

Dec 05, 18 12:22

build.bat

Page 1/1

```
mkdir bin
mkdir bin\tracker
mkdir bin\server
mkdir bin\client
cd src\tracker
call build.bat
if %errorlevel% neq 0 exit /b %errorlevel%
copy bin\tracker.jar ..\..\bin\tracker\
cd ..\
cd server
call build.bat
if %errorlevel% neq 0 exit /b %errorlevel%
copy bin\server.jar ..\..\bin\server\
cd ..\
cd client
call build.bat
if %errorlevel% neq 0 exit /b %errorlevel%
copy bin\client.jar ..\..\bin\client\
cd ..\..\

REM Copy Lib files
copy /Y lib\*.jar bin\server\
copy /Y tools\ffmpeg.exe bin\server\
```

Dec 05, 18 18:52

build.sh

Page 1/1

```
set -e
mkdir -p bin
mkdir -p bin/tracker
mkdir -p bin/server
mkdir -p bin/client
cd src/tracker
./build.sh
cp bin/tracker.jar ../../bin/tracker/
cd ../
cd server
./build.sh
cp bin/server.jar ../../bin/server/
cd ../
cd client
./build.sh
cp bin/client.jar ../../bin/client/
cd ../../

# Copy Lib files
cp lib/*.jar bin/server/
cp tools/ffmpeg bin/server/
```

Dec 04, 18 15:15

download_models.ps1

Page 1/1

```

$ErrorActionPreference = "Stop"
Import-Module BitsTransfer
if(!(Test-Path -Path "bin\server\models\ssd_inception_v2_coco_2017_11_17\")){
    new-item -Name bin\server\models -Force -ItemType directory
    Start-BitsTransfer -Source "http://download.tensorflow.org/models/object_detec
tion/ssd_inception_v2_coco_2017_11_17.tar.gz" -Destination "ssd_inception_v2_coc
o_2017_11_17.tar.gz"
    .\tools\7za.exe e "ssd_inception_v2_coco_2017_11_17.tar.gz"
    .\tools\7za.exe x "ssd_inception_v2_coco_2017_11_17.tar" -obin\server\models\
-r
    Remove-Item -Path ssd_inception_v2_coco_2017_11_17.tar
    Remove-Item -Path ssd_inception_v2_coco_2017_11_17.tar.gz
}
if(!(Test-Path "bin\server\labels\mscoco_label_map.pbtxt" -PathType Leaf)){
    new-item -Name bin\server\labels -Force -ItemType directory
    Start-BitsTransfer -Source "https://raw.githubusercontent.com/tensorflow/model
s/865c14c1209cb9ae188b2alb5f0883c72e050d4c/research/object_detection/data/mscoco
_label_map.pbtxt" -Destination "bin\server\labels\mscoco_label_map.pbtxt"
}
if(!(Test-Path "bin\server\labels\oid_bbox_trainable_label_map.pbtxt" -PathType
Leaf)){
    new-item -Name bin\server\labels -Force -ItemType directory
    Start-BitsTransfer -Source "https://raw.githubusercontent.com/tensorflow/model
s/865c14c1209cb9ae188b2alb5f0883c72e050d4c/research/object_detection/data/oid_bb
ox_trainable_label_map.pbtxt" -Destination "bin\server\labels\oid_bbox_trainabl
e_label_map.pbtxt"
}

```

Dec 04, 18 16:03

download_models.sh

Page 1/1

```
#!/bin/bash
set -e
if [ ! -d "bin/server/models/ssd_inception_v2_coco_2017_11_17/" ]; then
    mkdir -p bin/server/models/
    curl -L http://download.tensorflow.org/models/object_detection/ssd_inception_v2_coco_2017_11_17.tar.gz | tar -xz -C bin/server/models/
fi
if [ ! -f "bin/server/labels/mscoco_label_map.pbtxt" ]; then
    mkdir -p bin/server/labels/
    curl -L -o bin/server/labels/mscoco_label_map.pbtxt "https://raw.githubusercontent.com/tensorflow/models/865c14c1209cb9ae188b2a1b5f0883c72e050d4c/research/object_detection/data/mscoco_label_map.pbtxt"
    curl -L -o bin/server/labels/oid_bbox_trainable_label_map.pbtxt "https://raw.githubusercontent.com/tensorflow/models/865c14c1209cb9ae188b2a1b5f0883c72e050d4c/research/object_detection/data/oid_bbox_trainable_label_map.pbtxt"
fi
```

| Dec 05, 18 18:07 | README.md | Page 1/3 |
|--|-----------|----------|
| <pre><style> code{ color: #f92672 !important; border: 1px solid hsla(0, 0%, 89%, 1); background-color: hsla(0, 0%, 98%, 1); font-size: .875rem; border-radius: 2px; display: inline-block; padding: 3px 7px; } </style> # 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](#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 Definition Default ----- ----- ----- ----- image_analytics_simulate Turn simulate on or off see [Install Tensor flow models] (#install-Tensor-flow-models) 0 services Services that this server will provide. Split by comma (,) VideoAnalytics, VideoSplit, ImageAnalytics, ImageAnalyticsGraph </pre> | | |

| Dec 05, 18 18:07 | README.md | Page 2/3 |
|--|-----------|----------|
| <pre> 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 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 1008 rmi_registry_host The server rmi registry host/IP localhost ### Client File: server.config.properties Config Definition Default ----- ----- ----- rmi_registry_port Callback RMI Registry Port 1008 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> 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</pre> | | |

Dec 05, 18 18:07

README.md

Page 3/3

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.

Dec 04, 18 19:28

Callback.java

Page 1/1

```

import java.rmi.RemoteException;
import java.rmi.server.ServerNotActiveException;
import java.util.HashSet;
import java.util.Set;
import java.util.function.Consumer;

public class Callback extends java.rmi.server.UnicastRemoteObject implements ICallback {
    CallbackEvent cbe;
    Callback(CallbackEvent cbe) throws RemoteException {
        super();
        this.cbe = cbe;
    }

    public void NewCallback(String sn, String[] dt) throws RemoteException {
        cbe.broadcast(new EventArgs() {
            serviceName = sn;
            data = dt;
            ipAddress = "";
        });
    }

    public void NewCallback(String sn, byte[][] dt) throws RemoteException {
        cbe.broadcast(new EventArgs() {
            serviceName = sn;
            byteData = dt;
            ipAddress = "";
        });
    }

    public void NewExceptionCallback(String sn, Exception ex) {
        cbe.broadcast(new EventArgs() {
            serviceName = sn;
            ipAddress = "";
            exception = ex;
        });
    }
}

class EventArgs {
    String serviceName;
    String ipAddress;
    String[] data;
    byte[][] byteData;
    Exception exception;
}

class CallbackEvent {
    private Set<Consumer<EventArgs>> listeners = new HashSet<>();

    public void addListener(Consumer<EventArgs> listener) {
        listeners.add(listener);
    }

    public void broadcast(EventArgs args) {
        listeners.forEach(x -> x.accept(args));
    }
}

