

СИСТЕМА МЕЖОПЕРАТОРСКИХ РАСЧЕТОВ

Руководство по установке

Версия 10.1.0

nexign

Настоящая документация может быть использована только для поддержки работоспособности продуктов, установленных на основании договора с АО «Нэксайн». Документация может быть передана на основании договора, по которому производится (производилась или будет производиться) установка продуктов, или явно выраженного согласия АО «Нэксайн» на использование данной документации. Если данный экземпляр документации попал к вам каким-либо иным образом, пожалуйста, сообщите об этом в АО «Нэксайн» по адресу, приведенному ниже.

Все примеры, приведенные в документации (в том числе примеры отчетов и экранных форм), составлены на основании тестовой базы АО «Нэксайн». Любое совпадение имен, фамилий, названий компаний, банковских реквизитов и другой информации с реальными данными является случайным.

Все встречающиеся в тексте торговые знаки и зарегистрированные торговые знаки являются собственностью их владельцев и использованы исключительно для идентификации программного обеспечения или компаний.

Данная документация может не отражать некоторых модификаций программного обеспечения. Если вы заметили в документации ошибки или опечатки или предполагаете их наличие, пожалуйста, сообщите об этом в АО «Нэксайн».

Все имущественные авторские права сохраняются за АО «Нэксайн» в соответствии с действующим законодательством.

© АО «Нэксайн», 1992–2023

АО «Нэксайн»

Россия, 199155, Санкт-Петербург, ул. Уральская, д.4 лит.Б, помещение 22Н

Тел.: + 7 (812) 326-12-99; факс: + 7 (812) 326-12-98.

office@nexign.com; www.nexign.com

Содержание

1. Общие сведения	4
2. Предварительные условия	5
3. Установка продукта вручную	6
3.1. Установка серверной части продукта	6
3.2. Установка GUI	6
3.2.1. Настройка nginx	6
3.2.2. Настройка ITC_COMMON_GUI	9
3.3. Установка backend	9
3.3.1. Общие настройки	10
3.3.2. Настройки itc_authorization	10
3.3.3. Настройки itc_brd_loader	11
3.3.4. Настройки itc_imp_parser	11
3.3.5. Настройки itc_orchestrator	12
3.3.6. Настройка ZooKeeper	12
3.4. Установка ITC_EVENT_PROC	12
4. Обновление	14
5. Запуск и остановка продукта	15

1. Общие сведения

В настоящем документе описаны действия, необходимые для установки продукта.

2. Предварительные условия

Для корректной установки продукта обеспечьте наличие следующего программного обеспечения:

- операционная система: Red Hat® Enterprise Linux (RHEL) версии 7.0 или РЕД ОС версии 7.0;
- база данных: Oracle Database 19c или PostgreSQL не ниже 15;
- Java 8;
- Liquibase версии не ниже 4.17;
- Apache ZooKeeper версии не ниже 3.6.3;
- NGINX версии 1.20.1.

3. Установка продукта вручную

Порядок установки продукта:

1. [Установите серверную часть продукта.](#)
2. [Установите GUI.](#)
3. [Установите backend.](#)
4. [Установите клиентскую часть ITC_EVENT_PROC.](#)

3.1. Установка серверной части продукта

1. Создайте базу данных PostgreSQL (владелец – отдельный пользователь), внутри создайте одноимённую с пользователем схему, которая принадлежит тому же пользователю.
2. На созданную базу данных установите компоненты ITC_BASE_SCR и ITC_BASE_UTILS;
3. Установите схему ITC_SCR;
4. На схему ITC_SCR установите серверные части ITC_CONTROLS, ITC_ADMIN, ITC_VOLUME_PROC, ITC_EVENT_PROC, ITC_BRD_API, ITC_BRD_VALIDATOR, ITC_BRD_LOADER, расположенные в каталоге /db/postgresql:
 - в liquibase.properties настройте параметры подключения к базе данных на созданного пользователя;
 - в properties.xml настройте schema_name и остальные параметры табличных пространств;
 - запустите install.bat, либо вызовите команду liquibase --defaultsFile="liquibase.properties" --changeLogFile="changelog.xml" update.
5. Создайте новые базы данных для компонентов ITC_PARSER и ITC_ORCHESTRATOR, на них установите ITC_BASE_SCR и ITC_BASE_UTILS;
6. Установите схемы ITC_IMP_SCR, ITC_ORCHESTRATOR_SCR на соответствующие созданные базы данных;
7. На схему ITC_IMP_SCR установите серверные части ITC_IMP_PARSER_API, ITC_IMP_PARSER;
8. На схему ITC_ORCHESTRATOR_SCR установите серверные части ITC_ORCHESTRATOR_API, ITC_ORCHESTRATOR.

3.2. Установка GUI

Для установки пользовательского интерфейса:

1. Настройте nginx.
2. Установите компонент ITC_COMMON_GUI.

