**Asteroid Data Hunter - Front End Services Module Assembly**

**Deployment Guide**

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Author** | **Revision Number** | **Date** |
| TCSASSEMBLER | 1.0 | 11.09.2014 |
|  |  |  |
|  |  |  |

[Deployment Instructions](#h.30j0zll)

1. Organization of Submission

[2. Build Tools and Dependencies](#h.2et92p0)

[3. Application Setup](#h.tyjcwt)

[3.1. Set up the detector (algo)](#h.3dy6vkm)

[3.2. Setup H2 Database](#h.1t3h5sf)

[3.3. Compile the tester jar file](#h.4d34og8)

[3.4. Training Data (Optional Step)](#h.2s8eyo1)

[4. Configurations](#h.17dp8vu)

[5. Deploy the app](#h.3rdcrjn)

[6. Verification](#h.26in1rg)

[7. Resource Contact List](#h.lnxbz9)

**Deployment Instructions**

1. **Organization of Submission**

|  |  |
| --- | --- |
| **Path** | **Description** |
| docs | Contains the Deployment Guide (this documentation) |
| algo/linux | Contains the source code of c++ detector program for linux |
| algo/windows | The c++ detector program for Windows |
| algo/dist | The place you put the algo executable file. |
| hunter | Contains the source of hunter (The main module of this assembly) |
| tester | Contains the source of tester (use for testing and training). |
| run-jetty | The standalone runnable jetty program. |

1. **Build Tools and Dependencies**

**Platform**

This application must deploy and run on a Linux OS or Windows. I am using ubuntu 12.04.

**maven** 3.1.1 or above is used to build this app.

<http://maven.apache.org/docs/3.1.1/release-notes.html>

**jdk** 6 must be used

http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html

**g++** 4.8 or above

g++ 4.8 or above is used to build the c++ program. (the algo).

**Visual Studio 2013**

Use to compile the detector in Windows.

**Java Dependencies**: (Here only lists the main dependencies, for more detail, you can refer to pom.xml)

* Spring 3.2.8 <http://www.springsource.org/>

It is used to manage the configuration, and to provide the frontend MVC support.

Apache License 2

* Log4j 1.2.17 <http://logging.apache.org/log4j/1.2/>

It is used to perform logging.

Apache License 2

* Hibernate 4.3.6 <http://www.hibernate.org/>

It is used as the JPA provider that provides access to embedded H2 database.

LGPL

* H2 Database 1.3.176 <http://www.h2database.com/>

It will be used as embedded database.

**C++ Dependencies**

* BOOST

<http://www.boost.org/>

1. **Application Setup**

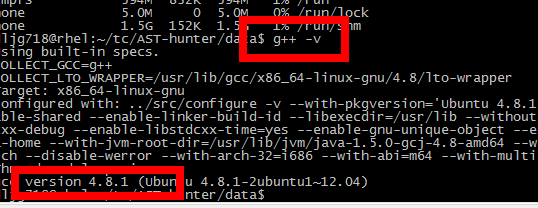
If you do not want to build the detector and use the detector program in my submission, you can ignore this section.

* 1. **Set up the detector (algo) in Linux**

a). Make sure your g++ version is 4.8 or above.

Run g++ -v,

You have to make sure your g++ version is 4.8.1 or above.



If not, here are the tips for you to install:

<http://askubuntu.com/questions/271388/how-to-install-gcc-4-8>

(Remember to replace the old gcc file with your new install one).

b) Install cmake

sudo apt-get install cmake

c) Install boost

sudo apt-get install libboost-all-dev

sudo apt-get install aptitude

aptitude search boost

d). compile the detector

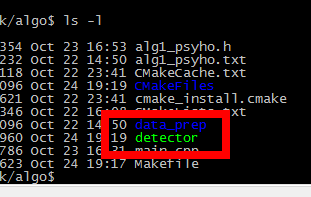
cd algo/linux

cmake CMakeLists.txt

make clean

make

You should see there will be an executable file named “detector” compiled:



e). copy this “detector” executable file to algo/dist with the file name “detector”

* 1. **Set up the detector (algo) in Windows**

1. Install Boost in windows

a). download boost:

<http://superb-dca2.dl.sourceforge.net/project/boost/boost/1.56.0/boost_1_56_0.zip>

b). unzip the file.

