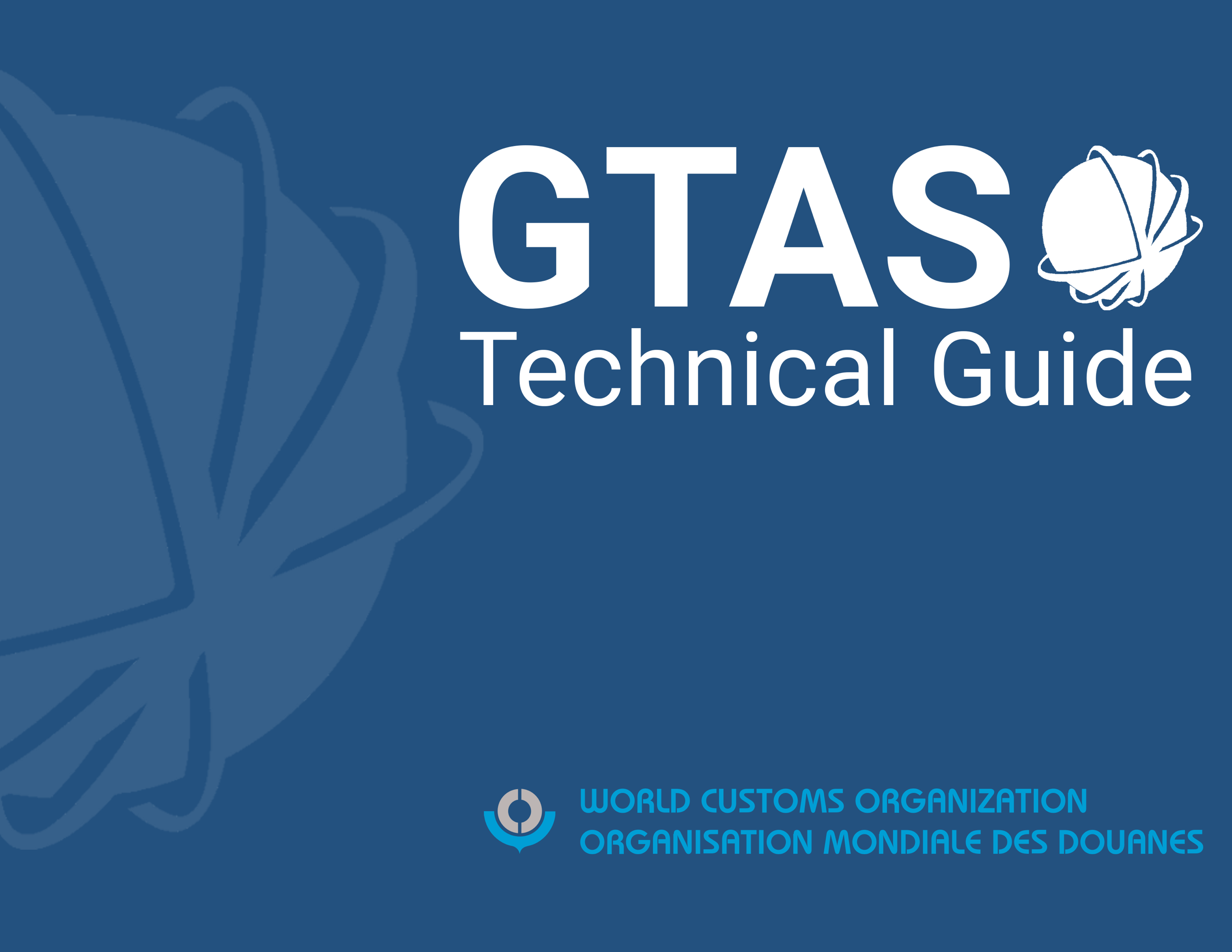
****exc

Table of Contents

[Foreword 4](#_Toc524351013)

[Ancillary Documents 5](#_Toc524351014)

[GTAS Installation / Upgrade 5](#_Toc524351015)

[Setting up the Environment 5](#_Toc524351016)

[Environment 6](#_Toc524351017)

[Download 6](#_Toc524351018)

[Pre-build Configuration 7](#_Toc524351019)

[Build 8](#_Toc524351020)

[Deploy 9](#_Toc524351021)

[Configure: Set country and UTC offset 10](#_Toc524351022)

[Configuring the Jaro-Winkler Distance and Double Metaphone Watchlist matching thresholds (optional) 12](#_Toc524351023)

[Starting the ActiveMQ, Redis, ElasticSearch, and the Tomcat Server 13](#_Toc524351024)

[Troubleshooting Immediately after Installation 14](#_Toc524351025)

[I can’t see any flights! 14](#_Toc524351026)

[I can’t access http://localhost:8080/gtas 15](#_Toc524351027)

[Redis 15](#_Toc524351028)

[Redis Commands 15](#_Toc524351029)

[Amazon Web Services (AWS) Deployment 15](#_Toc524351030)

[Installation of Message Queue Setup (Apache ActiveMQ) 15](#_Toc524351031)

[Installation Backup 16](#_Toc524351032)

[Browser Requirements 16](#_Toc524351033)

[ElasticSearch 16](#_Toc524351034)

[Installation 16](#_Toc524351035)

