



SMaRT AFIX
Powered by STC

SMaRT AFIX Installation Guide

v1.17.5.2



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/>.

Connect with Us on Social Media



Additional Information

This documentation describes the following: SMaRT AFIX installation.

(20170817)

Table of Contents

SMaRT AFIX Installation Instructions	1
Components	1
New Application Installations	2
Linux Instructions	4
Build a Linux (CentOs) Server	4
Install CentOS 7 Minimal and Configure	4
Forecaster Installation Instructions - Linux.....	5
Install Oracle Express	5
Install the SAF Database	6
Install the Web Application	6
Postgres Database Installation - Linux.....	8
Install on Linux CentOS 7	8
Modify Postgres Server Configuration Settings	9
SMaRT AFIX Database Initialization - Linux	10
Prerequisite Tasks	10
Initialize a New Database (for the First Time Only)	11
Upgrade/Migrate an Existing Database	12
Set Up a Daily Job / Drop Partition	13
SMaRT AFIX Online Tool Web Service Installation - Linux	13
Prerequisites.....	13
Installation Steps	14
Jaspersoft ETL Installation - Linux	14
Prerequisites.....	14
Install Jaspersoft ETL	15
Install the Apache Web Server and Subversion	16
Start/Stop Jaspersoft ETL Services.....	17
Configure the Admin Console	18
SMaRT AFIX Jaspersoft ETL Deployment - Linux	24
Pre-Migration Checklist.....	24
New Installation Instructions (for the First Time Only)	24
Upgrade Instructions.....	31
JasperReports Server Installation - Linux.....	34
Prerequisite	34
Installation Steps	35
Update the License File.....	36

Increase the JasperReports Session Timeout Period	36
Start/Stop the JasperReports Server	37
JasperReports Configuration with Keycloak.....	37
Steps to Debug Issues Related to Keycloak/JasperReports Server Integration	40
JasperReports Server Upgrade - Linux.....	42
Prerequisites.....	42
Plugin Installation Steps	42
Validation Steps	43
SMaRT AFIX JasperReports Deployment - Linux	44
Pre-Migration Checklist.....	45
Procedure Steps.....	45
Node.js Installation - Linux.....	49
SMaRT AFIX Application Installation Instructions - Linux	49
Prerequisites.....	50
Installation and Deployment of SMaRT AFIX, iQ, and VOMS	50
Install the Oracle Instant Client	50
Install Redis	50
Install the Deployment Package.....	51
Configure the Application Dashboard and Quick Links	51
Configure the Reverse Proxy	51
General Product Configuration Options	52
SMaRT AFIX-Specific Configuration Options	53
Standalone SMaRT AFIX vs. Multiple Application Deployment	53
Run the Application.....	54
Start the Products upon Bootup.....	54
Windows Instructions	55
Forecaster Installation Instructions - Windows.....	55
Install Oracle Express	55
Install the SAF Database	56
Install the Web Application	56
Postgres Database Installation - Windows	58
Install on Windows	58
Modify Postgres Server Configuration Settings	59
SMaRT AFIX Database Initialization - Windows	60
Prerequisite Tasks	60
Initialize a New Database (for the First Time Only)	61
Upgrade/Migrate an Existing Database	63
Set Up a Daily Job / Drop Partition	64

SMaRT AFIX Online Tool Web Service Installation - Windows	64
Prerequisites.....	64
Installation Steps	65
Jaspersoft ETL Installation - Windows.....	65
Prerequisites.....	65
Install Jaspersoft ETL	66
Install Subversion.....	66
Start/Stop Jaspersoft ETL Services.....	67
Configure the Admin Console	68
SMaRT AFIX Jaspersoft ETL Deployment - Windows.....	74
Pre-Migration Checklist.....	74
New Installation Instructions (for the First Time Only)	74
Upgrade Instructions.....	81
JasperReports Server Installation - Windows	84
Prerequisite	84
Installation Steps	85
Update the License File.....	86
Increase the JasperReports Session Timeout Period.....	86
Start/Stop the JasperReports Server	87
JasperReports Configuration with Keycloak.....	87
Steps to Debug Issues Related to Keycloak/JasperReports Server Integration	90
JasperReports Server Upgrade - Windows	92
Prerequisites.....	92
Plugin Installation Steps	92
Validation Steps	93
SMaRT AFIX JasperReports Deployment - Windows	94
Pre-Migration Checklist.....	94
Procedure Steps	95
Node.js Installation - Windows	98
SMaRT AFIX Application Installation Instructions - Windows	98
Prerequisites.....	98
Installation and Deployment of SMaRT AFIX, iQ, and VOMS	99
Install the Oracle Instant Client	99
Install Visual Studio 2013 Community Edition	99
Install Windows SDK v8.1.....	99
Install Redis	99
Install the Deployment Package.....	100
Configure the Application Dashboard and Quick Links	100

Configure the Reverse Proxy	100
General Product Configuration Options	100
SMaRT AFIX-Specific Configuration Options	101
Standalone SMaRT AFIX vs. Multiple Application Deployment	102
Run the Application.....	103
Start the Products upon Bootup.....	103
Keycloak Configuration for SMaRT AFIX.....	104
Import the Client and Roles	104
Configure the Client	104

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 <i>Keycloak Installation Guide for STC-Hosted Clients</i> or <i>Keycloak Installation Guide for Self-Hosted Clients</i> .	
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

Component	Server	Linux	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. Initially, the server is usually set up with the following:

- CentOS 7 Minimal (64-bit)
- PostgreSQL 9.4.7 (if a Postgres database will be hosted)
- MongoDB 2.6.11 (if a Mongo database will be hosted; required for STC | iQ)
- Node.js v4.3.x LTS (codename: Argon)
- ReactJS

General hardware requirements vary based on environment (dev/qa/uat/prod), but these are the general guidelines/specifications:

- 64-bit CPU (Xeon multi-core for production)
- 2GB RAM (4GB minimum for production)
- 50GB HD (100GB minimum for production)

These instructions 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
```

- Restart the sshd service:

```
sudo service sshd restart
```

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

```
sudo setenforce 0
```

- Edit the following file so *permissive* persists across reboots:

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

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

```
SELINUX=permissive
```

- Make sure the software is completely up to date:

```
sudo yum update
```

- 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.

- 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
```

- 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

To install the SAF database, navigate to the Application Components/AFIX/OXE/db/CDSI_V5169_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
```

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 Components/AFIX/SAF directory and execute this command:

```
$ cp forecaster-5.16.12.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.

Vaccine Family	Suppress Dose1 Forecast	Vaccination Grace Period	Minimum Display Age	Maximum Patient Age
MENINGOCOCCAL B, OMV	<input checked="" type="checkbox"/>	4 ▼		Dose 1
DTaP/DT/Tdap	<input type="checkbox"/>	4 ▼		Dose 1
HIB	<input type="checkbox"/>	4 ▼		Dose 1
POLIO	<input type="checkbox"/>	4 ▼		Dose 1
HEP-B 3 DOSE	<input type="checkbox"/>	4 ▼		Dose 1
MMR	<input type="checkbox"/>	4 ▼		Dose 1
VARICELLA	<input type="checkbox"/>	4 ▼		Dose 1
MENINGOCOCCAL	<input type="checkbox"/>	4 ▼		Dose 1
HEP-A	<input type="checkbox"/>	4 ▼		Dose 1
FLU	<input type="checkbox"/>	4 ▼		Dose 1
PNEUMO (PCV)	<input type="checkbox"/>	4 ▼		Dose 1
HEP-B 2 DOSE	<input type="checkbox"/>	4 ▼		Dose 1
MEASLES	<input checked="" type="checkbox"/>	4 ▼		Dose 1
MUMPS	<input checked="" type="checkbox"/>	4 ▼		Dose 1
RUBELLA	<input checked="" type="checkbox"/>	4 ▼		Dose 1
PNEUMO (PPSV)	<input type="checkbox"/>	4 ▼		Dose 1
ROTAVIRUS	<input type="checkbox"/>	4 ▼		Dose 1
HPV	<input type="checkbox"/>	4 ▼		Dose 1
HERPES ZOSTER	<input type="checkbox"/>	4 ▼		Dose 1

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](#) PHC Hub Database Initialization - Linux).

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 `Application Components/AFIX/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 `Application Components/AFIX/POSTGRES` directory and execute this command:


```
cd Application/ Components/AFIX/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 Application Components/AFIX/POSTGRES directory and execute this command:

```
cd Application/ Components/AFIX/POSTGRES/  
dos2unix flyway
```

6. State-specific eligibility: If your state uses one or more state-specific codes for VFC eligibility, you will need to make a modification to the Application Components/AFIX/POSTGRES/sql/OLAP/OLAP_V1_0_2__R1_Afix_OLAP_Initial_Data.sql file. Add the state-specific codes to the DIM_VFC_ELIGIBLE section as noted below. Be sure to increment the vfc_eligible_key value for each state-specific code you add. State-specific codes should be added exactly as they appear in your system database.

```
... --DB_script for DIM_VFC_ELIGIBLE----  
INSERT INTO olap.dim_vfc_eligible(vfc_eligible_key,  
vfc_eligible_code, created_ts)  
VALUES (1, 'Not VFC Eligible-Underinsured', current_timestamp),  
      (2, 'VFC Eligible - State Specific Eligibility',  
current_timestamp),  
      (3, 'Local-Specific Eligibility', current_timestamp),  
      (4, 'Not VFC Eligible', current_timestamp),  
      (5, 'American Indian/Alaska Native', current_timestamp),  
      (6, 'Uninsured', current_timestamp),  
      (7, 'Federally Qualified Health Center patient',  
current_timestamp),  
      (8, 'Medicaid', current_timestamp),  
--Add State Codes here----  
      (9, 'State Specific Code', current_timestamp),  
      (-1, 'Unknown', current_timestamp);  
...
```

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 Application Components/AFIX/POSTGRES directory and executing this command:

```
cd Application/ Components/AFIX/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 repair

./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 Application Components/AFIX/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 &&> $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 Application Components/AFIX/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

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-
V6.0.1-installer.zip
```

2. Place the installer into the installation directory:

```
/opt/TIBCOJaspersoft-JETL-Installer-20150908_1633-V6.0.1-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-V6.0.1-
installer.zip
```

6. Set the execute permission on the installer:

```
$ sudo chmod +x TIBCOJaspersoft-JETL-Installer-20150908_1633-V6.0.1-
linux64-installer.run
```

7. Execute the installer:

```
$ sudo ./TIBCOJaspersoft-JETL-Installer-20150908_1633-V6.0.1-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-6.0.1` 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.

