

STC | iQ

Installation Guide

v1.17.5



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



Copyrights and Trademarks

© 2017 by Scientific Technologies Corporation (STC). All rights reserved.

This documentation describes the following: STC|iQ 1.17.5 installation

No part of this publication may be altered, reproduced, transmitted, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, or otherwise, without the prior written permission of the copyright owner.

This document, along with any embedded media, is provided for informational purposes only. Scientific Technologies Corporation (STC) provides this material "as is" and does not assert that this document is error free. The information contained herein may not necessarily accurately represent the current or planned functions of the application, and may be subject to significant and frequent modification. Scientific Technologies Corporation (STC) may change this document, any embedded content, and the product described herein at any time. Any changes will be incorporated in new versions of this document.

Scientific Technologies Corporation (STC) recognizes the rights of the holders of all trademarks used in its publications.

This document may provide hyperlinks to third-party websites or access to third-party content. Links and access to third-party sites are provided for your convenience only. Scientific Technologies Corporation does not control, endorse, or guarantee third-party content and is not responsible for any content, associated links, resources or services associated with a third-party site. Scientific Technologies Corporation shall also not be liable for any loss or damage associated with your use of any third-party content. (20170922)

Table of Contents

STC iQ Installation Instructions.....	5
Components	5
New Application Installations	6
Linux Instructions	8
Build a Linux (CentOs) Server	8
Install CentOS 7 Minimal and Configure	8
Postgres Database Installation - Linux.....	9
Install on Linux CentOS 7	9
Modify Postgres Server Configuration Settings	10
iQ Database Initialization - Linux	11
Prerequisites.....	11
Initialization Steps.....	12
MongoDB Installation - Linux	13
Install MongoDB on Linux (CentOS 7).....	13
Make MongoDB Server Configuration Changes for Interop (iQ)	13
ActiveMQ Installation - Linux	15
Prerequisites.....	15
Installation Steps	15
Verify Installation	16
Autostart.....	17
ActiveMQ Credential Encryption - Linux	18
Encrypt Credentials for Web Admin User	18
Encrypt Password for JMS Connection.....	19
JMS Consumer Installation - Linux	23
Prerequisites.....	23
Installation Steps	23
Stop Consumer	25
Autostart.....	26
Jaspersoft ETL Installation - Linux	26
Prerequisites.....	26
Install Jaspersoft ETL	27
Install the Apache Web Server and Subversion	28
Start/Stop Jaspersoft ETL Services.....	29
Configure the Admin Console	30
STC iQ Jaspersoft ETL Deployment - Linux.....	36

Pre-Migration Checklist.....	36
Deploy the ETL Components	36
JasperReports Server Installation - Linux.....	42
Prerequisite	42
Installation Steps	42
Update the License File.....	44
Increase the JasperReports Session Timeout Period.....	44
Start/Stop the JasperReports Server	45
JasperReports Configuration with Keycloak.....	45
Steps to Debug Issues Related to Keycloak/JasperReports Server Integration	48
JasperReports Migration for STC iQ.....	50
STC iQ JasperReports Deployment - Linux	50
Pre-Migration Checklist.....	50
Procedure Steps	50
Node.js Installation - Linux.....	53
HL7 Transaction Log Viewer Installation - Linux	53
Extract the Files	54
Configure Process.json	54
Start Process.json	54
Set Up the Proxy	54
STC iQ Application Installation Instructions - Linux	54
Prerequisites.....	54
Installation and Deployment of iQ, SMaRT AFIX and VOMS	55
Install the Oracle Instant Client	55
Install Redis	55
Install the Deployment Package.....	56
Configure the Application Dashboard and Quick Links	56
Configure the Reverse Proxy	56
General Product Configuration Options	57
Run the Application.....	57
Start the Products upon Bootup.....	58
Windows Instructions	59
Postgres Database Installation - Windows	59
Install on Windows	59
Modify Postgres Server Configuration Settings	60
iQ Database Initialization - Windows	61
Prerequisites.....	61
Steps.....	61

MongoDB Installation - Windows	64
Make MongoDB Server Configuration Changes for Interop (iQ)	65
ActiveMQ Installation - Windows	66
Prerequisites	67
Steps	67
Verify Installation	68
Stop ActiveMQ	68
Create the Service	68
ActiveMQ Credential Encryption - Windows	69
Encrypt Credentials for Web Admin User	69
Encrypt Password for JMS Connection	70
JMS Consumer Installation - Windows	71
Prerequisites	71
Installation Steps (Basic)	72
Installation Steps (Advanced)	73
Jaspersoft ETL Installation - Windows	73
Prerequisites	73
Install Jaspersoft ETL	74
Install Subversion	74
Start/Stop Jaspersoft ETL Services	75
Configure the Admin Console	76
STC iQ Jaspersoft ETL Deployment - Windows	82
Pre-Migration Checklist	82
Deploy the ETL Components	82
JasperReports Server Installation - Windows	88
Prerequisite	88
Installation Steps	88
Update the License File	89
Increase the JasperReports Session Timeout Period	90
Start/Stop the JasperReports Server	90
JasperReports Configuration with Keycloak	90
Steps to Debug Issues Related to Keycloak/JasperReports Server Integration	93
JasperReports Migration for STC iQ	95
STC iQ JasperReports Deployment - Windows	95
Pre-Migration Checklist	95
Procedure Steps	95
STC iQ Application Installation Instructions - Windows	97
Prerequisites	98

Installation and Deployment of iQ, SMaRT AFIX, and VOMS	98
Install the Oracle Instant Client	98
Install Visual Studio 2013 Community Edition	98
Install Windows SDK v8.1	98
Install Redis	99
Install the Deployment Package.....	99
Configure the Application Dashboard and Quick Links	99
Configure the Reverse Proxy	100
General Product Configuration Options	100
Run the Application.....	101
Start the Products upon Bootup.....	101
STC iQ Application Configuration	102
STC iQ Web Server-Specific Options	102
STC iQ API-Specific Options.....	102
Organization/Facility Micro-Service-Specific Options	102
Keycloak Configuration for STC iQ.....	103
Import the Client and Roles	103
Configure the Client	103

STC | iQ Installation Instructions

The following provides an overview of the process for installing the STC | iQ application (also referred to as *iQ*).

Components

iQ is composed of several server components. The following includes their function and a brief technical description.

Apache ActiveMQ

ActiveMQ is an open-source message broker. ActiveMQ provides asynchronous message communication. PHC Hub sends transaction messages to iQ via ActiveMQ.

STC Consumer

This is a Java process that is responsible for processing the messages sent by PHC Hub to ActiveMQ. As messages are processed, they are parsed and inserted into the iQ ODS database. As noted above, messages in and out of ActiveMQ are asynchronous. That is, PHC Hub can send messages to ActiveMQ whether or not the consumer process is running.

PostgreSQL

iQ utilizes PostgreSQL 9.4 as a database. The name of the database is *interop*. It is created via several `.sql` scripts.

JasperReports Server (JRS)

JRS is a web application that provides the reports for iQ. These reports are imported into JRS via the JRS Management Console. See [iQ JasperReports Deployment - Linux](#) or [Windows](#) for installation instructions.

Jaspersoft ETL (JETL)

Transaction data asynchronously streams into iQ via a Message Queue (JMS). A nightly ETL (Extract, Load, Transform) batch job processes the data into a form optimized for BI (Business Intelligence).

iQ Application

The UI and application server logic are provided via the iQ application (formal name: STC | iQ). 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 iQ 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 iQ 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> .	
PostgreSQL Server (if not already installed)	Database and Batch (DBB)	Postgres Database Installation - Linux	Postgres Database Installation Windows
iQ Database	Database and Batch (DBB)	iQ Database Initialization - Linux	iQ Database Initialization - Windows
MongoDB Server	Database and Batch (DBB)	MongoDB Installation - Linux	MongoDB Installation - Windows
ActiveMQ Server	Service (SVC)	ActiveMQ Installation - Linux (and then ActiveMQ Credential Encryption - Linux)	ActiveMQ Installation - Windows (and then ActiveMQ Credential Encryption - Windows)
JMS Consumer	Service (SVC)	JMS Consumer Installation - Linux	JMS Consumer Installation - Windows
Jaspersoft ETL Server	Database and Batch (DBB)	Jaspersoft ETL Installation - Linux	Jaspersoft ETL Installation - Windows
iQ ETL	Database and Batch (DBB)	iQ Jaspersoft ETL Deployment - Linux	iQ Jaspersoft ETL Deployment - Windows
JasperReports Server	Application (APP)	JasperReports Server Installation - Linux	JasperReports Server Installation - Windows
iQ Reports	Application (APP)	iQ JasperReports Deployment - Linux	iQ JasperReports Deployment - Windows
Node.js Server (if not already installed)	Application (APP)	Node.js Installation - Linux	Not supported
Transaction Log Viewer Application	Application (APP)	HL7 Transaction Log Viewer Installation - Linux	Not supported
iQ Application Installation	Application (APP)	iQ Application Installation Instructions - Linux	iQ Application Installation Instructions - Windows
iQ Application Configuration	Application (APP)	iQ Application Configuration	

Component	Server	Linux	Windows
Keycloak Configuration for iQ	Service (SVC)	Keycloak Configuration for iQ	

Linux Instructions

The following instructions are for the Linux operating system.

Build a Linux (CentOs) Server

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

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

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

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

Install CentOS 7 Minimal and Configure

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

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

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

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

```
PermitRootLogin no
```

4. Restart the `sshd` service:

```
sudo service sshd restart
```

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

```
sudo setenforce 0
```

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

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

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

```
SELINUX=permissive
```

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

```
sudo yum update
```

9. Finally, install some basic packages:

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

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 iQ (Interop).

Install on Linux CentOS 7

Currently, iQ 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 iQ. 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 [iQ Database Initialization - Linux](#)).

iQ Database Initialization - Linux

The information below describes how to initialize the iQ database on Linux.

Prerequisites

1. Confirm that PostgreSQL has been installed.
2. Download the release distribution zip file from `Application Components/IQ/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/IQ/POSTGRES` directory and execute this command:

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

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

Initialization Steps

1. Act as user *postgres*:

```
sudo su - postgres
```

2. Prepare the tablespace by creating the directory for the tablespace to be used by Interopdb:

```
mkdir /var/lib/pgsql/9.4/interopdb  
mkdir /var/lib/pgsql/9.4/interopdb/data  
mkdir /var/lib/pgsql/9.4/interopdb/idx
```

3. From the downloaded release distribution zipped file, extract the database scripts for the database server:

```
Application Components/IQ/POSTGRES
```

4. Add postgres to the path:

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

5. Create the interopdb by navigating to the Application Components/IQ/POSTGRES directory and executing the following command:

```
cd Application Components/IQ/POSTGRES  
psql -U postgres -f setup_database.sql
```

6. Create the database users and roles by navigating to the Application Components/IQ/POSTGRES directory and executing the following command:

```
psql -U postgres -f setup_objects.sql
```

7. Create the ETL config schema by navigating to the Application Components/IQ/POSTGRES directory downloaded as part of the prerequisites and execute the following command:

```
./flyway.sh -url=jdbc:postgresql://localhost:5432/interopdb -  
schemas=etlconfig -sqlMigrationPrefix=ETL_V -user=interop migrate
```

8. Create the OLAP schema by navigating to the Application Components/IQ/POSTGRES directory downloaded as part of the prerequisites and execute the following command:

```
./flyway.sh -url=jdbc:postgresql://localhost:5432/interopdb -  
schemas=interopolap -sqlMigrationPrefix=OLAP_V -user=interop migrate
```

9. Create the ODS schema by navigating to the Application Components/IQ/POSTGRES directory downloaded as part of the prerequisites and execute the following command:

```
./flyway.sh -url=jdbc:postgresql://localhost:5432/interopdb -schemas=ods  
-sqlMigrationPrefix=ODS_V -user=interop migrate
```

MongoDB Installation - Linux

The following instructions describe how to install MongoDB on Linux CentOS 7 and how to set the MongoDB server configuration changes for Interop (iQ).

Install MongoDB on Linux (CentOS 7)

Interop (iQ) is currently using CentOS 7 as its Linux distribution. CentOS 7 also currently has MongoDB 2.6.11 in its repository.

Before continuing, make sure you have followed the sever preparation steps in [Building a Linux \(CentOS\) Server](#).

1. Trigger a packages update:

```
sudo yum update
```

2. Install the MongoD 2.6 server/packages:

```
sudo yum install -y mongodb-server mongodb
```

3. Set MongoD to run as a service and start MongoD:

```
sudo systemctl enable mongod  
sudo systemctl start mongod
```

Make MongoDB Server Configuration Changes for Interop (iQ)

1. Obtain the Mongo distribution files.

Copy the following two files and one folder from the Interop package to the Linux server. Make sure to run the following steps in the same folder where the files are placed. (Note:

If you are trying to copy the files from a Windows machine, you can use a tool such as [psftp.exe](#) to transfer the files.)

- `stc-mongo-admin-db.js`
- `stc-mongo-stc-db.js`
- `MONGO_STC_DB`

2. Set the passwords for the MongoDB users.

First, create the *superuser* account. By default, there is no password set (`pwd: ""`). Edit `stc-mongo-admin-db.js` and set a unique password, then save it and keep the file nearby. The file should look like this, but with the new password ("MySUpassword"):

```
db.createUser(
{
  user: "superuser",
  pwd: "MySUpassword",
  roles: [ "root" ]
})
```

Next, set the password for the STC account that will be used by the Interop portal application (iQ). Edit `stc-mongo-stc-db.js` and set a unique password, using the same process as above. This password should be different from the superuser account, however.

NOTE: The *stcadmin* user and password you selected will be needed during installation of the Interop portal application (iQ). It will be entered on the `./production/deploy.sh` command.

