
    echo "Drop CHANGEFEED for `TBG/${t}`"
    $yqlcmd yql -s "ALTER TABLE `TBG/${t}` DROP CHANGEFEED $change_feed_name"
    echo "Create CHANGEFEED for `TBG/${t}`"
    $yqlcmd yql -s "ALTER TABLE `TBG/${t}` ADD CHANGEFEED $change_feed_name WITH ( FORMAT =
'JSON', MODE = 'NEW_AND_OLD_IMAGES', VIRTUAL_TIMESTAMPS = TRUE, RETENTION_PERIOD =
Interval('PT${hours2keep}H') )"
    echo "Add consumer CDC_${t}_CONSUMER"
    $yqlcmd topic consumer add --consumer CDC_${t}_CONSUMER TBG/${t}/$change_feed_name
done

tables="INT_TR_ENTRIES"
change_feed_name=CDC_T_INT_TR_ENTRIES
for t in $tables; do
    echo "Drop consumer CDC_${t}_CONSUMER"
    $yqlcmd topic consumer drop --consumer CDC_T_${t}_CONSUMER TBG/${t}/$change_feed_name
    echo "Drop CHANGEFEED for `TBG/${t}`"
    $yqlcmd yql -s "ALTER TABLE `TBG/${t}` DROP CHANGEFEED $change_feed_name"
    echo "Create CHANGEFEED for `TBG/${t}`"
    $yqlcmd yql -s "ALTER TABLE `TBG/${t}` ADD CHANGEFEED $change_feed_name WITH ( FORMAT =
'JSON', MODE = 'NEW_AND_OLD_IMAGES', VIRTUAL_TIMESTAMPS = TRUE, RETENTION_PERIOD =
Interval('PT${hours2keep}H') )"
    echo "Add consumer CDC_${t}_CONSUMER"
    $yqlcmd topic consumer add --consumer CDC_T_${t}_CONSUMER TBG/${t}/$change_feed_name
done
```

### Скрипт check\_topics.sh

```
#!/bin/bash

ydbhome=/opt/ydb/yasubd-24.3.13.12
export LD_LIBRARY_PATH=$ydbhome/lib
grpc_url=localhost:2136
ydb_database=/Root/FXL
allerrors=0

checktopic() {
    tbname=$1
    cdcname=$2
    consumer=$3
```

```
    echo "Check topic for $tbname"
err=`$ydbhome/bin/ydb -vvv --endpoint grpc://$grpc_url --database $ydb_database --user root
--no-password \
    topic read $ydb_database/TBG/$tbname/$cdcname --consumer=$consumer --format=newline-
delimited --commit=false --limit 1 2>&1 | grep "The last error was: no path " | grep
"Closing session to cluster" | wc -l`
if [ "$err" -ge 1 ]; then
    echo "Not found topic for $tbname"
    allerrors=$((allerrors+1))
else
    echo "Topic for $tbname exists"
fi
}

checktopic INT_TR_ENTRIES CDC_T_INT_TR_ENTRIES CDC_T_INT_TR_ENTRIES_CONSUMER
checktopic MGC_SM_DOCUMENTS CDC_CHANGEFEED0 CDC_T_MGC_SM_DOCUMENTS_CONSUMER
checktopic MGC_TR_ENTRIES CDC_CHANGEFEED0 CDC_T_MGC_TR_ENTRIES_CONSUMER
checktopic SCS_DM_DOCUMENTS CDC_CHANGEFEED0 CDC_T_SCS_DM_DOCUMENTS_CONSUMER

if [ "$allerrors" -gt 0 ]; then
    echo "Not found $allerrors topics"
    exit -1
else
    echo "4 topics are exists"
    exit 0
fi
```

## **3. Порядок запуска и отслеживания успешности запуска сервисов программного обеспечения FXL 3.0**

### **3.1. Запуск YDB**

YDB storage сервер (сервис ydb-storage.service)

YDB database (сервис ydb-db-FXL.service. Запускать через 5-7 сек после запуска ydb-storage.service)

## 3.2. Запуск DataGrid

(запускать через 5-7 сек после сервисов ydb-storage.service и ydb-db-FXL.service)

### 3.3. Приложение и REST-сервис программного обеспечения FXL 3.0

Выполнять запуск ТОЛЬКО после полного запуска «DataGrid» (п.3.2). Критерий полного запуска «DataGrid» это результат запроса из «DataGrid» – *SELECT value FROM sys.DS\_ATOMICLONGS t WHERE t.name = 'CLUSTER\_ENV\_READY\_FLAG\_NAME'*. Результат должен быть равен «1». Полный запуск «DataGrid» занимает около 3 минут. Контроль завершения запуска «DataGrid» осуществляется скриптом `/opt/ignite/ignite-SE-17.6.0-beta1/check_fxl-ignite_ready.sh` в ExecStartPre в настройках зависимых сервисов

### 3.4. Отслеживания успешности запуска сервисов программного обеспечения FXL 3.0

YDB – в логах нет критичных ошибок работы сервиса

DataGrid – Запрос п.3.3 возвращает «1» и в логах нет критичных ошибок работы сервиса

Приложение – в логах нет критичных ошибок работы сервиса и наличие в логах информации о запуске:

```
... INFO [main] org.apache.catalina.startup.HostConfig.deployWAR Deployment of web application archive [/opt/axiom-libercat9-9.0.91/webapps/ROOT.war] has finished in [29 739] ms
```

```
... INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8090"]
```

```
... INFO [main] org.apache.catalina.startup.Catalina.start Server startup in [29827] milliseconds
```

REST-сервис – Обращение через веб-браузер к *<имя сервера или IP-адрес Rest-API>*:8091/actuator/health/. В ответе от Rest-API должно быть:

```
{"status": "UP" }
```

## 4. Описание сервисов программного обеспечения FXL 3.0 в systemctl

### ydb-storage.service (п.3.1.a)

```
[Unit]
Description=YDB storage service

[Service]
#ExecStartPre=/usr/bin/chown ydb /dev/disk/by-partlabel/ydb_FXL_disk_01
LimitNOFILE=16384
User=ydb
WorkingDirectory=/opt/ydb/yasubd-24.3.13.12-linux-amd64
ExecStart=/opt/ydb/yasubd-24.3.13.12-linux-amd64/bin/ydbd server --log-level 3 --
syslog --tcp --yaml-config /opt/ydb/yasubd-24.3.13.12-linux-
amd64/config/config_drive.yaml --grpc-port 2136 --ic-port 19001 --mon-port 8765 --
node static
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=YDB-storage
Restart=always
RestartSec=3

[Install]
WantedBy=multi-user.target
```

### ydb-db-FXL.service (п.3.1.b)

```
[Unit]
Description=YDB FXL database

[Service]
ExecStartPre=/bin/sleep 30
LimitNOFILE=16384
User=ydb
WorkingDirectory=/opt/ydb/yasubd-24.3.13.12-linux-amd64
ExecStart=/opt/ydb/yasubd-24.3.13.12-linux-amd64/bin/ydbd server --yaml-config
/opt/ydb/yasubd-24.3.13.12-linux-amd64/config/config_drive.yaml --tenant /Root/FXL
--node-broker vbnkinmdbtst01.bankexp.local:2136 --grpc-port 31011 --ic-port 31013
--mon-port 31012 --log-file-name /opt/ydb/yasubd-24.3.13.12-linux-
amd64/logs/FXL_reg.log
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=YDB-FXL-DB
Restart=always
RestartSec=3

[Install]
WantedBy=ydb-storage.service
```

### ignite.service (п.3.2)

```
[Unit]
Description=FXL Ignite Service
After=ydb-db-FXL.service
```

```
[Service]
LimitNOFILE=16384
WorkingDirectory=/opt/ignite/Ignite-SE-17.6.0-beta1
User=root
PrivateDevices=yes
ProtectSystem=full
Type=simple
ExecReload=/bin/kill -HUP $MAINPID
KillMode=mixed
KillSignal=SIGTERM
TimeoutStopSec=10
ExecStart=/opt/ignite/Ignite-SE-17.6.0-beta1/start_ignite_service.sh
SyslogIdentifier=FXL-Ignite
Restart=on-failure
RestartSec=5
```

```
[Install]
Alias=ignite.service
```

### FXL-front-app.service (n.3.3)