c). run boostrap.bat

d) run

b2.exe --toolset=msvc variant=release link=static threading=multi runtime-link=static stage

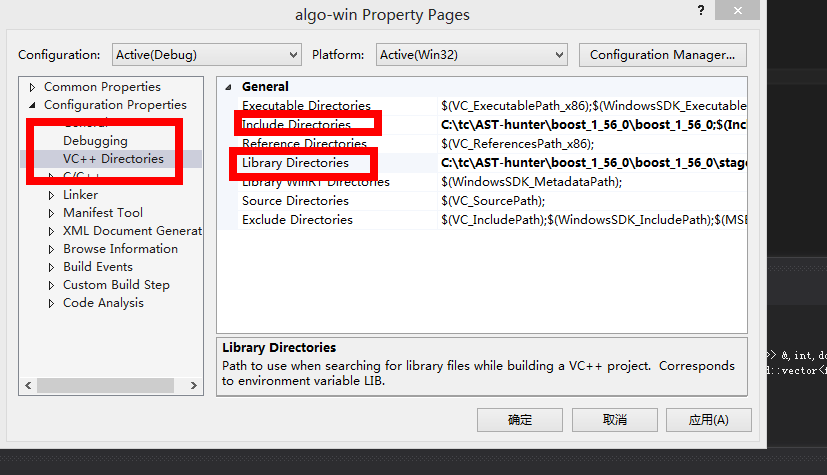
e) wait the compilation done

(note, you may need to open VS tool command first)

1. Open the project in VS: algo/windows/algo-win.vcxproj

(make sure you are using VS 2013)

1. Add the boost include path and boots lib path:

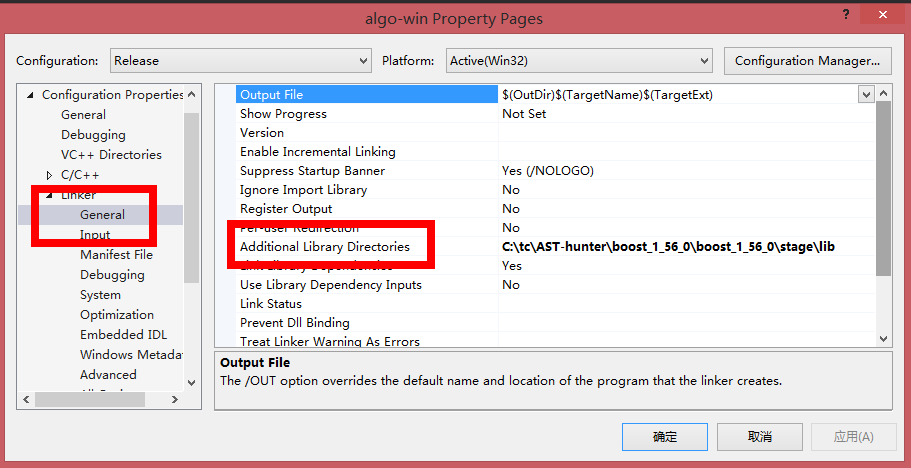
In “PROJECT” -> “algo win properties” -> VC++ Directories  


Here is my example:

Include path: C:\tc\AST-hunter\boost\_1\_56\_0\boost\_1\_56\_0\

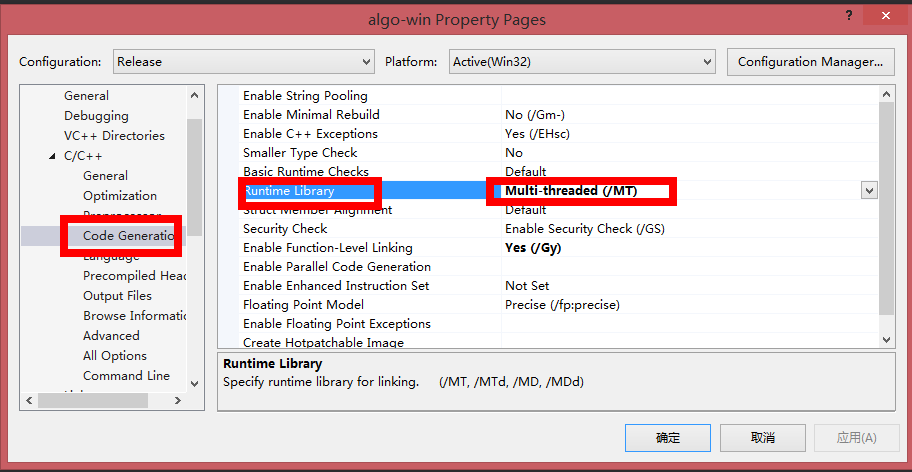
Lib path: C:\tc\AST-hunter\boost\_1\_56\_0\boost\_1\_56\_0\stage\lib