3. Apply the MongoDB scripts.

With MongoDB now installed, the server running and the passwords configured, you can run the client interface by entering the commands at the Linux prompt:

```
mongo admin stc-mongo-admin-db.js
mongorestore --host localhost --db stc --drop MONGO_STC_DB
mongo stc stc-mongo-stc-db.js
```

4. Edit/Update `/etc/mongod.conf`.

Using an editor, edit the `mongod` configuration file; `vi` is used here.

```
sudo vi /etc/mongod.conf
```

With the default recommendation, the services that will use MongoDB reside on the same server. If you have a different configuration, you must change the `bind_ip = 127.0.0.1` line. Examples:

```
# this would allow any IP to connect
```



```
bind_ip = 0.0.0.0
-or-
# this would allow the localhost as well as the interface IP address 1.2.3.4
(meaning the assigned IP address of this server machine running MongoDB)
bind_ip = 127.0.0.1,1.2.3.4
```

Remove the # from the line #auth to the following (around line 70 in the file):

```
...
# Run with/without security (without by default)
auth = true
#noauth = true
...
```

Now save the file and exit back to the Linux prompt. Restart the MongoDB sever with the following:

```
sudo systemctl restart mongod
```

MongodB for Interop (iQ) is now set up and complete.

ActiveMQ Installation - Linux

The following information describes the prerequisites and steps for installing ActiveMQ, plus how to verify the installation and set up the autostart. All actions are performed on Server 2 (Message Queue).

Prerequisites

The Java Runtime (1.7 or higher) JRE must be installed.

Installation Steps

1. If not done already, download the ActiveMQ distribution and configuration files:

```
Server Components/AMQ/apache-activemq-5.13.0-bin.tar.gz
Server Components/AMQ/activemq.xml
Server Components/AMQ/jetty-realm.properties
Server Components/AMQ/credentials-enc.properties
Server Components/AMQ/webconsole-embedded.xml
```

2. Place ActiveMQ into the temporary directory: /tmp
3. Unzip the ActiveMQ distribution file:

```
$ cd /tmp
$ gunzip /tmp/apache-activemq-5.13.0-bin.tar.gz
$ tar xvf /tmp/apache-activemq-5.13.0-bin.tar
```

4. Copy the ActiveMQ distribution file to the installation directory:

```
$ sudo mv apache-activemq-5.13.0 /usr/local/
```

5. Create a symbolic link:

```
$ sudo ln -sf /usr/local/apache-activemq-5.13.0 /usr/local/activemq
```

6. Create the *activemq* user:

```
$ sudo adduser activemq -s /bin/bash
```

7. Set a password (enter the password at the prompt):

```
$ sudo passwd activemq
```

8. Modify `/home/activemq/.bashrc` for the *activemq* user, adding `JAVA_HOME` and `PATH` to the Java executable (the actual path may be different on your system):

```
$ sudo vi /home/activemq/.bashrc

# add your JAVA_HOME location
export JAVA_HOME=/usr/lib/jvm/jre-1.8.0

# add java to your path
export PATH=$JAVA_HOME/bin:$PATH
```

9. Copy the configuration files:

```
$ sudo cp /tmp/Server Components/AMQ/activemq.xml
/usr/local/activemq/conf/
$ sudo cp /tmp/Server Components/AMQ/jetty-realm.properties
/usr/local/activemq/conf/
$ sudo cp /tmp/Server Components/AMQ/credentials-enc.properties
/usr/local/activemq/conf/
$ sudo cp /tmp/Server Components/AMQ/webconsole-embedded.xml
/usr/local/activemq/webapps/admin/WEB-INF/
```

10. Change the owner and group to *activemq* for all files in the installation directory:

```
$ sudo chown -R activemq /usr/local/apache-activemq-5.13.0
$ sudo chgrp -R activemq /usr/local/apache-activemq-5.13.0
$ sudo chown -R activemq /usr/local/activemq
$ sudo chgrp -R activemq /usr/local/activemq
```

11. Start ActiveMQ (ActiveMQ is started in the background):

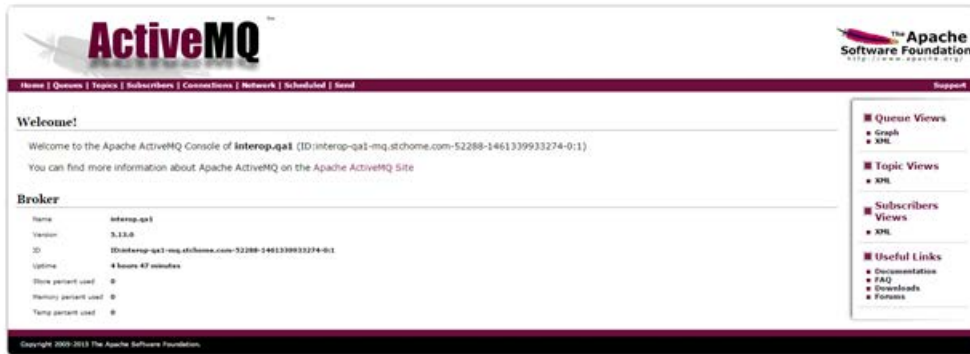
```
$ sudo /usr/local/apache-activemq-5.13.0/bin/activemq start
```

Verify Installation

Follow these steps to verify the installation was successful:

1. Open a browser to this URL (enter your ActiveMQ server address): `http://your-mq-server:8161/admin/`
2. Use the default credentials: user `admin` and password `t2X599apg`

You should see the welcome page. The actual values for your server will differ from this example:



CAUTION: The default password should be changed prior to production. See [ActiveMQ Credential Encryption - Linux](#).

Autostart

Follow these steps to set up autostart:

1. Copy the file:

```
cp /usr/local/apache-activemq-5.13.0/bin/activemq /etc/init.d/
chmod 755 /etc/init.d/activemq
```

2. Edit the `/etc/init.d/activemq` file and locate this code:

```
# Active MQ installation dir
if [ -z "$ACTIVEMQ_HOME" ] ; then
    ACTIVEMQ_HOME=`getActiveMQHome`
```

3. Change it to this:

```
# Active MQ installation dir
if [ -z "$ACTIVEMQ_HOME" ] ; then
    ACTIVEMQ_HOME="/usr/local/activemq"
```

4. Locate this code:

```
# Configure a user with non root privileges, if no user is specified do
not change user
if [ -z "$ACTIVEMQ_USER" ] ; then
    ACTIVEMQ_USER=""
```

5. And change it to:

```
# Configure a user with non root privileges, if no user is specified do
not change user
if [ -z "$ACTIVEMQ_USER" ] ; then
    ACTIVEMQ_USER="activemq"
```

6. Restart and enable ActiveMQ:

```
systemctl daemon-reload
systemctl enable activemq
```

ActiveMQ Credential Encryption - Linux

The instructions below detail how to encrypt credentials for the Web Admin user and for the JMS connection. Note that all paths are relative to your ActiveMQ installation directory (example: /usr/local/apache-activemq-5.13.0).

Encrypt Credentials for Web Admin User

There are two steps for this: generate the encrypted password, then set the encrypted password for the user.

Step 1: Generate the Encrypted Password

Use the jetty password utility to generate an encrypted password for a specific user. Note that [version] is the specific version of your distribution, [user] is the name of the user for whom the password is being generated, and [password] is the password to be encrypted.

```
$ java -cp ./lib/web/jetty-all-[version].jar
org.eclipse.jetty.util.security.Password [user] [password]
```

Example: Generate an encrypted password for the admin user (where the password is *myPassword*):

```
$ java -cp /usr/local/apache-activemq-5.13.0/lib/web/jetty-all-
9.2.13.v20150730.jar org.eclipse.jetty.util.security.Password admin
myPassword
```

Here is the output:

```
2016-08-25 14:45:58.469:INFO::main: Logging initialized @89ms
myPassword
OBF:luh41zly1qw01vu1lym71ym71vv91qxqlzlklugm
MD5:deb1536f480475f7d593219aa1afd74c
CRYPT:adtJskS8cjahg
```

Note that the encrypted password is *CRYPT:adtJskS8cjahg* in the last line of code above.

Step 2: Set the Encrypted Password for the User

Edit the Jetty realm properties (`apache-activemq-[version]/conf/jetty-realm.properties`). Replace the password with the CRYPT line from the generated output.

Example: Set the encrypted password (`CRYPT:adtJskS8cjahg` in the second-to-last line of code below):

\$ vi /usr/local/apache-activemq-5.13.0/conf/jetty-realm.properties

```
## -----
-
## Licensed to the Apache Software Foundation (ASF) under one or more
## contributor license agreements. See the NOTICE file distributed with
## this work for additional information regarding copyright ownership.
## The ASF licenses this file to You under the Apache License, Version 2.0
## (the "License"); you may not use this file except in compliance with
## the License. You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.
## -----
-
# Defines users that can access the web (console, demo, etc.)
# username: password [,rolename ...]
#admin: admin, admin
admin: CRYPT:adtJskS8cjahg, admin
user: user, user
```

Encrypt Password for JMS Connection

There are also two steps for this one: generate the encrypted password, then set the encrypted password for the connection users.

Step 1: Generate the Encrypted Password

Use the ActiveMQ password utility to generate an encrypted password for a specific user. Note that [password] is the encryptor's password and [input] is the password to be encrypted.

```
$ bin/activemq encrypt --password [password] --input [input]
```

Example: Generate an encrypted password (where the password is *myPassword*):

```
$ /usr/local/apache-activemq-5.13.0/bin/activemq encrypt --password activemq
--input myPassword
```

Here is the output:

```
ACTIVEMQ_HOME: /usr/local/apache-activemq-5.13.0
ACTIVEMQ_BASE: /usr/local/apache-activemq-5.13.0
ACTIVEMQ_CONF: /usr/local/apache-activemq-5.13.0/conf
ACTIVEMQ_DATA: /usr/local/apache-activemq-5.13.0/data
Encrypted text: n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp
```

Note that the encrypted password is `n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp` in the last line of code above.

Step 2: Set the Encrypted Password for the Connection Users

Edit the `conf/credentials-enc.properties` file, setting the encrypted value `ENC(encrypted password)`. Replace the password with the encrypted text line from the generated output.

Example: Set the encrypted password
(`password=ENC(n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp)` lines of code below):

\$ vi /usr/local/apache-activemq-5.13.0/conf/credentials-enc.properties

```
## -----
-
## Licensed to the Apache Software Foundation (ASF) under one or more
## contributor license agreements. See the NOTICE file distributed with
## this work for additional information regarding copyright ownership.
## The ASF licenses this file to You under the Apache License, Version 2.0
## (the "License"); you may not use this file except in compliance with
## the License. You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.
## -----
-
# Defines credentials that will be used by components (like web console) to
# access the broker
activemq.username=system
activemq.password=ENC(n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp)

guest.password=ENC(n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp)
user.password=ENC(n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp)
producer.password=ENC(n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp)

admin.password=ENC(n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp)
```

Then edit `conf/activemq.xml` and verify the lines:

```
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
```

contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
-->
<!-- START SNIPPET: example -->
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://activemq.apache.org/schema/core http://activemq.apache.org/schema/core/activemq-
core.xsd">

<bean id="configurationEncryptor"
class="org.jasypt.encryption.pbe.StandardPBEStrngEncryptor">
<property name="algorithm" value="PBEWithMD5AndDES"/>
<property name="password" value="activemq"/>
</bean>
<bean id="propertyConfigurer"
class="org.jasypt.spring31.properties.EncryptablePropertyPlaceholderConfigurer">
<constructor-arg ref="configurationEncryptor" />
<property name="location" value="file:${activemq.conf}/credentials-enc.properties"/>
</bean>

<!-- Allows accessing the server log -->
<bean id="logQuery" class="io.fabric8.insight.log.log4j.Log4jLogQuery" lazy-
init="false" scope="singleton" init-method="start" destroy-method="stop"></bean>
<!--
The <broker> element is used to configure the ActiveMQ broker. -->
<broker xmlns="http://activemq.apache.org/schema/core" brokerName="interop.qa1"
dataDirectory="${activemq.data}">

<plugins>
<simpleAuthenticationPlugin>
<users>
<authenticationUser username="admin" password="${admin.password}"
groups="users,admins"/>
<authenticationUser username="system" password="${activemq.password}"
groups="users,admins"/>
<authenticationUser username="user" password="${user.password}" groups="users"/>
<authenticationUser username="producer" password="${producer.password}"
groups="producers"/>
<authenticationUser username="guest" password="${guest.password}" groups="guests"/>
</users>
</simpleAuthenticationPlugin>
<authorizationPlugin>
<map>
<authorizationMap>
<authorizationEntries>
<authorizationEntry queue=">" read="admins" write="admins" admin="admins" />
<authorizationEntry queue="Interop.>" read="producers" write="producers"

```

```

admin="producers" />
<authorizationEntry topic=">" read="admins" write="admins" admin="admins" />

<authorizationEntry topic="ActiveMQ.Advisory.>"
read="guests,users,producers" write="guests,users,producers"
admin="guests,users,producers"/>
</authorizationEntries>
</authorizationMap>
</map>
</authorizationPlugin>
</plugins>

<destinationPolicy>
<policyMap>
<policyEntries>
<policyEntry topic=">">
<!-- The constantPendingMessageLimitStrategy is used to prevent slow topic consumers to
block producers and affect other consumers by limiting the number of messages that are
retained For more information, see: http://activemq.apache.org/slow-consumer-
handling.html -->
<pendingMessageLimitStrategy>
<constantPendingMessageLimitStrategy limit="1000"/>
</pendingMessageLimitStrategy>
</policyEntry>
</policyEntries>
</policyMap>
</destinationPolicy>
<!--
The managementContext is used to configure how ActiveMQ is exposed in JMX. By default,
ActiveMQ uses the MBean server that is started by the JVM. For more information, see:
http://activemq.apache.org/jmx.html -->
<managementContext>
<managementContext createConnector="false"/>
</managementContext>
<!--
Configure message persistence for the broker. The default persistence mechanism is the
KahaDB store (identified by the kahaDB tag). For more information, see:
http://activemq.apache.org/persistence.html -->
<persistenceAdapter>
<kahaDB directory="${activemq.data}/kahadb"/>
</persistenceAdapter>