3.2.1. Настройка nginx

1. В файле nginx.conf в секции http вместо стандартных секций server укажите:

```
include /etc/nginx/sites-enabled/*.conf;
```

2. В каталоге /etc/nginx создайте каталоги sites-available и sites-enabled.
3. В каталоге sites-enabled создайте конфигурационный файл itc-proxy.conf:
 - верхние секции настройте в соответствии с внутренним регламентом настройки HTTP(S)-

сервера;

- в секции `root` укажите путь до каталога с файлами сборки GUI приложения;
- в секции `location/` укажите `try_files $uri /index.html` (секция необходима для работы обновления страницы средствами браузера);
- в секциях `location/*` настройте перенаправление на те адреса, под которыми будут настроены backend приложения:

location	backend
<code>/auth</code>	<code>itc_authorization</code>
<code>/ItcOrchestratorApi</code>	<code>itc_orchestrator_api</code>
<code>/ItcImpParserApi</code>	<code>itc_imp_parser_api</code>
<code>/ItcBrdApi</code>	<code>itc_brd_api</code>
<code>/ItcControlsApi</code>	<code>itc_controls_api</code>

Пример заполнения:

```
server {
    listen      80;
    listen     [::]:80;
    server_name _;
    root       /srv/itc-site/build;
    index      index.html;

    location / {
        try_files $uri /index.html;
    }

    location /auth {
        proxy_pass http://127.0.0.1:5000;
    }

    location /ItcOrchestratorApi {
        rewrite ^/ItcOrchestratorApi/(.*) /$1 break;
        proxy_pass http://127.0.0.1:5010;
    }

    location /ItcImpParserApi {
        rewrite ^/ItcImpParserApi/(.*) /$1 break;
        proxy_pass http://127.0.0.1:5020;
    }

    location /ItcBrdApi {
        rewrite ^/ItcBrdApi/(.*) /$1 break;
        proxy_pass http://127.0.0.1:5030;
    }

    location /ItcControlsApi {
        rewrite ^/ItcControlsApi/(.*) /$1 break;
        proxy_pass http://127.0.0.1:5030;
    }

    error_page 404 /404.html;

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }
}
```

4. Создайте симлинк на настроенный конфигурационный файл:

```
sudo ln -s /etc/nginx/sites-available/itc-proxy.conf  
/etc/nginx/sites-enabled/itc-proxy.conf
```

5. Проверьте корректность конфигурационного файла:

```
sudo nginx -t
```

6. Запустите nginx:

```
sudo systemctl start nginx.service
```

В случае изменений конфигурационного файла необходим перезапуск nginx, для этого используйте команду:

```
sudo systemctl restart nginx.service
```



Примечание.

Данную команду `sudo` можно разрешить выполнять пользователю, от имени которого обычно будет происходить настройка ITC.

3.2.2. Настройка ITC_COMMON_GUI

Создайте каталог и поместите туда файлы `itc_common_gui` – веб-интерфейс ITC будет доступен по адресу сервера, в соответствии с настройками nginx.

3.3. Установка backend

1. В каталоге, где будут размещаться backend компонентов создайте подкаталоги для каждого:
 - `itc_authorization`;
 - `itc_orchestrator`;
 - `itc_orchestrator_api`;
 - `itc_imp_parser`;
 - `itc_imp_parser_api`;
 - `itc_brd_validator`;
 - `itc_brd_loader`;
 - `itc_brd_api`;
 - `itc_controls_api`.
2. В каждом подкаталоге разместите `*.jar`-файл одноименного компонента.
3. Для каждого компонента в подкаталоге `config` создайте файлы `application.properties` и `bootstrap.properties`.
4. Выполните настройку компонентов.

3.3.1. Общие настройки

Для каждого компонента в файле `application.properties` настройте:

```
server.address=localhost
server.port=5000
# Настройки логирования
logging.file.name=logs/itc_authorization.log
logging.file.max-size=10000000
logging.pattern.file: "%d %-5level [%thread] %logger : %msg%n"
# Ключ для шифрования пароля пользователя базы данных
secretkey=mySecretKey
```



Примечание.

`server.address` – адрес, по которому приложение слушает запросы. Если `nginx` установлен на внешнем сервере, то у `itc_authorization` и для всех `*_api` укажите внешний IP, а для других `backend` – адрес, где запущен `itc_orchestrator`.

В файле `bootstrap.properties` укажите строку соединения с ZooKeeper в виде:

```
spring.cloud.zookeeper.connect-string=<server>:<port>
```

Если файл не создан или настройки нет, то по умолчанию используется `127.0.0.1:2181`.

