webdyn

flexitron group

WebdynSunPM

User Manual

www.webdyn.com

Index

Glossary 5

About this Document 8

1. Presentation 10

1.1 General Description 10

1.2 Principles of Operation 10

1.3 The Interfaces 11

1.4 Supported Devices 12

1.5 Product References 12

1.6 Technical Specifications 13

1.6.1 General Specifications 13

1.6.2 Technical Specifications 14

1.6.3 Software Specifications 16

1.7 Safety Instructions 17

1.8 Regulation 17

2. Installation and Maintenance 19

2.1 Prerequisite 19

2.2 Unpacking 19

2.2.1 Content 19

2.2.2 Identification 19

2.3 Assembly 22

2.3.1 Opening/Closing the Box 22

2.3.2 Wall Mounting 22

2.4 Interface Description 23

2.4.1 Product Power Supply 23

2.4.2 Cellular Network 24

2.4.3 Indicators & Buttons 26

2.4.4 Extension Interface 30

2.4.5 Ethernet Interface 31

2.4.6 RS485/RS422 Serial Interface 32

2.4.7 Input/Output Interface 35

3. Configuration 41
3.1 ..... FTP/SFTP/WebDAV 41
3.1.1 Operating Principle 42
3.1.2 Configuration Files 42
3.1.3 Updates 69
3.2 Embedded Web Interface 70
3.2.1 Devices 72
3.2.2 Settings 93
3.2.3 System 136
3.3 Carte Micro SD 144
4. Operation 146
4.1 The FTP/SFTP/WebDAV Server 146
4.1.1 The Configuration “CONFIG” 149
4.1.2 “DEF”, the Definitions 150
4.1.3 “DATA” 151
4.1.4 “ALARM” Alarms 157
4.1.5 “CMD” Commands 159
4.1.6 “SCRIPTS” 159
4.1.7 “BIN” Update 160
4.1.8 “LOG” 160
4.1.9 Web Services 167
4.2 The MQTT/MQTTS server 171
4.2.1 Data format 172
4.3 microSD card 181
5. Commands 183
5.1 Principle 183
5.2 How it Works 183
5.2.1 Command file 183
5.2.2 MQTT command message 185
5.2.3 SMS 186
5.3 List of Commands 186

5.3.1 “connect”: Trigger a connection 188
5.3.2 “status”: Retrieval of the status of the concentrator 189
5.3.3 “factory”: Return to factory settings 190
5.3.4 “reboot”: Rebooting the concentrator 191
5.3.5 “updateFirmware”: Concentrator software update 191
5.3.6 “apn”: Modem configuration 192
5.3.7 “ftp”: Configuration of the FTP/SFTP server 193
5.3.8 “log”: Activation of equipment communication logs 194
5.3.9 “setRelay”: Changing the state of the relay 195
5.3.10 “discoverDevices”: Discovery of equipment 195
5.3.11 “getParameters”: Collection of parameters 196
5.3.12 “getData”: Collection of action code variables 6 or 7 197
5.3.13 “writeVariable”: Writing a variable on a device 198
5.3.14 “setKey”: Added keys for decrypting client scripts 198
5.3.15 “deleteKey”: Removing keys for decrypting client scripts 200
6. Update 202
6.1 Using the Web Interface 202
6.2 Using FTP/SFTP/WebDAV 202
6.3 By SMS or MQTT/MQTTS command 203
6.4 By micro SD card 203
7. Tools & Diagnostics 204
7.1 Diagnostics 204
7.2 Tools 204
8. FAQ 205
9. Appendices 210
9.1 Appendix A: “_config.ini” file 210
9.2 Appendix B: Time zone list 226
Offices & Support Contact 228

Glossary

NAME	DESCRIPTION
2G	Second Generation: second generation (2G) digital standard for cell phones including GSM, GPRS and EDGE.
3G	Third Generation: third generation (3G) digital standard for cell phones including UMTS, HSPA, HSPA+ and DC-HSPA+.
4G	Fourth Generation: fourth generation (4G) digital standard for mobile telephony including LTE-Advanced.
AES	Advanced Encryption Standard: symmetrical encryption algorithm.
APN	Access Point Name: the name of the access point the gateway uses to connect to the Internet via a mobile connection.
Broker	MQTT server in charge of receiving published information in order to transmit it to subscribed clients. The broker has a relay role.
CSV	Comma-separated values: open text format representing tabulated data in the form of values separated by semi-colons. This format makes it easy to use data with spreadsheet software such as Excel.
DNS	Domain Name System: distributed computer service used to translate Internet domain names to IP addresses.
FTP	File Transfer Protocol: communication protocol used to exchange files over a TCP/IP network.
HTTP	HyperText Transfer Protocol: client-server communication protocol developed for the Web.
IMEI	International Mobile Equipment Identity: number used to uniquely identify each modem.
IMSI	International Mobile Subscriber Identity: unique number stored in the SIM card used by a cell phone network to identify a user.
IP	Internet Protocol: message protocol in charge of addressing and sending TCP packets over the network.
Lua	Script language. See http://www.lua.org/ for more details.

Modbus	Modbus is a communication protocol routinely used by industry to dialogue with industrial equipment over a network. See http://www.modbus.org/ for more details.
MQTT	Message Queuing Telemetry Transport: publish-subscribe messaging protocol based the TCP/IP protocol.
MQTTS	Secure Message Queuing Telemetry Transport.
NTP	Network Time Protocol: protocol used to synchronise the local concentrator clock with a time reference via a computer network.
DIN rail	Standard 35 mm metal rail used in racked industrial control equipment in Europe.
RSSI	Received Signal Strength Indication: reception power level measurement of a signal issued by a radio antenna.
RTU	RTU mode is an RS422/485 hard-wired bus for Modbus.
S0	Standardised pulse from meters (water, gas, electricity, etc.) as per the NF EN 62053-31 standard.
SFTP	SSH File Transfer Protocol: communication protocol using a secure SSH communication protocol. Its use is similar to FTP.
IS	Information System: server with which the concentrator exchanges (configuration, data, alarms, etc.).
Sunspec	Open communication protocol for inverters based on Modbus and compliant with the SunSpec alliance standards (See https://sunspec.org/for more details).
TCP	Transmission Control Protocol: an Internet-based connection-oriented protocol that provides data packet segmenting services that the IP protocol sends over the network. This protocol provides a reliable data transfer service. See also IP.
TCP/IP	Transmission Control Protocol/Internet Protocol: a set of network protocols that provide interconnection services between computers of different hardware architectures and operating systems. TCP/IP includes standards for communication between computers and conventions for network interconnection and routing.
TIC	Customer remote information: digital data output from ERDF meters that permanently broadcasts the managed contractual parameters as well as the consumption magnitudes measured by the meter.
Topic	MQTT information channels that publishers use to send messages. These messages can be read by subscribers.

UDP	User Datagram Protocol: non connection-oriented protocol of the TCP/IP model transport layer. This protocol is very simple because it does not provide error checks (it is not connection-oriented...).
UTF-8	Universal Character Set Transformation Format1 - 8 bits: computer character encoding designed to encode all the characters from the 'Universal encoded character set".
WebDAV	Extension to the HTTP protocol to improve the management of remote files. As part of the WebdynSunPM, the WebDAV protocol is used over HTTPS. We are therefore talking about WebDAV-HTTPS.

About this Document

This guide describes all the WebdynSunPM product characteristics.

Its purpose is to help operators install and configure their WebdynSunPM and to allow operators to include collected data in their IS.

This manual is split into six separate sections:

- Section 1: General presentation
- Section 2: Installation
- Section 3: Configuration
- Section 4: Operation
- Section 5: Tools & diagnostics
- Section 6: FAQ

Scope

This technical description is valid for WebdynSunPM concentrators as from hardware version V1 and software version V3.0.0.

Target Audience

This guide is intended for all people involved in photovoltaic system supervision, in particular people in charge of local or remote installation maintenance, as well as for the developers of portals designed to use the sent data.

It is recommended to entrust the installation and commissioning of the WebdynSunPM to qualified and trained persons. Qualified persons must have the following skills:

- Detailed knowledge of network management services.
- Knowledge of IP-based network protocols.
- Knowledge of the specifications of the protocols used (Modbus, SunSPec, etc.) and of the equipment connected to the concentrator.
- Training in the installation and configuration of computer systems.
- Knowledge of and compliance with this document and all safety information.

Please contact your sales representative (contact@webdyn.com) to find out the list of partners.

Document Versions

VERSION	CONTENT
V2.05	Manual creation
V3.0	Micro SD card management Added WebDAV Added MQTT Specific appendixes for ICT and proprietary protocols Modification of the edition and creation of equipment Added WebdynSunPM 4G version
V4.2.17	Adding variables in the SCL script file Addition of the WebService Add option number of acquisitions in DATA files Script GUI update Added client encrypted LUA scripts Added Webdyn scripts with license Addition of forced write codes in definition files Addition of action codes 6, 7 and 9 in the definition files Changing the format of MQTT data (metadata) Added “getData” command

1. Presentation

1.1 General Description

The WebdynSunPM is a concentrator designed to monitor all types of photovoltaic installation. It is used to collect, analyse, monitor and control the on site devices.

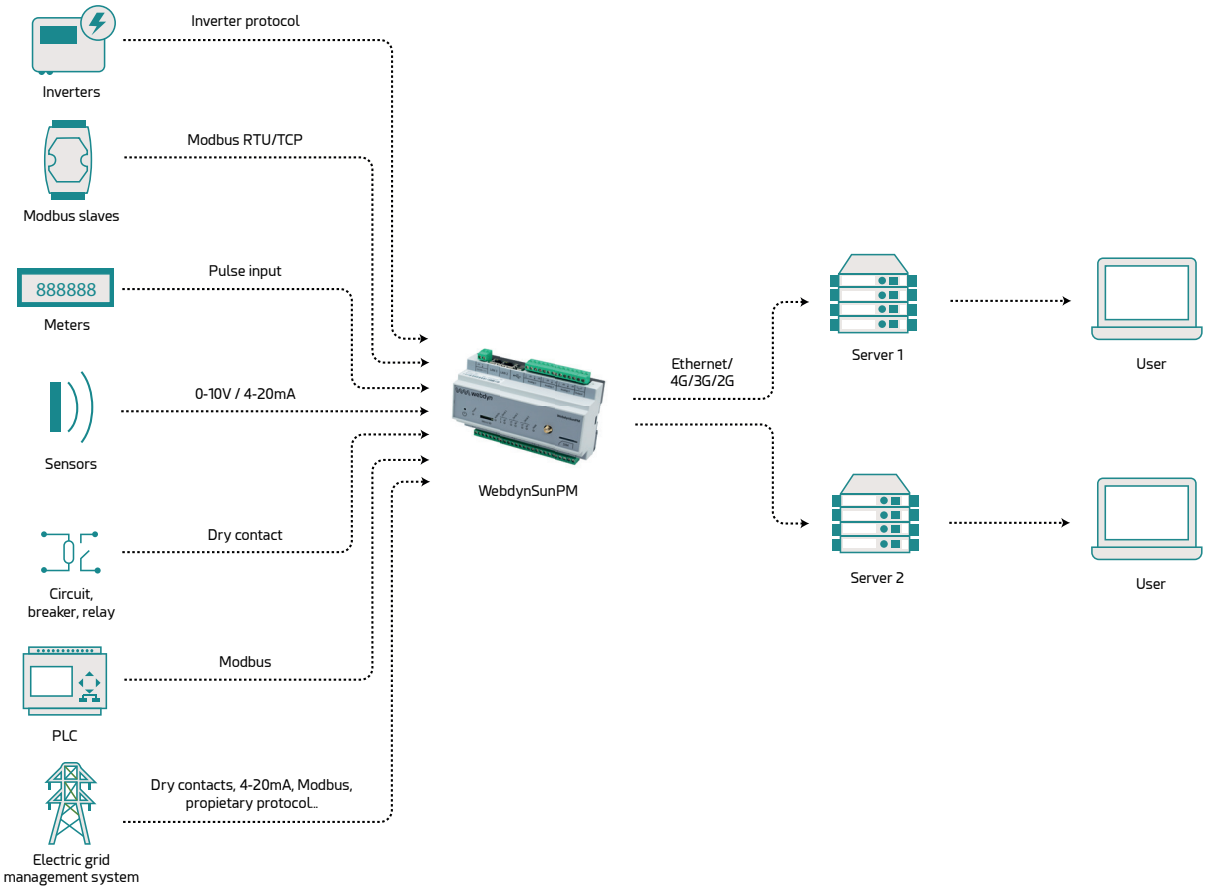
The collected information (data, parameters, etc.) is formatted before being sent to an Information System (IS). The concentrator provides the security and confidentiality of the exchanged information.

The automation of certain local actions, for example injection or self-consumption is managed using customisable scripts embedded in the concentrator.

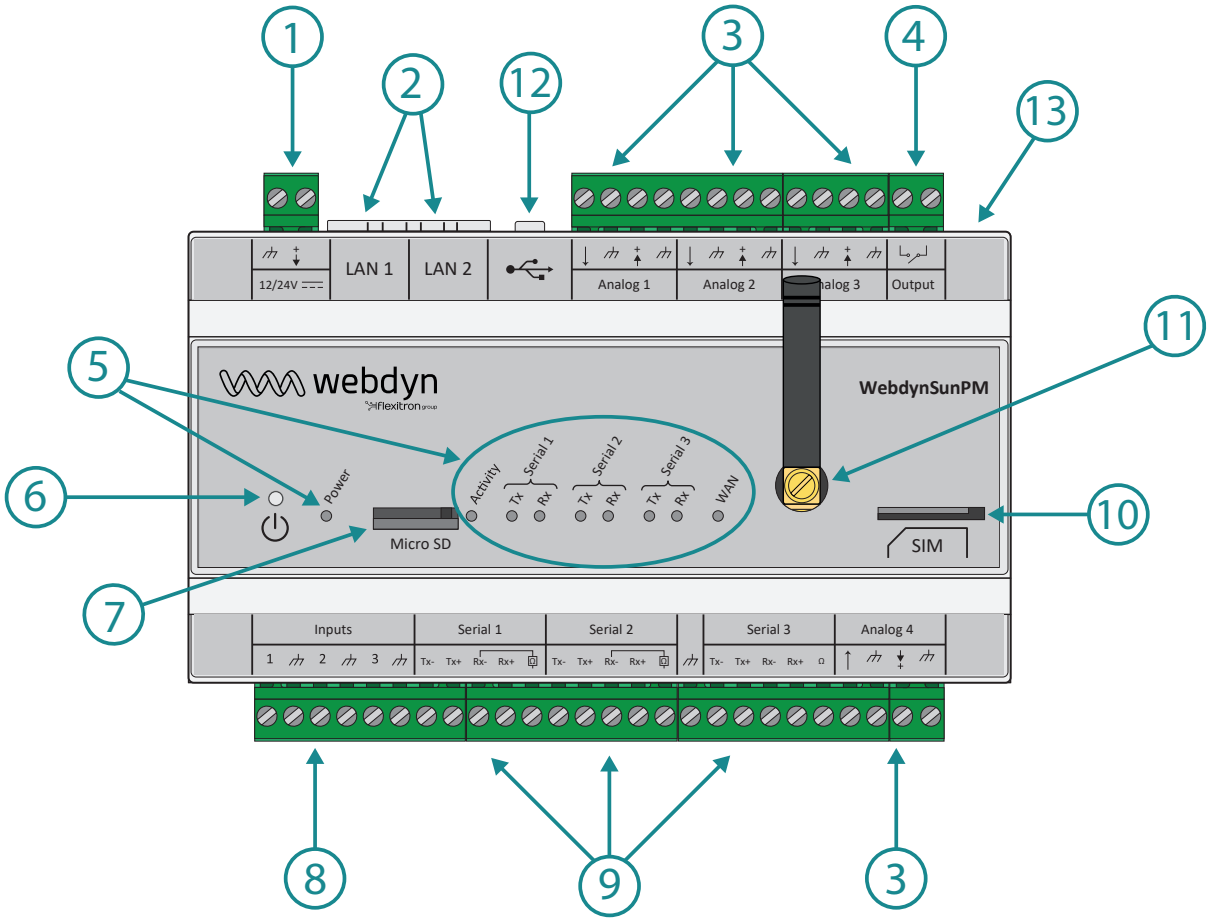
1.2 Principles of Operation

The WebdynSunPM concentrator can be fully integrated into photovoltaic installations. Devices such as inverters, sensors (pyranometers, temperature sensors, etc.), meters, displays, circuit breakers or relays can be connected to the concentrator using its many available interfaces. Information readings and device control are carried out by the WebdynSunPM continuously. The data is regularly formatted and uploaded to an FTP server and/or sent to an MQTT server using the modem or an Ethernet connection. The concentrator manages the SFTP and MQTTS protocols to secure exchanges with the servers. The WebdynSunPM can be configured by simply sending text message commands, but also using the embedded web pages.

Specific scripts can be created for installations requiring specific functions.



1.3 The Interfaces



1. 12-24 V power supply terminal block
2. 2x Ethernet ports (LAN 1 and LAN 2)
3. 4x 0-10 V or 4-20 mA analog inputs
4. 1x Relay output (24V/1A)
5. Indicators:
    - Power: 12-24 V power supply status
    - Activity: Product status
    - Serial1, Serial2, Serial3:
    - Tx: Data sent on the RS485/RS422 serial port
    - Rx: Data received on the RS485/RS422 serial port
    - WAN: Connection status
6. Power button
7. Micro SD card slot

8. 3x Digital inputs (ON-OFF or pulsed S0)

9. 3x RS485/RS422 ports

10. SIM card holder

11. Modem SMA antenna connector

12. USB port (for extension)

13. RESET button

1.4 Supported Devices

The WebdynSunPM is compatible with all devices that include one of the supported protocols.

Partial list of the supported protocols:

INVERTER PROTOCOL	PHYSICAL INTERFACE	SPECIFICATIONS
SMA-net	RS485/RS422 2 wires	100 max
Modbus TCP	Ethernet	254 max
Modbus RTU	RS485/RS422 2 wires or 4 wires	247 max
Power-One Aurora	RS485/RS422 2 wires	100 max
TIC	USB (cable optional)	1 max
Delta-Solivia	RS485/RS422 2 wires	100 max

Currently, much inverter equipment runs using Modbus. The concentrator accepts all Modbus RTU and TCP devices.

1.5 Product References

Product:

REFERENCES	DESCRIPTIONS
WG0517-A01	WebdynSunPM (Europe and India version)
WG0517-A02	WebdynSunPM (Monde version)
WG0517-A03-DEIE	WebdynSunPM in DEIE box (Europe and India version)

WG0517-A04	WebdynSunPM 4G (Europe and India version)

1.6 Technical Specifications

1.6.1 General Specifications

SPECIFICATIONS	DESCRIPTIONS
Power supply	+12V: 700mA (some functions are not supported on 12V) or +24V: 350mA
Battery	650mAh 3.7V, Lithium-polymer
Consumption	P: 5 W Pmax: 10 W
Dimensions	155x 106 x 58mm 9 DIN rail modules
Box	RoHS compliant DIN EN 60715 TH35 DIN VDE 0470-1 DIN 43880 size 1 REACH compliant VBG 4 IEC 529 Non leak tight
Fixing	DIN rail
Weight	0.330 kg
Operating temperature	-5 °C / +40 °C
Storage temperature	Storage: -20 °C / +85 °C
Humidity	25 - 75 %
Pollution rating	2
Certification	RED ROHS REACH

Regulation	CE marking created in the framework of European technical harmonisation legislation. It is mandatory for all products covered by one or more European regulatory texts (directives or regulations).
	Symbol indicating that the waste must be collected via a specific channel and must not be disposed of as household waste.
	Symbol indicating that the product must be recycled.

1.6.2 Technical Specifications

SPECIFICATIONS	DESCRIPTIONS
Memory capacity for data	DDR3 SDRAM: 512 Mb. Flash eMMc: 8 Gb in total (50Mb max per defined device)
SD card	MicroSD MMC/SD/SDIO (up to 32 Gb)
Cellular Interface Modem	Europe and India version modem • 2G (EDGE, GSM, GPRS): 900MHz, 1800MHz • 3G (HSPA): B1 and B8 Monde version modem: • 2G (EDGE, GSM, GPRS) : 850MHz, 900MHz, 1800MHz, 1900MHz • 3G (HSPA) : B1, B2, B5, B6, B8 et B19 Europe and India version 4G modem • 2G (EDGE, GSM, GPRS) : 850MHz, 900MHz, 1800MHz, 1900MHz • 4G (LTE) : B1, B3, B5, B7, B8, B20 et B28 Antenna: External SMA
SIM format	Standard SIM (mini SIM) 2FF format 1.8V and 3Vcompatible
Ethernet interface	2x 10/100 Mbits/s ports available
USB interface	1x USB2.0 port
Serial interface	3x RS485/RS422 ports

Input/Output interface	4x analog 0/10V or 4/20mA inputs 3x digital ON-OFF or pulsed S0 inputs (class A or B) 1x relay output (24V/1A)

Webdyn does not supply any SIM cards. Please contact an M2M operator that supports the 2G/3G or 2G/4G networks.

Connectivity data 2G/3G Europe and India version:

RF BAND	EMISSION FREQUENCIES	MAX POWER
UMTS B1	1922 MHz - 1978MHz	22.5 dBm(+-1.5dB)
UMTS B8	882 MHz - 913 MHz	22.5 dBm(+-1.5dB)
E-GSM 900	880 MHz - 915 MHz	33 dBm (+-2dB GSM,GPRS)
DCS 1800	1710 MHz - 1785 MHz	30 dBm (+-2dB GSM,GPRS)

Connectivity data 2G/3G Monde version:

RF BAND	EMISSION FREQUENCIES	MAX POWER
UMTS B1	1922 MHz – 1978 MHz	23 dBm(+-2dB)
UMTS B2	1852MHz – 1908 MHz	23 dBm(+-2dB)
UMTS B5	826 MHz – 847 MHz	23 dBm(+-2dB)
UMTS B6	832 MHz – 838 MHz	23 dBm(+-2dB)
UMTS B8	882 MHz - 913 MHz	23 dBm(+-2dB)
UMTS B19	832.4MHz – 842.6 MHz	23 dBm(+-2dB)
GSM 850	824 MHz – 849 MHz	33 dBm (+-2dB GSM,GPRS,EDGE)
E-GSM 900	880 MHz - 915 MHz	33 dBm (+-2dB GSM,GPRS,EDGE)
DCS 1800	1710 MHz - 1785 MHz	30 dBm (+-2dB GSM,GPRS,EDGE)

PCS 1900	1850 MHz – 1910 MHz	30 dBm (+-2dB GSM,GPRS,EDGE)

Connectivity data 2G/4G Europe and India version:

RF BAND	EMISSION FREQUENCIES	MAX POWER
GSM 850	824 MHz – 849 MHz	33 dBm (+-2dB)
E-GSM 900	880 MHz - 915 MHz	33 dBm (+-2dB)
DCS 1800	1710 MHz – 1785 MHz	30 dBm (+-2dB)
PCS 1900	1850 MHz – 1910 MHz	30 dBm (+-2dB)
LTE B1	1920 MHz- 1980 MHz	23 dBm (+-2dB)
LTE B3	1710 MHz – 1785 MHz	23 dBm (+-2dB)
LTE B5	824 MHz – 849 MHz	23 dBm (+-2dB)
LTE B7	2500 MHz – 2570 MHz	23 dBm (+-2dB)
LTE B8	880 MHz – 915 MHz	23 dBm (+-2dB)
LTE B20	832 MHz – 862 MHz	23 dBm (+-2dB)
LTE B28	703 MHz – 748 MHz	23 dBm (+-2dB)

1.6.3 Software Specifications

SPECIFICATIONS	PROTOCOLS/FORMATS
Embedded server	HTTP
Server communication protocol (IS)	• FTP • SFTP • WebDAV-HTTPS • MQTT • MQTTS (generic and compatible with AWS IoT, Azure IoT and Google Cloud IoT)

Modbus	• RTU • TCP
Clock synchronisation	NTP
File format for the server (IS)	• CSV for FTP/SFTP/WebDAV-HTTPS • JSON for MQTT/MQTTS

1.7 Safety Instructions

Follow all the safety instructions in this guide.

Failure to follow these instructions can damage equipment and endanger people.

Electric connection:

- All wiring work must be carried out by a specialised qualified electrician.
- Please follow all the safety instructions featured in the manufacturer's device documentation.

The WebdynSunPM product can be damaged by electrostatic discharges (ESD)

Class 3 equipment: the device operates on safety extra-low voltage (SELV) (50V maximum). The voltage reduction must be obtained using a safety transformer providing safe galvanic isolation between primary and secondary.

This equipment is not suitable for use on premises that may host children.

Do not install the equipment near a heat source or at a height greater than 2m.

To clean the product, only use a slightly damp cloth to gently clean and wipe the surfaces. Never use aggressive chemical agents or solvents that could alter the plastic material or corrode the metal parts.

It is essential to leave a 20 cm empty space around the antenna to optimise the Modem cell sensitivity.

1.8 Regulation

The product is compliant with the European directives according to the EU declaration of Conformity available from Webdyn or from the web site: www.webdyn.com.

Recycling:

The nationally enacted European directives covering batteries and waste electric and electronic equipment govern the actions required to limit the negative impact of the end of product life.

These products are collected separately. Use an authorised battery collection and processing centre or contact Webdyn.

2. Installation and Maintenance

2.1 Prerequisite

As the WebdynSunPM concentrator's role is to send the data it collects to an IS, installation requires knowledge of the concentrator but also of the IS it will upload its data to.

The following elements are required to guarantee a proper installation:

- To have this user manual to hand.
- To have a screwdriver suitable for the connector types available on the WebdynSunPM.
- To have knowledge of the parameters to connect to the IS information system.

It is also strongly recommended to have the elements described below for any intervention on site and to install the product.

- To have a SIM card with an activated M2M subscription (data and text messages (optional)) and knowledge of the supplier's APN. The SIM card call number can be useful.
- Use a remote antenna if radio or cellular modem reception is deteriorated.
- To have a PC for the product configuration or update using the concentrator's Web interface.

2.2 Unpacking

2.2.1 Content

The standard version of the WebdynSunPM concentrator is delivered with:

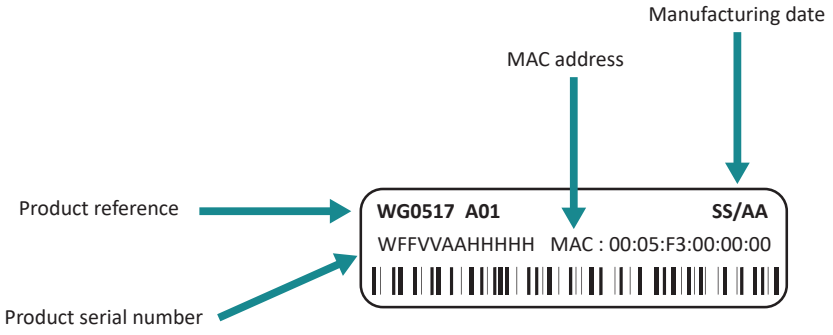
- An angled SMA antenna for the modem (taped to the back of the product).
- A battery (already in place in the product).

2.2.2 Identification

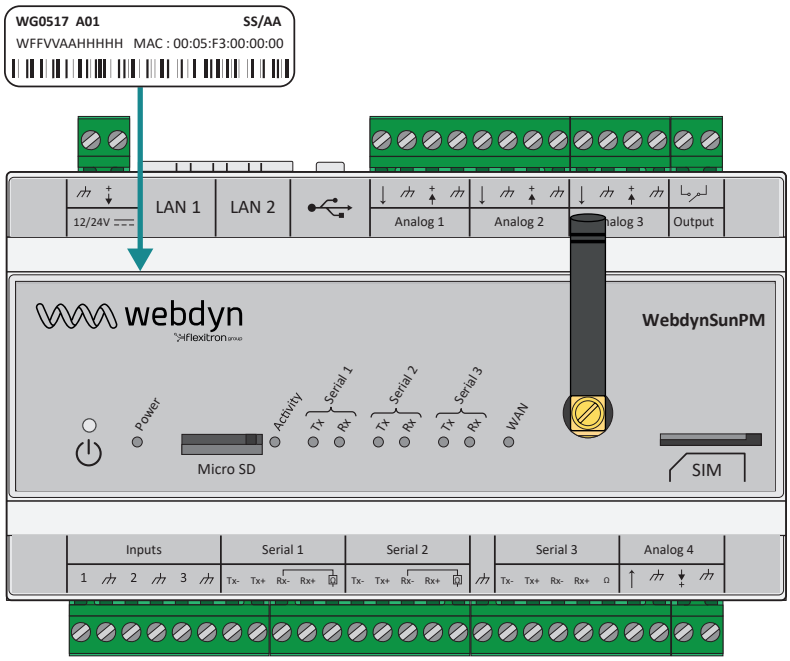
The commercial reference is composed as follows:

- **WG0517-A01**: WebdynSunPM Europe and India version
- **WG0517-A02**: WebdynSunPM in Monde version
- **WG0517-A03-DEIE**: WebdynSunPM in Europe and India version in a DEIE box
- **WG0517-A04**: WebdynSunPM 4G in Europe and India version

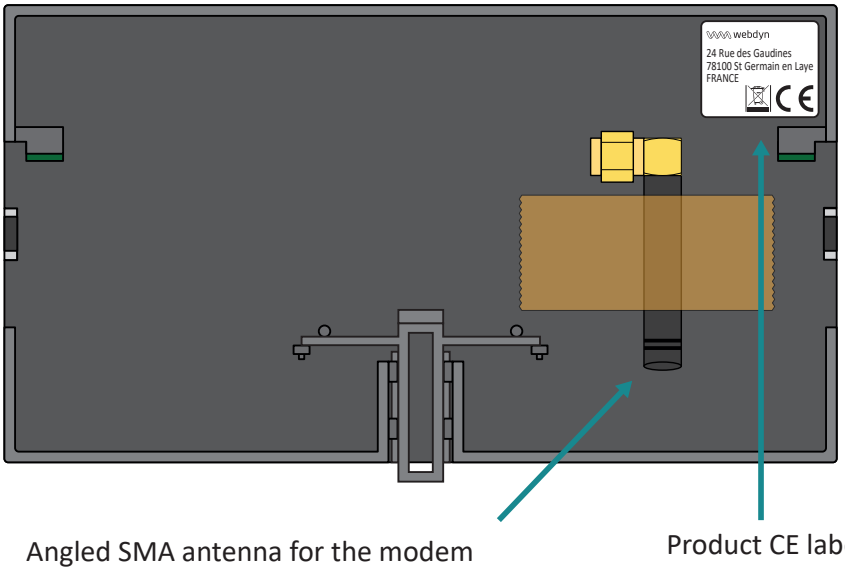
Each product is labelled with the following information:



This label is accessible on the top of the product:



The product CE label on the back of the box:



Software version:

The software version can be found on the concentrator web interface. The software version is available on the “Home” tab (See section3.2: “Embedded web interface”).