<!--
The systemUsage controls the maximum amount of space the broker will use before
disabling caching and/or slowing down producers. For more information, see:
http://activemq.apache.org/producer-flow-control.html -->
<systemUsage>
<systemUsage>
<memoryUsage>
<memoryUsage percentOfJvmHeap="70" />
</memoryUsage>
<storeUsage>
<storeUsage limit="100 gb"/>
</storeUsage>
<tempUsage>
<tempUsage limit="50 gb"/>
</tempUsage>

```



```

</systemUsage>
</systemUsage>
<!--
The transport connectors expose ActiveMQ over a given protocol to clients and other
brokers. For more information, see: http://activemq.apache.org/configuring-
transports.html -->
<transportConnectors>
<!-- DOS protection, limit concurrent connections to 1000 and frame size to 100MB -->
<transportConnector name="openwire"
uri="tcp://0.0.0.0:61616?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>
<transportConnector name="amqp"
uri="amqp://0.0.0.0:5672?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>
<transportConnector name="stomp"
uri="stomp://0.0.0.0:61613?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>
<transportConnector name="mqtt"
uri="mqtt://0.0.0.0:1883?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>
<transportConnector name="ws"
uri="ws://0.0.0.0:61614?maximumConnections=1000&wireFormat.maxFrameSize=104857600"/>
</transportConnectors>
<!-- destroy the spring context on shutdown to stop jetty -->
<shutdownHooks>
<bean xmlns="http://www.springframework.org/schema/beans"
class="org.apache.activemq.hooks.SpringContextHook" />
</shutdownHooks>
</broker>
<!--
Enable web consoles, REST and Ajax APIs and demos The web consoles requires by default
login, you can disable this in the jetty.xml file Take a look at
${ACTIVEMQ_HOME}/conf/jetty.xml for more details -->
<import resource="jetty.xml"/>
</beans>
<!-- END SNIPPET: example -->

```

JMS Consumer Installation - Linux

These instructions address the installation of ActiveMQ on Server 2 (Message Queue).

Prerequisites

1. PostgreSQL has been installed on the DBB server (see [Postgres Database Installation - Linux](#)).
2. ActiveMQ has been installed and configured on the SVC server (see [ActiveMQ Installation - Linux](#)).
3. The distribution release zip file has been downloaded and unzipped into your home directory.

Installation Steps

1. Create the *batch* user:

```
$ sudo adduser batch -s /bin/bash
```

2. Set the password (enter it at the prompt):

```
$ sudo passwd batch
```

3. Modify `/home/batch/.bashrc` for the batch user, adding `JAVA_HOME` and `PATH` to the Java executable:

```
$ sudo vi /home/batch/.bashrc

# add your JAVA_HOME location
export JAVA_HOME=/usr/lib/jvm/jre-1.8.0

# add java to your path
export PATH=$JAVA_HOME/bin:$PATH
```

4. Create the installation directory:

```
sudo mkdir /usr/local/batch
sudo mkdir /usr/local/batch/consumer
```

5. Copy `STCConsumer.zip` into the installation directory:

```
$ sudo cp Application Components/IQ/AMQ/STCConsumer.zip
/usr/local/batch/consumer/
```

6. Unzip `STCConsumer.zip`:

```
$ cd /usr/local/batch/consumer
$ sudo unzip STCConsumer.zip
```

7. Change ownership and group it to `batch`:

```
$ cd /usr/local
$ sudo chown batch -R batch
$ sudo chgrp batch -R batch
```

8. Log in as the `batch` user:

```
$ sudo su - batch
```

9. Edit `/usr/local/batch/consumer/etc/stcConsumer.properties` and set the ActiveMQ details and state code:

```
#activemq details
stc.mq.user=producer
stc.mq.password=g512x1773v
stc.mq.brokerUrl=failover://tcp://your-mq-server:61616
stc.mq.queueName=Interop.milestone
```

```
...
```

```
## State Code  
stateCode=??
```

10. Edit `/usr/local/batch/consumer/etc/hibernate.cfg.xml` and set the database connection details:

```
<!-- Database details: Start -->  
<property  
name="connection.driver_class">org.postgresql.Driver</property>  
<property name="connection.url">jdbc:postgresql://your-db-  
server:5432/interopdb</property>  
<property name="connection.username">interop</property>  
<property name="connection.password">your-db-password</property>  
<!-- Database details: End -->
```

11. Edit `/usr/local/batch/consumer/Start_IQ_Consumer.sh` and set `JAVA_HOME` to the location of your Java executable (see the last line in the code below - your path may differ from the example):

```
## Please make sure jdk 1.8 is installed and java command on the server  
is pointing to JDK8  
## set java executable to below variable  
## example set JAVA_HOME="/usr/jdk8/java"  
JAVA_HOME="/usr/lib/jvm/jre-1.8.0"
```

12. Set the execute permission in `STC_IQ_Consumer.sh`:

```
$ chmod 755 Start_IQ_Consumer.sh
```

13. Execute the `dos2unix` command in the `Start_IQ_Consumer.sh` file:

```
$ dos2unix Start_IQ_Consumer.sh
```

14. Start the Consumer process:

```
$ ./Start_IQ_Consumer.sh
```

Stop Consumer

Follow these steps to stop the Consumer process:

1. Set the execute permission on the `Stop_IQ_Consumer.sh` file:

```
$ chmod 755 Stop_IQ_Consumer.sh
```

2. Execute the `dos2unix` command on the `Stop_IQ_Consumer.sh` file:

```
$ dos2unix Stop_IQ_Consumer.sh
```

3. Stop the Consumer process:

```
$ ./Stop_IQ_Consumer.sh
```

Autostart

Follow these steps to set up the autostart:

1. Run this command:

```
cp /usr/local/batch/consumer/Start_IQ_Consumer.sh /etc/init.d/consumer
```

2. Edit the Consumer script to add the chkconfig requirements and the CONSUMER_HOME variable. First, add this to the very top of the file:

```
#!/bin/bash
#chkconfig: 234 90 10
#description: JMS Consumer
```

3. Next, add this after the JAVA_HOME line:

```
CONSUMER_HOME="/usr/local/batch/consumer"
```

4. Finally, modify this line:

```
nohup $JAVA_HOME/bin/java -jar lib/stcconsumer.jar > runtimeconsole.log
2>&1&
-to-
nohup $JAVA_HOME/bin/java -jar $CONSUMER_HOME/lib/stcconsumer.jar >
runtimeconsole.log 2>&1&
```

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
```

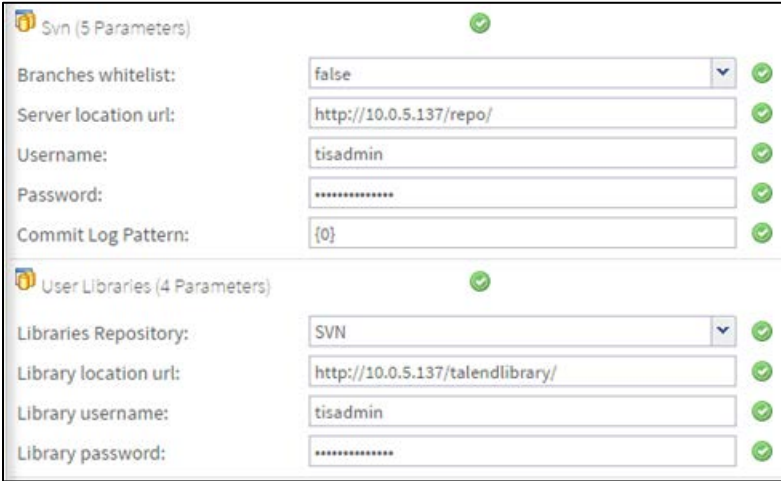
Command to start the service:

```
$ 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.talend.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 two configuration sections in the Admin Console, both marked with a green checkmark icon. The first section, titled "Svn (5 Parameters)", includes the following fields: "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 second section, titled "User Libraries (4 Parameters)", includes: "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 "*****"). Each field has a green checkmark to its right, indicating successful configuration.

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

Data

Login: admin@stchome.com

First name: admin

Last name: admin

Password: [change password](#)

Svn login: tisadmin

Svn password: *****

Type: Data Integration

Role: Administrator/Viewer

Active:

Connections stats

To the Administration Center

First: 2016-05-25 14:15:16

Last: 2016-05-25 14:15:16

Number: 1

To the Studio

First:

Last:

Number: 0

[Save](#) [Cancel](#)

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

Roles Selection

Role

Administrator

Viewer

Operation manager

Designer

[Validate](#)

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:

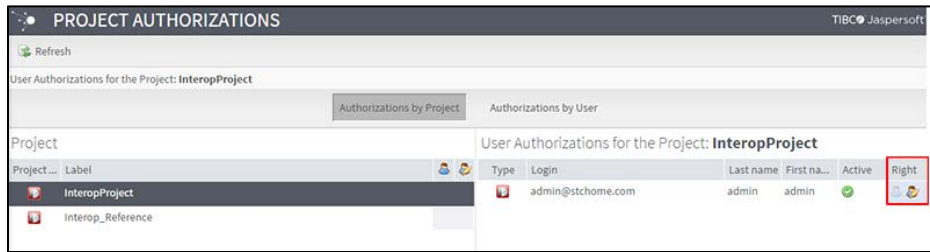
Description:

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.



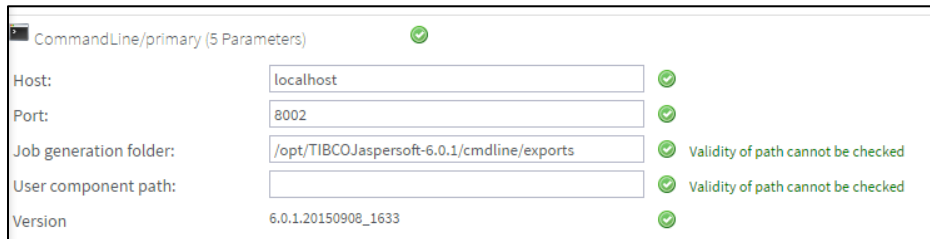
9. Create the following directories and files in the server:

```

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

SMTP (6 Parameters / 3 errors)

Use SMTP: true

Host:

Port:

Require SSL: false

Username: user@domain.com

Password:

STC|iQ Jaspersoft ETL Deployment - Linux

The following outlines how to import (migrate) Jaspersoft ETL for STC|iQ on Linux.

Pre-Migration Checklist

- Confirm that the Postgres database is installed and configured
- Retrieve the passwords for the SMaRT AFIX database (*interopdb*)
- 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
- Retrieve the link for the IWeb Org/Fac web services
- Retrieve the username and password for the PHC Hub database
- Copy these components from `Application Components/ETL` into your local directory:
 - `WF_Reset_DataMart.zip`
 - `WF_Int_DataMart_Map_x.x.zip` (where x.x is the version number)

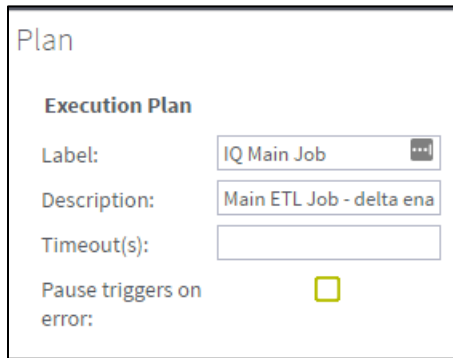
Deploy the ETL Components

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

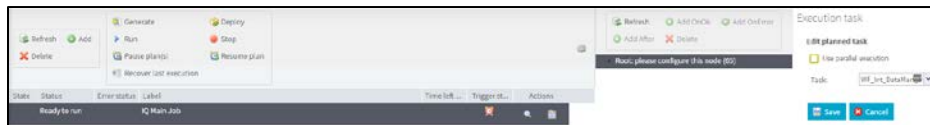
1. Copy these files to a convenient location to deploy:
`WF_Int_DataMart_Map_x.x.zip` and `WF_Reset_DataMart.zip`.
2. Navigate to the Job Conductor page.
 1. Select **Add > Normal Task** and add the following: Label = iQ Reset Data Mart; Description = iQ Reset Data Mart.

