Compatible with

STC ONE

SMaRT AFIX

Powered by STC

SMaRT AFIX

Installation Guide

v. July 2019

Support Services

For general support on this product, contact your system administrator or help desk. For up-to-date documentation, visit the STC Documentation Portal at https://documentation.stchome.com/.

Additional Information

This documentation describes the following: SMaRT AFIX installation.

(20190730)

Table of Contents

SMaRT AFIX Installation Instructions

The following provides an overview of the process for installing the SMaRT AFIX application.

Components

SMaRT AFIX is composed of several server components. The following includes their functions and brief technical descriptions.

Standalone Forecaster (SAF)

SAF is a web application that evaluates patient and vaccination data to produce a vaccination forecast for that patient. It provides both a user interface and a SOAP web service. For SMaRT AFIX, only the web service is used.

SAF runs under Apache Tomcat 7.0 and is deployed as a `.war` file. It utilizes an Oracle Express Database (XE) to store pre-configured vaccination schedules and configuration data.

Standalone Forecaster is installed on the *Service* box.

PostgreSQL

SMaRT AFIX utilizes PostgreSQL 9.4 as a database. An additional Postgres module, pg_partman, is required. The name of the database is *afixdb*. It is created via several `.sql` scripts. See SMaRT AFIX Database Initialization - Linux or Windows for the installation instructions.

JasperReports Server (JRS)

JRS is a web application that provides the reports for SMaRT AFIX. These reports are imported into JRS via the JRS Management Console. See SMaRT AFIX JasperReports Deployment - Linux or Windows for installation instructions.

Jaspersoft ETL (JETL)

Patient, vaccination, and provider data are batch-loaded into the SMaRT AFIX database on a nightly basis. Once this data has loaded, it is run through the Standalone Forecaster and aggregated into tables that are designed for reporting. These batch processes are scheduled and executed by Jaspersoft ETL.

SMaRT AFIX Application

The UI and application server logic are provided via the SMaRT AFIX application. This is a Node.js application and uses Keycloak for user authentication.

Keycloak

User authentication is provided by Keycloak. It is used by all STC Suite applications for Single Sign-On (SSO). Each application requires a client within Keycloak. The client contains references (URLs, ports, etc.) that are specific to each individual SMaRT AFIX application.

New Application Installations

The following refers to new application installations, not necessarily to upgrade installations. The instructions differ between Linux and Windows installations as well. If you are building a new Linux Server, see the Build a Linux (CentOS) Server instructions.

> **NOTE:** The SMaRT AFIX components should be installed in the order listed below.

Component	Server	Linux	Windows
Keycloak Installation	Service (SVC)	See either *Keycloak Installation Guide for STC-Hosted Clients* or *Keycloak Installation Guide for Self-Hosted Clients*.	
Standalone Forecaster	Service (SVC)	Forecaster Installation Instructions - Linux	Forecaster Installation Instructions - Windows
PostgreSQL Server (if not already installed)	Database and Batch (DBB)	Postgres Database Installation - Linux	Postgres Database Installation - Windows
SMaRT AFIX Database	Database and Batch (DBB)	SMaRT AFIX Database Initialization - Linux	SMaRT AFIX Database Initialization - Windows
SMaRT AFIX Online Tool	Service (SVC)	SMaRT AFIX Online Tool Web Service Installation Instructions - Linux	SMaRT AFIX Online Tool Web Service Installation Instructions - Windows
Jaspersoft ETL Server (if not already installed) (**NOTE:** Initial ETL loads can take several days; it's recommended to have them running while other components are being installed.)	Database and Batch (DBB)	Jaspersoft ETL Installation - Linux	Jaspersoft ETL Installation - Windows
SMaRT AFIX ETL	Database and Batch (DBB)	SMaRT AFIX Jaspersoft ETL Deployment - Linux	SMaRT AFIX Jaspersoft ETL Deployment - Windows
JasperReports Server (if not already installed)	Application (APP)	JasperReports Server Installation - Linux	JasperReports Server Installation - Windows
SMaRT AFIX Reports	Application (APP)	SMaRT AFIX JasperReports Deployment - Linux	SMaRT AFIX JasperReports Deployment -

			Windows
Node.js Server (if not already installed)	Application (APP)	Node.js Installation - Linux	Not supported
SMaRT AFIX Application Installation	Application (APP)	SMaRT AFIX Application Installation Instructions - Linux	SMaRT AFIX Application Installation Instructions - Windows
Keycloak Configuration for SMaRT AFIX	Service (SVC)	Keycloak Configuration for SMaRT AFIX (Linux and Windows)	

Linux Instructions

The following instructions are for the Linux operating system.

Build a Linux (CentOs) Server

The instructions below explain how to install CentOS 7 Minimal and configure it. This is a base configuration suitable for reuse for all servers. An image or snapshot should be taken, if possible (such as in a cloud or AWS environment), so that these steps do not need to be performed again. Initially, the server is usually set up with the following:

- CentOS 7 Minimal (64-bit)
- 64-bit CPU
- 2GB RAM minimum
- 50GB HD minimum

These requirements are only for the initial setup. CPU, RAM, and storage requirements vary based on server use. Check the STC Suite specifications document prior to installing additional components on the server.

The instructions below explain how to build a base server from scratch. After following these steps, an image should be created so that these steps do not need to be performed again.

Install CentOS 7 Minimal and Configure

CentOS 7 Minimal is installed in order to be able to start from a lean Linux installation. Packages and software components are added as needed.

1. Disable the SSH login as root - Run the remainder of this steps as root or use `sudo`.
2. Edit `sshd_config`. Vi is used here, but any editor can be used.

   ```
   sudo vi /etc/ssh/sshd_config
   ```

3. On the line that says *#PermitRootLogin yes*, update it to the following (remember to remove the #):

   ```
   PermitRootLogin no
   ```

4. Restart the sshd service:

   ```
   sudo service sshd restart
   ```

5. By default, SELinux is set to be enforced. Change this by running:

```
sudo setenforce 0
```

6. Edit the following file so *permissive* persists across reboots:

```
sudo vi /etc/selinux/config
```

7. Change the line *SELINUX=enforcing* to the following and save the changes:

```
SELINUX=permissive
```

8. Make sure the software is completely up to date:

```
sudo yum update
```

9. Finally, install some basic packages:

```
sudo yum -y install net-tools gcc-c++ openssl-devel make git unzip bzip2
  wget vim-enhanced ntp epel-release tmux dos2unix patch
```

Forecaster Installation Instructions - Linux

The installation of the Standalone Forecaster (SAF) is composed of three major tasks: install Oracle Express, install the SAF database into Oracle Express, and install the Forecaster web application into Tomcat. All three components are installed on the *Service* machine (SVC).

Install Oracle Express

Perform the following steps as **root** user.

1. Install the OS libraries and packages on the Linux machine on which XE will be installed:

```
yum install bc binutils flex gcc glibc glibc-common glibc-devel libaio
  libstdc libXtst make sysstat
```

2. Create the Oracle user and the Oinstall group:

```
/usr/sbin/groupadd oinstall
  /usr/sbin/groupadd dba
  /usr/sbin/useradd -g oinstall -G dba oracle
  passwd oracle (Oracle123)
  id oracle
```

3. Download the Oracle XE rpm, `oracle-xe-11.2.0-1.0.x86_64.rpm.zip`, from http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html and copy it to the server.

4. In the directory where the `oracle-xe-11.2.0-1.0.x86_64.rpm.zip` file is copied on the server, run these commands:

```
unzip -q oracle-xe-11.2.0-1.0.x86_64.rpm.zip
```

```
 cd Disk1
 rpm -i oracle-xe-11.2.0-1.0.x86_64.rpm
```

5. If the installation is successful, run this command as root. When it asks for an HTTP port, change it to 7080 so that it doesn't conflict with other installation applications. Set the initial SYS & SYSTEM password to WELCOME.

```
/etc/init.d/oracle-xe configure
```

6. Verify the connection to the database:

```
cd /u01/app/oracle/product/11.2.0/xe/bin
 . ./oracle_env.sh
 sqlplus system/<password>©XE
```

7. Get the database connection information:

```
[oracle@ ~]$ cd /u01/app/oracle/product/11.2.0/xe/network/admin
 [oracle@ ~]$ cat tnsnames.ora
```

Install the SAF Database (New Installation)

Copy the `/SVC/OXE/db/` directory and all of its contents to your server under the Oracle user's home directory.

It is recommended you change the owner and permissions of the db directory by running the below commands as root:

```
chown -R oracle:oinstall db
 chown -R 775 db
```

To install the SAF database, navigate to the `/db/AFIX_CDSI_Install/AFIX_CDSI_v5.16.9_Install/` directory and execute the following command as user oracle. Note that your path to `bin/oracle/env.sh` may differ.

```
./u01/app/oracle/product/11.2.0/xe/bin/oracle_env.sh
 sh Install.sh
```

Next, navigate to `/db/Patches/` and run the following:

```
/patch_v5_17_CDSI/forecast.sh
 /patch_v5_18_CDSI/forecast.sh
/AFIX_March_2019/forecast.sh
```

Run Database Patches (Upgrades)

To update the Forecaster database to the most current patch version, navigate to `/db/Patches/` and run the following:

```
/AFIX_March_2019/forecast.sh
```

Install the Web Application

Before installing the SMaRT AFIX web application, Java 7 must be installed on the SVC server and Tomcat 7.0 must also be installed and running on the SVC server.

1. Edit the `conf/context.xml` file - Add the lines below (between the <Context></Context> tags) to the `CATALINA_HOME/conf/context.xml` file. The added lines should be placed within the <Context></Context> tags in the file.

```
<Context>
    <Resource name="jdbc/com/stc/forecaster"
    auth="Container"
    type="javax.sql.DataSource"
    driverClassName="oracle.jdbc.OracleDriver"
    url="jdbc:oracle:thin:@localhost:1521/XE"
    username="SAFUSER"
    password="SAF"
    maxActive="100"
    maxIdle="10"
    maxWait="-1"/>
 </Context>
```

2. Navigate to the `Application /SVC/SAF` directory and execute this command:

```
$ cp forecaster.war $CATALINA_HOME/webapps/forecaster.war
```

3. Restart Tomcat.

4. Set the number of threads to the number of processor cores available on your server - Edit `$CATALINA_HOME/webapps/forecaster/WEB-INF/spring-ws-servlet.xml` and set the thread count (see the second line of code below):

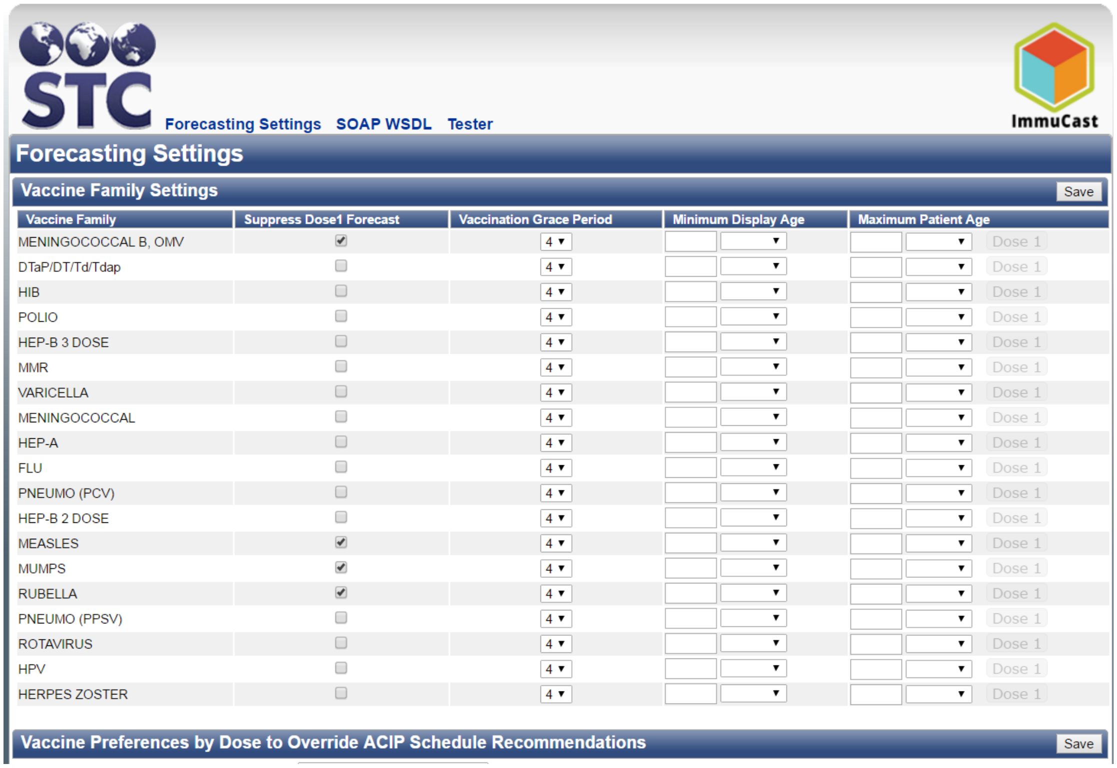
```
<bean id="forecastingEndpoint12"
 class="com.stchome.saf.ws.endpoint.ForecasterServiceEndpoint12"
 scope="prototype">
    <constructor-arg value="1" /><!-- Thread count -->
    <constructor-arg value="60" /><!-- Thread pool termination timeout in
 seconds-->
 </bean>
```

5. Increase the Tomcat memory usage on your server - Edit `$CATALINA_HOME/bin/setenv.sh` and set the thread count (see the second line of code below):

```
export CATALINA_OPTS="$CATALINA_OPTS -Xms2048m"
 export CATALINA_OPTS="$CATALINA_OPTS -Xmx2048m"
```

6. Restart Tomcat.

7. Verify the application is running correctly by navigating to `http://[YOUR SERVER]/forecaster`. The UI displays if the application was successfully installed and configured.



Forecasting Settings

Vaccine Family Settings Save

Vaccine Family	Suppress Dose1 Forecast	Vaccination Grace Period	Minimum Display Age	Maximum Patient Age
MENINGOCOCCAL B, OMV	☑	4 ▾	▾	▾ Dose 1
DTaP/DT/Td/Tdap	☐	4 ▾	▾	▾ Dose 1
HIB	☐	4 ▾	▾	▾ Dose 1
POLIO	☐	4 ▾	▾	▾ Dose 1
HEP-B 3 DOSE	☐	4 ▾	▾	▾ Dose 1
MMR	☐	4 ▾	▾	▾ Dose 1
VARICELLA	☐	4 ▾	▾	▾ Dose 1
MENINGOCOCCAL	☐	4 ▾	▾	▾ Dose 1
HEP-A	☐	4 ▾	▾	▾ Dose 1
FLU	☐	4 ▾	▾	▾ Dose 1
PNEUMO (PCV)	☐	4 ▾	▾	▾ Dose 1
HEP-B 2 DOSE	☐	4 ▾	▾	▾ Dose 1
MEASLES	☑	4 ▾	▾	▾ Dose 1
MUMPS	☑	4 ▾	▾	▾ Dose 1
RUBELLA	☑	4 ▾	▾	▾ Dose 1
PNEUMO (PPSV)	☐	4 ▾	▾	▾ Dose 1
ROTAVIRUS	☐	4 ▾	▾	▾ Dose 1
HPV	☐	4 ▾	▾	▾ Dose 1
HERPES ZOSTER	☐	4 ▾	▾	▾ Dose 1

Vaccine Preferences by Dose to Override ACIP Schedule Recommendations Save

Postgres Database Installation - Linux

The information below describes how to install the Postgres database on Linux CentOS 7, as well as how to modify relevant Postgres server configuration settings for SMaRT AFIX.

Install on Linux CentOS 7

Currently, SMaRT AFIX is using CentOS 7 as its Linux distribution. CentOS 7 has Postgres 9.2 included with its default repositories. Postgres 9.4 is required for SMaRT AFIX. The following steps detail how to upgrade to Postgres 9.4.

Before continuing, it is assumed that you have already followed the Build a Linux (CentOS) Server instructions.

1. Exclude the `postgresql` package from the CentOS base repositories - Edit `/etc/yum.repos.d/CentOS-Base.repo` and add the exclude lines below to the [base] and [updates] sections, then save the changes. Note that this must be `sudo`'d to act as root to edit the file:

```
[base]
 name=CentOS-$releasever - Base
 mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=
 os&infra=$infra
 #baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
 gpgcheck=1
 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
 exclude=postgresql*

 #released updates
 [updates]
 name=CentOS-$releasever - Updates
 mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=
 updates&infra=$infra
 #baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
 gpgcheck=1
 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
 exclude=postgresql*
```

2. Install the PGDG RPM file:

```
sudo yum install -y
 https://download.postgresql.org/pub/repos/yum/9.4/redhat/rhel-7-
 x86_64/pgdg-centos94-9.4-2.noarch.rpm
```

3. Trigger the packages update:

```
sudo yum update
```

4. Install the Postgres 9.4 server/packages:

```
sudo yum install -y postgresql94-server postgresql94-libs
```

5. Initialize the Postgres database:

```
sudo /usr/pgsql-9.4/bin/postgresql94-setup initdb
```

6. Set Postgres to run as a service and start Postgres:

```
sudo systemctl enable postgresql-9.4
 sudo systemctl start postgresql-9.4
```

Modify Postgres Server Configuration Settings

Some of the Postgres server configuration settings must be modified.