```

Dec 04, 18 20:41

CallbackServer.java

Page 1/1

```
import java.rmi.Naming;
import java.text.MessageFormat;
import java.util.Properties;

public class CallbackServer {
    ICallback cb;
    Properties p;
    CallbackEvent cbe;
    public CallbackServer(Properties p, CallbackEvent cbe){
        this.p = p;
        this.cbe = cbe;
    }
    public void start() {
        try {
            cb = new Callback(this.cbe);
            System.out.println("Server Ready");
            System.setProperty("java.rmi.server.hostname", p.getProperty("rmi_registry_host"));
            Naming.rebind(MessageFormat.format("rmi://{0}:{1}/Callback", p.getProperty("rmi_re
gistry_host"), p.getProperty("rmi_registry_port")), cb);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```


Dec 04, 18 20:43

Client.java

Page 1/1

```

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.util.Properties;

public class Client {
    static Registry r;
    static TrackerHandler trackerHandler;
    static CallbackEvent cbe;
    static Properties p;
    public static void main(String[] args) {
        System.setProperty("java.rmi.server.useLocalHostname", "true");
        p = new Properties();
        File file = new File("client.config.properties");
        if (!file.exists()) {
            try {
                System.out.println("Cant find properties file. Writing default.");
                FileOutputStream out = new FileOutputStream("client.config.properties");
                p.put("trackers", "localhost:1099");
                p.put("rmi_registry_host", "localhost");
                p.put("rmi_registry_port", "1088");
                p.store(out, null);
            } catch (Exception e) {
                System.err.println("Unable to save file client.config.properties");
                e.printStackTrace();
                return;
            }
        } else {
            try {
                FileInputStream in = new FileInputStream("client.config.properties");
                p.load(in);
            } catch (Exception e) {
                System.err.println("Unable to open file client.config.properties");
                e.printStackTrace();
                return;
            }
        }
        try {
            int port = Integer.parseInt(p.get("rmi_registry_port").toString());
            r = LocateRegistry.createRegistry(port);
        } catch (Exception e) {
            e.printStackTrace();
            return;
        }
        cbe = new CallbackEvent();
        // Setup callback
        new CallbackServer(p, cbe).start();
        trackerHandler = new TrackerHandler(p);
        if (!trackerHandler.Init()) {
            return;
        }
        new ClientJobVideoAnalytics(trackerHandler, cbe).run();
    }
}

```

Dec 05, 18 13:01

ClientJobVideoAnalytics.java

Page 1/2

```

import java.awt.Desktop;
import java.io.File;
import java.io.FileOutputStream;
import java.nio.file.Files;
import java.rmi.Naming;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.text.MessageFormat;
import java.util.Arrays;
import java.util.concurrent.TimeUnit;

public class ClientJobVideoAnalytics {
    TrackerHandler trackerHandler;
    CallbackEvent cbe;
    long startTime;
    public ClientJobVideoAnalytics(TrackerHandler trackerHandler, CallbackEvent cb
e) {
        this.trackerHandler = trackerHandler;
        this.cbe = cbe;
    }

    public void run() {
        cbe.addListener((x) -> {
            callBackFromVideoAnalytics(x);
        });
        startTime = System.currentTimeMillis();
        IServiceNode[] sn = trackerHandler.Query("VideoAnalytics");
        if (sn == null) {
            System.err.println("Cant find any service with name {VideoAnalytics} quit.");
            return;
        }
        IServiceInterface service = null;
        for (IServiceNode s : sn) {
            try {
                Registry cr = LocateRegistry.getRegistry(s.getIp(), s.getPort());
                System.out.println(Arrays.toString(cr.list()));
                service = (IServiceInterface) cr.lookup("Service");
                break;
            } catch (Exception e) {
                e.printStackTrace();
                System.out.println(MessageFormat.format("Cant get service with ip {0}. Next.", s.ge
tIp()));
            }
        }
        if (service == null) {
            System.err.println("No service found! Quit.");
            return;
        }
        try {
            File f = new File("Video.mp4");
            byte[] fileContent = Files.readAllBytes(f.toPath());
            service.RunServiceAsync("VideoAnalytics", new byte[][] { fileContent },
                new String[] { "Video.mp4" }, Client.p.getProperty("rmi_registry_port"));
        } catch (Exception e) {
            System.err.println("Unable to submit job.");
            e.printStackTrace();
        }
        System.out.println("Job submitted. Waiting for callback.");
    }

    public void callBackFromVideoAnalytics(EventArgs args) {
        if (args.serviceName.equals("VideoAnalytics")) {

```

Dec 05, 18 13:01

ClientJobVideoAnalytics.java

Page 2/2

```

        if (args.exception != null) {
            System.err.println("Error Callback from VideoAnalytics Service from " + args.ipAddress
);
            args.exception.printStackTrace();
            return;
        }
        System.out.println("Callback from VideoAnalytics service from " + args.ipAddress);
        try {
            File imgFile = new File("graph.png");
            try (FileOutputStream fos = new FileOutputStream(imgFile)) {
                fos.write(args.byteData[0]);
            }
            Desktop.getDesktop().open(imgFile);
            long stopTime = System.currentTimeMillis();
            long elapsedTime = stopTime - startTime;
            System.out.println("It took: " + elapsedTime + "ms");
            long minutes = TimeUnit.MILLISECONDS.toMinutes(elapsedTime);
            System.out.println("AKA: " + minutes + " minutes.");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Dec 04, 18 17:25

ICallback.java

Page 1/1

```
import java.rmi.RemoteException;

public interface ICallback extends java.rmi.Remote {
    public void NewCallback(String serviceName, String[] data) throws RemoteException;
    public void NewCallback(String serviceName, byte[][] data) throws RemoteException;
    public void NewExceptionCallback(String serviceName, Exception exception) throws RemoteException;
}
```

Dec 04, 18 16:35

IServiceInterface.java

Page 1/1

```
import java.rmi.RemoteException;

public interface IServiceInterface extends java.rmi.Remote {
    public String[] RunService(String service, String[] data) throws RemoteException;
    public boolean RunServiceAsync(String service, String[] data, String callbackPort) throws RemoteException;
    public byte[][] RunService(String service, byte[][] data, String[] data2) throws RemoteException;
    public boolean RunServiceAsync(String service, byte[][] data, String[] data2, String callbackPort) throws RemoteException;
}
```

