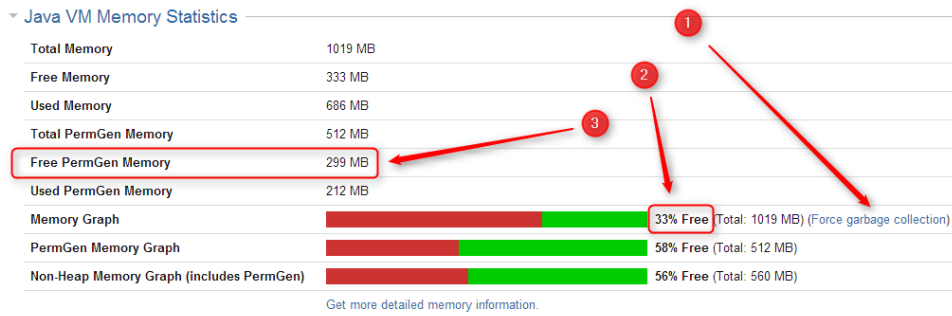


Memory Guidelines

On a production system, it is a good idea to check if you have enough free memory in JIRA's Java process before installing Structure (any other plugin too).

Assessing Available Memory

1. Open menu **Administration | Troubleshooting and Support | System Info** and scroll down to **Java VM Memory Statistics**.
2. Click **Force Garbage Collection**
3. Note the free % number of the **Memory Graph** (heap memory).
4. Note the absolute amount of **Free PermGen Memory** (non-heap memory for Java classes).



Memory Statistic	Recommended Value	Parameter in <code>setenv.sh</code> / <code>setenv.bat</code>
% of Free Heap Memory	25% – 50%	<code>JVM_MAXIMUM_MEMORY</code>
Free PermGen Memory	100 – 200 MB	<code>JIRA_MAX_PERM_SIZE</code>



All recommendations are for a general case and do not guarantee that you don't get `OutOfMemoryError`. Individual cases may vary.

Heap Memory Requirements

It is recommended that % of free heap memory is from 25% to 50%.

Structure requires about additional 100 MB of heap memory. You can take your current statistic of **Used Memory** and **Total Memory**, add 100 MB to the **Used Memory** and calculate the recommended value for the **Total Memory**.



If you already have recommended % of free memory, you can just increase total heap memory by 200 MB.

PermGen Memory Requirements

This section applies to JIRA running on Sun/Oracle Hotspot Java VM only.

PermGen space is used for Java classes and may be depleted if you uninstall, install or upgrade plugins frequently, or if you don't restart JIRA over a long period of time. Due to technical reasons PermGen space might not get cleaned up from the obsolete classes and you may end up with `OutOfMemoryError: PermGen space error`.

Structure classes use only about 10 MB of PermGen space. But for the reasons just mentioned, it is good to have a safety margin with a free PermGen space of at least 100 MB.



The out-of-the-box value for maximum PermGen space is 256 MB, and JIRA without any plugins takes up about 180 MB soon after it's started. The default value is not safe enough!

Changing Memory Parameters

To change memory parameters, edit `setenv.sh` (on Windows, `setenv.bat`).

- To change maximum amount of Heap space, edit `JVM_MAXIMUM_MEMORY` parameter near the top of the script.

```
JVM_MAXIMUM_MEMORY="2000m"
```

- To change maximum amount of PermGen space, edit `JIRA_MAX_PERM_SIZE=256m` line. Alternatively, you can add `MaxPermSize` parameter to `JVM_SUPPORT_RECOMMENDED_ARGS`, for example:

```
JVM_SUPPORT_RECOMMENDED_ARGS="-XX:MaxPermSize=400m"
```

You need to restart JIRA for these settings to take effect.

Use 64-Bit Java

It is imperative to use 64-bit Java when allocating large amount of memory to it (1 GB and more). To check if you're running 64-bit Java, look up **Java VM** parameter on the System Info page.

Physical Memory Requirements



Avoid swapping at all costs!

The amount of physical memory should be enough to accommodate the whole heap and non-heap memory. If you have other Java or memory-intensive applications running on the same host, they all should fit in physical memory, plus you need to reserve at least 1 GB for operating system, services and file cache.

Do not allocate more memory to JIRA if it cannot fit into physical memory! If Java running JIRA starts swapping actively used memory, it will be a performance disaster.

Sample calculations for a host running JIRA and Confluence, with Apache and MySQL:

JIRA	Heap: 2 GB Non-heap: 500 MB
Confluence	Heap: 2 GB Non-heap: 500 MB
Operating system Apache HTTPD MySQL	1 GB
Free memory margin / File buffers	2 GB
Total Physical Memory Required	8 GB