Install the Apache Web Server and Subversion

1. Install the Apache Web Server:

```
$ sudo yum install httpd
```

2. Start the Apache Web Server:

```
$ sudo service httpd restart
```

3. Configure the Apache Web Server to start on boot:

```
$ sudo chkconfig httpd on
```

4. Install SVN/Subversion:

```
$ sudo yum install httpd mod_dav_svn subversion
```

5. Configure an SVN repository for all of the Talend projects:

```
$ sudo mkdir /var/www/svn  
$ sudo svnadmin create /var/www/svn/repo  
$ sudo touch /var/www/svn/passwdFile  
$ sudo htpasswd -m /var/www/svn/passwdFile tisadmin  
$ sudo chown -R apache.apache /var/www/svn
```

6. Configure an SVN repository for the Talend library:

```
$ sudo svnadmin create /var/www/svn/talendlibrary  
$ sudo chown -R apache.apache /var/www/svn
```

7. Configure the settings:

```
# sudo vim /etc/httpd/conf.d/subversion.conf  
<Location /repo>  
DAV svn  
SVNPath /var/www/svn/repo  
Require valid-user
```

```
AuthType Basic
AuthName "Subversion"
AuthUserFile /var/www/svn/passwdFile
</Location>
<Location /talendlibrary>
DAV svn
SVNPath /var/www/svn/talendlibrary
Require valid-user
AuthType Basic
AuthName "Subversion"
AuthUserFile /var/www/svn/passwdFile
</Location>
```

8. Restart the Apache Web Server:

```
$ sudo systemctl restart httpd
```

9. Special instructions for AWS instances: If ETL and SVN are configured on the same AWS instance, execute this command. ServerName is the name (IP address) of the AWS server. Example: iptables -t nat -A OUTPUT -p all -d 52.42.200.159 -j DNAT --to-destination 127.0.0.1

```
Syntax: iptables -t nat -A OUTPUT -p all -d ServerName -j DNAT --to-destination 127.0.0.1
```

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-6.0.1/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-6.0.1/cmdline
$ sudo ./stop_cmdline.sh
```

Command to start the service:

```
$ cd /opt/TIBCOJaspersoft-6.0.1/cmdline
```

```
$ sudo ./start_cmdline.sh
```

Start/Stop Jaspersoft ETL Job Server

Command to stop the service:

```
$ cd /opt/TIBCOJaspersoft-6.0.1/jobserver  
$ sudo ./stop_jobserver.sh
```

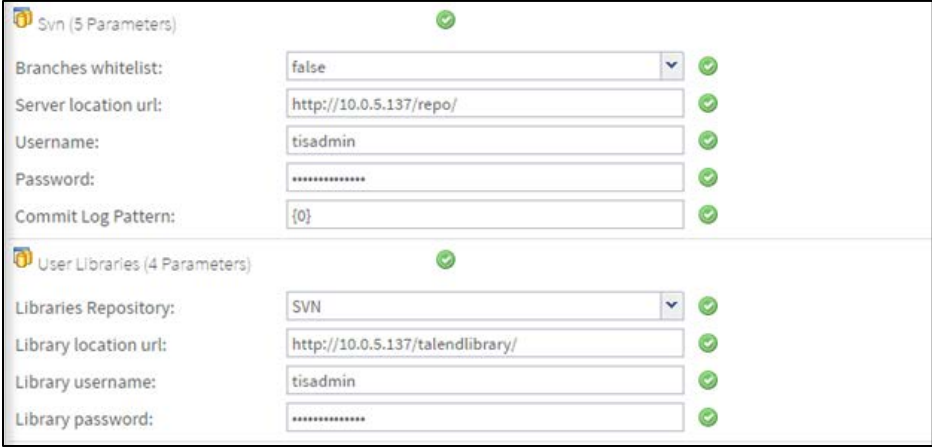
Command to start the service:

```
$ cd /opt/TIBCOJaspersoft-6.0.1/jobserver  
$ sudo ./start_jobserver.sh
```

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.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 **Configuration** on the left menu and update the SVN and library information.



The screenshot shows the configuration page for the Jaspersoft Admin Console. It is divided into two sections: 'Svn (5 Parameters)' and 'User Libraries (4 Parameters)'. Each section has a green checkmark icon in the top right corner, indicating that the settings are valid. The 'Svn' section includes fields for 'Branches whitelist' (set to 'false'), 'Server location url' (set to 'http://10.0.5.137/repo/'), 'Username' (set to 'tisadmin'), 'Password' (masked with '*****'), and 'Commit Log Pattern' (set to '{0}'). The 'User Libraries' section includes fields for 'Libraries Repository' (set to 'SVN'), 'Library location url' (set to 'http://10.0.5.137/talendlibrary/'), 'Library username' (set to 'tisadmin'), and 'Library password' (masked with '*****').

3. Navigate to **User Settings** on the left menu and update the username (if needed), SVN, and library information.

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

5. Navigate to **Project settings** from the left menu and add a new project by clicking the **Add** button.
6. Create two projects. Name the first one **InteropProject**, select the **Active** option, enter **Interop ETL** as the description and **admin admin** as the Author, and select **SVN** as the storage, then save it.

Project

Label:

Active:

Reference:

Description:

Author:

Storage: SVN None

Advanced settings

7. Name the second project **Interop_Reference**, select both the **Active** and **Reference** options, do not enter a description, use **admin admin** as the Author, and select **SVN** as the storage again, then save it.

Project

Label:

Active:

Reference:

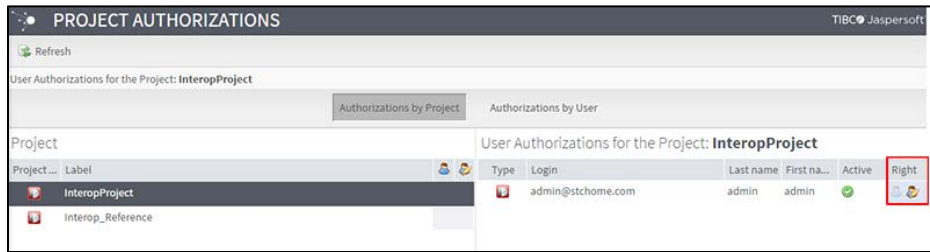
Description:

Author:

Storage: SVN None

Advanced settings

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



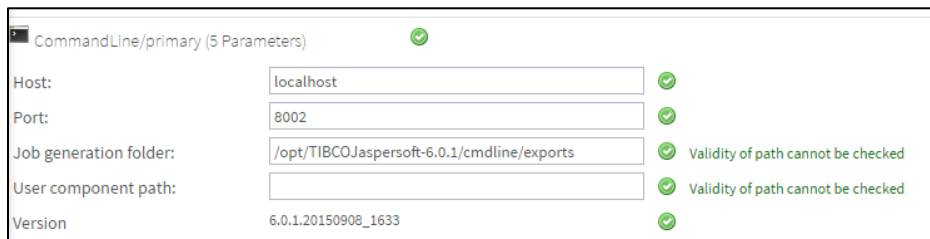
9. Create the following directories and files in the server:

```

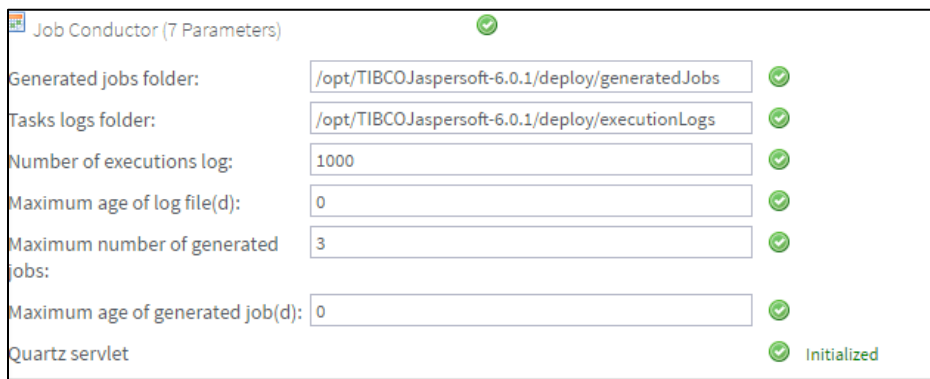
sudo mkdir /opt/TIBCOJaspersoft-6.0.1/deploy
sudo mkdir /opt/TIBCOJaspersoft-6.0.1/deploy/generatedJobs
sudo mkdir /opt/TIBCOJaspersoft-6.0.1/deploy/executionLogs
sudo mkdir /opt/TIBCOJaspersoft-6.0.1/cmdline/exports
sudo mkdir /opt/TIBCOJaspersoft-6.0.1/Audit
sudo mkdir /opt/TIBCOJaspersoft-6.0.1/Audit/reports
sudo mkdir /opt/TIBCOJaspersoft-6.0.1/joblogs
sudo touch /opt/TIBCOJaspersoft-6.0.1/joblogs/log

```

10. In Primary Command Line configuration, set the path for the Job generation folder to `/opt/TIBCOJaspersoft-6.0.1/cmdline/exports`



11. In Job Conductor configuration, set the path for the Generated jobs folder to `/opt/TIBCOJaspersoft-6.0.1/deploy/generatedJobs` and the path to the Tasks logs folder to `/opt/TIBCOJaspersof-6.0.1/deploy/executionLogs`



12. In Logging configuration, set the Technical log file path to `/opt/TIBCOJaspersoft-6.0.1/joblogs/log`

Logging (4 Parameters / 1 errors)

Technical log file path: ✓

Technical log threshold: ✓

Business log file path: ✓

13. In Audit configuration, set the Reports stored path to `/opt/TIBCOJaspersoft-6.0.1/Audit/reports`

Audit (6 Parameters) ✓

Url: ✓

User: ✓

Password: ✓


Driver: ✓

Web Console: ✓

Reports stored path: ✓

14. 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:	<input type="text" value="qa0-etl"/>
Description:	<input type="text"/>
Host:	<input type="text" value="10.0.5.137"/>
Time zone:	<input type="text"/> 
Command port:	<input type="text" value="8000"/>
File transfer port:	<input type="text" value="8001"/>
Monitoring port:	<input type="text" value="8888"/>
Timeout on unknown state (s):	<input type="text" value="120"/>
Username:	<input type="text" value="admin@stchome.com"/>
Password:	<input type="password" value="....."/>
Use SSL:	<input type="checkbox"/>
Active:	<input checked="" type="checkbox"/>
<input type="checkbox"/> Jaspersoft Runtime	

15. Log in via SSH to the Unix server using the ETL server owner.
16. Copy the `etl.zip` file (from `Application Components/AFIX/ETL`) to a location convenient for you on the ETL Jaspersoft server to unzip the file, such as a home directory.
17. 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`
18. Set up SMTP in Configuration only if the user needs an email notification on failure of jobs. Use the following settings:
 - Use SMTP > False
 - Host: company's email SMTP server
 - Port: host's port number
 - Username: host's username
 - Password: user's password

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 `Application Components/AFIX/ETL` into your local directory:
 - `API_Vaildation_Sequential_Execution_0.1`
 - `etl`
 - `etl_201703`
 - `Forecaster_Sequential_Execution_0.1`
 - `IWEB_Staging_Sequential_Execution_0.1`
 - `ODS_Sequential_Execution_0.1`
 - `OLAP_Sequential_Execution_0.1`
 - `AFIX_Assessment_Benchmark_0.1`
- Identify whether the installation is an upgrade from prior versions or a fresh installation from scratch.

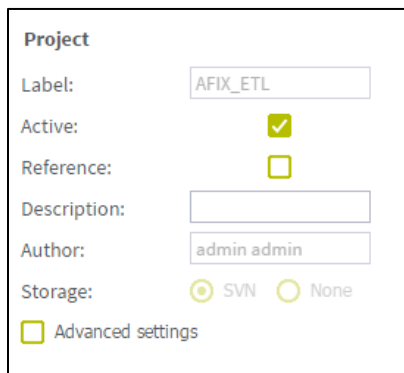
New Installation Instructions (for the First Time Only)

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:

Active:

Reference:

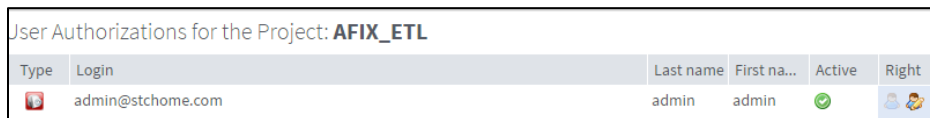
Description:

Author:




Storage: SVN None

Advanced settings

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




User 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 sever 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:	<input type="text" value="qa0-etl"/>
Description:	<input type="text"/>
Host:	<input type="text" value="10.0.5.137"/>
Time zone:	<input type="text"/> 
Command port:	<input type="text" value="8000"/>
File transfer port:	<input type="text" value="8001"/>
Monitoring port:	<input type="text" value="8888"/>
Timeout on unknown state (s):	<input type="text" value="120"/>
Username:	<input type="text" value="admin@stchome.com"/>
Password:	<input type="password" value="....."/>
Use SSL:	<input type="checkbox"/>
Active:	<input checked="" type="checkbox"/>
<input type="checkbox"/> Jaspersoft Runtime	

6. Log in via SSH to the Unix server using the ETL server owner.
7. Copy the `etl.zip` file (from `Application Components/AFIX/ETL`) to a location convenient for you on the ETL Jaspersoft server to unzip the file, such as a home directory.
8. 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
3. Navigate to the Job Conductor page.
4. Select **Add > Normal Task** and add the following: Label = AFIX IWEB API Execution; Description = AFIX IWEB API Execution.
5. 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.
6. Click **Save**.
7. Select **Add > Normal Task** and add the following: Label = AFIX IWEB API Validation; Description = AFIX IWEB API Validation.
8. 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.
9. Click **Save**.
10. Select **Add > Normal Task** and add the following: Label = AFIX Stage 2 ODS Load; Description = AFIX Stage 2 ODS Load.
11. 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.
12. Click **Save**.
13. Select **Add > Normal Task** and add the following: Label = AFIX Forecast Execution; Description = AFIX Forecast Execution.
14. 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.

15. Click **Save**.
16. Select **Add > Normal Task** and add the following: Label = AFIX OLAP Execution; Description = AFIX OLAP Execution.
17. 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.
18. Click **Save**.
19. Select **Add > Normal Task** and add the following: Label = AFIX Assessment Benchmark; Description = AFIX Assessment Benchmark.
20. 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.
21. Click **Save**.
22. 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

23. Save when finished.

24. Initial load instructions:

- Go to the task **AssessmentBenchmark** and 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



- 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



- 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



- 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. 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. 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:

25. 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

26. 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_201703.zip` file (located at `Application Components/AFIX/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_201703.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`
3. Navigate to the Job Conductor page.
4. Select **Add > Normal Task** and add the following: Label = AFIX Assessment Benchmark; Description = AFIX Assessment Benchmark.
5. Click **Import Generated Jobs** and select the `AFIX_Assesment_Benchmark.zip` file downloaded as part of the prerequisites. Add the following: Execution server = `AFIX_JobServer`; On Unavailable JobServer = Recover Task.
6. Click **Save**.
7. Navigate to the task **AFIX IWeb API Execution**.
8. 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.

9. Click **Save**.
10. Navigate to the task **AFIX IWeb API Validation**.
11. 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.
12. Click **Save**.
13. Navigate to the task **AFIX Stage 2 ODS Load**.
14. 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.
15. Click **Save**.
16. Navigate to the task **AFIX Forecast Execution**.
17. 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.
18. Click **Save**.
19. Navigate to the task **AFIX OLAP Execution**.
20. 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.
21. Click **Save**.
22. 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.
23. Save when finished.
24. 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
25. 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.

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.

Prerequisite

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
- jasperreports-server-6.2.1-windows-64bit-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
```