[Useful Commands 16](#_Toc524351036)

[Rest API | External Queries 17](#_Toc524351037)

[Security 17](#_Toc524351038)

[Digitally-Signed .war Files 17](#_Toc524351039)

[(Coming Soon) Configuring Tomcat to support SSL or https 17](#_Toc524351040)

[(Coming Soon) Industry Standard Security Checks 17](#_Toc524351041)

[System Design 17](#_Toc524351042)

[Drag and drop file uploader configuration 18](#_Toc524351043)

[Miscellaneous Commands 18](#_Toc524351044)

[Tomcat Server Commands 18](#_Toc524351045)

[Starting & Stoping the server 18](#_Toc524351046)

[Apache Tomcat 18](#_Toc524351047)

[What Java services are running? 19](#_Toc524351048)

[ElasticSearch 19](#_Toc524351049)

[Elastic Search Commands 19](#_Toc524351050)

[Saving Rules as a SQL script. Be sure to select SQL when exporting 20](#_Toc524351051)

[The Data: Industry Standard Airline Data 20](#_Toc524351052)

[APIS | Advance Passenger Information System | WCO/IATA/ICAO PASSENGER LIST MESSAGE (PAXLST) Version 3.0 20](#_Toc524351053)

[PNR | Passenger Name Record | PNRGOV version 13.1 21](#_Toc524351054)

[Useful Links 21](#_Toc524351055)

[Anticipated Message Volume 22](#_Toc524351056)

[Load Testing Results 24](#_Toc524351057)

[World Customs Organization and United States Customs and Border Protection Partnership 24](#_Toc524351058)

[ActiveMQ 24](#_Toc524351059)

[Indexed Fields 25](#_Toc524351060)

# Foreword

The deployment of an API/PNR screening system can take months to years. Often times these systems need to be redesigned and rewritten after critical flaws are discovered late in the software development lifecycle. The **Global Travel Assessment System** (GTAS) is built on a proven model that continues to produce results across the globe. It is free to customize and tailor to your agency’s needs and your country’s legislative requirements. It is meant to be a powerful tool in the border security toolkit, but not the only tool. You’re people and their collective experience are your greatest assets for guarding your borders and promoting legitimate travel.

The human element is critical to a successful border security agency. GTAS relies completely on two types of information to make it work; Advance Passenger Information (API) and Intelligence Information. The Intelligence Information must be provided by the agency/organization operating the application. Agents need to know the API data elements inside and out, and connect that information to their existing intel to generate rules and watchlists.

Many border security agencies are already using similar systems, however they are tedious, iterative, and manual. GTAS automates these tasks, freeing up valuable time for analysts to do more with what they know. Dozens of countries have implemented systems similar to GTAS, often times at a high cost not only because they’re compelled to by the United Nations, but because they’re worth it. The total lifecycle cost pales in comparison to the benefits, and GTAS shortens the time to deployment and reduces the overall cost burden.

There are several hurdles to implementing an API/PNR system. The legislative authority to collect the data is the biggest. Next is securing the Airline Data Agreements, either directly with individual airlines or with a third party data provider (SITA, Rockwell Collins, Amadeus, Sabre, etc.). This technical document is meant to help you over the technical hurdle of physically deploying the application.

This document contains everything you’ll need to know to deploy GTAS successfully.

# Ancillary Documents

1. Source Data Dictionary.xlsx (in github/US-CBP/GTAS repository)
2. PNR Specification (PNRGOV 13.1)
   1. <https://www.iata.org/iata/passenger-data-toolkit/assets/doc_library/04-pnr/PNRGOV%20EDIFACT%20Implementation%20Guide%2013_1.pdf>
3. APIS Specification UNEDIFACT PAXLIST 16B
   1. <http://www.unece.org/trade/untdid/d16b/tred/tred7059.htm>
4. GTAS Implementation Guide

# GTAS Installation / Upgrade

The installation process involves setting up an environment to build the application from its human-readable source code into binary form, and deploying it to a Tomcat server. There are two programs that need to run on the server; the **web application**, and the **job scheduler**. These programs are .war files. Installation also includes creating a database, and connecting it to GTAS. This walkthrough is for mac/Linux-based machines, but the high-level process is the same regardless of the operating system. GTAS has been designed for deployment on Linux, in the spirit of having a no-cost software stack. However, GTAS is a java application, and java applications are platform agnostic. It can run on Windows or any other operating system.

## Setting up the Environment

The following software is required to build and run the GTAS application.

**Java 8** - *development kit*

**Apache Tomcat 8** *- server*

**MariaDB 10.0 Stable** *- database*

**Maven 3.3 -** *software build management*

**Redis 4.0.9** *- in-memory data structure store, used as a database, cache and message broker*

Helpful instructions for installing Redis on a Mac

<https://medium.com/@djamaldg/install-use-redis-on-macos-sierra-432ab426640e>

**Apache ActiveMQ** *- message queue*

**ElasticSearch 2.3.2** (Optional, but highly recommended) *- free-text search tool*

**Sequel Pro** or **Heidi SQL** (database viewer, recommended)

## Environment

1. If this is an initial installation, install the following dependencies and skip to step 3. If this is an upgrade, proceed to step 2
   1. Java 8
   2. Apache Tomcat 8
   3. MariaDB 10.0 Stable
   4. Maven 3.3
   5. Redis 4.0.9
   6. Apache ActiveMQ
   7. Google Chrome (recommended)
   8. Sequel Pro or Heidi SQL (database viewer, recommended)
   9. npm
   10. yarn (package manager)
2. Remove existing gtas-job-scheduler.war and gtas.war files from the server. Remove existing directories from old installations.

