

# Distributed Processing Program

---

## Prerequisite

1. java >1.8
2. Make sure that java and javac is in path
3. Tensor flow (if want to run with simulation off) (Installation instruction is below)

## Build

1. Run `build.bat`

## Install Tensor flow models

If you pull from github, no download is needed. Skip all step to [Run](#) section. One of the service provided is to run Tensor Flow Object Detection. This will require the tensor flow pre-trained models and labels. To download them run `download_models.ps1` or `download_models.sh`. To save time, you can also turn on simulation so that the it does not actually run the tensor flow command but return a hard coded value. See configuration below to find out how to turn it on.

## Configuration

If a configuration file is not found. First startup will create default config file.

### Tracker

File: tracker.config.properties

Config	Definition	Default
rmi_registry_port	Tracker RMI Port	1099

### Server

File: server.config.properties

Config	Definition	Default
image_analytics_simulate	Turn simulate on or off see <a href="#">Install Tensor flow models</a>	0
services	Services that this server will provide. Split by comma (,)	VideoAnalytics,VideoSplit,ImageAnalytics,ImageAnalyticsGraph
image_analytics_label	The label file for tensor flow. Not needed if simulation is 1. Or if the server is not providing ImageAnalyticsService.	labels/mscoco_label_map.pbtxt

Config	Definition	Default
image_analytics_model_dir	The model directory for tensor flow. Not needed if simulation is 1. Or if the server is not providing ImageAnalyticsService.	models/ssd_inception_v2_coco_2017_11_17/saved_model
tracker	The tracker server	localhost:1099
rmi_registry_port	The server rmi registry port	1000
rmi_registry_host	The server rmi registry host/IP	localhost
ffmpeg_command	The server ffmpeg command to run. Not needed if not providing VideoSplitService.	./ffmpeg

## Client

File: server.config.properties

Config	Definition	Default
rmi_registry_port	Callback RMI Registry Port	1088
rmi_registry_host	Callback RMI Registry Host/IP	localhost
trackers	Tracker servers to find servers. Comma seperated (,)	localhost:1099

## Run

### Run Tracker

Go to `bin\tracker` and run `java -jar tracker.jar` in command line / terminal

### Run Server

Go to `bin\server` and run `java -cp server.jar;xchart-3.5.2.jar Server` for windows in command line. Run `java -cp server.jar:xchart-3.5.2.jar Server` for linux in terminal

### Run Client

Go to `bin\client` and run `java -jar client.jar` in command line / terminal

## Acknowledgements & License

### TensorFlow demo Program

The ImageAnalytics potion of the program uses an external jar application to do the analytics. Mainly the TensorFlow demo program that can be found at

[https://github.com/tensorflow/models/tree/master/samples/languages/java/object\\_detection](https://github.com/tensorflow/models/tree/master/samples/languages/java/object_detection)

Since the jar demo application does not provide an easy way to extract info out, we modified the code to output a CSV file. It's then package as a standalone JAR file. A copy of the license can be found at lib/detect-object-LICENSE

## Xchart

The ImageAnalyticsGraph part of the program uses an external library called xchart <https://knowm.org/open-source/xchart/>. No modification is made on the source. A copy of the license can be found at lib/xchart-LICENSE

## ffmpeg

The VideoSplit part of the program uses an external application called ffmpeg <https://www.ffmpeg.org/> to split videos to multiple images.

## 7za

No part of the application is using 7za.exe but it is used for the `download_models.ps1` script.