Nov 30, 18 14:13

IServiceNode.java

Page 1/1

```
import java.io.Serializable;

public interface IServiceNode extends Serializable {
    String getIp();

    int getPort();

    String[] getService();
}
```

Dec 04, 18 12:42

ServiceNode.java

Page 1/1

```
public class ServiceNode implements IServiceNode {
    public String Ip;
    public int Port;
    public String[] Services;

    public String getIp(){
        return Ip;
    }
    public int getPort(){
        return Port;
    }
    public String[] getService(){
        return Services;
    }

    @Override
    public int hashCode() {
        int hc = 0;
        for (String s : Services) {
            hc += s.hashCode();
        }
        return Ip.hashCode() + hc;
    }
}
```

Nov 30, 18 13:52

TrackerHandler.java

Page 1/1

```

import java.net.MalformedURLException;
import java.rmi.Naming;
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.text.MessageFormat;
import java.util.ArrayList;
import java.util.Properties;

public class TrackerHandler {
    Properties p;
    ArrayList<TrackerInterface> trackers;

    public TrackerHandler(Properties p) {
        this.p = p;
    }

    public Boolean Init() {
        String[] trackerHosts = p.get("trackers").toString().split(",");
        trackers = new ArrayList<>();
        for (int i = 0; i < trackerHosts.length; i++) {
            String tracker = trackerHosts[i];
            try {
                TrackerInterface ti = (TrackerInterface) Naming
                    .lookup(MessageFormat.format("rmi://{0}/TrackerService", tracker));
                System.out.println(MessageFormat.format("Connected to tracker: {0}", tracker));
                trackers.add(ti);
            } catch (NotBoundException | RemoteException e) {
                e.printStackTrace();
                System.err.println(MessageFormat.format("Unable to connect to tracker with host: {0}.",
                    tracker));
            } catch (MalformedURLException e) {
                e.printStackTrace();
                System.err.println(MessageFormat.format("The tracker host configured is not valid: {0}.",
                    tracker));
            }
        }
        if (trackers.size() == 0) {
            System.err.println("No tracker is connected. Unable to continue.");
            return false;
        }
        return true;
    }

    public IServiceNode[] Query(String service) {
        for (TrackerInterface tracker : trackers) {
            try {
                IServiceNode[] serviceNodes = (IServiceNode[])tracker.GetMeService(service);
                if (serviceNodes.length != 0) {
                    return serviceNodes;
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return null;
    }
}

```

Dec 04, 18 19:30

TrackerInterface.java

Page 1/1

```
import java.rmi.RemoteException;
import java.rmi.server.ServerNotActiveException;

public interface TrackerInterface extends java.rmi.Remote {
    public Boolean RegisterMeAsService(String[] services, int port, String ipAddr)
    throws RemoteException;

    public IServiceNode[] GetMeService(String service) throws RemoteException;

    public IServiceNode[] GetMeServices(String[] services) throws RemoteException;

    public Boolean RemoveMe(String ipAddr) throws RemoteException, ServerNotActive
Exception;
}
```


Nov 30, 18 13:39

build.bat

Page 1/1

```
mkdir bin
javac -d bin *.java
if %errorlevel% neq 0 exit /b %errorlevel%
copy manifest.txt bin
cd bin
jar cvfm client.jar manifest.txt *.class
cd ../
```

Dec 05, 18 18:21

build.sh

Page 1/1

```
mkdir -p bin
javac -d bin *.java
cp manifest.txt bin
cd bin
jar cvfm client.jar manifest.txt *.class
cd ../
```

| | | |
|---|--------------|----------|
| Nov 22, 18 10:21 | manifest.txt | Page 1/1 |
| <div>Manifest-Version: 1.0 Main-Class: Client</div> | | |

Nov 30, 18 14:12

IServiceNode.java

Page 1/1

```
import java.io.Serializable;

public interface IServiceNode extends Serializable {
    String getIp();

    int getPort();

    String[] getService();
}
```

Nov 30, 18 14:12

ServiceNode.java

Page 1/1

```
public class ServiceNode implements IServiceNode {
    public String Ip;
    public int Port;
    public String[] Services;

    public String getIp(){
        return Ip;
    }
    public int getPort(){
        return Port;
    }
    public String[] getService(){
        return Services;
    }

    @Override
    public int hashCode() {
        int hc = 0;
        for (String s : Services) {
            hc += s.hashCode();
        }
        return Ip.hashCode() + hc;
    }
}
```

Dec 04, 18 19:27

Tracker.java

Page 1/1

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.util.Properties;

public class Tracker {
    static Registry r;
    public static void main(String[] args) {
        Properties p = new Properties();
        File file = new File("tracker.config.properties");
        if (!file.exists()) {
            try {
                System.out.println("Cant find properties file. Writing default.");
                FileOutputStream out = new FileOutputStream("tracker.config.properties");
                p.put("rmi_registry_port", "1099");
                p.store(out, null);
            } catch (Exception e) {
                System.err.println("Unable to save file tracker.config.properties");
                e.printStackTrace();
                return;
            }
        } else {
            try {
                FileInputStream in = new FileInputStream("tracker.config.properties");
                p.load(in);
            } catch (Exception e) {
                System.err.println("Unable to open file tracker.config.properties");
                e.printStackTrace();
                return;
            }
        }
        try {
            int port = Integer.parseInt(p.get("rmi_registry_port").toString());
            r = LocateRegistry.createRegistry(port);
        } catch (Exception e) {
            e.printStackTrace();
            return;
        }
        new TrackerServer(p).start();
    }
}
```

Dec 05, 18 11:02

TrackerImpl.java

Page 1/2

```

import java.rmi.RemoteException;
import java.text.MessageFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.LinkedHashSet;
import java.util.Map;
import java.util.Set;
import java.rmi.server.ServerNotActiveException;

class TrackerImpl extends java.rmi.server.UnicastRemoteObject implements TrackerInterface {
    private Map<String, ArrayList<IServiceNode>> serviceDict;
    private ArrayList<IServiceNode> serviceNodes;