1. Act as the *postgres* user to perform admin level changes:

```
sudo su - postgres
 cd 9.4/data/
```

2. Add the Postgres `/bin/` directory to your path. Edit the `~/.bash_profile` and add the following to the end of the file, then save the file:

```
PATH=/usr/pgsql-9.4/bin:$PATH
 export PATH
```

3. Run the following to update the bash environment PATH:

```
source ~/.bash_profile
```

4. Edit `postgres.conf` with your favorite editor and add the following line right after **# - Connection Settings -**, then save the file:

   ```
   listen_addresses = '*'
   ```

5. Edit the `pg_hba.conf` file to allow connections from other STC servers/applications and secure the connection using md5 instead of clear text passwords. Change the following line:

   ```
   host   all      all      127.0.0.1/32      md5
   ```

   to something like the following, which will enforce md5-encrypted password communication. Let's assume your internal sub-net falls under 192.168.x.x, which would update the line to:

   ```
   host   all      all      192.168.0.0/16       md5
   ```

6. Save the changes. If you are technical and familiar enough with CIDR notation, you can further customize and restrict connection access as needed.
7. Trigger Postgres to reload with the new configuration changes by running:

   ```
   pg_ctl restart
   ```

8. Complete the installation by initializing the database (see SMaRT AFIX Database Initialization - Linux).

Add Fuzzy Search Module to Postgres DB

In independent deployments of SMaRT AFIX, provider data is pulled from the SMaRT AFIX Postgres database. This requires that an additional plugin be added to the database to enable fuzzy searching.

1. Log in to the database server (dbb) as a root user:

   ```
   sudo su - root
   ```

2. Install the contrib package:

   ```
   yum install postgresql-contrib
   ```

3. Enter "Y" when prompted.
4. Switch to postgres user:

   ```
   sudo su - postgres
   ```

5. Log in to afixdb and create an extension for the fuzzy search:

   ```
   psql -d afixdb
    afixdb=# create extension fuzzystrmatch;
    CREATE EXTENSION
   ```

SMaRT AFIX Database Initialization - Linux

The information below describes how to initialize the database for the first time, how to upgrade/migrate an existing database, and how to set up daily jobs.

Prerequisite Tasks

Perform the following tasks before initializing the database:

1. Confirm that PostgreSQL has been installed.
2. Download the release distribution zip file from `/DBB/POSTGRES` and extract the database scripts for the database server.
3. As a root user, give read, write, and execute permissions on the flyway command - Navigate to the `/DBB/POSTGRES` directory and execute this command:

   ```
   cd /DBB/POSTGRES/
    chmod 755 flyway
   ```

4. As a root user, install the dos2unix package:

   ```
   yum install dos2unix
   ```

5. As a Postgres user, run the dos2unix command for the flyway file to make it compatible for Linux - Navigate to the `/DBB/POSTGRES` directory and execute this command:

   ```
   cd /DBB/POSTGRES/
    dos2unix flyway
   ```

Initialize a New Database (for the First Time Only)

If this is the first time you are initializing the database, follow the steps below. Otherwise, see the Upgrade/Migrate an Existing Database instructions.

1. Act as the postgres user:

   ```
   sudo su - postgres
   ```

2. Prepare the tablespace by creating the directories that will be used by afixdb:

   ```
   mkdir -p /var/lib/pgsql/9.4/afixdb
    mkdir -p /var/lib/pgsql/9.4/afixdb/data
    mkdir -p /var/lib/pgsql/9.4/afixdb/olapdata
   ```

3. Add postgres to your path:

   ```
   export PATH=$PATH:/usr/pgsql-9.4/bin
   ```

4. Create the afixdb by navigating to the `/DBB/POSTGRES` directory and executing this command:

```
cd /DBB/POSTGRES/
 psql -a -f setup_database.sql
```

5. Create the database users and roles:

```
psql -d afixdb -a -f setup_objects.sql
```

6. Create the APP configuration schema:

```
./flyway -url=jdbc:postgresql://localhost:5432/afixdb -schemas=app -
 sqlMigrationPrefix=APP_V -user=afix migrate
```

7. Create the ODS schema:

```
./flyway -url=jdbc:postgresql://localhost:5432/afixdb -schemas=ods -
 sqlMigrationPrefix=ODS_V -user=afix migrate
```

8. Create the OLAP schema:

```
./flyway -url=jdbc:postgresql://localhost:5432/afixdb -schemas=olap -
 sqlMigrationPrefix=OLAP_V -user=afix migrate
```

Upgrade/Migrate an Existing Database

If this is not the first time you are initializing the database, follow the steps below. Otherwise, see the Initialize a New Database (for the First Time Only) instructions.

1. Act as the postgres user:

```
sudo su - postgres
```

2. Migrate the APP configuration schema:

```
./flyway -url=jdbc:postgresql://localhost:5432/afixdb -schemas=app -
 sqlMigrationPrefix=APP_V -user=afix migrate
```

3. Migrate the ODS schema:

```
./flyway -url=jdbc:postgresql://localhost:5432/afixdb -schemas=ods -
 sqlMigrationPrefix=ODS_V -user=afix migrate
```

4. Migrate the OLAP schema:

```
./flyway -url=jdbc:postgresql://localhost:5432/afixdb -schemas=olap -
 sqlMigrationPrefix=OLAP_V -user=afix migrate
```

Set Up a Daily Job / Drop Partition

1. Copy `partition.sql` from the `/DBB/POSTGRES` directory to `/var/lib/pgsql/9.4/afixdb`:

```
cp partition.sql /var/lib/pgsql/9.4/afixdb
```

2. Add a crontab entry to run the partition script daily at 1:00 a.m. to create future partitions and drop old partitions:

```
[postgres@interop-qa0-dbb ~]$ crontab -e
 #Add below 2 entries and save it (Esc + Shift + : + wq + Enter)
 LOGFILE_AFIXDB=/var/lib/pgsql/9.4/afixdb/partition.log
 0 1 * * * echo "--------------------------------" >> $LOGFILE_AFIXDB ;
 date >> $LOGFILE_AFIXDB ; psql -d afixdb -U postgres -f
 /var/lib/pgsql/9.4/afixdb/partition.sql &&gt;> $LOGFILE_AFIXDB
```

SMaRT AFIX Database Initialization - RDS

The information below describes how to initialize the database for the first time, how to upgrade/migrate an existing database, and how to set up daily jobs.

Prerequisite Tasks

Perform the following tasks before initializing the database:

1. Confirm that PostgreSQL has been installed.
2. Download the release distribution zip file from `/DBB/POSTGRES` and extract the database scripts for the database server.
3. As a root user, give read, write, and execute permissions on the flyway command - Navigate to the `/DBB/POSTGRES` directory and execute this command:

```
cd /DBB/POSTGRES/
 chmod 755 flyway
```

4. As a root user, install the dos2unix package:

```
yum install dos2unix
```

5. As a Postgres user, run the dos2unix command for the flyway file to make it compatible for Linux - Navigate to the `/DBB/POSTGRES` directory and execute this command:

```
cd /DBB/POSTGRES/
 dos2unix flyway
```

Initialize a New Database (for the First Time Only)

If this is the first time you are initializing the database, follow the steps below. Otherwise, see the Upgrade/Migrate an Existing Database instructions.

1. Add postgres to your path:

```
export PATH=$PATH:/usr/pgsql-9.4/bin
```

2. Create the afixdb by navigating to the `/usr/pgsql-9.4/bin` directory and executing this command. Enter the password at the end.

```
cd /usr/pgsql-9.4/bin
 psql -h <rds_hostname> -d <default_db_created_with_RDS> -p <DB_port> -U
 <RDS-superuser> -f /DBB/POSTGRES/setup_database_Rds.sql
```

3. Create the database users and roles:

```
cd /usr/pgsql-9.4/bin
 psql -h <rds_hostname> -d afixdb -p <DB_port> -U <RDS-superuser> -f
 /DBB/POSTGRES/setup_objects.sql
```

4. Create the APP configuration schema:

```
./flyway -url=jdbc:postgresql://<RDS-hostname>:5432/afixdb -schemas=app -
 sqlMigrationPrefix=APP_V -user=afix migrate
```

5. Create the ODS schema:

```
./flyway -url=jdbc:postgresql://<RDS-hostname>:5432/afixdb -schemas=ods -
 sqlMigrationPrefix=ODS_V -user=afix migrate
```

6. Create the OLAP schema:

```
./flyway -url=jdbc:postgresql://<RDS-hostname>:5432/afixdb -schemas=olap
 -sqlMigrationPrefix=OLAP_V -user=afix migrate
```

7. Create the .pgpass file under the home directory (usually root **or** talenduser):

```
vi ~/.pgpass
```

8. Add these contents into the ~/.pgpass file and save it:

```
<RDS_hostname>:<port>:<database>:<username>:<password>
```

9. give the permission

```
chmod 0600 ~/.pgpass
```

Upgrade/Migrate an Existing Database

If this is not the first time you are initializing the database, follow the steps below. Otherwise, see the Initialize a New Database (for the First Time Only) instructions.

1. Act as the postgres user:

```
sudo su - postgres
```

2. Migrate the APP configuration schema:

```
./flyway -url=jdbc:postgresql://<RDS-hostname>t:5432/afixdb -schemas=app
  -sqlMigrationPrefix=APP_V -user=afix migrate
```

3. Migrate the ODS schema:

```
./flyway -url=jdbc:postgresql://<RDS-hostname>:5432/afixdb -schemas=ods -
  sqlMigrationPrefix=ODS_V -user=afix migrate
```

4. Migrate the OLAP schema:

```
 ./flyway -url=jdbc:postgresql://<RDS-hostname>:5432/afixdb -schemas=olap -
sqlMigrationPrefix=OLAP_V -user=afix clean
```

```
./flyway -url=jdbc:postgresql://<RDS-hostname>:5432/afixdb -schemas=olap
  -sqlMigrationPrefix=OLAP_V -user=afix migrate
```

Set Up a Daily Job / Drop Partition

1. Copy `partition.sql` from the `/DBB/POSTGRES` directory to `/opt/partitionDir`:

```
cp partition.sql /opt/partitionDir
  chmod 755 /opt/partitionDir/partition.sql
```

2. Add a crontab entry to run the partition script daily at 1:00 a.m. to create future partitions and drop old partitions:

```
[postgres@interop-qa0-dbb ~]$ crontab -e
  #Add below 2 entries and save it (Esc + Shift + : + wq + Enter)
  LOGFILE_AFIXDB=/opt/partitionDir/partition.log
  0 1 * * * echo "---------------------------------" >> $LOGFILE_AFIXDB ;
  date >> $LOGFILE_AFIXDB ; cd /usr/psql-9.4/bin ; psql -d afixdb -h <RDS-
  hostname> -U <username> -f /opt/partitionDir/partition.sql &>>
  $LOGFILE_AFIXDB
```

SMaRT AFIX Database Migration from EC2 Postgres to RDS

The following instructions explain how to perform a database migration from EC2 Postgres to RDS.

Prerequisite Tasks

Perform the following tasks before initializing the database:

1. Confirm that the PostgreSQL client (Psql) has been installed.

2. Download the release distribution zip file from `/DBB/POSTGRES` and extract the database scripts for the database server.

3. As a root user, give read, write, and execute permissions on the flyway command - Navigate to the `/DBB/POSTGRES` directory and execute this command:

   ```
   cd /DBB/POSTGRES/
    chmod 755 flyway
   ```

4. As a root user, install the dos2unix package:

   ```
   yum install dos2unix
   ```

5. As a Postgres user, run the dos2unix command for the flyway file to make it compatible for Linux - Navigate to the `/DBB/POSTGRES` directory and execute this command:

   ```
   cd /DBB/POSTGRES/
    dos2unix flyway
   ```

Create Dump File from Postgres

Use these instructions to create a dump file from Postgres installed in the EC2 instance:

1. Log in as a Postgres user and create a dump file using this command:

   ```
   su - postgres
    pg_dump dbname=afixdb -f myafixdump.sql
   ```

2. Save this SQL file (`myafixdump.sql`) into the most convenient directory.

Initialize a New Database (for the First Time Only)

Follow these steps to initialize the new database for the first time only:

1. Add postgres to your path:

   ```
   export PATH=$PATH:/usr/pgsql-9.4/bin
   ```

2. Create the afixdb by navigating to the `/usr/pgsql-9.4/bin` directory and executing this command. Enter the password at the end.

   ```
   cd /usr/pgsql-9.4/bin
    psql -h <rds_hostname> -d <default_db_created_with_RDS> -p <DB_port> -U
    <RDS-superuser> -f /DBB/POSTGRES/setup_database_Rds.sql
   ```

3. Create the database users and roles:

   ```
   cd /usr/pgsql-9.4/bin
    psql -u <rds_hostname> -d afixdb -p <DB_port> -u <RDS-superuser> -f
    /DBB/PSTGRES/setup_objects_Rds_Migration_From_Ec2.sql
   ```

4. Restore the dump file created in the previous section into the RDS console using this command:

   ```
   cd /usr/pgsql-9.4/bin
    psql -h <rds_hostname> -p <DB_port> -U <RDS-superuser> -d afixdb -f
    /var/lib/gsql/myafixdump.sql
   ```

5. Create the APP configuration schema:

   ```
   ./flyway -url=jdbc:postgresql://<RDS-hostname>:5432/afixdb -schemas=app -
    sqlMigrationPrefix=APP_V -user=afix migrate
   ```

6. Create the ODS schema:

   ```
   ./flyway -url=jdbc:postgresql://<RDS-hostname>:5432/afixdb -schemas=ods -
    sqlMigrationPrefix=ODS_V -user=afix migrate
   ```

7. Create the OLAP schema:

   ```
   ./flyway -url=jdbc:postgresql://<RDS-hostname>:5432/afixdb -schemas=olap
    -sqlMigrationPrefix=OLAP_V -user=afix migrate
   ```

8. Create the .pgpass file under the home directory (usually root):

   ```
   vi ~/.pgpass
   ```

9. Add these contents into the `.pgpass` file and save it:

   ```
   <RDS_hostname>:<port>:<database>:<username>:<password>
   ```

Set Up a Daily Job / Drop Partition

1. Copy `partition.sql` from the `/DBB/POSTGRES` directory to `/opt/partitionDir`:

   ```
   cp partition.sql /opt/partitionDir
    chmod 755 /opt/partitionDir/partition.sql
   ```

2. Add a crontab entry to run the partition script daily at 1:00 a.m. to create future partitions and drop old partitions:

   ```
   [postgres@interop-qa0-dbb ~]$ crontab -e
    #Add below 2 entries and save it (Esc + Shift + : + wq + Enter)
    LOGFILE_AFIXDB=/opt/partitionDir/partition.log
    0 1 * * * echo "--------------------------------" >> $LOGFILE_AFIXDB ;
    date >> $LOGFILE_AFIXDB ; cd /usr/psql-9.4/bin ; psql -d afixdb -h <RDS-
    hostname> -U <username> -f /opt/partitionDir/partition.sql &>>
    $LOGFILE_AFIXDB
   ```

SMaRT AFIX Online Tool Web Service Installation - Linux

The SMaRT AFIX Online Tool web service is used by the SMaRT AFIX application to generate an XML file to be uploaded to the CDC's AFIX online tool application. The web service is a simple web application that is installed alongside the Stand Alone Forecaster on the Service machine (SVC).

Prerequisites

1. The Postgres database is installed and configured.
2. The Postgres password for the SMaRT AFIX database is known.
3. The Tomcat instance is installed on the SVC server.
4. All necessary ports are open across all firewalls.
5. The release distribution file for the SMaRT AFIX Online Tool web service is extracted from `/SVC/ONLINE_TOOL/afix.export.ws.war` and copied to your local drive.

Installation Steps

1. Copy the `afix.export.ws.war` file to your Tomcat webapps folder ($CATALINA_HOME is your Tomcat installation directory):

   ```
   $ cp afix.export.ws.war $CATALINA_HOME/webapps/afix.export.ws.war
   ```

2. Restart Tomcat.
3. Update the application properties to point to the proper Postgres database. Make sure the username and password are correct.
4. Edit the `$CATALINA_HOME/webapps/afix.export.ws/WEB-INF/app.properties` file:

   ```
   driverClassName=org.postgresql.Driver
    url=jdbc:postgresql://127.0.0.1:5432/afixdb
    username=afixUser
    password=afixPassword
   ```

5. Restart Tomcat.
6. Verify the installation with this command. Note that you may need to adjust the port from 8080, depending on your Tomcat configuration.