2. Click **Import Generated Jobs** and select the `WF_Reset_DataMart.zip` file downloaded as part of the prerequisites.
3. On successful import of the file, verify these properties:
 - Active: Selected (checked)
 - Project: InteropProject (auto-populated based on the imported file)
 - Branch: (auto-populated based on the imported file)
 - Name: WF_Int_DataMart_Map (auto-populated based on the imported file)
 - Version: (auto-populated based on the imported file)
 - Context: Config (auto-populated based on the imported file)
 - Apply Context to Children: Selected (checked)
 - Execution Server: Interop_JobServer
4. Select **Add > Normal Task** and add the following: Label = `WF_Int_DataMart_Map`; Description = `iQ Execution Plan`.
5. Click **Import Generated Jobs** and select the `WF_Int_DataMart_Map_x.x.zip` file downloaded as part of the prerequisites.
6. On successful import of the file, verify these properties:
 - Active: Selected (checked)
 - Project: InteropProject (auto-populated based on the imported file)
 - Branch: (auto-populated based on the imported file)
 - Name: WF_Int_DataMart_Map (auto-populated based on the imported file)
 - Version: (auto-populated based on the imported file)
 - Context: Config (auto-populated based on the imported file)
 - Apply Context to Children: Selected (checked)
 - Execution Server: Interop_JobServer
7. 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.
 - Web service related parameters:
 - Ws_Url: `http://[IWeb server name]:8080/iweb/api/v1/OrgFac/`
 - Ws_Api_Key: Secret key for IWeb
 - Database-related parameters:
 - DB_JDBC_ETLCONFIG:

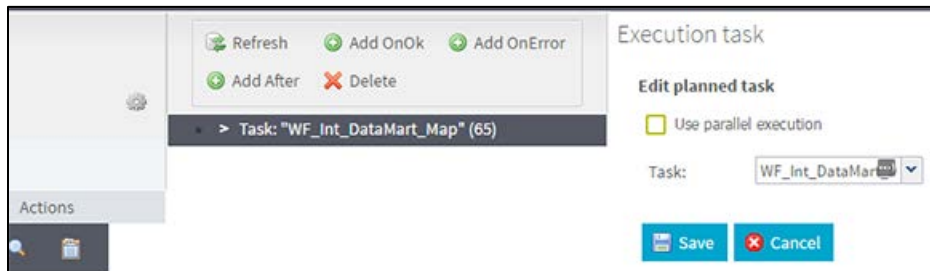
- DB_JDBC_ETLCONFIG_ --> [jdbc:postgresql://(PostgresServer: 5432/interopdb]
 - DB_JDBC_ETLCONFIG_Login --> username for the database
 - DB_JDBC_ETLCONFIG_Password --> password for the database user
 - DB_ORCL_PHCHUB:
 - DB_ORCL_PHCHUB_Schema --> CPWEB
 - DB_ORCL_PHCHUB_Login --> cpweb
 - DB_ORCL_PHCHUB_Password --> password for the login user
 - DB_ORCL_PHCHUB_SID --> CPWEB
 - DB_ORCL_PHCHUB_Port --> 1521 or other port if selected during installation
 - DB_PGRS_INTEROPOLAP:
 - DB_PGRS_INTEROPOLAP_Schema --> interopolap
 - DB_PGRS_INTEROPOLAP_Login --> interop (modify if different)
 - DB_PGRS_INTEROPOLAP_Password --> password for the login user
 - DB_PGRS_INTEROPOLAP_Database --> interopdb
 - DB_PGRS_INTEROPOLAP_Port --> 5432
 - DB_PGRS_ODS:
 - DB_PGRS_ODS_Schema --> ods
 - DB_PGRS_ODS_Login --> interop (modify if different)
 - DB_PGRS_ODS_Password --> password for login user
 - DB_PGRS_ODS_Database --> interopdb
 - DB_PGRS_ODS_Port --> 5432
8. Save when finished.
 9. Verify that the jobs deployed correctly: Click on WF_Int_DataMart_Map created in Step 4 above, then click **Run**. Deployment succeeded if no errors occur.
 10. Go to the execution plan and click **Add**, then enter the details of the execution plan on the right side of the screen and click **Save**.



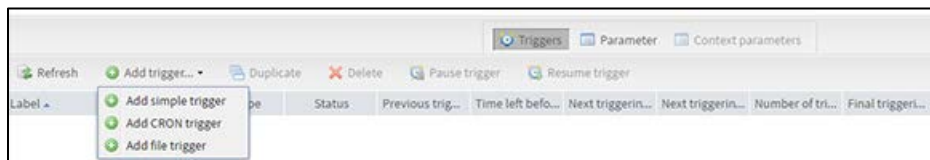
11. Add jobs to the execution plan: Click on the execution plan created in the previous step, then in the middle of the window select Root. On the third window, select the WF_Int_DataMart_Map task and click **Save**.



12. After the last step, the execution plan looks like this:



13. On that same page, click the Triggers tab for the execution plan at the bottom right to schedule the plan. Click the **Add trigger...** drop-down menu to open it and then select **Add CRON trigger**.



14. On the Add Cron trigger page, enter IQ Main Job Trigger in the label and click the **Open UI configurator** button.

Add Cron trigger

Label:

Description:

Time zone strategy:

[Open UI configurer](#) [Open Cron Help](#)

Minutes *:

Hours *:

Days of month **:

Months *:

Days of week **:

Years:

Pause trigger on error:

15. Configure how often the job should run and click **Apply Modifications**.

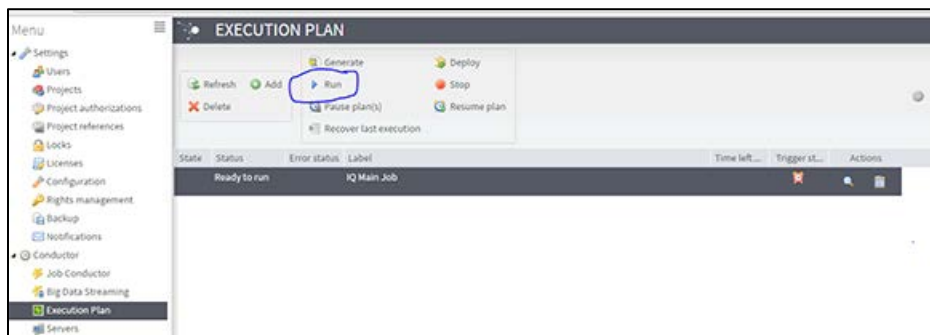
Cron UI Trigger configuration

Minutes *	Hours *	Days of	Months *	Days of	Years	
00	30	00	1	January	Sunday	2014
01	31	01	2	February	Monday	2015
02	32	02	3	March	Tuesday	2016
03	33	03	4	April	Wednesday	2017
04	34	04	5	May	Thursday	2018
05	35	05	6	June	Friday	2019
06	36	06	7	July	Saturday	2020
07	37	07	8	August		2021
08	38	08	9	September		2022
09	39	09	10	October		2023
10	40	10	11	November		2024
11	41	11	12	December		2025
12	42	12	13			2026
13	43	13	14			2027
14	44	14	15			2028
15	45	15	16			
16	46	16	17			
17	47	17	18			
18	48	18	19			
19	49	19	20			
20	50	20	21			
21	51	21	22			
22	52	22	23			
23	53	23	24			
24	54		25			
25	55		26			
26	56		27			
27	57		28			
28	58		29			
29	59		30			
			31			
			last day of month			

[Apply modifications](#)

16. Note that after the trigger times are selected, the Add Cron trigger looks similar to this:

17. To manually run an ETL job, click on the iQ main job created here and click **Run**.



18. 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 **iQ 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.

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
- 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
```

12. It should look something like this:

```
# Highcharts static export properties
com.jaspersoft.jasperreports.highcharts.render.require.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/require/require-2.1.6.src.js
com.jaspersoft.jasperreports.highcharts.jquery.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/jquery-1.10.2.min.js
com.jaspersoft.jasperreports.highcharts.highcharts.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/highcharts-4.2.1.src.js
com.jaspersoft.jasperreports.highcharts.highcharts.more.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/highcharts-more-4.2.1.src.js
com.jaspersoft.jasperreports.highcharts.highcharts.heatmap.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/highcharts-heatmap-4.2.1.src.js
com.jaspersoft.jasperreports.highcharts.highcharts.treemap.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/highcharts-treemap-4.2.1.src.js
com.jaspersoft.jasperreports.highcharts.highcharts.3d.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/highcharts-3d-4.2.1.src.js
com.jaspersoft.jasperreports.highcharts.highcharts.solid.gauge.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/render/scripts/highcharts-solid-gauge-4.2.1.src.js
com.jaspersoft.jasperreports.highcharts.data.service.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/services/require/data.service.js
com.jaspersoft.jasperreports.highcharts.default.service.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/services/require/default.service.js
com.jaspersoft.jasperreports.highcharts.item.hyperlink.service.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/services/require/item.hyperlink.service.js
com.jaspersoft.jasperreports.highcharts.dual.pie.service.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/services/require/dual.pie.service.js
com.jaspersoft.jasperreports.highcharts.treemap.service.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/charts/services/require/treemap.service.js
com.jaspersoft.j1.adhoc.highcharts.setting.service.js$context.url=reportresource?resource=com/jaspersoft/j1/adhoc/j1/require/adhocHighchartsSettingService.js
com.jaspersoft.jasperreports.highcharts.chart.producer.js$context.url=reportresource?resource=com/jaspersoft/jasperreports/highcharts/chart/producer.js
com.jaspersoft.jasperreports.highcharts.function.properties.allowed=true

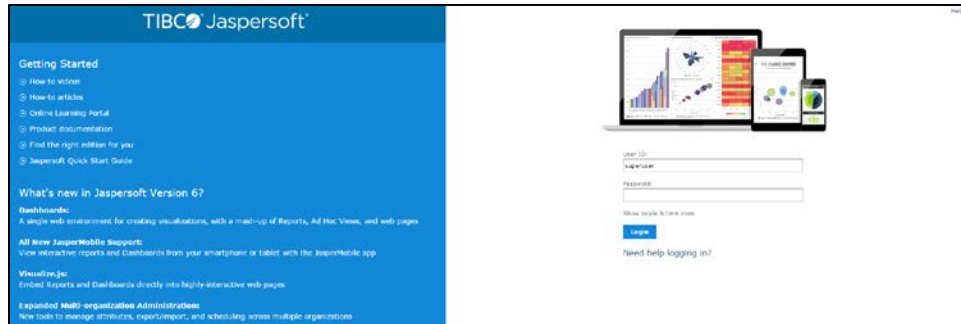
# highcharts: phantomjs's executable absolute path
com.jaspersoft.jasperreports.highcharts.phantomjs.executable.path=path/to/phantomjs
```

13. Save the file and restart the JasperReports Server (see [Start/Stop the JasperReports Server](#) below).

14. 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`

15. 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 using FTP or WinSCP.
2. Replace the license file in the JasperReports installation directory (`/opt/jasperreports-server-6.2.1/`) with the license file from the above step, overwriting the existing `jasperserver.license` file.
3. Restart the JasperReports server (see [Start/Stop the JasperReports Server](#) below).

Increase the JasperReports Session Timeout Period

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

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

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

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

JasperReports Server Modifications

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

Key	XML Tag Name (s) to Update	Value
[KEYCLOAK_URL]	<pre><authorization_location> <token_location> <userdetails_location> <logoutUrl></pre>	<p>URL of the Keycloak server. Example: <code>https://sso-dev.stchome.com/</code></p>
[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 <code><keycloak_URL></code>. 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: <pre><property name="client_id"> <value>jasper-aws-dev</value></pre></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: <pre><bean id="oAuthUserDetailsService" class="com.jaspersoft.jasperserver.ps.OAuthUserDetailsServiceImpl"> <property name="state"> <value>WA</value> </property> </bean></pre></p>
Jaspersoft Roles	<pre><bean id="mtExternalUserSetupProcessor" class="com.jaspersoft.jasperserver. multipleTenancy.security.externalAuth. processors.MTExternalUserSetupProcessor" parent="abstractExternalProcessor" > <property name="organizationRoleMap"></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 <code><entry></code> tag, the key value should match with the client role defined at the Keycloak end. Currently, the following roles are configured:</p> <ol style="list-style-type: none"> 1. Administrator

Key	XML Tag Name (s) to Update	Value
	<pre><map> <!-- Example of mapping customer roles to JRS roles --> <entry> <key> <value>ROLE_ADMIN_EXTERNAL_ORGA NIZATION</value> </key> <value>ROLE_ADMINISTRATOR</value > </entry> </map></pre>	<p>2. Superuser</p> <p>3. User Role</p>

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

ID

Name

Consent Required OFF

Mapper Type

Token Claim Name

API-KEY

API-URI

Response format

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 Migration for STC|iQ

See [iQ JasperReports Deployment - Linux](#) for specific instructions for iQ.

STC|iQ JasperReports Deployment - Linux

The following outlines how to import (migrate) Jaspersoft Reports for STC|iQ on Linux.

Pre-Migration Checklist

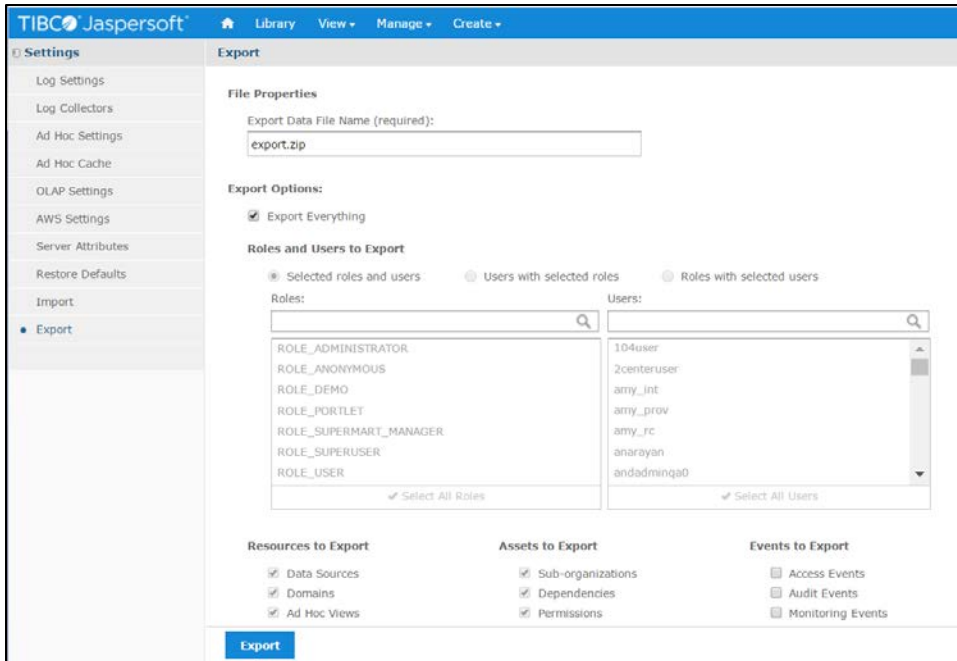
- Confirm that the Postgres database is installed and configured
- Retrieve the passwords for the STC|iQ database (*interopdb*)
- Retrieve the Jaspersoft superuser username and password
- Confirm that all necessary ports are open across all firewalls (the default port is 80)
- Confirm that the release distribution zip file for Jasper Reports is extracted from `Application Components/JRS/iQ_Report_Repository.zip` and copied to your local drive.

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 `iQ_Report_Repository.zip` file.
4. Use the defaults and click the **Import** button at the bottom.
5. After the import completes, it should display a confirmation message. Click **Close** at the top.

Import the Content via the Command Line

1. Confirm that `JAVA_HOME` is set and your path includes `jre\bin` or `jdk\bin`.
2. Stop the Tomcat instance from running by using this command in the `/opt/jasperreports-server-6.2.1` directory:

```
sudo ./ctlscript.sh stop tomcat
```

3. Change to the `buildomatic` directory on the JasperReports server:

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

4. Execute the migrate command:

```
sudo ./js-import.sh --input-zip "[home
```

```
directory]/iQ_Report_repository.zip" --update --include-access-events --include-audit-events --include-monitoring-events --include-server-settings --include-themes
```

5. Start the Tomcat server:

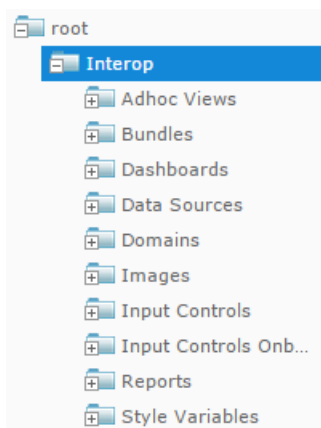
```
sudo ./ctlscript.sh start tomcat
```

6. Update the data source with the database connection information.

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/interop` (select **View > Repositories**).



2. Validate that the user can execute and view a report under `interop/reports` as defined by their class or organization membership.

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.

Deploy the LatoFont.jar File

This installation only needs to be performed once.

1. Copy the LatoFont.jar executable file from Server Components\JRS into jasperreports-server-6.2.0\apache-tomcat\webapps\jasperserver-pro\WEB-INF\lib.
2. Restart the JasperReports server. See [Start/Stop the JasperReports Server](#) for instructions on how to do this.

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
```

HL7 Transaction Log Viewer Installation - Linux

STC|iQ is currently using CentOS 7 as its Linux distribution server. The HL7 Transaction Log Viewer requires Node.js 6.9. If necessary, see [Build a Linux \(CentOS\) Server](#) and/or [Node.js Installation - Linux](#) first and then return to these instructions.

Extract the Files

Copy the installation file to the server and extract it. Run:

```
cp interop-logviewer.zip /home/node/  
unzip interop-logviewer.zip
```

Configure Process.json

Update the application configuration in the `process.json` file. The following fields must be configured to match your environment. Other fields in the `process.json` file can generally be left at their defaults, unless you are customizing the installation.

Name	Value
SCRIPT	/full/path/to/app.js
APP_HOST	The hostname where the log viewer is installed.
APP_PATH	The path, if the log viewer does not run at the root of the hostname. If customizing, the value must have a leading slash (/) but no training slash.
SAML_ENTRY_POINT	https://[Keycloak URL]/auth/realms/[Realm Name]/protocol/saml
POSTGRES_CONNECT	postgres://username:password@host/database_name

Start Process.json

Run this command:

```
pm2 start process.json
```

Set Up the Proxy

The node application runs unencrypted via HTTP on port 3500 by default. It is necessary to proxy the connection through an HTTP server, preferably SSL-encrypted. Any HTTP server that supports proxy pass can be used, such as Apache, IIS, or Nginx.

STC|iQ 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; general configuration options; and information on how to run the application and start the application upon bootup.

Prerequisites

The following should already be installed and configured:

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

Installation and Deployment of iQ, SMaRT AFIX and VOMS

If you are installing iQ alongside the SMaRT AFIX 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

1. Ensure epel repo is enabled and run the following with sudo/root:

```
yum install redis
```

2. Edit the /etc/redis.conf file and make sure the following variable line is set as follows (the default is usually set to an empty string ""):

```
notify-keyspace-events EKx
```

3. Set redis to autostart across system boots and start the service:

```
systemctl enable redis
systemctl start redis
```

Install the Deployment Package

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

Configure the Application Dashboard and Quick Links

If you are installing iQ alongside the SMaRT AFIX 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 SMaRT AFIX, 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/iQ_dark.svg",
  "name": "iQ",
  "url": "http://iq.stchome.com/",
  "description": "An application that allows for an efficient assessment of HL7
Data quality, and provides tools and metrics to help onboard new providers
and improve data quality.",
  "accessRole": "Access_interop"
}
```

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 VOMS 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.

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.

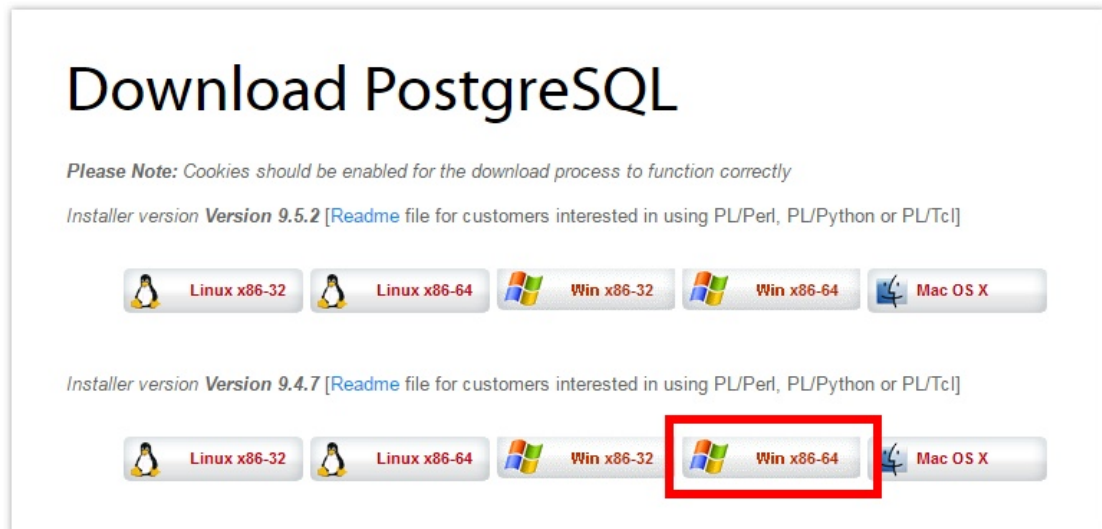
Postgres Database Installation - Windows

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 iQ (Interop).

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\
```

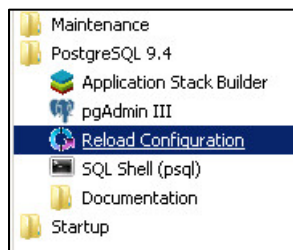
1. Edit the `pg_hba.conf` file to allow connections from other STC servers/applications. Copy/paste the following line:

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

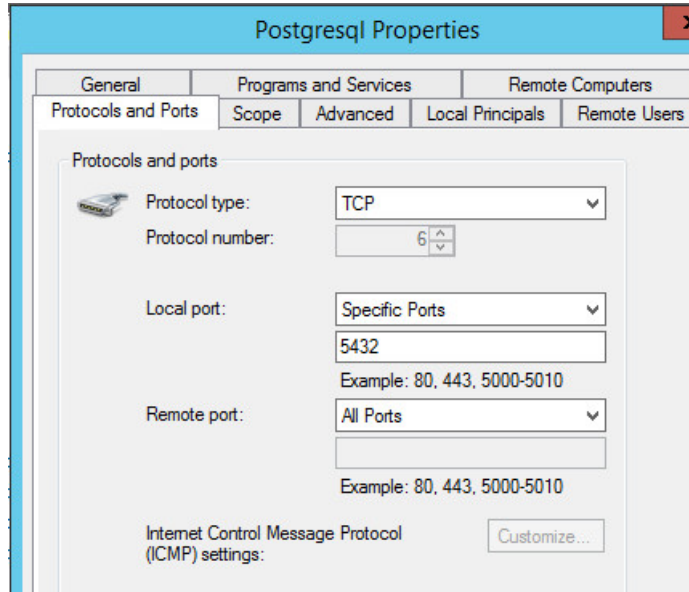
to make the file look something like the following. Let's assume your internal subnet falls under `192.168.x.x`, which would update the line to:

```
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 [iQ Database Initialization - Windows](#)).

iQ Database Initialization - Windows

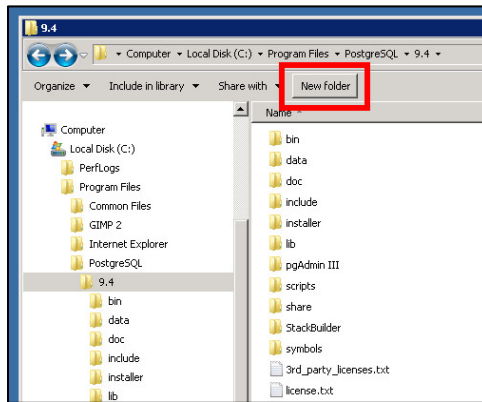
The information below describes how to initialize the iQ database on Windows.

Prerequisites

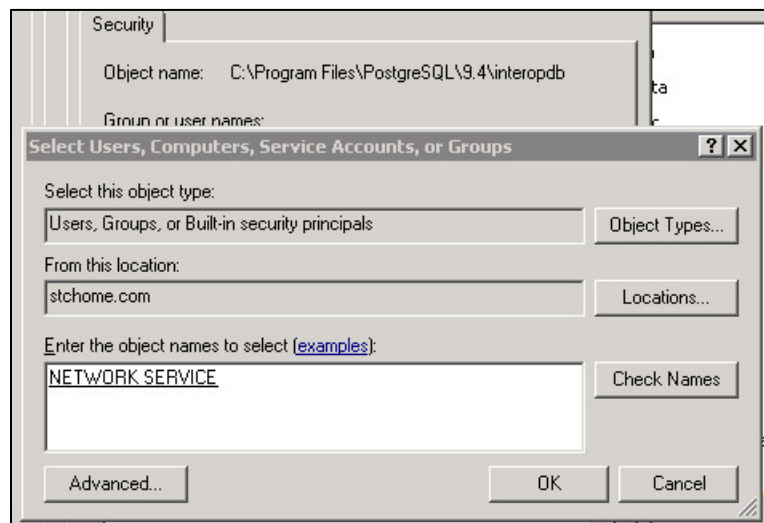
PostgreSQL has been installed.

Steps

1. Create a tablespace folder - Use Windows Explorer to create the directory for the tablespace to be used by iQ. Navigate to the folder `C:\Program Files\PostgreSQL\9.4` and then click **New Folder**. Name the new folder *interopdb*.



2. Set the folder permissions for the *interopdb* folder:
 - a. In Windows Explorer, right-click on the *interopdb* folder and select **Properties**.
 - b. Click on the Security tab.
 - c. Click the **Edit** button.
 - d. Click the **Add** button.
 - e. Enter the name *NETWORK SERVICE* and click on the **Check Names** button. The window should look like the following image, although your location will be different and local to your network:



3. Still in the same window, click OK.
4. Make sure *NETWORK SERVICE* is selected and also select the checkbox under *Allow for Full control*.
5. Click **OK** on the remaining open windows to close them.

6. Back in Windows Explorer (interopdb directory), create the *data* directory:

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

7. Create the *idx* directory:

```
C:\Program Files\PostgreSQL\9.4\interopdb/idx
```

8. Download the release distribution zipped file and extract the database scripts for the database server in the interopdb folder:

```
Application Components\IQ\POSTGRES
```

9. Edit the Application Components\IQ\POSTGRES\setup_database.sql script, changing the following:

```
create tablespace odstbs owner interop location
'/var/lib/pgsql/9.4/interopdb';
-to-
create tablespace odstbs owner interop location 'C:\Program
Files\PostgreSQL\9.4\interopdb';
```

10. Edit the Application Components\IQ\POSTGRES\setup_objects.sql script, changing the following:

```
-- Create Tablespace
CREATE TABLESPACE "interopdata" OWNER interop LOCATION
'/var/lib/pgsql/9.4/interopdb/data';
CREATE TABLESPACE "interopidx" OWNER interop LOCATION
'/var/lib/pgsql/9.4/interopdb/idx';
-to-
-- Create Tablespace
CREATE TABLESPACE "interopdata" OWNER interop LOCATION 'C:\Program
Files\PostgreSQL\9.4\interopdb\data';
CREATE TABLESPACE "interopidx" OWNER interop LOCATION 'C:\Program
Files\PostgreSQL\9.4\interopdb/idx';
```

11. Create the interopdb database - From the command prompt, navigate to the Application Components\IQ\POSTGRES directory and execute the following command. Enter the password that was configured during the Postgres installation, if prompted.

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

12. Create the database users and roles - From the command prompt, navigate to the Application Components\IQ\POSTGRES directory and execute the following command. Enter the password that was configured during the Postgres installation, if prompted.

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

13. Create the ETL config schema by navigating to the Application Components\IQ\POSTGRES directory and execute the following command:

```
flyway -url=jdbc:postgresql://localhost:5432/interopdb -  
schemas=etlconfig -sqlMigrationPrefix=ETL_V -user=interop migrate
```

14. Create the OLAP schema by navigating to the Application Components\IQ\POSTGRES directory and execute the following command, using the password *InteropQA*:

```
flyway -url=jdbc:postgresql://localhost:5432/interopdb -  
schemas=interopolap -sqlMigrationPrefix=OLAP_V -user=interop migrate
```

15. Create the ODS schema by navigating to the Application Components\IQ\POSTGRES directory and execute the following command, using the password *InteropQA*:

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

MongoDB Installation - Windows

The following instructions describe how to install MongoDB on Windows Server and how to set the MongoDB server configuration changes for Interop (iQ).

1. Install MongoDB.

Since the August 2016 release, Interop (iQ) requires MongoDB v2.6.11 or 2.6.12. You can find the download link at <https://www.mongodb.com/download-center> (search for previous versions, either 2.6.11 or 2.6.12).

Follow the instructions for installing MongoDB that are listed here: <https://docs.mongodb.com/v2.6/tutorial/install-mongodb-on-windows/>. It is recommended that you simply copy all of the files in the \bin directory inside the zip file directly into c:\mongodb (**NOT** into c:\mongodb\bin).

2. Create the configuration file.

For the purpose of these instructions, the following reference folders are used (substitute custom folders as needed):

- c:\mongodb - MongoDB binaries
- c:\data\db - STC database folder
- c:\data\log - Mongo logs

Create the configuration file in the folder `c:\mongodb` and name it `mongod.cfg`. Use these contents, or copy this file from the STC deployment folder at `\Application Components\DB\IQ\MONGO` from the configuration instructions in the next section below.