    TrackerImpl() throws RemoteException {
        super();
        serviceDict = new HashMap<String, ArrayList<IServiceNode>>();
        serviceNodes = new ArrayList<>();
    }

    private void Log(String log) {
        System.out.println(log);
    }

    public Boolean RegisterMeAsService(String[] services, int port, String ipAddress)
    throws RemoteException {
        Log(MessageFormat.format("Request: {0}", ipAddress));
        for (IServiceNode sn : serviceNodes) {
            if (sn.getIp().equals(ipAddress)) {
                Log(MessageFormat.format("IP: {0} already added", ipAddress));
                return false;
            }
        }
        IServiceNode serviceNode = new ServiceNode();
        serviceNode.Ip = ipAddress;
        serviceNode.Services = services;
        serviceNode.Port = port;
        for (String s : services) {
            ArrayList<IServiceNode> sn = serviceDict.get(s);
            if (sn == null) {
                sn = new ArrayList<IServiceNode>();
                serviceDict.put(s, sn);
            }
            sn.add(serviceNode);
        }
        Log(MessageFormat.format("IP: {0}:{1} added with services {2}", ipAddress, port, Arrays.toString(services)));
        serviceNodes.add(serviceNode);
        return true;
    }

    public IServiceNode[] GetMeService(String service) throws RemoteException{
        ArrayList<IServiceNode> sn = serviceDict.get(service);
        if(sn == null) return new IServiceNode[0];
        return sn.toArray(new IServiceNode[sn.size()]);
    }

    public IServiceNode[] GetMeServices(String[] services) throws RemoteException
    {
        ArrayList<IServiceNode> sns = new ArrayList<>();
        for (String s : services) {

```

Dec 05, 18 11:02

TrackerImpl.java

Page 2/2

```

        ArrayList<IServiceNode> sn = serviceDict.get(s);
        if (sn != null) {
            sns.addAll(sn);
        }
    }
    return MergeServiceNodes(sns);
}

public Boolean RemoveMe(String ipAddress) throws RemoteException, ServerNotActiveException {
    IServiceNode removeNode = null;
    for (IServiceNode sn : serviceNodes) {
        if (sn.getIp().equals(ipAddress)) {
            removeNode = sn;
            break;
        }
    }
    if(removeNode == null){
        return false;
    }
    Log(MessageFormat.format("IP: {0} removed", ipAddress));
    for (String service : removeNode.getService()) {
        serviceDict.get(service).remove(removeNode);
    }
    serviceNodes.remove(removeNode);
    return true;
}

private IServiceNode[] MergeServiceNodes(ArrayList<IServiceNode> services) {
    Set<IServiceNode> set = new LinkedHashSet<>();
    set.addAll(services);
    services.clear();
    services.addAll(set);
    return services.toArray(new IServiceNode[services.size()]);
}
}

```

Dec 04, 18 19:30

TrackerInterface.java

Page 1/1

```
import java.rmi.RemoteException;
import java.rmi.server.ServerNotActiveException;

public interface TrackerInterface extends java.rmi.Remote {
    public Boolean RegisterMeAsService(String[] services, int port, String ipAddr)
    throws RemoteException;

    public IServiceNode[] GetMeService(String service) throws RemoteException;

    public IServiceNode[] GetMeServices(String[] services) throws RemoteException;

    public Boolean RemoveMe(String ipAddr) throws RemoteException, ServerNotActive
Exception;
}
```


Nov 30, 18 14:13

TrackerServer.java

Page 1/1

```
import java.rmi.Naming;
import java.rmi.server.UnicastRemoteObject;
import java.text.MessageFormat;
import java.util.Properties;

public class TrackerServer {
    TrackerImpl t;
    Properties p;
    public TrackerServer(Properties p){
        this.p = p;
    }
    public void start() {
        try {
            t = new TrackerImpl();
            System.out.println("Server Ready");
            Naming.rebind(MessageFormat.format("rmi://localhost:{0}/TrackerService", p.get("rmi_registry_port")), t);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Nov 30, 18 13:38

build.bat

Page 1/1

```
mkdir bin
javac -d bin *.java
if %errorlevel% neq 0 exit /b %errorlevel%
copy manifest.txt bin
cd bin
jar cvfm tracker.jar manifest.txt *.class
cd ../
```

Dec 05, 18 18:21

build.sh

Page 1/1

```
mkdir -p bin
javac -d bin *.java
cp manifest.txt bin
cd bin
jar cvfm tracker.jar manifest.txt *.class
cd ../
```

| | | |
|--|--------------|----------|
| Nov 21, 18 11:45 | manifest.txt | Page 1/1 |
| <div>Manifest-Version: 1.0 Main-Class: Tracker</div> | | |

Dec 04, 18 17:22

CallBackHandler.java

Page 1/1

```
import java.rmi.RemoteException;

public class CallBackHandler{
    private ICallback callback;
    CallBackHandler(ICallback callback){
        this.callback = callback;
    }
    public boolean ExecuteCallback(String serviceName, String[] data){
        try {
            callback.NewCallback(serviceName, data);
        } catch (RemoteException e) {
            e.printStackTrace();
            return false;
        }
        return true;
    }
    public boolean ExecuteCallback(String serviceName, byte[][] data){
        try {
            callback.NewCallback(serviceName, data);
        } catch (RemoteException e) {
            e.printStackTrace();
            return false;
        }
        return true;
    }
    public boolean ExecuteExceptionCallback(String serviceName, Exception e){
        try {
            callback.NewExceptionCallback(serviceName, e);
        } catch (RemoteException ex) {
            ex.printStackTrace();
            return false;
        }
        return true;
    }
}
```

Dec 04, 18 17:10

ICallback.java

Page 1/1

```
import java.rmi.RemoteException;

public interface ICallback extends java.rmi.Remote {
    public void NewCallback(String serviceName, String[] data) throws RemoteException;
    public void NewCallback(String serviceName, byte[][] data) throws RemoteException;
    public void NewExceptionCallback(String serviceName, Exception exception) throws RemoteException;
}
```

Dec 05, 18 18:40

IService.java

Page 1/1

```
public interface IService{  
    String[] run(String[] data) throws Exception;  
    byte[][] run(byte[][] data, String[] data2) throws Exception;  
}
```

Dec 04, 18 16:35

IServiceInterface.java

Page 1/1

```
import java.rmi.RemoteException;

public interface IServiceInterface extends java.rmi.Remote {
    public String[] RunService(String service, String[] data) throws RemoteException;
    public boolean RunServiceAsync(String service, String[] data, String callbackPort) throws RemoteException;
    public byte[][] RunService(String service, byte[][] data, String[] data2) throws RemoteException;
    public boolean RunServiceAsync(String service, byte[][] data, String[] data2, String callbackPort) throws RemoteException;
}
```


Dec 04, 18 12:30

IServiceNode.java

Page 1/1

```
import java.io.Serializable;

public interface IServiceNode extends Serializable {
    String getIp();

    int getPort();

    String[] getService();
}
```