```
[Unit]
Description=FXL front application Service
After=ignite.service

[Service]
ExecStartPre=/opt/ignite/Ignite-SE-17.6.0-beta1/check_fxl-ignite_ready.sh
Type=forking
Environment="JAVA_HOME=/opt/java/jdk-pro-11.0.25-full"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom -
Djbo.tmpdir=/opt/axiom-libercat9-9.0.91/temp/ -Djbo.doconnectionpooling=false -
Djbo.ampool.initpoolsize=0 -Djbo.ampool.maxpoolsize=100 -
Djbo.ampool.maxavailablesize=100 -Djbo.recyclethreshold=100 -
Djbo.ampool.minavailablesize=0 -Djbo.ampool.timetolive=-1 -
Djbo.ampool.maxinactiveage=2400000 -Djbo.ampool.monitorsleepinterval=1200000 -
Djbo.max.cursors=100 -Djbo.locking.mode=pessimistic -Djbo.sql92.LockTrailer= -
Djbo.maxpassivationstacksize=100 -Djbo.txn.disconnect_level=1 -
Dcom.flexsoft.fxl.oracleFormsRunMode=JWS -
Dcom.flexsoft.fxl.maxQueryElapsedTime=3000 -
Dcom.flexsoft.fxl.useSessionKeepAlive=true -
Dcom.flexsoft.fxl.sessionKeepAliveTimeout=10900000 -
Djbo.def.mgr.listener=com.flexsoft.fxl.masterapp.model.CustomDefMgrListener -
Dfxl.dualCoreMode=true -
Doracle.as.jmx.framework.spi=oracle.as.jmx.framework.standardmbeans -
Djbo.SQLBuilder=com.flexsoft.fxl.commons.adf.modelutils.baseclasses.FxlOracleSQLBu
ilderImpl -Doracle.mds.cache=simple -Dorg.apache.el.parser.SKIP_IDENTIFIER=true -
Djbo.passivationstore=file -Djbo.tmpdir=/opt/axiom-libercat9-9.0.91/temp/ -
Dorg.apache.el.parser.SKIP_IDENTIFIER_CHECK=true "
Environment="CATALINA_PID=/var/run/libercat/%i.pid"
Environment="CATALINA_BASE=/opt/axiom-libercat9-9.0.91/"
Environment="CATALINA_HOME=/opt/axiom-libercat9-9.0.91/"
ExecStart=/opt/axiom-libercat9-9.0.91/bin/startup.sh
ExecStop=/opt/axiom-libercat9-9.0.91/bin/shutdown.sh
WorkingDirectory=/opt/axiom-libercat9-9.0.91
User=root
PrivateDevices=yes
ExecReload=/bin/kill -HUP $MAINPID
KillMode=mixed
KillSignal=SIGTERM
TimeoutStopSec=10
```

```
SyslogIdentifier=FXL-front-app  
Restart=on-failure  
RestartSec=5
```

```
[Install]  
Alias=FXL-front-app.service
```

### **FXL-rest-api.service (п.3.3)**

```
[Unit]  
[Unit]  
Description=FXL rest-API Service  
After=ignite.service  
  
[Service]  
ExecStartPre=/opt/ignite/Ignite-SE-17.6.0-beta1/check_fxl-ignite_ready.sh  
WorkingDirectory=/opt/rest-api  
User=root  
PrivateDevices=yes  
ProtectSystem=full  
Type=simple  
ExecReload=/bin/kill -HUP $MAINPID  
KillMode=mixed  
KillSignal=SIGTERM  
TimeoutStopSec=10  
Environment="JAVA_HOME=/opt/java/jdk-pro-11.0.25-full"  
Environment="PORT=8091"  
#Environment="PORT=8099"  
Environment="URL_IGNITE=localhost:10800/TBG"  
Environment="SESSION_USER=USR2"  
ExecStart=/opt/java/jdk-pro-11.0.25-full/bin/java -jar ./fxl-rest-api-1.0.0.jar -  
DIGNITE_ALLOW_DML_INSIDE_TRANSACTION=true  
SyslogIdentifier=FXL-rest-api  
Restart=on-failure  
RestartSec=5  
  