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>
```

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

See Keycloak Client Setup.

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: <code>https://sso-dev.stchome.com/</code>
[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 <i>Dev-Integration</i> : <code>https://sso-dev.stchome.com/auth/realms/Dev-Integration</code>
[JASPERSERVER_URL]	<property name="redirecturl"> <property name="logoutUrl">	URL of the Jasper server. Example: <code>http://<server_name>:8080/jasperserver-pro/oauth</code>
[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.OAuthUserDetailsServiceImpl"> <property name="state"> <value>WA</value> </property> </bean>
Jaspersoft Roles	<bean id="mtExternalUserSetupProcessor" class="com.jaspersoft.jasper	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

Key	XML Tag Name(s) to Update	Value
	<pre> server.multipleTenancy.security.externalAuth.processors.MTEExternalUserSetupProcessor" 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> </pre>	<p>match with the client role defined at the Keycloak end.</p> <p>Currently, the following roles are configured:</p> <ol style="list-style-type: none"> 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.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 Jasper server 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/web/api/v1/OrgFac/

Response format ? application/json

Save Cancel

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/web/api/v1/OrgFac/?user=user1

GET POST PUT DELETE Other methods

Raw headers Headers form Headers sets

API-KEY: 43434
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.

```
/Application Components/AFIX/JRS/webservices_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.

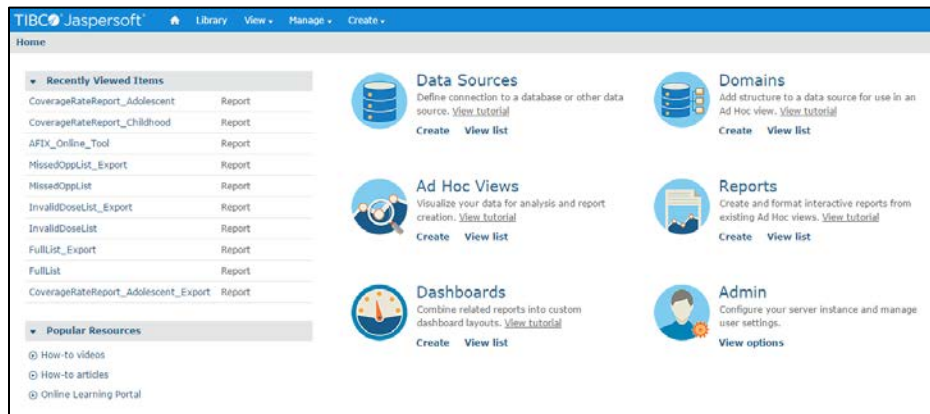
```
cd /$WS_DS/JRS/WEB-INF/  
cp applicationContext-WebServiceDataSource.xml /$JRS_HOME/WEB-INF/
```

5. Restart the web server running JasperReports server.

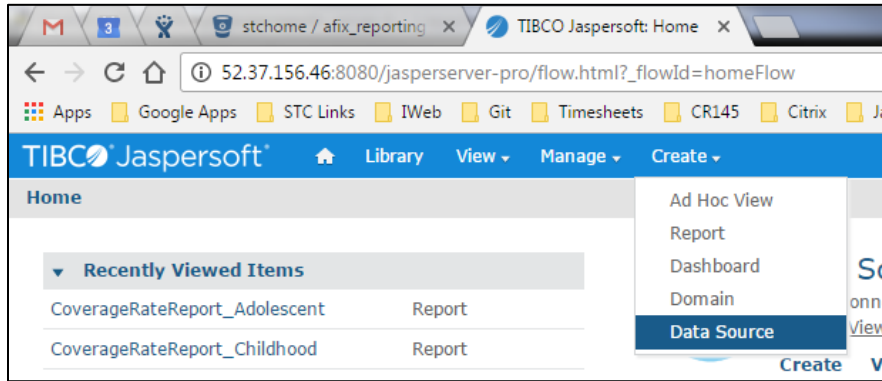
Validation Steps

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

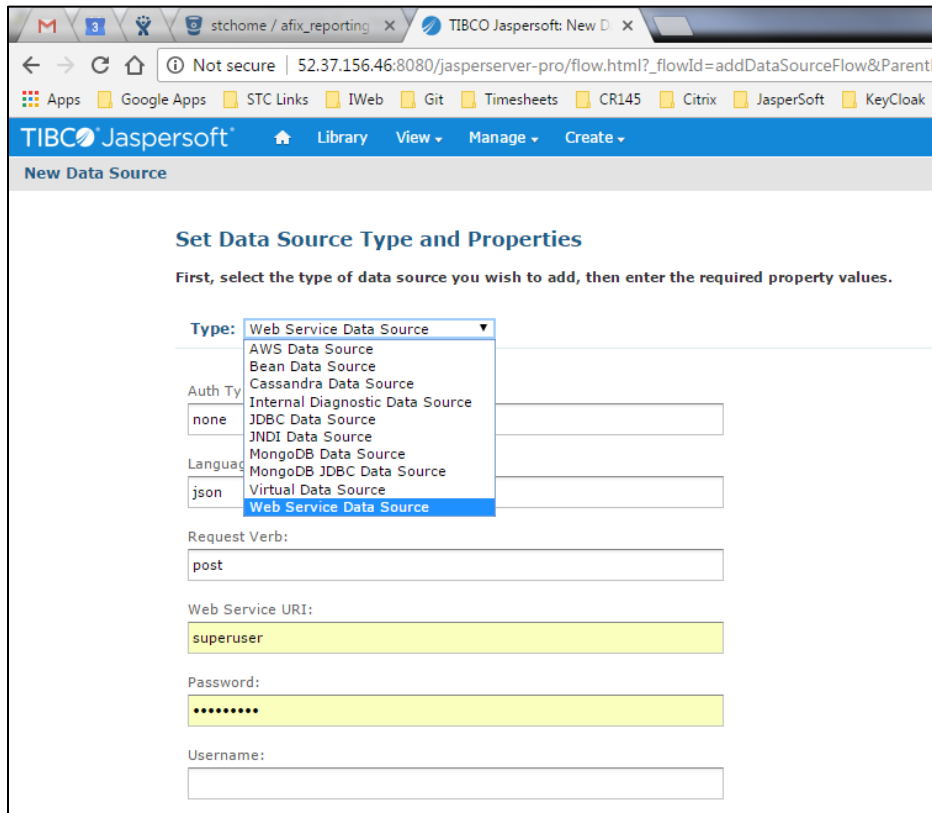
1. Navigate to your JasperReports server admin console.
2. Log in as an admin-level user.



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



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



SMaRT AFIX JasperReports Deployment - Linux

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

Pre-Migration Checklist

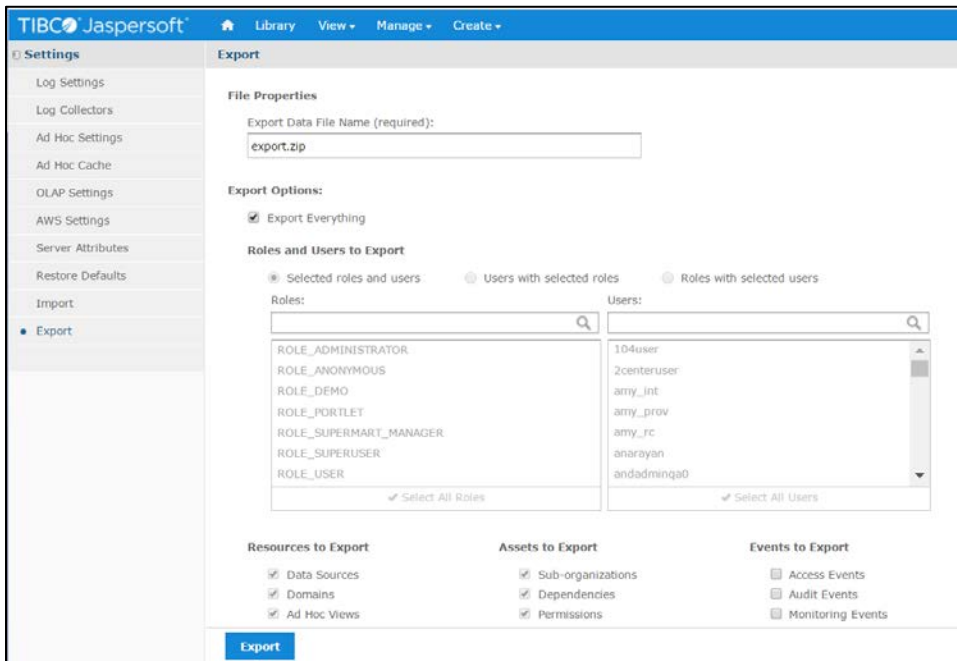
- Confirm that the Postgres database is installed and configured
- Retrieve the passwords for the SMaRT AFIX database (*afixdb*)
- Retrieve the Jaspersoft superuser username and password
- Confirm that the SMaRT AFIX Online Tool web service is installed
- Confirm that all necessary ports are open across all firewalls (the default port is 80)
- Confirm that the release distribution zip file for Jasper Reports is extracted from `Application Components/JRS/AFIX/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.

