

# Out of Memory

- **Problem**
  - GUI mode
  - CLI mode
- **Solution**
  - Change the memory parameters
  - Modify `conf/analyzer.properties` file
- **Related articles**

## Problem

Under this general category are memory-related problems (out of memory).

Every Kiuwan analysis step is executed as a forked process running in a JVM. That forked execution is run with some default memory parameters (min and max).

By default, forked JVM is configured to run with a maximum of 1024 Mb. If the analysis needs more memory, the execution will finish with this error.

The most common out-of-memory error messages are:

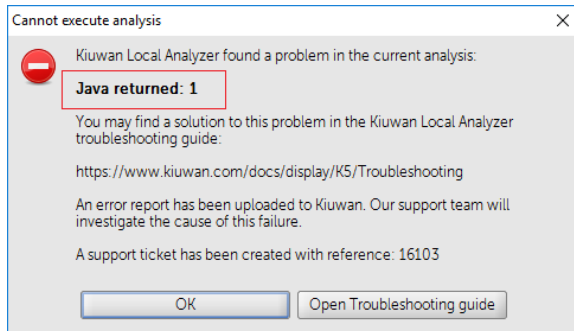
```
Exception in thread "main" com.als.core.OptimythException: GC overhead
limit exceeded
Caused by: java.lang.OutOfMemoryError: GC overhead limit exceeded

Exception in thread "main" com.als.core.OptimythException: Java heap space
Caused by: java.lang.OutOfMemoryError: Java heap space
```

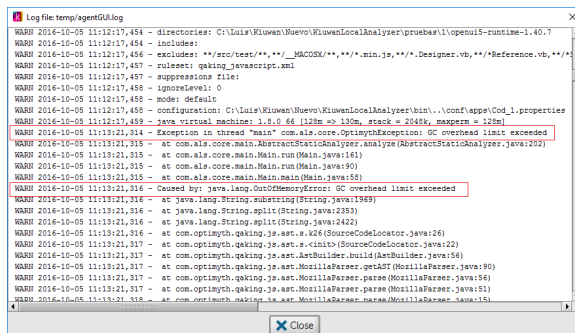
All of them can be solved increasing the memory allocated to Kiuwan analysis.

## GUI mode

If you are using Kiuwan Local Analyzer in GUI mode, a dialog will open indicating this error:



Click **Analyzer log** to open a window where the log file (`temp/agentGUI.log`) is displayed. You will see the following error message.



You will find some lines as the following indicating that an Out of Memory error is produced.

```
WARN 2016-10-05 11:13:21,314 - Exception in thread "main" com.als.core.OptimythException: GC overhead limit exceeded
WARN 2016-10-05 11:13:21,316 - Caused by: java.lang.OutOfMemoryError: GC overhead limit exceeded
```

## CLI mode

If you are using Kiuwan Local Analyzer in CLI mode, this error is reported to the standard output.

```
java virtual machine 1.8.0_60 (16m * 16m, stack = 1048k, maxperm = 10m)
Exception in thread "main" com.als.core.OptimythException: GC overhead limit exceeded
    at com.als.core.main.AbstractStaticAnalyzer.analyze(AbstractStaticAnalyzer.java:282)
    at com.als.core.main.Main.run(Main.java:161)
    at com.als.core.main.Main.run(Main.java:98)
    at com.als.core.main.Main.main(Main.java:52)
Caused by: java.lang.OutOfMemoryError: GC overhead limit exceeded
    at java.util.concurrent.CopyOnWriteArrayList.add(String.java:368)
    at java.lang.String.<init>(String.java:107)
    at java.lang.String.substring(String.java:198)
    at java.lang.String.split(String.java:135)
    at java.lang.String.split(String.java:142)
    at com.optinyth.qaking.js.ast.s.<init>(SourceCodeLocator.java:26)
    at com.optinyth.qaking.js.ast.s.<init>(SourceCodeLocator.java:22)
    at com.optinyth.qaking.js.ast.Builder.build(AstBuilder.java:16)
    at com.optinyth.qaking.js.ast.NoillParser.getLast(NoillParser.java:98)
    at com.optinyth.qaking.js.ast.NoillParser.parse(NoillParser.java:98)
    at com.optinyth.qaking.js.ast.NoillParser.parse(NoillParser.java:15)
    at com.optinyth.qaking.js.ast.NoillParser.parse(NoillParser.java:15)
    at com.als.core.language.languagesupport.parser.validate(Languagesupport.java:173)
    at com.als.core.language.languagesupport.parser(Languagesupport.java:167)
    at com.als.core.main.AbstractStaticAnalyzer.parseFile(AbstractStaticAnalyzer.java:171)
    at com.als.core.main.AbstractStaticAnalyzer.analyzeSingleFile(AbstractStaticAnalyzer.java:118)
    at com.als.core.main.SingleTechnologyStaticAnalyzer.<init>(SingleTechnologyStaticAnalyzer.java:112)
    at com.als.core.main.SingleTechnologyStaticAnalyzer.analyze(SingleTechnologyStaticAnalyzer.java:98)
    at com.als.core.main.SingleTechnologyStaticAnalyzer.analyze(SingleTechnologyStaticAnalyzer.java:75)
    at com.als.core.main.AbstractStaticAnalyzer.analyze(AbstractStaticAnalyzer.java:160)
    ... 7 more
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=10m; support was removed in 8.0
Kiuwan Local Analyzer found a problem in the current analysis:
The following error occurred while executing this line:
C:\Users\Kiuwan\Huevo\KiuwanLocalAnalyzer\bin\agent.xml:114: The following error occurred while executing this line:
C:\Users\Kiuwan\Huevo\KiuwanLocalAnalyzer\bin\agent.xml:118: The following error occurred while executing this line:
C:\Users\Kiuwan\Huevo\KiuwanLocalAnalyzer\bin\agent.xml:124: The following error occurred while executing this line:
C:\Users\Kiuwan\Huevo\KiuwanLocalAnalyzer\bin\rules.xml:121: The following error occurred while executing this line:
C:\Users\Kiuwan\Huevo\KiuwanLocalAnalyzer\bin\analyzerCommon.xml:79: The following error occurred while executing this line:
C:\Users\Kiuwan\Huevo\KiuwanLocalAnalyzer\bin\analyzerCommon.xml:79: Java returned: 1
An error report has been uploaded to Kiuwan. Our support team will investigate the cause of this failure.
A support ticket has been created with reference: 16107
Exit code = 1
```