2.3 Assembly

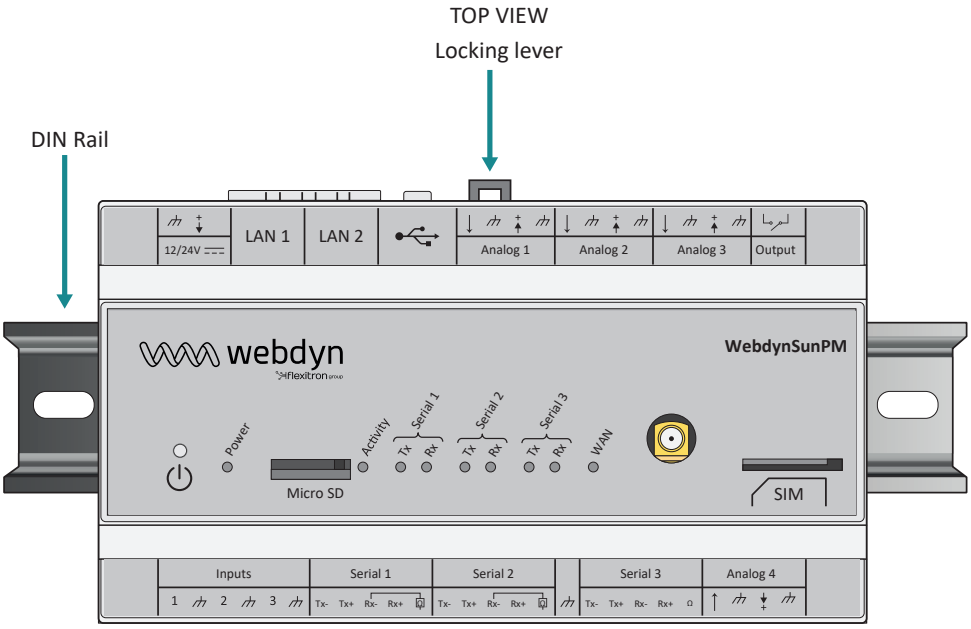
2.3.1 Opening/Closing the Box

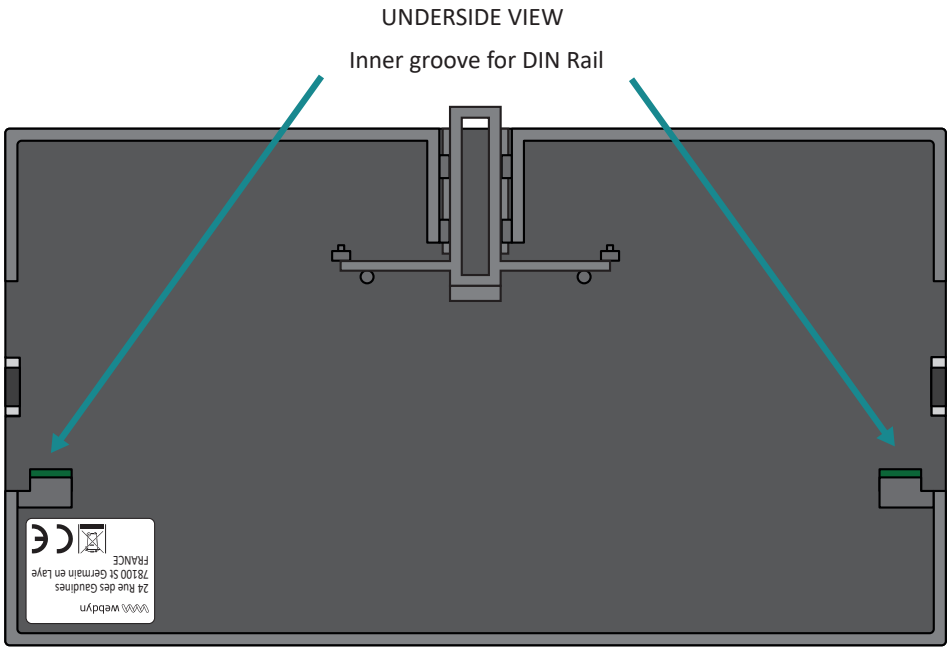
Users must not open the product.

The WebdynSunPM must be returned to after sales for all work (sav@webdyn.com).

2.3.2 Wall Mounting

The WebdynSunPM is designed to be fixed onto a DIN rail.





Follow the steps below to fix the concentrator to a DIN rail:

- Tilt and position the concentrator's lower grooves (see underside view) onto the bottom of the DIN rail.
- Push the concentrator to pivot it upwards.
- Push on the concentrator until it clicks into place.

Follow the steps below to remove the concentrator from a DIN rail:

- Raise the locking lever (see top view) on the concentrator to be removed to the high position. This opens the locking mechanism and makes it possible to remove the concentrator.
- Pivot the concentrator downwards.

Before fixing or removing the concentrator, make sure:

- To cut the power supply to the equipment.
- Remove the antenna taped to the back of the box.

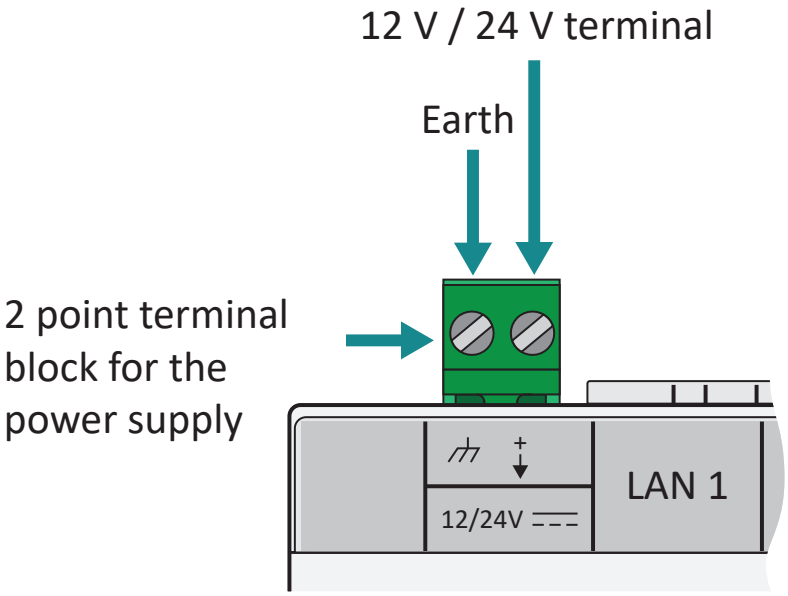
2.4 Interface Description

2.4.1 Product Power Supply

2.4.1.1 External Power Supply

The WebdynSunPM concentrator can be powered using 12V or 24V direct current. The power supply is implemented using the 2 point unpluggable terminal block marked "12/24V" located at the top left of the concentrator.

End users must use a CE certified power supply of less than 15 watts. The distance between the power supply and the product must not exceed 3 metres. End users must make sure their installation meets applicable EMC standards.



Make sure the power supply wires are connected to the proper terminals.

Product power consumption varies depending on its configuration. Make sure the power supply used can provide at least 15 Watts of power.

2.4.1.2 Battery

The WebdynSunPM concentrator has a battery that is used to send an alarm to notify of a power failure fault and switch the product to safety mode until the power supply returns. The battery recharges on the concentrator's external power supply.

The battery may not have time to recharge if the concentrator suffers too frequent or long power cuts.

If the battery status does not allow for the immediate issue of the power loss alarm, it will be sent when the concentrator reboots.

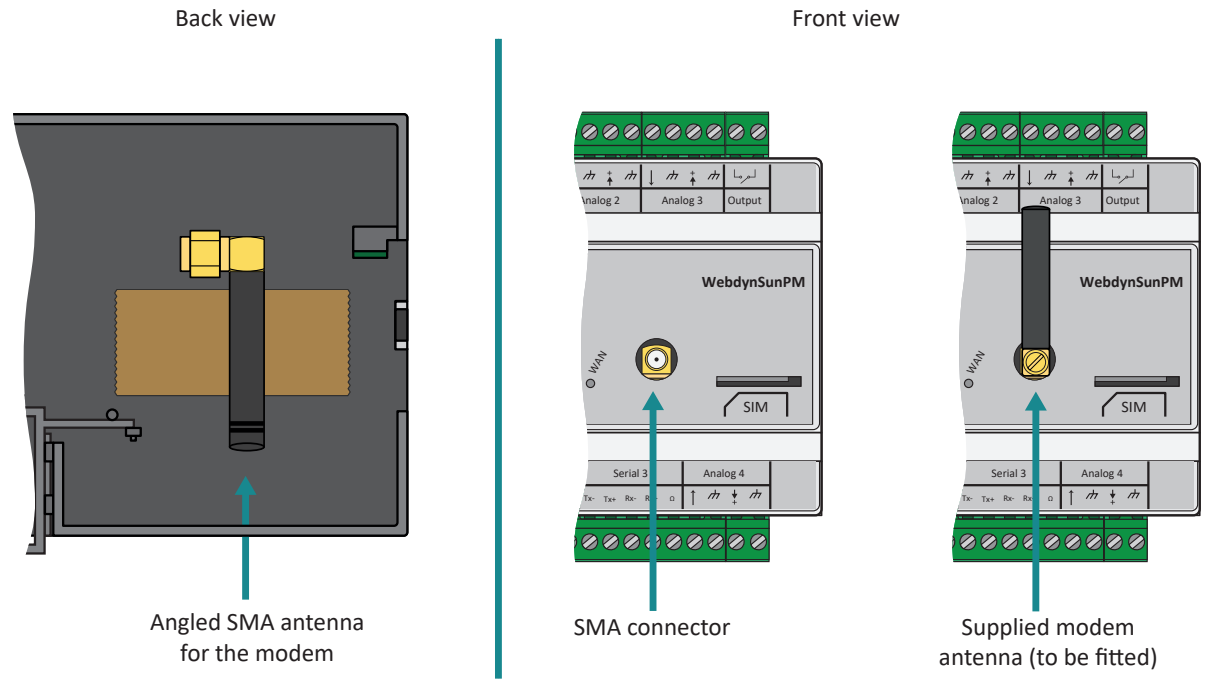
2.4.2 Cellular Network

The WebdynSunPM concentrator has a built-in 2G/3G or 2G/4G network compatible modem.

2.4.2.1 Antenna

The concentrator has a female SMA connector available at the front of the product to connect a modem antenna. The product is delivered with an angled SMA antenna taped to the back of the box. It can be replaced with other compatible antennas.

If the WebdynSunPM concentrator is installed in a metal box or in a location that does not have proper signal reception, the use of an offset antenna is strongly recommended. Be careful to use an antenna compatible with the connector and frequencies used.



End users must make sure their installation using remote antennas meets applicable EMC standards.

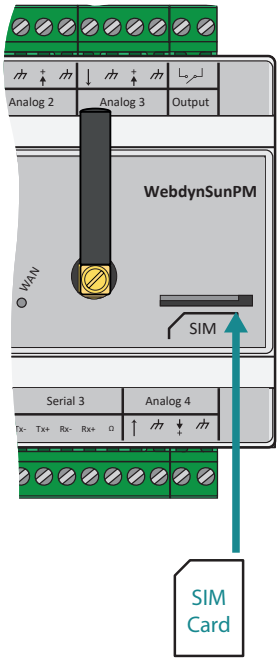
2.4.2.2 SIM Card

To use the 2G/3G or 2G/4G modem connection and allow the concentrator to communicate with the remote server or servers, a mini SIM format SIM card must be inserted in the SIM card housing on the front of the concentrator.

The concentrator is compatible with all market operators as well as with all mini SIM 2FF 25 x 15mm format SIM cards.

To guarantee proper WebdynSunPM operation, insert a SIM card with the following specifications:

- Possibility of sending and receiving text messages (preferable but not essential),
- 2G/3G or 2G/4G communication included.



Turning off the concentrator is recommended before inserting the SIM card to avoid any electrostatic discharge risks.

To insert the SIM card into the product, insert it into the slot on the front of the concentrator until it clicks into position.

To remove the SIM card from the concentrator, briefly press the end of the SIM card protruding from the product until it clicks, then release it. You can then recover the SIM card.

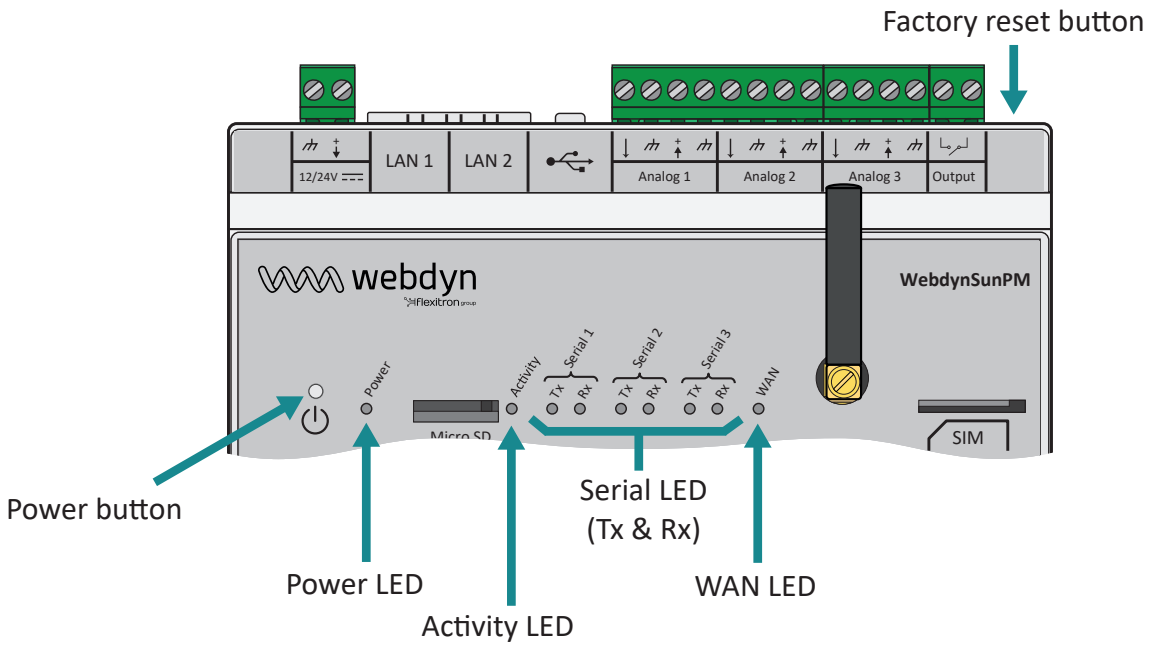
Webdyn does not supply any SIM cards. Please contact an M2M operator that supports the 2G/3G or 2G/4G network or a partner portal.

To find out the information to enter to configure the modem, contact your SIM card provider.

2.4.3 Indicators & Buttons

The concentrator is fitted with:

- 2 push buttons
- 9 indicators



2.4.3.1 Power Button

The Power button on the front of the product is used to shut down and reboot the concentrator.

A long press on the Power button causes the “Activity” indicator to flash every second and is used to trigger the following actions:concentrator reboot: a 2-second press (2 Activity LED pulses)

- Concentrator shut down: a press longer than 5 seconds (At least 5 Activity LED pulses).

When the product has been shut down, it can be rebooted by simply pressing the Power button.

2.4.3.2 Factory Reset Button

The Factory Reset button is used to recover the product in any situation (loss of IP address, loss of login and password, etc.).

A long press on the Factory Reset button causes the “Activity” indicator to flash every second and is used to trigger the following actions:

- IP parameter reinitialisation: a press of between 4 and 14 seconds (at least 4 and at maximum 14 Activity LED pulses).
- Reinitialise all parameters and all data: a press longer than 15 seconds (at least 15 Activity LED pulses).

2.4.3.3 Power LED

The Power indicator shows the WebdynSunPM concentrator power up. The indicator is:

- On: the concentrator is running. There is a voltage of between 12 and 24 volts at the power supply terminals.
- Off: the concentrator is stopped. There is no voltage on the power supply terminal block.

2.4.3.4 Activity LED

The Activity LED shows the product running statuses, which are:

- Fast flashing: Occurs in 3 cases which are:
  - One of the buttons has been pressed.
  - A concentrator update is in progress.
  - The concentrator start-up phase.
- Slow flashing: Normal concentrator operation.

Pressing the Power button changes the Activity indicator default operation. During a long press on the Power button, the Activity indicator will flash every second to help the user complete an action.

2.4.3.5 Serial LED

The bus has 3 RS485/422 serial buses and each bus has an associated Tx indicator and RX indicator.

The Tx indicator will flash depending on the serial transmission dialogue on the concentrator (sending from the concentrator to device). If the Tx indicator does not flash, it means there is no transmission on the serial bus.

The Rx indicator will flash depending on the serial reception dialogue on the concentrator (sending from a device to the concentrator). If the Rx indicator does not flash, it means there is no reception on the serial bus.

For 2-wire wiring (see section 2.4.6: “RS485/RS422 Serial interface”), the data emitted by the concentrator is received by echo. These do not cause the “Rx” indicators to flash.

2.4.3.6 WAN LED

The purpose of the WAN indicator is to help the user know the connection status. The indicator can have 3 different colours (green, orange and red) and depends on the primary server configuration.

Primary server on Ethernet interface:

WAN INDICATOR	STATUS	MEANING
	Off	No connection attempts
green	Slow flashing	Last FTP/WebDAV connection OK
green	Fast flashing	FTP/WebDAV connection in progress
orange	Slow flashing	NTP synchronisation problem
red	Slow flashing	FTP/WebDAV connection problem

Primary server on Modem interface:

WAN INDICATOR	STATUS	MEANING
	Off	No connection attempts
green	X slow flashes followed by a 1 second pause	Indicates the modem signal level by flashing: 1. Unstable 2. Limit 3. Correct 4. Good 5. Excellent
green	Fast flashing	FTP/WebDAV connection in progress
orange	Slow flashing	NTP synchronisation problem
red	Slow flashing	FTP/WebDAV connection problem
red	Fast flashing	Problem attaching to the cell network or unstable reception signal (RSSI < -89 dBm)
red	Steady	SIM card error (missing, no PIN code, PUK code)

Primary server on “SD Card” interface:

WAN INDICATOR	STATUS	MEANING
	Extinct	No attempt to use SD card
green	Flashes slowly	Indicates that the configuration is correct: the SD card has been detected
red	Fixed	SD card error. The SD card was not detected. Check that it has been inserted correctly
red	Blinks quickly	SD card error. A write error has been detected on the SD card. Check that it is well formatted and that the directories have all been correctly created. If necessary, a replacement of the SD card is recommended

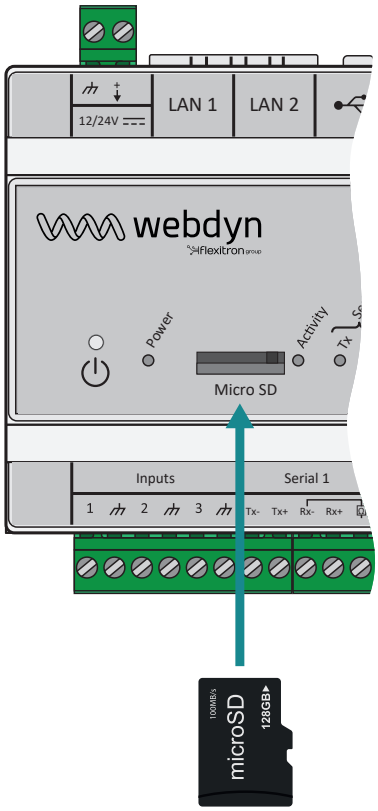
If an error occurs, the WAN LED stays on the last error until the next conclusive attempt or the product reboots.

2.4.4 Extension Interface

2.4.4.1 External Memory Medium (MicroSD)

A micro SD slot is available in front of the concentrator. The WebdynSunPM is compatible with micro SDXC cards (15 x 11 mm) with a capacity of up to 32 GB.

The SD card is used to store the configuration, perform updates, or memorize the data read from the various equipment locally, thus making it possible to dispense with a remote server.

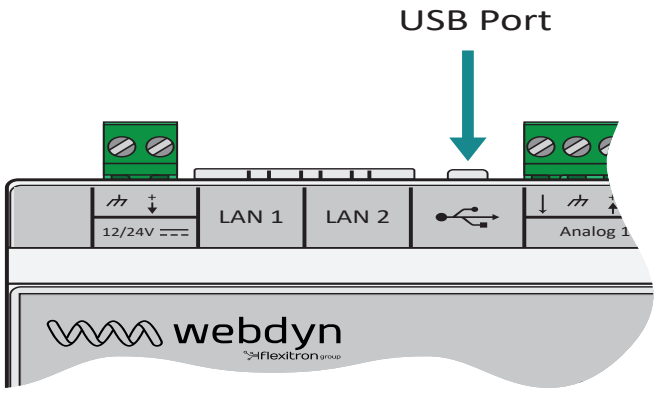


To insert the microSD card into the product, insert the microSD card into the slot on the front of the hub until you hear a click.

Webdyn does not provide any SD card. Please contact a computer hardware retailer.

2.4.4.2 USB Interface

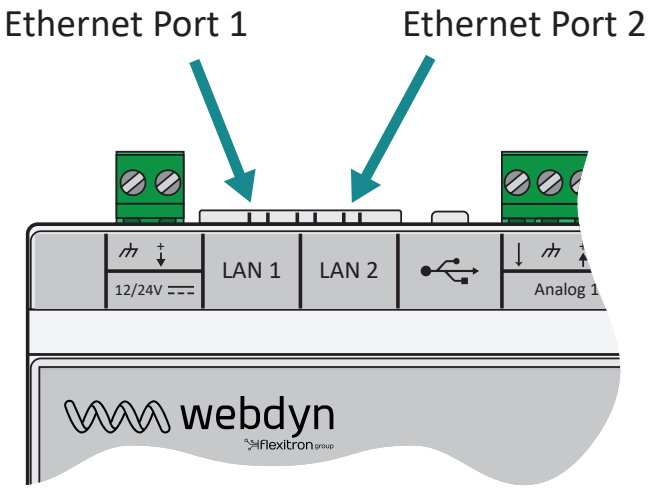
A USB port is available on top of the product next to the LAN connectors and analog inputs. The USB port is used to connect the TIC accessory to remotely read the information from electricity meters.



2.4.5 Ethernet Interface

The WebdynSunPM concentrator has 2 Ethernet interfaces (LAN1 and LAN2) which are separate from each other.

These Ethernet interfaces allow the concentrator to be part of 2 different Ethernet networks to communicate with local IP devices belonging to 2 separate networks or to communicate with the IS using Ethernet.



Ethernet port default parameters:

PARAMETERS	LAN1	LAN2
IP address	192.168.1.12	192.168.2.12
Subnet mask	255.255.255.0	255.255.255.0

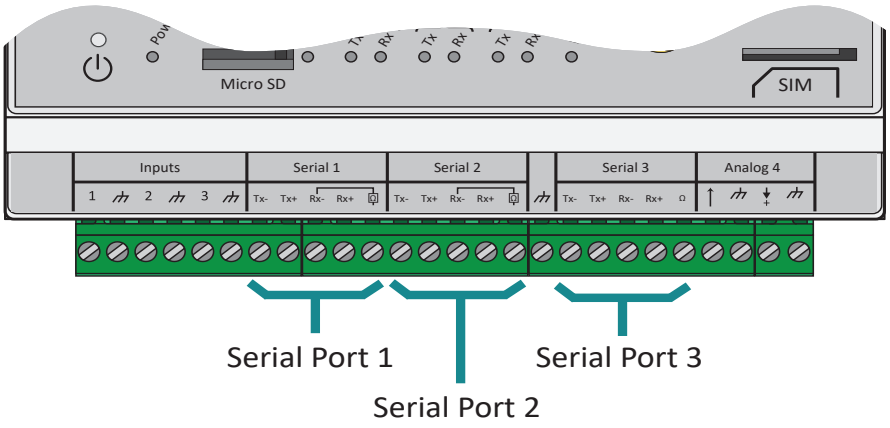
The Ethernet ports each support and include:

- A 10Base / 100Base-Tx IEEE 802.3 link.
- 2 indicators:
  - “Link” (green): Used to check that the physical link with another network device is available.
  - “Status” (orange): Used to view network traffic. It flashes depending on the traffic.
- Automatic signal detection and crossing.
- A speed (10/100 Mbps) and mode (half/full duplex) auto-negotiation.

If you want to connect several IP devices to the same network, the devices must have different IP addresses but belong to the same subnet. Never use the same IP address twice.

2.4.6 RS485/RS422 Serial Interface

The WebdynSunPM concentrator has 3 RS485/RS422 serial ports marked “Serial” on the bottom of the product which are only used for modbus in RTU mode. This interface is Half Duplex (2 wires) and Full Duplex (4 wires) compatible.



If several modbus RTU devices are connected, “serial” or “daisy chain” wiring is required. The cable arrives at a modbus module and exits towards the next one.

To guarantee proper data bus operation, an RS485 bus must feature a 120 Ohm terminator at each end. The WebdynSunPM concentrator can be at the end of the RS485 communication bus or in the middle. As the concentrator has a 120 Ohms resistor, it made need to be enabled depending on the concentrator position on the bus. (See wiring below)

There are 3 separate considerations for the choice of cable type:

- On installations requiring short lengths with no electric interference, plan on using a 2 pair 6/10 rigid screened cable.
- On larger installations of which the cable length is less than 500 m, plan on a 2 pair 8/10 rigid screened cable.
- When the cable distance is more than 500 m, and even more so if there is electric interference, plan for a shielded 2 pair 0.34 mm2 cable.

The maximum RS485 bus length is 1000 metres. (for 19200 bauds max). If the length is long, remember to reduce the device transmission speeds if communication is difficult.

Recommendations for RS485/RS422 BUS wiring:

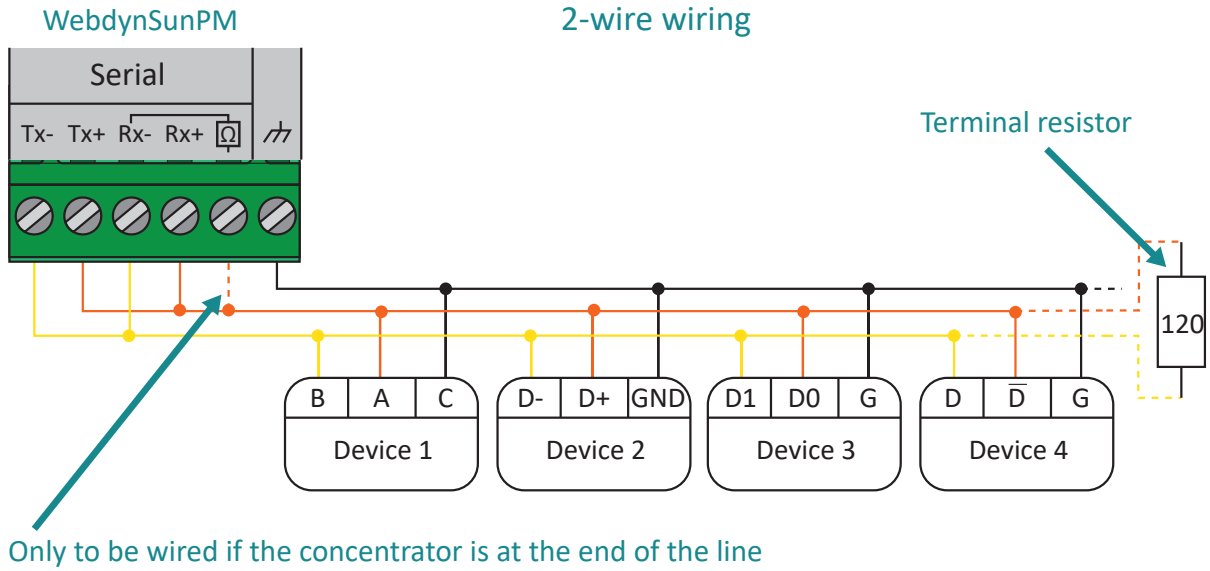
- The modules must be connected one after the other.
- Star connections are prohibited.
- The cables must either be screened or shielded, twisted pair per pair (see above: “cable type for RS485 bus connection”).
- The cable screen or shielding must be connected to the concentrator box earth and not to the 0 V (only connect one end of the screen).
- Avoid any return trips in the same cable.

Concentrator side RS485 wiring:

- Strip the RS485 communication cable sheath over about 4 cm.
- Shorten the shielding down to the cable sheath.
- Strip the wires over about 6 mm.
- Connect the conductors to the terminal block marked “Serial” following the assignments in your RS485 communication bus.

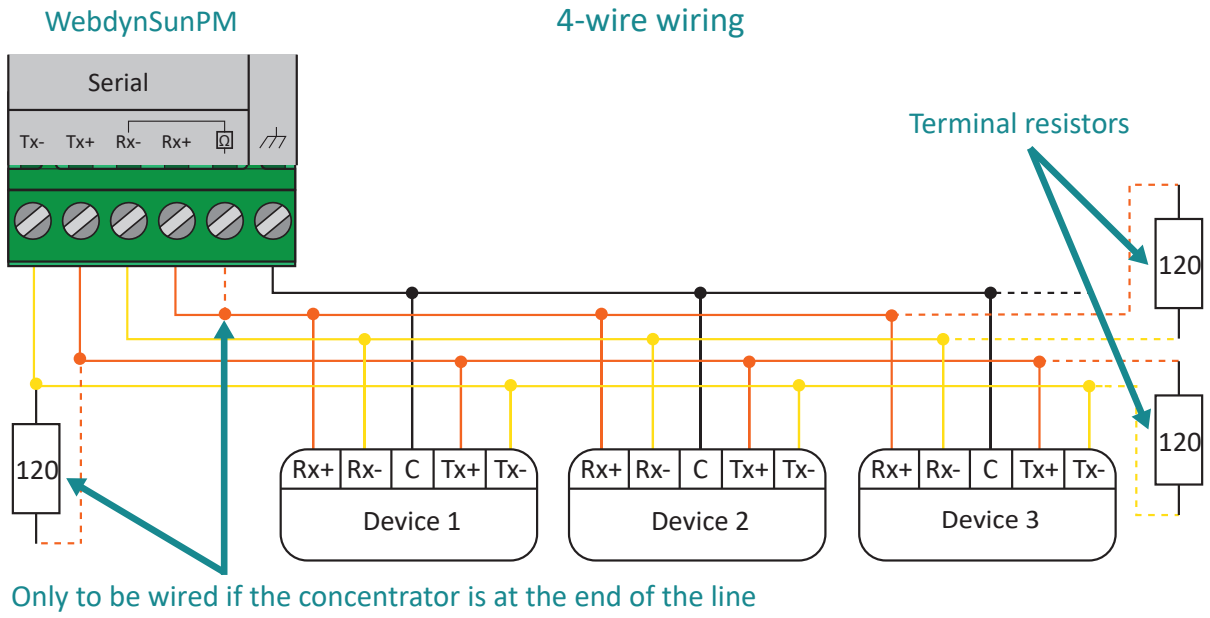
2-wire RS485 wiring (Half-Duplex):

This is the most common RS485 standard use. A single pair of wires is used for data transmission and reception. Several systems are linked in bus form as shown on the following figure. Different RS485 systems use different notations to indicate the correct connection form per differential communication pair. The following figure shows some of the notations used.



4-wire RS485/ RS485 wiring (Full-Duplex):

This type of connection uses two pairs of wires for communication. One pair of wires carries data sent from the concentrator to the device and the other pair the data sent from the device to the concentrator. Several systems can be connected to the bus as shown on the following figure.



The RS485 standard imposes a differential level of at least 200 mV to detect the signal level. To do that, the polarisation resistors must be at one end of the bus, usually at the master level. A simple method to check the correct polarisation consists in positioning the polarisation source at the start of the bus (master side) and to check the voltage level at the other end of the bus.

The common terminal (ground) must be interconnected with the corresponding terminals on each appliance to make sure the voltage between them is balanced. If the common conductor is not installed between all the devices, they must be properly grounded in compliance with the manufacturer recommendations for each network device. This requirement implies the use of an extra wire which, although not part of the communication process, is essential to guarantee the electric integrity of the network devices.

For more information on the RS4875/RS422 standards and device wiring, refer to the EIA-485 and EIA RS-422-A standards.

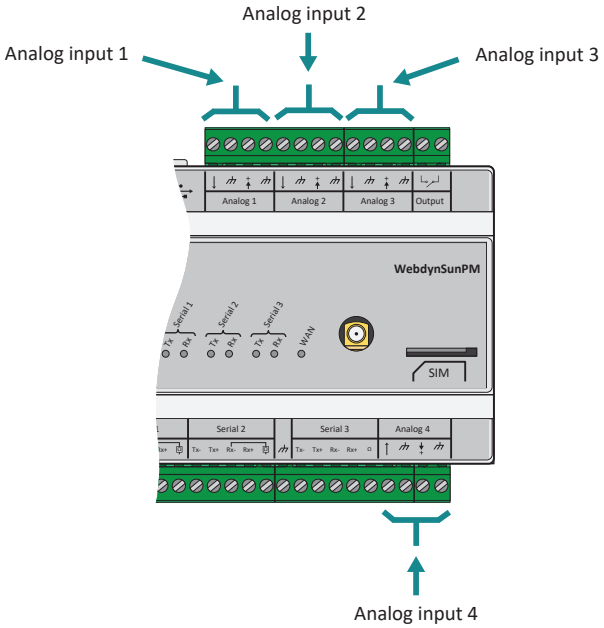
2.4.7 Input/Output Interface

The WebdynSunPM is fitted with:

- 4 analog inputs
- 3 digital inputs
- 1 relay output

2.4.7.1 Analog 0-10V or 4-20mA Inputs

The WebdynSunPM has 4 analog inputs marked “Analog” used to measure current of between 4 and 20 mA or a voltage of between 0 and 10 V. Each analog terminal block has a power output that can be used to power a sensor. The voltage delivered by this power output is equal to the concentrator’s power supply voltage. The earth on each analog terminal block is common.