   ```
   curl -XGET
    'http://localhost:8080/afix.export.ws/api/v1/sitevisit/?org_id=
    0&fac_id=0&cohort=All' -H "API-KEY: 123456" -H "accept:application/xml"
   ```

7. Verify that you have the following result:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
 <AFIXExport xsi:schemaLocation="http://www.cdc.gov/afix/online-
 tool/1.1/exportSchema CDC_AFIX_Export.xsd"
 xmlns="http://www.cdc.gov/afix/online-tool/1.1/exportSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Header>
       <AwardeeName>No Provider Found</AwardeeName>
    </Header>
 </AFIXExport>
```

Jaspersoft ETL Installation – Linux automation Scripts

Run provision.sh script from the bitbucket link below

https://bitbucket.org/stchome/afix_automation/src/master/ETL_Install_ShellScript/

Jaspersoft ETL Installation - Linux

Follow these instructions to install Jaspersoft ETL, Apache Web Server, and Subversion, and to configure the administration console. This installation only needs to occur once.

Prerequisites

1. Java JDK 8 (1.8.91) has been installed.
2. The environment variables JAVA_HOME, JDK_HOME, and PATH have been set. Example:

```
export JAVA_HOME=/usr/lib/jvm/jdk1.8_91
 export JDK_HOME=/usr/lib/jvm/jdk1.8_91
 export PATH=$JAVA_HOME/bin:$PATH
 export PATH=$JDK_HOME/bin:$PATH
```

3. Copy the components from the distribution `/Server Components/ETL` directory into your local directory. These files include:
   - license.txt
   - TIBCOJaspersoft-JETL-Installer-20150908...zip
   - etl.zip

Install Jaspersoft ETL

1. Download the Jaspersoft ETL installer zip file:

```
Server Components/ETL/TIBCOJaspersoft-JETL-Installer-20150908_1633-
 V7.1.1-installer.zip
```

2. Place the installer into the installation directory:

   ```
   /opt/TIBCOJaspersoft-JETL-Installer-20150908_1633-V7.1.1-installer.zip
   ```

3. Navigate to the installation directory:

   ```
   $ cd /opt
   ```

4. Install the Unzip utility if it is not already present:

   ```
   $ sudo yum install zip unzip -y
   ```

5. Unzip the installer:

   ```
   $ sudo unzip TIBCOJaspersoft-JETL-Installer-20150908_1633-V-installer.zip
   ```

6. Set the execute permission on the installer:

   ```
   $ sudo chmod +x TIBCOJaspersoft-JETL-Installer-20150908_1633-V-linux64-
    installer.run
   ```

7. Execute the installer:

   ```
   $ sudo ./TIBCOJaspersoft-JETL-Installer-20150908_1633-V7.1.1-linux64-
    installer.run
   ```

8. Follow the prompts to accept the license agreement.
9. Follow the prompts to specify the installation folder. Specify the `/opt/TIBCOJaspersoft` directory.
10. Follow the prompts to select the install style of item 1 (Easy).
11. Follow the prompts to select the install type of item 1 (Server).
12. Follow the prompts to select the license file (provide the location of the license file).
13. Wait until the installation completes.

Start/Stop Jaspersoft ETL Services

The following are the instructions to start and/or stop Jaspersoft ETL services. These steps are not mandatory on a first-time installation because the services will be running by default.

Start/Stop Jaspersoft ETL Admin Console

Command to stop the service:

```
$ cd /opt/TIBCOJaspersoft-/jac
```

```
 $ sudo ./stop_tac.sh
```

Command to start the service:

```
$ cd /opt/TIBCOJaspersoft-6.0.1/jac
 $ sudo ./start_tac.sh
```

Start/Stop Jaspersoft ETL Command Line

Command to stop the service:

```
$ cd /opt/TIBCOJaspersoft/cmdline
 $ sudo ./stop_cmdline.sh
```

Command to start the service:

```
$ cd /opt/TIBCOJaspersoft/cmdline
 $ sudo ./start_cmdline.sh
```

Start/Stop Jaspersoft ETL Job Server

Command to stop the service:

```
$ cd /opt/TIBCOJaspersoft/jobserver
 $ sudo ./stop_jobserver.sh
```

Command to start the service:

```
$ cd /opt/TIBCOJaspersoft/jobserver
 $ sudo ./start_jobserver.sh
```

Auto Start/Stop Jaspersoft ETL Services

Follow these instructions to automatically start and/or stop Jaspersoft ETL services.
Note: These steps are not mandatory for a first-time installation because the services run by default.

Create New User

Use this command to create a new user:

```
$ sudo adduser talenduser
```

Assign ownership of `/etl` and `/opt/TIBCOJaspersoft` to the new user created above:

```
$ sudo chown -R talenduser:talenduser /etl
 $ sudo chown -R talenduser:talenduser /opt/TIBCOJaspersoft
```

Edit Startup Scripts

Follow these steps to edit the startup scripts.

1. Copy these scripts from the package folder `DBB/ETL/Autostart` into a convenient directory:

    - `jaspersoft-cmdline`

    - `jaspersoft-jac`
    - `jaspersoft-rjs`

2. Copy the above files into the `/etc/init.d` directory.
3. Edit all three scripts and modify the environment variables as shown here:

```
* export JAVA_HOME=[Path for JAVA_HOME directory. This should be Oracle
 JDK.]
 * RUN_AS_USER=talenduser [If you are running as a root user, comment out
 this statement.]
```

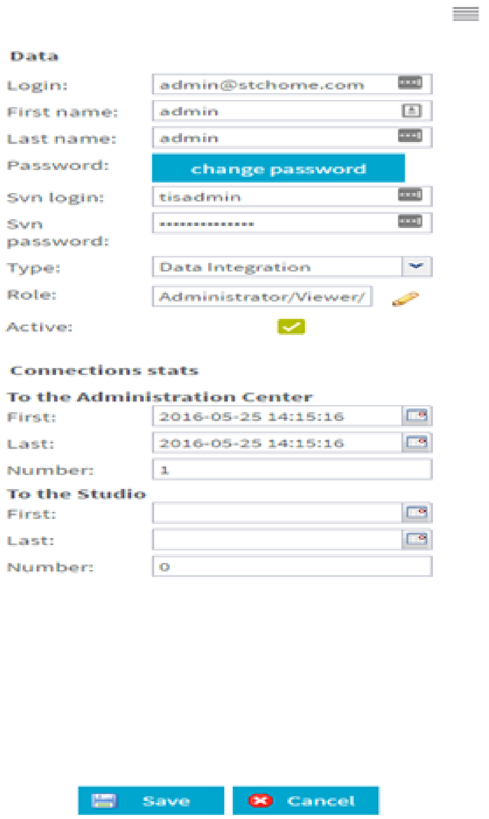
4. Edit the `chkconfig` setting to start the script on system reboot:

```
* chkconfig --level 235 jaspersoft-jac on
 * chkconfig --level 235 jaspersoft-rjs on
 * chkconfig --level 235 jaspersoft-cmdline on
```

Configure the Admin Console

These steps are used one time only. They should be completed during the server installation.

1. Navigate to `http://[host name]:8080/org.jasper.administrator` and enter the default username and password (admin). For the username, use `admin@yourdomain.com` or replace it with an email group if the user needs email notifications on ETL job failure (example: `email_group@yourdomain.com`).
2. Navigate to **User Settings** on the left menu and update the username (if needed) and library information.



Data

Login:	admin@stchome.com
First name:	admin
Last name:	admin
Password:	change password
Svn login:	tisadmin
Svn password:	•••••••••••••••
Type:	Data Integration
Role:	Administrator/Viewer/
Active:	☑

Connections stats

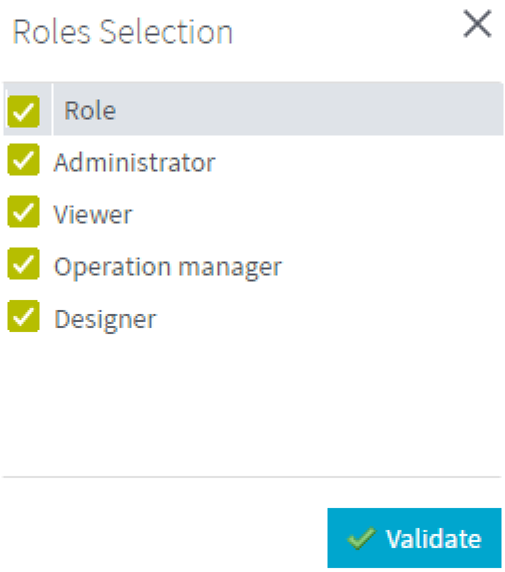
To the Administration Center

First:	2016-05-25 14:15:16
Last:	2016-05-25 14:15:16
Number:	1

To the Studio

First:	
Last:	
Number:	0

3. Click on **Role** and select all of the roles.



Roles Selection

☑ Role

☑ Administrator

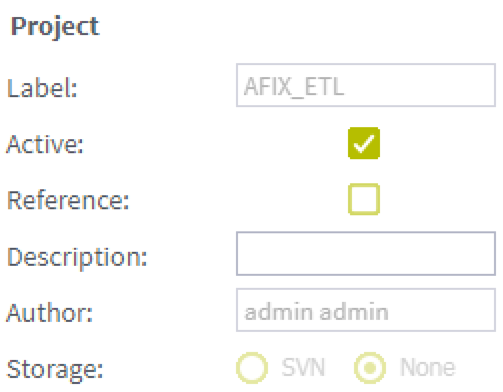
☑ Viewer

☑ Operation manager

☑ Designer

4. Navigate to **Project settings** from the left menu and add a new project by clicking the **Add** button.

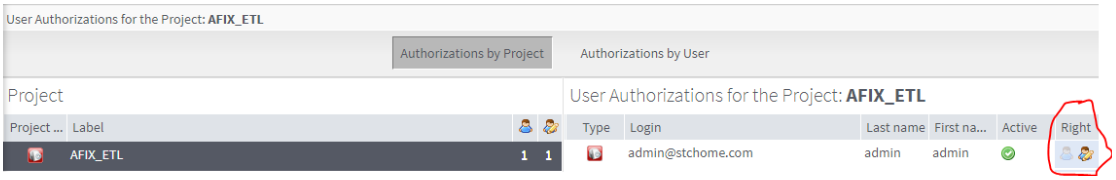
5. Name the project as **AFIX_ETL**, select the **Active** option, enter **AFIX_ETL** as the description and **admin admin** as the Author, select **None** as the storage, and save it.



Project

Label:	AFIX_ETL
Active:	☑
Reference:	□
Description:	
Author:	admin admin
Storage:	○ SVN ● None

6. Navigate to **Project authorization** from the left menu. Click on each of the newly created projects and assign read/write permissions to the user.



User Authorizations for the Project: **AFIX_ETL**

Project

Project ...	Label		
	AFIX_ETL	1	1

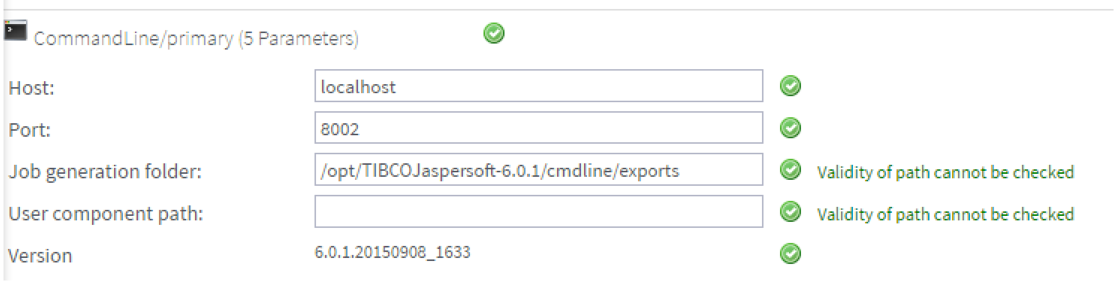
User Authorizations for the Project: **AFIX_ETL**

Type	Login	Last name	First na...	Active	Right
	admin@stchome.com	admin	admin	✓	

7. Create the following directories and files in the server:

```
sudo mkdir /opt/TIBCOJaspersoft/deploy
  sudo mkdir /opt/TIBCOJaspersoft/deploy/generatedJobs
  sudo mkdir /opt/TIBCOJaspersoft/deploy/executionLogs
  sudo mkdir /opt/TIBCOJaspersoft/cmdline/exports
  sudo mkdir /opt/TIBCOJaspersoft/Audit
  sudo mkdir /opt/TIBCOJaspersoft/Audit/reports
  sudo mkdir /opt/TIBCOJaspersoft/joblogs
  sudo touch /opt/TIBCOJaspersoft/joblogs/log
```

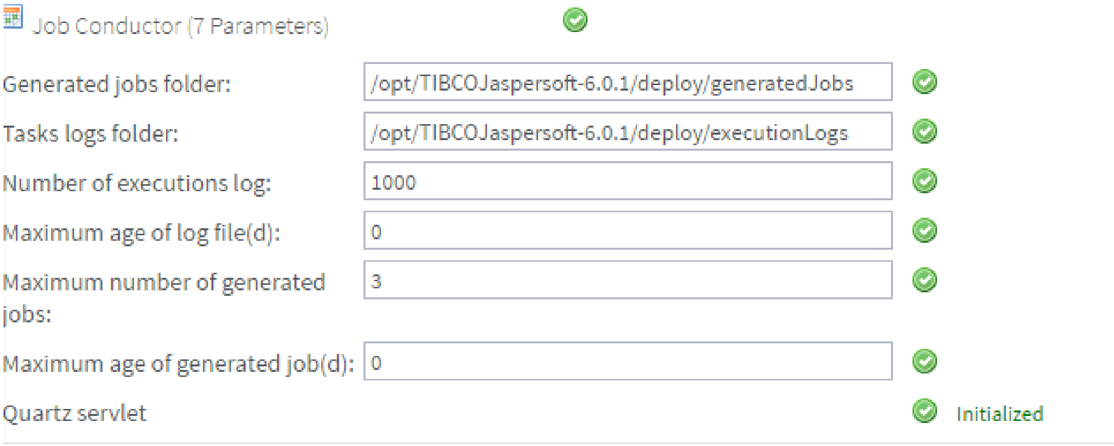
8. In Primary Command Line configuration, set the path for the Job generation folder to `/opt/TIBCOJaspersoft/cmdline/exports`



CommandLine/primary (5 Parameters) ✓

Host:	localhost	✓	
Port:	8002	✓	
Job generation folder:	/opt/TIBCOJaspersoft-6.0.1/cmdline/exports	✓	Validity of path cannot be checked
User component path:		✓	Validity of path cannot be checked
Version	6.0.1.20150908_1633	✓	

9. In Job Conductor configuration, set the path for the Generated jobs folder to `/opt/TIBCOJaspersoft/deploy/generatedJobs` and the path to the Tasks logs folder to `/opt/TIBCOJaspersof/deploy/executionLogs`



Job Conductor (7 Parameters) ✓

Generated jobs folder: /opt/TIBCOJaspersoft-6.0.1/deploy/generatedJobs ✓

Tasks logs folder: /opt/TIBCOJaspersoft-6.0.1/deploy/executionLogs ✓

Number of executions log: 1000 ✓

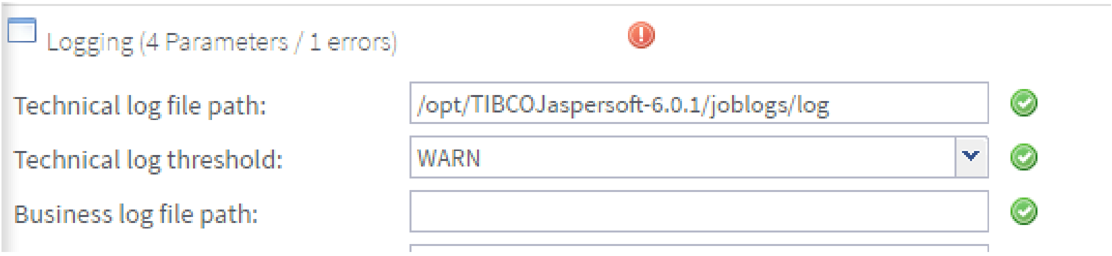
Maximum age of log file(d): 0 ✓

Maximum number of generated jobs: 3 ✓

Maximum age of generated job(d): 0 ✓

Quartz servlet ✓ Initialized

10. In Logging configuration, set the Technical log file path to `/opt/TIBCOJaspersoft/joblogs/log`



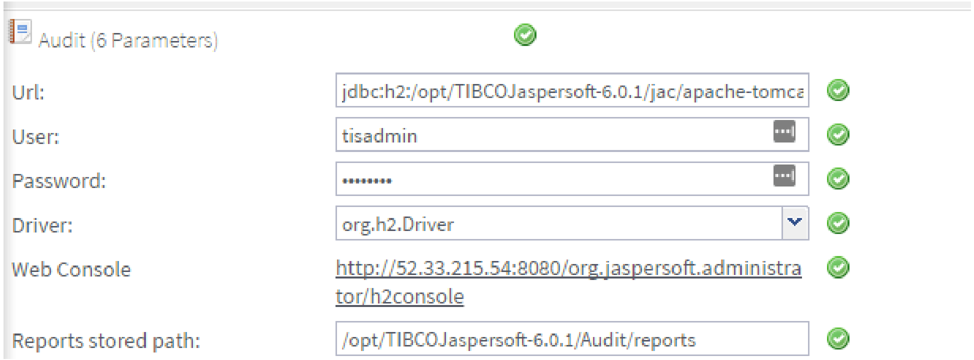
Logging (4 Parameters / 1 errors) !

Technical log file path: /opt/TIBCOJaspersoft-6.0.1/joblogs/log ✓

Technical log threshold: WARN ✓

Business log file path: ✓

11. In Audit configuration, set the Reports stored path to `/opt/TIBCOJaspersoft/Audit/reports`



Audit (6 Parameters) ✓

Url: jdbc:h2:/opt/TIBCOJaspersoft-6.0.1/jac/apache-tomca ✓

User: tisadmin ✓

Password: •••••••• ✓

Driver: org.h2.Driver ✓

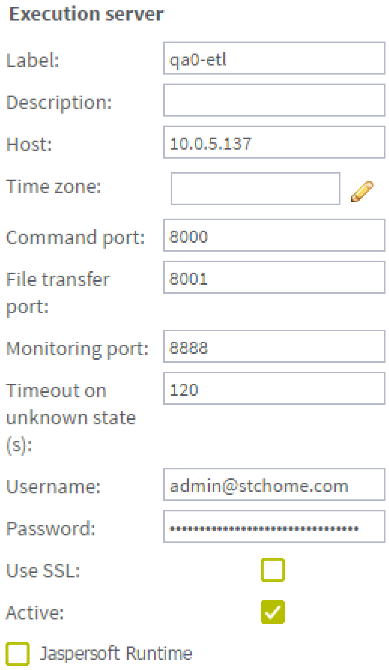
Web Console http://52.33.215.54:8080/org.jaspersoft.administrator/h2console ✓

Reports stored path: /opt/TIBCOJaspersoft-6.0.1/Audit/reports ✓

12. Navigate to **Servers page** from the menu and create a new execution sever with the following settings:

- Label: Name of the job server
- Host: ETL Server IP (it usually ends with *dbb*)
- Command port: 8000
- File transfer port: 8001
- Monitoring port: 8888

- Active: Select this option (enable)
- Username: the admin username for the ETL
- Password: the password for the admin ETL user



Execution server

Label:	qa0-etl
Description:	
Host:	10.0.5.137
Time zone:	
Command port:	8000
File transfer port:	8001
Monitoring port:	8888
Timeout on unknown state (s):	120
Username:	admin@stchome.com
Password:	••••••••••••••••••••••••••••••••
Use SSL:	□
Active:	☑

□ Jaspersoft Runtime

2. Log in via SSH to the Unix server using the ETL server owner.
3. Copy the `etl.zip` file (from `/DBB/ETL`) to a location convenient for you on the ETL Jaspersoft server to unzip the file, such as a home directory.
4. Unzip the `etl.zip` file to the server in the root directory (Unix), which creates a directory structure similar to this:
   - /etl
   - /etl/data
   - /etl/archive

SMaRT AFIX Jaspersoft ETL Deployment - Linux

The following outlines how to import (migrate) Jaspersoft ETL for SMaRT AFIX.

Pre-Migration Checklist

- Confirm that the Postgres database is installed and configured
- Retrieve the passwords for the SMaRT AFIX database (*afixdb*)

- Retrieve the Jaspersoft ETL admin username and password
- Confirm that port 8080 is open across all firewalls
- Confirm that all of the steps in the Admin Console configuration are completed on a first-time installation
- Copy these components from `/DBB/ETL` into your local directory:
  - API_Vaildation_Sequential_Execution_0.1
  - etl
  - Forecaster_Sequential_Execution_0.1
  - ODS_Sequential_Execution_0.1
  - OLAP_Sequential_Execution_0.1
  - AFIX_Assessment_Benchmark_0.1
  - LoadVFCEligibleStatus_OnDemand_0.1
  - IWeb_User_API_Sequential_Execution_0.1
- Identify whether the installation is an upgrade from prior versions or a fresh installation from scratch.

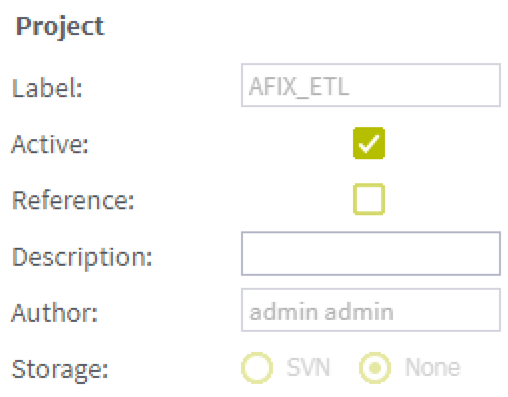
New Installation Instructions (for the First Time Only)

If this is the first time you deploying Jaspersoft ETL, follow the steps below. Otherwise, see the Upgrade Instructions.

The steps below only need to be performed once. They should be completed during the server installation.

Configure the Admin Console

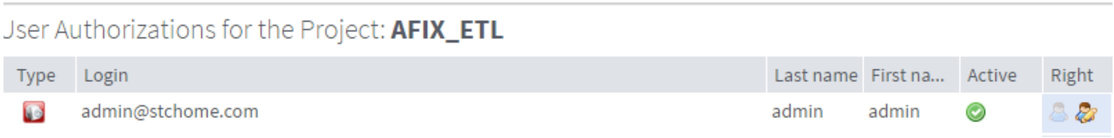
1. Navigate to `http://[host name]:8080/org.jaspersoft.administrator` and enter the default username and password (admin). For the username, use `admin@yourdomain.com` or replace it with an email group if the user needs email notifications on ETL job failure (example: `email_group@yourdomain.com`).
2. Navigate to **Project settings** from the left menu and add a new project by clicking the **Add** button.
3. Name the new project **AFIX_ETL**. Select the **Active** option, enter **admin admin** as the Author, and select **SVN** as the Storage.



Project

Label:	AFIX_ETL
Active:	☑
Reference:	□
Description:	
Author:	admin admin
Storage:	○ SVN ● None

4. Navigate to **Project authorization** from the left menu. Click on the newly created project and assign read/write permission to the user.

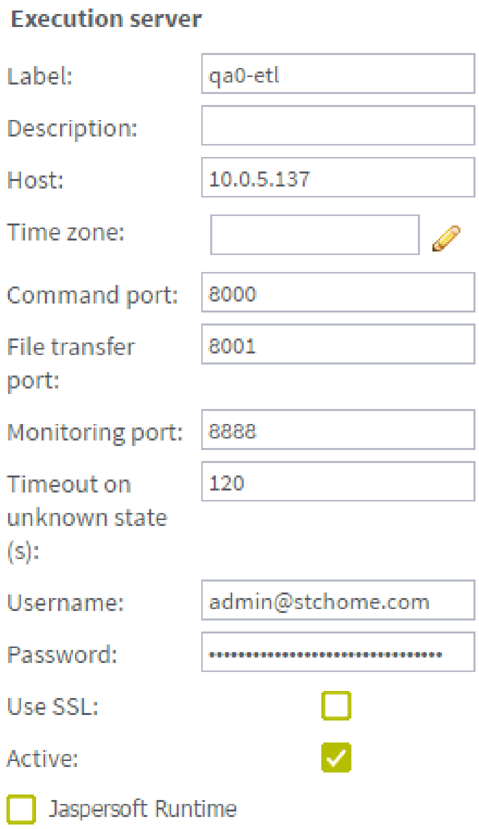


Jser Authorizations for the Project: **AFIX_ETL**

Type	Login	Last name	First na...	Active	Right
	admin@stchome.com	admin	admin	✓	

5. Navigate to **Servers page** from the menu and create a new execution server with the following settings:

   - Label: AFIX_jobServer
   - Host: ETL Server IP (it usually ends with *dbb*)
   - Command port: 8000
   - File transfer port: 8001
   - Monitoring port: 8888
   - Active: Select this option (enable)
   - Username: the admin username for the ETL
   - Password: the password for the admin ETL user



Execution server

Label:	qa0-etl
Description:	
Host:	10.0.5.137
Time zone:	
Command port:	8000
File transfer port:	8001
Monitoring port:	8888
Timeout on unknown state (s):	120
Username:	admin@stchome.com
Password:	••••••••••••••••••••••••••••••••
Use SSL:	□
Active:	☑

□ Jaspersoft Runtime

2. Log in via SSH to the Unix server using the ETL server owner.
3. Copy the `etl.zip` file (from `/DBB/ETL`) to a location convenient for you on the ETL Jaspersoft server to unzip the file, such as a home directory.
4. Unzip the `etl.zip` file to the server in the root directory (Unix), which creates a directory structure similar to this:
   - /etl/afix
   - /etl/afix/api
   - /etl/afix/archive
   - /etl/afix/bulkload
   - /etl/afix/dataout
   - /etl/afix/input
   - /etl/afix/rejects
   - /etl/afix/tmp
   - /etl/afix/validation

Deploy the ETL Components

1. If a previous ETL Projects already exists, perform a backup of it.

2. Copy the following files from the installation files to a convenient location to deploy:

    - IWEB_Staging_Sequential_Execution.zip
    - API_Validation_Sequential_Execution.zip
    - ODS_Sequential_Execution.zip
    - Forecaster_Sequential_Execution_0.1.zip
    - OLAP_Sequential_Execution.zip
    - AFIX_Assesment_Benchmark.zip
    - LoadVFCEligibleStatus_OnDemand_0.1
    - IWeb_User_API_Sequential_Execution_0.1

2. Navigate to the Job Conductor page.

3. Select **Add > Normal Task** and add the following: Label = AFIX IWEB API Execution; Description = AFIX IWEB API Execution.

4. Click **Import Generated Jobs** and select the `IWEB_Staging_Sequential_Execution.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.

5. Click **Save**.

6. Select **Add > Normal Task** and add the following: Label = AFIX IWEB API Validation; Description = AFIX IWEB API Validation.

7. Click **Import Generated Jobs** and select the `API_Validation_Sequential_Execution.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.

8. Click **Save**.

9. Select **Add > Normal Task** and add the following: Label = AFIX Stage 2 ODS Load; Description = AFIX Stage 2 ODS Load.

10. Click **Import Generated Jobs** and select the `ODS_Sequential_Execution.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.

11. Click **Save**.

12. Select **Add > Normal Task** and add the following: Label = AFIX Forecast Execution; Description = AFIX Forecast Execution.

13. Click **Import Generated Jobs** and select the `Forecaster_Sequential_Execution.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.

14. Click **Save**.

15. Select **Add > Normal Task** and add the following: Label = AFIX OLAP Execution; Description = AFIX OLAP Execution.

16. Click **Import Generated Jobs** and select the `OLAP_Sequential_Execution.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.

17. Click **Save**.

18. Select **Add > Normal Task** and add the following: Label = AFIX Assessment Benchmark; Description = AFIX Assessment Benchmark.

19. Click **Import Generated Jobs** and select the `AFIX_Assessment_Benchmark.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.

20. Click **Save**.

21. Select **Add > Normal Task** and add the following: Label = Load VFC_Eligible; Description = Load VFC Eligible.

22. Click **Import Generated Jobs** and select the LoadVFCEligibleStatus_Ondemand.zip file downloaded as part of the prerequisites. Add the following: Execution Server = AFIX_JobServer; On Unavailable JobServer = Wait.

23. Click **Save**.

24. Select **Add > Normal Task** and add the following: Label = IWeb User API; Description = IWeb User API.

25. Click **Import Generated Jobs** and select the IWeb_User_API_Sequential_Execution_0.1 file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable Job Server = Wait.

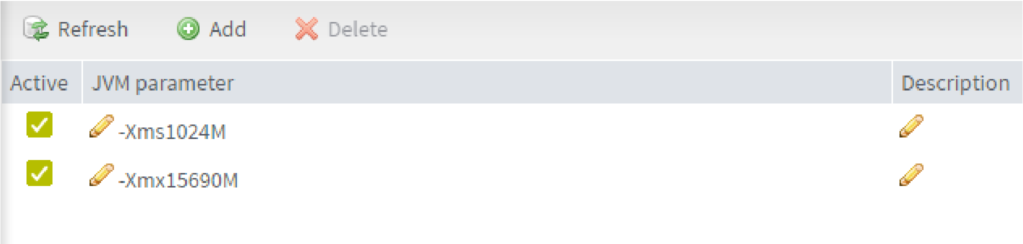
26. Click **Save**.

27. For each job created above, update the following Contexts within the Context tab located at the bottom of the page. Select custom values as needed. If the client is configured to receive email notifications about failures/successes, then Email_active should be True; otherwise it should be false. For Email_Active=True, all of the parameters in the Email section should be keyed in with proper values.

    - Email_To
    - Email_SMTP_Host
    - Email_SMTP_Port
    - AFIX_Email_Notification
    - Email_Active
    - Email_UserName

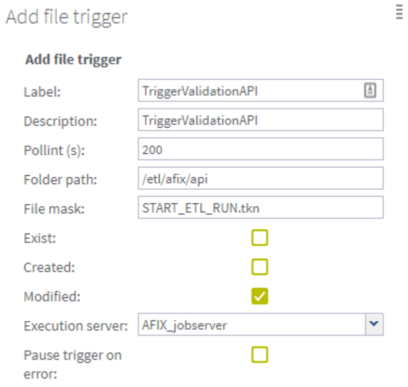
- Email_Password
- Web service related parameters (Forecaster):
  - SOAP_ACTION
  - ENDPOINT
- Database-related parameters:
  - AFIX_DB - Login, Password, Port, Database, and Server
  - Source_DB_Context - Login, Password, Port, Database, and Server
- RDS-related parameters for ODS_sequential_job
  - Psql_Path - Path of the psql (by default, it is `/usr/pgsql-9.4/bin`)
  - Rds_Postgres - A boolean parameter; set it to True if the database is RDS
  - Windows_Execution_Environment - A boolean parameter; set it to False; the ID of the ETL execution environment is UNIX.

2. Save when finished.
3. Initial load instructions:
   - Run the **AssessmentBenchmark** task
   - Go to the task **AFIX IWeb API Execution**. Add the two JVM parameters: -Xms1024M and -Xmx15690M. Run the AFIX IWeb API Execution task and wait to see if it executes without issues. After it starts to make progress, continue with the next item



Active	JVM parameter	Description
☑	-Xms1024M	
☑	-Xmx15690M	

   - Go to the task **IWeb API Validation** and add a file trigger with the following values: Label = TriggerValidationAPI, Folder Path = /etl/afix/dataout, File Mask = API_EXTRACT_Complete.tkn, Pollint(s) = 200

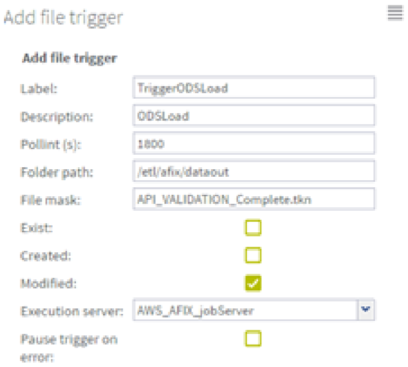


Add file trigger

Add file trigger

Label:	TriggerValidationAPI
Description:	TriggerValidationAPI
Pollint (s):	200
Folder path:	/etl/afix/api
File mask:	START_ETL_RUN.tkn
Exist:	□
Created:	□
Modified:	☑
Execution server:	AFIX_jobserver
Pause trigger on error:	□

- Go to the task **AFIX Stage 2 ODS Load** and add a file trigger with these values: Label = TriggerODSLoad, Folder Path = /etl/afix/dataout, File Mask = API_VALIDATION_Complete.tkn, Pollint(s) = 200

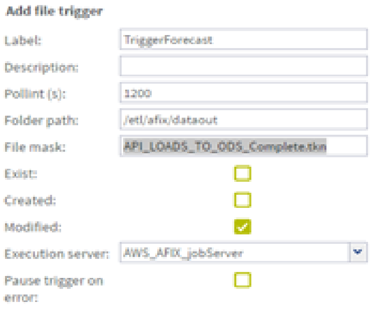


Add file trigger

Add file trigger

Label:	TriggerODSLoad
Description:	ODSLoad
Pollint (s):	1800
Folder path:	/etl/afix/dataout
File mask:	API_VALIDATION_Complete.tkn
Exist:	□
Created:	□
Modified:	☑
Execution server:	AWS_AFIX_jobServer
Pause trigger on error:	□

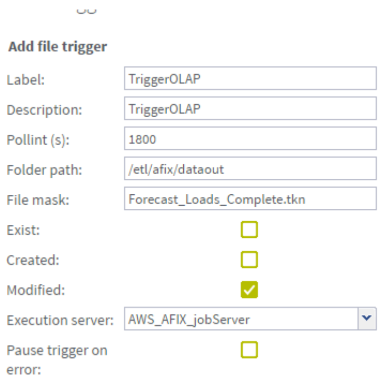
- Go to the task **AFIX Forecast Execution** and add a file trigger with the following values: Label = TriggerForecast, Folder Path = /etl/afix/dataout, File Mask = API_LOADS_TO_ODS_Complete.tkn, Pollint(s) = 200. Add the JVM parameters -Xms1024M and -Xmx15690M



Add file trigger

Label:	TriggerForecast
Description:	
Pollint (s):	1200
Folder path:	/etl/afix/dataout
File mask:	API_LOADS_TO_ODS_Complete.tkn
Exist:	□
Created:	□
Modified:	☑
Execution server:	AWS_AFIX_jobServer
Pause trigger on error:	□

- Go to the task **AFIX OLAP Execution** and add a file trigger with the following values: Label = TriggerOLAP, Folder Path = /etl/afix/dataout, File Mask = Forecast_Loads_Complete.tkn, Pollint(s) = 200. Add the JVM parameters -Xms1024M and Xmx15690M

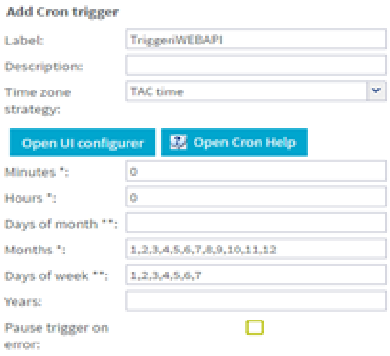


Add file trigger

Label:	TriggerOLAP
Description:	TriggerOLAP
Pollint (s):	1800
Folder path:	/etl/afix/dataout
File mask:	Forecast_Loads_Complete.tkn
Exist:	☐
Created:	☐
Modified:	☑
Execution server:	AWS_AFIX_jobServer
Pause trigger on error:	☐

4. Once the initial loads have completed, the ODS and OLAP tables are populated. Now the daily sequence can be scheduled. Go to the AFIX IWeb API Execution task and add the Cron trigger *Trigger_IWeb_API*. Enter the following values and save:

    - Timezone Strategy: TAC Time
    - Minutes: 0
    - Hours: 0
    - Days of Month: Leave blank
    - Months: 1,2,3,4,5,6,7,8,9,10,11,12
    - Days of the Week: 1,2,3,4,5,6,7
    - Years: leave blank



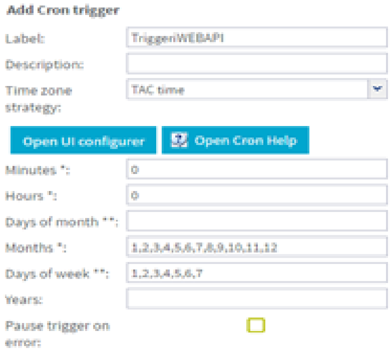
Add Cron trigger

Label:	TriggenWEBAPI
Description:	
Time zone strategy:	TAC time
Minutes *:	0
Hours *:	0
Days of month **:	
Months *:	1,2,3,4,5,6,7,8,9,10,11,12
Days of week **:	1,2,3,4,5,6,7
Years:	
Pause trigger on error:	☐

5. Go to the IWeb User API Execution task and add the Cron trigger *Trigger_IWeb_User_API*. Enter the following values and save:

    - Timezone Strategy: TAC Time
    - Minutes: 0
    - Hours: 0
    - Days of Month: Leave blank
    - Months: 1,2,3,4,5,6,7,8,9,10,11,12

- Days of the Week: 1,2,3,4,5,6,7
- Years: leave blank



Add Cron trigger

Label:	TriggenWEBAPI
Description:	
Time zone strategy:	TAC time
Minutes *:	0
Hours *:	0
Days of month **:	
Months *:	1,2,3,4,5,6,7,8,9,10,11,12
Days of week **:	1,2,3,4,5,6,7
Years:	
Pause trigger on error:	□

6. Turn on the notify/email feature. **Important Note:** SMTP needs to be configured beforehand.
   - Select **Notifications** from the left menu
   - Click **Add**
   - Select the **Tasks** category
   - Select **On task failed**
   - Select **Enabled**
   - Select **Recipients** to specify the user's email. It is better to create an email group as a recipient so that notifications are sent to multiple people
   - Select **AFIX ETL Workflow**

If the upgrade needs to be aborted for any reason, import the previously created backup. Alternatively, redeploy the previous version of the content.

Upgrade Instructions

If this is not the first time you are deploying Jaspersoft ETL (i.e., it was deployed earlier), follow the steps below. Otherwise, see the New Installation Instructions.

The steps below only need to be performed once. They should be completed during the server installation.

Configure the Admin Console

1. Navigate to `http://[host name]:8080/org.jaspersoft.administrator` and enter the default username and password (admin). For the username, use `admin@yourdomain.com` or replace it with an email group if the user needs email notifications on ETL job failure (example: `email_group@yourdomain.com`).
2. Log in via SSH to the Unix server using the ETL server owner.

3. Copy the `etl.zip` file (located at `/DBB/ETL`) to the Jaspersoft ETL server. Use a location convenient for you to unzip the file, such as a home directory.
4. Unzip the `etl.zip` file to the server in the Unix root directory (Unix), which adds the following files that were created as part of this release:
    - /etl/afix
    - /etl/afix/api
    - /etl/afix/archive
    - /etl/afix/bulkload
    - /etl/afix/dataout
    - /etl/afix/input
    - /etl/afix/rejects
    - /etl/afix/tmp
    - /etl/afix/validation

Deploy the ETL Components

1. If a previous ETL Projects already exists, perform a backup of it.
2. Copy the following files from the installation files to a convenient location to deploy:
    - IWEB_Staging_Sequential_Execution.zip
    - API_Validation_Sequential_Execution.zip
    - ODS_Sequential_Execution.zip
    - Forecaster_Sequential_Execution_0.1.zip
    - OLAP_Sequential_Execution.zip
    - AFIX_Assesment_Benchmark.zip
    - LoadVFCEligibleStatus_OnDemand_0.1
    - IWeb_User_API_Sequential_Execution_0.1

2. Navigate to the Job Conductor page.
3. Navigate to the task **AFIX Assessment Benchmark**.
4. Click **Import Generated Jobs** and select the `AFIX_Assessment_Benchmark.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.
5. Click **Save**.
6. Navigate to the task **AFIX IWeb API Execution**.

7. Click **Import Generated Jobs** and select the `IWEB_Staging_Sequential_Execution.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.
8. Click **Save**.
9. Navigate to the task **AFIX IWeb API Validation**.
10. Click **Import Generated Jobs** and select the `API_Validation_Sequential_Execution.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.
11. Click **Save**.
12. Navigate to the task **AFIX Stage 2 ODS Load**.
13. Click **Import Generated Jobs** and select the `ODS_Sequential_Execution.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.
14. Click **Save**.
15. Navigate to the task **AFIX Forecast Execution**.
16. Click **Import Generated Jobs** and select the `Forecaster_Sequential_Execution.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.
17. Click **Save**.
18. Navigate to the task **AFIX OLAP Execution**.
19. Click **Import Generated Jobs** and select the `OLAP_Sequential_Execution.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.
20. Click **Save**.
21. Select **Add > Normal Task** and add the following; Label = Load VFC Eligible; Description = Load VFC Eligible.
22. Click **Import Generated Jobs** and select the `LoadVFCEligibleStatus_Ondemand_0.1.zip` file downloaded as part of the prerequisites. Add the following: Execution server = AFIX_JobServer; On Unavailable JobServer = Recover Task.
23. Select **Add > Normal Task** and add the following: Label = IWeb User API; Description = IWeb User API.
24. Click **Import Generated Jobs** and select the `IWeb_User_API_Sequential_Execution_0.1` file downloaded as part of the prerequisites. Add the following: Execution Server = AFIX_JobServer; On Unavailable JobServer = Wait.

25. For each job created above, update the following Contexts within the Context tab located at the bottom of the page. Select custom values as needed. If the client is configured to receive email notifications about failures/successes, then Email_active should be True; otherwise it should be false. For Email_Active=True, all of the parameters in the Email section should be keyed in with proper values.

    - Email_To
    - Email_SMTP_Host
    - Email_SMTP_Port
    - AFIX_Email_Notification
    - Email_Active
    - Email_UserName
    - Email_Password
    - Custom Date: Set to the current date +1. For example, if the current date is 2017-03-06, the context value should be set to 2017-03-07.
    - RDS-related parameters for ODS_sequential_job

        - Psql_Path - Path of the psql (by default, it is `/usr/pgsql-9.4/bin`)
        - Rds_Postgres - A boolean parameter; set it to True if the database is RDS
        - Windows_Execution_Environment - A boolean parameter; set it to False; the ID of the ETL execution environment is UNIX.

2. Save when finished.

3. Initial load instructions: Go to the task AFIX IWeb API Execution. Run the task and wait to see if the job executes without any issues. Once it has made progress, make sure each of these tasks is triggered and completes:

    - IWeb API Validation
    - State 2 ODS Load
    - Forecast Execution
    - OLAP Execution

4. Once the initial loads have completed, the ODS and OLAP tables are populated. Now the daily sequence can be scheduled. Resume the CRON trigger *Trigger_IWeb_API*. Go to the task OLAP_Sequential_Execution and change the context parameter *custom date* to the original value in OLAP_Sequential_Execution and save.

5. Go to the IWeb User API Execution task and add the Cron trigger *Trigger_IWeb_User_API*. Enter the following values and save:

    - Timezone Strategy: TAC Time
    - Minutes: 0

- Hours: 0
- Days of Month: Leave blank
- Months: 1,2,3,4,5,6,7,8,9,10,11,12
- Days of the Week: 1,2,3,4,5,6,7
- Years: leave blank

If the upgrade needs to be aborted for any reason, import the previously created backup. Alternatively, redeploy the previous version of the content.

JasperReports Server Installation - Linux

The following instructions explain how to install the JasperReports Server.

Prerequisites

Download all of the JasperReports Server contents from the distribution site. The JRS folder contains the following files:

- applicationContext-externalAuth-oAuth
- applicationContext-security
- applicationContext-security-web
- InstallCert$SavingTrustManager.class
- InstallCert.class
- jasperreports-server-6.2.1-linux-x64-installer.exe
- jasperserver.license
- LatoFont
- stc-jaspersoft-oauth-0.0.1-SNAPSHOT

Installation Steps

1. Download the JasperReports Server installer from the release distribution:

   ```
   /Server Components/JRS/jasperreports-server-6.2.1-linux-x64-installer.run
   ```

2. Place the JasperReports Server installer into the target directory:

   ```
   /opt/jasperreports-server-6.2.1-linux-x64-installer.run
   ```

3. Navigate to the target directory:

   ```
   $ cd /opt
   ```

4. Set the execute permissions for the JasperReports Server installer:

   ```
   $ sudo chmod u+x ./jasperreports-server-6.2.1-linux-x64-installer.run
   ```

5. Run the installer:

   ```
   $ sudo ./jasperreports-server-6.2.1-linux-x64-installer.run
   ```

6. Follow the prompts to accept the license agreement.
7. Follow the prompts to select an install option. Select option 1 - **Install All Components and Samples**.

8. Follow the prompts to specify the installation folder. Specify the following directory:

```
/opt/jasperreports-server-6.2.1
```

9. When the installer completes, start the JasperReports Server (see Start/Stop the JasperReports Server below).

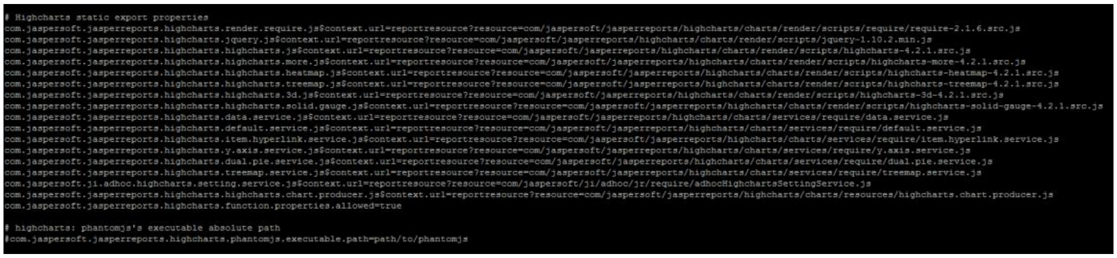
10. Before continuing, you need to update a configuration file on the JasperReports Server. Navigate to:

```
cd /opt/jasperreports-server-6.2.1/apache-tomcat/webapps/jasperserver-
  pro/WEB-INF/classes/
```

11. Open `jasperreports.properties` for editing and add the following under the section marked *# Highcharts static export properties*:

```
com.jaspersoft.jasperreports.highcharts.function.properties.allowed=true
```

It should look something like this:



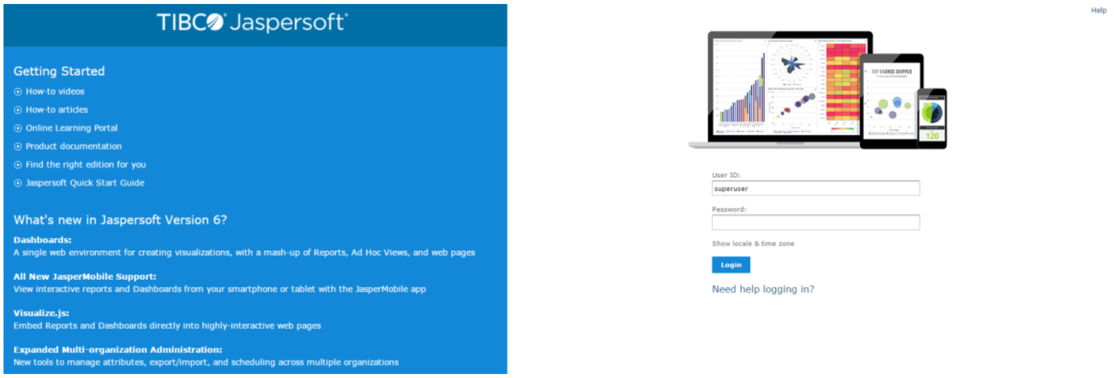
```
# Highcharts static export properties
com.jaspersoft.jasperreports.highcharts.render.require.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/require/require-2.1.6.src.js
com.jaspersoft.jasperreports.highcharts.jquery.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/jquery-1.10.2.min.js
com.jaspersoft.jasperreports.highcharts.highcharts.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/highcharts-4.2.1.src.js
com.jaspersoft.jasperreports.highcharts.highcharts.more.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/highcharts-more-4.2.1.src.js
com.jaspersoft.jasperreports.highcharts.highcharts.heatmap.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/highcharts-heatmap-4.2.1.src.js
com.jaspersoft.jasperreports.highcharts.highcharts.treemap.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/highcharts-treemap-4.2.1.src.js
com.jaspersoft.jasperreports.highcharts.highcharts.3d.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/highcharts-3d-4.2.1.src.js
com.jaspersoft.jasperreports.highcharts.highcharts.solid.gauge.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/highcharts-solid-gauge-4.2.1.src.js
com.jaspersoft.jasperreports.highcharts.data.service.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/services/require/data.service.js
com.jaspersoft.jasperreports.highcharts.default.service.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/services/require/default.service.js
com.jaspersoft.jasperreports.highcharts.item.hyperlink.service.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/services/require/item.hyperlink.service.js
com.jaspersoft.jasperreports.highcharts.y.axis.service.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/services/require/y.axis.service.js
com.jaspersoft.jasperreports.highcharts.dual.pie.service.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/services/require/dual.pie.service.js
com.jaspersoft.jasperreports.highcharts.treemap.service.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/services/require/treemap.service.js
com.jaspersoft.ji.adhoc.highcharts.setting.service.js$context.url=reportresource?resource=com/jaspersoft/ji/adhoc/jr/require/adhocHighchartsSettingService.js
com.jaspersoft.jasperreports.highcharts.highcharts.chart.producer.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/resources/highcharts.chart.producer.js
com.jaspersoft.jasperreports.highcharts.function.properties.allowed=true

# highcharts: phantomjs's executable absolute path
#com.jaspersoft.jasperreports.highcharts.phantomjs.executable.path=path/to/phantomjs
```

12. Save the file and restart the JasperReports Server (see Start/Stop the JasperReports Server below).

13. Validate the installation by logging in to the JasperReports Server console. Use your browser to access the server. Replace `<hostname>` below with the name or IP address of the server:

```
http://<hostname>:8080/jasperserver-pro
```

14. Log in with the user ID of *superuser* and the default password of *superuser*.



TIBCO Jaspersoft

Getting Started

- How-to videos
- How-to articles
- Online Learning Portal
- Product documentation
- Find the right edition for you
- Jaspersoft Quick Start Guide

What's new in Jaspersoft Version 6?

Dashboards:
A single web environment for creating visualizations, with a mash-up of Reports, Ad Hoc Views, and web pages

All New JasperMobile Support:
View interactive reports and Dashboards from your smartphone or tablet with the JasperMobile app

Visualize.js:
Embed Reports and Dashboards directly into highly-interactive web pages

Expanded Multi-organization Administration:
New tools to manage attributes, export/import, and scheduling across multiple organizations

Help

User ID: superuser

Password:

Show locale & time zone

Login

Need help logging in?

Update the License File

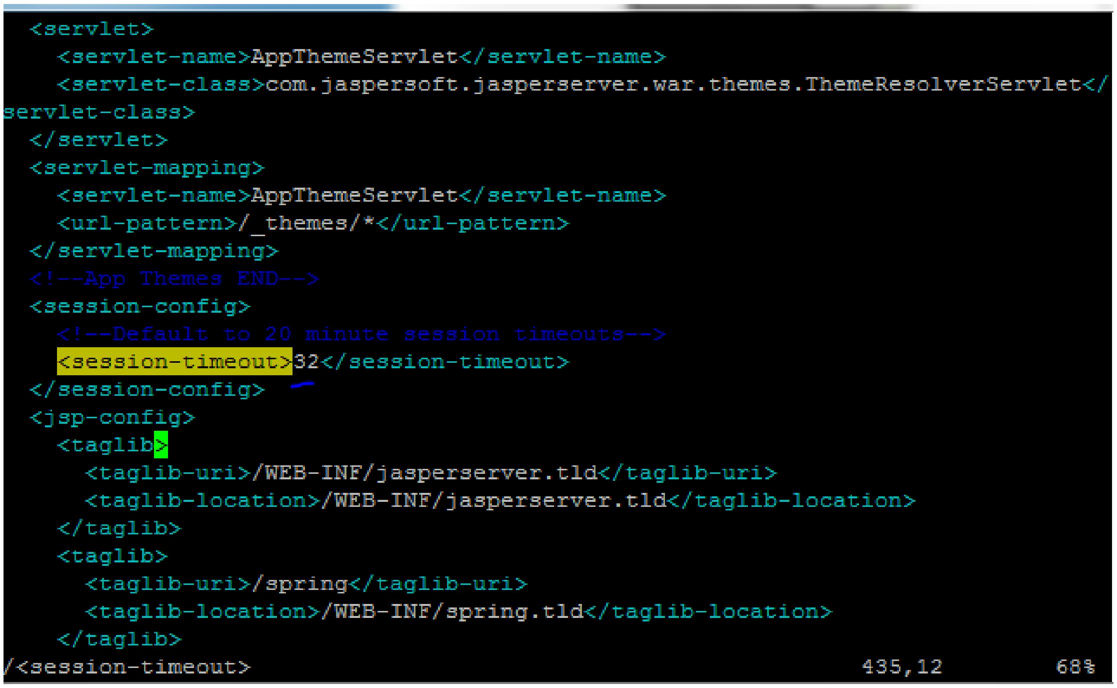
Follow these steps to update the license file for the JasperReports server:

1. Copy the license file (`jasperserver.license`) from the distribution site and move it to your home directory using FTP or WinSCP.
2. Replace the license file in the JasperReports installation directory (`/opt/jasperreports-server-6.2.1/`) with the license file from the above step, overwriting the existing `jasperserver.license` file.
3. Restart the JasperReports server (see Start/Stop the JasperReports Server below).

Increase the JasperReports Session Timeout Period

By default, the JasperReports timeout period is set to 20 minutes. However, the timeout period should be reset to 32 minutes in order to match with Keycloak.

1. Open `web.xml` from the path `/opt/jasperreports-server-6.2.1/apache-tomcat/webapps/jasperserver-pro/WEB-INF/web.xml`
2. Search for the property `<session-timeout>` under `<session-config>` and update the value to **32**. After this, the property and value should be similar to the image below:



```
  <servlet>
    <servlet-name>AppThemeServlet</servlet-name>
    <servlet-class>com.jaspersoft.jasperserver.war.themes.ThemeResolverServlet</
servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>AppThemeServlet</servlet-name>
    <url-pattern>/_themes/*</url-pattern>
  </servlet-mapping>
  <!--App Themes END-->
  <session-config>
    <!--Default to 20 minute session timeouts-->
    <session-timeout>32</session-timeout>
  </session-config>
  <jsp-config>
    <taglib>
      <taglib-uri>/WEB-INF/jasperserver.tld</taglib-uri>
      <taglib-location>/WEB-INF/jasperserver.tld</taglib-location>
    </taglib>
    <taglib>
      <taglib-uri>/spring</taglib-uri>
      <taglib-location>/WEB-INF/spring.tld</taglib-location>
    </taglib>
/<session-timeout>                                              435,12        68%
```

3. Restart the JasperReports server (see Start/Stop the JasperReports Server below).

Start/Stop the JasperReports Server

Use these commands to start and stop the JasperReports server.

Start

```
$ cd /opt/jasperreports-server-6.2.1
 $ ./ctlscript.sh start
```

Stop

```
$ cd /opt/jasperreports-server-6.2.1
 $ ./ctlscript.sh stop
```

JasperReports Configuration with Keycloak

A few configuration changes need to be made in order for JasperReports to be used with Keycloak.

Prerequisite

Keycloak must be installed and set up, if the SSO-enabled version of IWeb and VOMS is being used.

JasperReports Server Modifications

1. Edit the following values in the `applicationContext-externalAuth-oAuth.xml` file:

Key	XML Tag Name(s) to Update	Value
[KEYCLOAK_URL]	<authorization_location> <token_location> <userdetails_location> <logoutUrl>	URL of the Keycloak server. Example: `https://sso-dev.stchome.com/`
[KEYCLOAK_REALM]	<authorization_location> <token_location> <userdetails_location> <logoutUrl>	Keycloak realm that the client belongs to. This value needs to be changed, along with <keycloak_URL>. In this example, the realm name is *Dev-Integration*: `https://sso-dev.stchome.com/auth/realms/`**`Dev-Integration`**
[JASPERSERVER_URL]	<property name="redirecturl"> <property name="logoutUrl">	URL of the Jasper server. Example: `http://<server_name>:8080/jasperserver-pro/oauth`
[CLIENT_ID]	<property name="client_id">	Client ID as set on Keycloak. Example: <property name="client_id"> <value>jasper-aws-dev</value>

[CLIENT_SECRET]	<property name="clientsecret"> <property name="userdetails_secret">	Credentials secret key generated by Keycloak.
State_ID	<property name="state">	This is the global setting used to populate the STATE_ID attribute in Jaspersoft. Example: <bean id="oAuthUserDetailsService" class="com.jaspersoft.jasperserver.ps.OAuth.OAuthUserDetailsServiceImpl"> <property name="state"> <value>WA</value> </property> </bean>
Jaspersoft Roles	<bean id="mtExternalUserSetupProcessor" class="com.jaspersoft.jasperserver.multipleTenancy.security.externalAuth.processors.MTExternalUserSetupProcessor" parent="abstractExternalProcessor"> <property name="organizationRoleMap"> <map> <!-- Example of mapping customer roles to JRS roles --> <entry> <key> <value>ROLE_ADMIN_EXTERNAL_ORGANIZATION</value> </key> <value>ROLE_ADMINISTRATOR</value> </entry> </map>	All the Jaspersoft Roles are configured inside this tag. If any new Jaspersoft Roles are needed, they must be configured here. In the <entry> tag, the key value should match with the client role defined at the Keycloak end. Currently, the following roles are configured: 1. Administrator 2. Superuser 3. User Role

2. Copy the `applicationContext-externalAuth-oAuth.xml` file (edited in Step 1) to `<JASPERSERVER_DIR>/apache-tomcat/webapps/jasperserver-pro/WEB-INF`.

3. Copy stc-jaspersoft-oauth-1.0.0.jar (available in the JRS distribution folder) to `<JASPERSERVER_DIR>/apache-tomcat/webapps/jasperserver-pro/WEB-INF/lib`.

4. Create and save a backup of the `<JASPERSERVER_DIR>/apache-tomcat/webapps/jasperserver-pro/WEB-INF/applicationContext-security.xml` file.

5. In `<JASPERSERVER_DIR>/apache-tomcat/webapps/jasperserver-pro/WEB-INF/applicationContext-security.xml`, replace the access value for the method `com.jaspersoft.jasperserver.api.metadata.user.service.ProfileAttributeService.putProfileAttribute` with **ROLE_USER**.

6. Save a backup copy of the `<JASPERSERVER_DIR>/apache-tomcat/webapps/jasperserver-pro/WEB-INF/applicationContext-security-web.xml` file again and replace the constructor-arg value for org.springframework.security.web.authentication.LoginUrlAuthenticationEntryPoint with **/oauth**.

7. Add the SSL certificate used by Keycloak to the Java installation that Jaspersoft is using. (By default, Java only knows about a small number of root certificates.) To do this, follow these steps:

    a. Copy `InstallCert$SavingTrustManager.class` and `InstallCert.class` (both available in the `STC Suite Distribution/Server Components/JRS` directory) to the `<JASPERSERVER_DIR>/java/bin` directory.

    b. NOTE for sso-dev.stchome.com: This is only for Dev SSO SSL certificate installation. This URL changes based on your server hostname of the Keycloak/SSO server.

    c. Go to the `<JASPERSERVER_DIR>/java/bin` directory and execute these commands:

```
sudo ./java InstallCert sso-dev.stchome.com
 sudo ./keytool -exportcert -alias sso-dev.stchome.com-1 -keystore
 jssecacerts -storepass changeit -file sso-dev.stchome.com.cer
 sudo ./keytool -importcert -alias sso-dev.stchome.com -keystore
 ../lib/security/cacerts -storepass changeit -file sso-
 dev.stchome.com.cer
```

8. Make database changes with the following commands. Jaspersoft stores all of its metadata in the Postgres database. All of the user-based profile attributes are stored in the *jiprofileattribute* table in Jaspersoft. By default, user attributes support a character size of 200. In Step E below, the data type size is changed from 200 to 1,000.

    a. Navigate to the `<Jasper Installation Directory>/postgres` directory.

    b. Run this command: `sudo bin/psql -U postgres -W` (the password is *postgres*).

    c. Now you should see at the prompt: `postgres=#`

    d. Enter this command to connect to the Jasperserver database in Postgres: `\c jasperserver;` (the password is *postgres*).

    e. Execute this SQL alter script: `ALTER TABLE jiprofileattribute ALTER COLUMN attrvalue TYPE character varying(100000);`

9. Restart the Jasper server and navigate to `JasperUrl/oauth` (see example). You should now be redirected to Keycloak to log in. Example: `http://52.10.228.158:8080/jasperserver-pro/oauth`

Steps to Debug Issues Related to Keycloak/JasperReports Server Integration

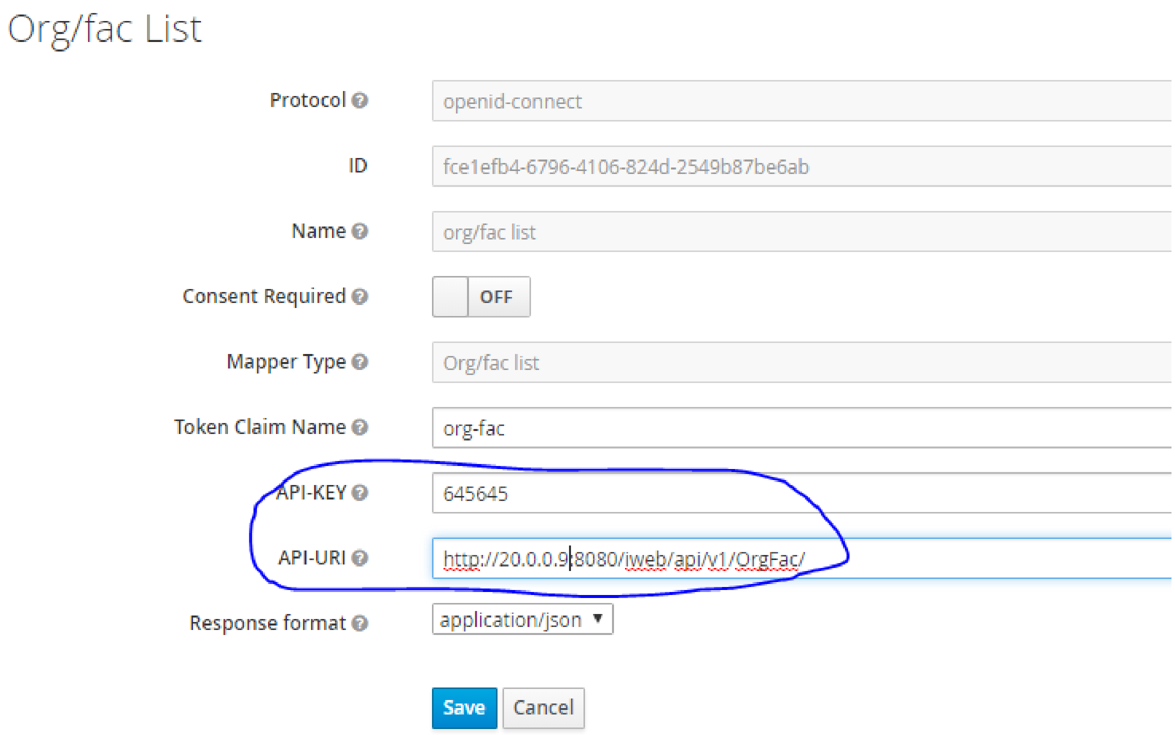
If there are any issues with integrating Keycloak with Jasper, follow these suggestions below.

Check the applicationContext-externalAuth-oAuth.xml File

Make sure that `applicationContext-externalAuth-oAuth.xml` is configured with the correct Keycloak and Jasper server parameters and with the correct client_secret.

Keycloak Mappers

Make sure that the Jasper client has the below mappers mapped correctly in Keycloak, and check if the Org/fac List mapper contains the correct parameters. (Note that your API-KEY and API-URI will differ from the image below.



Org/fac List

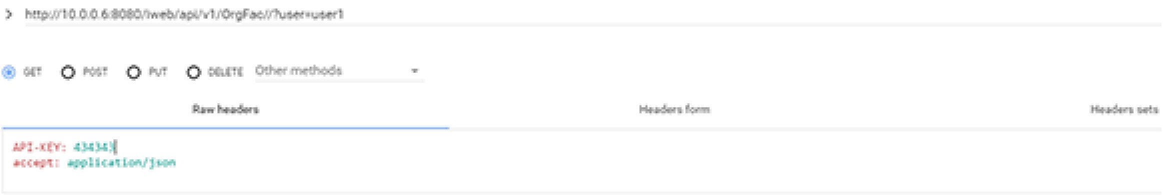
Protocol	openid-connect
ID	fce1efb4-6796-4106-824d-2549b87be6ab
Name	org/fac list
Consent Required	OFF
Mapper Type	Org/fac list
Token Claim Name	org-fac
API-KEY	645645
API-URI	http://20.0.0.9:8080/iweb/api/v1/OrgFac/
Response format	application/json ▾

Name	Category	Type
org/fac list	Token mapper	Org/fac list
email	Token mapper	User Property
groups	Token mapper	Group Membership
username	Token mapper	User Property
realm roles	Token mapper	Realm Role List

roles	Token mapper	Client Role List
given name	Token mapper	User Property
family name	Token mapper	User Property
full name	Token mapper	User's full name

Validate the IWeb Web Service

Make sure that the IWeb web service URL is accessible. The output of this should return the org/fac list. If there is an error, it needs to be fixed.



http://10.0.0.6:8080/iweb/api/v1/OrgFac//?user=user1

```
API-KEY: 434343
accept: application/json
```

Check the Whitelist Rules

If Jaspersoft is installed in an AWS environment, the outbound connection on port 8080 should be enabled for the IWeb server.

If Keycloak is installed in an AWS environment, the outbound connection on port 8080 should also be enabled for the IWeb server.

Check the Firewall Rules

If there are any special firewall rules preventing the Jasper and/or Keycloak servers from accessing IWeb through port 8080, the need to be fixed so that the port is enabled.

JasperReports Server Upgrade - Linux

These instructions are only required when deploying SMaRT AFIX for the first time or when upgrading from a version of SMaRT AFIX older than 1.17.4. If you have already deployed SMaRT AFIX 1.17.4, you should already have the web service plugin installed on your JasperReports server and can skip these instructions.

As a part of SMaRT AFIX 1.17.4, the new SMaRT AFIX Online Tool web service was introduced to help support the CDC's AFIX Online Tool application. In order to integrate this tool with SMaRT AFIX, a plugin must be added to the existing JasperReports server.

Prerequisites

1. JasperReports server has been installed.
2. The web service plugin zip file has been downloaded to your application server (APP) and the contents extracted. The unzipped folder is referenced as $WS_DS from this point on.