rm /usr/local/apache-tomcat-8.0.47/webapps/gtas-job-scheduler.war

rm /usr/local/apache-tomcat-8.0.47/webapps/gtas.war

sudo rm -r /usr/local/gtas-parent

sudo rm -r /Users/david.j.ertel/GTAS

## 

## Download

1. Download the latest code, extract the file, and copy the gtas-parent folder into the desired directory

git clone --branch dev --single-branch https://github.com/US-CBP/GTAS.git

sudo mv /Users/david.j.ertel/GTAS/gtas-parent/ /usr/local/

## 

## Pre-build Configuration

There are a number of properties and configuration files that need to be updated prior to the build process. The creation of inbound and outbound message directories is required for the queue that handles messages. Inbound messages are messages that have been sent from the data provider to the receiver, and are ready for processing by GTAS. Outbound messages have been handled by the queue manager, and have begun the loading process. In this example, two directories have been created; one inbound (gtas\_in) and one outbound (gtas\_out).

1. Update .properties files to match appropriate inbound / outbound message directories. If you’re not sure what the address is for Redis, ping it and edit the address in this file to reflect the true Redis address. Please note that Windows requires two \\’s following a drive letter i.e. C:\\
   1. jobScheduler

Open /usr/local/gtas-parent/gtas-job-scheduler-war/src/main/resources/jobScheduler.properties

##################### MessageLoader #####################

inputType=two\_dirs

message.dir.origin=/data/gtas\_in

message.dir.processed=/data/gtas\_out

loader.fixedDelay.in.milliseconds=120000

loader.initialDelay.in.milliseconds=20000

loader.isRunning=false

maxNumofFiles=5

###JMS###

inbound.loader.jms.queue=GTAS\_INBOUND\_Q\_REDIS

outbound.loader.jms.queue=GTAS\_LOADER\_Q

#########

### REDIS ###

redis.connection.string=redis://0.0.0.0:6379

#############

##### Passenger ID Tag and BookingDetail Scheduler settings ####

cleanup.fixedDelay.in.milliseconds=120000

cleanup.initialDelay.in.milliseconds=60000

#########

##################### RuleRunner #####################

ruleRunner.fixedDelay.in.milliseconds=120000

ruleRunner.initialDelay.in.milliseconds=50000

ruleRunner.isRunning=false

## Build

1. Set paths

M2\_HOME=/usr/local/apache-maven-3.3.9

M2=$M2\_HOME/bin

export PATH=$M2:$PATH

1. Change directory to parent and clean install

cd /usr/local/gtas-parent

mvn clean install

OR install without tests

cd /usr/local/gtas-parent

mvn clean install -Dskip.unit.tests=true

## Deploy

1. Log into mysql (makes sure it’s running. If you get an error, see script to start msyql)

mysql –u root –p

(password: admin)

if error, try this. This depends on where mysql is installed.

sudo /usr/local/bin/mysql.server start

or you may need to specify the database at login (i.e.)

mysql -u root -p -h127.0.0.1

1. If new installation, skip to step 9. If upgrading:

use gtas\g

drop database gtas;

1. Create database

create database gtas;