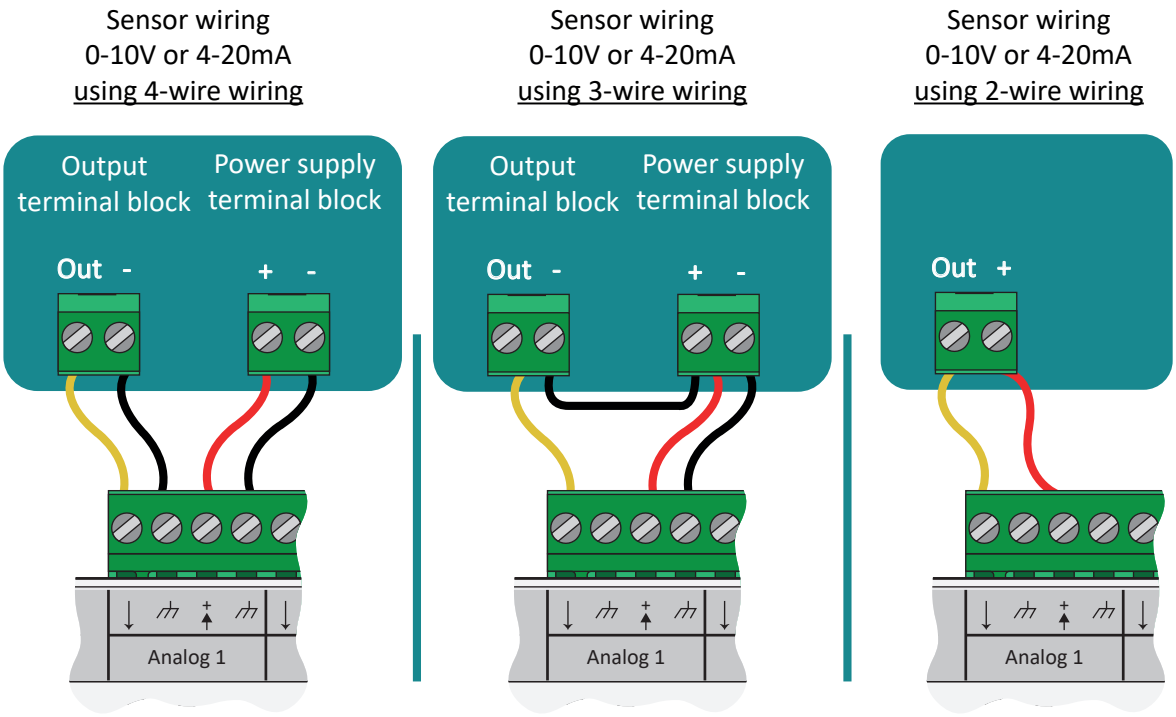


Each analog input can be software configured to 0-10V or 4-20mA.

The concentrators analog/digital converters (CAN) have 12 bit resolution making possible in:

- 0-10 V mode: to have .578 mV resolution.
- 4-20 mA mode: to have 5.578 µV resolution.

To connect, power off the concentrator and the 0-10V or 4-20mA sensor. Take into account the wiring information provided by the sensor manufacturer.



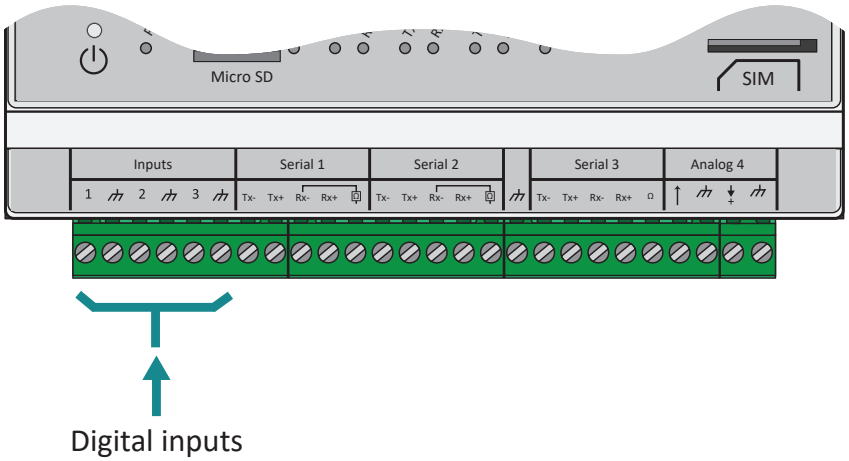
Do not apply a voltage higher than 12V or a current higher than 24mA to the analog inputs.

The voltage available for the sensors on the terminal block is the same as the power supply block voltage used to power the concentrator. The maximum power available for all the sensors must not exceed 7 watts.

2.4.7.2 Digital ON-OFF/S0 (pulsed) Inputs

The WebdynSunPM concentrator has 3 inputs that can be configured to ON-OFF mode or S0 pulsed mode (pulse counting).

These inputs are located at the bottom left of the WebdynSunPM concentrator.



The cable length for these inputs must not exceed 100m.

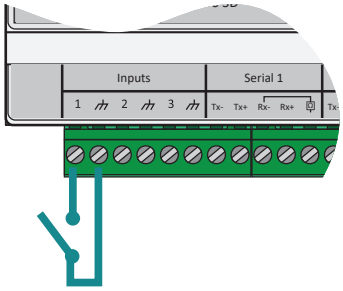
To prevent any damage to the concentrator, do not inject current or voltage onto the digital inputs.

In ON-OFF mode:

The concentrator can detect dry contact openings and closures to report device status or trigger status change alarms.

ON-OFF INPUTS	DIGITAL INPUTS
Type	Collector open / Drain open / Dry contact
Max voltage / current	4mA @5V
“0” Switching threshold disabled	> 3.5 V
“1” Switching threshold enabled	< 1 V
Pulse counters	> 20 ms

On-OFF Input wiring



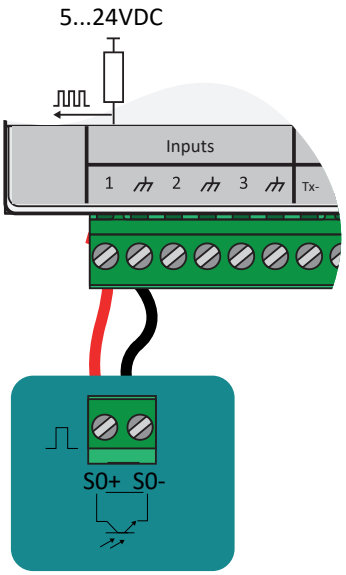
In S0 (pulse) mode:

The WebdynSunPM concentrator manages meters that have class A (24 V) and B (5 V) pulse outputs as per the IEC 62053-31-1998 standard.

The concentrator runs in “Sink” type mode, meaning that voltage is applied to e meter’s S0+ terminal using an (internal) pull-up resistor and a 0V voltage is applied to the meter’s S0- connection.

S0 PULSE INPUTS	CLASS A (24 V) Current pulses	CLASS B (5 V) Current pulses
“LOW” Switching threshold disabled	< 8 mA	< 1 mA
“HIGH” Switching threshold enabled	> 15 mA	> 2.5 mA
¨Power supply voltage	Internal 24 V	Internal 5 V
Pulse counters	> 20ms	> 20ms

S0 Input wiring

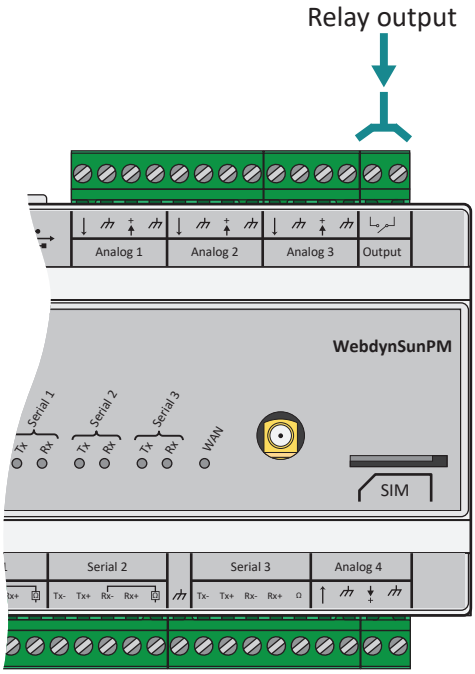


Meters with class A pulse outputs should be used for long distance transmissions. Meters with class B pulse outputs can only be used for short distances and make it possible to have reduced power consumption.

If the concentrator does not have a 24V power supply, there is a risk class A pulses will not work. The voltage for class A will be same as the power supply block's connected to the concentrator.

2.4.7.3 Relay Output

The WebdynSunPM concentrator has a potential-free relay output.



This output has the following specifications:

SPECIFICATIONS	MAX VALUES
Voltage	24 V
Current	1A

The concentrator relay does not allow high power to be driven directly. In this case, it is necessary to use an intermediate external relay.

The relay output can be controlled by:

- Command file (FTP, SFTP or WebDAV), MQTT/MQTTS message and SMS. (See chapter 5.3.9: ""setRelay": Changing the state of the relay").
- LUA scripts. (See chapter 3.2.3.1: "Scripts").

3. Configuration

The concentrator can be configured in several ways:

- Using configuration files uploaded to an FTP, SFTP or WebDAV-HTTPS server.
- Through configuration files placed on a micro SD card.
- Using the concentrator's web interface.
- Using text message commands.

The use of a server configured by FTP/SFTP/WebDAV-HTTPS is the preferred solution to remotely configure and maintain devices to service the equipment pool. If the local web interface and the FTP files are modified at the same time, the FTP files take precedence. Any modifications to files on the FTP server will overwrite any possible local modifications.

It should be noted that for the first start, the following elements must at least be configured using SMS (see chapter 5.2.3: "SMS") or the local web interface (see chapter 3.2.2.5: "Servers (servers))Servers (servers)"):

- FTP server name.
- FTP server authentication: login and password.
- Interface to use: modem or Ethernet properly configured to access external devices.

The text messages make it possible to issue some basic requests to launch the configuration file retrieval on the FTP server. They are never a substitute for full FTP or web configuration.

3.1 FTP/SFTP/WebDAV

As indicated previously, this configuration method takes precedence over local configuration using the web interface.

This operating mode requires an FTP, SFTP or WebDAV server accessible by the concentrator. This server is accessible using an address, a login and a password.

The server can be hosted on any server type (Windows, Linux, etc.). What is important is that its address is accessible by the concentrator using the configured network interface (Ethernet or modem). Please note that Windows FTP servers are unable to manage the case in file names. Therefore, to avoid all problems, it is recommended to only use Linux FTP servers.

The use of an SFTP or WebDAV server is preferable because of the extra security layers compared to a classic FTP server. Otherwise, different types of servers operate in the same way.

SFTP based on login and password.

3.1.1 Operating Principle

Configuration is based on a directory tree structure and files on an FTP, SFTP or WebDAV server.

By default, the concentrator identifier (uid) is calculated using the product MAC address, taking its last 6 characters prefixed by “WPM”. Thus, for a product with a MAC dress of “00:8D:00:00:BD:E4, the default uid will be “WPM00BDE4.

This uid is then used to prefix all the configuration files uploaded to the server. This is why it is essential for the uid to be unique for the entire installation.

The files are broken down as follows:

- Concentrator configuration files: these are files that contain data specific to the concentrator. By default they are stored in the “/CONFIG” directory on the server.
- Device definition files: the files used to configure/manage the different devices connected to the concentrator. By default they are stored in the “/DEF” directory on the server.

Please note that the concentrator does not create a configured tree structure. Therefore, all the configured directories need to be created manually. Thus, by default, the following directories at least should exist on the server:

- /CONFIG
- /ALARM
- /LOG
- /BIN
- /CERT
- /DATA
- /CMD
- /DEF
- /SCRIPT

These directories should have read-write rights for the configured user. (See section4.1 : “The FTP/SFTP /WebDAV server”) for more details on the required access rights.

2 separate servers can be configured. Both servers are used to upload data and backup the configuration.

Server 1 is called the main server, server 2 the secondary server.

If files are modified on the server, only the files on the main server are taken into account. Therefore, there are no risks of conflicts if the configuration files are modified on both servers.

Modifications made on the main server are propagated to the secondary server at the next connection.

3.1.2 Configuration Files

There are several types of configuration file.

There are files to configure concentrator operation (connection to the server, NTP management, passwords, modem, etc.) as well as connected device definition files.

This section describes all those files.

When the files are modified on the main server, the changes are carried over onto thee configured secondary server.

3.1.2.1. Concentrator Operation

The concentrator configuration files are located in the indicated configuration directory. By default "/CONFIG".

The files are the following:

- <uid>_config.ini: this file contains the following configuration elements:
  - Server tree structure configuration
  - Server type configuration: FTP, SFTP, WebDAV, MQTT, etc.
  - NTP configuration
  - Concentrator name and description
- <uid>_scl.ini: this file contains the configuration parameters for the scripts installed on the concentrator
- <uid>_var.ini: this file contains the concentrator connection scheduling configuration parameters
- <uid>_daq.csv: this file contains the concentrator connection and monitored device interfaces, namely:
  - Modem configuration:
    - PIN code
    - APN
    - Login/password/authentication type
  - Ethernet interface configuration:
    - IP address
    - Gateway
    - DNS
  - Serial port configuration:
    - Speed
    - Parity

        - Data bits
        - Protocol type used: Modbus etc.
    - Declaration of each connected equipment:
        - Index
        - Name
        - Interface
        - Address
        - Definition file
- <uid>_licence.ini: this file contains the licenses of the Lua Webdyn “.luaw” scripts

With <uid> the identifier of the concentrator.

3.1.2.1.1 “<uid>_config.ini” file

At least a certain number of parameters must be provided in this configuration file to provide communication with the concentrator.

First, define the configuration name: Take the device uid as explained previously and suffix “_config.ini” to it. Thus, in the previous example, the generated file name will be “WPM00BDE4_config.ini”.

If the base configuration is created using the embedded web server, this file is created automatically the first time the concentrator connects.

If the file is detected when connecting to the remote server, it is uploaded and the configuration is applied immediately regardless of the local configuration.

For an FTP or SFTP server:

- SERVER_Address=<Address or name of the FTP/SFTP server to use for the configuration>
- SERVER_TYPE=ftp or sftp
- If the connection is by Ethernet:
    - SERVER_Interface=ethernet
- If the connection is by modem:
    - SERVER_Interface=modem
- FTP_Login=<Login to use for the server>
- FTP_Password=<Password for the login>

For a WebDAV-HTTPS server:

- SERVER_Address=<WebDAV-HTTPS server address or name>
- SERVER_TYPE=webdav
- If the connection is made by ethernet:
  - SERVER_Interface=ethernet
- If the connection is made by modem:
  - SERVER_Interface=modem (by default)

Details of the configuration parameters for this file can be found in section 10.1 Appendix A: "_config.ini" configuration file.

For an MQTT server:

The WebdynSunPM can connect to an MQTT server in order to upload its data and alarms. It is also possible to issue commands to the hub through the MQTT server. To do this, you must indicate the topics subscribed to the concentrator in its settings.

The configuration of the concentrator cannot be done by the MQTT server, for this you will have to either go through an FTP server or use the embedded web interface.

The hub supports 5 different types of MQTT server, which are:

- MQTT: MQTT server without security.
- MQTTS: secure MQTT server.
- MQTTS aws: Amazon's "AWS IoT" server.
- MQTTS gcloud: Google's "Google Cloud IoT" server.
- MQTTS azure: Microsoft's "Azure IoT Hub" server.

Please contact the MQTT server manager to obtain the settings to be made as well as the certificates and secure keys to import to the concentrator.

Details of the configuration parameters for this file can be found in section 10.1 Appendix A: "_config.ini" configuration file.

For MQTTS servers, certificates must be imported into the concentrator. These certificates have a lifespan, it is your responsibility to renew them and import them before they expire.

3.1.2.1.2 "<uid>_var.ini" file

The "_var.ini" file contains the scheduling list configured on the concentrator.

The file name is defined as follows: Take the device uid as explained previously and suffix "_var.ini" to it. Thus, in the previous example, the generated file name will be "WPM00BDE4_var.ini".

If the base configuration is created using the embedded web server, this file is created automatically the first time the concentrator connects.

If the file is detected when connecting to the remote server, it is uploaded and the configuration is applied immediately regardless of the local configuration.

The file has one row per configured schedule. Each row contains the schedule number with its corresponding parameters.

The format is the following:

VARIABLE	DEFINITION	DEFAULT VALUE
SCHEDULE_Params[n]	Schedule parameters for the Server 1 connection	
SCHEDULE2_Params[n]	Schedule parameters for the Server 2 connection	

Where “n” is replaced by the schedule index number. The number starts at 0.

The parameters for each row are in the following format:

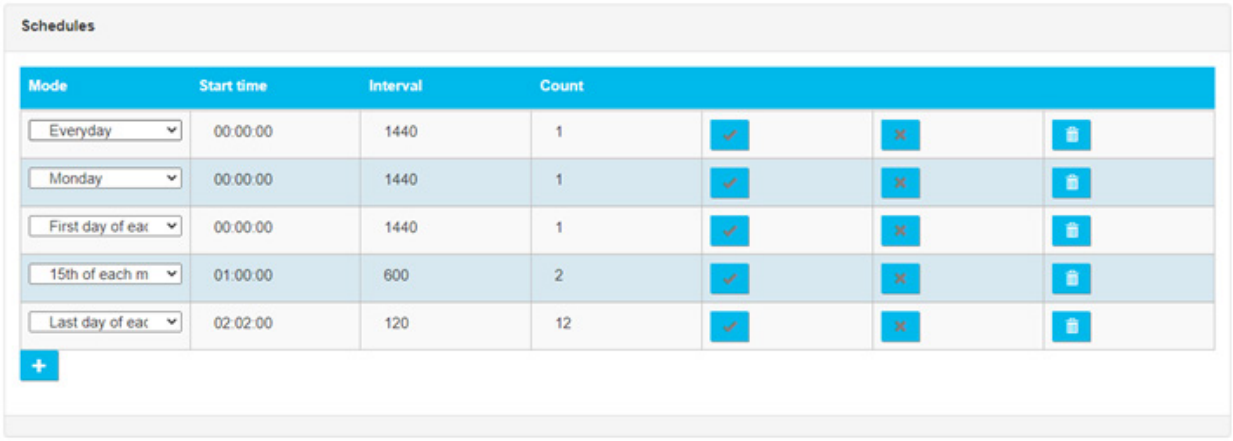
```
SCHEDULE_Params[index]=Id|Type|StartTime|Interval|Count
```

Each configured row therefore contains the following configuration information:

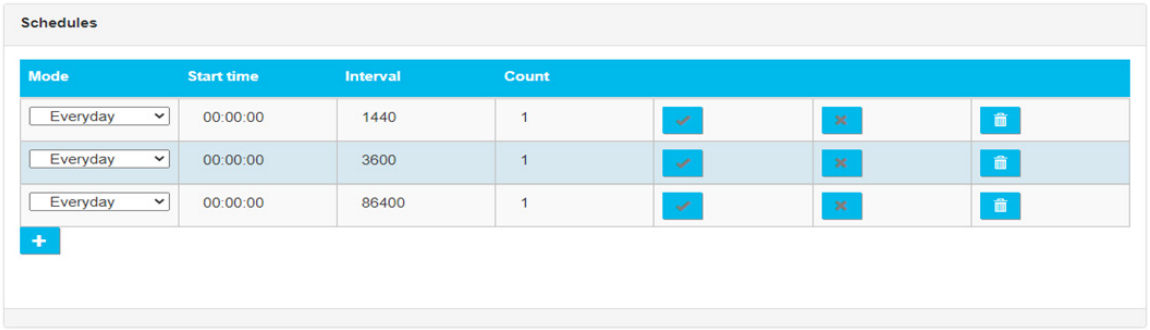
PARAMETER	DEFINITION	DEFAULT VALUE
Id	Configuration line identifier. Identifier must be unique	
Type	Indicates the schedule type. The authorised values are: • everyday: the schedule will be run every day • Monday: the schedule will be run every Monday • Tuesday: the schedule will be run every Tuesday • Wednesday: the schedule will be run every Wednesday • Thursday: the schedule will be run every Thursday • Friday: the schedule will be run every Friday • Saturday: the schedule will be run every Saturday • Sunday: the schedule will be run every Sunday • first: the schedule will be run on the 1st of the month • middle: the schedule will be run on the 15th of the month • last: the schedule will be run on the last day of the month	Everyday

StartTime	Indicates the task start time in the following format: “HH:MM:SS”	00:00:00
Interval	Indicates the connection repeat interval in minutes	1440
Count	Indicates the maximum number of connections in one day	1

Thus, with the following schedule on the Schedule 1 connection:



And the following schedule on the Schedule 2 connection:



The following configuration file is obtained:

```
SCHEDULE_Params[0]=1|everyday|00:00:00|1440|1
SCHEDULE_Params[1]=2|monday|00:00:00|1440|1
SCHEDULE_Params[2]=3|first|00:00:00|1440|1
SCHEDULE_Params[3]=4|middle|01:00:00|600|2
SCHEDULE_Params[4]=5|last|02:02:00|120|12
SCHEDULE2_Params[0]=1|everyday|00:00:00|1440|1
SCHEDULE2_Params[1]=2|everyday|00:00:00|3600|1
SCHEDULE2_Params[2]=3|everyday|00:00:00|86400|1
```

The index number is calculated automatically by the concentrator starting with 0. If this file is modified manually, make sure there are no duplicate index numbers as this would result in the configuration being rejected. Please note that the index number is separate for “SCHEDULE_Params” and “SCHEDULE2_Params”.

3.1.2.1.3 “<uid>_daq.csv” file

The “<uid>_daq.ini” file contains the list of devices configured on the concentrator and their interface configurations.

The file name is defined as follows: Take the device uid as explained previously and suffix “_daq.csv” to it. Thus, in the previous example, the generated file name will be “WPM00BDE4_daq.csv”.

If the base configuration is created using the embedded web server, this file is created automatically the first time the concentrator connects.

If the file is detected when connecting to the remote server, it is uploaded and the configuration is applied immediately regardless of the local configuration.

Contrary to the previous files, this one is in CSV format (delimiter “;”), i.e. directly editable using spreadsheet software such as Microsoft Excel®.

This file has 4 separate parts:

- Modem configuration
- Ethernet connection configuration
- Serial port configuration
- Connected device configuration

3.1.2.1.3.1 Modem Configuration

Modem configuration is based on the following parameters:

PARAMETER	DESCRIPTION	DEFAULT VALUE
Type	Device type. The only possible value here is “MODEM”	MODEM
Pin	Used to define the modem PIN code value, if defined.	
APN	This field contains the APN name to connect to using the SIM card. This APN depends on the selected operator and subscription. This field MUST NOT be empty to be able to use the modem connection.	
Login	Login to use to establish the connection. This login is provided by the operator and depends on it and the subscription type. This field can be empty.	
Password	Password for the login for authentication to use to establish the connection. This password is provided by the operator and depends on it and the subscription type. This field can be empty.	
Authentification	Authentication type to use for the connection. This value depends on the operator and subscription type. This information is supplied by the operator. The possible values are: • None: No authentication requested by the remote server • PAP: PAP type authentication requested by the remote server	None

Below is a modem configuration example:

```
type;pin;apn;login;password;authentification
MODEM;;m2minternet;;;None
```

In this example, the PIN code is empty, the APN is set to “m2minternet”, the login and password are empty and the authentication type is defined as “None”.

When editing this file using Excel, the following display is shown using the CSV format:

type	pin	apn	login	password	authentification
MODEM		m2minternet			none

If the file is modified using “Excel” type spreadsheet software, the format may be modified and the “;” delimiters replaced by “,”, making it unusable by the concentrator. Always make sure to indicate the delimiter format when saving.

3.1.2.1.3.2 Ethernet Connection Configuration

Ethernet interface configuration is based on the following parameters:

PARAMETER	DESCRIPTION	DEFAULT VALUE
Type	Device type. The possible values are: • LAN1: to configure the LAN 1 interface. • LAN2: to configure the LAN 2 interface.	LAN1 for the first line LAN2 for the second line
Ip	The local IP address assigned to the Ethernet interface. This value has an effect on the local IP address at which the box can be contacted on the relevant Ethernet interface. This field cannot be empty.	192.168.1.12 for the first line 192.168.2.12 for the second line
Mask	This field contains the subnet mask used jointly with the configured IP address. This field cannot be empty.	255.255.255.0
Gateway	Routing device configuration to use for the concentrator to be able to communicate with devices not present on its local network.	
	IP address for a DNS server the concentrator is to use to resolve names.	
DNS1	IP address for a DNS server the concentrator is to use to resolve names if “DNS1” fails to respond.	

Below is an Ethernet interface configuration example:

```
type;ip;mask;gateway;dns1;dns2
LAN1;192.93.121.37;255.255.255.0;192.93.121.1;192.93.121.8;
LAN2;192.168.2.12;255.255.255.0;;;
```

In this example, the first network interface (LAN 1) is configured at IP address “192.93.121.37”, with a subnet mask of “255.255.255.0” allowing it to access all machines connected using the “193.93.121.xxx” address. This interface also uses a router at the “192.93.121.1” address to communicate with external devices and a DNS server accessible at the “193.93.121.8” address. DNS2 is not configured.

Similarly, a second network interface is left with its default configuration, namely an IP address on “192.1682.12” and a subnet mask at “255.255.255.0”. All the other parameters are empty.

Ethernet interframe configuration:

When using these network interfaces to collect data on equipment to be monitored, it may be necessary to space out the sending of requests so as not to saturate the equipment queried.

To do this, configure the ethernet interframe parameter as follows:

```
tcpInterFrameMs;<value>
```

PARAMETER	DESCRIPTION	DEFAULT VALUE
Value	Waiting time between 2 frames in modbus TCP. This time is expressed in ms	0

The operating principle is the same as for the serial ports. Each time a frame is sent in modbus TCP, the concentrator will leave a silence corresponding to “tcpInterFrameMs” between the device response and the next query to network devices.

The interframe parameter is global, meaning it applies to all modbusTCP communications and can therefore result in the slowing down of data reading if it is too high.

This parameter is to be used for certain specific modbusTCP devices. See “3.1.2.1.3.3 - Serial port configuration“ for more details on how the inter frame time operates.

When editing this file using Excel, the following display is shown using the CSV format:

type	ip	mask	gateway	dns1	dns2
LAN1	192.93.121.37	255.255.255.0	192.93.121.1	192.93.121.8	0
LAN2	192.168.2.12	255.255.255.0			
tcpInter FrameMs	0				

If the file is modified using “Excel” type spreadsheet software, the format may be modified and the “;” delimiters replaced by “,”, making it unusable by the concentrator. Always make sure to indicate the delimiter format when saving.

3.1.2.1.3.3 Serial Port Configuration

Serial port configuration is based on the following parameters:

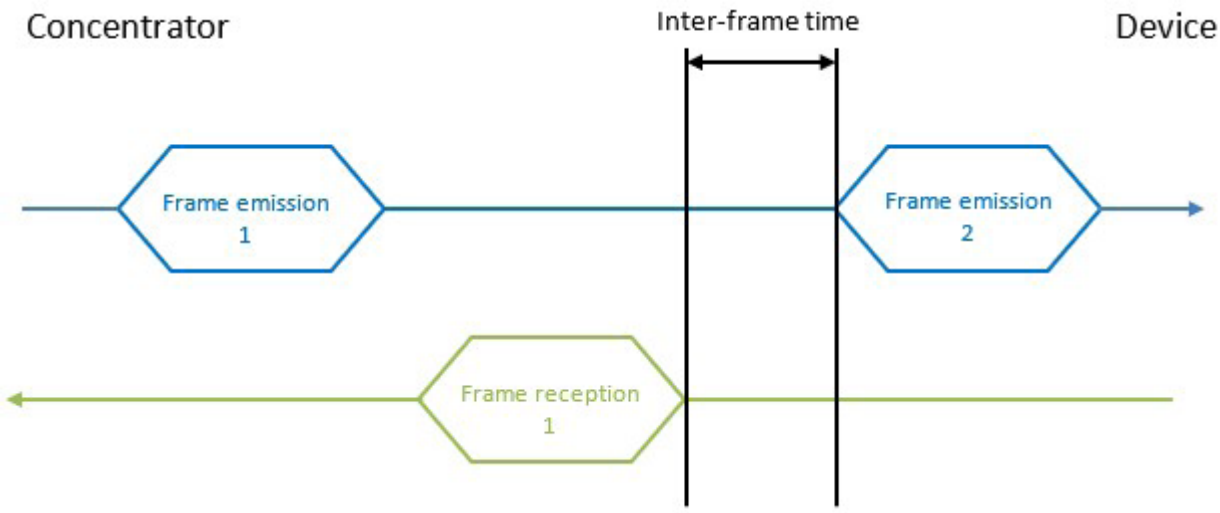
PARAMETER	DESCRIPTION	DEFAULT VALUE
Type	Device type. The possible values are: • SERIAL1: to configure the “Serial 1” interface • SERIAL2: to configure the “Serial 2” interface • SERIAL3: to configure the “Serial 3” interface	SERIAL1 for the 1st line SERIAL2 for the 2nd line SERIAL3 for the 3rd line
Baudrate	Speed in bauds to use for the serial connection. The possible values are: • 1200 • 2400 • 4800 • 9600 • 19200 • 38400 • 57600 • 115200 • 230400 • 460800	19200

DataBits	Number of data bits per byte. The possible values are: • 7 • 8	8
Parity	The parity type to apply to validate data on the serial link. The possible values are: • N: no parity • O: odd parity • E: even parity	N
StopBits	Number of stop bits between 2 bytes. The possible values are: • 1 • 2	1
Wires	The number of wires to use on the serial interface. The possible values are: • 2: the same 2 wires are used to send and receive data frames • 4: data emission and reception use separate pairs of wires	2
Protocol	The protocol type for this serial interface. The possible values are: • ModbusRTU: the serial port is reserved for Modbus RTU communications • proprietary protocol (see specific appendix on proprietary protocols)	ModbusRTU
InterFrame	The waiting time between 2 frames exchanges on the serial port. This time is expressed in ms. See below for a detailed explanation of how this parameter operates.	0

Forwarded TCP port	Forwarded TCP port. If there is a value in this field, the concentrator opens a modbusTCP port on the entered port number. When modbusTCP devices connect to this port, all sent requests are directly forwarded to the ModbusRTU bus and the response is returned to the connected device using this modbusTCP port. This option is used to create a communication tunnel between modbusTCP devices and the local ModbusRTU network. The requests are slotted between the concentrator's internal monitoring requests.

The "InterFrame" parameter is used to define a silence time on the serial bus to allow certain devices to switch to waiting for data. Some manufacturers call this the "return time".

The operating principle is the following:



When the concentrator receives a response from the device, it will impose a delay equivalent to the "InterFrame" parameter between the last byte of the last received frame and the first byte of the next frame it sends to that device. The time is valid for the entire bus. So, if a new frame is emitted to another device than the previous one, the delay will nevertheless be applied. Below is a serial interface configuration example:

```
type;baudrate;data_bits;parity;stop_
bits;wires;protocol;interframe(ms)
SERIAL1;9600;8;N;1;2;Modbus;0
SERIAL2;1200;8;N;1;4;SMANET;0
SERIAL3;19200;8;N;1;2;PW1;0
```

In this example, the first serial port is configured at 9600 bauds, 8 data bits, no, parity, 1 stop bit, 2 communication wires, “modbus” protocol and no inter-frame time.

The 2nd serial port is configured at 1200 bauds, 4 wires to be used with the SMA-Net protocol.

The 3rd serial port is configured at 19200 bauds, 2 wires to be used with the PowerOne protocol.

When editing this file using Excel, the following display is shown using the CSV format:

type	br	data_b	parity	stop_bits	wires	protocol	interframe
SERIAL1	9600	8	N	1	2	Modbus	0 ms
SERIAL2	1200	8	N	1	4	SMANET	0 ms
SERIAL3	19200	8	N	1	2	PW1	0

If the file is modified using “Excel” type spreadsheet software, the format may be modified and the “;” delimiters replaced by “,”, making it unusable by the concentrator. Always make sure to indicate the delimiter format when saving.

3.1.2.1.3.4 Connected Device Declaration

Connected device interface configuration is based on the following parameters:

PARAMETER	DESCRIPTION	DEFAULT VALUE
Index	Index for the device to be defined. This field contains a numeric value representing the identification number for the device to be configured. Each number must be unique, failing which the device will not be visible in the web interface. The following index values are also accepted: • IO: is used to indicate that the definition file is for the concentrator IO. • TIC1: is used to indicate that the definition file is for a TIC type device.	
Interface	For “IO” or “TIC” devices, this field is not used. It must be left empty. For modbus devices, it contains the following information: • SERIAL1: the device is connected to serial port 1. • SERIAL2: the device is connected to serial port 2. • SERIAL3: the device is connected to serial port 3. • “IP address” the device is of the modbus TCP type and is accessible at the indicated IP address. • “IP address-Port:” the device is of the modbus TCP type and is accessible at the indicated IP address at the indicated TCP port number.	
Name	This field describes the name given to the device in the web interface. This name is also used for MQTT and the scripts. This is why it is essential that it be unique in the configuration.	
Address	This field is not used for “IO” and “TIC” type devices. For “modbus” devices, it contains the device address on the modbus network. Its value is therefore from 1 to 240. For “TIC”, “IO” and “SMA-Net” devices, this field is not used.	
AcqPeriod	This field is used to specify the period for recording data in the file (equipment collection is continuous). It is expressed in seconds.	600

Timeout	Device response timeout configuration. If the device does not respond within this time limit, the concentrator considers the request to have failed. This time is expressed in ms. This field is not used by the “IO”, “TIC” and “SMA-Net” protocols. For SMA-Net, the value is 3000ms.	1000
SerialNumber	Some devices require a serial number. This is especially the case for “TIC” devices. This field is used to indicate it. It is not used by the “IO” and “Modbus” protocols	
Parameters	This field is only used in the following cases: • ModbusTCP devices: used to indicate if “KeepAlive” messages are sent to keep the connection open. If the field value is “1”, a regular message is sent. If the field value is “0”, no message is sent. The default value is 1 • Proprietary equipment: (see specific appendix on proprietary protocols).	
Category	This field allows you to specify the category to which the equipment belongs. It should be noted that this field is filled in automatically by the concentrator.	
Model	This field indicates the device model name. Note that the concentrator fills in this field automatically.	
DefFile	This field is used to indicate the definition file that exactly describes all the variables and data exposed by the device. See the specific definition file section (Connected Device Definition).	