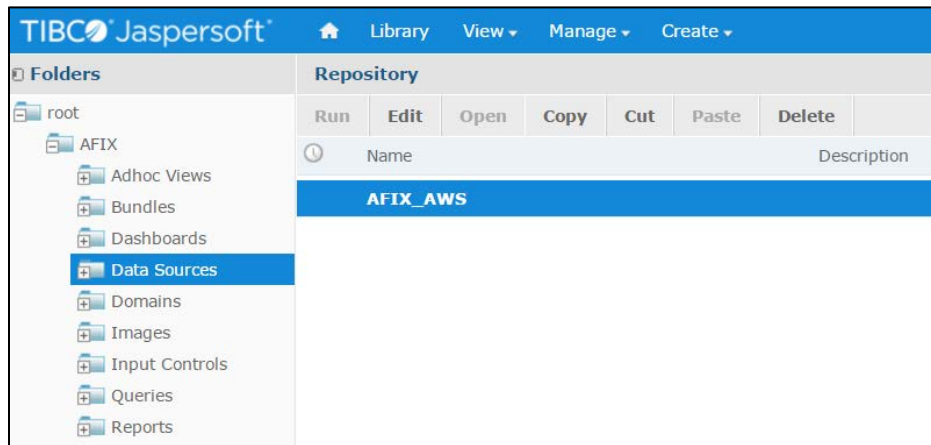


Import the Content via the Portal

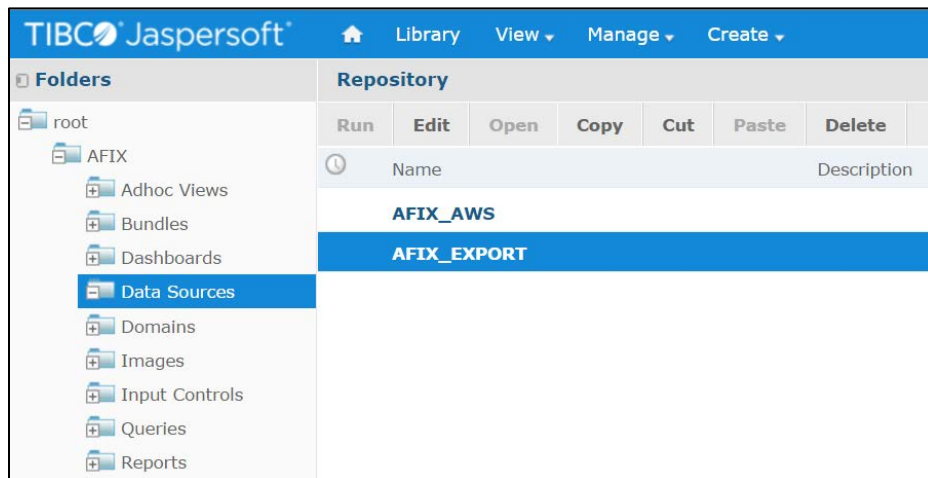
1. Log in to the Jaspersoft BI Portal as the administrator user.
2. Navigate to **Server Settings > Import**.
3. Import the AFIX_Reports_[version number].zip file.
4. Update the data source with the database connection information.

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.

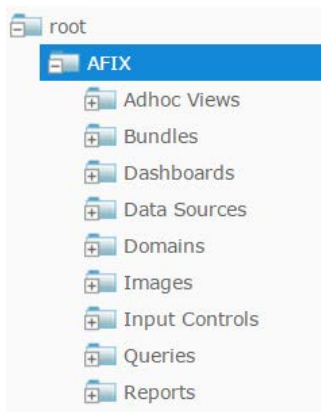


6. Leave the defaults for Auth Type, Language, and Request Verb. Update the web service URI to point to your SMaRT AFIX Online Tool web service deployment. Note that even though the Password field is populated, no password is present or required; leave it as it is. Click **Save** when finished.

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.

NOTE: There is a new application build for the AFIX 1.17.5.02 patch. It can be found under `/afix-1.17.5.02/Application Components/AFIX/`

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 <i>localhost</i>).
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 <i>true</i> .
ENABLE_SSL	When set to <i>true</i> , this enables HTTPS support. If REVERSE_PROXY is set to <i>true</i> , this should be set to <i>false</i> 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.

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)))",
}
},
```

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
```

Windows Instructions

The following instructions are for the Windows operating system.

Forecaster Installation Instructions - Windows

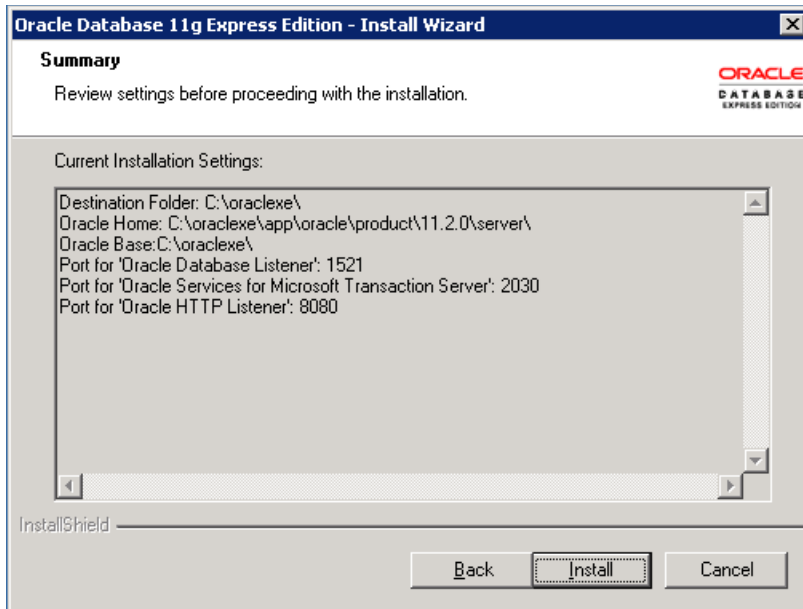
The installation of the Standalone Forecaster (SAF) is composed of three major tasks: install Oracle Express, install the SAF database into Oracle Express, and install the Forecaster web application into Tomcat. All three components are installed on the *Service* machine (SVC).

SAF can be installed to Oracle 11g or Oracle Express 11g. The Oracle product you choose will depend on your state's licensing. Oracle Express 11g is free to use for this type of installation. All performance testing, etc., for SAF has been conducted using Oracle Express. These instructions only cover the installation of Oracle Express 11g.

Install Oracle Express

1. If necessary, create an Oracle account at <http://www.oracle.com>.
2. If using Oracle Express (free), download the appropriate version. The Windows version is approximately 312MB in size.
3. Unzip the downloaded file and run the `DISK1/setup.exe`. When using the installation wizard, the installer creates a database named XE. The installer uses these ports: TNS = 1521, MTS = 2030, HTTP = 8080.
4. There may be two error messages about a missing file named `KEY_XE.reg`. Just acknowledge them and continue with the installation.
5. Disable the HTTP port by running the following at a command prompt after the installation is complete:

```
echo EXEC DBMS_XDB.SETHTTPPORT(0); | sqlplus / as sysdba
```



Install the SAF Database

Database objects are created with 5.16.9 CDSi install version.

1. Navigate to the Application Components/AFIX/OXE/db/CDSI_V5169_Install directory.
2. Run Install.bat.
3. Read the ImmuCast_CDSi_readme.txt file to verify the current SAF version.

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>
```

- Copy the Application Components/AFIX/SAF/forecaster-5.16.12.war to webapps/forecaster.war under your Tomcat directory.
- Restart Tomcat.
- 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>
```

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

STC Forecasting Settings SOAP WSDL Tester 

Forecasting Settings

Vaccine Family Settings Save

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

Vaccine Preferences by Dose to Override ACIP Schedule Recommendations Save

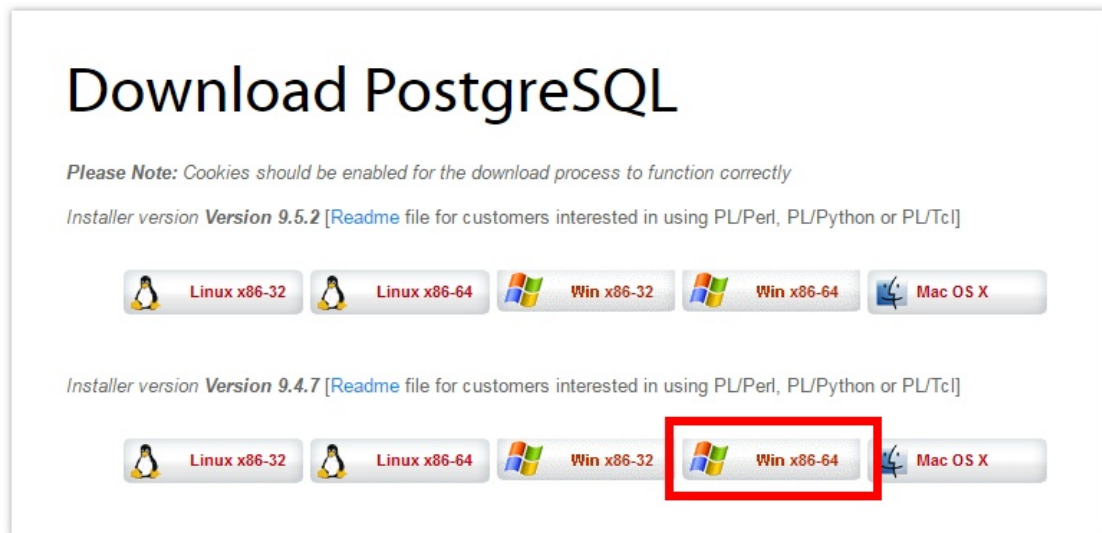
Postgres Database Installation - Windows

The information below describes how to install the Postgres database on Windows Server 2008 R2 64-bit, as well as how to modify relevant Postgres server configuration settings for SMaRT AFIX.

Install on Windows

The following information covers general installation instructions for installing PostgreSQL onto Windows. Windows Server 2008 R2 64-bit was used as a reference, but installation on other versions of Windows should be similar.

1. Visit [this page](#) and download the latest 64-bit version of Postgres. Version 9.4.x was used for this installation example.



2. Run the downloaded executable to start the installation. It is recommended to use the default installation directories, but they can be customized as needed (make note of the new locations).
3. Choose a password for the Postgres admin user *postgres* and store it securely for when administrative maintenance is required.
4. Use the default port 5432 (recommended). An alternate port can be used, but make sure to configure connecting applications to the Postgres database.
5. Use the default locale.
6. On the last step post installation, you can uncheck the *Stack Builder may be...* option and click **Finish**.