```
/APP/JRS/webserviceds_v1.0_3.zip
```

3. Locate your JasperReports server web application folder, which is referred to as $JRS_HOME from this point on. It is recommended that you back up this folder before beginning the plugin installation.

   ```
   /opt/jasperreports-server-6.2.1/apache-tomcat/webapps/jasperserver-pro
   ```

Plugin Installation Steps

The installation should be performed as root user.

1. Stop your web server running JasperReports server.
2. Open the folder `$WS_DS/JRS/WEB-INF/bundles` and copy the contents to `$JRS_HOME/WEB-INF/bundles/`. Note that this should not overwrite any existing files.

   ```
   cd /$WS_DS/JRS/WEB-INF/bundles/
     cp * /$JRS_HOME/WEB_INF/bundles/
   ```

3. Open the folder `$WS_DS/JRS/WEB-INF/lib` and copy the contents to `$JRS_HOME/WEB-INF/lib/`. Note that this should not overwrite any existing files.

   ```
   cd /$WS_DS/JRS/WEB-INF/lib/
     cp * /$JRS_HOME/WEB_INF/lib/
   ```

4. Open the `$WS_DS/JRS/WEB-INF/` folder and copy the `applicationContext-WebServiceDataSource.xml` file to `$JRS_HOME/WEB-INF/`. Note that this should not overwrite any existing files.

   ```
   cd /$WS_DS/JRS/WEB-INF/
     cp applicationContext-WebServiceDataSource.xml /$JRS_HOME/WEB_INF/
   ```

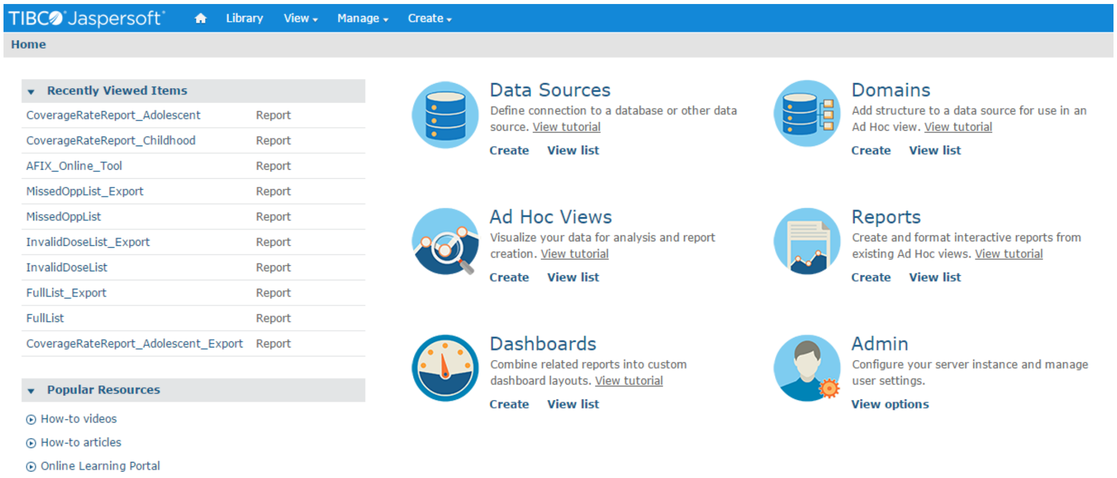
5. Restart the web server running JasperReports server.

Validation Steps

Follow these steps to confirm that the plugin has been successfully installed:

1. Navigate to your JasperReports server admin console.

2. Log in as an admin-level user.



Recently Viewed Items

CoverageRateReport_Adolescent	Report
CoverageRateReport_Childhood	Report
AFIX_Online_Tool	Report
MissedOppList_Export	Report
MissedOppList	Report
InvalidDoseList_Export	Report
InvalidDoseList	Report
FullList_Export	Report
FullList	Report
CoverageRateReport_Adolescent_Export	Report

Popular Resources

- How-to videos
- How-to articles
- Online Learning Portal

Data Sources
Define connection to a database or other data source. View tutorial
Create View list

Domains
Add structure to a data source for use in an Ad Hoc view. View tutorial
Create View list

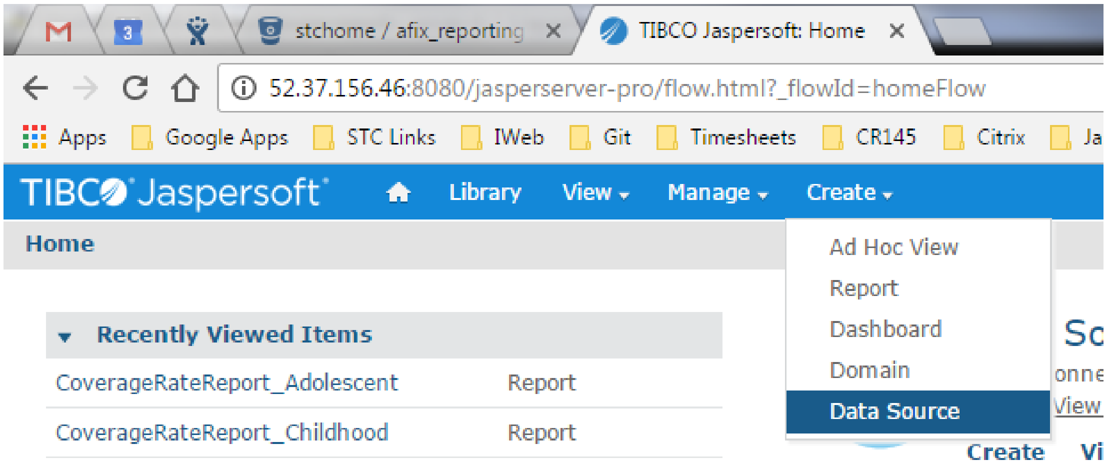
Ad Hoc Views
Visualize your data for analysis and report creation. View tutorial
Create View list

Reports
Create and format interactive reports from existing Ad Hoc views. View tutorial
Create View list

Dashboards
Combine related reports into custom dashboard layouts. View tutorial
Create View list

Admin
Configure your server instance and manage user settings.
View options

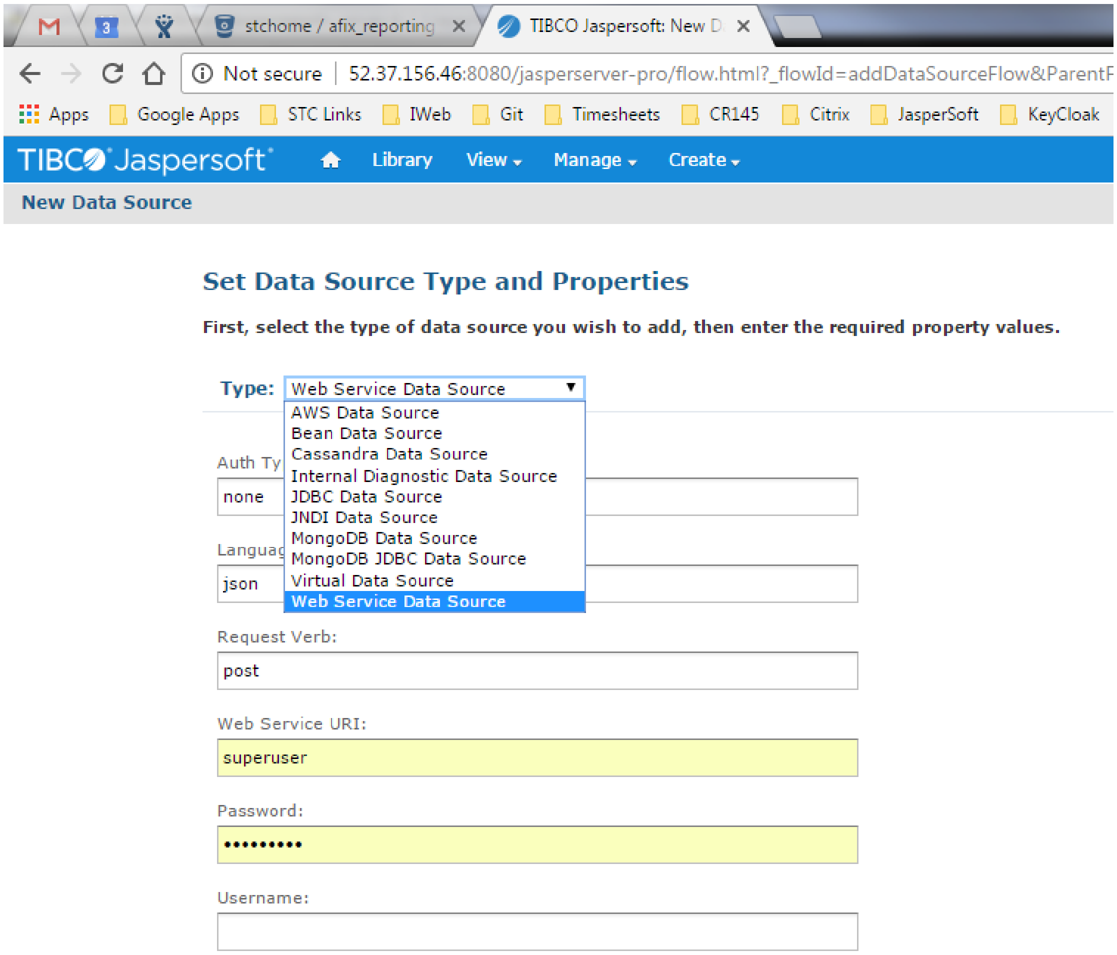
3. Select **Create > Data Source** from the top menu.



Recently Viewed Items

CoverageRateReport_Adolescent	Report
CoverageRateReport_Childhood	Report

4. Confirm that **Web Service Data Source** is listed in the Type drop-down list.



New Data Source

Set Data Source Type and Properties

First, select the type of data source you wish to add, then enter the required property values.

Type: Web Service Data Source

Auth Ty
none

Languag
json

Request Verb:
post

Web Service URI:
superuser

Password:
•••••••••

Username:

SMaRT AFIX JasperReports Deployment - Linux

The following outlines how to import (migrate) Jaspersoft Reports for SMaRT AFIX.

Pre-Migration Checklist

- Confirm that the Postgres database is installed and configured
- Retrieve the passwords for the SMaRT AFIX database (*afixdb*)
- Retrieve the Jaspersoft superuser username and password
- Confirm that the SMaRT AFIX Online Tool web service is installed
- Confirm that all necessary ports are open across all firewalls (the default port is 80)

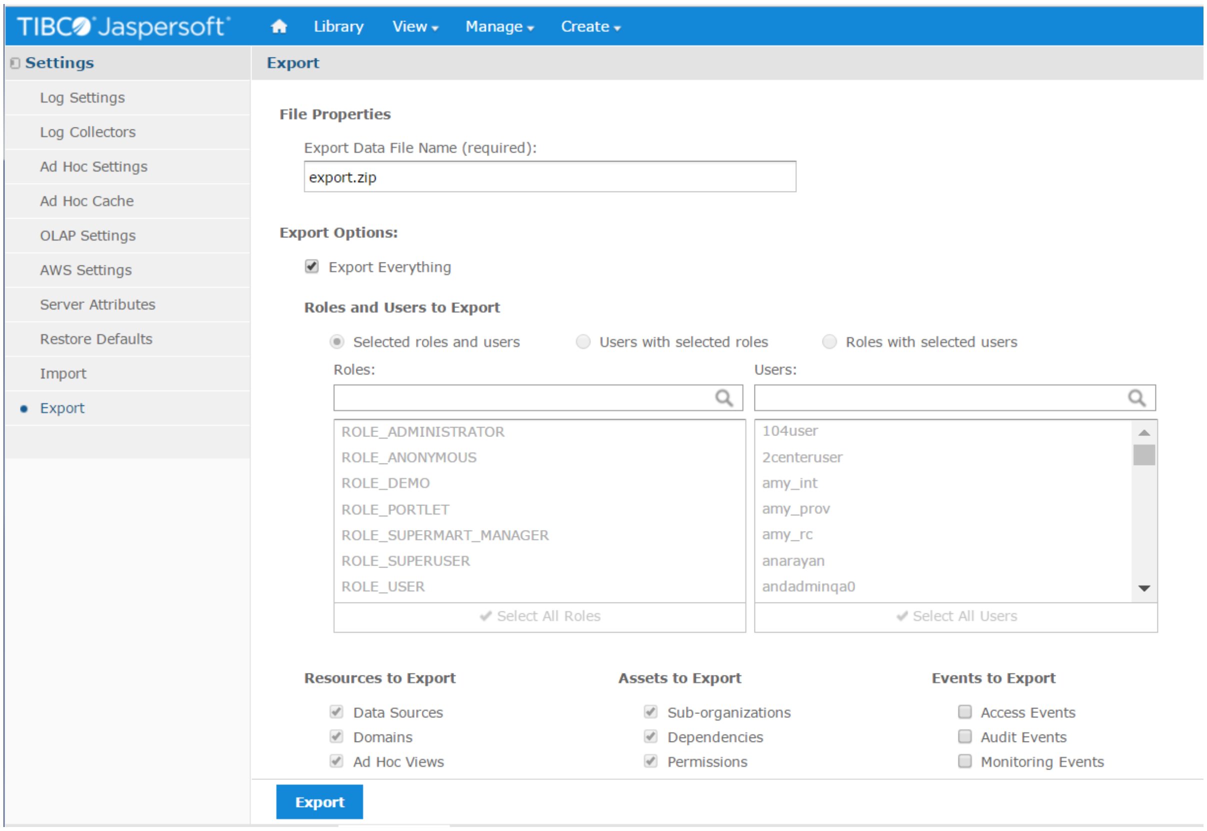
- Confirm that the release distribution zip file for Jasper Reports is extracted from `/APP/JRS/AFIX_Reports-[version number].zip` and copied to your local drive.
- **Important:** Be sure to update the JasperReports server by following the instructions in SMaRT AFIX JasperReports Server Upgrade - Linux.

Procedure Steps

Follow these steps to deploy Jaspersoft Reports.

Export the Jaspersoft Repository for Backup

Export the repository by selecting **Manage > Server Settings > Export (Export Everything)** in Jaspersoft. Save the file as a backup.



Export

File Properties

Export Data File Name (required): export.zip

Export Options:

☑ Export Everything

Roles and Users to Export

● Selected roles and users ○ Users with selected roles ○ Roles with selected users

Roles:

ROLE_ADMINISTRATOR
ROLE_ANONYMOUS
ROLE_DEMO
ROLE_PORTLET
ROLE_SUPERMART_MANAGER
ROLE_SUPERUSER
ROLE_USER

✓ Select All Roles

Users:

104user
2centeruser
amy_int
amy_prov
amy_rc
anarayan
andadminqa0

✓ Select All Users

Resources to Export	Assets to Export	Events to Export
☑ Data Sources	☑ Sub-organizations	□ Access Events
☑ Domains	☑ Dependencies	□ Audit Events
☑ Ad Hoc Views	☑ Permissions	□ Monitoring Events

Import the Content via the Portal

1. Log in to the Jaspersoft BI Portal as the administrator user.
2. Navigate to **Server Settings > Import**.
3. Import the `AFIX_Reports-[version number].zip` file.
4. Update the data source with the database connection information.

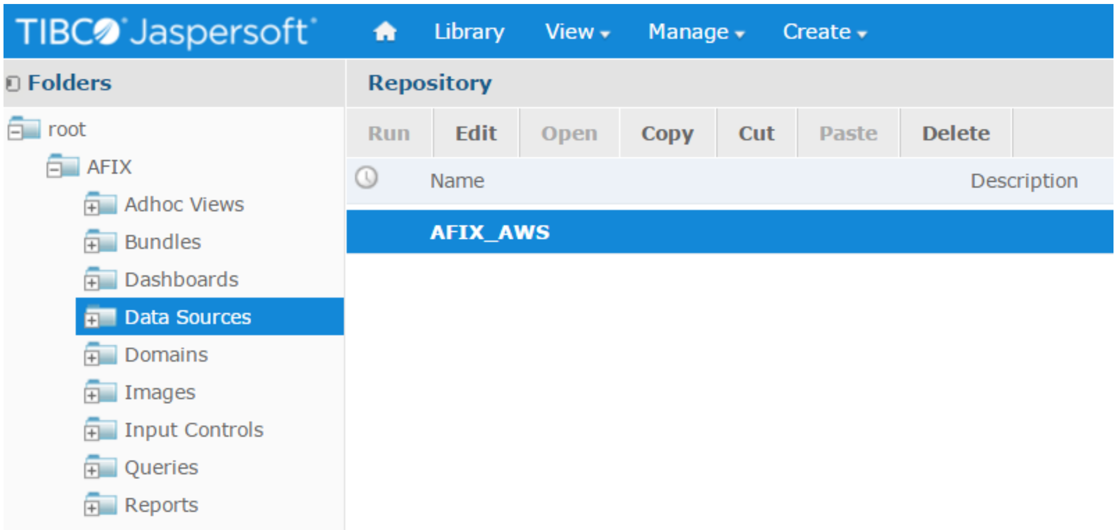
Import the Content via the Command Line

1. Confirm that JAVA_HOME is set and your path includes `jre\bin` or `jdk\bin`.

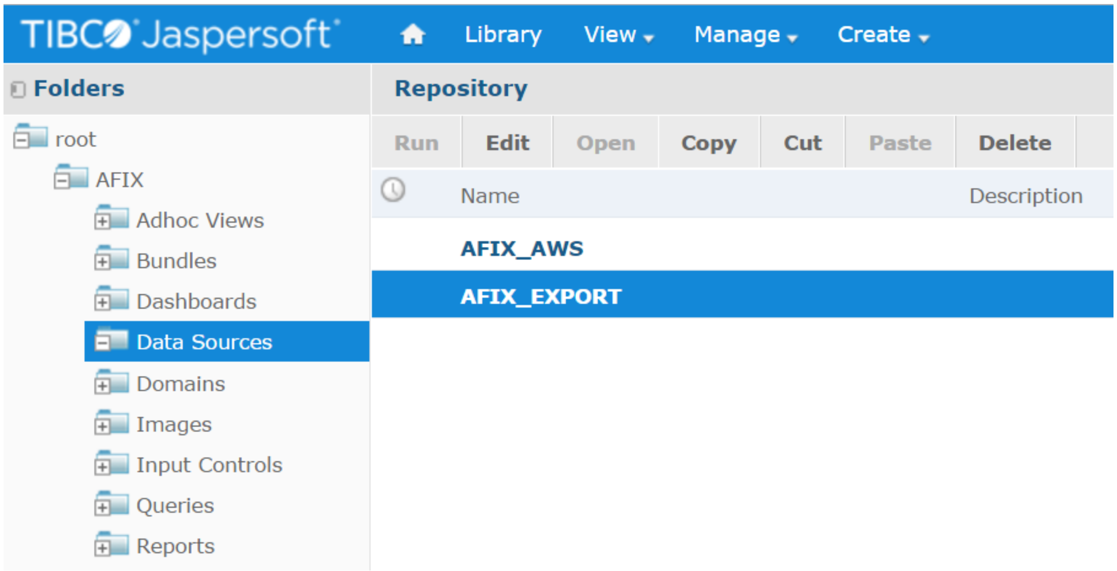
2. Change to the buildomatic directory on the JasperReports server: `C:\Jaspersoft\jasperreports-server-6.2.1\buildomatic`

3. Execute the migrate command (adding in the correct version number): `js-import.sh --input-zip "AFIX_Reports-[version number].zip" --update --skip-user-update --include-access-events --include-audit-events --include-monitoring-events`

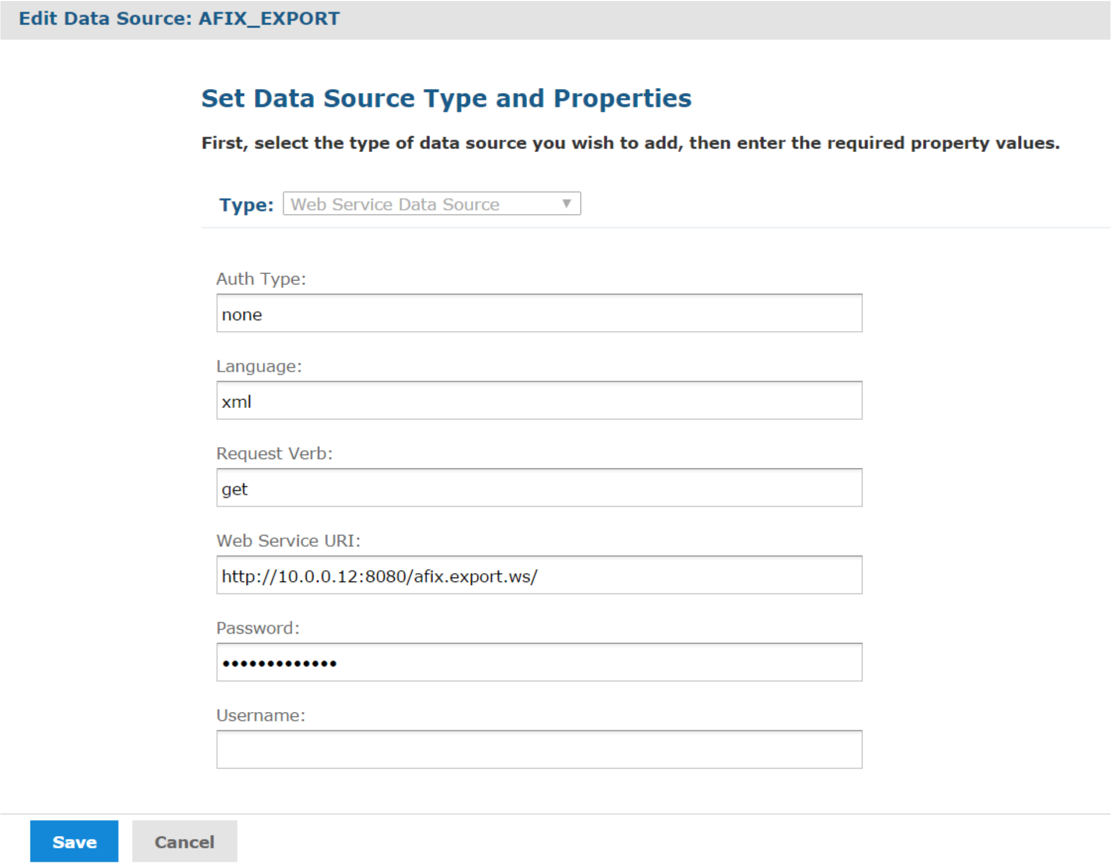
4. Update the data source with the database connection information: From the left menu, go to **root > AFIX > Data Sources**. Select **AFIX_AWS**, then click the Edit tab. Specify your database connection properties for Host, Port, and Password, then click **Save**.



5. Select **AFIX_EXPORT**, then click the Edit tab. This is the web service data source linked to the SMaRT AFIX Online Tool web service.



6. Leave the defaults for Auth Type, Language, and Request Verb. Update the web service URI to point to your SMaRT AFIX Online Tool web service deployment. Note that even though the Password field is populated, no password is present or required; leave it as it is. Click **Save** when finished.



Edit Data Source: AFIX_EXPORT

Set Data Source Type and Properties

First, select the type of data source you wish to add, then enter the required property values.

Type: Web Service Data Source

Auth Type: none

Language: xml

Request Verb: get

Web Service URI: http://10.0.0.12:8080/afix.export.ws/

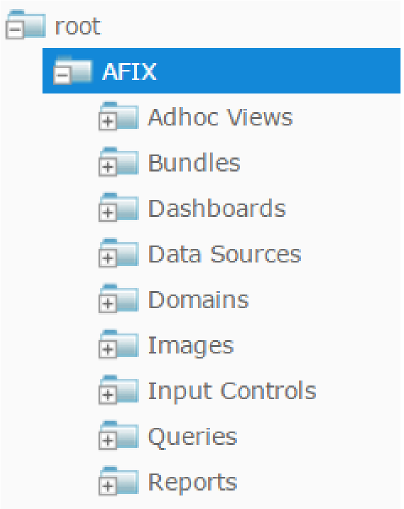
Password: •••••••••••••

Username:

Validate the New Content

The following steps perform a spot check on the newly created content to validate it and to check for any serious errors. It does not perform a full check of all possible issues and should be followed up with a functional validation of the content.

1. Log on to the Jaspersoft BI server as a user able to run a report. The user should be able to log in successfully and see the subfolders under `/root/AFIX` (select **View > Repositories**).



2. Validate that the user can execute and view a report as defined by their class or organization membership: Select the report from `AFIX/Reports/AFIX_Online_Tool`, select one report from any of the other sub-directories of `AFIX/Reports`, and confirm that both reports execute successfully.

Restore the Previous Content (if Necessary)

If there are any uncorrectable issues and the release upgrade needs to be aborted, follow these steps to revert to the previous content:

1. Using the backup exported at the beginning of these steps, import using the steps above. Note that a re-import of the previous version of the content will also achieve the same back-out.
2. Perform the validation steps listed above.

Node.js Installation - Linux

Follow these steps to install Node.js on Linux.

1. Create the user environment: A user needs to be created to run Node.js applications. However, it is not recommended that Node.js applications be run as a privileged user.