Below is a connected device interface configuration example:

```
index;interface;name;address;acqPeriod(s);timeout(ms);serialNumber;
parameters;category;model;defFile
IO;;Io;;36000;;;;WebdynSunPM;ioSunPM;WPM00C715_IO.csv
0;SERIAL1;serial1_52224;0;600;0;2000388220;1;Inverter;WR21TL09;
WPM00C715_SMA_Inverter_SMA_WR21TL09.csv
1;192.93.121.23:502;ethernet_192.93.121.23_502_126;126;600;5000;;1;
Inverter;Solar_Inverter;WPM00BDE4_SunSpec_inverter_SMA_Solar_Inverter
_9301_modbusTCP.csv
```

There are 3 devices in this example:

- An “IO” type device corresponding to the concentrator inputs/outputs
- A device with a proprietary protocol on serial bus
- A modbus SMA device on Ethernet

Explanation of the different devices:

- IO:
  - Index: IO. The IO value indicates that the device is of the “IO” type.
  - Interface: empty. As the inputs/outputs are built into the box, this field is ignored.
  - Name : Io. The name that will be displayed on the local web pages.
  - Address: empty. This information does not concern inputs/outputs.
  - AcqPeriod: The acquisition period is set to 36000 seconds, or one record every 10 hours.
  - Timeout: empty. This information does not concern inputs/outputs.
  - SerialNumber: empty. This information does not concern inputs/outputs.
  - Parameters: empty. This information does not concern inputs/outputs.
  - Category: WebdynSunPM.
  - Model: ioSunPM.
  - DefFIle: WPM00C715_IO.csv. The definition file name that describes the different configured inputs/outputs.

- INV proprietary protocol (see proprietary protocols specific appendix):
- modbus ethernet:
  - Index: 1. Index for the second modbus device configured on the concentrator.
  - Interface: 192.93.121.23:502. This value indicates that the device is of the Ethernet type at IP address 192.93.121.23 using the default port number, i.e. 502.
  - Name: ethernet_192.93.121.23_502_126. The name that will be displayed on the local web pages.
  - Address: 126. This device responds at the modbus 126 address.
  - AcqPeriod: 600. The acquisition period is set to 600 seconds, or one record every 10 minutes.
  - Timeout: 5000. The timeout is set to 5000ms.
  - SerialNumber: empty. Not applicable.
  - Parameters: 1. The connection is kept alive.
  - Category: Inverter.
  - Model: Solar_Inverter.
  - DefFIle: WPM00C715_SunSpec_inverter_SMA_Solar_Inverter_9301_modbusTCP.csv. The definition file name that describes the different variables configured for this device detected using SunSpec.

When editing this file using Excel, the following display is shown using the CSV format:

index	interface	name	address	acqPeriod	timeout	serial number	param.	category	model	defFile
IO		Io		36000 s				Webdyn SunPM	ioSunPM	WPM00C715_IO.csv
0	SERIAL1	Serial1_52224	0	600 s	0 ms	2000388220	1	Inverter	WR21TL09	WPM00C715_SMA_Inverter_SMA_WR21TL09.csv
1	192.93.121.23:502	Ethernet_192.93.121.23_502_126	126	600 s	5000 ms		1	Inverter	Solar_Inverter	WPM00C715_SunSpec_Inverter_SMA_Solar_Inverter_9301_modbusTCP.csv

If the file is modified using “Excel” type spreadsheet software, the format may be modified and the “;” delimiters replaced by “,”, making it unusable by the concentrator. Always make sure to indicate the delimiter format when saving.

3.1.2.1.4 “<uid>_sci.ini” file

The “<uid>_sci.ini” file contains the list of scripts configured on the concentrator.

If the base configuration is created using the embedded web server, this file is created automatically the first time the concentrator connects.

If the file is detected when connecting to the remote server, it is uploaded and the configuration is applied immediately regardless of the local configuration.

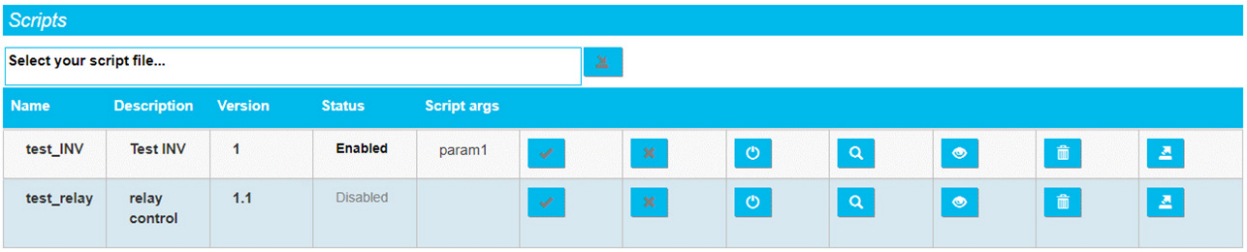
The file has three lines per configured script. Each line contains the script number.

The format is the following:

VARIABLE	DEFINITION	DEFAULT VALUE
SCRIPT_File[n]	Scenario name	Script file name with extension
SCRIPT_Enable[n]	Enabling the script	0
SCRIPT_Args[n]	Script parameter	

Where “n” is replaced by the script index number. The number starts at 0.

Thus, using the following script configuration:



The configuration file will contain the following information:

```
SCRIPT_File[0]=test_INV.lua
SCRIPT_Enable[0]=1
SCRIPT_Args[0]=param1
SCRIPT_File[1]=test_relay.lua
SCRIPT_Enable[1]=0
SCRIPT_Args[1]=
```

The index number is calculated automatically by the concentrator starting with 0. If this file is modified manually, make sure there are no duplicate index numbers as this would result in the configuration being rejected.

3.1.2.1.5 “<uid>_licence.ini” file

The “<uid>_licence.ini” file contains the Lua Webdyn “.luaw” script licenses. The license file must be placed in the CONFIG directory of the remote server.

The license file makes it possible to obtain specific scripts designed by Webdyn. In this case, please contact the Webdyn sales department who will be able to advise you and redirect you to the relevant contacts: contact@webdyn.com

When connecting to the remote server, if the file is detected, it is downloaded and the license is immediately applied. The WebdynSunPM will not upload the license file if it is deleted.

The license file is specific to a WebdynSunPM. It is not possible to use the same file on several concentrators. It is forbidden to modify the contents of the license file, under penalty of blocking the management of the concentrator licenses.

3.1.2.2 Connected Device Definition

The definition file is standardised to process the different device cases. It manages the following types: IO, TIC, Modbus RTU, Modbus TCP, etc.

A definition file is needed fore each configured device. A same definition file can be used to define several devices.

The device definition files are stored in the configured server directory. By default it is “/DEF”.

For the definition files to be taken into account, they must be referenced in the “_daq.ini” file described previously (3.1.2.1.3.4 - Declaration of devices) in the “DefFile” field.

3.1.2.2.1 Definition File Naming

The file name is free and can be modified by the client at will, the gateway will use the name given in the daq.csv file.

The automatically generated files are prefixed with the concentrator ID of which the default value is “WPM” followed by the last 6 digits of the MAC address of the device that generated them.

The concentrator ID can be modified by the client, however, if it is modified after a file has been generated, there will be no resulting modification of the file name or its declaration in the daq.ini file.

The prefix is separated from the remainder of the file name by an underscore. The purpose of this is to prevent overwriting files that previously existed at the IS level.

The files will have a csv extension. If another extension (.ini for example) is given in the daq file, it will be accepted.

Files generated automatically or created by the local web interface will be named as follows:

3.1.2.2.1.1 IO

The IO file names are defined as follows:

<uid>_IO.csv

3.1.2.2.1.2 Modbus

There are two types of modbus file:

- The files generated by the installer/integrator.
- The files generated by SunSpec auto-detection.

For the files generated by the installer/integrator, there are no file naming rules. The file name is free form. There are no rules.

For the files generated by SunSpec detection, the name is composed as follows:

<uid>_SunSpec_<Category>_<Manufacturer>_<Model>_<Options>_<Protocol>.csv:

- Category: the device type. Currently only the “Inverter” type is used.
- Manufacturer: the manufacturer name. The value is obtained from the ”Manufacturer” field in the SunSpec tables.
- Model: model name. The value is obtained from the ”Model” field in the SunSpec tables.
- Options: options detected on the device. The value is obtained from the ”Options” field in the SunSpec tables.
- Protocol: the protocol name. The values are:
  - ModbusRTU: for a device on the serial connection.
  - modbusTCP: for modbus devices on Ethernet.

3.1.2.2.1.3 Proprietary Protocol

Proprietary protocol file names are detailed in the proprietary protocols application note.

3.1.2.2.2 Definition File Content

The file is in csv format, it is composed of text rows each composed of “;” delimited fields.

The first row in the file contains the following information:

```
Protocol ; Category ; Manufacturer ; Model ; Forced written code
```

The fields are configured as follows:

FIELD	DESCRIPTION
Protocol	Protocol name used for this device. The possible values are: • ModbusRTU: the device is accessed using the modbus RTU serial connection • modbusTCP: the device is accessed using TCP connection • io: the device is of the IO type • tic: the device is of the TIC type
Category	Device category This name will be displayed as is on the local web site
Manufacturer	Manufacturer name. This name will be displayed as is on the local web site
Model	Device model name. This name will be displayed as is on the local web site
Forced written code	Indicates whether the Modbus function code used for writing should be forced to 0x10. The possible values are: • 0 (default value): Writing a single register is done with function code 0x06 while writing multiple registers is done with function code 0x10. This is the classic behavior for a Modbus device. • 1: writing is done only with function code 0x10, even in the case of a simple register. This behavior is required for some inverters. For example some GoodWe brand inverters.

The Category, Manufacturer and Model are used to find the device's associated definition file from the web pages.

Following this first row, all the following rows will contain the device variable definitions.

Each row fully describes a variable.

Each row will have the following format:

```
Index ; Info1 ; Info2 ; Info3 ; Info4 ; Name ; Tag ; CoefA ; CoefB ;
Unit ; Action
```

The field meanings are the following:

FIELD	DESCRIPTION
Index	Contains the unique variable identifier in the file. It is free form for the client as long as it remains unique. This field is used to identify the variables in the data file, the logs or the command files.

Info1	This field contains information specific to the protocol used on the device. Refer to the specific protocol documentation below.
Info2	This field contains information specific to the protocol used on the device. Refer to the specific protocol documentation below.
Info3	Variable format. The allowed formats are: •U8: unsigned integer on 8 bits (1 byte) •U16: 16-bit unsigned integer (2 bytes, or 1 register) •U32: 32-bit unsigned integer (4 bytes, or 2 registers) •U64: 64-bit unsigned integer (8 bytes, or 4 registers) •I8: 8-bit signed integer (1 byte) •I16: 16-bit signed integer (2 bytes, or 1 register) •I32: 32-bit signed integer (4 bytes, or 2 registers) •I64: 64-bit signed integer (8 bytes, or 4 registers) •F32: floating on 32 bits (4 bytes, or 2 registers) •F64: floating on 64 bits (8 bytes, or 4 registers) •String: the variable is a character string. It is then necessary to use the notation “Address_Size” for the field “Info2” •Bits: the variable is of the bit field type. You must then use the notation “Address_1st bit_Number of bits” for the “Info2” field •IP: the variable is of the IP V4 address type and is therefore coded on 4 bytes (2 registers) •IPV6: the variable is of the IP V6 address type and is therefore coded on 16 bytes (8 registers) •MAC: the variable is of the MAC address type, in “EUI48” format. It is therefore coded on 6 bytes (3 registers) It should be noted that it is possible to modify the whole types by adding a suffix. The allowed modifiers are: •_W: the words are exchanged, i.e. the content of the variable registers is exchanged in blocks of 2 bytes •_B: the bytes are exchanged, i.e. the content of the variable registers is exchanged at byte level, one by one •_WB: the words AND the bytes are exchanged. The 2 modifiers above are applied. So, for example, the notation “I32_W” indicates that it is a variable so bytes 1 and 2 will be exchanged with bytes 3 and 4. Similarly, the notation “U16_B” indicates that bytes 1 and 2 of the variable will be exchanged. This corresponds to a “Little Endian/Big Endian” conversion.
Info4	This field contains information specific to the protocol used on the device. Refer to the specific protocol documentation below.
Name	Contains the variable name, which is free form as long as it is unique.

Tag	Contains an identification making it possible to use the variable in scripts. (Calculation of totals, issuing of commands to multiple devices, etc.). This name must be unique to allow unambiguous identification and use in the scripts.
CoefA	Contains the multiplier to apply to the variable so that it complies with the unit described in the “unit” field. This multiplier is a floating point number using a decimal point “. “
CoefB	Contains the offset to apply to the variable so that it complies with the unit described in the “unit” field. This offset is a floating point number using a decimal point “. “ Factors A and B will contain the appropriate value by default if it is known, otherwise A is 1 and B is 0. If the values are missing, these values will be considered to be 1 and 0. The factors are there to inform users and are not applied on the data sent in the data files. This operation is the responsibility of operators who, if they want to convert the sent raw data, must carry out the “Ax+B” operation on data x, in particular to obtain a value in the unit indicated in the “unit” field These factors are applied to the values used by the scripts and by the return mechanism to a display to obtain a value indicated in the “unit” field.
Unit	Contains the required unit. As for the factors, this field contains the appropriate information if it is known, otherwise the field is empty. This field is information for the user and it is up to the client to make sure it matches the entered A and B factors.

Action	Contains the code describing the processing to be carried out by the product on this variable when the files for the IS are created. The possible actions are: • 0: variable disabled. The variable will not appear in the data files • 1: the variable is of the parameter type and is read only. It will not therefore appear in the data files. • 2: the variable is of the min/max/mean type. In that case 3 files fields will contain this variable in the data file to log the minimum, maximum and mean value for this data. • 4: the variable is of the instant value type. The data read at collection time will be stored in the data file using a single field. • 6: the variable is equivalent to a variable defined with code 4 with regard to acquisition. But the variable will also be targeted by an instantaneous collection performed via the getData command. • 7: the variable is equivalent to a variable defined with code 2 with regard to acquisition. But the variable will also be targeted by an instantaneous collection performed via the getData command. • 8: the variable is of the alarm type. When a change in the value of this variable is detected, an alarm is triggered. The data read at collection time will be stored in the alarm file, using a single field. A connection is caused immediately in order to transmit said file. • 9: the variable is equivalent to a variable defined with code 8, so it is an alarm. But the alarm file containing the new value will only be deposited during the next connection to the server (scheduled or manual). There is no automatic connection in the case of this action code.

The “Info1”, “Info2”, “Info3” and “Info4” fields are specific to each protocol and are therefore configured as follows:

3.1.2.2.2.1 IO

FIELD	DESCRIPTION
Info1	Input/output type. The authorised values are: • 1:analog input (0-10V or 4-20mA) • 2: digital input • 3: switching relay output
Info2	Input/output number. The authorised values are: • 1 to 3: if “Info1” equal to 1 (analog input) • 1 to 4: if “Info1” equal to 2 (digital input) • 1: if “Info1” equal to 3 (switching relay output)

Info3	Clarification on the input or output type. • If “Info1” equals 1 (analog input), the authorised values are: 1: analog input of the 4- 20ma type 2: analog input of the 0-10V type • If “Info1” equals 2 (digital input), the authorised values are: 1: ON-OFF input 2: S0 pulse input • If “Info3” equals 3 (relay type), this field is not used
Info4	Not used

3.1.2.2.2.2 Modbus

FIELD	DESCRIPTION
Info1	The type of register read. This type results in the function codes that will be used to read and write the data. The authorised values are: • 1: “coil . The modbus read function code will be 0x01. The write function code will be0x05 • 2: “discrete inputs“. The modbus read function code will be 0x03. This type is for input reading. Writing is therefore not possible. • 3: “holding register”. The modbus read function code will be 0x03. The write function code will be 0x10 • 4: “input”. The modbus read function code will be 0x04. As this type is for input reading, there is no associated write function.

Info2	Address and size of the register or input to read. The possible forms are the following: • Register address. This is the most common case. Here, we find the register address in its classic format. Example: “40000” causes the register to be read at address 40000. • Register address and size. This format is used to indicate the size of the data to be read, expressed in bytes. This format is used to read character strings for example. The format is the following: Register address_Size. Thus, for example, the value “40000 10” configures a variable of which the data is at register 40000 and is 10 bytes long. If the type is “U8” or “I8”, the “Size” information corresponds to the offset to be applied to the register to obtain the information. Thus, for example, for an 8-bit integer (1 byte), the value “40000_1” implies that we want to read the 2nd byte in modbus register 40000. • Register address, 1st bit and number of bits. This format is used to indicate the register at which a bit field starts and the bit field size. Note that if the number of bits is 1, there is no need to indicate the number of bits. The format is the following: Register address_1st bit_Number of bits. Thus, for example, the value “40005_4_8” configures a variable of which the data is at register 40005 at its 4th bit and is 8 bits long. Similarly, the value “40008_1” configures a variable of which the data is at register 40008 at its 1st bit and of which the value is only 1 bit long considering that the number of bits is not indicated.

Info3	Variable type. The authorised types are the following: • U8: unsigned integer on 8 bits (1 byte) • U16: unsigned integer on 16 bits (2 bytes, or 1 register) • U32: unsigned integer on 32 bits (4 bytes, or 2 registers) • U64: unsigned integer on 64 bits (8 bytes, or 4 registers) • I8: signed integer on 8 bits (1 byte) • I16: signed integer on 16 bits (2 bytes, or 1 register) • I32: signed integer on 32 bits (4 bytes, or 2 registers) • I64: signed integer on 64 bits (8 bytes, or 4 registers) • F32: floating on 32 bits (4 bytes, or 2 registers) • F64: floating on 64 bits (8 bytes, or 4 registers) • String: the variable is a character string. In that case the “Address_Size” notation should be used for the “Info2” field • Bits: the variable is of the bit field type. In that case the “Address_1st bit_ Number of bits” notation should be used for the “Info2” field • IP: the variable is of the IP V4 address type and is therefore coded on 4 bytes (2 registers) • IPV6: the variable is of the IP V6 address type and is therefore coded on 16 bytes (8 registers) • MAC: the variable is of the MAC address type in “EUI48” format. It is therefore coded on 6 bytes (3 registers) Note that the integer types can be modified by adding a suffix. The authorised modifiers are: • _W: the words are exchanged, i.e. the variable register content is exchanged in 2 byte blocks • _B: the bytes are exchanged, i.e. the variable register content is exchanged in at byte level, one by one • _WB: the words AND the bytes are exchanged. The 2 previous modifiers are applied. Thus, for example, the “I32_W” notation indicates that it is a variable of which bytes 1 and 2 will be exchanged with bytes 3 and 4. Similarly, the “U16_B” notation indicates that bytes 1 and 2 of the variable are exchanged. This is a “Little endian/Big endian” conversion.
Info4	Scale Factor: when the variable was generated automatically by SunSpec, this field contains the variable name that determines its scale factor when applicable. When the configured variable value is calculated, the read variable will have its decimal point position offset by as many digits as the value of its “scale factor”. The formula is var * 10sf” with “var” being the variable value that is read and “sf” the variable value indicated by the “scale factor”. For example, for a variable “var1” with scale factor variable “sf var1”. If “var1” is equal to “1234” and “sf var1” equal to “3”, the decimal point for “var1” will be offset by 3 digits to the right to obtain “1234000”. If “var1” is equal to “1234” and “sf var1” equal to “-2”, the decimal point for “var1” will be offset by 2 digits to the left to obtain “12.34”.

Modbus Frames: Modbus requests are grouped whenever possible, meaning that contiguous variables are processed using the minimum number of requests using the allocated resources to the maximum. On the other hand, when there is a free memory zone between 2 variables, the concentrator will generate 2 frames and will not attempt to group the 2 variables together in a single request.

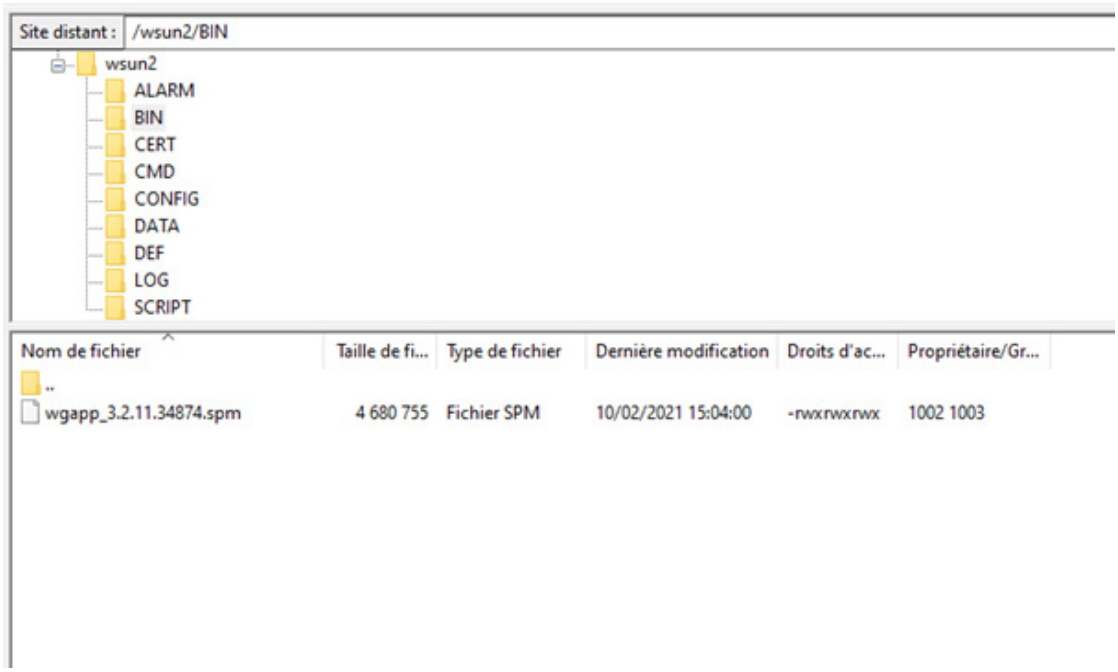
3.1.2.2.2.3 Proprietary Protocol

FIELD	DESCRIPTION
Info1	(see proprietary protocols specific appendix)
Info2	(see proprietary protocols specific appendix)
Info3	(see proprietary protocols specific appendix)
Info4	(see proprietary protocols specific appendix)

3.1.3 Updates

To update via the server:

- Upload the new file to the server in the configured directory as follows:



- Load the “config.ini” file
    - Put the binary name in “BIN_FileName”.

- Put the binary validation checksum that was provided with the binary in the “BIN_ Checksum “field.
- If the example below, the modified line will therefore be:

```
BIN_Checksum=26d00b3496803378cdc8820649cf9535
BIN_FileName=wgapp_3.2.11.34874.spm
```

3.2 Embedded Web Interface

To access the concentrator’s embedded web interface, follow the steps below:

- Launch the web browser: the web interface is compatible with the latest browser versions: Firefox, Chrome and Edge. Older versions may work but they are not supported (IE 7 for example).
- Enter the concentrator’s IP address in your web browser (the default address is: http://192.168.1.12 for LAN1 and http://192.168.2.12 for LAN2) to access the WebdynSunPM home page. (see section 2.4.5: “Ethernet interface”).

PARAMETERS	LAN1	LAN2
IP address	192.168.1.12	192.168.2.12

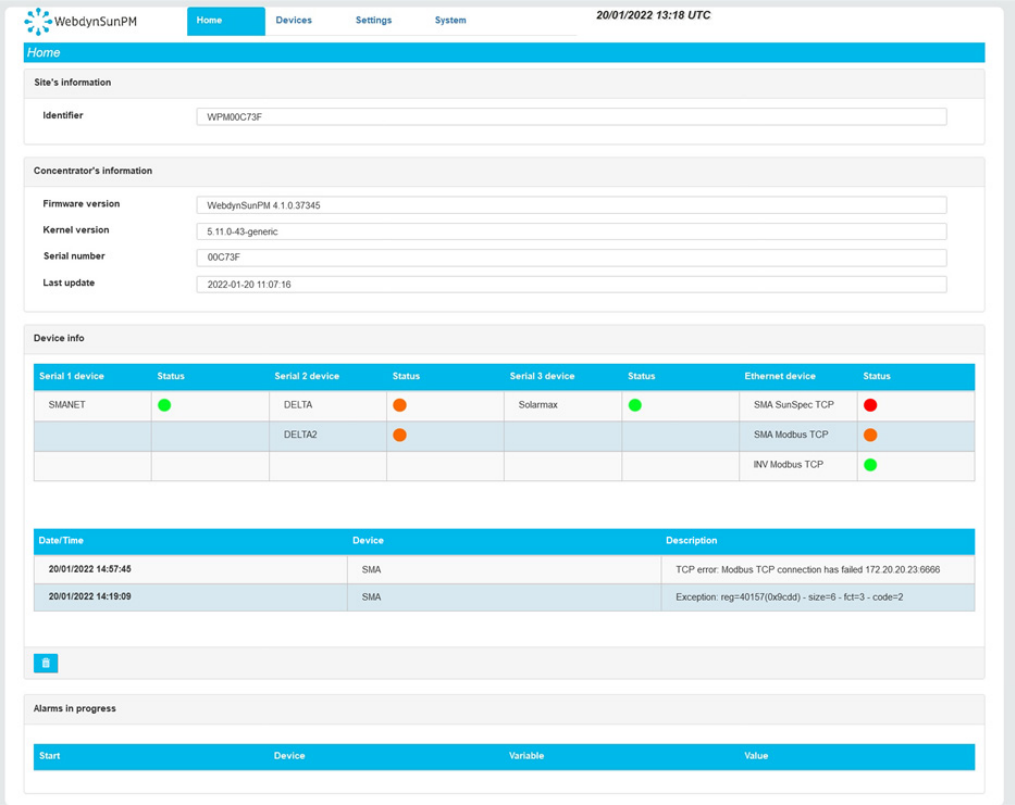
- An authentication window is displayed:

Enter your login and password:

LOGIN	PASSWORD
userhigh	high

Password: to secure access to the concentrator, we recommend changing the default passwords following the first configuration. Passwords are modified using the web interface (see section 3.2.2.7: “Password”) or using the configuration file (see section 3.1.2.1.1: ““<uid>_ config.ini” file”).

- The Home page is displayed:



The “Home” tab provides site and concentrator information as well as the general status of the equipment, current alarms and communication errors.

At the top of the screen is the essential information of the site.

The “Device info” panel contains the list of configured devices with their status as well as a list of errors detected on the various configured devices. Only the first error is displayed.

If new errors occur, they are not displayed. The 4 possible statuses for equipment are:

	The equipment has been found and the current configuration is functional
	The equipment has been found but one or more variables in the definition file are not functional
	The device was not found or the current configuration is not functional
	Unknown equipment status

Clicking on a piece of equipment gives direct access to its settings page.

Clicking on the trash can icon in the “Device info” panel resets the list of detected errors. In this case, all new errors will be displayed.

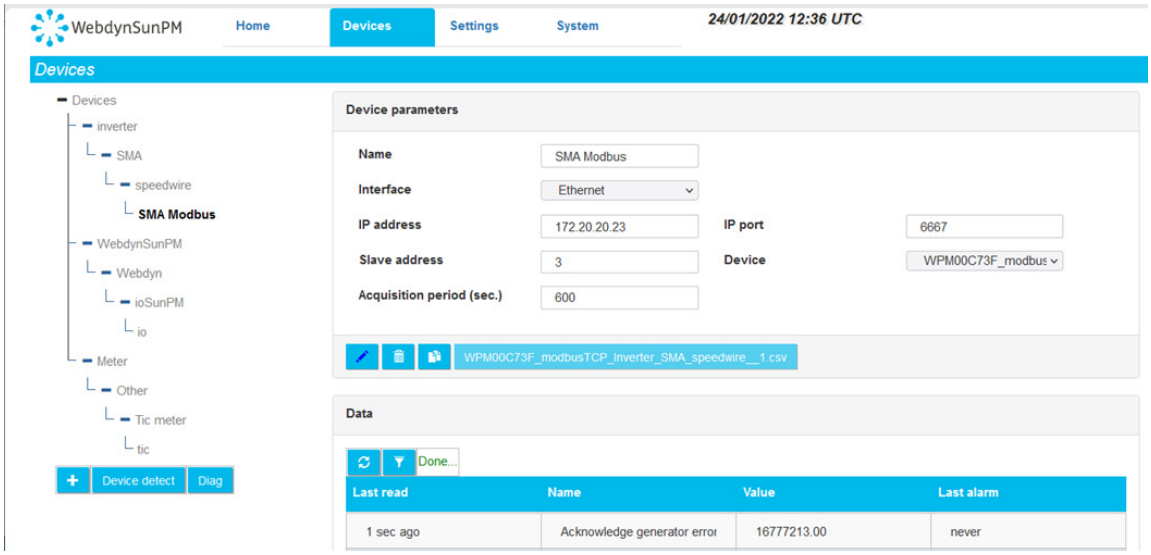
Further down, the “Alarms in progress” panel contains current alarms.

3.2.1 Devices

Devices can be configured in several ways:

- By editing or importing existing concentrator files, as described in section 3.1.2.1.3 - “<uid>_daq.csv” file.
- By running automatic device detection from the web interface or a text message.
- By manually editing the configuration using the web interface.

Device configuration using the web interface will be detailed in this section and can be accessed by clicking the “Devices” tab at the top of the screen.



3.2.1.1 Automatic Device Detection

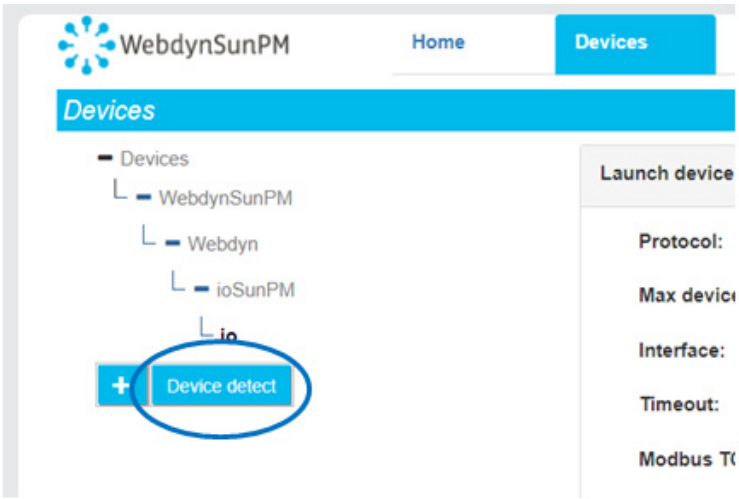
The concentrator can detect a certain number of devices meeting specific standards automatically. These devices are the following:

- SunSpec: The modbus devices in the tables meeting the SunSpec standard specification (http://www.sunspec.org/) can be detected and configured automatically using the Ethernet (modbus TCP) or serial (modbus RTU) connection.
- Proprietary protocol: (see specific appendix for proprietary protocols).