\q

1. Run hibernate to script the database, populate tables (airports, countries, users, etc.)

cd /usr/local/gtas-parent/gtas-commons

mvn hibernate4:export

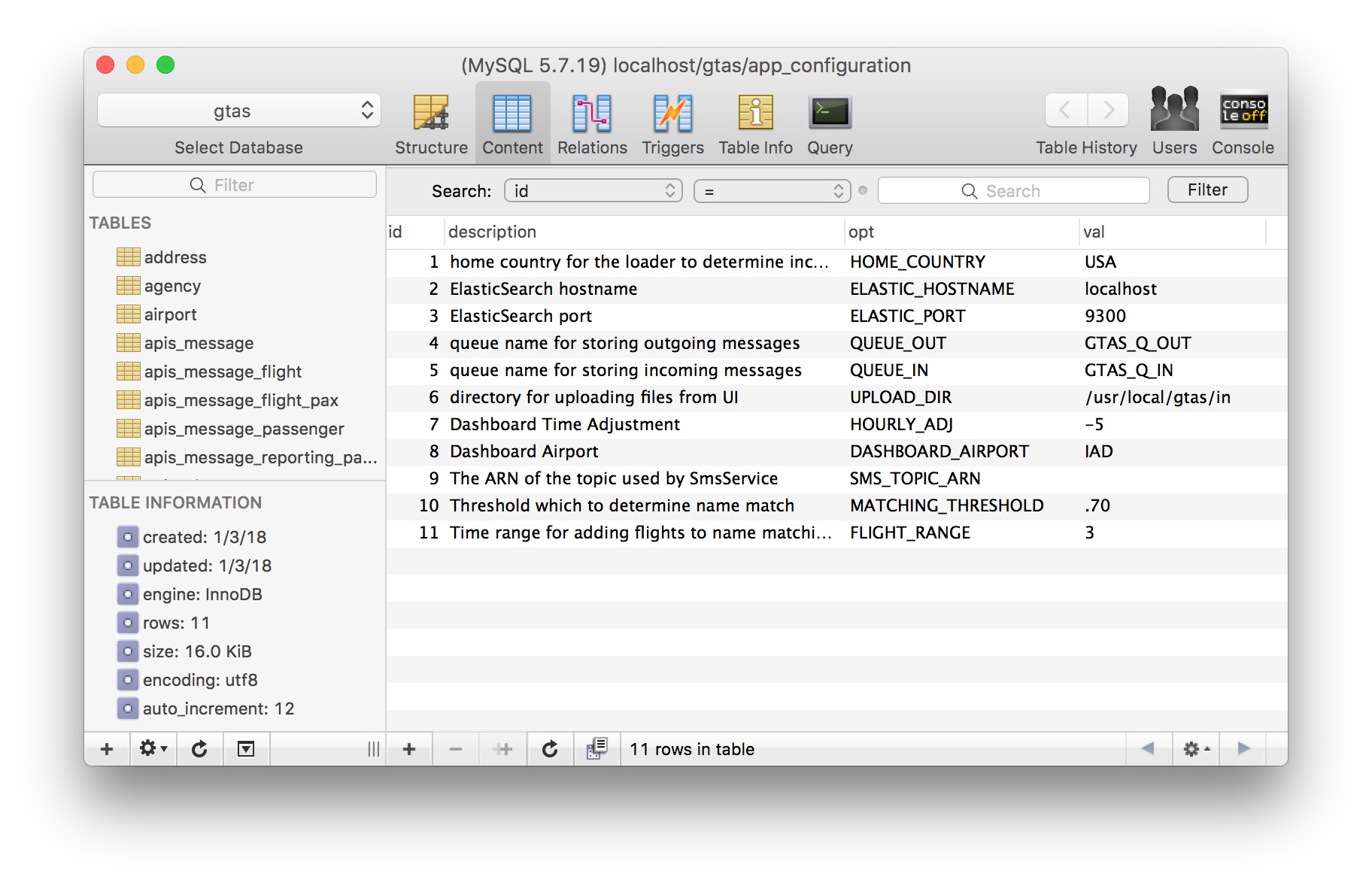
1. Move the .war files to the Tomcat server

cp /usr/local/gtas-parent/gtas-webapp/target/gtas.war /usr/local/apache-tomcat-8.0.47/webapps

cp /usr/local/gtas-parent/gtas-job-scheduler-war/target/gtas-job-scheduler.war /usr/local/apache-tomcat-8.0.47/webapps

## Configure: Set country and UTC offset

Update the app\_configuration table to reflect desired drag-and-drop folder, installation country etc. Database Table\_App



1. Update the javascript files to customize the dashboard for your country/organization
   1. Go to: /gtas-parent/gtas-webapp/src/main/webapp/dashboard/DashboardController.js

"title": "IAD",

var map = AmCharts.makeChart("chartdiv", {

"type": "map",

"theme": "light",

"dataProvider": {

"map": "worldLow",

"linkToObject": "WashDC",

"images": [{

"id": "WashDC",

"color": "#000000",

"svgPath": targetSVG,

"title": "EBB",

"latitude": anchorLat[0],

"longitude": anchorLong[0],

"scale": 1.5,

"zoomLevel": 5,

"zoomLongitude": 42.5903,

"zoomLatitude": 11.8251,

This section sets the anchor point and the level for the zooming animation and give the title to the home airport icon, which is highlighted. The location for the home airport icon is in a subsequent step.

* i.e. Djibouti is a good midpoint reference between AUH and EBB
* A zoom level of 5 is good for the pilot since it only needs to capture that one flight route
  1. In the same file, locate this section;

"label": "Flights To IAD",

"images": [{

"label": "Flights To EBB",

"svgPath": planeSVG,

"left": 100,

"top": 45,

"labelShiftY": 5,

"color": "#D35400",

"labelColor": "#D35400",

"labelRollOverColor": "#D35400",

"labelFontSize": 20

This section configures the message on the label for the amchart on the dashboard.

* 1. In the same file, locate this section

var arrOfLats = [];

var arrOfLongs = [];

var imagesArrOfAirports = [];

var anchorLat = [0.043600];

var anchorLong = [32.441800];