Also add the Lib Path to: Linker -> General -> Additional Library Directories

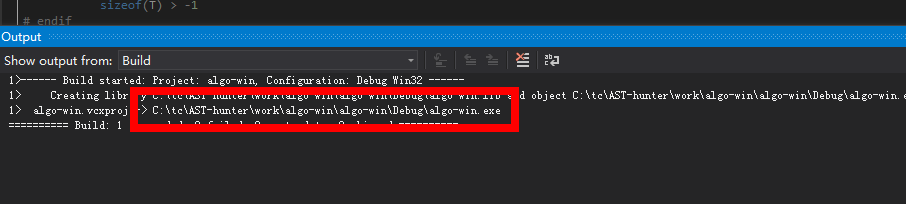


We choose static builds, so:

In c/c++ -> Code Generation -> Runtime Library, we choose **Multi-threaded (/MT)**



BUILD the solution, you should see a built exe file generated:



Copy the file to algo/dist, with the file name: detector.exe

That means: algo/dist/detector.exe is the detector file.

The detector may use a lot of memories. So, use JDK 64 is better.

* 1. **Training Data (Optional Step)**

I have already provided trained data in folder:

hunter/src/main/resources/trained-data

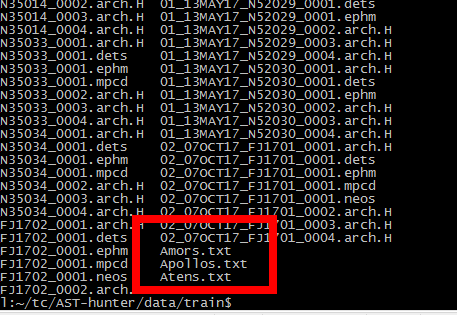
I only trained with 5 data set for test purpose. If you want to train it with your own data, steps are:

1. Download the train data. For example, use: algo/data\_prep/wget\_traindata.sh

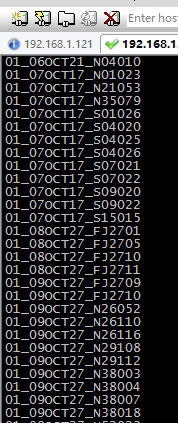
b). Put the downloaded files to a folder, let’s called this FOLDER A.

c). Copy the hunter/src/main/resources/trained-data/\*.txt to FOLDER A.

i.e, FOLDER A contains 2 kinds of files, the training data and the \*.txt files. Like:

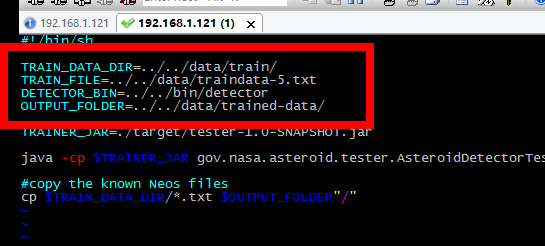


d). prepare your traindata.txt file. This file contains many lines(abitrary), each line is the name of the data set. (Make sure the data set data in the traindata.txt are already downloaded in step a). The file looks like:



e). Modify the configuration in:

tester/train.sh



TRAIN\_DATA\_DIR is the dir that store the data to train (i.e, FOLDER A)

TRAIN\_FILE is the path of your traindata.txt file.

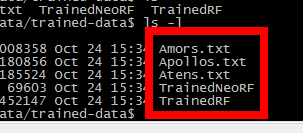
DETECTOR\_BIN is the executable file of the detector (i.e, the algo c++ program)

OUTPUT\_FOLDER is the the place to store the output trained data. (it must be an existing folder).

f). Then run:

./train.sh

After it finished, you should see the trained files are generated in your output folder:



g). copy the generated files in your output folder to:

hunter/src/main/resources/trained-data

and replace the existing files.