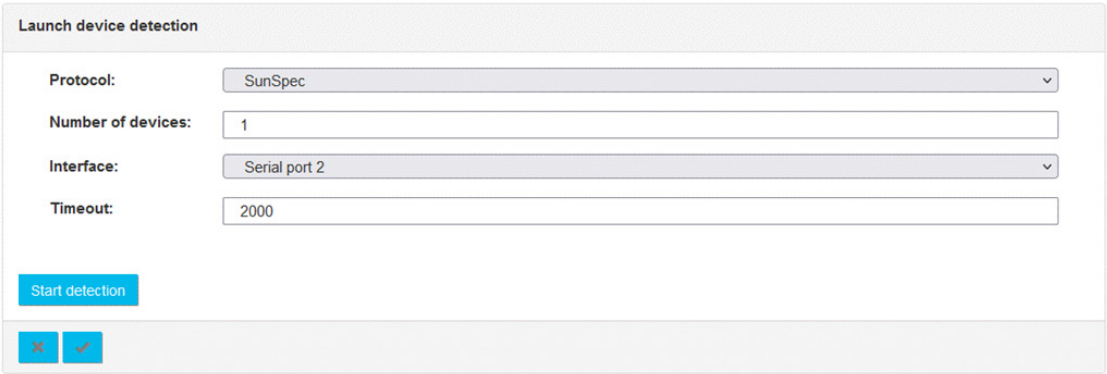
3.2.1.1.1 SunSpec Device Detection

The automatic detection of SunSpec devices requires the following steps:

- Connect the device to the concentrator on one of the serial connections or the Ethernet network.
- Check device configuration: for serial connections check that the configuration is the same on the concentrator and the device. For an Ethernet connection, check that the network configuration is compatible between the two devices.
- Go to the device page and click “Device detect”:



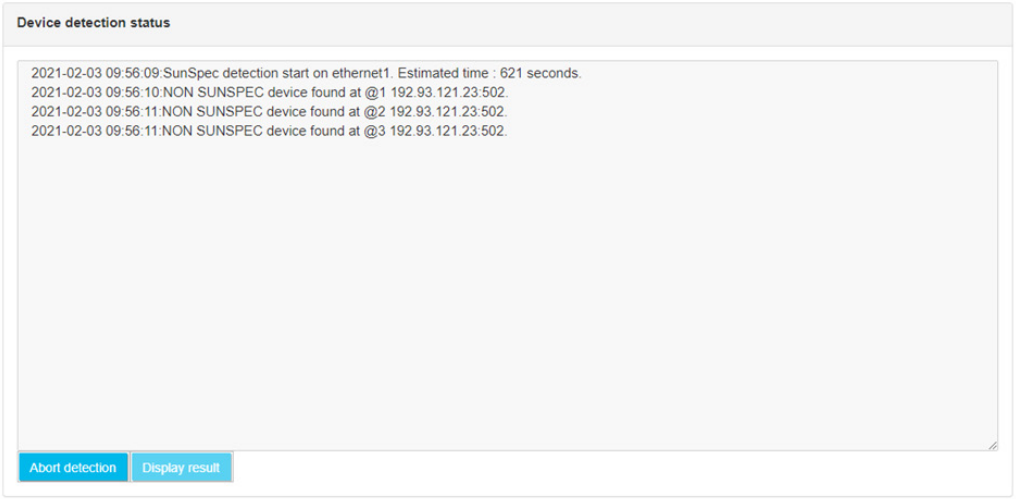
- The detection page is displayed:



- Select the SunSpec protocol from the drop-down list on the first “Protocol” field.

- In the “Number of devices” field, enter the number of devices to be detected. The default value is 1 “).
- Select the interface to use for detection in the “Interface” field. If the device responds in modbus TCP, the “LAN port 1” or “LAN port 2” value must be selected depending on the device IP address and the concentrator IP configuration. If the device responds on the serial connection, select the serial connection for the site installation: “Serial port 1”, “Serial port 2”, “Serial port 3” for concentrator serial ports 1 to 3. For the serial link to be selectable, you must first configure the serial link in the “Settings” tab then “Serial” and choose the “Modbus” protocol.
- Enter a timeout value. The default value of 2000 ms should be suitable for most installations. For slower devices, the timeout value can be increased.

- Then click the “Start detection” button to launch the detection. The progress window below is displayed:



On the first line, the window displays the SunSpec detection start date and time, the interface used and an estimate of the total detection time. The detection progress is displayed on the web page. :

- “NON SUNSPEC device found” means that a non SunSpec modbus device was detected at the indicated modbus address:

```
2021-02-03 09:56:10:NON SUNSPEC device found at @1
192.93.121.23:502
```

This means that a modbus device that does not meet SunSpec specifications was found at IP address “192.93.121.23” with modbus address 1.

- “SunSpec device found” means that a modbus device meeting SunSpec specifications was detected at the indicated modbus address:

```
2021-02-03 09:56:21:SunSpec device found at @126
192.93.121.23:502
```

This means that a modbus device that meets SunSpec specifications was found at IP address “192.93.121.23” with modbus address “126”.

- “Found table” means that a SunSpec table was detected on the device. The information line then indicates the table identifier, its size claimed by the device as well as technical information on the device and the modbus start register for the table:

```
2021-02-03 09:56:21:Found table 1:66 at 40004@126
192.93.121.23:502
```

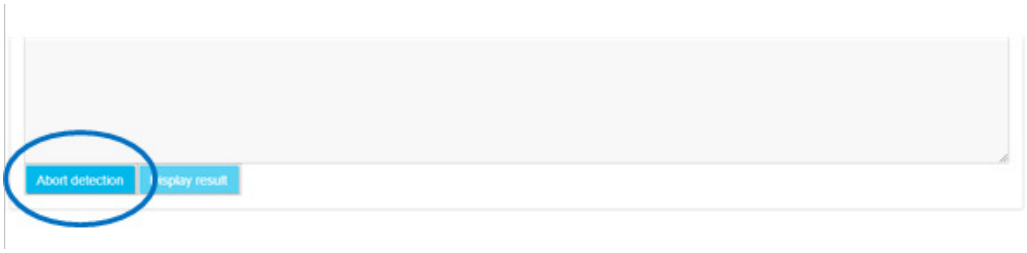
Means that table identifier “1” of size “66” registers was detected on the modbus TCP device at IP address “192.93.121.23” at register “40004” and modbus address “126”.

- “End of SunSpec detection” means that SunSpec detection is complete. The line indicates the number of detected devices:

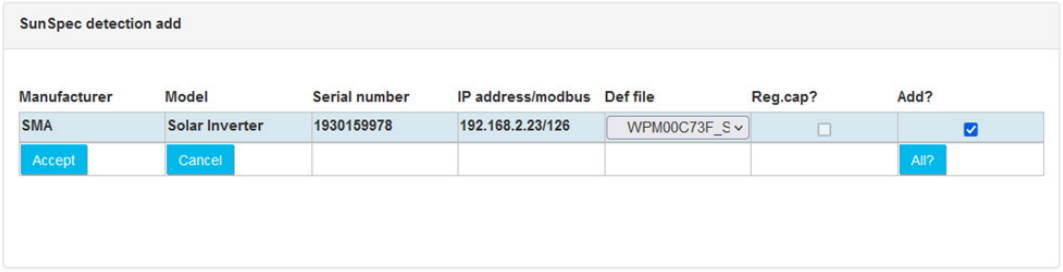
```
2021-02-03 09:56:53:End of SunSpec detection on
ethernet1. 1 devices found.
```

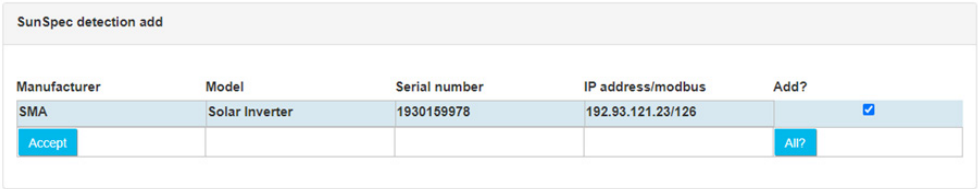
SunSpec detection completed on the Ethernet interface. 1 device was detected.

- It is always possible to interrupt detection by clicking the “Abort detection” button. But this is not recommended, because some equipment can be disturbed if the detection is stopped in the middle of a discovery.



- At the end of the SunSpec detection the last detection page is used to view all detected devices and eventually to add them to the configuration.





This screen therefore displays all the detected devices, as well as a certain amount of information read in the SunSpec device tables (model, serial number, address, manufacturer) as well as the name of the definition file associated with it.

There is also a checkbox at the right to select the devices to add to the configuration. Note that if the detected device is already part of the configuration, the checkbox is not checked by default. Otherwise the checkbox is checked for automatic addition.

Once the devices have been selected, a click on the “Accept” button imports the new configuration to the concentrator and the device appears in the configured devices.

The variables for this new device will be generated based on the detected SunSpec tables. Thus, for each detected SunSpec table, the following variables will be created:

- <idTable>_tableId: this variable will contain the identifier of the table in numeric form, as a 16-bit integer
- <idTable>_tableSize: this variable contains the size of the table in number of registers, as a 16-bit integer
- <idTable>_<variableName>: for each variable in the table declared in the SunSpec standards, a corresponding variable will be associated with the device. The variable name will consist of the table identifier, followed by the variable name
- <idTable>_<repeatBlock>_<variableName>: in the case of variables that come from a repeating block, the variables are created using the table identifier, the repetition number, as well as the name of the variable, so that the generated name is unique.

The variables generated are, by default, of the “Parameter” type and will therefore have the “Action” code 1, with the exception of the variables of tables 101, 102, 103, 111, 112, 113, 123, 160 and 401 which will be created with the “Immediate” type, i.e. code 4.

It should also be noted that the following variables will have a tag automatically applied:

- WMaxLimPct of table 123 receives the tag “cmdPwrPercent”
- WMaxLimPct_RmpTms of table 123 receives the tag “WMaxLimPct_RmpTms”
- WMaxLimPct_Ena of table 123 receives the tag “WMaxLimPct_Ena”
- VarPct_Mod of table 123 receives the tag “VArPct_Mod”

Note that if the device already existed in the configuration and the user forces a new import, the previous device is not overwritten. A new device is created in addition to the pre-existing device.

If the user clicks on the “Cancel” button, this list is erased, and the page displays the detection type selection screen again.

3.2.1.1.2 Detection of Proprietary Protocol Equipment

(see proprietary protocol specific appendix)

3.2.1.2 Manual Device Management

Devices can be managed manually on the web site.

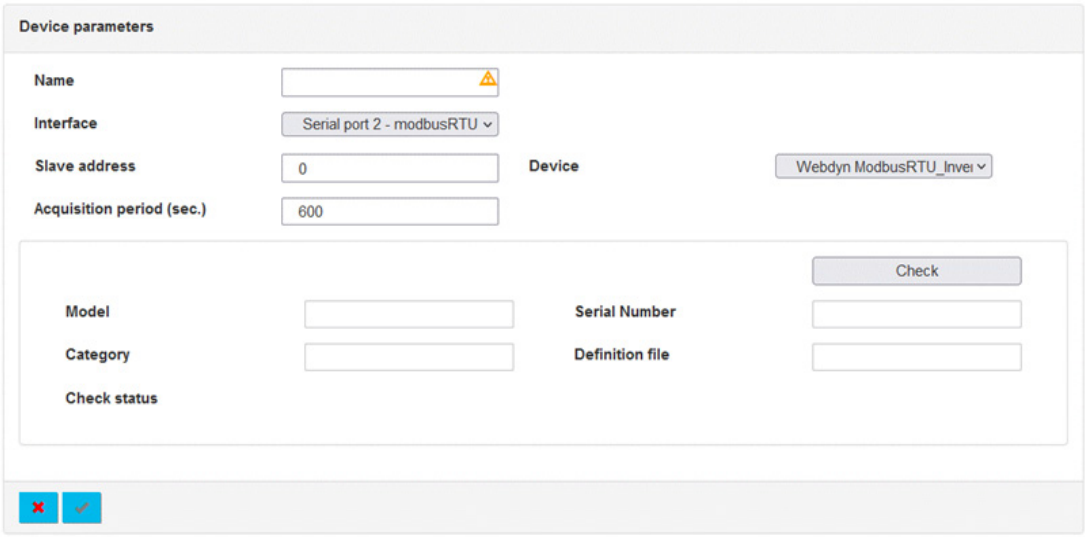
Everything is managed from the “Devices” tab described previously (see section 3.2.1: “Devices”).

3.2.1.2.1 Add a Device

To add a device, first click the “+” button:

Q

The next page is displayed:

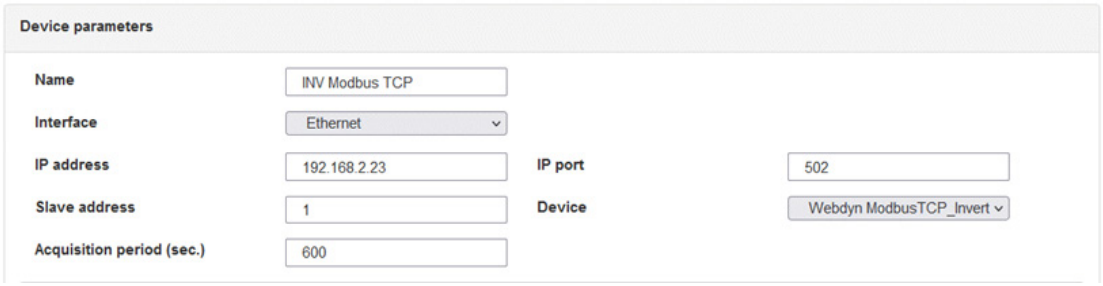


In this form, it is possible to select a piece of equipment in relation to its corresponding definition file and to choose its interface, its name, its address as well as its acquisition period. Start by entering a name for this equipment. This name must be unique.

Choose the interface where the equipment is connected:

- Ethernet port: if the device is connected to the Ethernet network (for example for a device that makes “modbus TCP”).
- Serial port: if the device is only connected to one of the serial ports of the WebdynSunPM (for example for a device that does “modbus RTU”). Only serial ports whose settings are compatible with the selected protocol are displayed.

In this example, these are therefore “Modbus” type devices.



Since the device is of the TCP type (in this example: “modbusTCP”), the following fields also appeared:

- IP address: enter the IP address of the modbusTCP device on the network
- Port: Enter the port number to access the device. Usually 502
- Slave address: modbus address of the device

When the device is of the serial type (in this example: “ModbusRTU”), the following fields appear:

- Slave address: modbus address of the device

Once these parameters have been entered, it is also possible to enter the periodicity of data acquisition.

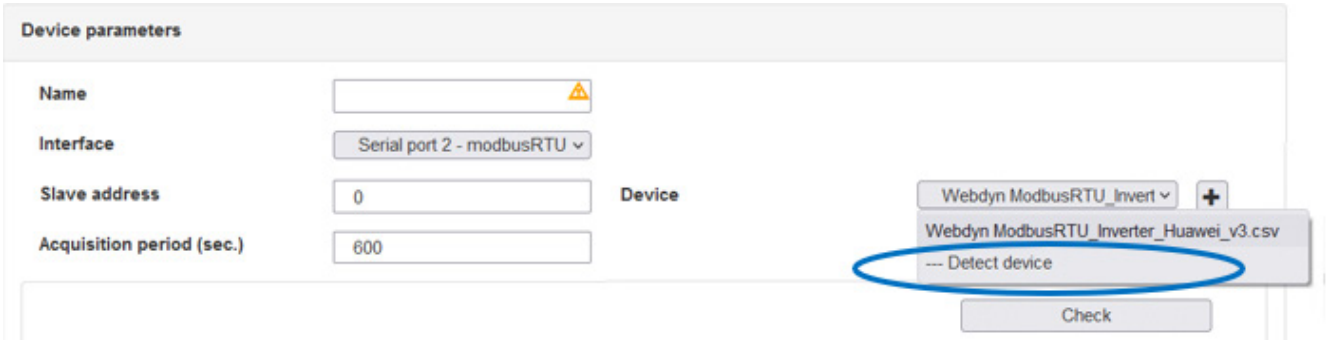
For the choice of equipment in the “Device” drop-down list, it is possible to:

- Launch an equipment detection
- Select the definition file present in the drop-down list

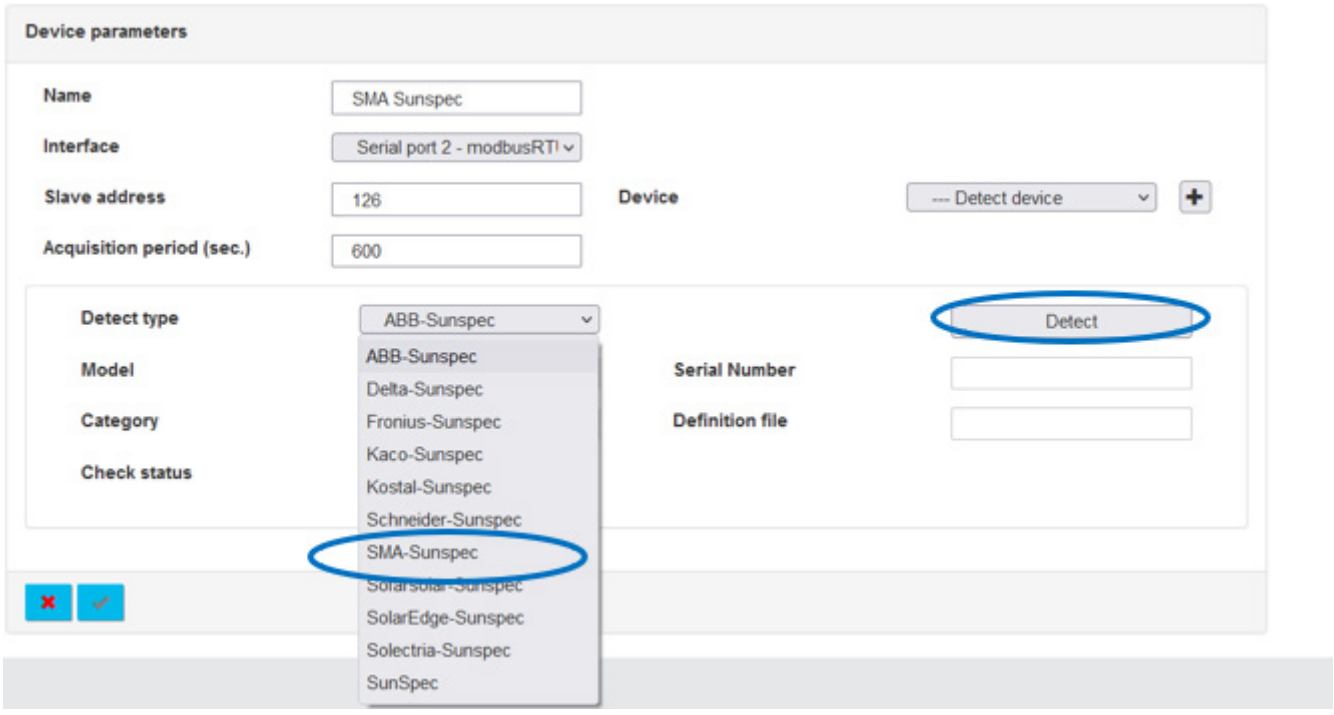
It is possible to import a new definition file by clicking on the “+” symbol.

Launch an equipment detection:

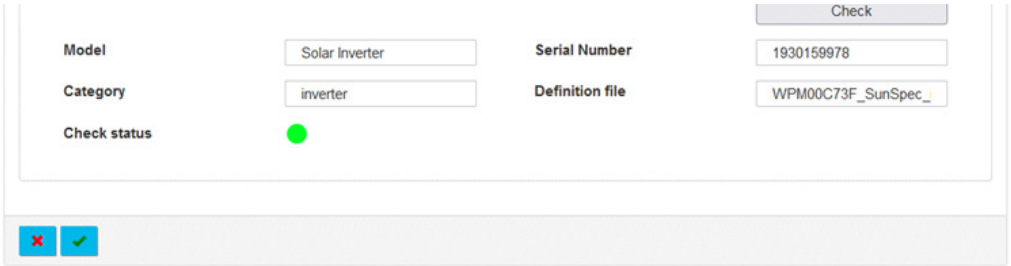
Select “Detect device” in “Device”:



Choose the type of equipment to detect (in this example “SMA-Sunspec”), then start the detection by clicking on the “Detect” button:



Examine the result:

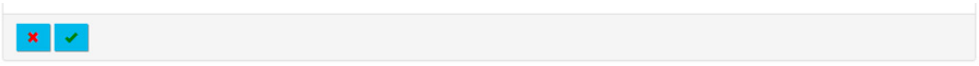


The model, category, serial number, definition file as well as the status of the equipment is displayed when the equipment is found.

If the “Check status” is:

	The equipment has been found and the current configuration is functional
	The device was not found or the current configuration is not functional

Validate the addition of the equipment by clicking on the checkmark at the bottom of the page.

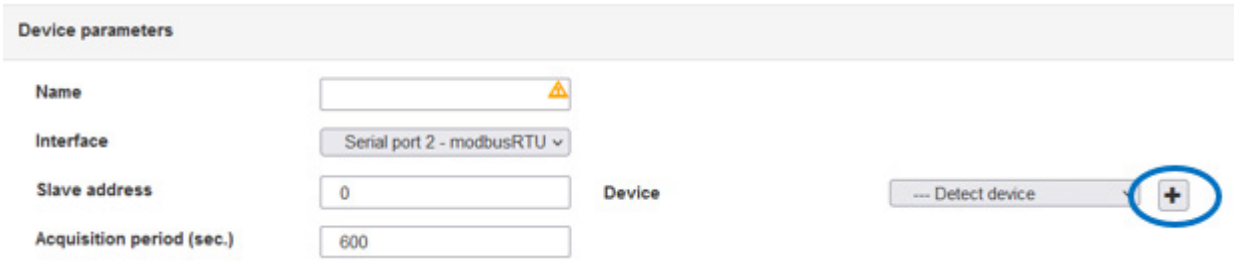


The new equipment then appears in the tree structure on the left side of the screen:

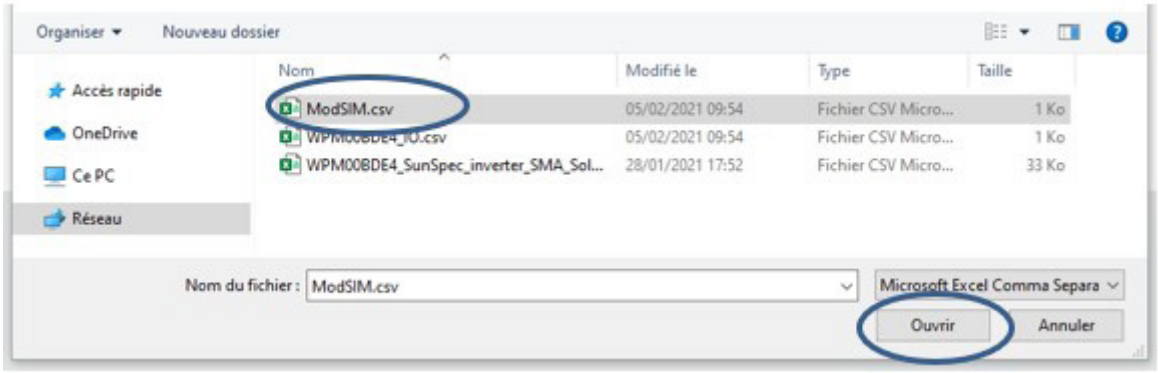
Import a new definition file:

If the desired device is not found in the “Device” drop-down lists, you must import the corresponding definition file.

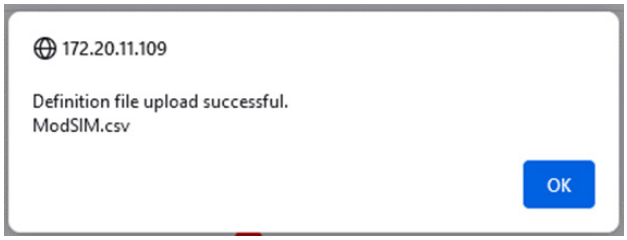
To do this, first click on the button:



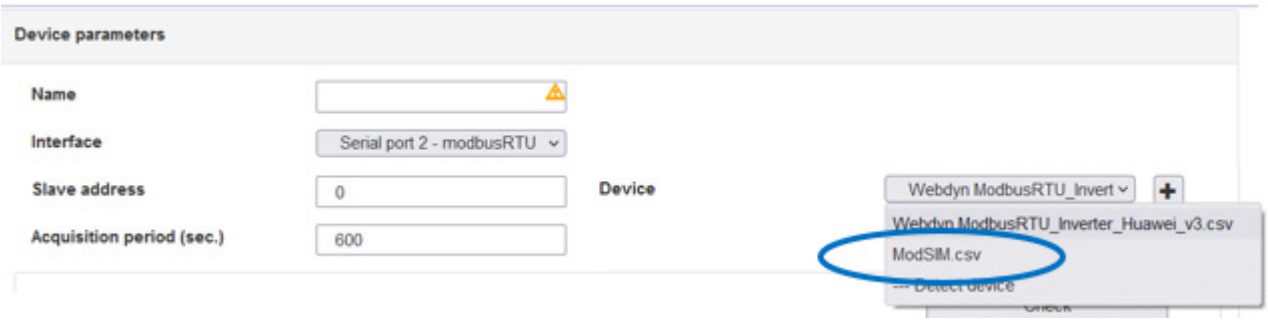
The next window is displayed, select the definition file to import:



Then select the file to import and click the “Open” button.



The message “Definition file upload successful” then appears on the web interface to indicate the success of the operation, click on the OK button to finalize the import of the definition file.



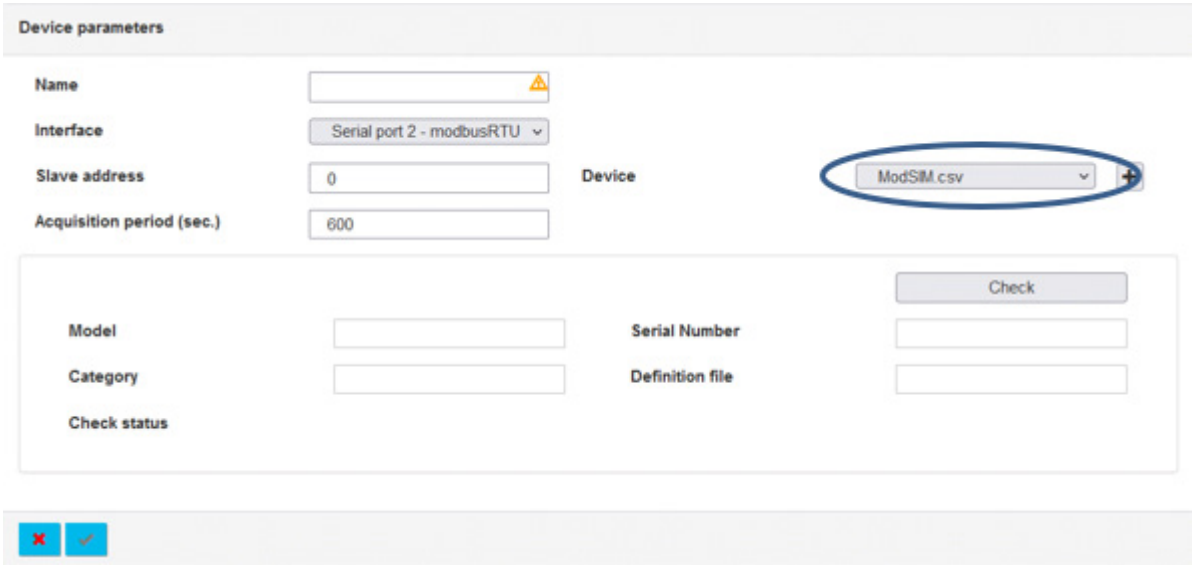
Once the definition file is loaded, it is possible to select the required device characteristics in the drop-down lists.

For the test file, the header contains the following information:

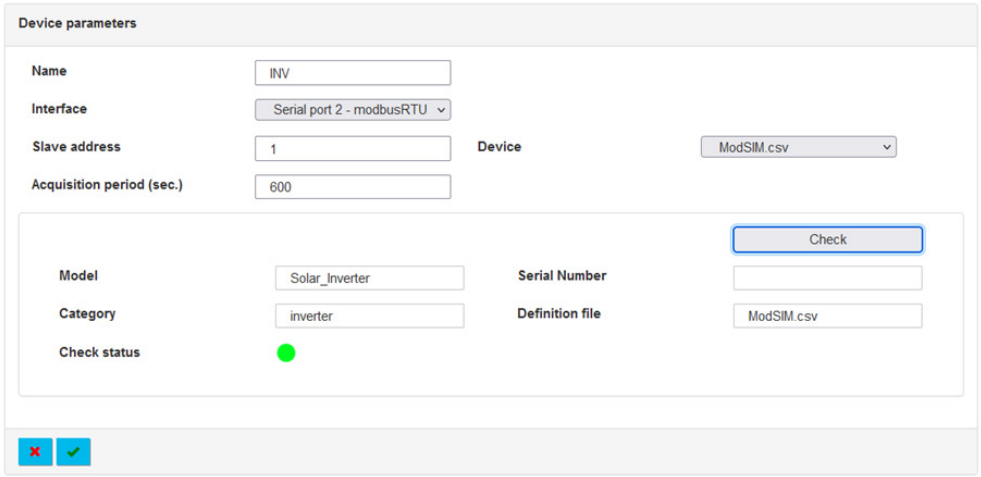
modbusTCP	modSIM	Webdyn	test_PC1
1	3	2	U16
2	3	3	U16

Select the definition file present in the drop-down list:

It is therefore possible to enter the corresponding device:



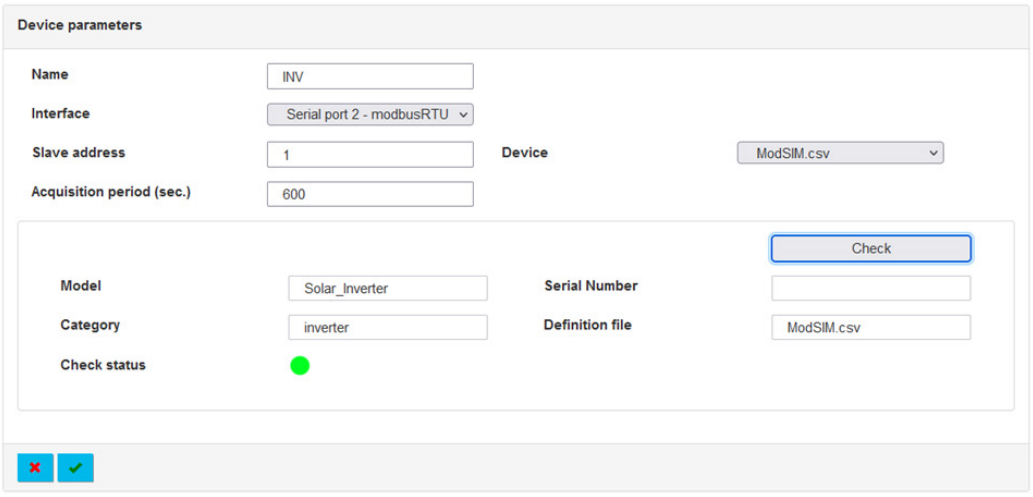
When the configuration is satisfactory, click on the “Check” button to launch the check on the equipment:



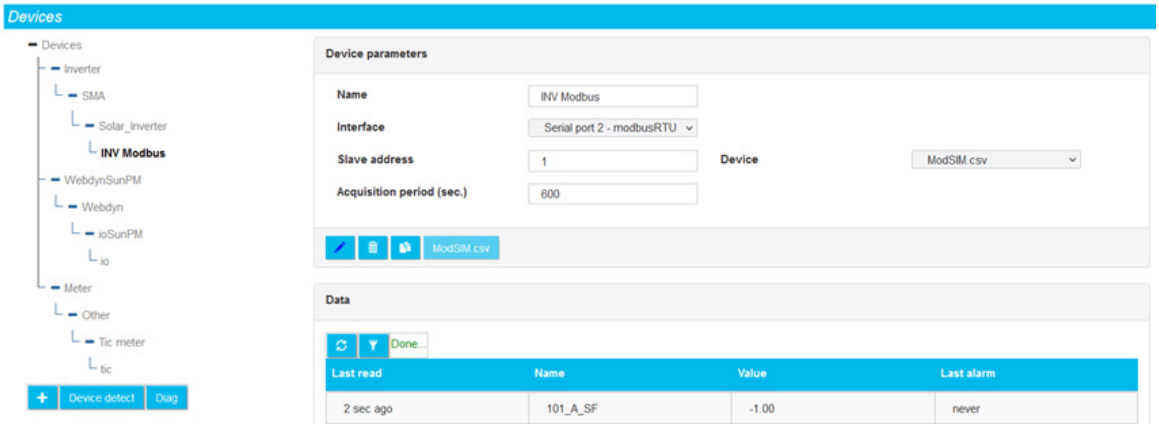
If the “Check status” is:

	The equipment has been found and the current configuration is functional
	The device was not found or the current configuration is not functional

Validate the addition of the equipment by clicking on the checkmark at the bottom of the page.



