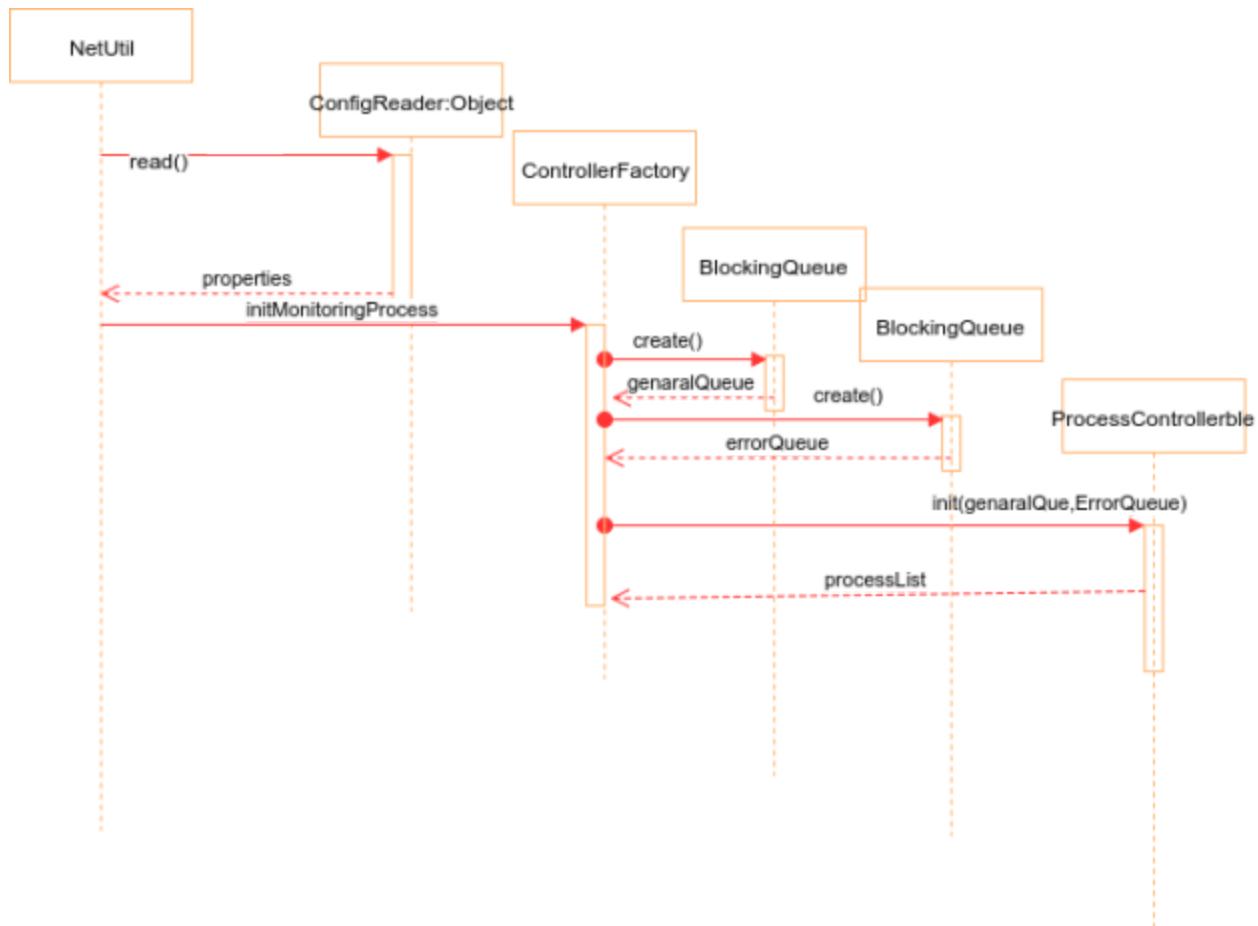


Abstract

This a java service which can be used to continuously monitor a given host/hosts using ping ,TCP/IP protocol,Traceroute. Since the java does not supports the low level ICMP protocol i have used the java Process utility to run the native ping/traceroute command. These native commands are provided by the underlying os. This implementation only tested on Ubuntu 16.04 but it can be improved to used under windows as well.