Dec 05, 18 18:32

Server.java

Page 1/1

```

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.util.Properties;

public class Server {
    static Registry r;
    static Properties p;
    public static void main(String[] args) {
        Properties p = new Properties();
        Server.p = p;
        File file = new File("server.config.properties");
        if (!file.exists()) {
            try {
                System.out.println("Cant find properties file. Writing default.");
                FileOutputStream out = new FileOutputStream("server.config.properties");
                p.put("tracker", "localhost:1099");
                p.put("services", "VideoAnalytics,VideoSplit,ImageAnalytics,ImageAnalyticsGraph");
                p.put("rmi_registry_host", "localhost");
                p.put("rmi_registry_port", "1000");
                p.put("image_analytics_model_dir", "models/ssd_inception_v2_coco_2017_11_17/saved_model");
                p.put("image_analytics_label", "labels/mscoco_label_map.pbtxt");
                p.put("image_analytics_simulate", "0");
                p.put("ffmpeg_command", "./ffmpeg");
                p.store(out, null);
            } catch (Exception e) {
                System.err.println("Unable to save file server.config.properties");
                e.printStackTrace();
                return;
            }
        } else {
            try {
                FileInputStream in = new FileInputStream("server.config.properties");
                p.load(in);
            } catch (Exception e) {
                System.err.println("Unable to open file server.config.properties");
                e.printStackTrace();
                return;
            }
        }
        try {
            System.setProperty("java.rmi.server.hostname", p.getProperty("rmi_registry_host"));
            int port = Integer.parseInt(p.getProperty("rmi_registry_port").toString());
            r = LocateRegistry.createRegistry(port);
        } catch (Exception e) {
            e.printStackTrace();
            return;
        }
        new ServerService(p).run();
    }
}

```

Dec 05, 18 11:03

ServerService.java

Page 1/1

```

import java.rmi.Naming;
import java.text.MessageFormat;
import java.util.Map;
import java.util.Properties;

public class ServerService {
    class ShutdownThread extends Thread{
        public TrackerInterface t;
        public Properties properties;
        @Override
        public void run() {
            try {
                System.out.println("Removing myself from tracker.");
                t.RemoveMe(properties.getProperty("rmi_registry_host"));
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
    Properties properties;
    String[] services;
    public static TrackerInterface t;

    ServerService(Properties p) {
        properties = p;
        p.list(System.out);
        services = p.get("services").toString().split(",");
    }

    void run() {
        try {
            ShutdownThread st = new ShutdownThread();
            st.properties = properties;
            Runtime.getRuntime().addShutdownHook(st);
            t = (TrackerInterface) Naming.lookup(MessageFormat.format("rmi://{0}/TrackerService", properties.get("tracker")));
            st.t = t;
            t.RegisterMeAsService(services, Integer.parseInt(properties.getProperty("rmi_registry_port").toString()), properties.getProperty("rmi_registry_host"));
            System.out.println("Connected to tracker");
            IServiceInterface service = new ServiceImpl();
            Naming.rebind(MessageFormat.format("rmi://{0}/Service", properties.getProperty("rmi_registry_host") + ":" + properties.getProperty("rmi_registry_port")), service);
            System.out.println("Service Published");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Dec 05, 18 18:45

ServiceImageAnalytics.java

Page 1/2

```

import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.ArrayList;
import java.util.Arrays;

public class ServiceImageAnalytics implements IService {
    public String[] run(String[] data) throws Exception {
        throw new Exception("This is not implemented. Please use byte[][] variant.");
    }
    public byte[][] run(byte[][] data, String[] data2) throws Exception {
        if(Server.p.getProperty("image_analytics_simulate").equals("0")){
            return ExecuteImageAnalytics(data);
        }
        return ExecuteImageAnalyticsSimulate();
    }
    private byte[][] ExecuteImageAnalytics(byte[][] data) throws Exception{
        // Generate Files
        File csvFile = new File("out.csv");
        ArrayList<String> cmdA = new ArrayList<>();
        cmdA.add("java");
        cmdA.add("-jar");
        cmdA.add("detect-object.jar");
        cmdA.add(Server.p.getProperty("image_analytics_model_dir"));
        cmdA.add(Server.p.getProperty("image_analytics_label"));
        cmdA.add(csvFile.toString());
        try {
            File basePath = new File("image_analytics");
            basePath.mkdir();
            File dir = Files.createTempDirectory(basePath.toPath(), "image_analytics").toF
ile();
            for (int i = 0; i < data.length; i++) {
                File f = new File(dir, i + ".png");
                byte[] b = data[i];
                try (FileOutputStream fos = new FileOutputStream(f)) {
                    fos.write(b);
                }
                cmdA.add(f.toString());
            }
        } catch (Exception e) {
            e.printStackTrace();
            throw e;
        }
        try {
            String[] cmd = new String[cmdA.size()];
            cmdA.toArray(cmd);
            System.out.println(Arrays.toString(cmd));
            ProcessBuilder ps = new ProcessBuilder(cmd);
            ps.redirectErrorStream(true);
            Process pr = ps.start();
            BufferedReader in = new BufferedReader(new InputStreamReader(pr.getInputSt
ream()));
            String line;
            while ((line = in.readLine()) != null) {
                System.out.println(line);
            }
            in.close();
            pr.waitFor();

```

Dec 05, 18 18:45

ServiceImageAnalytics.java

Page 2/2

```

        } catch (Exception e) {
            e.printStackTrace();
            throw e;
        }
        try {
            return new byte[][] { Files.readAllBytes(csvFile.toPath()) };
        } catch (Exception e) {
            e.printStackTrace();
            throw e;
        }
    }
    private byte[][] ExecuteImageAnalyticsSimulate() {
        return new byte[][] { "cat,15\ndog,11\nturtle,22\nbob,100\norange,3".getBytes() };
    }
}

```

Dec 05, 18 18:45

ServiceImageAnalyticsGraph.java

Page 1/1

```

import org.knowm.xchart.BitmapEncoder;
import org.knowm.xchart.PieChart;
import org.knowm.xchart.PieChartBuilder;

import java.io.IOException;

public class ServiceImageAnalyticsGraph implements IService {
    public String[] run(String[] data) throws Exception {
        throw new Exception("This is not implemented. Please use byte[][] variant.");
    }

    public byte[][] run(byte[][] data, String[] data2) throws Exception {
        byte[] csvStringByte = data[0];
        String csvString = new String(csvStringByte);
        String[] splitNewLineCsv = csvString.split("\n");
        PieChart chart = new PieChartBuilder().width(800).height(600).title("Pie Chart")
        ).build();
        chart.getStyler().setCircular(false);
        for (String s : splitNewLineCsv) {
            String[] splitStr = s.split(",");
            chart.addSeries(splitStr[0], Integer.parseInt(splitStr[1]));
        }
        byte[] imageData;
        try {
            imageData = BitmapEncoder.getBitmapBytes(chart, BitmapEncoder.BitmapFormat
            .PNG);
        } catch (IOException e) {
            e.printStackTrace();
            throw e;
        }
        return new byte[][]{imageData};
    }
}

```

Dec 04, 18 17:30

ServiceImpl.java

Page 1/2