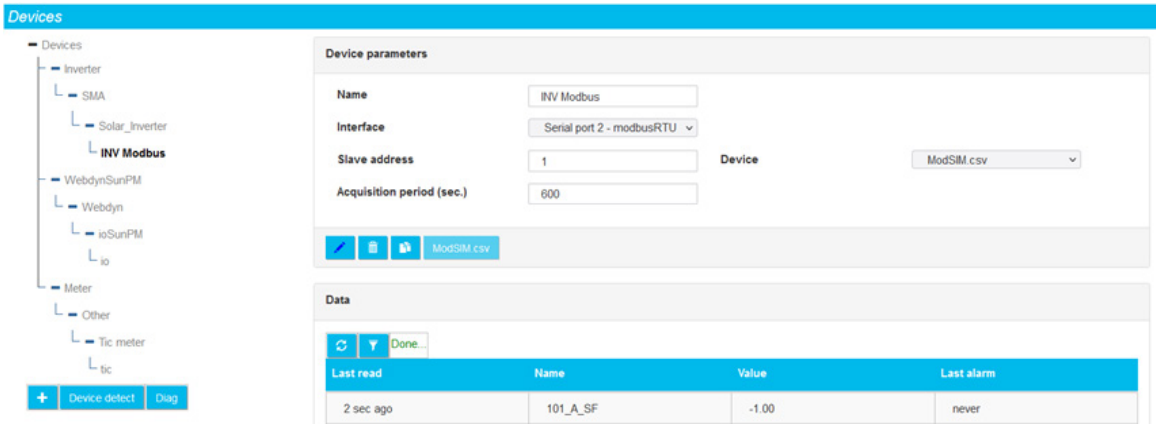
The new device then appears in the tree structure on the left of the screen:



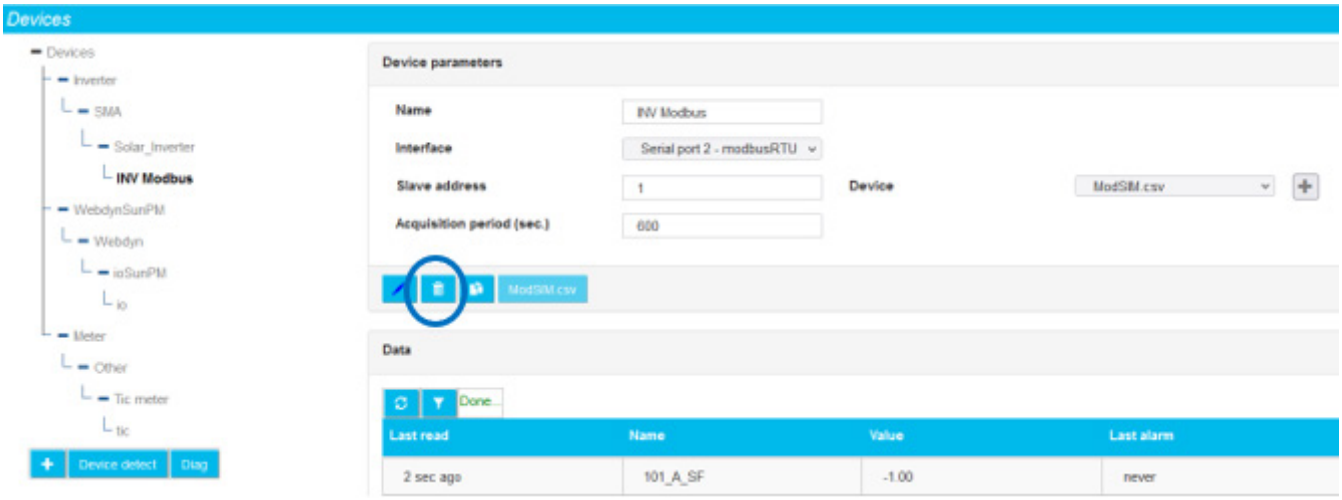
Any definition file imported into the WebdynSunPM which is not used for 24 hours will be automatically erased by the concentrator during its next remote connection.

3.2.1.2.2 Deleting a Device

To delete a device, first select the device to delete:



Then click the trash icon under the device description:



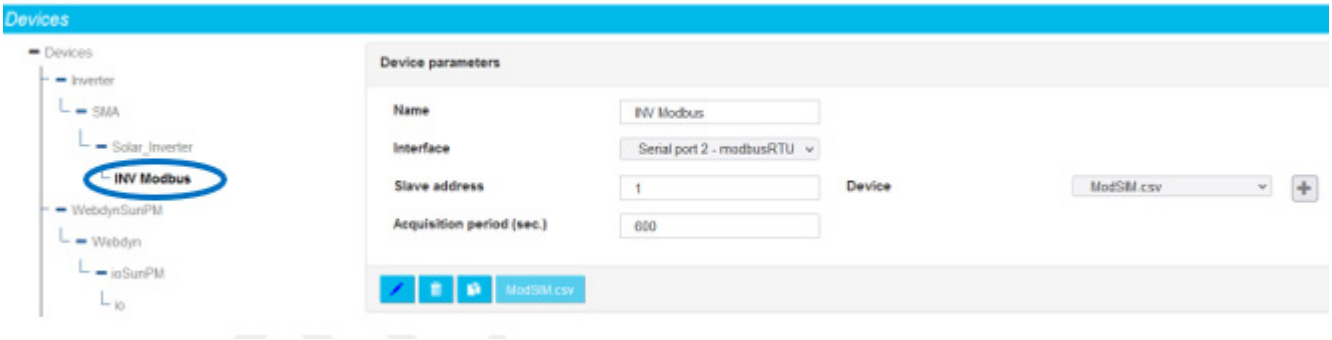
A dialogue box is displayed requesting confirmation.

After confirmation, the deleted device is removed from the concentrator databases.

The modification will be carried over to the configuration files at the next server connection.

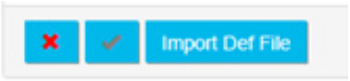
3.2.1.2.3 Editing a Device

To edit a device, first select the device to edit:



Then click the edit button:

As soon as the button is pressed, the device page changes. The device management button bar switches to edit mode:



It is then possible to modify the different device fields and thus change the name, interface, acquisition frequency, model and the specific protocol parameters: IP address and port number for IP devices, slave address for modbus devices, timeout, etc.

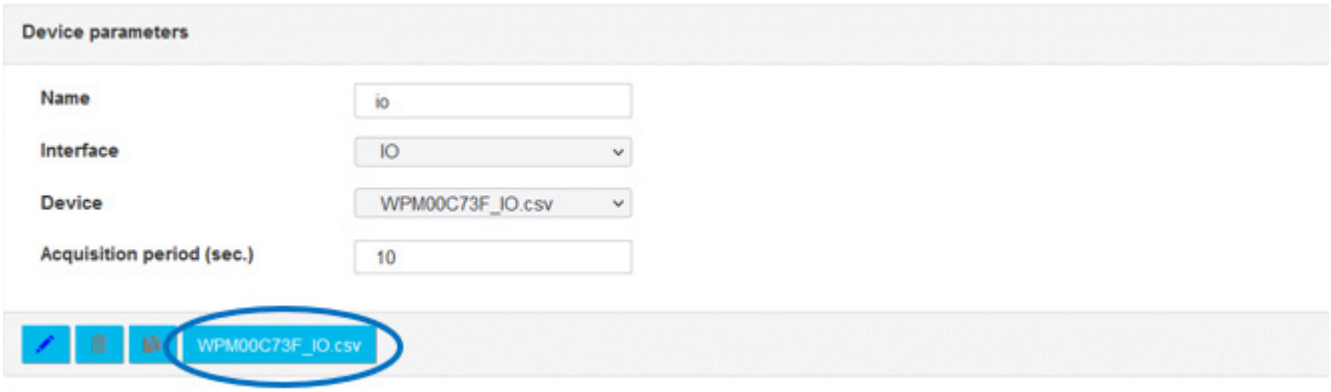
Pressing the validation button validates the data entry. Pressing the cancel button ignores any changes made on the data entry form.

It is also possible to modify the equipment variables by importing a new definitions file. To do this, simply click on “Import new definition file” in the “Device” drop-down menu and select the new definition file (See chapter 3.1.2.1.3.4: “- Declaration of equipment”). Once the file has been imported, you must reselect the definition file to correspond to the new imported equipment in order to reload the variables, as when adding equipment (see chapter 3.2.1.2.1: “- Adding equipment”).

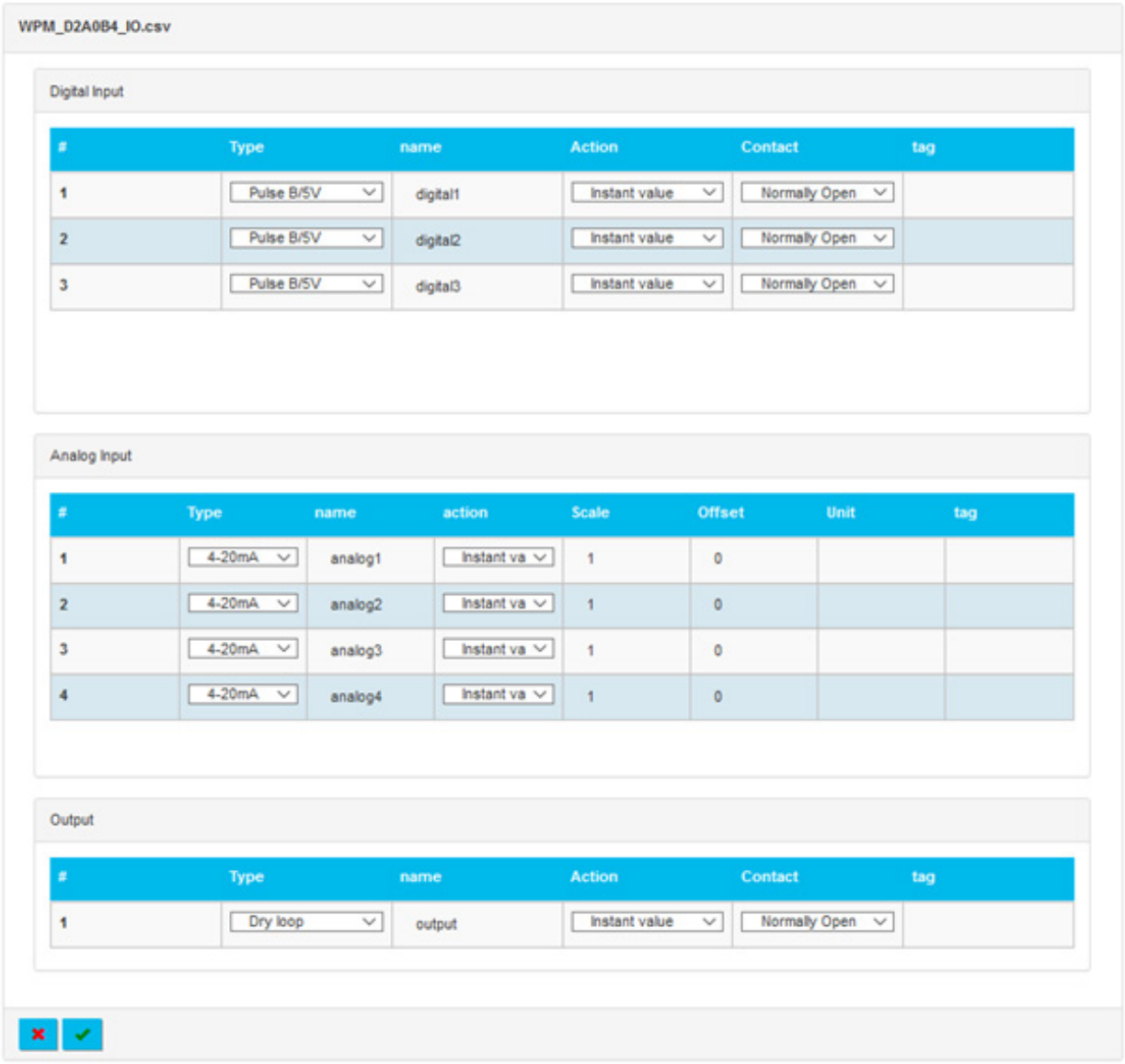
3.2.1.2.4 Modifying the Input/Output Configuration

Input/output configuration can be done directly from the device management page.

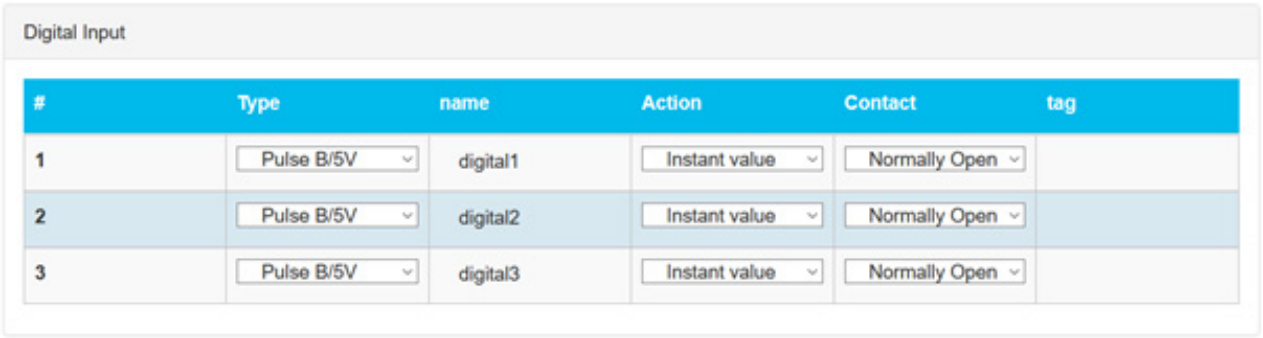
Just click the concentrator’s IO definition file button:



The input/output status page display is then replaced by the following display:



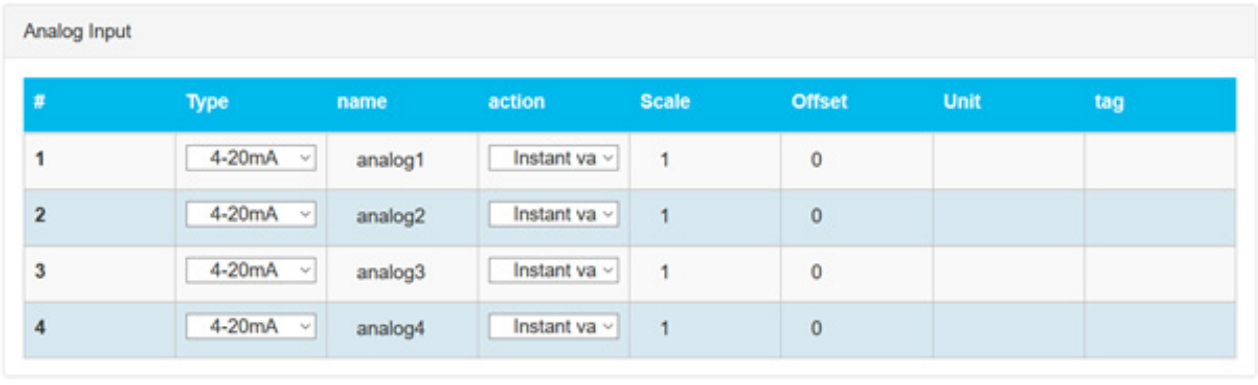
The first part of the screen is used to configure the 3 digital inputs:



The fields are:

FIELD	DESCRIPTION
Type	Digital input type. The authorised values are: • Dry Loop: dry loop (ON-OFF) • Pulse B/5V: S0 (see chapter 2.4.7.2: “Discrete digital inputs/S0 (pulse)”) • Pulse A/24: S0 (see chapter 2.4.7.2: “Discrete digital inputs/S0 (pulse)”)
Name	Input name. This value will be used on the web pages and on the MQTT servers The name on the concentrator must be unique
Action	Action associated with the input. The possible actions are: • Ignored: the value will be ignored. The variable is disabled. It will not appear in the data files • Instant value: the input is of the instant value type. The data read at collection time will be stored in the data file using a single field. • Alarm on change: the input is of the alarm type. When a status change is detected, an alarm is triggered. The data read at collection time will be stored in the data file using a single field.
Contact	Indicates the “normal” status of this input, namely its standby state. The possible values are: • Normally open: the input standby state is 0. When the input is activated, its value switches to 1. • Normally closed: the input standby state is 1. When the input is activated, its value switches to 0. This difference is used on the web pages for display and in the scripts. It has no impact on the data files.
Tag	Contains an identification making it possible to use the variable in scripts. (Calculation of totals, issuing of commands to multiple devices, etc.). This name must be unique to allow unambiguous identification and use in the scripts.

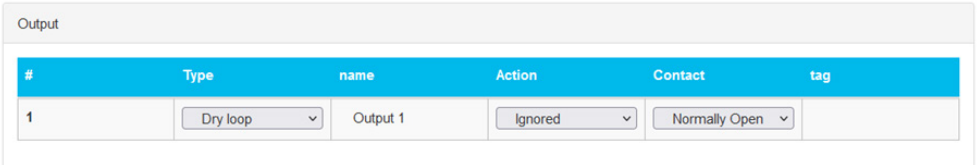
The second part of the input/output edit screen is used to configure the 4 analog inputs:



The fields are:

FIELD	DESCRIPTION
Type	Analog input type. The allowed values are: • 4-20mA • 0-10V
Name	Input name. This value will be used on the web pages and on the MQTT servers The name on the concentrator must be unique
Action	Action associated with the input. The possible actions are: • Ignored: the value will be ignored. The variable is disabled. It will not appear in the data files • Instant value: the input is of the instant value type. The data read at collection time will be stored in the data file using a single field. • Min/Max/Average: the input is of Min/Max/Average type. The data read at the time of collection will be stored in the data file, using three separate fields. Over the acquisition period, the first field will memorize the minimum value, the second field the maximum value and the third field will be an average over all the variables acquired over this period.
Contact	Indicates the “normal” status of this input, namely its standby state. The possible values are: • Normally open: the input standby state is 0. When the input is activated, its value switches to 1. • Normally closed: the input standby state is 1. When the input is activated, its value switches to 0. This difference is used on the web pages for display and in the scripts. It has no impact on the data files.
Tag	Contains an identification making it possible to use the variable in scripts. (Calculation of totals, issuing of commands to multiple devices, etc.). This name must be unique to allow unambiguous identification and use in the scripts.
Contact	Indicates the “normal” status of this input, namely its standby state. The possible values are: • Normally open: the input standby state is 0. When the input is activated, its value switches to 1. • Normally closed: the input standby state is 1. When the input is activated, its value switches to 0. This difference is used on the web pages for display and in the scripts. It has no impact on the data files.
Tag	Contains an identification making it possible to use the variable in scripts. (Calculation of totals, issuing of commands to multiple devices, etc.). This name must be unique to allow unambiguous identification and use in the scripts.

The last part of the input/output editing screen is dedicated to the configuration of the relay output:

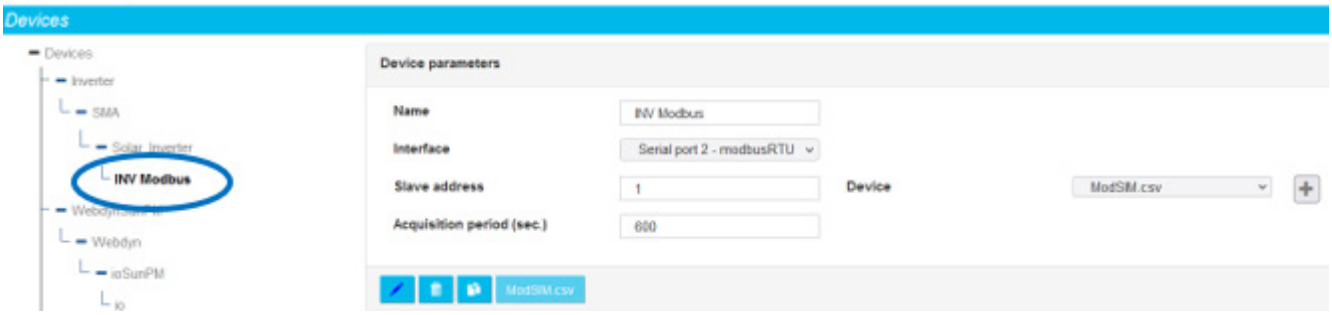


The fields are:

FIELD	DESCRIPTION
Type	Output type. The allowed values are: • Dry loop: dry contact
Name	Name to give to the entry. This value will be used in web pages and on MQTT servers. The name must be unique on the hub.
Action	Action associated with the entry. The allowed values are: Possible actions are: • Ignored: The value will be ignored. The variable is disabled. It will not appear in the data files. • Instant value: the input is of instant value type. The data read at the time of collection will be stored in the data file, using only one field. • Alarm on change: the input is of the alarm type. When a change of state is detected, an alarm is triggered. The data read at the time of collection will be stored in the data file, using only one field.
Contact	State of the relay at rest: • Normally Open: normally open • Normally Closed: normally closed
Tag	Contains an identification allowing the use of the variable in question in scripts. (Calculation of accumulation, sending of command to multiple equipment, etc...). This name must therefore be unique to allow unequivocal identification and use in scripts.

3.2.1.2.5 Duplicating a Device

To duplicate a device, first select the device to duplicate:



The click the duplication button:

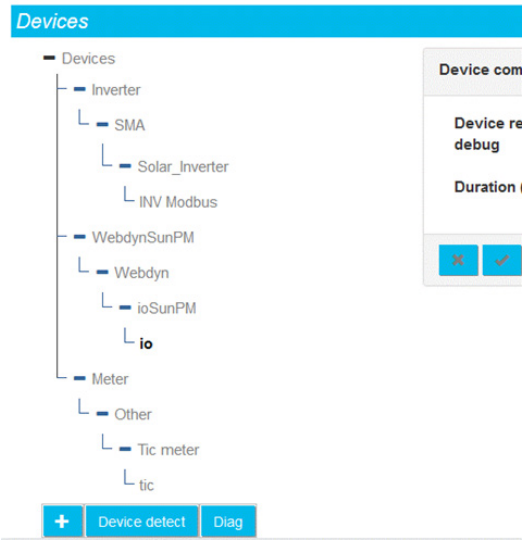
When the button is pressed, the device is duplicated.

The new instance uses the same definition file as the original device. It then becomes possible to rename and change the definition file by editing the device. (see section 3.2.1.2.3: “Editing a device”).

3.2.1.3 Equipment diagnostic tools

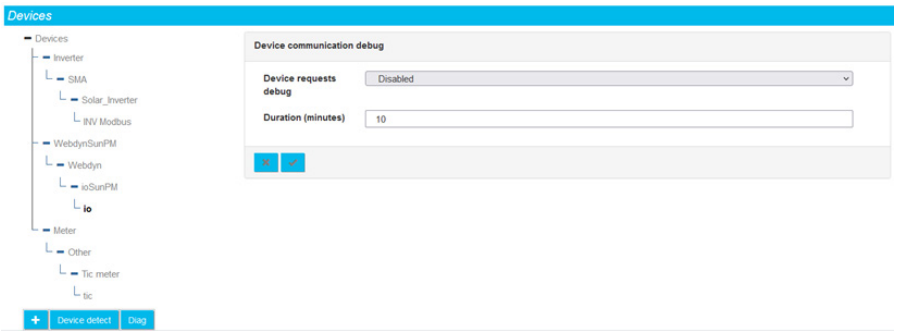
Diagnostic tools are made available to analyze the frames sent to the equipment as well as the frames received. These tools help to understand what happens in case of configuration problems.

3.2.1.3.1 Enable communication logs



Access to the equipment communication diagnostic tools pages is done by clicking on the “Diag” button in the equipment page.

The following page then appears:



The first parameter is dedicated to the communication logs with the devices, via the serial ports or the IOs.

When communication traces are selected for a given interface, all communication on this interface will be logged and sent to the server as a log timestamped to the millisecond.

Given the amount of information transmitted and received, these logs are only activated for a defined period, to be configured in the “Duration” field. This period is expressed in minutes.

Once the time expires, the activation of the log switches back to “Disabled” automatically.

3.2.1.3.2 Exploitation of logs

When communication logs are available on the concentrator, they will be deposited during the next connection to the server, in the directory configured for the logs.

3.2.1.3.2.1 Serial and ethernet interface logs

The name of the log files for communications with the devices is built according to the same principle as the other log files of the gateway, namely: “uid”_ “interface”_“date”.log.gz.

The interface corresponds to the one chosen for the logs, namely:

- serial1
- serial2
- serial3
- ethernet

The logs contain the following information:

```
datetime_1;data;
datetime_2;data
...
datetime_Y;data
```

“datetime” is in the format DD/MM/YYYY hh:mm:ss.mmm, with DD the day of the month, MM the number of the month, YYYY the year, hh the hour of the communication, mm the minute, ss for seconds, and mmm for milliseconds.

The “data” field contains the transmitted data as well as the meaning. For outgoing communications, the meaning is “=>”. For incoming communications, the meaning is “<=”.

If the communication interface is of the “serial” type, the data in hexadecimal is then provided.

If the communication interface is of the “ethernet” type, the IP address of the equipment to be monitored is logged, then the data is provided in hexadecimal format, as for the serial protocol.

It should be noted that in the case of modbus, the complete modbus TCP frame is provided if the link is of the ethernet type. Otherwise it is the modbus RTU frame.

In the case of modbus errors, the frame can be prefixed with the following messages:

- “*** CRC ***”: A CRC error was detected on the incoming frame. The frame is therefore invalid. This is a hardware communication error. Too many CRC errors is a sign of a problem in the installation. The frame is ignored.
- “*** BAD SLAVE ***”: A slave with an incorrect number responded to the request. This may be due to an equipment configuration error which may disrupt the correct operation of the installation. The frame is ignored.
- “*** EXCEPTION ***”: The slave responded to an exception to the request. This means that the request that was sent is incorrect for the equipment in question.
- “*** INVALID ID ***”: A slave responded to a ModbusTCP request with an incorrect ID. The frame was discarded.

- “*** INVALID FCT ***”: The response contains an incorrect function code. The frame is ignored.

3.2.1.3.2.2 Input/output logs

The name of the input/output log files is built according to the same principle as the other gateway log files, namely: “uid”_IO_“date”.log.gz.

The logs contain the following information:

```
datetime_1;data
datetime_2;data
...
datetime_Y;data
```

“datetime” is in the format DD/MM/YYYY hh:mm:ss.mmm, with DD the day of the month, MM the number of the month, YYYY the year, hh the hour of the communication, mm the minute, ss for seconds, and mmm for milliseconds.

The “data” field contains the input-output data.

First appears the type of information:

- “In”: indicates that it is an input
- “Out”: indicates that it is an output

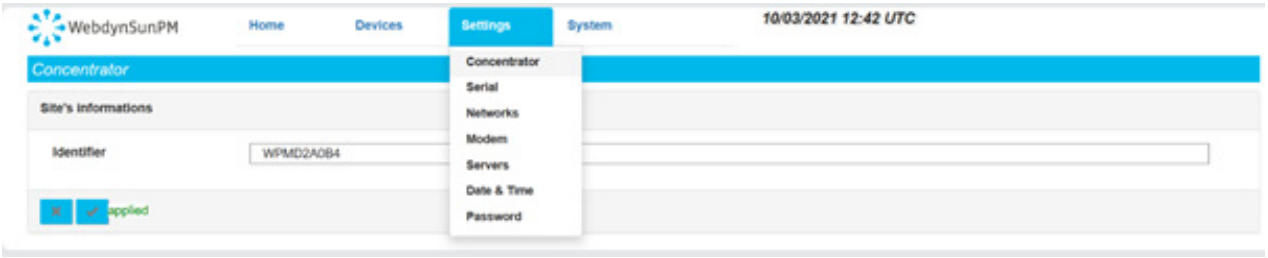
Then, the name of the impacted input/output, as defined in the configuration

Then, the new state of the input/output:

- 0: the input/output is closed
- 1: input/output is open
- Any other value contains the value of an analog input

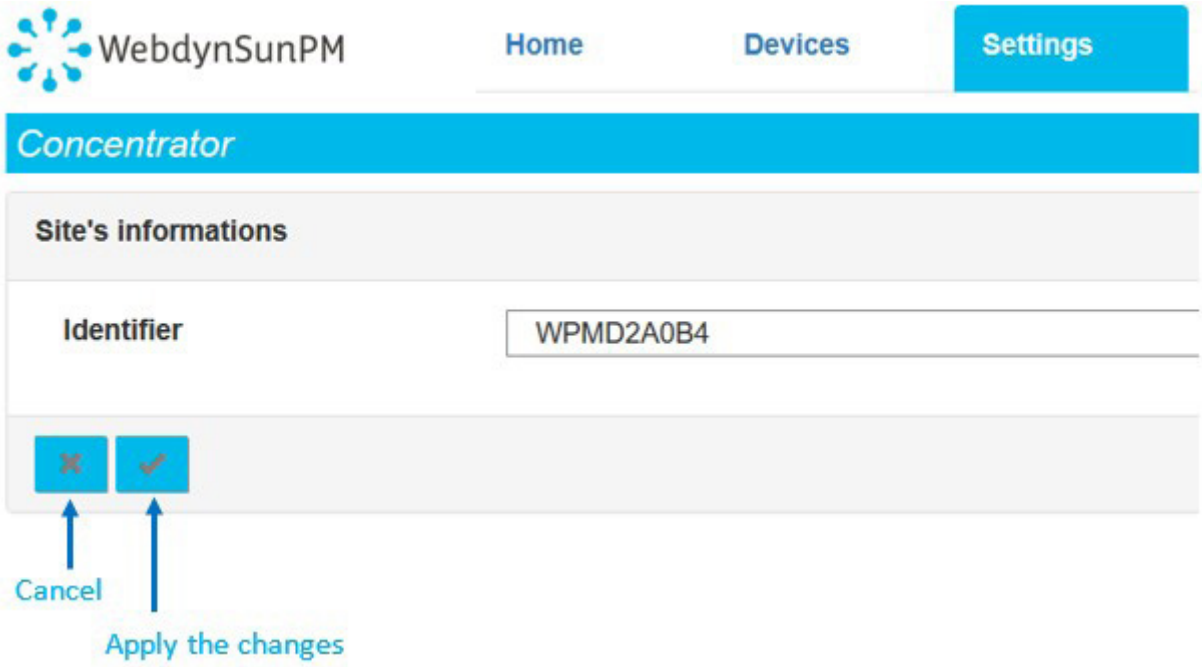
3.2.2 Settings

All the concentrator settings are grouped together on the “Settings” tab. The settings are split into several parts on the menu.



3.2.2.1 Concentrator

The “Concentrator” part is used to change the concentrator’s identifier and description.



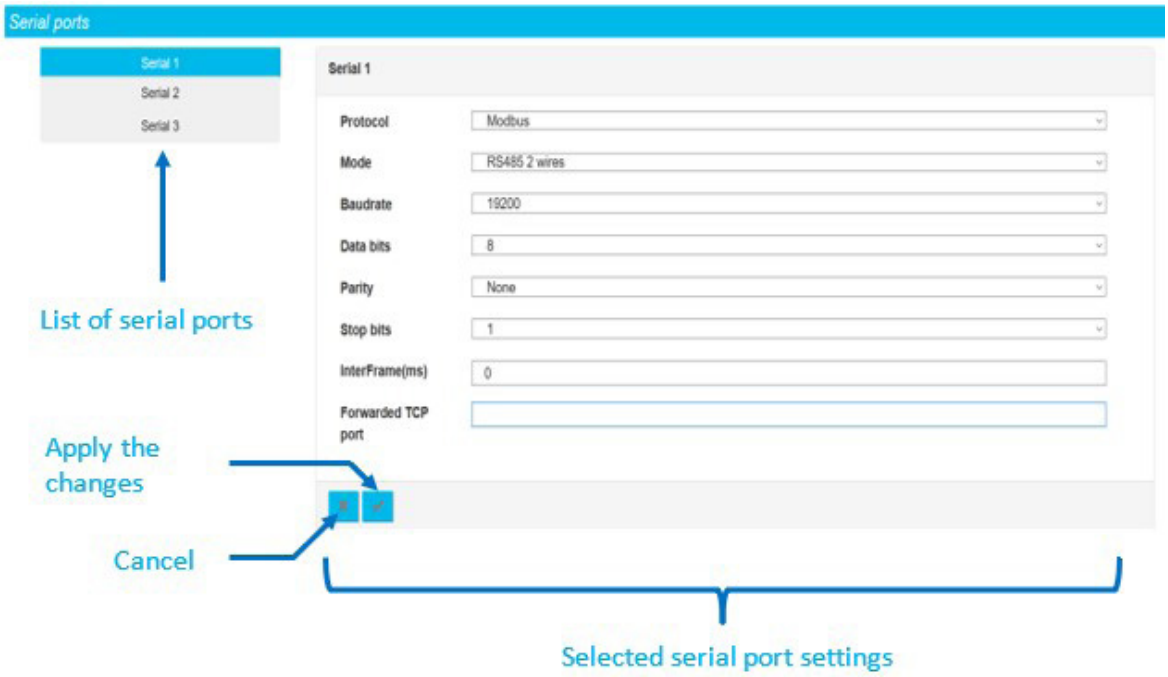
WEB INTERFACE	PARAMETER <uid>_config.ini configuration file	DESCRIPTION
Identifier	Concentrator_Identifier	Concentrator identifier

By default, a unique “WPMxxxxxx” identifier is filled in. The “xxxxxx” corresponds to the last 6 characters of the concentrator’s MAC address indicated on the product label. (see section 2.2.2: “Identification”)

Identifier: the identifier is used when creating the names of the files uploaded to the server. It is important that it be unique to be able to know where the files on the remote server come from. The concentrator identifier is identified as follows in the document: <uid>.

3.2.2.2. Serial

The “serial” part is used to configure 3 RS485/422 serial ports each with their own settings and output. (see section 2.4.6: “RS485/RS422 Serial interface”).