This section above sets the coordinates for the Home airport anchor

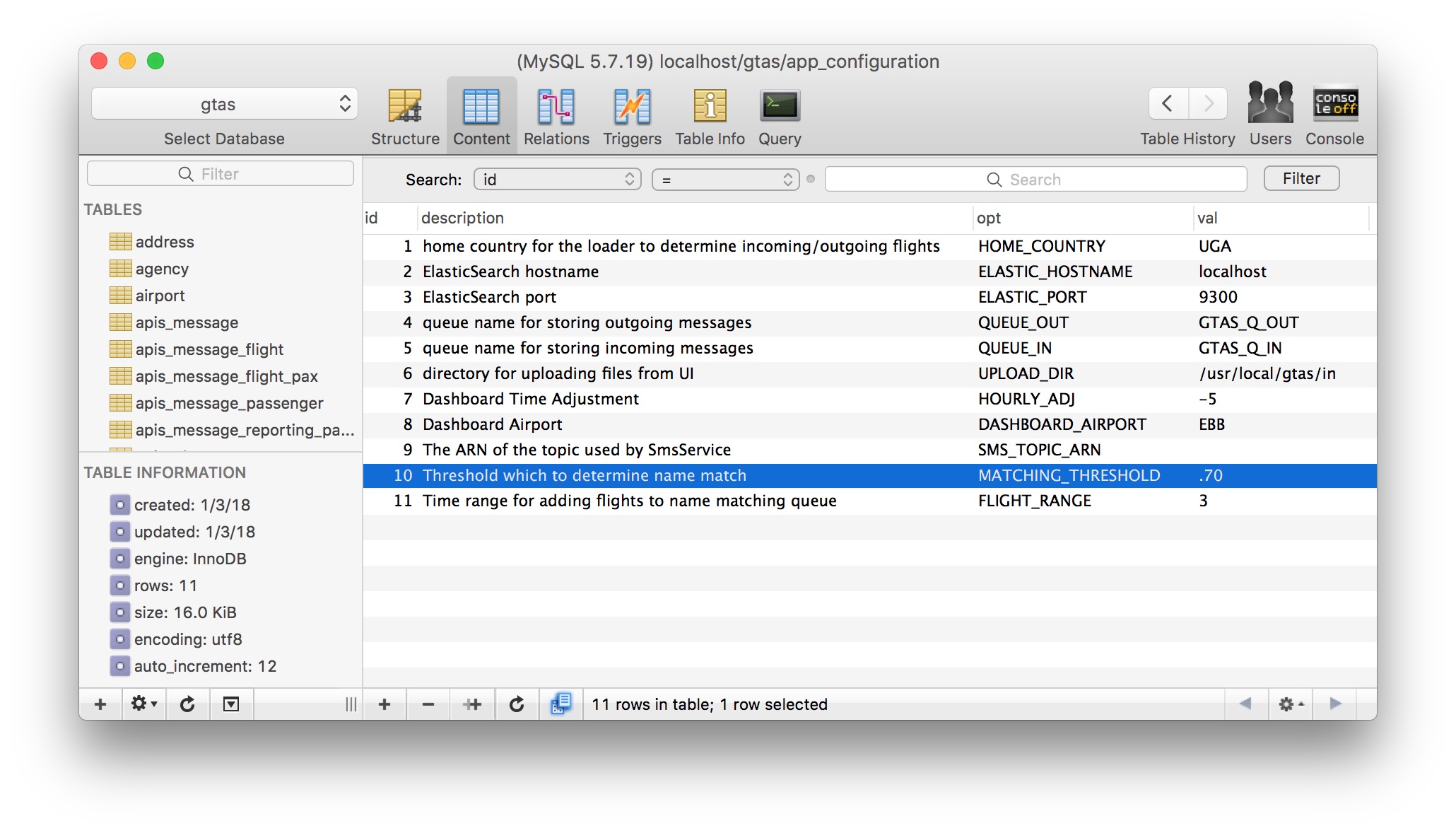
* For latitude, Positive is North, Negative is South
* For Longitude Positive is East, Negative is West
  + i.e. Latitude for Entebbe airport is 0.043600
  + i.e. Longitude is 32.441800

## Configuring the Jaro-Winkler Distance and Double Metaphone Watchlist matching thresholds (optional)

This can be configured from the front-end through the admin settings.

The watchlist service will automatically generate hits if the first and last names of a given traveler are determined to be phonetic matches to an entry on the watchlist. The Double Metaphone algorithm matches words based on approximations of their sounds, so this portion is not configurable. If a match is determined, the Jaro-Winkler Distance portion of the watchlist service will not run.

The watchlist service will also generate hits if the Jaro-Winkler distance algorithm determines a text string-based match to a watchlist record. This threshold is configurable, and is highlighted in the table below.



## Starting the ActiveMQ, Redis, ElasticSearch, and the Tomcat Server

1. Start ActiveMQ

/usr/local/apache-activemq-5.15.3/bin/macosx/activemq start

1. Start Redis

cd /usr/local/redis-4.0.9

cd src

./redis-server

1. Turn off “protected mode” in Redis (in new terminal window)

redis-cli

CONFIG SET protected-mode no

1. Start ElasticSearch

/usr/local/elasticsearch-2.3.2/bin/elasticsearch

1. Start Tomcat server with log

/usr/local/apache-tomcat-8.0.47/bin/catalina.sh run

1. Clear browser cache if performing an upgrade (command + shift + r in google chrome)
2. Access site <http://localhost:8080/gtas>

## Troubleshooting Immediately after Installation

This section covers common issues seen immediately after installation.

### I can’t see any flights!

This issue is only seen in some mac installations that were installed for development purposes. It has to do with an odd sql mode setting. Most developers have Heidi SQL installed, and are able to run this query script to correct the issue. This error can be detected in the system log, and requires restarting the Tomcat server after script execution.

**set** sql\_mode = "";

**SET** **GLOBAL** sql\_mode = '';

**select** @@**GLOBAL**.sql\_mode="";

**SET** sql\_mode = ''

**select** @@**GLOBAL**.sql\_mode;

### I can’t access <http://localhost:8080/gtas>

Some IT policies will recognize port 8080 as being unusual and will block it. This is the default port set by Apache Tomcat.