Modify Postgres Server Configuration Settings

Some of the Postgres server configuration settings must be modified. If you used the default installation directory, the following file reference locations should be located in the directory path:

```
C:\Program Files\PostgreSQL\9.4\data\
```

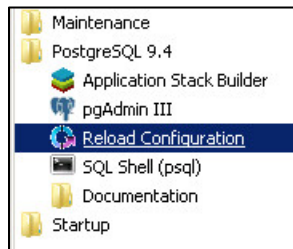
1. Edit the `pg_hba.conf` file to allow connections from other STC servers/applications. Copy/paste the following line:

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

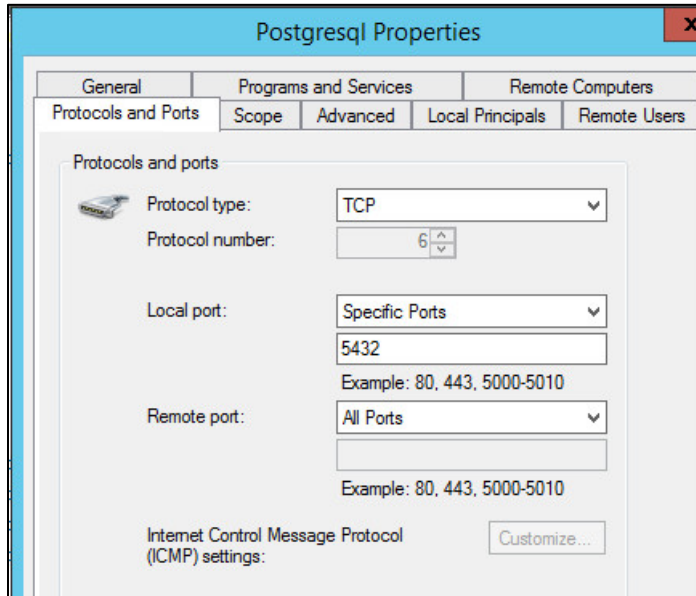
to make the file look something like the following. Let's assume your internal subnet falls under `192.168.x.x`, which would update the line to:

```
host    all        all        127.0.0.1/32    md5
host    all        all        192.168.0.0/16  md5
```

2. Save the changes. If you are technical and familiar enough with CIDR notation, you can further customize and restrict connection access as needed.
3. Trigger Postgres to reload with the new configuration changes by running **Reload Configuration**, found under PostgreSQL 9.4 in the Windows Start Menu:



4. Create a custom firewall rule to allow inbound connections on port 5432:



5. Complete the installation by initializing the database (see [SMaRT AFIX Database Initialization - Windows](#)).

SMaRT AFIX Database Initialization - Windows

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. State-specific eligibility: If your state uses one or more state-specific codes for VFC eligibility, you will need to make a modification to the `Application Components\AFIX\POSTGRES\sql\OLAP\OLAP_V1_0_2__R1_Afix_OLAP_Initial_Data.sql` file. Add the state-specific codes to the `DIM_VFC_ELIGIBLE` section as noted below. Be sure to increment the `vfc_eligible_key` value for each state-specific code you add. State-specific codes should be added exactly as they appear in your system database.

```
... --DB_script for DIM_VFC_ELIGIBLE----  
INSERT INTO olap.dim_vfc_eligible(vfc_eligible_key,  
vfc_eligible_code, created_ts)  
VALUES (1, 'Not VFC Eligible-Underinsured', current_timestamp),  
       (2, 'VFC Eligible - State Specific Eligibility',
```

```

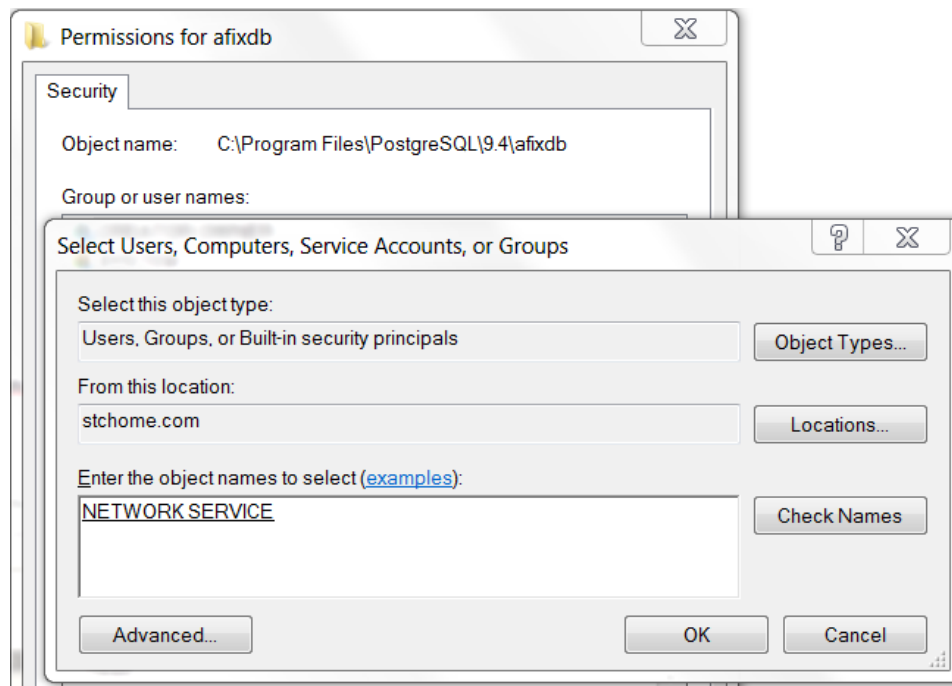
current_timestamp),
    (3, 'Local-Specific Eligibility', current_timestamp),
    (4, 'Not VFC Eligible', current_timestamp),
    (5, 'American Indian/Alaska Native', current_timestamp),
    (6, 'Uninsured', current_timestamp),
    (7, 'Federally Qualified Health Center patient',
current_timestamp),
    (8, 'Medicaid', current_timestamp),
--Add State Codes here----
    (9, 'State Specific Code', current_timestamp),
    (-1, 'Unknown', current_timestamp);
...

```

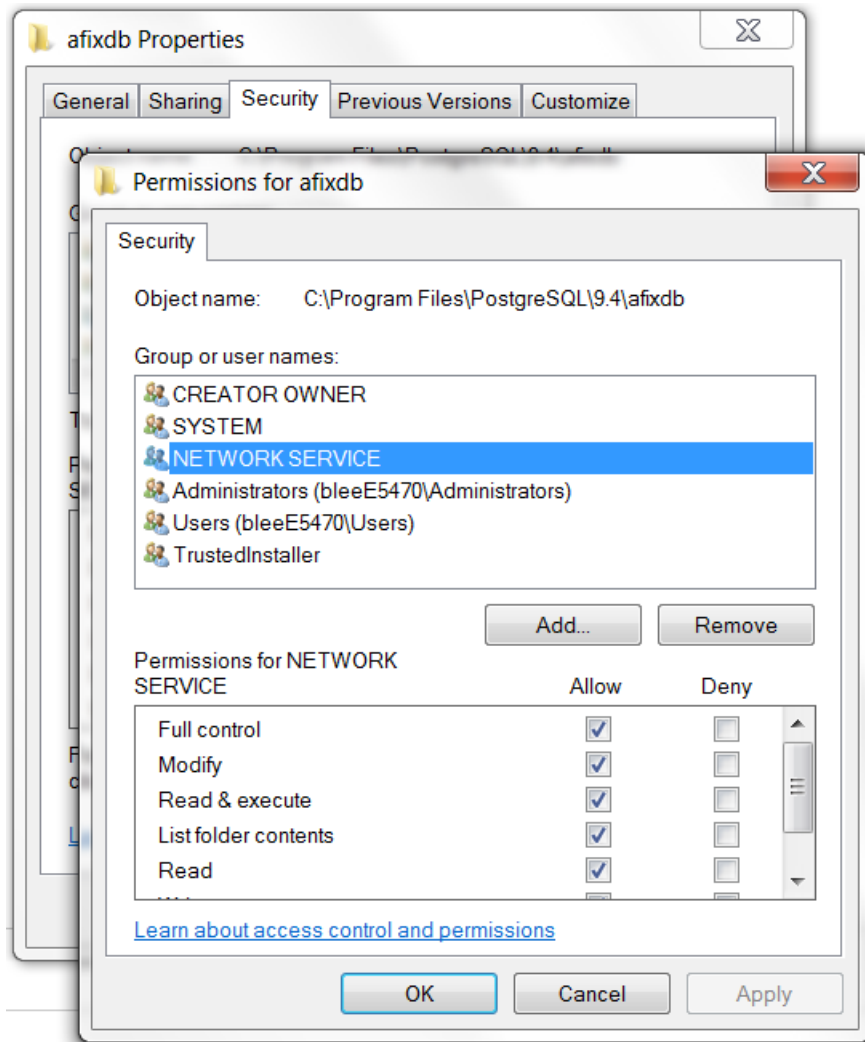
Initialize a New Database (for the First Time Only)

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. Create a tablespace folder - Use Windows Explorer to create the directory for the tablespace to be used by SMART AFIX. Navigate to the folder C:\Program Files\PostgreSQL\9.4, click the **New Folder** button, and name the new folder *afixdb*.
2. Set the folder permissions for *afixdb* - In Windows Explorer, right-click on the *afixdb* folder and select **Properties**. Click on the Security tab, then click the **Edit** button and then the **Add** button. Enter the name *network service* and click on the **Check Names** button.
3. Your window should like the image below, although your location will be different and local to your network. Click **OK**.



4. Make sure NETWORK SERVICE is selected and that the Full Control option under the Allow column is selected. When finished, click **OK** for all of the remaining open windows to close them.



5. Create the data directory:

```
C:\Program Files\PostgreSQL\9.4\afixdb\data
```

6. Create the olapdata directory:

```
C:\Program Files\PostgreSQL\9.4\afixdb\olapdata
```

7. Download the release distribution zip file and extract the database scripts for the database server in the afixdb folder:

```
Application Components\AFIX\POSTGRES
```

8. Create the afixdb by navigating to the Application Components/AFIX/POSTGRES directory and executing the following command. If prompted, enter the password that was entered during the Postgres installation.

```
"c:\Program Files\PostgreSQL\9.4\bin\psql.exe" -Upostgres -f  
setup_database_win.sql
```

9. Create the database users and roles by navigating to the Application Components/AFIX/POSTGRES directory and executing the following command. If prompted, enter the password that was entered during the Postgres installation.

```
"c:\Program Files\PostgreSQL\9.4\bin\psql.exe" -Upostgres -f  
setup_objects.sql afixdb
```

10. Create the APP configuration schema by navigating to the Application Components/AFIX/POSTGRES directory and executing the following command. Enter the password **AfixQA**.

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

11. Create the ODS schema by navigating to the Application Components/AFIX/POSTGRES directory and executing the following command. Enter the password **AfixQA**.

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

12. Create the OLAP schema by navigating to the Application Components/AFIX/POSTGRES directory and executing the following command. Enter the password **AfixQA**.

```
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. Migrate the APP configuration schema by navigating to the Application Components/AFIX/POSTGRES directory and executing the following command. Enter the password **AfixQA**.

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

2. Migrate the ODS schema by navigating to the Application Components/AFIX/POSTGRES directory and executing the following command. Enter the password **AfixQA**.

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

3. Migrate the OLAP schema by navigating to the Application Components/AFIX/POSTGRES directory and executing the following command. Enter the password **AfixQA**.

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

```
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 and partition.bat from the Application Components/AFIX/POSTGRES directory to C:\Program Files\PostgreSQL\9.4\afixdb.
2. Navigate to the Application Components/AFIX/POSTGRES directory and execute this command:

```
copy partition.sql C:\Program Files\PostgreSQL\9.4\afixdb  
copy partition.bat C:\Program Files\PostgreSQL\9.4\afixdb
```

3. Create a Task Scheduler to run the C:\Program Files\PostgreSQL\9.4\afixdb\partition.bat script daily at 1:00 a.m. to create future partitions and drop old partitions.

SMaRT AFIX Online Tool Web Service Installation - Windows

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 `Application Components/AFIX/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.
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.

