

## Sizing Recommendations for Sage X3 Web Server (Syracuse)

The Sage X3 web server architecture is based on node.js instances that are load balanced. Every node.js process can manage pages for several users.

The Syracuse load balancer distributes user sessions over the different node.js processes. The distribution is based on an algorithm that considers the response time and occupation rate (number of sessions) of every node.js process.

The load balancer will also check their memory consumption, will stop assigning new sessions to a process if necessary (“Restricted mode”), will create new processes to supplement restricted ones, and will even terminate a process if the memory consumption and/or the response time exceeds some parameter values that can be fine-tuned.

It is critical to size the number of nodes and to tune their CPU and memory consumption to get the best performance. node.js is a standard framework that works in asynchronous mode. It is therefore difficult to accurately and conclusively estimate the number of node.js processes and to size them for a customer instance without extensive analysis.

The sizing requirement differs according to the type of page and X3 functions the users run. The heaviest consumption is encountered with classic pages containing grids with a large number of lines (over 1000).

The guidelines in this document are a starting point but should be tuned and modified on site according to observation, if the result is not optimal.

The setup of the configuration is based on two elements:

- ❖ **Number of node.js child processes:** The number of node.js processes is set-up in the host administration page (Administration menu > Administration > Servers > Hosts)
- ❖ **Memory sizing:** Memory sizing for each node.js process is defined in a configuration file: **nodelocal.js** (Located in ..\Syracuse\bin in the Syracuse installation directory).

### Number of node.js Processes

The number of Node processes depends on the maximum number of session requests during peak activity.

You can use a set of simple rules to initially size a Sage X3 Syracuse (Web) server:

- For normal usage, a node.js process can serve up to about 25 typical interactive user **sessions**. If some or all those sessions have a high **activity level** this number should be lowered (see below for examples of high activity levels).
- A **session** is a single browser tab running a Classic Page function, or the set of tabs opened by a user running functions in Syracuse mode.

- The **activity Level** of a session means the actual operations performed by a session. As an example, if there are 50 users who perform occasional entry of small sales orders or supplier invoices, or who occasionally perform small enquiries, but whose sessions are relatively inactive most of the time, then it might be possible to serve them all through a couple of node.js processes. Inversely, if they perform heavy operations intensively, such as creating hundreds of sales order lines per minute or using large enquiries with thousands of lines, then you need to spread the sessions over more node.js processes.
- A single dedicated logical processor (CPU core) can run 2 to 4 node.js processes (depending on their activity rate).
- The rule above also applies to node.js processes that are dedicated to Web Services, but in that case memory consumption is normally lower because Web Service sessions are usually stateless.

The number of node.js processes can be set up through the **Administration menu**: *Administration > Servers > Hosts*, mainly through the **Number of Child Processes** setting. You do not need to restart the server when increasing the number of Child Processes, but when decreasing them you need to make sure you are the only user connected to the application

Some recommendations can be given to the users to reduce memory and CPU pressure on the node.js processes:

- Please note that all the tabs opened in the same browser of a user *are assigned to a single node.js process* because they belong to the same session. If your tabs contain large classic screens with lots of lines, it is preferable to *limit the number of simultaneous browser tabs* to avoid memory over-consumption for a single node.
- The memory associated to classic pages is only released *when the function is closed*. Make sure the browser tabs are closed when not needed anymore.

## Sizing of node.js Processes

A default Syracuse installation has default sizing settings that define the memory consumption of each node.js process, but you can modify them and also use advanced parameters defined in the nodelocal.js configuration file.

The configuration file is located on the Syracuse Server (**..\Syracuse\bin\nodelocal.js**) and contains default parameters settings. The nodelocal.js file is never modified during an update and must always be set up manually. The modification will only be effective when you restart your Syracuse Server.

**Warning:** It is recommended you backup the file before any modification. Incorrect Json syntaxes in nodelocal.js will prevent the Syracuse server from starting.

The default **nodelocal.js** file supplied at installation does not contain any of the advanced sizing parameters but experience shows that *it might be necessary to change default values* to cover most heavy activity classic page users. The table below lists the settings and their recommended values.