```
systemLog:
  destination: file
  path: c:\data\log\mongod.log
storage:
  dbPath: c:\data\db
net:
  bindIp: 127.0.0.1
  port: 27017
setParameter:
  enableLocalhostAuthBypass: false
```

3. Add Mongo Binaries Directory to Command Path

Add `c:\mongodb` (or use the location you selected, if different) to your Windows Command Path. This is usually found in Windows under **Control Panel > System > Advanced > Environment Variables**.

Make MongoDB Server Configuration Changes for Interop (iQ)

1. Obtain the Mongo distribution files.

Copy these three files into your `c:\mongodb` directory. These are located in your STC distribution directory (`\Application Components\DB\IQ\MONGO`):

- `mongod.cfg`
- `stc-mongo-admin-db.js`
- `stc-mongo-stc-db.js`

Copy the Mongo `MONGO_STC_DB` directory into the `c:\mongodb` directory.

NOTE: It is highly recommended to install MongoDB as a service. This process is documented under the section [Configure a Windows Service for MongoDB](https://docs.mongodb.com/v2.6/tutorial/install-mongodb-on-windows/) here: <https://docs.mongodb.com/v2.6/tutorial/install-mongodb-on-windows/>.

Open a command window, `cd` to `c:\mongodb`, and execute the following:

```
mongorestore.exe -d stc MONGO_STC_DB
```

2. Set the passwords for the MongoDB users.

First, create the *superuser* account. By default, there is no password set (`pwd: ""`). Edit `stc-mongo-admin-db.js` and set a unique password, then save it and keep the file nearby. The file should look like this, but with the new password ("MySUpassword"):

```
db.createUser(
{
  user: "superuser",
  pwd: "MySUpassword",
  roles: [ "root" ]
})
```

Next, set the password for the STC account that will be used by the Interop portal application (iQ). Edit `stc-mongo-stc-db.js` and set a unique password, using the same process as above. This password should be different from the superuser account, however.

NOTE: The *stcadmin* user and password you selected will be needed during installation of the Interop portal application (iQ). It will be entered on the `./production/deploy.sh` command.

3. Apply the MongoDB scripts.

With MongoDB now installed, the server running and the passwords configured, you can run the client interface by entering the commands at a CMD prompt. Make sure you are in the `c:\mongodb` directory. Enter the following commands:

```
mongo admin stc-mongo-admin-db.js
mongo stc stc-mongo-stc-db.js
```

4. Edit/Update `c:\mongodb\mongod.cfg`.

Edit the `mongod.cfg` file, editing the `net` section as follows (or customize it to your network subnet), then add the following security section lines to the file. Save the file when finished.

```
net:
  bindIp: 0.0.0.0
security:
  authorization: enabled
```

Restart the MongoDB sever with the following:

```
net stop mongodb
net start mongodb
```

MongoDB for Interop (iQ) is now set up and complete.

ActiveMQ Installation - Windows

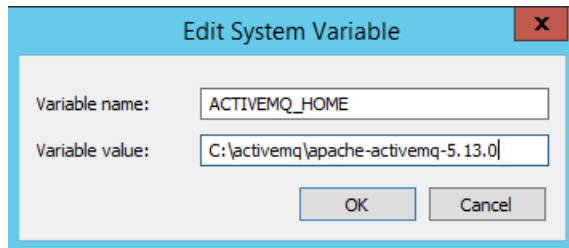
The following information describes the prerequisites and steps for installing ActiveMQ on Windows.

Prerequisites

The Java Development Kit (1.7 or higher) JDK must be installed.

Steps

1. Create the ACTIVEMQ_HOME environmental variable in Windows with the path of C:\activemq\apache-activemq-5.13.0.



2. Copy the ActiveMQ release distribution installation file into the directory of your choice. For the purposes of these instructions, this location is referred to as *ACTIVEMQ_HOME*.

```
Server Components\AMQ\apache-activemq-5.13.0-bin.zip
```

3. Unzip the file into the C:\activemq directory (this is ACTIVEMQ_HOME).
4. Once unzipped, the directory looks like this:

```
c:\activemq\apache-activemq-5.13.0
```

5. Verify that java.exe is in your PATH and that the environment variable JAVA_HOME is set.
6. Copy the configuration files:

```
copy Server Components\AMQ\activemq.xml %ACTIVEMQ_HOME%\apache-activemq-5.13.0\conf\  
copy Server Components\AMQ\jetty-realm.properties  
%ACTIVEMQ_HOME%\apache-activemq-5.13.0\conf\  
copy Server Components\AMQ\credentials-enc.properties  
%ACTIVEMQ_HOME%\apache-activemq-5.13.0\conf\  
copy Server Components\AMQ\webconsole-embedded.xml  
%ACTIVEMQ_HOME%\apache-activemq-5.13.0\webapps\admin\WEB-INF\  

```

7. Start ActiveMQ - from the command prompt, navigate to the %ACTIVEMQ_HOME%\apache-activemq-5.13.0 directory and execute the following command:

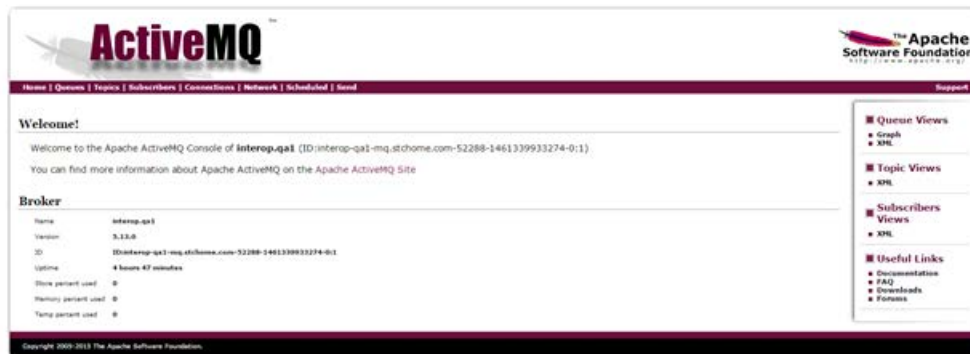
```
bin\activemq start
```

Verify Installation

Follow these steps to verify the installation was successful:

1. Open a browser to this URL (enter your ActiveMQ server address): `http://your-mq-server:8161/admin/`
2. Use the default credentials: user *admin* and password *t2X599apg*

You should see the welcome page. The actual values for your server will differ from this example:



CAUTION: The default password should be changed prior to production. See [ActiveMQ Credential Encryption - Windows](#).

Stop ActiveMQ

From the command window, press **CTRL+C** on the keyboard.

Create the Service

Follow these steps to create the service:

1. From an administrative command prompt, go to `ACTIVEMQ_HOME/bin/win64`.
2. Run `InstallService.bat`.

After running `InstallService.bat`, the ActiveMQ service should be added to the list of NT services. It is not started by default.

To verify, go to the **Control Panel > Administrative Tools > Services** and look for the ActiveMQ service. Here you can start the ActiveMQ service or configure it to start automatically every time the system boots.

ActiveMQ Credential Encryption - Windows

The instructions below detail how to encrypt credentials for the Web Admin user and for the JMS connection. Note that all paths are relative to your ActiveMQ installation directory (example: `c:\activemq\apache-activemq-5.13.0`).

Encrypt Credentials for Web Admin User

There are two steps for this: generate the encrypted password, then set the encrypted password for the user.

Step 1: Generate the Encrypted Password

Use the Jetty password utility to generate an encrypted password for a specific user. Note that [version] is the specific version of your distribution, [user] is the name of the user for whom the password is being generated, and [password] is the password to be encrypted.

```
$ java -cp .\lib\web\jetty-all-[version].jar  
org.eclipse.jetty.util.security.Password [user] [password]
```

Example: Generate an encrypted password for the admin user (where the password is *myPassword*):

```
$ java -cp c:\activemq\apache-activemq-5.13.0\lib\web\jetty-all-  
9.2.13.v20150730.jar org.eclipse.jetty.util.security.Password admin  
myPassword
```

Here is the output:

```
2016-08-25 14:45:58.469:INFO::main: Logging initialized @89ms  
myPassword  
OBF:1uh41zly1qw01vu1lym71ym71vv91qxq1zlk1ugm  
MD5:deb1536f480475f7d593219aalafd74c  
CRYPT:adtJskS8cjahg
```

Note that the encrypted password is *CRYPT:adtJskS8cjahg* in the last line of code above.

Step 2: Set the Encrypted Password for the User

Edit the jetty realm properties (`apache-activemq-[version]\conf\jetty-realm.properties`). Replace the password with the CRYPT line from the generated output.

Example: Set the encrypted password (*CRYPT:adtJskS8cjahg* in the second-to-last line of code below):

```
c:\activemq\apache-activemq-5.13.0\conf\jett-realm.properties
```

```

## -----
-
## Licensed to the Apache Software Foundation (ASF) under one or more
## contributor license agreements. See the NOTICE file distributed with
## this work for additional information regarding copyright ownership.
## The ASF licenses this file to You under the Apache License, Version 2.0
## (the "License"); you may not use this file except in compliance with
## the License. You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the License is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.
## -----
-
# Defines users that can access the web (console, demo, etc.)
# username: password [,rolename ...]
#admin: admin, admin
admin: CRYPT:adtJskS8cjahg, admin
user: user, user

```

Encrypt Password for JMS Connection

There are also two steps for this one: generate the encrypted password, then set the encrypted password for the connection users.

Step 1: Generate the Encrypted Password

Use the ActiveMQ password utility to generate an encrypted password for a specific user. Note that [password] is the encryptor's password and [input] is the password to be encrypted.

```
$ bin\activemq encrypt --password [password] --input [input]
```

Example: Generate an encrypted password (where the password is *myPassword*):

```
$ c:\activemq\apache-activemq-5.13.0\bin\activemq encrypt --password activemq
--input myPassword
```

Here is the output:

```

ACTIVEMQ_HOME: c:\activemq\apache-activemq-5.13.0
ACTIVEMQ_BASE: c:\activemq\apache-activemq-5.13.0
ACTIVEMQ_CONF: c:\activemq\apache-activemq-5.13.0\conf
ACTIVEMQ_DATA: c:\activemq\apache-activemq-5.13.0\data
Encrypted text: n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp

```

Note that the encrypted password is *n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp* in the last line of code above.

Step 2: Set the Encrypted Password for the Connection Users

Edit the `conf\credentials-enc.properties` file, setting the encrypted value `ENC(encrypted password)`. Replace the password with the encrypted text line from the generated output.

Example: Set the encrypted password
(`password=ENC(n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp)` lines of code below):

c:\activemq\apache-activemq-5.13.0\conf\credentials-enc.properties

```
## -----  
-  
## Licensed to the Apache Software Foundation (ASF) under one or more  
## contributor license agreements. See the NOTICE file distributed with  
## this work for additional information regarding copyright ownership.  
## The ASF licenses this file to You under the Apache License, Version 2.0  
## (the "License"); you may not use this file except in compliance with  
## the License. You may obtain a copy of the License at  
##  
## http://www.apache.org/licenses/LICENSE-2.0  
##  
## Unless required by applicable law or agreed to in writing, software  
## distributed under the License is distributed on an "AS IS" BASIS,  
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
## See the License for the specific language governing permissions and  
## limitations under the License.  
## -----  
-  
# Defines credentials that will be used by components (like web console) to  
# access the broker  
activemq.username=system  
activemq.password=ENC(n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp)  
  
guest.password=ENC(n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp)  
user.password=ENC(n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp)  
producer.password=ENC(n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp)  
  
admin.password=ENC(n8PLSJGxRKx0wSaCWsSXWfnbK79F5kmp)
```

JMS Consumer Installation - Windows

These instructions address the installation of ActiveMQ on Windows.

Prerequisites

1. PostgreSQL has been installed on the DBB server (see [Postgres Database Installation - Linux](#)).
2. ActiveMQ has been installed and configured on the SVC server (see [ActiveMQ Installation - Linux](#)).

3. The distribution release zip file has been downloaded and unzipped into your home directory.
4. Decide whether to perform a [basic](#) installation or to install JMS Consumer as a Windows service ([advanced](#)).

Installation Steps (Basic)

1. Copy the Application Components\iQ\AMQ\STCConsumer.zip file to c:\consumer.
2. Unzip the STCConsumer.zip file.
3. Verify that the java.exe file is in your PATH and the environment variable JAVA_HOME is set.
4. Edit c:\consumer\etc\stcConsumer.properties to change the stc.mq.brokerUrl to the hostname of the ActiveMQ server and to set the ActiveMQ details and state code:

```
#frequency of the scheduler to check the queue
stc.consumer.cron=0 0/1 * * * ?
#frequency of the scheduler to process the open messages
stc.payload.cron=0 0/3 * * * ?
#activemq details
stc.mq.user=admin
stc.mq.password=t2X599apg
stc.mq.brokerUrl=failover://tcp://52.39.220.0:61616
stc.mq.queueName=Interop.milestone

## Run MQ Active Queue
mq.enable=true
## Retrieving data in milestone dump table once
run.once=false
## State Code
stateCode=AZ
```

5. Edit RunSTCConsumer.bat to set JAVA_EXC to the path of your java.exe file:

```
ECHO OFF
REM Please make sure jdk 1.8 is installed and java command on the server
is pointing to JDK8
REM set java executable to below variable
REM example set JAVA_EXC="/usr/jdk8/java"
set JAVA_EXC="c:\java\bin\java.exe"
REM export JAVA_EXC
echo %JAVA_EXC%
IF EXIST "consumer.lck" (
echo "Consumer process is currently running. Please make sure only one
process is running"
) ELSE (
copy /y nul consumer.lck
%JAVA_EXC%\bin\java -jar lib/stcconsumer.jar
del consumer.lck
echo "Running"
)
```

6. Start the Consumer process:

```
$ ./Start_IQ_Consumer.sh
```

Installation Steps (Advanced)

Follow these steps to install JMS Consumer as a Windows service. The service installation components are located in `Application Components\iQ\AMQ\STCConsumerService.zip`.

1. Download the `STCConsumerService.zip` file and unzip it into your installation directory (`c:\consumer`). Your installation directory should have these files:

```
etc
lib
RunSTCConsumer.bat
Start_IQ_Consumer.sh
Stop_IQ_Consumer.sh
STCConsumer.exe
STCConsumer.xml
STCConsumer.zip
```

2. Follow the basic steps 2-6 above.
3. Install the service: From the command prompt, execute the following:

```
C:\consumer> STCConsumer install
```

4. Start the service:

```
C:\consumer> STCConsumer start
```

5. Stop the service:

```
C:\consumer> STCConsumer stop
```

6. Monitor the service:

```
C:\consumer> STCConsumer status
```

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
```

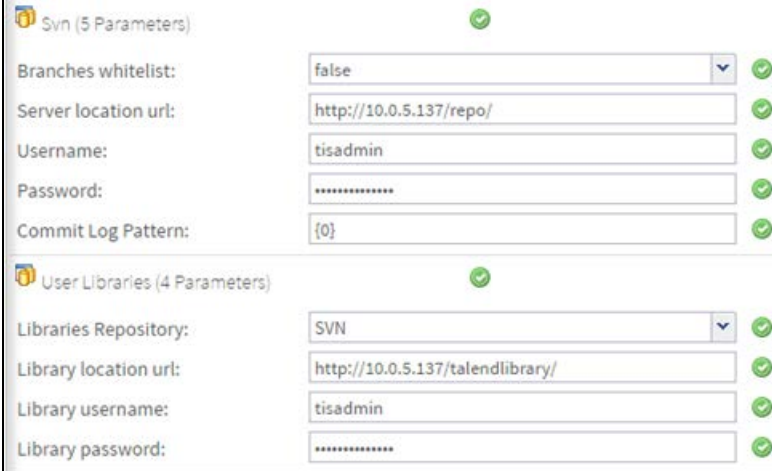
Command to start the service:

```
$$ 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 contains several input fields with green checkmarks indicating they are correctly configured.

Section	Parameter	Value	Status
Svn (5 Parameters)	Branches whitelist:	false	✓
	Server location url:	http://10.0.5.137/repo/	✓
	Username:	tisadmin	✓
	Password:	*****	✓
	Commit Log Pattern:	{0}	✓
User Libraries (4 Parameters)	Libraries Repository:	SVN	✓
	Library location url:	http://10.0.5.137/talendlibrary/	✓
	Library username:	tisadmin	✓
	Library password:	*****	✓

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:

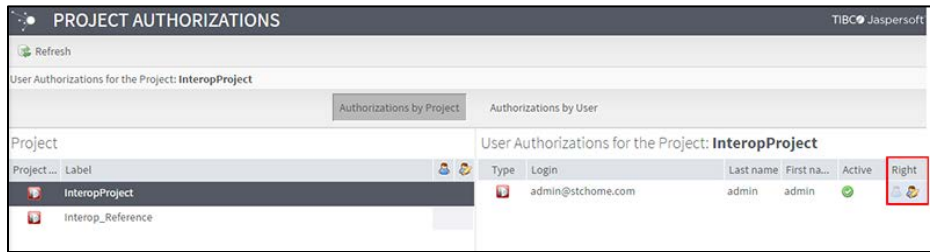
Description:

Author:

Storage: SVN None

Advanced settings

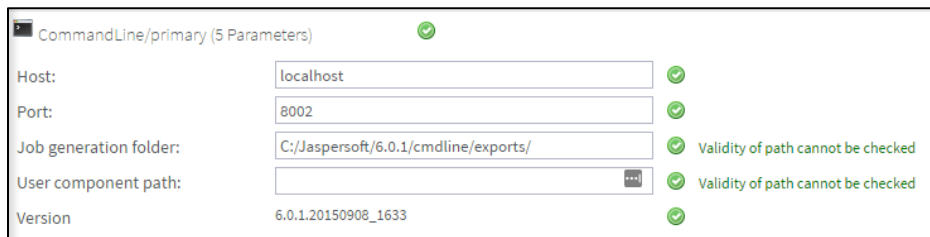
8. Navigate to **Project authorization** from the left menu. Click on each of the newly created projects and assign read/write permissions to the user.



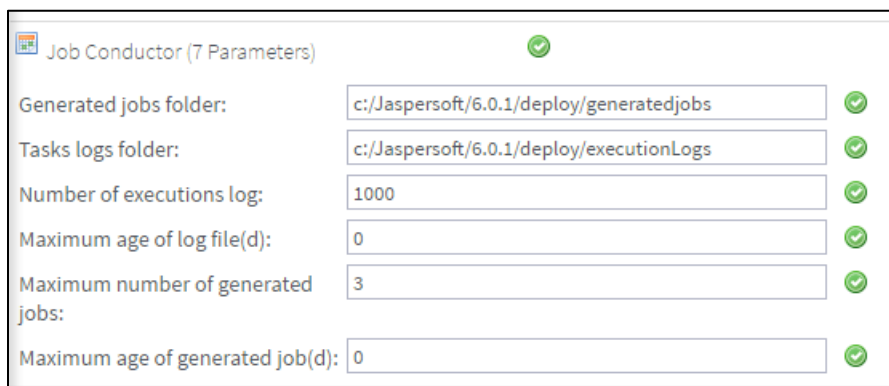
9. Create the following directories and files in the server:

```
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

The screenshot shows a configuration window for SMTP settings. The title bar reads 'SMTP (6 Parameters / 3 errors)' with a red error icon. The fields are as follows:

Use SMTP:	true
Host:	
Port:	
Require SSL:	false
Username:	user@domain.com
Password:	*****

STC|iQ Jaspersoft ETL Deployment - Windows

The following outlines how to import (migrate) Jaspersoft ETL for STC|iQ on Windows.

Pre-Migration Checklist

- Confirm that the Postgres database is installed and configured
- Retrieve the passwords for the SMaRT AFIX database (*interopdb*)
- 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
- Retrieve the link for the IWeb Org/Fac web services
- Retrieve the username and password for the PHC Hub database
- Copy these components from `Application Components/ETL` into your local directory:
 - `WF_Reset_DataMart.zip`
 - `WF_Int_DataMart_Map_x.x.zip` (where x.x is the version number)

Deploy the ETL Components

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

1. Copy these files to a convenient location to deploy:
`WF_Int_DataMart_Map_x.x.zip` and `WF_Reset_DataMart.zip`.
2. Navigate to the Job Conductor page.

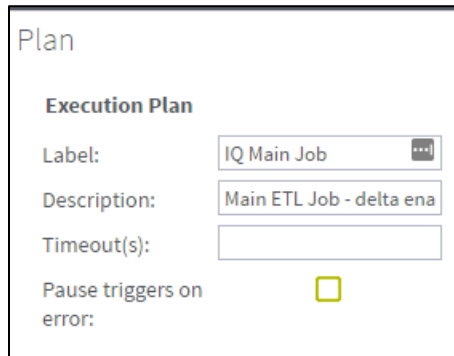
3. Select **Add > Normal Task** and add the following: Label = iQ Reset Data Mart; Description = iQ Reset Data Mart.
4. Click **Import Generated Jobs** and select the WF_Reset_DataMart.zip file downloaded as part of the prerequisites.
5. On successful import of the file, verify these properties:
 - Active: Selected (checked)
 - Project: InteropProject (auto-populated based on the imported file)
 - Branch: (auto-populated based on the imported file)
 - Name: WF_Int_DataMart_Map (auto-populated based on the imported file)
 - Version: (auto-populated based on the imported file)
 - Context: Config (auto-populated based on the imported file)
 - Apply Context to Children: Selected (checked)
 - Execution Server: Interop_JobServer
6. Select **Add > Normal Task** and add the following: Label = WF_Int_DataMart_Map; Description = iQ Execution Plan.
7. Click **Import Generated Jobs** and select the WF_Int_DataMart_Map_x.x.zip file downloaded as part of the prerequisites.
8. On successful import of the file, verify these properties:
 - Active: Selected (checked)
 - Project: InteropProject (auto-populated based on the imported file)
 - Branch: (auto-populated based on the imported file)
 - Name: WF_Int_DataMart_Map (auto-populated based on the imported file)
 - Version: (auto-populated based on the imported file)
 - Context: Config (auto-populated based on the imported file)
 - Apply Context to Children: Selected (checked)
 - Execution Server: Interop_JobServer
9. 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.
 - Web service related parameters:
 - Ws_Url: http://[IWeb server name]:8080/iweb/api/v1/OrgFac/
 - Ws_Api_Key: Secret key for IWeb

- Database-related parameters:
 - DB_JDBC_ETLCONFIG:
 - DB_JDBC_ETLCONFIG_ --> [jdbc:postgresql://(PostgresServer: 5432/interopdb]
 - DB_JDBC_ETLCONFIG_Login --> username for the database
 - DB_JDBC_ETLCONFIG_Password --> password for the database user
 - DB_ORCL_PHCHUB:
 - DB_ORCL_PHCHUB_Schema --> CPWEB
 - DB_ORCL_PHCHUB_Login --> cpweb
 - DB_ORCL_PHCHUB_Password --> password for the login user
 - DB_ORCL_PHCHUB_SID --> CPWEB
 - DB_ORCL_PHCHUB_Port --> 1521 or other port if selected during installation
 - DB_PGRS_INTEROPOLAP:
 - DB_PGRS_INTEROPOLAP_Schema --> interopolap
 - DB_PGRS_INTEROPOLAP_Login --> interop (modify if different)
 - DB_PGRS_INTEROPOLAP_Password --> password for the login user
 - DB_PGRS_INTEROPOLAP_Database --> interopdb
 - DB_PGRS_INTEROPOLAP_Port --> 5432
 - DB_PGRS_ODS:
 - DB_PGRS_ODS_Schema --> ods
 - DB_PGRS_ODS_Login --> interop (modify if different)
 - DB_PGRS_ODS_Password --> password for login user
 - DB_PGRS_ODS_Database --> interopdb
 - DB_PGRS_ODS_Port --> 5432

10. Save when finished.

11. Verify that the jobs deployed correctly: Click on WF_Int_DataMart_Map created in Step 4 above, then click **Run**. Deployment succeeded if no errors occur.

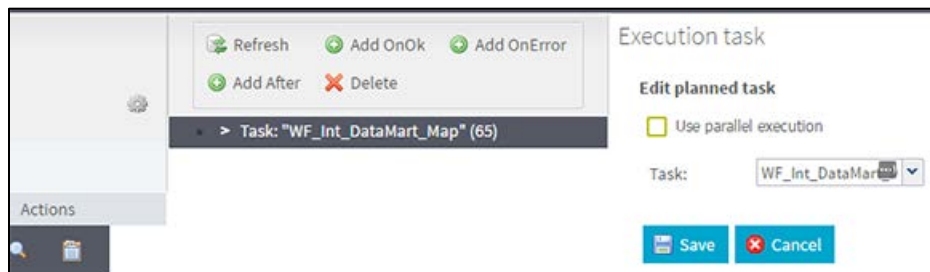
- Go to the execution plan and click **Add**, then enter the details of the execution plan on the right side of the screen and click **Save**.



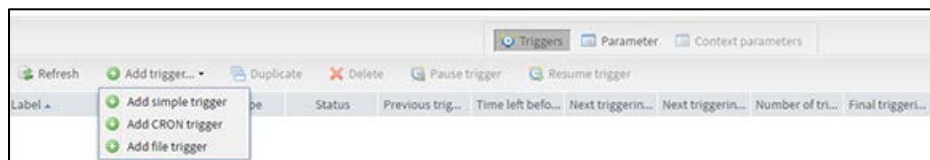
- Add jobs to the execution plan: Click on the execution plan created in the previous step, then in the middle of the window select Root. On the third window, select the WF_Int_DataMart_Map task and click **Save**.



- After the last step, the execution plan looks like this:



- On that same page, click the Triggers tab for the execution plan at the bottom right to schedule the plan. Click the **Add trigger...** drop-down menu to open it and then select **Add CRON trigger**.



- On the Add Cron trigger page, enter IQ Main Job Trigger in the label and click the **Open UI configurator** button.