```
sudo adduser node
```

2. Become the node user:

```
sudo su - node
```

3. To install Node.js for the previously created node user, Node Version Manager is recommended. This allows for more flexible control over the environment and an easier future upgrade path.

```
curl -o-
 https://raw.githubusercontent.com/creationix/nvm/v0.32.1/install.sh |
 bash
 source ~/.bashrc
```

4. Install the required version of Node.js:

```
nvm install 6.9.1
 nvm use 6.9.1
 nvm alias default 6.9.1
```

5. Install the PM2 node process manager:

```
npm install -g pm2
```

SMaRT AFIX Application Installation Instructions - Linux

The following instructions explain how to install the application, the Oracle Instant Client, and Redis; how to configure the application dashboard and quick links; specific configuration options; and information on standalone versus multiple application deployment.

If you are upgrading from a previously installed version of SMaRT AFIX, you can skip directly to the Install the Deployment Package section and continue from there.

Prerequisites

The following should already be installed and configured:

- Linux CentOS 7
- PostgreSQL server (DBB)
- Oracle server (COR)
- Node.js (APP)
- PM2 (APP)

Installation and Deployment of SMaRT AFIX, iQ, and VOMS

If you are installing SMaRT AFIX alongside the iQ and/or VOMS applications, you only need to install the Oracle Instant Client and Redis with the first application installation. Once these are installed, for subsequent applications that are running on the same server, you can skip down to the Install the Deployment Package section and continue from there instead.

Install the Oracle Instant Client

1. Download and install the Oracle Instant Client (both basic and SDK). The `.rpm` files can be found at http://www.oracle.com/technetwork/topics/linuxx86-64soft-092277.html. Install them with the RPM command.

2. Add the following to the users: `.bashrc`, `.cshrc`, and `.zshrc`, or use the default per-interactive-shell startup file of the default shell of the user for which you want to run the application. Optionally, instead of adding them on a per-user basis, the following can be added to `/etc/profile` or `/etc/profile.d/oracle.sh`:

   ```
   export OCI_LIB_DIR=/usr/lib/oracle/12.1/client64/lib
    export OCI_INC_DIR=/usr/include/oracle/12.1/client64
   ```

3. Create a file in `/etc/ld.so.conf.d/oracle.conf` with the path to the Oracle Instant Client libraries, which by default are as follows:

   ```
   /usr/lib/oracle/12.1/client64/lib
   ```

Install Redis

When installing Redis, make sure `epel repo` is enabled:

```
yum install redis
 systemctl enable redis
 systemctl start redis
```

Install the Deployment Package

Copy the provided zip/tar file to the target server. Extract the archive and change into the parent of the directory where it was extracted.

Configure the Application Dashboard and Quick Links

If you are installing SMaRT AFIX alongside the iQ and/or VOMS applications, the steps below only need to be completed the first time an application is installed. This file should be saved for all subsequent installations.

1. Download the `apps.json` file from the release distribution application components shared folder.

2. Edit this file with a text editor and replace the placeholders DASHBOARD_URL, IWEB_URL, AFIX_URL, PHC-HUB_URL, IQ_URL, and VOMS_URL (if applicable for your installation) with their respective URLs.

3. Save this file and keep it handy as you will need to use it in the installation of iQ, VOMS, and the SSO dashboard.

An example of one such configuration object in the JSON file is shown below. In most cases, only the URL value needs to change:

```
{
 "icon": "/static/public/img/STC - SmartAFIX.svg",
 "id": "AFIX",
 "name": "SMaRT AFIX",
 "url": "https://afix.stchome.com ",
 "description": "Utilized to make AFIX assessments efficient, standardized and
 meaningful.",
 "hideIfUnavailable": true,
 "accessRole": "Access_afix"
  }
```

Copy this file to the following location:

```
<application_install_dir>/src/shared/helpers/appActions/
```

Configure the Reverse Proxy

A reverse proxy is recommended to provide access to the various products. This is not explicitly required unless the products are running with the ENABLE_SSL option set to *true*. However, this will generally be the case when the products are being run in conjunction with IWeb on the same server. See the example Apache configurations at `package/docs/apache`.

General Product Configuration Options

Once the package containing the products has been installed, there are some configuration options that can be set depending on the environment. The majority of these lie in `package/process.json`, which is simply a JSON configuration file for pm2. The options listed in bold below need to be changed to match your environment. Other options not in bold might not need to change, and in most cases can be the default setting. Some options in the `process.json` file are not listed at all below; those options should be left unchanged.

The options are as follows:

Option	Description
script	Location of the startup script application. The path to the script may need to be customized.
cwd	The path to the application. May need to be customized.
PRODUCT	The name of the product.
HOST	The host name for the product.
APIHOST	The host name of the product's respective API server (generally

	localhost).
PORT	The port for the product.
APIPORT	The port for the product's API server.
NODE_ENV	This should always be **production**.
SAML_ISSUER	The SAML issuer (client) from Keycloak.
SAML_ENTRY_POINT	The SAML entry point from Keycloak.
JASPER_SERVER	The domain name of the JasperReports server.
REDIS_URL	The Redis URL used for storing client sessions.
REVERSE_PROXY	When configuring SMaRT AFIX with a reverse proxy, this should be set to *true*.
ENABLE_SSL	When set to *true*, this enables HTTPS support. If REVERSE_PROXY is set to *true*, this should be set to *false* even when the environment is configured for HTTPS.
SECRET_SESSION	The secret key for session storage. **IMPORTANT NOTE:** This value must be exactly the same for every application, in both the server and the API environment variables.

SMaRT AFIX-Specific Configuration Options

The following options are specific to SMaRT AFIX configuration settings:

Option	Description
VOMS_ORACLE_USER	The username for IWeb's SIIS database. This is set to the default user for the SIIS, but may need to be updated depending on your configuration.
VOMS_ORACLE_PASSWORD	The password for IWeb's SIIS database. This is set to the default password for the SIIS, but may need to be updated depending on your configuration.
VOMS_ORACLE_CONNECTION	Connection information for IWeb's SIIS database. This needs to be updated to point to your IWeb's Oracle database instance.
PG_USER	The user account for the SMaRT AFIX PostgreSQL database.
PG_PASSWORD	The password for the associated PG_USER for the SMaRT AFIX PostgreSQL database.
PG_HOST	The hostname for the SMaRT AFIX PostgreSQL database.
PG_PORT	The port number for the SMaRT AFIX PostgreSQL database.
AFIX_EXPORT_URI	The URL for the SMaRT AFIX Online Tool web service. Only the hostname and possibly the port number for the URL need to be updated.
AFIX_EXPORT_KEY	The API Key for the SMaRT AFIX Online Tool web service. This should be left as the default.
CDSI_FORECASTER_VERSION	The current version of the CDSI Forecaster deployed. For the SMaRT AFIX 1.17.5.4 release, it should be set to **5.17.10**.

API_URL_PREFIX	This should be set to **/api**

Standalone SMaRT AFIX vs. Multiple Application Deployment

The SMaRT AFIX bundle you were provided contains an Org-Fac-Micro-Service include in the `process.json` file that looks something like the example below. The same micro-service is included in the `process.json` file for the iQ application. If you are deploying SMaRT AFIX alongside iQ, this service should only be included in the `process.json` file for the first application you deploy. It needs to be removed from the `process.json` file for all other applications.

```
{
    "name"   : "org-fac-micro-service",
    "script"   : "afix-pre-windows/services/bin/service.js",
    "instances"   : 2,
    "exec_mode"   : "cluster",
    "cwd"   : "afix-pre-windows/services",
    "env": {
    "NODE_ENV"   : "production",
    "SERVICE"   : "facilitySearch",
    "SERVICE_PORT"   : 3060,
    "SAML_ISSUER"   : "dev-portal",
    "SAML_ENTRY_POINT"   : "https://sso-dev.stchome.com/auth/realms/Dev-
 Integration/protocol/saml",
    "JASPER_SERVER"   : "interop-aws-jaspersoft.stchome.com",
    "REDIS_URL"   : "redis://localhost:6379",{
    "VOMS_ORACLE_USER"   : "user",
    "VOMS_ORACLE_PASSWORD"   : "password",
    "VOMS_ORACLE_CONNECTION"   : "(DESCRIPTION = (ADDRESS = (PROTOCOL =
 TCP)(HOST = 10.0.0.1)(PORT = 1521)) (CONNECT_DATA = (SID = SID)))",
    }
 },
```

NOTE: Due to variations in the org-fac-micro-service between the different applications (SMaRT AFIX, VOMS, and iQ), it is recommended that you deploy the service that is bundled with SMaRT AFIX instead of the one bundled with other applications.

Run the Application

Use pm2 to run the deployment package as shown below. In many cases, environment-specific configuration options must be set prior to running an application's package. See the configuration option sections above for more details.

```
pm2 start package-parent/process.json
```

Start the Products upon Bootup

To have the products automatically start on every boot, do the following once it has been verified that they are running correctly:

```
pm2 save
 pm2 startup
```