- **memoryThreshold1:** Defines the limit (in MB) beyond which a node.js process will be placed in restricted mode, i.e. it will not accept any new connections. This value can be extended up to 8000 (if max\_old\_space\_size has been set-up accordingly, see below). When a node.js process is placed in restricted mode, it stops being part of the load balancing group and another process is created to take its place. The current (restricted) process continues to handle its current sessions normally but will not accept any new connection. The new process (that will show in the process list as “no\_cleanup\_Nx”) runs just like a normal node.js process.
- **memoryThreshold2:** defines the memory consumption limit (in MB) for a node.js process. When the memory consumed exceeds that value, the load balancer will terminate the node.js process. This value must always be higher than memoryThreshold1. The no\_cleanup\_Nx node is also terminated when it exceeds memoryThreshold1.
- **maxRestrictedChildren:** Maximum number of Node processes in restricted mode. Any additional node that turns to restricted mode is terminated.
- **max\_old\_space\_size:** This is the maximum memory allocation possible for the “old space” of a node process, in MB. The “old space” is the section of node’s memory where expired memory objects are stored.
- **childPingStatusTimeout:** When a child process takes longer than the number of milliseconds specified here to respond to the load balancer’s ping, it is terminated.

**The recommended values for those advanced parameters are as follows:**

|                        | <b>Value recommended for normal/medium usage</b><br>(Inquiries < 1000 lines , no large Excel exports)  | <b>Value recommended with large classic screens</b> (Inquiries with more than 1000 lines, large Excel exports, multiple tabs with large screens...) |
|------------------------|--|---|
| memoryThreshold1       | 2500 MB  | 4000 MB   |
| memoryThreshold2       | 3000 MB  | 5000 MB   |
| childPingStatusTimeout | 30000 ms   |   |
| max_old_space_size     | 8192 MB  |   |
| maxRestrictedChildren  | Depends on the number of node processes : <ul style="list-style-type: none"> <li>▪ <b>2</b> if 5 nodes or less</li> <li>▪ <b>1</b> if more than 5 nodes</li> </ul> |   |

## Nodelocal.js example

Here is an example of a configuration that can be adapted:

```
39  "session": {
40    "timeout": 20,
41    "checkInterval": 60,
42    "auth": "basic"
43  },
44  "searchEngine": {
45    "hostname": "x3erpvl1vm",
46    "port": 9200
47  },
48  "x3fusion": {
49    "records": {
50      "dumpPath": "C:\\Sage\\SAFEX3\\X3ERPv11SYRSVR/syracuse/cache/_cvg__USERNAME_"
51    }
52  },
53  "nanny": {
54    // Number of Nodes in restricted mode before killing the next Node turning in Restricted Mode:
55    "maxRestrictedChildren": 1,
56    // childPingStatusTimeout (milliseconds)
57    "childPingStatusTimeout": 30000,
58  },
59  "system": {
60    // load balancer will not assign new sessions any more when the heap usage exceeds this value (in MB)
61    "memoryThreshold1": 2500,
62    // process will be killed during load balancer ping operation when the heap usage exceeds this value (in MB)
63    "memoryThreshold2": 3000,
64  },
65  "hosting": {
66    // node.js options
67    "nodeOptions": "--max_old_space_size=8192"
68  }
69 };
70
```

## Server Resources

The total memory and CPU capacity on the Syracuse server must be tailored to your settings, of course. Total memory must be able to accommodate all node processes *at their peak usage*, and also the additional “no\_cleanup” node processes that are allowed by *maxRestrictedChild*, which must be taken into account. The maximum memory consumption of a node is defined by *memoryThreshold2*.

In addition, you must consider the operating system and any other applications running on that machine. Make sure you have enough memory available! Also make sure you look at appropriate swap space on the server's disks.

The total number of cores is calculated similarly based on the guidelines in this document. Also make sure you have enough physical cores, proper socket distribution and the proper power management settings (set to 'Performance', and not 'Balanced') to account for the number of vCPUs if you are running a virtualized environment.

## Memory Consumption Examples

Memory consumed by a Sage X3 connection is directly linked to the activity of that session. As an example, running a large classic inquiry or requester (a grid screen) with 10 000 lines displayed on screen will cause the related node.js process to typically consume several hundred Megabytes holding the full data structure. Syracuse pages do not cause as much memory consumption as paging/caching is used natively. Here are a few typical examples as an illustration:

| <b>Sage X3 Functions</b>                 | <b>Display</b>                | <b>Session's node.js Memory Consumption</b> |
|--|-------------------------------|---|
| Account Inquiry (CONSCPT): 10 000 lines  | 500 lines per page, 20 pages  | ~600MB                                      |
| Manual Matching (LETTARGE): 30 000 lines | 15 lines per page, 2000 pages | ~950Mb                                      |
| Manual Matching (LETTARGE): 30 000 lines | 500 lines per page, 20 pages  | ~850MB                                      |