# Redis

GTAS uses Redis to prevent redundant message payloads from hitting the entire loading process. Redis creates a hash code, or “ key” for each message received, and uses an in-memory store to save the keys for a designated period of time (currently seven days). If the key already exists, the system knows the message should be disregarded, since it doesn’t contain any valuable updates. The definition of the payload is configurable, and will need some tuning with each production deployment for regional differences and variation from carrier-to-carrier.

## Redis Commands

Delete all stored keys

redis-cli

fushall

# Amazon Web Services (AWS) Deployment

Cloud formation scripts are available for GTAS deployment.

# Installation of Message Queue Setup (Apache ActiveMQ)

Download the lastest zip file

Unzip it

Move it to desired directory

Cd to bin directory

Activemq start

Localhost:8161

# Installation Backup

## Browser Requirements

GTAS requires a modern browser that supports angular. A full list of browsers supported can be found at:

https://v2.angular.io/docs/ts/latest/guide/browser-support.htm

## ElasticSearch

### Installation

Running ElasticSearch for GTAS is an optional component that adds fast, free-text field searching on indexed data elements. It also allows the user to perform basic link analysis, and view connections visually through a force-directed graph. GTAS currently requires ElasticSearch 2.3.2 to work properly. It can be downloaded at <https://www.elastic.co/downloads/past-releases/elasticsearch-2-3-2> . ElasticSearch must be running prior to message loading in order for it to be indexed, and searchable by the end-user.

### Useful Commands

To view a list of the fields currently indexed, run the following script when ElasticSearch is running.

curl -XGET 'localhost:9200/\_all/\_mapping?pretty'

Clear ElasticSearch Cache. This will clear the entire cache of indexed values. All information previously stored by ElasticSearch will be lost.

curl -XDELETE localhost:9200/\*

# Rest API | External Queries

GTAS has a feature where a user with admin privileges can authorize a user to query the database for the installation. This functionality is useful if an organization using GTAS wishes to share passenger and flight information with a partner organization. This information is returned in the form of a JSON object for easy ingestion into the partner’s system, or even into a separate installation of GTAS.

cd /usr/local/gtas-rest-api/target/

java -jar gtas-rest-api-1.jar

# Security

## Digitally-Signed .war Files

The .war files can be digitally signed to determine where its source code came from

## (Coming Soon) Configuring Tomcat to support SSL or https

## (Coming Soon) Industry Standard Security Checks

* Need a license scanner for compliance violations
* Fossology?
* MojoHaus?

# System Design

GTAS is based on ingesting industry-standard airline messages, updating a relational database with the most current information, and presenting it in a manageable form for the user. Fields with one-to-many relationships will be updated to include the latest information, but will not delete records if they are not present in the latest message. The latest message is determined by the order in which it enters the ETL process. It is critical to configure the message queue so that messages arrive in the “in” folder in the correct order. (there is some question about the systems’ ability to do this, and is being investigated)

### Drag and drop file uploader configuration

Configuring the drag-and-drop file uploader to point at the proper directories requires two operations that are explained in detail within the installation instructions

1. editing the jobscheduler.properties file prior to the build process. It is located in the directory:

/usr/local/gtas-parent/gtas-job-scheduler-war/src/main/resources

1. Updating the “directory for uploading files form UI” record of the app\_configuration table in the database to reflect the desired directory (i.e. /usr/local/gtas/in)

# Miscellaneous Commands

## Tomcat Server Commands

### Starting & Stoping the server

/usr/local/apache-tomcat-8.0.47/bin/catalina.sh start

and

/usr/local/apache-tomcat-8.0.47/bin/catalina.sh stop

Running the server in the same window it was kicked-off in:

/usr/local/apache-tomcat-8.0.47/bin/catalina.sh run

# Apache Tomcat

/usr/local/apache-tomcat-8.5.15/bin/catalina.sh run

/usr/local/apache-tomcat-8.5.15/bin/catalina.sh stop

### What Java services are running?

Ps -el | grep java

## ElasticSearch

### Elastic Search Commands

Kicking off ElasticSearch

/usr/local/elasticsearch-2.3.2/bin/elasticsearch

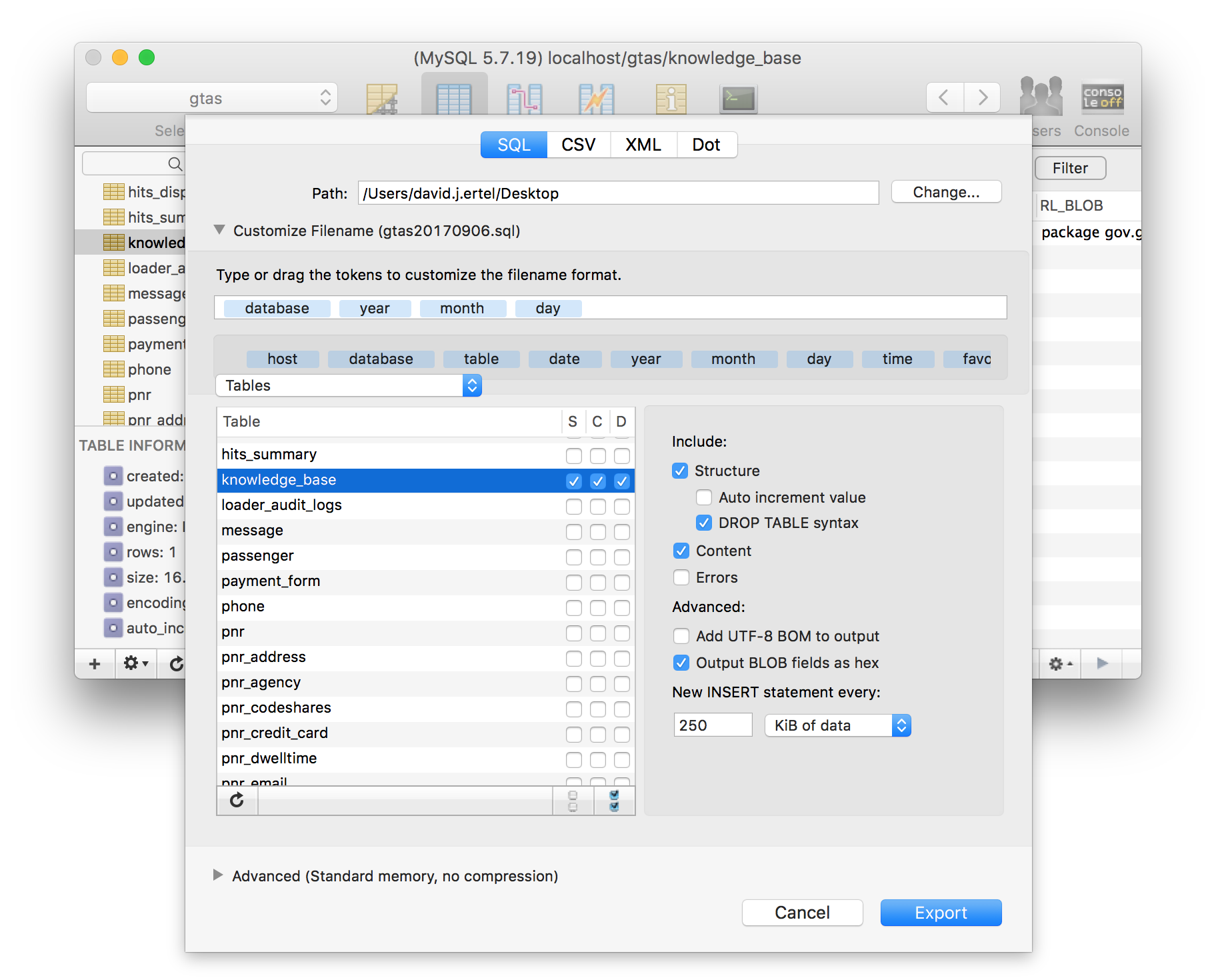
Identifying all indexed fields in ElasticSearch

curl -XGET 'localhost:9200/\_all/\_mapping?pretty'

Clearing out entire ElasticSearch Cache

curl -XDELETE localhost:9200/\*

## Saving Rules as a SQL script. Be sure to select SQL when exporting



There are four key tables that store rules:

* knowledge\_base
* rule
* rule\_meta
* udr\_rule

rule

rule category

rule\_meta

udr\_rule

udr\_rule\_cat

# The Data: Industry Standard Airline Data

Please refer to the latest Entity Relationship Diagram and related spreadsheet for the source and description of Data Elements included in APIS and PNR messages

GTAS relies completely on two types of data to make it work; Industry Standard Airline Data and Intelligence Data provided by the agency/organization operating the application.

## APIS | Advance Passenger Information System | WCO/IATA/ICAO PASSENGER LIST MESSAGE (PAXLST) Version 3.0

The system has been designed to ingest WCO/IATA/ICAO PASSENGER LIST MESSAGE (PAXLST) Version 3.0, which is a subset of the UN/EDIFACT PAXLST message. GTAS supports the most commonly used message types–02B, and 05B.

## PNR | Passenger Name Record | PNRGOV version 13.1

The system has been designed to ingest PNRGOV version 13.1 messages. PNR messages are a rich data source, and are critical to passenger screening and analysis.

PNRGOV (Element 1225 = 22) Push PNR data to States

PNRGOV (Element 1225 = 141) Update (used for update push)

The second message (24 hours prior to departure) in a sequence of five PNR pushes is not considered an update from the first message (72 hours prior to departure). If the sender updates information that applies to the second push, and resends it, would be considered an update (both messages are for 24 hours prior to departure).

Interchange Control Reference

This is the ID for the series of messages pertaining to that particular reservation.