```

import java.net.MalformedURLException;
import java.rmi.Naming;
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.text.MessageFormat;

public class ServiceImpl extends java.rmi.server.UnicastRemoteObject implements I
ServiceInterface {
    private CallBackHandler callbackHandler;
    ServiceImpl() throws RemoteException {
        super();
    }

    private IService getServiceClass(String serviceName) throws ClassNotFoundExcep
tion, InstantiationException, IllegalAccessException{
        Class serviceClass = Class.forName("Service" + serviceName);
        return (IService) serviceClass.newInstance();
    }

    private void RegisterCallback(String host, String clientPort) throws RemoteExc
eption, NotBoundException, MalformedURLException {
        if(this.callbackHandler != null) return;
        ICallback callback = (ICallback)Naming.lookup(MessageFormat.format("rmi://{0}:{
1}/Callback", host, clientPort));
        this.callbackHandler = new CallBackHandler(callback);
    }

    public String[] RunService(String service, String[] data) throws RemoteExcepti
on {
        try {
            return getServiceClass(service).run(data);
        } catch (Exception e) {
            e.printStackTrace();
            throw new RemoteException("Unable to run service", e);
        }
    }

    public boolean RunServiceAsync(String service, String[] data, String callbackP
ort) throws RemoteException {
        try {
            String clientHost = getClientHost();
            RegisterCallback(clientHost, callbackPort);
        } catch (Exception e) {
            e.printStackTrace();
            throw new RemoteException("Failed to register callback. Unable to get host", e);
        }
        Thread t = new Thread(() -> {
            try {
                String[] result = RunService(service, data);
                callbackHandler.ExecuteCallback(service, result);
            } catch (Exception e) {
                e.printStackTrace();
                callbackHandler.ExecuteExceptionCallback(service, e);
            }
        });
        t.start();
        return true;
    }

    public byte[][] RunService(String service, byte[][] data, String[] data2) thro
ws RemoteException {
        try {

```

Dec 04, 18 17:30

ServiceImpl.java

Page 2/2

```

        return getServiceClass(service).run(data, data2);
    } catch (Exception e) {
        e.printStackTrace();
        throw new RemoteException("Unable to run service", e);
    }
}

    public boolean RunServiceAsync(String service, byte[][] data, String[] data2,
String callbackPort) throws RemoteException {
    try {
        String clientHost = getClientHost();
        RegisterCallback(clientHost, callbackPort);
    } catch (Exception e) {
        e.printStackTrace();
        throw new RemoteException("Failed to register callback. Unable to get host", e);
    }
    Thread t = new Thread(() -> {
        try {
            byte[][] result = RunService(service, data, data2);
            callbackHandler.ExecuteCallback(service, result);
        } catch (Exception e) {
            e.printStackTrace();
            callbackHandler.ExecuteExceptionCallback(service, e);
        }
    });
    t.start();
    return true;
}
}

```

Dec 04, 18 12:40

ServiceNode.java

Page 1/1

```
public class ServiceNode implements IServiceNode {
    public String Ip;
    public int Port;
    public String[] Services;

    public String getIp(){
        return Ip;
    }
    public int getPort(){
        return Port;
    }
    public String[] getService(){
        return Services;
    }

    @Override
    public int hashCode() {
        int hc = 0;
        for (String s : Services) {
            hc += s.hashCode();
        }
        return Ip.hashCode() + hc;
    }
}
```

Dec 05, 18 18:47

ServiceVideoAnalytics.java

Page 1/3

```

import java.rmi.Naming;
import java.text.MessageFormat;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

class ServiceVideoAnalytics implements IService {
    public String[] run(String[] data) throws Exception {
        throw new Exception("This is not implemented. Please use byte[][] variant.");
    }

    // data = video file
    // data2 = filename
    public byte[][] run(byte[][] data, String[] data2) throws Exception {
        // Setup all service nodes first before executing
        ArrayList<IServiceInterface> videoSplitNode = GetServices("VideoSplit", 1);
        ArrayList<IServiceInterface> imageAnalyticsNode = GetServices("ImageAnalytics",
4);
        ArrayList<IServiceInterface> imageAnalyticsGraphNode = GetServices("ImageAnal
yticsGraph", 1);
        if (videoSplitNode == null || imageAnalyticsNode == null || imageAnalyticsGr
aphNode == null) {
            System.err.println("Unable to find one of the node.");
            throw new Exception("Unable to find one of the node.");
        }
        byte[][] fileBytes = GenerateVideoThumb(data[0], data2[0], videoSplitNode.ge
t(0));
        byte[] nameToOccurance = ImageAnalytics(fileBytes, imageAnalyticsNode);
        byte[] graphImg = GraphIt(nameToOccurance, imageAnalyticsGraphNode.get(0));
        return new byte[][] {graphImg};
    }

    private byte[][] GenerateVideoThumb(byte[] data, String filename, IServiceInte
rface service) throws Exception {
        byte[][] fileBytes;
        try {
            fileBytes = service.RunService("VideoSplit", new byte[][] { data }, new Stri
ng[] { filename });
        } catch (Exception e) {
            e.printStackTrace();
            throw e;
        }
        return fileBytes;
    }

    private byte[] ImageAnalytics(byte[][] files, ArrayList<IServiceInterface> ser
vices) throws Exception {
        ArrayList<byte[][]> splitedFiles = SplitFiles(files, services.size());
        Thread[] threads = new Thread[services.size()];
        byte[][] result = new byte[services.size()][];
        for (int i = 0; i < services.size(); i++) {
            final int index = i;
            final ArrayList<byte[]> filesToRun = splitedFiles[i];
            final IServiceInterface serviceToRunAt = services.get(i);
            Thread t = new Thread() -> {
                byte[][] bts = new byte[filesToRun.size()][];
                for (int y = 0; y < filesToRun.size(); y++) {
                    try {
                        bts[y] = filesToRun.get(y);
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            };
            threads[index] = t;
            threads[index].start();
        }
        for (int i = 0; i < threads.length; i++) {
            try {
                threads[i].join();
            } catch (Exception e) {
                e.printStackTrace();
                throw e;
            }
        }
        StringBuilder sb = new StringBuilder();
        Map<String, Integer> nameToOccurance = new HashMap<>();
        for (int i = 0; i < result.length; i++) {
            byte[] csvB = result[i];
            if (csvB == null) {
                continue;
            }
            String csvContent = new String(csvB);
            csvContent = csvContent.replace("\r", "");
            String[] newLineSplit = csvContent.split("\n");
            for (String s : newLineSplit) {
                if (!s.equals("")) {
                    String name = s.split(",")[0];
                    if (nameToOccurance.get(name) == null) {
                        nameToOccurance.put(name, 0);
                    }
                    nameToOccurance.put(name, nameToOccurance.get(name) + 1);
                }
            }
        }
        if (nameToOccurance.keySet().size() == 0) {
            throw new Exception("No result from Image Analytics.");
        }
        for (String key : nameToOccurance.keySet()) {
            sb.append(key).append(",").append(nameToOccurance.get(key)).append("\n");
        }
        System.out.println("CSV Return: " + sb.toString());
        return sb.toString().getBytes();
    }