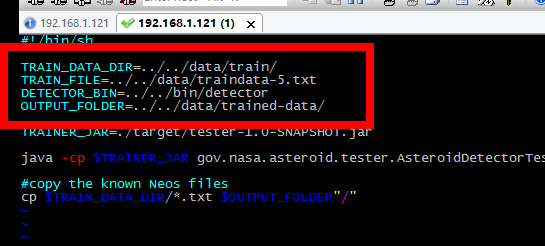
1. **Configurations**
2. The Hunter configuration:

hunter/src/main/resources/applicationContext.properties

|  |  |
| --- | --- |
| jdbc.driverClassName | The jdbc driver |
| ~~jdbc.url~~ | ~~The jdbc url~~ |
| jdbc.username | The database user name |
| jdbc.password | The database password |
|  |  |
| ~~detector.asteroids\_detection\_exe~~ | ~~The executable detector program path~~ |
| ~~detector.base\_directory~~ | ~~The base directory (the work directory when testing)~~ |
| detector.max\_number\_of\_answers | The max number of answers we accept from the detector. |

1. The trainer Configuration (optional)

tester/train.sh



TRAIN\_DATA\_DIR is the dir that store the data to train

TRAIN\_FILE is the path of your traindata.txt file.

DETECTOR\_BIN is the executable file of the detector (i.e, the algo c++ program)

OUTPUT\_FOLDER is the the place to store the output trained data. (it must be an existing folder).

1. spring-dispatcher.xml

You can config the static content’s cache:

<mvc:interceptor>

<mvc:mapping path="/\*\*"/>

<bean id="webContentInterceptor" class="org.springframework.web.servlet.mvc.WebContentInterceptor">

<property name="cacheSeconds" value="36000"/>

</bean>

</mvc:interceptor>

We choose 10 hours for the image cache.

1. **Deploy the app**
   1. **Deploy the app without installer**

For linux:

#1 Build the app:

build.sh

#2 run the app:

dist/start.sh

For windows:

#1 Build the app:

build.bat

#2 run the app:

dist/start.bat

It will open your browser and visit:

<http://localhost:8080/dashboard>

* 1. **Build the Installer**

You can build the installer:

For linux, run:

build.sh installer

For Windows, run:

build.bat installer

You can see that, a folder name **installer-dist** is generated. That folder is the installer, you can zip it and deploy it to the user who wants to install the app.

Note, before building the installer, you must make sure the system environment JAVA\_HOME is properly set and point to your JDK.

1. **Verification**

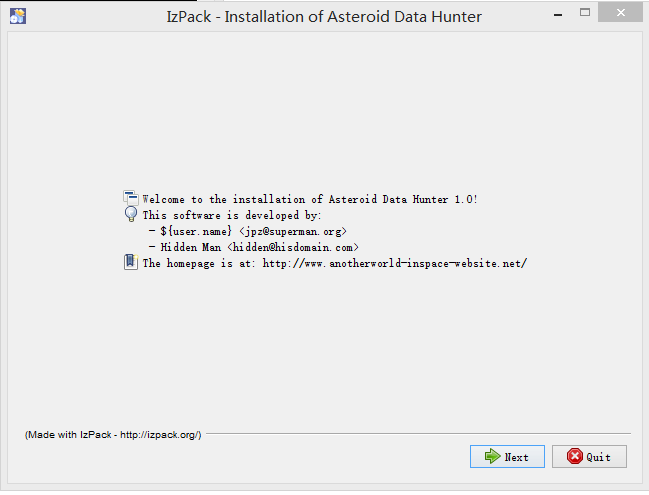
**Installer**

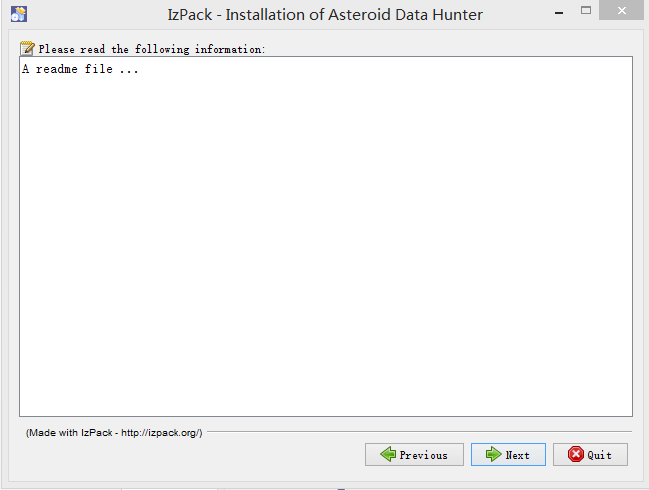
Follow the 5.2 to build the installer.