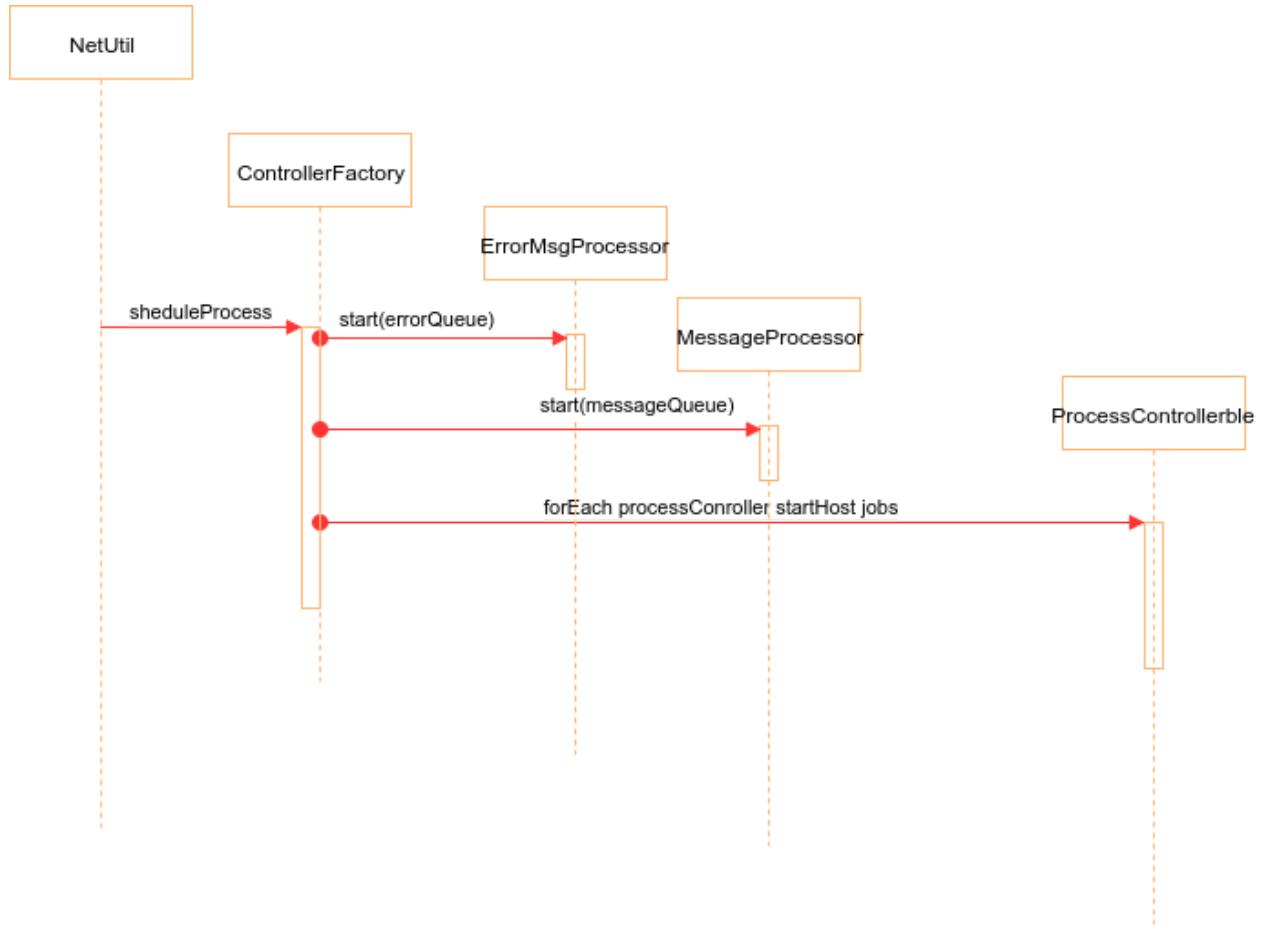
Initialization process

Main sequence follow as follows:



- **NetUtil** is the main class which start the process .
- **ConfigReader** - read the configuration from the config file.The default config file is located at config/config.properties but it can customized.
- **ConrollerFactory** - The main factory which constructs the concrete implementations, glue the each component.
- **ErrorQueue** - Instance of **BlockingQueue** which hold the error messages.
- **GenaralQueue** - Instance of **BlockingQueue** which hold the error messages.
- **ProcessControllerble** - interface which can be extends to implement any network monitoring service.For the moment there are three services :
 - **ICMPController** - service for controlling ping service
 - **TCPIPController**- implementation using tcplp protocol. This is implemented based on socket programming.
 - **TraceRTController** - implementation for traceroute

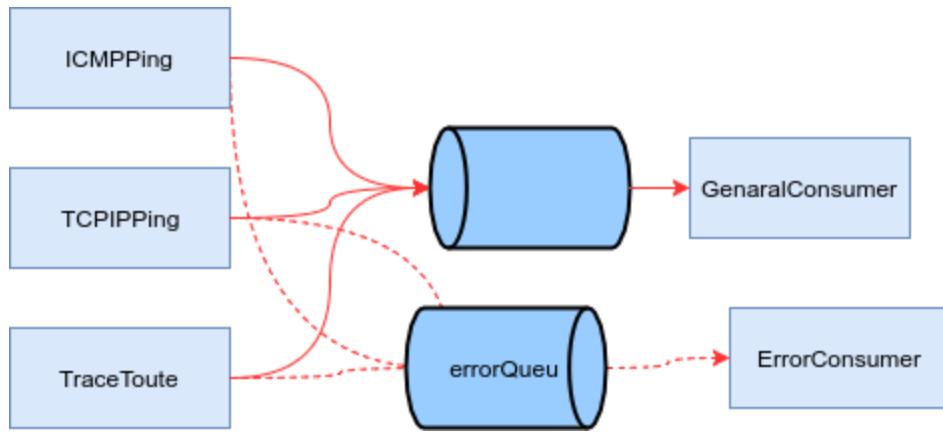
Schedule process



ErrorMsgProcess- listener for error queue.

MessageProcess - listener for general Queue.

General Queue consuming process



ICMPPing, TCPIP, TraceToute are jobs they produce messages to queue. On Error a error message publish to error Queue.

GenaralConsumer - responsible for recording the messages.

ErrorConsumer - responsible for processing error messages and posting error reprot to given url..

How to

How to build

Change the director to project folder and run `mvn clean install`.

How to Run

The service use the java logging service to log. The default file is located at config/logging.properties. It can be configured according to standard java logging configuration.default set to `WARNING`.

```
java -Djava.util.logging.config.file=config/logging.properties -jar  
target/net-util-0.0.1-SNAPSHOT.jar
```

File structure

```
└── net-util  
    ├── config  
    │   ├── config.properties  
    │   └── logging.properties  
    ├── pom.xml  
    └── src  
  
└── report-recorder  
    ├── deploy  
    │   └── config.yml  
    ├── pom.xml  
    ├── README.md  
    └── src
```

Supporting services

Report-recorder

this a sample rest service which consume the json post request and log the output to console.This is a microservice build using dropwizard,maven.

How to build

Change the directory to above folder run “mvn clean install”

How to run

It can be start the service using following command:

```
report-recorder$ java -jar target/report-recorder-1.0-SNAPSHOT.jar server deploy/config.yml
```