    private byte[] GraphIt(byte[] nameToOccuranceCsv, IServiceInterface service) t
hrows Exception {
        byte[][] fileBytes;
        try {
            fileBytes = service.RunService("ImageAnalyticsGraph", new byte[][] { nameToOcc
uranceCsv }, null);
        } catch (Exception e) {
            e.printStackTrace();
            throw e;
        }
        return fileBytes[0];
    }

```

Dec 05, 18 18:47

ServiceVideoAnalytics.java

Page 2/3

```

    }
    try {
        System.out.println("Running service with T: " + Thread.currentThread().getId(
));
        byte[][] r = serviceToRunAt.RunService("ImageAnalytics", bts, null);
        result[index] = r[0];
    } catch (Exception e) {
        e.printStackTrace();
    }
    });
    t.start();
    threads[i] = t;
}
for (int i = 0; i < threads.length; i++) {
    try {
        threads[i].join();
    } catch (Exception e) {
        e.printStackTrace();
        throw e;
    }
}
StringBuilder sb = new StringBuilder();
Map<String, Integer> nameToOccurance = new HashMap<>();
for (int i = 0; i < result.length; i++) {
    byte[] csvB = result[i];
    if (csvB == null) {
        continue;
    }
    String csvContent = new String(csvB);
    csvContent = csvContent.replace("\r", "");
    String[] newLineSplit = csvContent.split("\n");
    for (String s : newLineSplit) {
        if (!s.equals("")) {
            String name = s.split(",")[0];
            if (nameToOccurance.get(name) == null) {
                nameToOccurance.put(name, 0);
            }
            nameToOccurance.put(name, nameToOccurance.get(name) + 1);
        }
    }
}
if (nameToOccurance.keySet().size() == 0) {
    throw new Exception("No result from Image Analytics.");
}
for (String key : nameToOccurance.keySet()) {
    sb.append(key).append(",").append(nameToOccurance.get(key)).append("\n");
}
System.out.println("CSV Return: " + sb.toString());
return sb.toString().getBytes();
}

private byte[] GraphIt(byte[] nameToOccuranceCsv, IServiceInterface service) t
hrows Exception {
    byte[][] fileBytes;
    try {
        fileBytes = service.RunService("ImageAnalyticsGraph", new byte[][] { nameToOcc
uranceCsv }, null);
    } catch (Exception e) {
        e.printStackTrace();
        throw e;
    }
    return fileBytes[0];
}

```


Dec 05, 18 18:47

ServiceVideoAnalytics.java

Page 3/3

```

    }

    private ArrayList<byte[][]> SplitFiles(byte[][] files, int parts) {
        ArrayList<byte[][]> splitedFiles = new ArrayList<byte[][]>(parts);
        for (int i = 0; i < parts; i++) {
            splitedFiles[i] = new ArrayList<byte[][]>();
        }
        for (int i = 0; i < files.length; i++) {
            splitedFiles[i % parts].add(files[i]);
        }
        return splitedFiles;
    }

    private ArrayList<IServiceInterface> GetServices(String serviceName, int max)
    {
        ArrayList<IServiceInterface> services = new ArrayList<>();
        try {
            IServiceNode[] sn = (IServiceNode[]) ServerService.t.GetMeService(serviceName);
            for (IServiceNode s : sn) {
                try {
                    services.add((IServiceInterface) Naming
                        .lookup(MessageFormat.format("rmi://{0}/Service", s.getIp() + ":" + s.getPort())));
                    if (services.size() == max) {
                        break;
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                    System.out.println(MessageFormat.format("Cant get service with ip {0}. Next.", s.getIp()));
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
        if (services.size() == 0) {
            System.err.println("Unable to find any node named " + serviceName);
            return null;
        }
        return services;
    }
}

```

Dec 05, 18 18:48

ServiceVideoSplit.java

Page 1/1

```

import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.nio.file.Files;
import java.util.Arrays;

class ServiceVideoSplit implements IService {
    public String[] run(String[] data) throws Exception {
        throw new Exception("This is not implemented. Please use byte[][] variant.");
    }

    // data = video file
    // data2 = filename
    public byte[][] run(byte[][] data, String[] data2) throws Exception {
        // Save the file
        try {
            new File("temp/").mkdirs();
            new File("temp/" + data2[0] + "out/").mkdirs();
            try (FileOutputStream fos = new FileOutputStream("temp/" + data2[0])) {
                fos.write(data[0]);
            }
        } catch (Exception e) {
            e.printStackTrace();
            throw e;
        }
        // split file
        try {
            final String[] cmd = { Server.p.getProperty("ffmpeg_command"), "-i", "temp/"
+ data2[0], "-vf", "scale=720:-1,fps=4", "temp/" + data2[0] + "out/out%d.png" };
            System.out.println(Arrays.toString(cmd));
            ProcessBuilder ps = new ProcessBuilder(cmd);
            ps.redirectErrorStream(true);
            Process pr = ps.start();
            BufferedReader in = new BufferedReader(new InputStreamReader(pr.getInputStream()));
            String line;
            while ((line = in.readLine()) != null) {
                System.out.println(line);
            }
            in.close();
            pr.waitFor();
            if (pr.exitValue() != 0)
                throw new Exception("ffmpeg return non 0 exit code.");
        } catch (Exception e) {
            e.printStackTrace();
            throw e;
        }
        File dir = new File("temp/" + data2[0] + "out/");
        File[] files = dir.listFiles();
        byte[][] bytes = new byte[files.length][];
        for (int i = 0; i < files.length; i++) {
            try {
                bytes[i] = Files.readAllBytes(files[i].toPath());
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return bytes;
    }
}

```

Dec 04, 18 19:30

TrackerInterface.java

Page 1/1

```
import java.rmi.RemoteException;
import java.rmi.server.ServerNotActiveException;

public interface TrackerInterface extends java.rmi.Remote {
    public Boolean RegisterMeAsService(String[] services, int port, String ipAddr)
    throws RemoteException;

    public IServiceNode[] GetMeService(String service) throws RemoteException;

    public IServiceNode[] GetMeServices(String[] services) throws RemoteException;

    public Boolean RemoveMe(String ipAddr) throws RemoteException, ServerNotActive
Exception;
}
```

Dec 04, 18 15:16

build.bat

Page 1/1