3.3.2. Настройки `itc_authorization`

В файле `application.properties`:

```
# известный адрес LDAP для Active Directory
ad.url=ldap://<url>/
# в cmd пишем whoami /upn, после знака @ будет указан домен
ad.domain=<domain>
# в cmd пишем whoami /fqdn для вывода информации о пользователе и в
конце смотрим корневой DN
ad.rootDN=<rootDN>
```

В файл `application.yaml`:

```
itc:
  relations:
    -
      appModule: itc_imp_parser_api
      adGroup: <группа active directory>
    -
      appModule: itc_brd_api
      adGroup: <группа active directory>
    -
      appModule: itc_orchestrator_api
      adGroup: <группа active directory>
    -
      appModule: itc_controls_api
      adGroup: <группа active directory>
```

где `<группа active directory>` – группу active directory, в которую включены доменные пользователи, которым необходим доступ до определенного API через веб-интерфейс.

3.3.3. Настройки `itc_brd_loader`

В файле `application.properties`:

```
# Префикс и постфикс в создаваемых сущностях ITC
entity.name.prefix=[
entity.name.postfix=]
# Разделитель атрибутов в создаваемых сущностях ITC
entity.name.separator=-
# DRGR_ID группы направлений для совокупных зон (должен относиться к
типу группы, полученному из таблицы APP_PARAMETERS с PRMT_ID =
20010050")
drgrIdL2=<DRGR_ID>
```

3.3.4. Настройки `itc_imp_parser`

В файле `application.properties`:

```
# Символ разделения дробной и десятичной части
decimal.symbol=,
#формат даты для вычисляемых на уровне базы данных полей
itc.date.format=DD.MM.YYYY (DD.MM.YYYY HH:mi:ss)
#Установка локализации
itc.imp.default.locale=ru
```

3.3.5. Настройки itc_orchestrator

В файле application.properties:

```
# Частота запроса на запуск джоба (мс)
itc.queue.start.interval = 10000
# Частота запроса на формирование очереди (мс)
itc.queue.build.interval = 10000
```

3.3.6. Настройка ZooKeeper

Для каждого компонента настройте параметры соединения с базой данных для определённого сценария:

- ps/config/apps/<component_name>/<scenary_name>/dbUrl – URL в формате jdbc:oracle:thin:@<сервер>:<порт>/<имя экземпляра базы данных>;
- ps/config/apps/<component_name>/<scenary_name>/dbUser – имя пользователя, созданного с помощью скриптов Role_User_Part в поставке продукта;
- ps/config/apps/<component_name>/<scenary_name>/dbPassword – зашифрованный пароль пользователя; для шифрования пароля запустите утилиту PasswordManager.jar (из компонента itc_orchestrator) следующим образом:

```
java -jar PasswordManager.jar encrypt mySecretKey <пароль>
```

где mySecretKey – стандартный ключ шифрования.

Для использования другого ключа, в настройках каждого приложения поменяйте его с помощью параметра secretkey в файле application.properties.

3.4. Установка ITC_EVENT_PROC

1. Создайте целевой каталог (например: /opt/itc/itc_tar) и скопируйте в него исполняемый файл компонента и конфигурационный файл, входящий в поставку, файлы динамически загружаемых библиотек, предназначенных для конвертации файлов входных учетных записей (dll-файлы);
2. Создайте каталоги для размещения:
 - входных файлов от коммутатора (например: /opt/itc/itc_tar/in);
 - обработанных файлов (например: /opt/itc/itc_tar/out);
 - отвергнутых файлов (например: /opt/itc/itc_tar/bad);
 - файлов работы системы обработки вызовов (например: /opt/itc/itc_tar/log).



Примечание.

Исполняемый файл должен иметь права на чтение, создание и запись файлов в созданных каталогах. Имена каталогов не должны содержать символов пробела.

3. Отредактируйте настроечные параметры компонента и файл конфигурации:

- укажите пути до созданных каталогов;
- настройте все необходимые параметры файла конфигурации;
- добавьте в процедуру старта операционной системы запуск компонента, с указанием в качестве параметра полного пути до конфигурационного файла.

4. Обновление

Вначале установите продукт на тестовую базу данных, а уже после тестового просмотра устанавливайте его на оригинальную рабочую базу данных (production). Кроме этого, устанавливая продукт на базу production, желательно предусмотреть резервное время в случае выявления неисправностей.

Для успешной установки новых версий ИТС:

- внимательно изучите документацию;
- установите серверную часть;
- установите клиентские приложения;
- создайте роли и пользователей компонентов.

5. Запуск и остановка продукта

Для запуска компонентов используйте команду:

```
java -jar <JAR_NAME>
```

Для запуска на Linux в фоновом режиме используйте команду:

```
nohup java -jar <JAR_NAME> > /dev/null 2> /dev/null &
```

Перед запуском компонента ITC_BRD, в интерфейсе ITC необходимо вручную создать следующие сущности:

- класс времени **По умолчанию**;
- схему часов **По умолчанию** с единственным входящим классом времени **По умолчанию**.

Остановка продукта осуществляется штатными средствами операционной системы.