Jaspersoft ETL Installation - Windows

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 variable `JAVA_HOME` has been set:

```
JAVA_HOME=[drive]:\directory\Java\1.8_91
```

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`
 - `VisualSVN-Server-3.5.6-x64`

Install Jaspersoft ETL

1. Download the Jaspersoft ETL installer zip file and the license file:

```
Server Components\ETL\TIBCOJaspersoft-JETL-Installer-20150908_1633-  
V6.0.1-installer.zip  
Server Components\ETL\license.txt
```

2. Copy the installer and license file into a temporary directory:

```
c:\user\[user name]\Downloads
```

3. Unzip the installer into the temporary directory. Several files will be extracted. Keep all of the files in the temporary directory for the duration of the installation.
4. From the temporary directory, execute the Windows installer by clicking on:

```
TIBCOJaspersoft-JETL-Installer-20150908_1633-V6.0.1-windows-  
installer.exe
```

5. Follow the prompts to accept the license agreement.
6. Follow the prompts to specify the installation directory, installation type, and license file. Specify [drive:]\Jaspersoft\6.0.1 for the installation directory, **Easy Install** for the installation type, and `license.txt` (found in the temporary directory).
7. Click the **Next** button and wait until the installation finishes.
8. Click the **Start** button, search for `services.msc`, and then search for any service named TIBCO Jaspersoft Logserver Collector 6.0.1. Stop this service and set it to **Manual**. If there is no service named this, ignore this step.

Install Subversion

1. Download the SVN installer: `Setup-Subversion-1.8.16.msi`.
2. Click on the installer to execute.
3. Follow the prompt for the security warning by selecting **Run**.
4. Click **Next** on the Welcome prompt.
5. Click **Next** on the Information prompt.
6. Use the default destination folder or specify a different folder, then click **Next**.
7. Use the default version on the Apache Version prompt, then click **Next**.
8. Click **Finish** on the Complete Install page.
9. Configure the SVN repository for all of the Talend projects and library:


```
svnadmin create "c:\projects"  
svnadmin create "c:\talendlibrary"
```

10. Configure the user settings - edit the c:\talendlibrary\conf\passwd file and add the following line:

```
admin@company.com = admin
```

11. Configure the server settings:

```
anon-access = none  
auth-access = write  
password-db = passwd
```

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 C:\Jaspersoft\6.0.1\jac  
$ stop_tac.bat
```

Command to start the service:

```
$ cd C:\Jaspersoft\6.0.1\jac  
$ start_tac.bat
```

Start/Stop Jaspersoft ETL Command Line

Command to stop the service:

```
$ cd C:\Jaspersoft\6.0.1\cmdline  
$ stop_cmdline.bat
```

Command to start the service:

```
$ cd C:\Jaspersoft\6.0.1\cmdline  
$ start_cmdline.bat
```

Start/Stop Jaspersoft ETL Job Server

Command to stop the service:

```
$ cd C:\Jaspersoft\6.0.1\jobserver  
$ stop_jobserver.bat
```

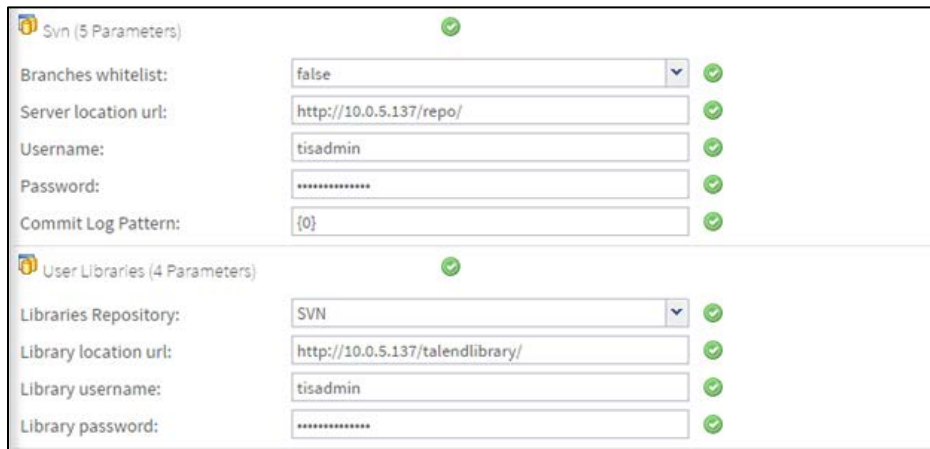
Command to start the service:

```
$$ cd C:\Jaspersoft\6.0.1\jobserver  
$ start_jobserver.bat
```

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.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 **Configuration** on the left menu and update the SVN and library information.



The screenshot displays the configuration interface for the Admin Console, divided into two sections: 'Svn (5 Parameters)' and 'User Libraries (4 Parameters)'. Each section has a green checkmark icon in the top right corner, indicating that the settings are valid. The 'Svn' section includes fields for 'Branches whitelist' (set to 'false'), 'Server location url' (set to 'http://10.0.5.137/repo/'), 'Username' (set to 'tisadmin'), 'Password' (masked with asterisks), and 'Commit Log Pattern' (set to '{0}'). The 'User Libraries' section includes fields for 'Libraries Repository' (set to 'SVN'), 'Library location url' (set to 'http://10.0.5.137/talendlibrary/'), 'Library username' (set to 'tisadmin'), and 'Library password' (masked with asterisks).

3. Navigate to **User Settings** on the left menu and update the username (if needed), SVN, and library information.

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

5. Navigate to **Project settings** from the left menu and add a new project by clicking the **Add** button.
6. Create two projects. Name the first one **InteropProject**, select the **Active** option, enter **Interop ETL** as the description and **admin admin** as the Author, and select **SVN** as the storage, then save it.

Project

Label:

Active:

Reference:

Description:

Author:

Storage: SVN None

Advanced settings

7. Name the second project **Interop_Reference**, select both the **Active** and **Reference** options, do not enter a description, use **admin admin** as the Author, and select **SVN** as the storage again, then save it.

Project

Label:

Active:

Reference:

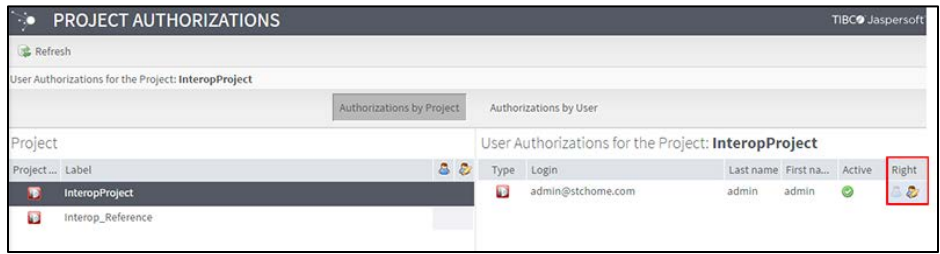
Description:

Author:

Storage: SVN None

Advanced settings

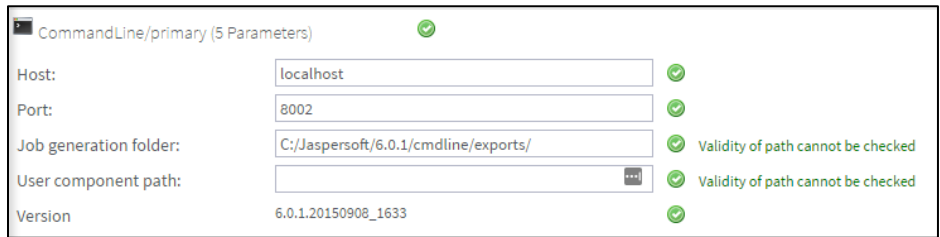
8. Navigate to **Project authorization** from the left menu. Click on each of the newly created projects and assign read/write permissions to the user.



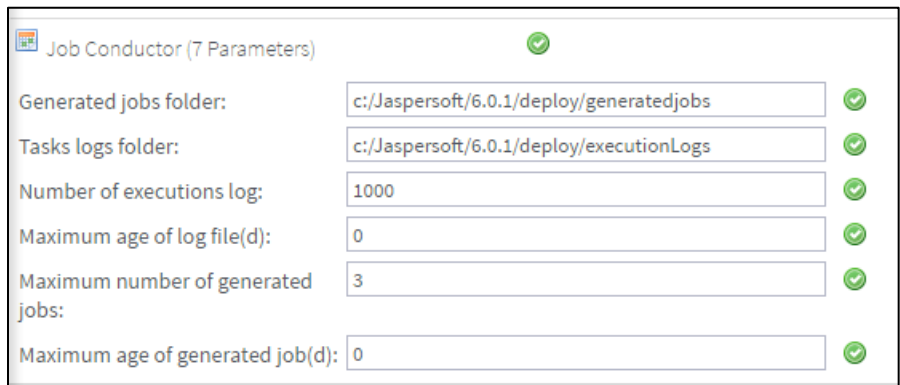
9. Create the following directories and files in the server:

```
C:\Jaspersoft\6.0.1\deploy
C:\Jaspersoft\6.0.1\deploy\generatedJobs
C:\Jaspersoft\6.0.1\deploy\executionLogs
C:\Jaspersoft\6.0.1\cmdline\exports
C:\Jaspersoft\6.0.1\Audit
C:\Jaspersoft\6.0.1\Audit\reports
C:\Jaspersoft\6.0.1\joblogs
C:\Jaspersoft\6.0.1\joblogs\log.txt
```


10. In Primary Command Line configuration, set the path for the Job generation folder to C:/Jaspersoft/6.0.1/cmdline/exports





11. In Job Conductor configuration, set the path for the Generated jobs folder to c:/Jaspersoft/6.0.1/deploy/generatedJobs and the path to the Tasks logs folder to c:/Jaspersoft/6.0.1/deploy/executionLogs




12. In Logging configuration, set the Technical log file path to c:/Jaspersoft/6.0.1/joblogs/log


Logging (4 Parameters / 1 errors) 


Technical log file path: 


Technical log threshold: 


Business log file path: 


13. In Audit configuration, set the Reports stored path to `c:/Jaspersoft/6.0.1/Audit/reports`


Audit (6 Parameters) 


Url: 

User: 

Password: 


Driver: 

Web Console: <http://iq-demo-dbb:8080/org.jaspersoft.administrator/h2console> 

Reports stored path: 

14. 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:	<input type="text" value="qa0-etl"/>
Description:	<input type="text"/>
Host:	<input type="text" value="10.0.5.137"/>
Time zone:	<input type="text"/> 
Command port:	<input type="text" value="8000"/>
File transfer port:	<input type="text" value="8001"/>
Monitoring port:	<input type="text" value="8888"/>
Timeout on unknown state (s):	<input type="text" value="120"/>
Username:	<input type="text" value="admin@stchome.com"/>
Password:	<input type="password" value="....."/>
Use SSL:	<input type="checkbox"/>
Active:	<input checked="" type="checkbox"/>
<input type="checkbox"/> Jaspersoft Runtime	

15. Log in via Remote Desktop to Windows Server using ETL server owner.
16. Copy the `etl.zip` file (from `Application Components/AFIX/ETL`) to a location convenient for you on the ETL Jaspersoft server to unzip the file, such as a home directory.
17. Unzip the `etl.zip` file to the server in `(C:\)` directory (Windows), which creates a directory structure similar to this:
 - `C:\etl`
 - `C:\etl\data`
 - `C:\etl\archive`
18. Set up SMTP in Configuration only if the user needs an email notification on failure of jobs. Use the following settings:
 - Use SMTP > False
 - Host: company's email SMTP server
 - Port: host's port number
 - Username: host's username
 - Password: user's password

SMTP (6 Parameters / 3 errors)	
Use SMTP:	true
Host:	
Port:	
Require SSL:	false
Username:	user@domain.com
Password:	*****

SMaRT AFIX Jaspersoft ETL Deployment - Windows

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 `Application Components/AFIX/ETL` into your local directory:
 - `API_Vaildation_Sequential_Execution_0.1`
 - `etl`
 - `etl_201703`
 - `Forecaster_Sequential_Execution_0.1`
 - `IWEB_Staging_Sequential_Execution_0.1`
 - `ODS_Sequential_Execution_0.1`
 - `OLAP_Sequential_Execution_0.1`
 - `AFIX_Assessment_Benchmark_0.1`
- Identify whether the installation is an upgrade from prior versions or a fresh installation from scratch.

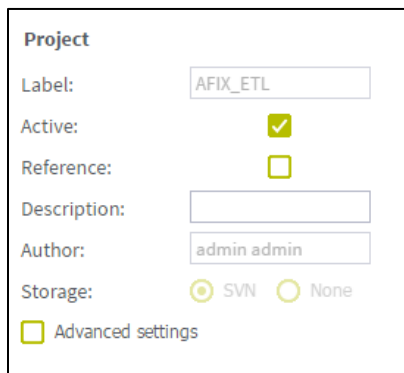
New Installation Instructions (for the First Time Only)

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

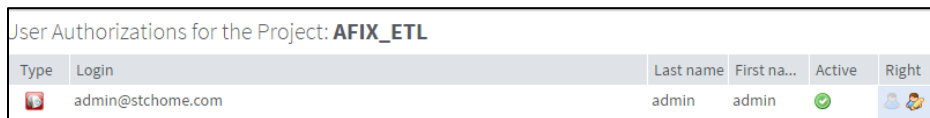
1. Navigate to `http://[host name]:8080/org.jaspersoft.administrator` and enter the default username and password (admin). For the username, use `admin@yourdomain.com` or replace it with an email group if the user needs email notifications on ETL job failure (example: `email_group@yourdomain.com`).
2.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






The screenshot shows a form titled "Project" with the following fields and options:

- Label:
- Active:
- Reference:
- Description:
- Author:
- Storage: SVN None
- Advanced settings

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



The screenshot shows a table titled "User Authorizations for the Project: AFIX_ETL" with the following data:

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 sever 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:

Description:

Host:

Time zone: 

Command port:

File transfer port:

Monitoring port:

Timeout on unknown state (s):

Username:

Password:

Use SSL:

Active:

Jaspersoft Runtime

6. Log in via SSH to the Unix server using the ETL server owner.
7. Copy the `etl.zip` file (from `Application Components/AFIX/ETL`) to a location convenient for you on the ETL Jaspersoft server to unzip the file, such as a home directory.
8. Unzip the `etl.zip` file to the server in the root directory (Unix), which creates a directory structure similar to this:
 - `c:\etl\afix`
 - `c:\etl\afix\api`
 - `c:\etl\afix\archive`
 - `c:\etl\afix\bulkload`
 - `c:\etl\afix\dataout`
 - `c:\etl\afix\input`
 - `c:\etl\afix\rejects`
 - `c:\etl\afix\tmp`
 - `c:\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
3. Navigate to the Job Conductor page.
4. Select **Add > Normal Task** and add the following: Label = AFIX IWEB API Execution; Description = AFIX IWEB API Execution.
5. 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.
6. Click **Save**.
7. Select **Add > Normal Task** and add the following: Label = AFIX IWEB API Validation; Description = AFIX IWEB API Validation.
8. 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.
9. Click **Save**.
10. Select **Add > Normal Task** and add the following: Label = AFIX Stage 2 ODS Load; Description = AFIX Stage 2 ODS Load.
11. 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.
12. Click **Save**.
13. Select **Add > Normal Task** and add the following: Label = AFIX Forecast Execution; Description = AFIX Forecast Execution.
14. 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.

15. Click **Save**.
16. Select **Add > Normal Task** and add the following: Label = AFIX OLAP Execution; Description = AFIX OLAP Execution.
17. 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.
18. Click **Save**.
19. Select **Add > Normal Task** and add the following: Label = AFIX Assessment Benchmark; Description = AFIX Assessment Benchmark.
20. 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.
21. Click **Save**.
22. 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

23. Save when finished.

24. Initial load instructions:

- Go to the task **AssessmentBenchmark** and 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



- 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



- 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



- 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. Add the JVM parameters -Xms1024M and -Xmx15690M

- 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. Add the JVM parameters - Xms1024M and Xmx15690M

25. 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

26. 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_201703.zip` file (located at `Application Components/AFIX/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_201703.zip` file to the server in the Unix root directory (Unix), which adds the following files that were created as part of this release:
 - `c:\etl\afix`
 - `c:\etl\afix\api`
 - `c:\etl\afix\archive`
 - `c:\etl\afix\bulkload`
 - `c:\etl\afix\dataout`
 - `c:\etl\afix\input`
 - `c:\etl\afix\rejects`
 - `c:\etl\afix\tmp`
 - `c:\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`
3. Navigate to the Job Conductor page.
4. Select **Add > Normal Task** and add the following: Label = AFIX Assessment Benchmark; Description = AFIX Assessment Benchmark.
5. Click **Import Generated Jobs** and select the `AFIX_Assesment_Benchmark.zip` file downloaded as part of the prerequisites. Add the following: Execution server = `AFIX_JobServer`; On Unavailable JobServer = Recover Task.
6. Click **Save**.
7. Navigate to the task **AFIX IWeb API Execution**.
8. 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.

9. Click **Save**.
10. Navigate to the task **AFIX IWeb API Validation**.
11. 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.
12. Click **Save**.
13. Navigate to the task **AFIX Stage 2 ODS Load**.
14. 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.
15. Click **Save**.
16. Navigate to the task **AFIX Forecast Execution**.
17. 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.
18. Click **Save**.
19. Navigate to the task **AFIX OLAP Execution**.
20. 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.
21. Click **Save**.
22. 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.
23. Save when finished.
24. 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
25. 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.

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 - Windows

The following instructions explain how to install the JasperReports Server on Windows.

Prerequisite

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
- jasperreports-server-6.2.1-windows-64bit-installer.exe
- jasperserver.license

- LatoFont
- stc-jaspersoft-oauth-0.0.1-SNAPSHOT

Installation Steps

1. Download the JasperReports Server for Windows and the license file from the release distribution:

```
Server Components\JRS\jasperreports-server-6.2.1-windows-64bit-  
installer.exe  
Server Components\JRS\jasperserver.license
```

2. Run the JasperReports Server Installer by right-clicking on the file and selecting **Run as administrator**:

```
jasperreports-server-6.2.1-windows-64bit-installer.exe
```

3. Follow the prompts to accept the license agreement.
4. Select the **Install All Components and Samples** option.
5. Follow the prompts to specify the installation folder. Specify the following directory:

```
C:\Jaspersoft\jasperreports-server-6.2
```

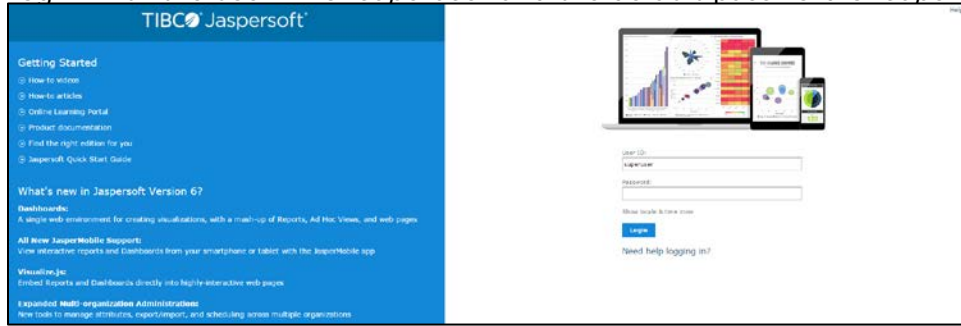
6. On completion of the installation, start the JasperReports server by selecting the following:

```
Launch JasperReports Server Now (for bundled Tomcat and PostgreSQL only)  
If you're installing on Linux, don't close the terminal window running  
the start script.
```

7. 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
```

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



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/desktop.
2. Replace the license file in the JasperReports installation directory (`C:\Jaspersoft\jasperreports-server-6.2\`) 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 `C:\Jaspersoft\jasperreports-server-6.2\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>
</jsp-config>
</session-timeout>

```

- Restart the JasperReports server (see [Start/Stop the JasperReports Server](#) below).

Start/Stop the JasperReports Server

To start or stop the JasperReports server, click the Start button, go to **All Programs > JasperReports Server > Start or Stop Services**, and then select either **Start Service** or **Stop Service**.

JasperReports Configuration with Keycloak

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

Prerequisite

See Keycloak Client Setup.

JasperReports Server Modifications

- 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>	URL of the Keycloak server. Example: https://sso-dev.stchome.com/

Key	XML Tag Name(s) to Update	Value
	<pre><userdetails_location> <logoutUrl></pre>	
[KEYCLOAK_REALM]	<pre><authorization_location> <token_location> <userdetails_location> <logoutUrl></pre>	<p>Keycloak realm that the client belongs to. This value needs to be changed, along with <keycloak_URL>.</p> <p>In this example, the realm name is <i>Dev-Integration</i>: <code>https://sso-dev.stchome.com/auth/realms/Dev-Integration</code></p>
[JASPERSERVER_URL]	<pre><property name="redirecturl"> <property name="logoutUrl"></pre>	<p>URL of the Jasper server. Example: <code>http://<server_name>:8080/jasperserver-pro/oauth</code></p>
[CLIENT_ID]	<pre><property name="client_id"></pre>	<p>Client ID as set on Keycloak. Example: <code><property name="client_id"> <value>jasper-aws-dev</value></code></p>
[CLIENT_SECRET]	<pre><property name="clientsecret"> <property name="userdetails_secret"></pre>	<p>Credentials secret key generated by Keycloak.</p>
State_ID	<pre><property name="state"></pre>	<p>This is the global setting used to populate the STATE_ID attribute in Jaspersoft. Example: <code><bean id="oAuthUserDetailsService" class="com.jaspersoft.jasperserver.ps.OAuth.OAuthUserDetailsServiceImpl"> <property name="state"> <value>WA</value> </property> </bean></code></p>
Jaspersoft Roles	<pre><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 --></pre>	<p>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.</p> <p>Currently, the following roles are configured:</p> <ol style="list-style-type: none"> 1. Administrator 2. Superuser 3. User Role

Key	XML Tag Name(s) to Update	Value
	<pre> <entry> <key> <value>ROLE_ADMIN_EX TERNAL_ORGANIZATION </value> </key> <value>ROLE_ADMINIST RATOR</value> </entry> </map> </pre>	

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.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. The [sso_link] differs based on the environment; an example might be something like `sso-dev.stchome.com`.

```
java -cp .\ InstallCert [sso_link]
keytool -exportcert -alias [sso_link]-1 -keystore jssecacerts -
storepass changeit -file [sso_link].cer
keytool -importcert -alias [sso_link] -keystore
../lib/security/cacerts -storepass changeit -file [sso_link].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: `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 Jasper server 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/web/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.



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 - Windows

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.

```
..\Application Components\AFIX\JRS\webservices_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.