Add Cron trigger

Label: IQ Main Job Trigger

Description: Triggers the Incremental Loz

Time zone strategy: TAC time

Open UI configurer Open Cron Help

Minutes *:

Hours *:

Days of month **:

Months *:

Days of week **:

Years:

Pause trigger on error:

17. Configure how often the job should run and click **Apply Modifications**.

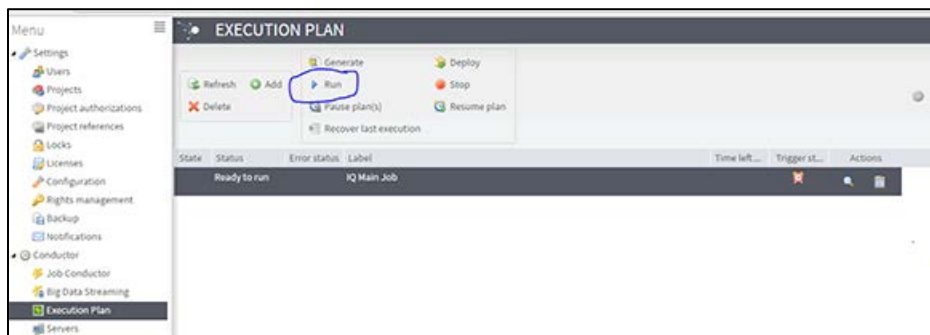
Cron UI Trigger configuration

Minutes *	Hours *	Days of	Months *	Days of	Years
00	30	00	1	January	Sunday 2014
01	31	01	2	February	Monday 2015
02	32	02	3	March	Tuesday 2016
03	33	03	4	April	Wednesday 2017
04	34	04	5	May	Thursday 2018
05	35	05	6	June	Friday 2019
06	36	06	7	July	Saturday 2020
07	37	07	8	August	2021
08	38	08	9	September	2022
09	39	09	10	October	2023
10	40	10	11	November	2024
11	41	11	12	December	2025
12	42	12	13		2026
13	43	13	14		2027
14	44	14	15		2028
15	45	15	16		
16	46	16	17		
17	47	17	18		
18	48	18	19		
19	49	19	20		
20	50	20	21		
21	51	21	22		
22	52	22	23		
23	53	23	24		
24	54		25		
25	55		26		
26	56		27		
27	57		28		
28	58		29		
29	59		30		
			31		
			last day of month		

Apply modifications

18. Note that after the trigger times are selected, the Add Cron trigger looks similar to this:

19. To manually run an ETL job, click on the iQ main job created here and click **Run**.



20. 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 **iQ 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.

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-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
```

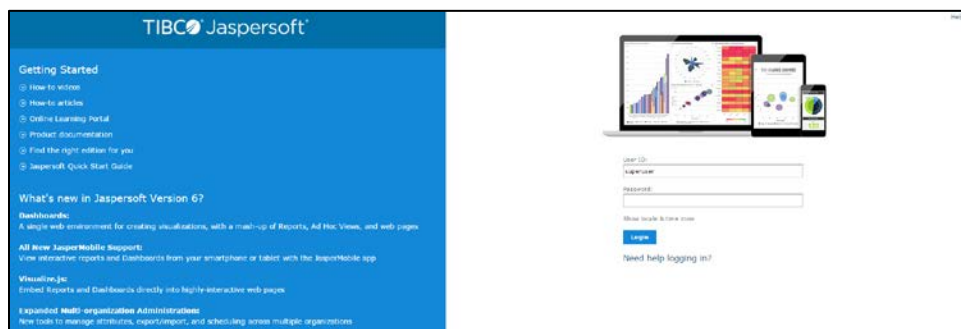
6. On completion of the installation, start the JasperReports server by selecting the following:

```
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

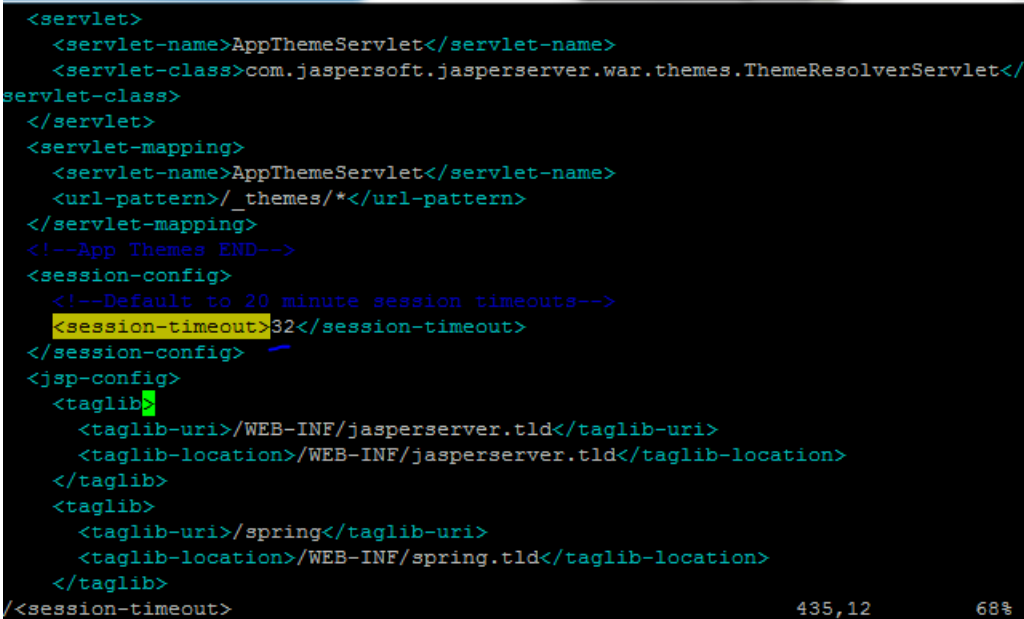
Follow these steps to update the license file for the JasperReports server:

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>
```

3. 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

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

JasperReports Server Modifications

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

Key	XML Tag Name(s) to Update	Value
[KEYCLOAK_URL]	<pre><authorization_location > <token_location> <userdetails_location> <logoutUrl></pre>	URL of the Keycloak server. Example: <code>https://sso-dev.stchome.com/</code>
[KEYCLOAK_REALM]	<pre><authorization_location > <token_location> <userdetails_location> <logoutUrl></pre>	Keycloak realm that the client belongs to. This value needs to be changed, along with <code><keycloak_URL></code> . In this example, the realm name is <i>Dev-Integration</i> : <code>https://sso-dev.stchome.com/auth/realms/Dev-Integration</code>
[JASPERSERVER_URL]	<pre><property name="redirecturl"> <property name="logoutUrl"></pre>	URL of the Jasper server. Example: <code>http://<server_name>:8080/jasperserver-pro/oauth</code>
[CLIENT_ID]	<pre><property name="client_id"></pre>	Client ID as set on Keycloak. Example: <pre><property name="client_id"> <value>jasper-aws-dev</value></pre>
[CLIENT_SECRET]	<pre><property name="clientsecret"> <property name="userdetails_secret"></pre>	Credentials secret key generated by Keycloak.
State_ID	<pre><property name="state"></pre>	This is the global setting used to populate the STATE_ID attribute in Jaspersoft. Example: <pre><bean id="oAuthUserDetailsService" class="com.jaspersoft.jasperserver.ps.OAuth.OAuthUserDetailsServiceImpl"> <property name="state"> <value>WA</value> </property> </bean></pre>
Jaspersoft Roles	<pre><bean id="mtExternalUserSetu</pre>	All the Jaspersoft Roles are configured inside this tag. If any new Jaspersoft Roles are

Key	XML Tag Name(s) to Value Update	
	<pre> pProcessor" class="com.jaspersoft.ja sperserver.multipleTena ncy.security.externalAut h.processors.MTEexternal UserSetupProcessor" parent="abstractExterna IProcessor"> <property name="organizationRole Map"> <map> <!-- Example of mapping customer roles to JRS roles --> <entry> <key> <value>ROLE_ADMIN_E XTERNAL_ORGANIZATIO N</value> </key> <value>ROLE_ADMINIS TRATOR</value> </entry> </map> </pre>	<p>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

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 Jasperserver database in Postgres: `\c jasperserver;` (the password is `postgres`).
 - e. Execute this SQL alter script: `ALTER TABLE jiprofileattribute ALTER COLUMN attrvalue TYPE character varying(100000);`
9. Restart the Jasper server and navigate to `JasperUrl/oauth` (see example). You should now be redirected to Keycloak to log in. Example:
`http://52.10.228.158:8080/jasperserver-pro/oauth`

Steps to Debug Issues Related to Keycloak/JasperReports Server Integration

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

Check the `applicationContext-externalAuth-oAuth.xml` File

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

Keycloak Mappers

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

The screenshot shows the configuration for an 'Org/fac List' mapper in Keycloak. The fields are as follows:

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

Buttons for 'Save' and 'Cancel' are located at the bottom of the form.

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 Migration for STC|iQ

See [iQ JasperReports Deployment - Windows](#) for specific instructions for iQ.

STC|iQ JasperReports Deployment - Windows

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

Pre-Migration Checklist

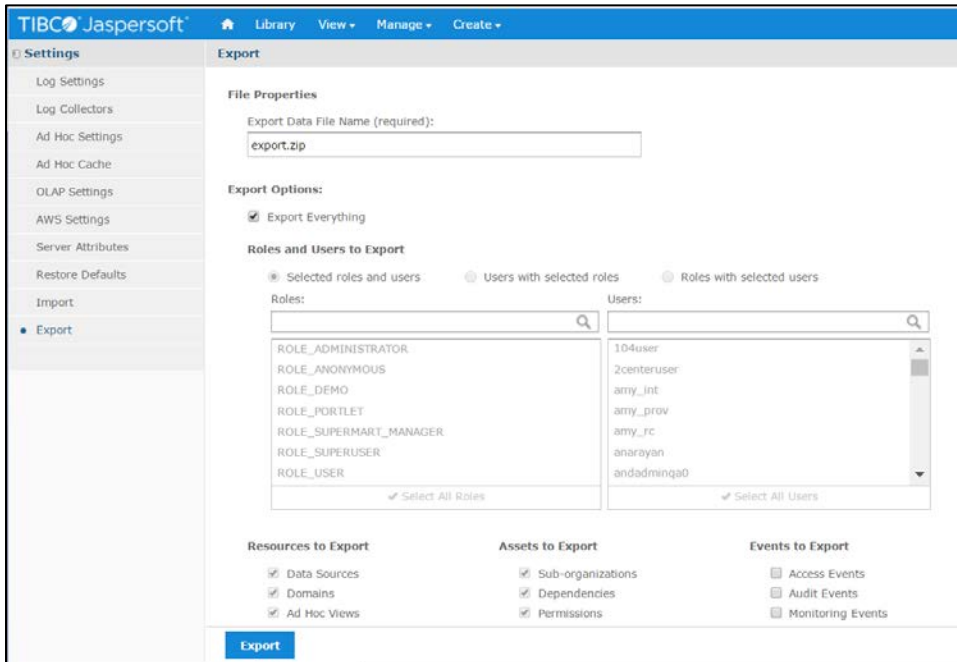
- Confirm that the Postgres database is installed and configured
- Retrieve the passwords for the SMaRT AFIX database (*interopdb*)
- Retrieve the Jaspersoft superuser username and password
- Confirm that all necessary ports are open across all firewalls (the default port is 80)
- Confirm that the release distribution zip file for Jasper Reports is extracted from `Application Components/JRS/iQ_Report_Repository.zip` and copied to your local drive.

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 `iQ_Report_Repository.zip` file.
4. Update the data source with the database connection information.

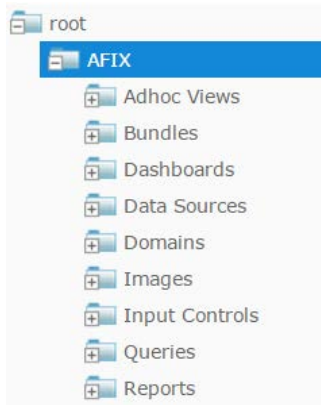
Import the Content via the Command Line

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:
`js-import.bat --input-zip "C:\source\interop-portal\develop\Reporting\iQ_Report_Repository.zip" --update --skip-user-update --include-access-events --include-audit-events --include-monitoring-events`
4. Update the data source with the database connection information.

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/Interop` (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 `interop/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.

STC|iQ 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 iQ, 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)
- Mongo server (DBB)
- Oracle server (COR)
- PM2 (APP)

Installation and Deployment of iQ, SMaRT AFIX, and VOMS

If you are installing iQ alongside the SMaRT AFIX 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

See <https://github.com/MicrosoftArchive/redis> for more information. Download the release at <https://github.com/MicrosoftArchive/redis/releases/tag/win-3.2.100>.

1. Use this command to install Redis as a Windows Service:

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

2. Edit the `redis.widnows-service.conf` file which is located (in default installations) in `C:\Program Files\Redis`. Update the following variable line as follows (the default is usually set to an empty string `""`):

```
notify-keyspace-events EKx
```

3. Restart the Redis Windows Service for the above change to take effect.

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 iQ alongside the SMaRT AFIX 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/iQ_dark.svg",
  "name": "iQ",
  "url": "http://iq.stchome.com/",
  "description": "An application that allows for an efficient assessment of HL7 Data quality, and provides tools and metrics to help onboard new providers and improve data quality.",
  "accessRole": "Access_interop"
}
```

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 VOMS with a reverse proxy, this should be set to <i>true</i> .

Option	Description
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.

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
```

STC | iQ Application Configuration

The following information describes how to configure the STC | iQ application.

STC | iQ Web Server-Specific Options

There are currently no options available for the STC | iQ web server.

STC | iQ API-Specific Options

The table below lists the available API-specific options:

Option	Description
INTEROP_POSTGRES_USER	The iQ PostgreSQL user.
INTEROP_POSTGRES_PASSWORD	The iQ PostgreSQL password.
INTEROP_POSTGRES_CONNECTION_STRING	The iQ PostgreSQL connection string.
INTEROP_MONGODB_CONNECTION_STRING	The iQ MongoDB connection string.
HL7_LOGVIEWER_URL	The URL for the HL7 Transaction Log Viewer.
STATE_LOGO_PATH	The path to the state logo image file.

Organization/Facility Micro-Service-Specific Options

The table below lists the available options for the organization/facility micro-service:

Option	Description
VOMS_ORACLE_USER	The username to the Oracle database.
VOMS_ORACLE_PASSWORD	The password to the Oracle database.
VOMS_ORACLE_CONNECTION	The Oracle database connection string for the IWeb database.

Keycloak Configuration for STC | iQ

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 VOMS release `Application Components/iQ/SSO/JSON` directory.
5. Select the `interop.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 *interop* 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.