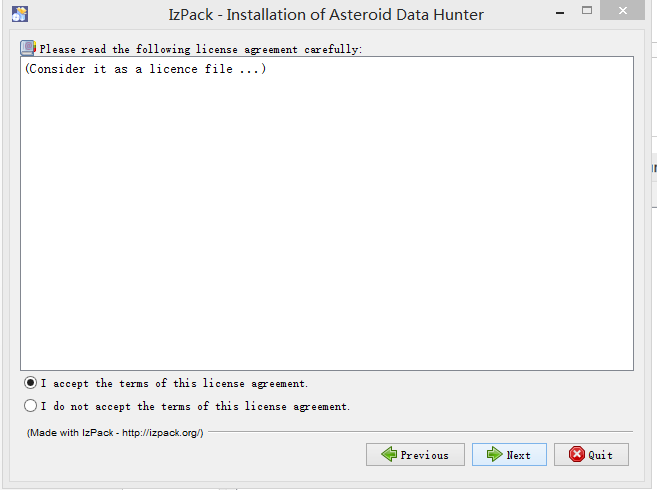
In the generated folder **installer-dist**, run:

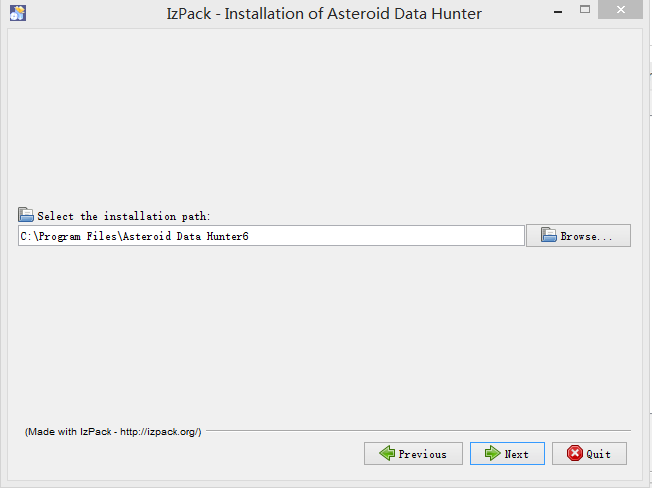
installer.bat (Windows) or install.sh (Linux)

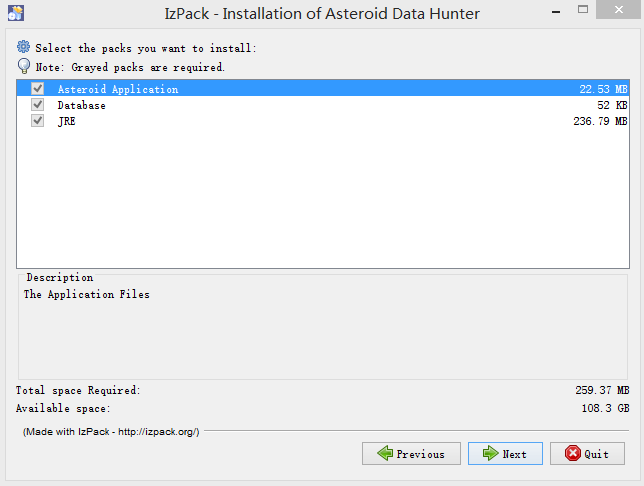
The installer will start:











Start the app:

In the install path, run start.bat (for windows) or start.sh (for Linux)

Then the app will start working.

**Bug Fixes**

Check:

<https://apps.topcoder.com/bugs/browse/BUGHUNTDATAHUNTER>

My comments are under the tickets.

1. **Resource Contact List**

|  |  |
| --- | --- |
| **Name** | **Resource Email** |
| TCSASSEMBLER | Use contact option from tc profile |