

Optimize Neo4j memory allocation

- [Introduction](#)
 - [How do performance issues manifest themselves?](#)
- [Update process](#)
- [Memrec optimization test](#)
 - [Microsoft Windows via traditional installer](#)
 - [Docker](#)
- [Database size test](#)

Introduction

By default Neo4j is set to use the following RAM:

CAST Imaging 2.5.0	8GB of RAM memory (initial and maximum)
CAST Imaging 2.4.x	3GB of RAM memory (initial and maximum)

If you are facing performance issues, it is possible to modify the allocated memory. See also https://neo4j.com/docs/operations-manual/current/performance/memory-configuration/#_considerations for more information.

How do performance issues manifest themselves?

In general, a sure sign that you need to modify the RAM allocated to Neo4j is when objects fail to load in CAST Imaging - particularly if there are a large number of objects to display - this is because the query used to fetch the objects for display is not completing before a timeout occurs. This timeout will manifest itself as a 504 (gateway timeout) error in the browser. Allocating additional RAM to Neo4j can help reduce the time required to complete the query.

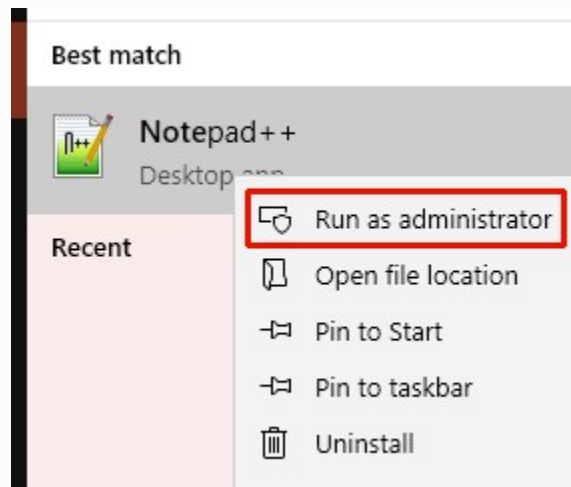
Update process

Open and edit the following file:

```
Microsoft Windows traditional installer:  
%APPDATA%\CAST\ImagingSystem\neo4j\neo4j.conf  
  
Docker Installer extension (located in the folder created when unzipping the extension):  
neo4j\configuration\neo4j.conf
```

Microsoft Windows via traditional installer

This file is located in the protected %APPDATA% location, therefore you must open the file with **elevated permission** (this is usually achieved by right clicking your text editor in the Windows start menu and selecting **Run as administrator**):



Linux

You may need to use elevated permissions to edit this file (for example use **sudo**).

Alter the following lines and increase the RAM allocation as necessary:

```
dbms.memory.heap.initial_size=8G
dbms.memory.heap.max_size=8G
```

Restart the **Neo4j Windows service** / **Docker container** to ensure the changes are taken into account.

Memrec optimization test

You can also run an optimization test (**memrec**) that will recommend memory allocation and database settings for you. To do so, run the following commands:

Microsoft Windows via traditional installer

Input the amount of available RAM on the server (for a server with 16GB of RAM, enter "16G" as in the example below):

```
cd %PROGRAMFILES%\CAST\ImagingSystem\neo4j\bin
neo4j-admin memrec --memory=16G
```

The tool will recommend settings as follows:

```
# Based on the above, the following memory settings are recommended:
dbms.memory.heap.initial_size=5g
dbms.memory.heap.max_size=5g
dbms.memory.pagecache.size=7g
```

Docker

Login to the neo4j docker container to run the **memrec** command. Make sure your neo4j container is up and running.

```
$ docker exec -it neo4j /bin/bash
root@2d9e51d4be43:/var/lib/neo4j# cd bin
root@2d9e51d4be43:/var/lib/neo4j/bin# ./neo4j-admin memrec --memory=16G
```

The tool will output the recommended settings. Exit the container:

```
root@2d9e51d4be43:/var/lib/neo4j/bin# exit
```

Database size test

The database size test should only be performed once you have imported Application data into CAST Imaging. It displays the size in MB of the Neo4j database, which can then be used to modify the following settings:

- `dbms.memory.pagecache.size`

Run the following command:

```
cd %PROGRAMFILES%\CAST\ImagingSystem\neo4j\bin
neo4j-admin memrec --database=graph.db
```

The tool will recommend settings as follows:

```
# The numbers below have been derived based on your current data volume in database and index configuration of
database 'graph.db'.
# They can be used as an input into more detailed memory analysis.
# Lucene indexes: 0.0
# Data volume and native indexes: 114647.0
```

As opposed to the memory recommendations from the Neo4j tool, you can also set the **`dbms.memory.pagecache.size`** value comparable to your data value.