```
mkdir bin
javac -classpath ../../lib/detect-object.jar;../../lib/xchart-3.5.2.jar -d bin *.java
if %errorlevel% neq 0 exit /b %errorlevel%
copy manifest.txt bin
cd bin
jar cvfm server.jar manifest.txt *.class
cd ../
```

Dec 05, 18 18:21

build.sh

Page 1/1

```
mkdir -p bin
javac -classpath '../lib/detect-object.jar:../lib/xchart-3.5.2.jar' -d bin *.java
cp manifest.txt bin
cd bin
jar cvfm server.jar manifest.txt *.class
cd ../
```

| | | |
|---|--------------|----------|
| Dec 03, 18 18:21 | manifest.txt | Page 1/1 |
| <div>Manifest-Version: 1.0 Main-Class: Server</div> | | |

| Dec 05, 18 18:53 | Table of Content | Page 1/1 |
|---|-----------------------------------|-----------|
| Table of Contents | | |
| 1 <i>build.bat</i> | sheets 1 to 1 (1) pages 1- 1 | 24 lines |
| 2 <i>build.sh</i> | sheets 2 to 2 (1) pages 2- 2 | 22 lines |
| 3 <i>download_models.ps1</i> .. | sheets 3 to 3 (1) pages 3- 3 | 19 lines |
| 4 <i>download_models.sh</i> .. | sheets 4 to 4 (1) pages 4- 4 | 12 lines |
| 5 <i>README.md</i> | sheets 5 to 6 (2) pages 5- 7 | 94 lines |
| 6 <i>Callback.java</i> | sheets 7 to 7 (1) pages 8- 8 | 56 lines |
| 7 <i>CallbackServer.java</i> .. | sheets 8 to 8 (1) pages 9- 9 | 24 lines |
| 8 <i>Client.java</i> | sheets 9 to 9 (1) pages 10- 10 | 57 lines |
| 9 <i>ClientJobVideoAnalytics.java</i> | sheets 10 to 10 (1) pages 11- 12 | 84 lines |
| 10 <i>ICallback.java</i> | sheets 11 to 11 (1) pages 13- 13 | 8 lines |
| 11 <i>IServiceInterface.java</i> | sheets 12 to 12 (1) pages 14- 14 | 9 lines |
| 12 <i>IServiceNode.java</i> ... | sheets 13 to 13 (1) pages 15- 15 | 10 lines |
| 13 <i>ServiceNode.java</i> | sheets 14 to 14 (1) pages 16- 16 | 25 lines |
| 14 <i>TrackerHandler.java</i> .. | sheets 15 to 15 (1) pages 17- 17 | 56 lines |
| 15 <i>TrackerInterface.java</i> | sheets 16 to 16 (1) pages 18- 18 | 13 lines |
| 16 <i>bin</i> | sheets 16 to 16 (1) pages 18- 18 | 1 lines |
| 17 <i>build.bat</i> | sheets 17 to 17 (1) pages 19- 19 | 8 lines |
| 18 <i>build.sh</i> | sheets 18 to 18 (1) pages 20- 20 | 7 lines |
| 19 <i>manifest.txt</i> | sheets 19 to 19 (1) pages 21- 21 | 3 lines |
| 20 <i>IServiceNode.java</i> ... | sheets 20 to 20 (1) pages 22- 22 | 10 lines |
| 21 <i>ServiceNode.java</i> | sheets 21 to 21 (1) pages 23- 23 | 25 lines |
| 22 <i>Tracker.java</i> | sheets 22 to 22 (1) pages 24- 24 | 44 lines |
| 23 <i>TrackerImpl.java</i> | sheets 23 to 23 (1) pages 25- 26 | 94 lines |
| 24 <i>TrackerInterface.java</i> | sheets 24 to 24 (1) pages 27- 27 | 13 lines |
| 25 <i>TrackerServer.java</i> .. | sheets 25 to 25 (1) pages 28- 28 | 22 lines |
| 26 <i>bin</i> | sheets 25 to 25 (1) pages 28- 28 | 1 lines |
| 27 <i>bin</i> | | |
| | sheets 25 to 25 (1) pages 28- 28 | 1 lines |
| 28 <i>build.bat</i> | sheets 26 to 26 (1) pages 29- 29 | 8 lines |
| 29 <i>build.sh</i> | sheets 27 to 27 (1) pages 30- 30 | 7 lines |
| 30 <i>manifest.txt</i> | sheets 28 to 28 (1) pages 31- 31 | 3 lines |
| 31 <i>CallBackHandler.java</i> | sheets 29 to 29 (1) pages 32- 32 | 36 lines |
| 32 <i>ICallback.java</i> | sheets 30 to 30 (1) pages 33- 33 | 8 lines |
| 33 <i>IService.java</i> | sheets 31 to 31 (1) pages 34- 34 | 5 lines |
| 34 <i>IServiceInterface.java</i> | sheets 32 to 32 (1) pages 35- 35 | 9 lines |
| 35 <i>IServiceNode.java</i> ... | sheets 33 to 33 (1) pages 36- 36 | 10 lines |
| 36 <i>Server.java</i> | sheets 34 to 34 (1) pages 37- 37 | 55 lines |
| 37 <i>ServerService.java</i> .. | sheets 35 to 35 (1) pages 38- 38 | 47 lines |
| 38 <i>ServiceImageAnalytics.java</i> | sheets 36 to 36 (1) pages 39- 40 | 76 lines |
| 39 <i>ServiceImageAnalyticsGraph.java</i> | sheets 37 to 37 (1) pages 41- 41 | 33 lines |
| 40 <i>ServiceImpl.java</i> | sheets 38 to 38 (1) pages 42- 43 | 84 lines |
| 41 <i>ServiceNode.java</i> | sheets 39 to 39 (1) pages 44- 44 | 25 lines |
| 42 <i>ServiceVideoAnalytics.java</i> | sheets 40 to 41 (2) pages 45- 47 | 155 lines |
| 43 <i>ServiceVideoSplit.java</i> | sheets 42 to 42 (1) pages 48- 48 | 61 lines |
| 44 <i>TrackerInterface.java</i> | sheets 43 to 43 (1) pages 49- 49 | 13 lines |
| 45 <i>bin</i> | sheets 43 to 43 (1) pages 49- 49 | 1 lines |
| 46 <i>build.bat</i> | sheets 44 to 44 (1) pages 50- 50 | 8 lines |
| 47 <i>build.sh</i> | sheets 45 to 45 (1) pages 51- 51 | 7 lines |
| 48 <i>manifest.txt</i> | sheets 46 to 46 (1) pages 52- 52 | 3 lines |