```
C:\Jaspersoft\jasperreports-server-6.2\apache-tomcat\webapps\jasperserver-pro\
```

Plugin Installation Steps

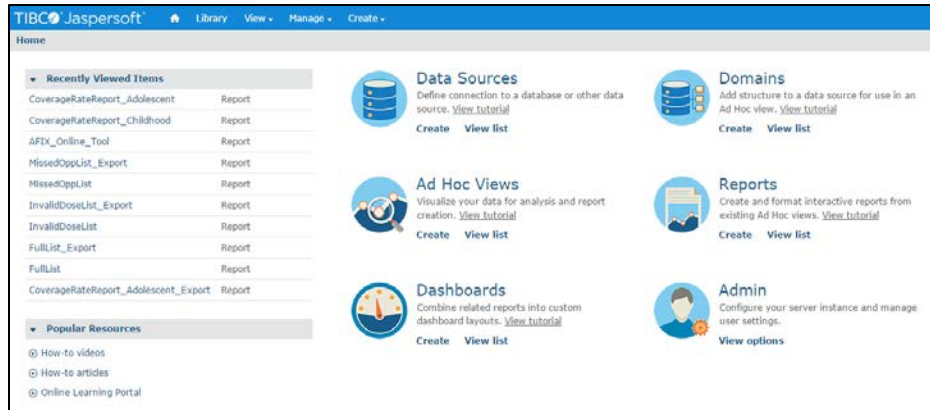
The installation should be performed as root user.

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.
3. Restart the JasperReports server.

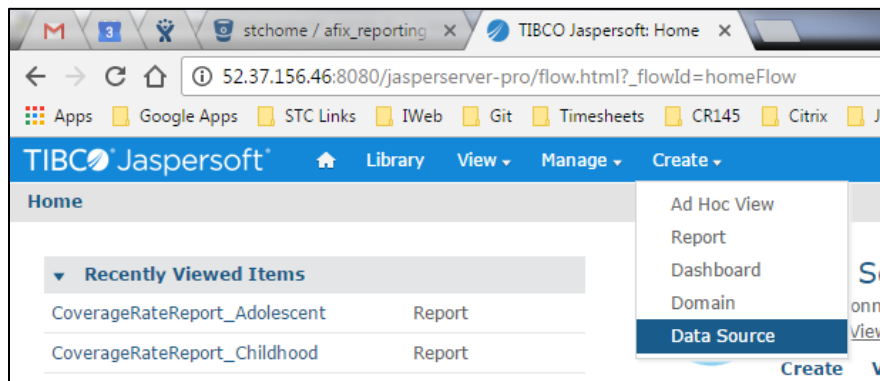
Validation Steps

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.



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



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

The screenshot shows the 'New Data Source' configuration page in TIBCO JasperSoft. The 'Type' dropdown menu is open, and 'Web Service Data Source' is selected. The 'Auth Ty' field is set to 'none', 'Language' is 'json', 'Request Verb' is 'post', 'Web Service URI' is 'superuser', 'Password' is masked with dots, and 'Username' is empty.

SMaRT AFIX JasperReports Deployment - Windows

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 `Application Components/JRS/AFIX/AFIX_Reports_[version number].zip` and copied to your local drive.

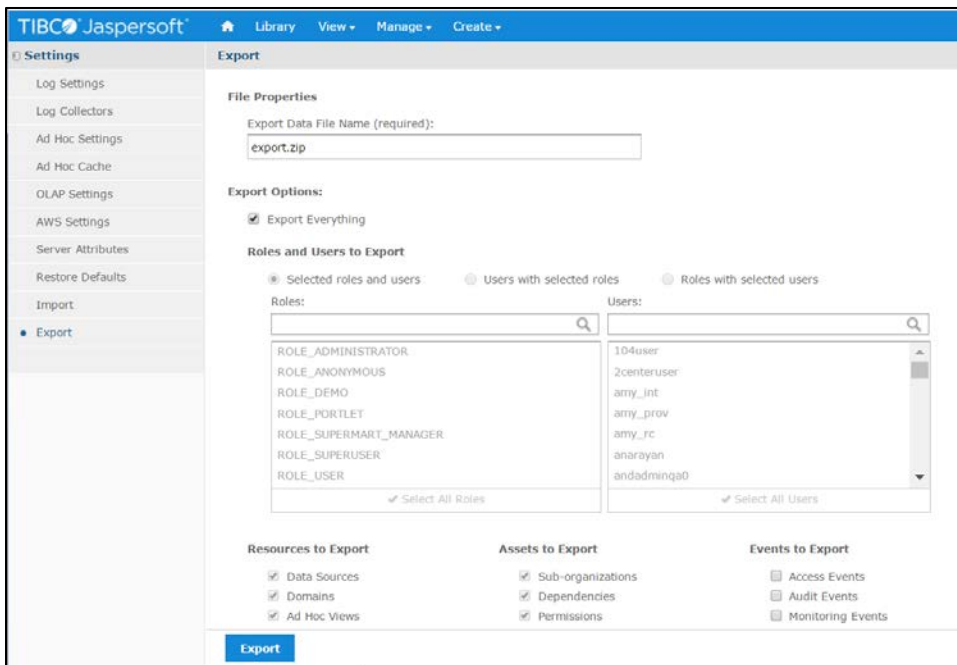
- **Important:** Be sure to update the JasperReports server by following the instructions in [SMaRT AFIX JasperReports Server Upgrade - Windows](#).

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.



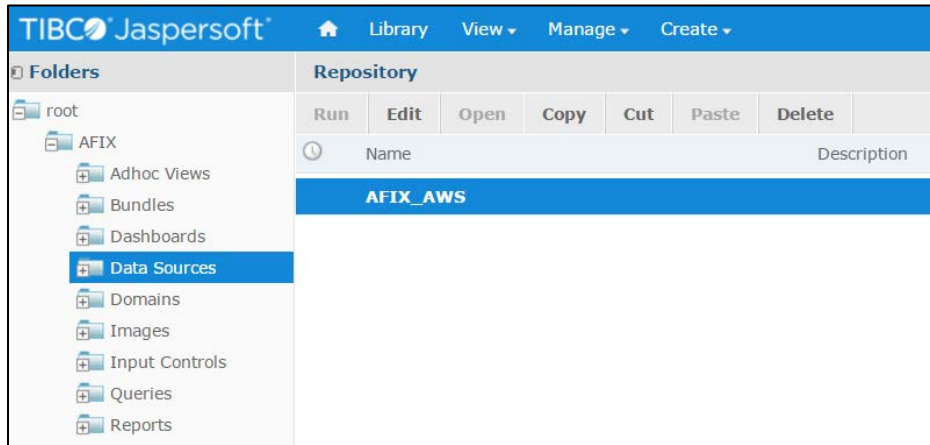
Import the Content via the Portal

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.

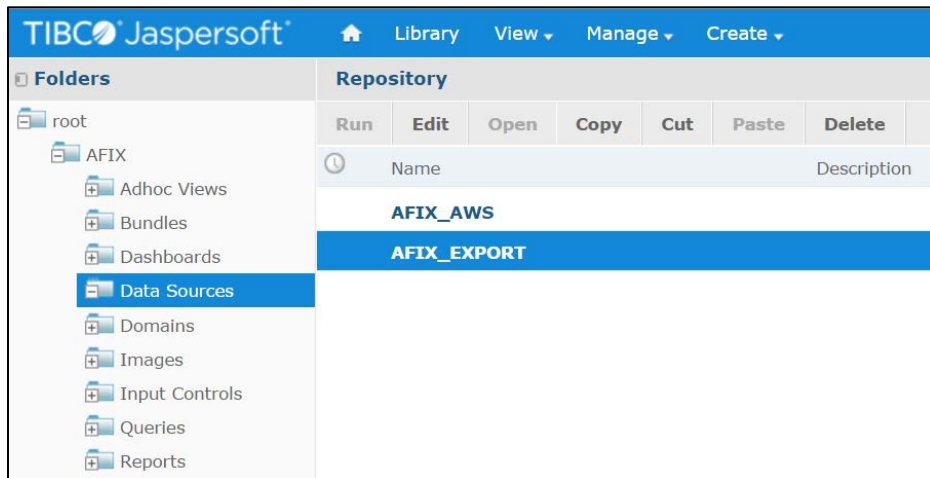
Import the Content via the Command Line

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

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



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



- 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:

Auth Type:

Language:

Request Verb:

Web Service URI:

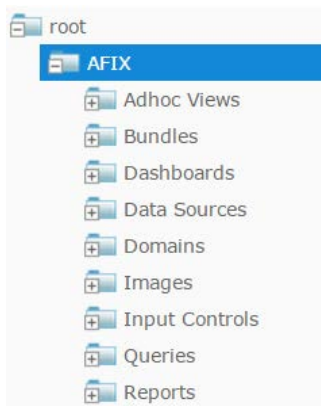
Password:

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 - Windows

Node.js installation is not supported in Windows.

SMaRT AFIX Application Installation Instructions - Windows

The following instructions explain how to install the application, the Oracle Instant Client, Visual Studio 2013 Community Edition, Windows SDK v8.1, 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:

- Windows Server 2012
- 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, Visual Studio, Windows SDK, 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 Oracle Instant Client Basic and SDK for Windows. Extract Instant Client Basic somewhere on the disk (i.e., C:\oracle\instantclient_12_1), then extract Instant Client SDK into the same folder.
2. Add the path you extracted Oracle Instant Client into to the PATH environment variable. This can be accessed via the Advanced System Settings window.
3. Add the following system variables. (See <https://github.com/oracle/node-oracledb/blob/master/INSTALL.md#instwin> and <https://community.oracle.com/docs/DOC-931127> for more information.)
 - OCI_LIB_DIR to C:\oracle\instantclient_12_1\sdk\lib\msvc
 - OCI_INC_DIR to C:\oracle\instantclient_12_1\sdk\include

Install Visual Studio 2013 Community Edition

See <https://www.visualstudio.com/en-us/news/releasenotes/vs2013-community-vs> for download and installation instructions.

Install Windows SDK v8.1

See <https://developer.microsoft.com/en-us/windows/downloads/windows-8-1-sdk> for download and installation instructions.

Install Redis

Use this command:

```
msiexec /i Redis-x64-3.2.100.msi
```

See <https://github.com/MicrosoftArchive/redis> and <https://github.com/MicrosoftArchive/redis/releases/tag/win-3.2.100> for more information.

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 <i>localhost</i>).
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 <i>true</i> .
ENABLE_SSL	When set to <i>true</i> , this enables HTTPS support. If REVERSE_PROXY is set to <i>true</i> , this should be set to <i>false</i> 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.

Option	Description
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.

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)))",
```

```
},  
}
```

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:

```
npm install -g pm2-windows-service  
pm2-service-install [-n service-name]  
pm2 save
```

Keycloak Configuration for SMaRT AFIX

This task is to be performed during SSO installation of the application. The table below lists each application and component, along with the JSON file name, client name, and URL placeholder name for each. Make sure to use the correct file and other information corresponding to the specific application/component installation.

Application Name	JSON File Name	Client Name	URL Placeholder Name
IWeb	iweb.json	iweb	IWEB-URL
PHC-Hub	phchub.json	phc-hub	PHC-HUB-URL
Jasper Report Server	jasper.json	jasper	JASPER-URL
iQ	interop.json	interop	INTEROP-URL
VOMS	voms.json	voms	VOMS-URL
Logviewer	interop-logviewer.json	interop-logviewer	LOGVIEWER-URL
Dashboard	dashboard.json	dashboard	DASHBOARD-URL

Import the Client and Roles

These steps explain how to import the client and configure the roles.

Prerequisites

The location of the JSON file for the application being installed.

Import Steps

1. Log in to Keycloak.
2. Select the desired realm for the application suite in the top left drop-down list.
3. Click **Import** on the left side menu.
4. Click the **Select File** button and browse to the SMaRT AFIX release Application Components/AFIX/SSO/JSON directory.
5. Select the `afix.json` file and click **Open**.
6. **Important:** In the *If a resource exists* drop-down list, select **Skip**.

Configure the Client

1. On the left side menu, click **Clients**.
2. Locate the client named *afix* you just imported (see above) and click **Edit**.
3. On the Settings tab, enter the URL to the application everywhere there is a URL placeholder.

4. If the Client Protocol is SAML, expand the Fine Grain SAML Endpoint Configuration section at the bottom of the page and make sure the URL placeholders are replaced with the application URL.
5. Click the **Save** button at the bottom.
6. Click on the Mappers tab.
7. Locate the *user-type* mapper and click the **Edit** button.
8. In the API-URI box, replace the URL placeholder with the URL to your IWeb installation.
9. Verify the API-KEY is set to the API key of your IWeb instance.
10. Click the **Save** button at the bottom.