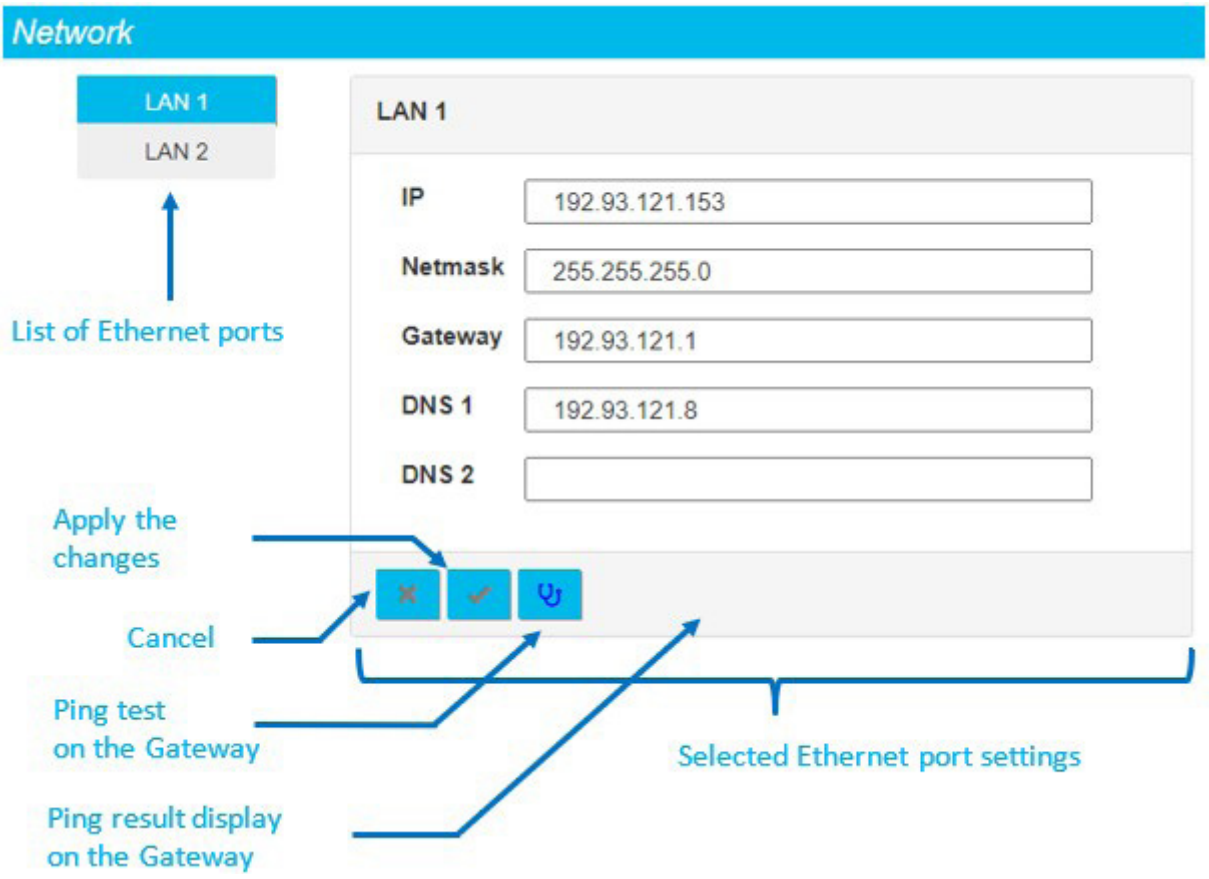
The possible settings for each serial port are:

WEB INTERFACE	PARAMETER <uid>_daq.csv configuration file	DESCRIPTION
Protocol	protocol	The protocol type for this serial interface: • Modbus: serial port configured in modbus RTU mode • Proprietary protocol: serial port configured for the proprietary protocol (see proprietary protocols specific appendix). • PW1: serial port configured for the PowerOne protocol
Mode	wires	Serial interface mode: • RS485 2 wires: Half-Duplex (2 wires) RS485 serial connection • RS485 4 wires: Full-Duplex (4 wires) RS485 or RS422 serial connection

Baudrate	baudrate	Serial connection speed in bauds: • 1200 • 2400 • 4800 • 9600 • 19200 • 38400 • 57600 • 115200 • 230400 • 460800
Data bits	data_bits	Number of data bits: • 7 • 8
Parity	parity	Serial connection parity: • None: no parity • Odd: odd parity • Even: even parity
Stop bits	stop_bits	Number of stop bits: • 1 • 2
InterFrame	interframe	The waiting time between 2 frames exchanged on the serial port. This time is expressed in ms. See the explanation in section “3.1.2.1.3.3 - Serial port configuration”.
Forwarded TCP port	forwarded_port	Forwarded TCP port. If there is a value in this field, the concentrator opens a modbusTCP port on the entered port number. When modbusTCP devices connect to this port, all sent requests are directly forwarded to the ModbusRTU bus and the response is returned to the connected device using this modbusTCP port. This option is used to create a communication tunnel between modbusTCP devices and the local ModbusRTU network. The requests are slotted between the concentrator’s internal monitoring requests.

3.2.2.3. Networks

The “Networks” part is used to configure the 2 Ethernet interfaces (LAN1 and LAN2) available on the concentrator. These Ethernet interfaces make it possible for the concentrator to belong to 2 different Ethernet networks (see section 2.4.5: “Ethernet Interface”).



To be able to run a ping test on the Gateway IP address, it must have been entered and the configuration applied before clicking the Test button.

The 2 Ethernet interface settings are:

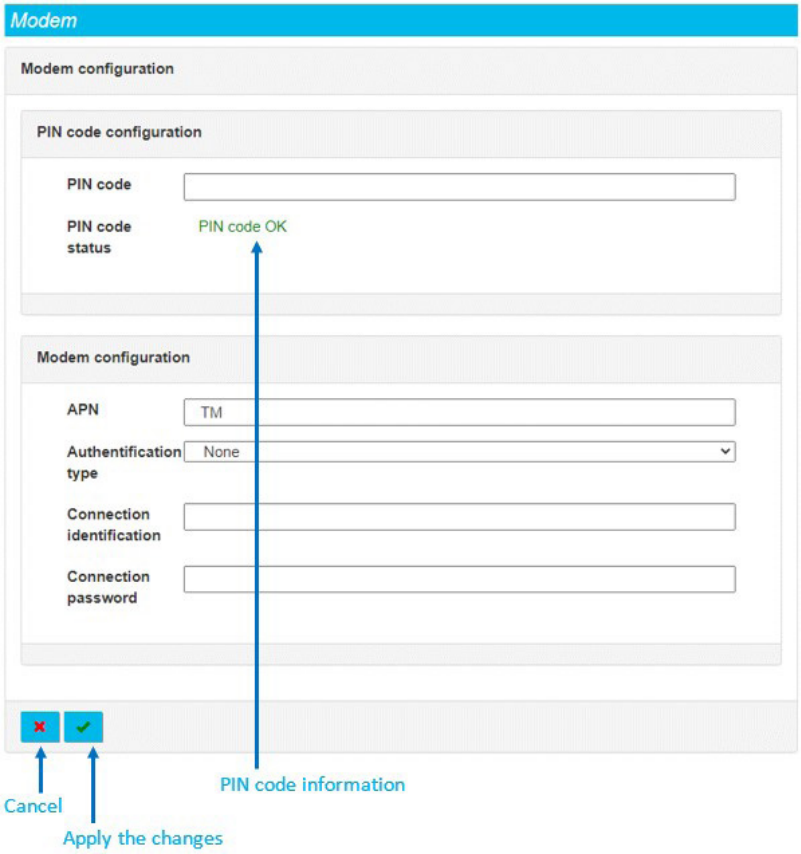
WEB INTERFACE	PARAMETER <uid>_daq.csv configuration file	DESCRIPTION
IP	ip	IP address at which the concentrator is accessible using the Ethernet network.
Netmask	mask	Your Ethernet network subnet mask. This mask limits the Ethernet network to defined IP addresses and separates the network ranges from each other.

Gateway	gateway	Your Ethernet network gateway address. The gateway address is the IP address for the device that connects to the internet. The address entered here is usually your ADSL/fibre router address.
DNS 1	dns1	DNS 1 server. DNS (Domain Name System) servers translate explicit internet addresses (for example, www.webdyn.com) into their corresponding IP addresses. Enter the DNS server addresses you received from your internet service provider (ISP) here. You can also enter your router IP address. You can also use the google DNS: “8.8.8.8”.
DNS 2	dns2	DNS 2 server. If the DNS1 server fails.

If your local network is managed by a network administrator, contact them before including the WebdynSunPM gateway in your network.

3.2.2.4. Modem

The “Modem” part is used to configure the modem and get network information. To use the modem, the SIM card must first be inserted into the product (see section 2.4.2.2: “SIM card”).



The modem parameters are:

WEB INTERFACE	PARAMETER <uid>_daq.csv configuration file	DESCRIPTION
PIN code	pin	The SIM card PIN code to be entered if it has one
APN	apn	Your mobile operator's APN name (required for an IP connection)
Authentification type	authentication	Operator authentication type (optional depending on the operator): • None: no authentication • PAP: PAP type authentication The login and password must be entered below
Connection identification	login	Your mobile operator's user name (optional depending on the operator)
Connection password	password	Your mobile operator's password (optional depending on the operator)

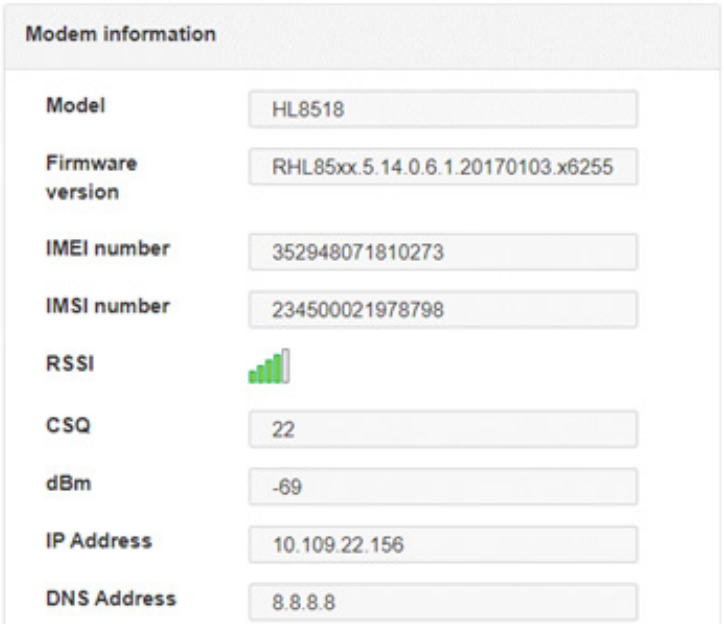
Contact your SIM card provider to find out what information to enter to configure the modem.

The PIN code status gives information about the SIM code which can be:

PIN CODE STATUS	DESCRIPTION
PIN code OK	The modem can access the SIM card. Either the SIM card PIN code is correct or the SIM card has no PIN code
PIN code required	The SIM card expects a code that must be entered in the "PIN code" field
Unknown	Miscellaneous modem errors
PUK code required	The SIM card is locked due to too many incorrect attempted codes
SIM card not inserted	There is no SIM card in the concentrator

If the SIM card has an activated PIN code and the PIN code entered into the concentrator is incorrect, the SIM card can lock. It can be unlocked using a mobile phone using the PUK code provided by the operator.

The displayed modem information is:



This information is:

MODEM INFO	DESCRIPTION
Model	The model model built into the concentrator
Firmware version	The firmware version for the model built into the concentrator
IMEI number	Unique international identification number for the modem built into the concentrator
IMSI number	IMSI number for the inserted SIM card allowing a network to identify a user
RSSI	Indication of the modem's reception power level in number of bars
CSQ	Reception signal level returned by the modem of 0 to 31
dBm	Signal level returned by the modem interpreted in dBm of -113 to -51
IP address	IP address assigned automatically by the mobile operator
DNS address	DNS address assigned automatically by the mobile operator
Network type	The type of network the modem is connected to (2G/3G/4G.)
Carrier name	The name of the operator to which the modem is currently connected.

The reception signal level is very important when the product is being installed.

It is used to find out the connection status between the modem and the operator to which the concentrator is attached.

This is very important to avoid transmission errors and too long exchange times between the concentrator and the remote server. Refer to the table below.

CSQ VALUE	RSSI IN DBM	DESCRIPTION
CSQ > 20	RSSI > -73	Level Excellent
15 < CSQ < 20	-83 < RSSI < -73	Level good
12 < CSQ < 14	-89 < RSSI < -85	Level limit
CSQ < 11	RSSI < -91	Level unstable

If the reception level is limit or unstable, the use of an offset antenna or a test with a different mobile operator is recommended to improve the connection.

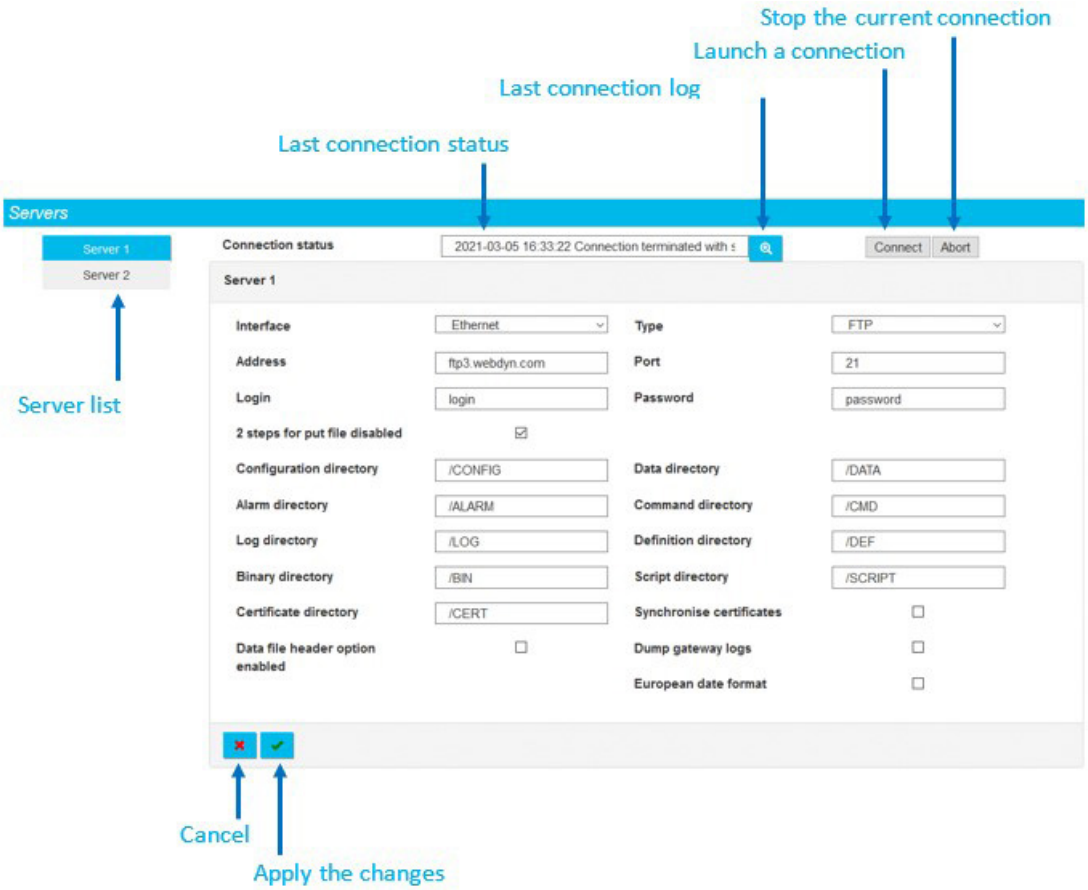
If the IP and DNS addresses are not displayed, there may be several causes:

- The APN has not been entered or is incorrect.
- If authentication is required, the login and/or password are incorrect.

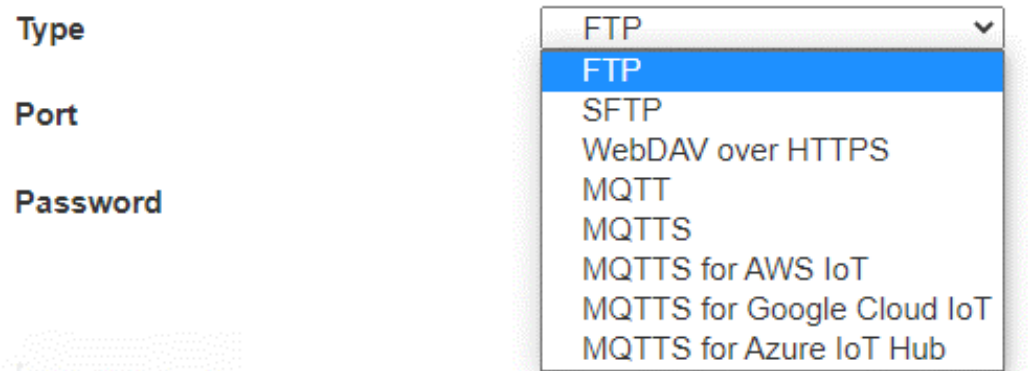
As long as the problem persists, the concentrator will not be able to use the modem to connect to the remote server. On the other hand, the text message commands will function.

3.2.2.5 Servers

The “servers” part allows you to configure the 2 servers available on the concentrator and to plan synchronization schedules. This synchronization can also be done locally on an SD card.



The concentrator supports 8 different types of remote servers, which are:



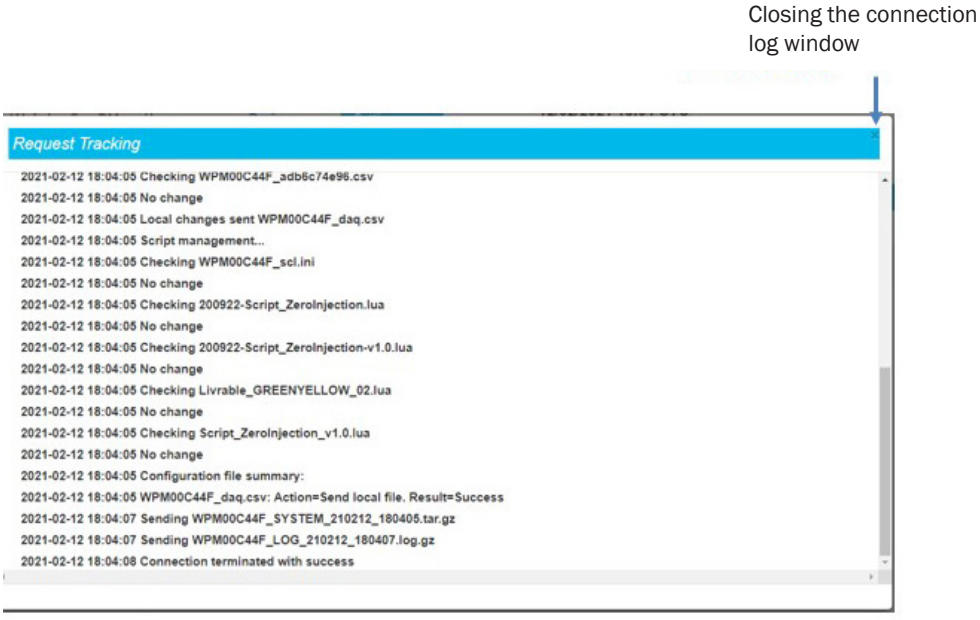
The concentrator can also store data locally on an SD card, for this you must select in the “SD card” interface:

The choice of the type of server or the SD Card interface modifies the parameters to be entered. The server settings remain the same.

Server 2 is only used to back up configurations. Only server 1 manages the synchronization of configuration files. The MQTT/MQTTS/MQTTS AWS IOT/MQTTS Google Cloud IoT/MQTTS Azure IoT Hub is only available on server 2 (backup).

To verify that the server configuration is correctly configured, it is advisable to click on the “Connect” button. A window with the connection log is displayed, allowing you to see all the files exchanged between the concentrator and the remote server. The last line allows you to quickly know if the connection went well or if a failure occurred.

Stopping the current connection by pressing the “Abort” button is not immediate, the stop occurs between two actions. If an action is in progress, it must end first. An action corresponds to each line described in the connection log.

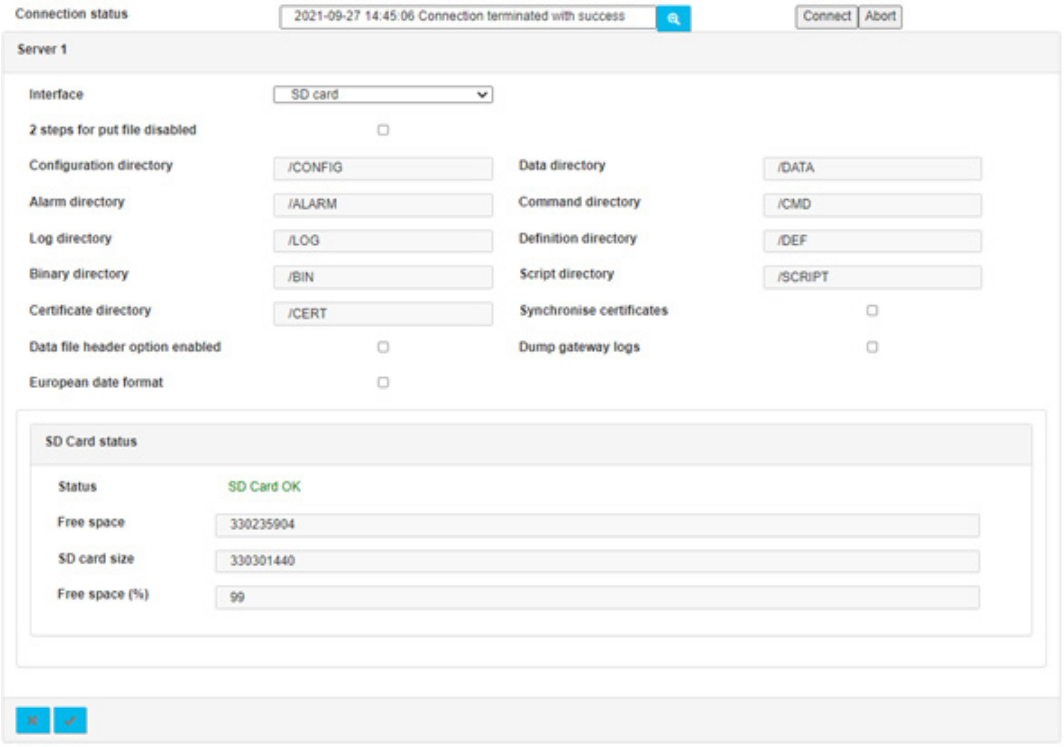


In the event of an error, check all the server parameters.

Please contact the administrator of the server to which you want to connect in order to obtain the parameters to be entered on the concentrator and, if necessary, the certificates and the key for encryption and authentication.

3.2.2.5.1 SD Card

When the “SD card” interface is selected, the display removes unnecessary fields and the directory configuration cannot be modified in this interface. In addition, a frame appears with the current information about the SD card:



In order to guarantee a compatible tree structure on all SD cards, the directory configuration cannot be modified.

All server settings for the SD card interface are only accessible via the web interface and not stored in the hub configuration file. To use the concentrator’s SD card interface, you must activate it via the web interface.

The parameters on the 2 servers are:

WEB INTERFACE	PARAMETER <uid>_config.ini configuration file	DESCRIPTION
Interface	SERVER_Interface SERVER2_Interface	Choice of the network interface to be used by the server: Sdcard

If the directories do not exist on the SD card, they will be created automatically by the concentrator.

The SD card information box contains the following information:

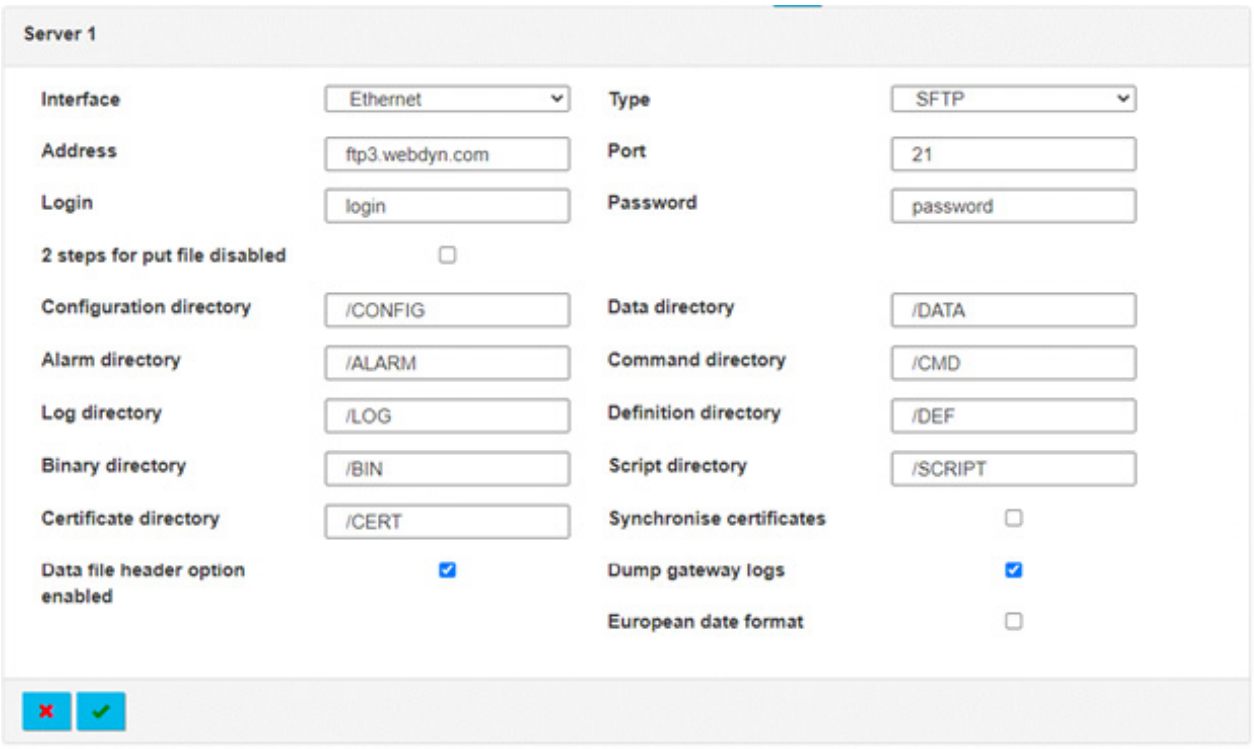
- Status: Status of the SD card, the possible states are:
  - Unknown: the state is not known because no access to it has been made so far.
  - SD Card OK: The last attempt to write to the SD card was successful.
  - Failed to mount SD Card: SD card was not detected. Check that the card is correctly inserted and that it is correctly formatted (FAT32 or exFAT).
  - SD Card read error: the SD card was detected but there were errors reading the information it contains. Check that the card is correctly formatted (FAT32 or exFAT).
- Free space: memory size available on the SD card (in bytes).
- SD card size: capacity of the SD card (in bytes).
- Free space (%): SD card occupation percentage. Allows you to easily and quickly monitor the filling of the SD card.

At each connection, the hub will deposit data and retrieve information from the SD card. The “SD Card status” insert reflects the information read from the card:

It is of course possible to force a connection by clicking on the “Connect” button present on the page.

3.2.2.5.2 FTP/SFTP

FTP and SFTP servers have the same settings.



The parameters on the 2 servers are:

WEB INTERFACE	PARAMETER <uid>_config.ini configuration file	DESCRIPTION
Interface	SERVER_Interface SERVER2_Interface	Choice of the network interface to be used by the server: • Ethernet (see chapter 3.2.2.3: “Networks”) • Modem (see chapter 3.2.2.4: “Modem”)
Type	SERVER_Type SERVER2_Type	Choice of server protocol: • FTP: FTP server • SFTP: SFTP server
Address	SERVER_Address SERVER2_Address	IP address or server name
Port	FTP_Port FTP2_Port	FTP/SFTP server port

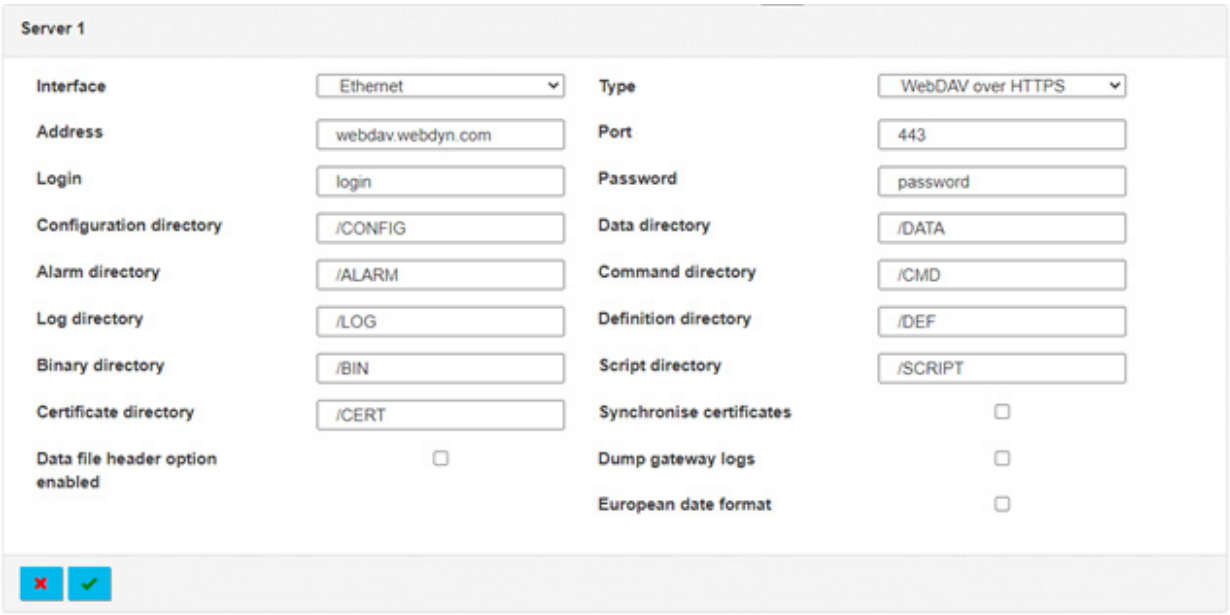
Login	FTP_Login FTP2_Login	Username used by the hub to connect to the remote FTP/SFTP server
Password	FTP_Password FTP2_Password	Password used by the hub to connect to the remote FTP/SFTP server
2 steps for put file disabled	FTP_TwoStepsSendingDisabled FTP2_TwoStepsSendingDisabled	Choice of file transfer in 2 steps via a temporary file until the file is complete on the remote server: • Checked: disabled • Unchecked: enabled
Configuration directory	FTP_DirConfig FTP2_DirConfig	Directory of configuration files on the FTP/SFTP server
Data directory	FTP_DirData FTP2_DirData	Directory of data files on the FTP/SFTP server
Alarm directory	FTP_DirAlarm FTP2_DirAlarm	Directory of alarm files on the FTP/SFTP server
Command directory	FTP_DirCmd FTP2_DirCmd	Directory of command files on the FTP/SFTP server
Log directory	FTP_DirLog FTP2_DirLog	Directory of log files on the FTP/SFTP server
Definition directory	FTP_DirDef FTP2_DirDef	Directory of definition files on the FTP/SFTP server
Binary directory	FTP_DirBin FTP2_DirBin	Directory of update files on the FTP/SFTP server
Script directory	FTP_DirScript FTP2_DirScript	Directory of script files on the FTP/SFTP server
Certificate directory	FTP_DirCert FTP2_DirCert	Directory of certificate files on the FTP/SFTP server
Synchronise certificates	FTP_SynchroniseCertificates FTP2_SynchroniseCertificates	Choice of synchronization of certificates: • Checked: Enable certificate synchronization • Unchecked: No synchronization of certificates

Enable data file header option	FTP_HeaderOption FTP2_HeaderOption	Choice of optional headers in the data files uploaded to the FTP/SFTP server: • Checked: With optional headers • Unchecked: Without optional headers
Enable advanced data option	FTP_EnableAdvancedData FTP2_EnableAdvancedData	Addition of the number of complete readings over this acquisition period in the data files deposited on the FTP/SFTP server: • Checked: Addition of the number of complete readings • Unchecked: No addition of the number of complete readings
Dump gateway logs	FTP_UploadLog FTP2_UploadLog	Choice of depositing system log files on the FTP/SFTP server: • Checked: Filing of system log files on a schedule. • Unchecked: No deposit of system log files on a schedule. System logs are systematically filed on a manual action.
European date format	FTP_EuroDateFormat FTP2_EuroDateFormat	Choice of timestamp type for data deposited on the FTP/SFTP server: • Checked: European format (DD/MM/YY-HH:MM:SS) • Unchecked: ISO format (YY/MM/DD-HH:MM:SS)
Enable Web Services	FTP_WebServicesEnable FTP2_WebServicesEnable	Activation of web services associated with FTP actions: • Checked: web services are enabled • Unchecked: web services are not enabled
Web Services URL	FTP_WebServicesUrl FTP2_WebServicesUrl	URL to call when FTP actions have been performed and web services are enabled

The directory structure on the remote FTP/SFTP server must be created before any connection. (see chapter 4.1: “The FTP/SFTP/ server”).

