

WEB GARAGE

APPLICATION

- The WebGarage platform responds to this challenge: to offer our customers a state-of-the-art supervision and programming environment, ensuring the continuity of field installations and the integration of new generations of devices.
- The system integrates the basic functionality of a Building Automation System (BAS) which supervises and controls HVAC systems, lighting and other plants. It offers the additional benefits of a Building Operating System (BOS) to govern, display and give value to relevant data. The technology on which it is developed uses tagging and data modeling to ensure functionality and potential still unexplored.
- WebGarage not only makes it easy to create dashboards but it also offers advanced analytics functions to support the diagnosis of maintenance gaps and optimize building management performance. Based on the open standard of the Haystack project, WebGarage opens up a multitude of possible integrations and solutions from the Haystack collaborative community

Click on yhe two items below

- Fill in the form to apply for the WEBGARAGE license:
http://www.costergroup.eu/admin/mgr_moduli/Risposte_Utente.asp?FUN_ID=23&MEC_ID=412&MEC_IDFiglie=0&MGRCONTROL=S

Features

RECOMENDATION

- RAM: 8-16GB
- Operating System: Windows, Linux, Mac
- Interface: Gigabit: LAN
- Flash: 32GB SSD
- CPU: x86-64 CPU (Intel Xeon, Intel Core, AMD Ryzen)
- CPU Power: 2GHz+
- The technical specifications should be adapted to the size of the project or projects. With a high number of historicized points and/or high number of logics, it is recommended to consistently size the CPU and RAM for better use.
- The most important resource for server sizing is RAM. For any medium to large project, you must properly tune your JVM heap size.
- A good rule of thumb is to plan for 1GB to 2GB of heap for every 10K points.
- Heap size more than 1.5GB will need to run a 64-bit JVM (the actual max heap size of a 32-bit JVM will vary slightly). 85% of Java Heap consumption is roughly our recommended max before considering to upgrade the amount of space.
- Open JDK is recommended but you can also use Java 11
- Another important aspect to consider is that the historian is mostly limited by I/O performance (speed of disk drive). Using a SSD will likely have a significant performance boost for reading and writing history data. Disk space is typically not a issue for projects due to the Folio's highly efficient compression techniques. Analysis of projects has shown an average about 1.5 bytes per timestamp sample which equates to about 51KB per year per point for 15min interval data.
- Depending on the apps/features you use (and don't use) will have a significant impact on performance and how much horsepower you will need to run smoothly. Running multiple projects on a single host consumes more overhead; 100 little projects of 100 points are more resource consuming than a single 10K point project

Code	Description	Data Sheet
CWS 200	Server Version license, 200 points managed (of which up to 40 historicised)	–
CWS 500	Server Version license, 500 points managed (of which up to 100 historicised)	–
CWS 01K	Server Version license, 1000 points managed (of which up to 200 historicised)	–
CWS 02K	Server Version license, 2000 points managed (of which up to 400 historicised)	–
CWS 05K	Server Version license, 5000 points managed (of which up to 1000 historicised)	–
CWS 10K	Server Version license, 10000 points managed (of which up to 2000 historicised)	–
CWS 50K	Server Version license, 50000 points managed (of which up to 10000 historicised)	–
CWS 1HK	Server Version license, 100000 points managed (of which up to 20000 historicised)	–
CWC 100	Add-On license for Server Version. Increases the number of historicized points by 100 points (only for Server Version license)	–
CWC 01K	Add-On license for Server Version. Increases the number of historicized points by 1000 points (only for Server Version license)	–
YHC CWE	Network manager ModBus with embedded WEBGarage license of 200 points (of which up to 40 historicised)	–