Specification Documents

## Useful Links

IATA toolkit

<http://www.iata.org/iata/passenger-data-toolkit/presentation.html>

<https://www.eraa.org/policy/security/advance-passenger-information-api-and-passenger-notifications-records-pnr>

<http://www.iata.org/publications/pages/api-pnr-toolkit.aspx>

<https://www.icao.int/Security/FAL/ANNEX9/Pages/Publications.aspx>

<https://www.icao.int/Security/FAL/SitePages/API%20Guidelines%20and%20PNR%20Reporting%20Standards.aspx>

<https://store.icao.int/index.php/guidelines-on-passenger-name-record-pnr-data-doc-9944-english-printed.html>

Link to the guide from the “API-PNR” toolkit, the guide costs $31 USD

<https://www.icao.int/Security/FAL/Documents/1.API%20Guidelines%202013%20Main_%20Text_E.pdf>

https://www.eraa.org/system/files/directive\_681-2016\_on\_pnr\_-\_final.pdf

Notoriety

<http://www.wcoomd.org/en/media/newsroom/2017/december/wco-and-uganda-revenue-authority-launch-global-travel-assessment-system-gtas-pilot.aspx> - Article

<https://www.cbp.gov/frontline/cbp-national-targeting-center> - Article

## Anticipated Message Volume

It is critical to understand your country’s expected air travel volume. Every passenger taking off or landing in your country will produce X number of messages depending on the arrangements with airlines and/or third party data providers. This message volume can grow exponentially depending on the message types and periodicity of transmission chosen. Be mindful of the rate that your system can ingest data and run rules. Have a plan for what do when feed outages occur and the message queue backs up. It may not recover for hours, or possibly days depending on the nature and severity. Anything from weather to changing internet service providers and impact connections and queues.

A considerable number of the messages ingested by any API/PNR system contain no new information. GTAS hashes every message to determine uniqueness and message determined to have redundant contents to a previously encountered message will not be parsed or stored in the database.

GTAS in its current state can easily load the volume of an APIS-only implementation, even for countries with high travel volume. A typical APIS-only implementation receives one APIS message per flight, and is known as a “One-Times-APIS” system.

A typical APIS+PNR implementation will have considerably higher volume. The following example will illustrate the volume and complexity of an API/PNR system

|  |  |
| --- | --- |
| **Order Received** | **Description** |
| 1 | 72 Hours Prior To Departure |
| 2 | 24 Hours Prior To Departure |
| 3 | 2 Hours Prior To Departure |
| 4 | 1 Hour Prior To Departure |
| 5 | Wheels Up |

This table outlines a sample periodicity of five PNR pushes, which will happen for each distinct flight reservation. This is known as a “Five-Times-PNR” system. Some countries use up to seven pushes. Some use fewer.

A hypothetical PNR reservation with this system will have five periodic pushes. Depending on the number of travelers on a reservation, this could be up to 1,250 messages for a single flight if there are 250 passengers that made reservations without co-travelers.

A more reasonable expectation of how many messages to expect would be about 840 messages for a 250 passenger flight, given the passengers are distributed among reservations in the table below. This exercise is for illustrative purposes, and is hypothetical. There is no statistical basis behind the distribution of passengers per reservation. A sample of production data through a pilot installation is the easiest way to gauge volume. Several factors must be considered with the sample, and it’s best to consult with an expert to account for seasonal highs and lows. Another possible solution is to start with your national carrier or a specific flight route.

A One-Time-APIS+Five-Times-PNR system vs. APIS-Only implementation would have an 841:1 volume ratio comparison if every flight fit this model. This certainly doesn’t account for last minute reservations, cancelled reservations, cancelled flights etc.

|  |  |  |  |
| --- | --- | --- | --- |
| **Passengers Per Reservation** | **Probability** | **Reservations** | **Messages** |
| 1 | 50.00% | 125.0 | 625 |
| 2 | 20.00% | 25.0 | 125 |
| 3 | 10.00% | 8.3 | 42 |
| 4 | 7.50% | 4.7 | 23 |
| 5 | 5.00% | 2.5 | 13 |
| 6 | 4.50% | 1.9 | 9 |
| 7 | 1.75% | 0.6 | 3 |
| 8 | 1.25% | 0.4 | 2 |

The expected load for a given day under the same assumptions is 67,280 Messages for a One-Time-APIS+Five-Times-PNR system. This equates to about 7.3 million travelers per year at the same rate.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **4 AM** | **5 AM** | **6 AM** | **7 AM** | **8 AM** | **9 AM** | **10 AM** | **11 AM** | **12 PM** | **1 PM** | **2 PM** | **3 PM** | **4 PM** | **5 PM** | **6 PM** | **7 PM** | **8 PM** | **9 PM** |
| 840 | 1,680 | 4,205 | 4,205 | 4,205 | 4,205 | 4,205 | 4,205 | 4,205 | 4,205 | 4,205 | 4,205 | 4,205 | 4,205 | 4,205 | 4,205 | 3,365 | 2,525 |

It should be noted that most international airlines allow passengers to answer security-related questions as well as enter travel documentation as soon as 48 hours before departure. This information will certainly change the payload of the PNR messages from the previous Push.

## Load Testing Results

# World Customs Organization and United States Customs and Border Protection Partnership

The Global Travel Assessment System source code is developed and maintained through a partnership between the World Customs Organization (WCO) and United States Customs and Border Protection (CBP). Various other government agencies have offered their support to the GTAS project as well. GTAS is a WCO product.

## ActiveMQ

ActiveMQ is the Open Source alternative to IBM MQ software. Messages tend to be about 4 Kb or less. It is expected that an hour’s worth of messages should be transferred in a few seconds.

The performance stats can be found at <http://activemq.apache.org/performance.html>

## Indexed Fields