An error code will be returned to the calling process.

Please visit [Local Analyzer Return Codes](#) for further information on return codes.

## Solution

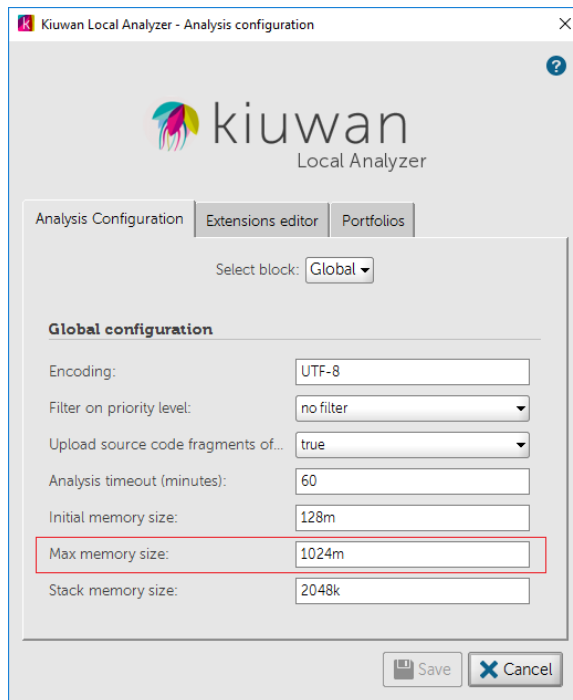
The solution consists of providing more memory to Kiuwan.

You can do it in two ways:

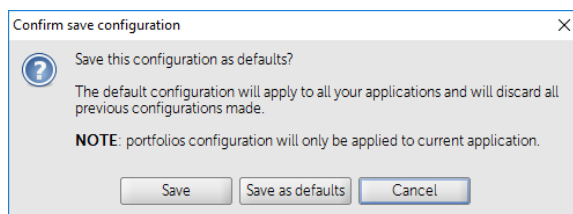
1. Change the memory parameters
2. Modify the conf/analyzer.properties file

### Change the memory parameters

1. Click **Advanced** in the analysis configuration window
2. Select **Analysis Configuration**
3. Modify the **Max memory size** property (set by default to 1024 Mb)



4. After modifying the value, click **Save** to open the next dialog



5. **Save** = this configuration will apply only to the current applications.  
**Save as defaults** = the configuration changes will apply to all the applications of your Kiuwan Local Analyzer installation.

## Modify conf/analyzer.properties file

An alternative way to change this memory value is by **modifying the conf/analyzer.properties** file.

To do it, edit "**memory-max**" property:

```
# Maximum size for heap memory (1024m = 1 Gigabyte)
memory.max=1024m
```

If you have a specific configuration for an application and you want the change only to be applied to that specific application, you will need to modify this property in the **conf/apps/<name\_of\_your\_app>.properties** file.



In case you increase this parameter too high, it's possible that the Operating System cannot allocate such free memory.

If this happens, you will get a Not enough Memory error. In this case, please read the [next section](#).

## Related articles

- [SSO - Form-based authentication fails](#)
- [SSO - HTTP authentication fails](#)
- [SSO - WIA is not working](#)
- [SSO - Cannot authenticate with credentials](#)
- [Basic Authentication Error when Exporting Action Plan to Atlassian JIRA](#)