[Install]  
Alias=FXL-rest-api.service
```

## 5. Управление сервисами производится с помощью утилиты `systemctl`. Краткая справка по утилите `systemctl`

`start PATTERN...`

Запускает (активирует) один или более сервисов (далее «юнитов»), указанных в командной строке. Следует иметь в виду, что шаблоны поиска работают только с первичными именами юнитов, загруженных в память. Юниты, которые неактивны и при этом не завершили работу с ошибкой, обычно не хранятся в памяти, и не будут найдены ни одним шаблоном. Также, в случае экземпляров юнитов, `systemd` зачастую не знает имена экземпляров, которые еще не были запущены. Следовательно, использование шаблонов поиска со `start` имеет ограниченное применение. Также не производится поиск по псевдонимам юнитов.

`stop PATTERN...`

Останавливает (деактивирует) один или более юнитов, указанных в командной строке.

`reload PATTERN...`

Перезагружает конфигурацию всех юнитов, перечисленных в командной строке. Это затрагивает только конфигурацию связанных служб. Для перезагрузки конфигурации юнита следует использовать команду `daemon-reload`. Эту команду не следует путать с командой `daemon-reload`.

`restart PATTERN...`

Перезапускает один или более юнитов, указанных в командной строке. Если юниты не запущены, они будут запущены.

`try-restart PATTERN...`

Перезапускает один или более юнитов, указанных в командной строке. Если запущенных юнитов нет, ничего не происходит.

`reload-or-restart PATTERN...`

Перезагружает один или более юнитов, если они поддерживают данную операцию. В противном случае, перезапускает их. Незапущенные юниты будут запущены.

`try-reload-or-restart PATTERN...`

Перезагружает один или более юнитов, если они поддерживают данную операцию. В противном случае, перезапускает их. Не затрагивает незапущенные юниты.

`isolate NAME`

Запускает указанный юнит и все его зависимости, останавливает все остальные юниты. Если имя юнита указано без расширения, предполагается расширение `".target"`.

Это подобно изменению уровня выполнения в традиционной системе `init`. Команда `isolate` немедленно останавливает процессы, которые не включены в новом юните, в том числе, потенциально, используемое графическое окружение или терминал. Следует иметь в виду, что это разрешено только для юнитов с включенным параметром `AllowIsolate=`. См. `systemd.unit(5)`

`kill PATTERN...`

Посылает сигнал одному или более процессам юнита. Для указания конкретных процессов, которые нужно завершить, используется опция --kill-who=. Чтобы указать, какой сигнал отправить, следует использовать --signal=.

`is-active PATTERN...`

Проверяет, являются ли какие-либо из указанных юнитов активными (например, запущенными). Возвращает код возврата 0, если хотя бы один юнит активен, в противном случае возвращает ненулевое значение. Также выводит информацию о статусе юнитов, если не указана опция --quiet.

`is-failed PATTERN...`

Проверяет указанные юниты на состояние "failed". Возвращает код возврата 0, если хотя бы один такой юнит найден, в противном случае возвращает ненулевое значение. Также выводит информацию о статусе юнитов, если не указана опция --quiet.

`status [PATTERN... |PID...]`

Выводит сжатую информацию о статусе одного или более юнитов, за которой следуют самые недавние данные из журнала. Если юниты не указаны, выводит информацию о состоянии системы. При использовании с опцией --all также выводит статус всех юнитов (с учетом ограничений, наложенных опцией -t). Если в качестве параметра передан PID процесса, будет выведена информация о юните, которому принадлежит данный процесс. Данная функция генерирует человекочитаемый вывод. Для машиночитаемого вывода следует использовать команду show. По умолчанию, одновременно отображается 10 строк вывода, сокращенных многоточиями, чтобы уместиться в окно терминала. Это можно изменить с помощью команд --lines и --full (см. выше). Также стоит заметить, что команды journalctl --unit=NAME или journalctl --user-unit=NAME используют похожий фильтр сообщений и могут быть более удобны в использовании.

`show [PATTERN... |JOB...]`

Выводит список свойств одного или более юнитов, заданий или самого менеджера. Если не передан аргумент, будут выведены свойства менеджера. Если указано имя юнита, будут выведены свойства данного юнита. Если указан ID задания, будут выведены свойства данного задания. По умолчанию пустые свойства не отображаются, для их отображения следует использовать опцию --all. Для отображения только определенных свойств следует использовать опцию --property=. Команда предназначена для получения машиночитаемого вывода. Для получения человеко читаемого вывода следует использовать команду status.

## 6. Логирование сервисов программного обеспечения FXL 3.0

Логи сервера приложений в каталоге `/opt/axiom-libercat9-9.0.91/logs`

Логи прочих компонентов находятся в каталоге `/var/log/flexsoft/`

`fxl-ignite.log` – лог «DataGrid»

`fxl-rest-api.log` – лог Rest-API

`fxl-ydb-storage.log` – лог YDB storage server

`fxl-ydb-db.log` – лог YDB БД