3.2.2.5.3 WebDAV over HTTPS

The WevDAV over HTTPS server is a secure server with identification with a username and password.



The parameters on the 2 servers are:

WEB INTERFACE	PARAMETER <uid>_config.ini configuration file	DESCRIPTION
Interface	SERVER_Interface SERVER2_Interface	Choice of the network interface to be used by the server: • Ethernet (see chapter 3.2.2.3: “Networks”) • Modem (see chapter 3.2.2.4: “Modem”)
Type	SERVER_Type SERVER2_Type	Choice of server protocol: • WebDAV over HTTPS
Address	SERVER_Address SERVER2_Address	IP address or server name
Port	HTTP_Port HTTP2_Port	WebDAV server port
Login	HTTP_Login HTTP2_Login	Username used by the hub to connect to the remote WebDAV server

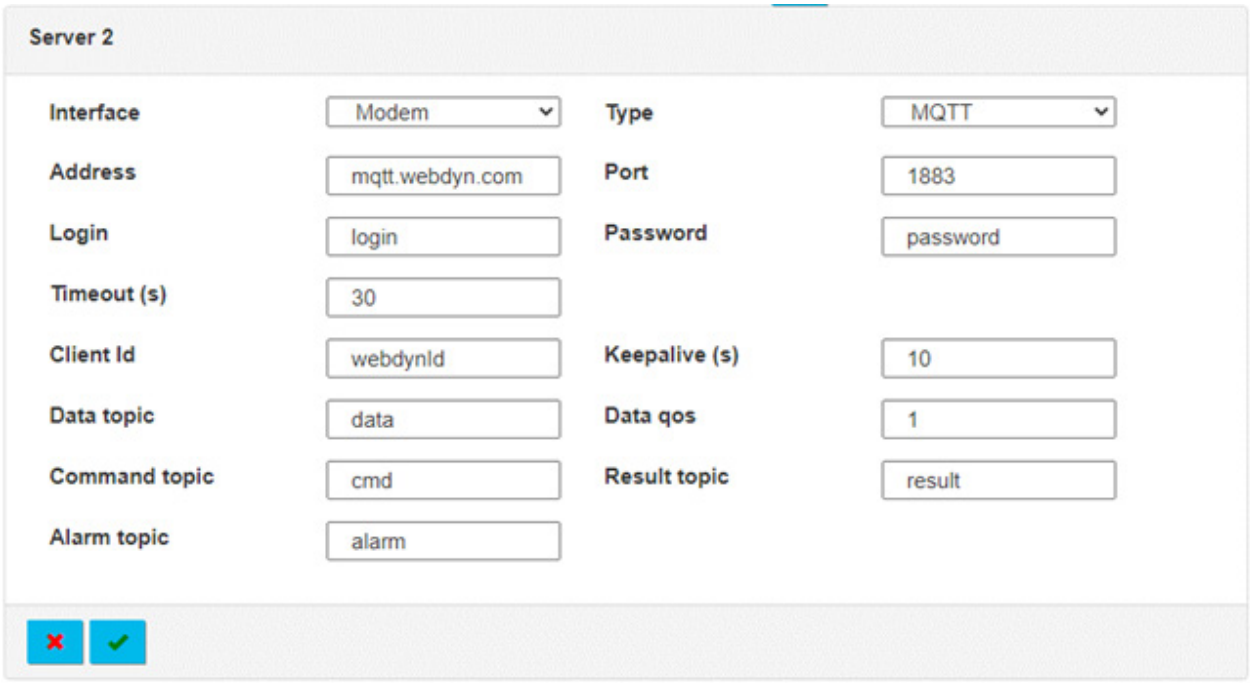
Password	HTTP_Password HTTP2_Password	Password used by the hub to connect to the remote WebDAV server
Configuration directory	HTTP_DirConfig HTTP2_DirConfig	Directory of configuration files on the WebDAV server
Data directory	HTTP_DirData HTTP2_DirData	Directory of data files on the WebDAV server
Alarm directory	HTTP_DirAlarm HTTP2_DirAlarm	Directory of alarm files on the WebDAV server
Command directory	HTTP_DirCmd HTTP2_DirCmd	Directory of command files on the WebDAV server
Log directory	HTTP_DirLog HTTP2_DirLog	Directory of log files on the WebDAV server
Definition directory	HTTP_DirDef HTTP2_DirDef	Directory of definition files on the WebDAV server
Binary directory	HTTP_DirBin HTTP2_DirBin	Directory of update files on the WebDAV server
Script directory	HTTP_DirScript HTTP2_DirScript	Directory of script files on the server
Certificate directory	HTTP_DirCert HTTP2_DirCert	Directory of certificate files on the WebDAV server
Synchronise certificates	HTTP_SynchroniseCertificates HTTP2_SynchroniseCertificates	Choice of synchronization of certificates: • Checked: Enable certificate synchronization • Unchecked: No synchronization of certificates
Data file header option enabled	HTTP_HeaderOption HTTP2_HeaderOption	Choice of optional headers in the data files uploaded to the WebDAV server: • Checked: With optional headers • Unchecked: Without optional headers

Dump gateway logs	HTTP_UploadLog HTTP2_UploadLog	Choice of depositing system log files on the WebDAV server: • Checked: Filing of system log files on a schedule. • Unchecked: No deposit of system log files on a schedule. System logs are systematically filed on a manual action.
European date format	HTTP_EuroDateFormat HTTP2_EuroDateFormat	Choice of timestamp type for data deposited on the WebDAV server: • Checked: European format (DD/MM/YY-HH:MM:SS) • Unchecked: ISO Format (YY/MM/DD-HH:MM:SS)
Enable Web Services	FTP_WebServicesEnable FTP2_WebServicesEnable	Activation of web services associated with FTP actions: • Checked: web services are enabled • Unchecked: web services are not enabled See the “Web Services” section for more details on how it works.
Web Services URL	FTP_WebServicesUrl FTP2_WebServicesUrl	URL to call when FTP actions have been performed and web services are enabled. See the “Web Services” section for more details on how it works.

The directory structure on the remote WebDAV-HTTPS server must be created before any connection. (see chapter 4.1: “The FTP/SFTP/ server”).

3.2.2.5.4 MQTT

The MQTT server is an insecure server with an identification with a username and password.



Server 2 settings are:

WEB INTERFACE	PARAMETER <uid>_config.ini configuration file	DESCRIPTION
Interface	SERVER2_Interface	Choice of the network interface to be used by the server: • Ethernet (see chapter 3.2.2.3: “Networks”) • Modem (see chapter 3.2.2.4: “Modem”)
Type	SERVER2_Type	Choice of server protocol: • MQTT: MQTT server
Address	SERVER2_Address	IP address or server name
Port	MQTT2_Port	MQTT server port (default 1883)
Login	MQTT2_Login	Username used by the hub to connect to the MQTT server

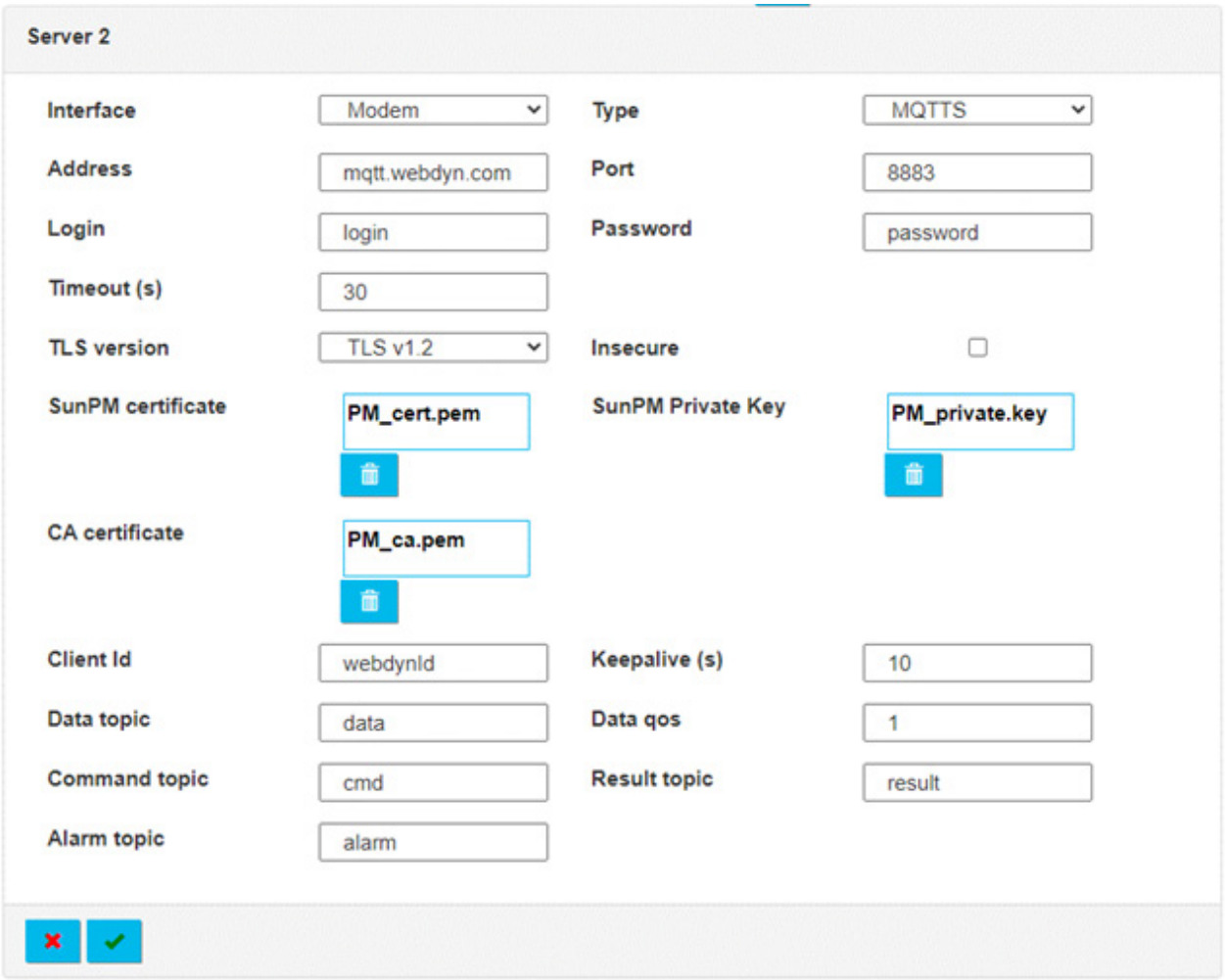
Password	MQTT2_Password	Password used by the hub to connect to the MQTT server
Timeout (s)	MQTT2_Timeout	Maximum wait time in seconds for the response from the MQTT server. If the server has not responded within the allotted time, the send is stopped and retried on the next schedule. Functional only in QoS 1 or QoS 2.
Client Id	MQTT2_ClientId	Customizable identifier of the equipment on the MQTT server. This parameter is to be retrieved from your MQTT server.
Keepalive (s)	MQTT2_KeepAlive	If no exchange made with the MQTT server for the time defined in seconds, the concentrator sends a ping to the MQTT server in order to verify the connection to it. If the value is “0”, KeepAlive is disabled. If the hub is in persistent connection mode with the MQTT server and a disconnection is detected after a KeepAlive, the hub will automatically reconnect to the MQTT server.
Data topic	MQTT2_Topic	Name of the topic for the data deposited by the concentrator.
Data qos	MQTT2_QoS	Guaranteed service number for sending messages (Quality Of Service). The possible values are: • 0: The message will be delivered at most once, i.e. with no guarantee of reception. • 1: The message will be delivered at least once, i.e. the concentrator will transmit several times if necessary until the broker confirms that it has been transmitted. • 2: The message will necessarily be saved by the concentrator and will always send it as long as the broker does not confirm its sending. (avoids duplication of messages)

Command topic	MQTT2_ControlTopic	Name of the topic for the commands to be retrieved by the concentrator. The MQTT2_ResultTopic parameter must be populated for using commands. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTT server.
Result topic	MQTT2_ResultTopic	Name of the topic for the results of the commands passed to the concentrator. The MQTT2_ControlTopic parameter must be populated for using commands. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTT server.
Alarm topic	MQTT2_AlarmTopic	Name of the alarm topic that you want to publish. If the field is empty, no alarm will be published to the broker. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTT server.
Enable advanced data option	MQTT2_EnableAdvancedData	Publication of the number of complete readings over this acquisition period in the data topic. Possible values are: • 0: Disabled • 1: Enabled

MQTT is only available on server 2 (backup).

3.2.2.5.5 MQTTS

The MQTTS server is a secure server with identification with a username and password. It is necessary to import certificates and a private key in order to secure the connection between the concentrator and the MQTTS server.



Server 2 settings are:

WEB INTERFACE	PARAMETER <uid>_config.ini configuration file	DESCRIPTION
Interface	SERVER2_Interface	Choice of the network interface to be used by the server: • Ethernet (see chapter 3.2.2.3: “Networks”) • Modem (see chapter 3.2.2.4: “Modem”)

Type	SERVER2_Type	Choice of server protocol: • MQTTS: secure MQTT server
Address	SERVER2_Address	IP address or server name
Port	MQTT2_Port	MQTTS server port (default 8883)
Login	MQTT2_Login	Username used by the hub to connect to the MQTTS server
Password	MQTT2_Password	Password used by the concentrator to connect to the MQTTS server
Timeout (s)	MQTT2_Timeout	Maximum wait time in seconds for the response from the MQTTS server. If the server has not responded within the allotted time, the send is stopped and retried on the next schedule.
TLS version	MQTT2_TlsVersion	TLS version supported by the MQTTS server. The possible values are: • TLS v1.1 • TLS v1.2
Insecure	MQTT2_Insecure	Disable verification of the host name specified in certificates. The possible values are: • Unchecked: Verification enabled • Checked: Verification disabled.
SunPM certificate	MQTT2_CertFile	Name of the hub-specific certificate used for the connection. The certificate is to be retrieved from your MQTTS server and must be imported to the concentrator by FTP or by the web interface.
SunPM Private Key	MQTT2_KeyFile	Name of the file including the private key specific to the concentrator used for the connection. The file is to be retrieved from your MQTTS server and must be imported to the concentrator by FTP or by the web interface.

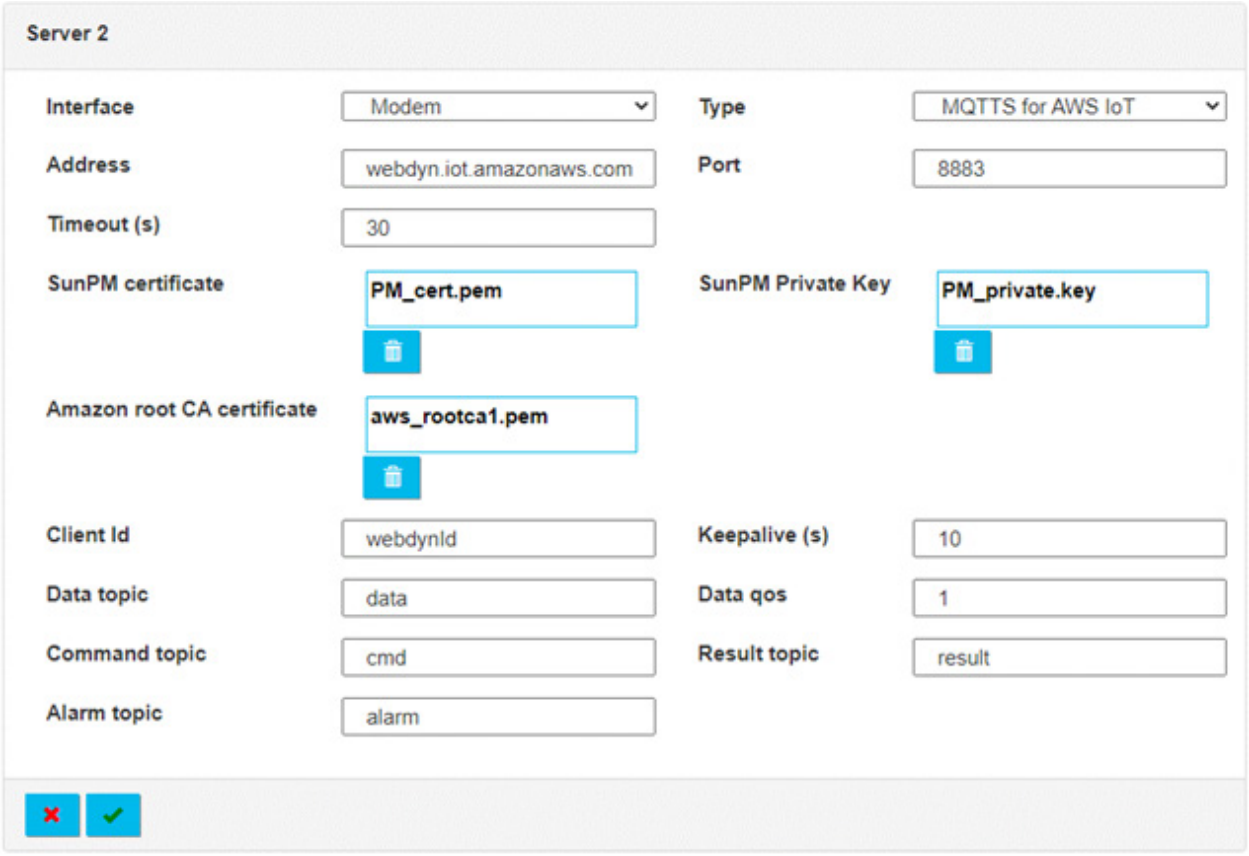
CA certificate	MQTT2_CaCertFile	Name of the certificate used to authenticate the specified MQTTS server. The certificate is to be retrieved from your MQTTS server and must be imported to the concentrator by FTP or by the web interface.
Client Id	MQTT2_ClientId	Customizable identifier of the equipment on the MQTTS server. This parameter is to be retrieved from your MQTT server.
Keepalive (s)	MQTT2_KeepAlive	If no exchange has been made with the MQTTS server for the time defined in seconds, the concentrator sends a ping to the MQTTS server in order to verify the connection to it. If the value is “0”, KeepAlive is disabled. If the hub is in permanent connection mode with the MQTTS server and a disconnection is detected after a KeepAlive, the hub will automatically reconnect to the MQTTS server.
Client Id	MQTT2_Topic	Name of the topic for the data deposited by the concentrator.
Data qos	MQTT2_QoS	Guaranteed service number for sending messages (Quality Of Service). The possible values are: • 0: The message will be sent only once, i.e. with no guarantee of receipt. • 1: The message will be sent at least once, i.e. the concentrator will send several times if necessary until the broker confirms that it has been sent. • 2: The message will necessarily be saved by the concentrator and will always send it as long as the broker does not confirm its sending. (avoids duplication of messages)

Command topic	MQTT2_ControlTopic	Name of the topic for the commands to be retrieved by the concentrator. The MQTT2_ResultTopic parameter must be populated for using commands. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTTS server.
Result topic	MQTT2_ResultTopic	Name of the topic for the results of the commands passed to the concentrator. The MQTT2_ControlTopic parameter must be populated for using commands. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTTS server.
Alarm topic	MQTT2_AlarmTopic	Name of the alarm topic to be published. If the field is empty, no alarm will be published to the broker. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTTS server.

MQTTS is only available on server 2 (backup).

3.2.2.5.6 MQTTS AWS IoT

The MQTTS AWS IoT server is a secure Amazon server with certificate identification. It is necessary to import certificates and a private key into the hub.



Server 2 settings are:

WEB INTERFACE	PARAMETER <uid>_config.ini configuration file	DESCRIPTION
Interface	SERVER2_Interface	Choice of the network interface to be used by the server: • Ethernet (see chapter 3.2.2.3: “Networks”) • Modem (see chapter 3.2.2.4: “Modem”)
Type	SERVER2_Type	Choice of server protocol: • MQTTS for AWS IoT: MQTTS server on “AWS IoT”
Address	SERVER2_Address	IP address or server name

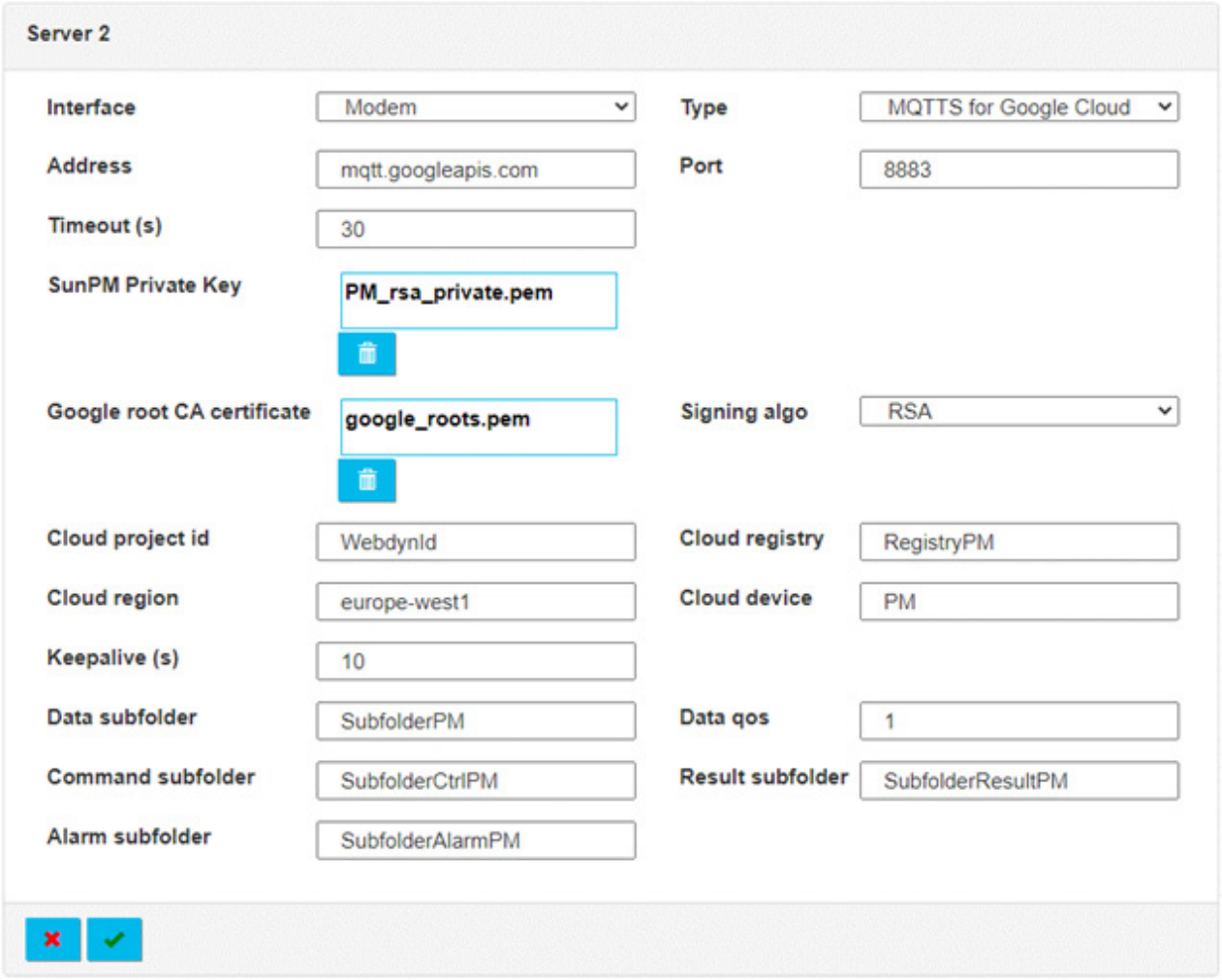
Port	MQTT2_Port	MQTT server port (default 8883)
Timeout (s)	MQTT2_Timeout	Maximum wait time in seconds for the response from the MQTT server. If the server has not responded within the allotted time, the send is stopped and retried on the next schedule. Functional only in QoS 1 or QoS 2.
SunPM certificate	MQTT2_CertFile	Name of the hub-specific certificate used for the connection. The certificate is to be retrieved from your MQTTS server and must be imported to the concentrator by FTP or by the web interface.
SunPM Private Key	MQTT2_KeyFile	Name of the file including the private key specific to the concentrator used for the connection. The file is to be retrieved from your MQTTS server and must be imported to the concentrator by FTP or by the web interface.
Amazon root CA certificate	MQTT2_CaCertFile	Name of the certificate used to authenticate the specified MQTTS server. The certificate is to be retrieved from your MQTTS server and must be imported to the concentrator by FTP or by the web interface.
Client Id	MQTT2_ClientId	Customizable identifier of the equipment on the MQTT server.
Keepalive (s)	MQTT2_KeepAlive	If no exchange made with the MQTT server for the time defined in seconds, the concentrator sends a ping to the MQTT server in order to verify the connection to it. If the value is “0”, KeepAlive is disabled. If the hub is in persistent connection mode with the MQTT server and a disconnection is detected after a KeepAlive, the hub will automatically reconnect to the MQTT server.
Data topic	MQTT2_Topic	Name of the topic for the data deposited by the concentrator.

Data qos	MQTT2_QoS	Guaranteed service number for sending messages (Quality Of Service).The possible values are: • 0: The message will be sent only once, i.e. with no guarantee of receipt. • 1: The message will be sent at least once, i.e. the concentrator will send several times if necessary until the broker confirms that it has been sent. • 2: Not managed by the AWS IoT MQTTS server
Command topic	MQTT2_ControlTopic	Name of the topic for the commands to be retrieved by the concentrator. The MQTT2_ResultTopic parameter must be populated for using commands. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTTS server.
Result topic	MQTT2_ResultTopic	Name of the topic for the results of the commands passed to the concentrator. The MQTT2_ControlTopic parameter must be populated for using commands. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTTS server.
Alarm topic	MQTT2_AlarmTopic	Name of the alarm topic to be published. If the field is empty, no alarm will be published to the broker. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTTS server.
Enable advanced data option	MQTT2_EnableAdvancedData	Publication of the number of complete readings over this acquisition period in the data topic. The possible values are: • 0: Disabled • 1: Enabled

AWS IoT MQTTS is only available on server 2 (backup).

3.2.2.5.7 MQTTS Google Cloud IoT

The Google Cloud IoT MQTTS server is a secure Google server with certificate identification. It is necessary to import a certificate and a private key into the concentrator.



Server 2 settings are:

WEB INTERFACE	PARAMETER <uid>_config.ini configuration file	DESCRIPTION
Interface	SERVER2_Interface	Choice of the network interface to be used by the server: • Ethernet (see chapter 3.2.2.3: “Networks”) • Modem (see chapter 3.2.2.4: “Modem”)

Type	SERVER2_Type	Choice of server protocol: • MQTTS for Google Cloud IoT: MQTTS server on “Google Cloud IoT”
Address	SERVER2_Address	IP address or server name
Port	MQTT2_Port	MQTT server port (default 8883)
Timeout (s)	MQTT2_Timeout	Maximum wait time in seconds for the response from the MQTT server. If the server has not responded within the allotted time, the send is stopped and retried on the next schedule. Functional only in QoS 1 or QoS 2.
SunPM Private Key	MQTT2_KeyFile	Name of the file including the private key specific to the concentrator used for the connection. The file is to be retrieved from your MQTTS server and must be imported to the concentrator by FTP or by the web interface.
Google root CA certificate	MQTT2_CaCertFile	Name of the certificate used to authenticate the specified MQTTS server. The certificate is to be retrieved from your MQTTS server and must be imported to the concentrator by FTP or by the web interface.
Signing algo	MQTT2_CloudSigningAlgo	Type of key used to verify the signature of the MQTTS server certificate. This parameter is to be retrieved from your Google IoT Cloud server. The possible values are: • “RSA” for the RSA key • “Elliptic Curve” for the elliptical curve key
Cloud project Id	MQTT2_CloudProjectId	Customizable unique identifier of the project defined on the MQTT server. This parameter is to be retrieved from your MQTT server and corresponds to “projectId” on Google Cloud IoT.

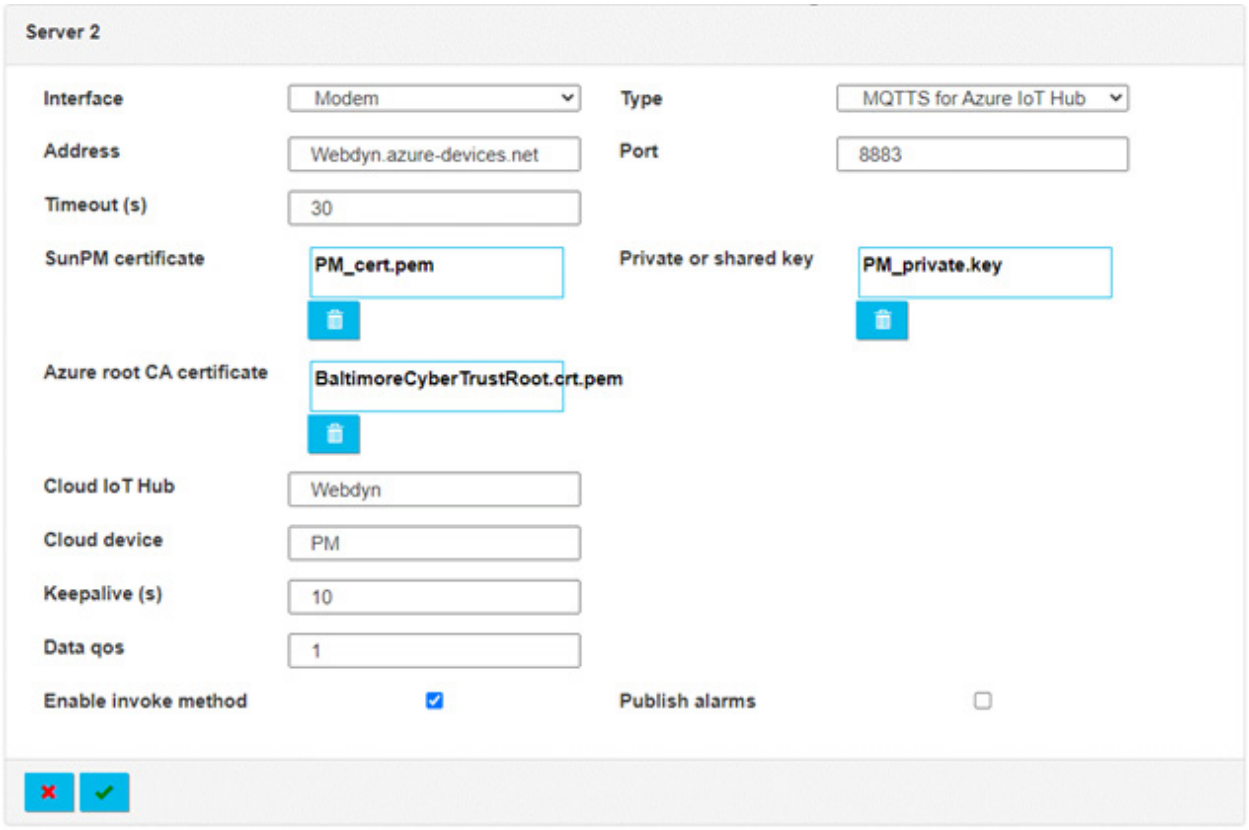
Cloud registry	MQTT2_CloudRegistry	Customizable registry name defined on the MQTT server. This parameter is to be retrieved from your MQTT server and corresponds to “deviceRegistryId” on Google Cloud IoT.
Cloud region	MQTT2_CloudRegion	Device registry MQTT server region. This parameter is to be retrieved from your MQTT server and corresponds to “deviceRegistryLocation” on Google IoT Cloud. For example: “europe-west1”
Cloud device	MQTT2_CloudDevice	Customizable unique identifier of the equipment in a register defined on the MQTT server. This parameter is to be retrieved from your MQTT server and corresponds to “deviceId” on Google IoT Cloud.
Keepalive (s)	MQTT2_KeepAlive	If no exchange made with the MQTT server for the time defined in seconds, the concentrator sends a ping to the MQTT server in order to verify the connection to it. If the value is “0”, KeepAlive is disabled. If the hub is in persistent connection mode with the MQTT server and a disconnection is detected after a KeepAlive, the hub will automatically reconnect to the MQTT server.
Data subfolder	MQTT2_Topic	Name of the topic for the data deposited by the concentrator.
Data qos	MQTT2_QoS	Guaranteed service number for sending messages (Quality Of Service).The possible values are: • 0: The message will be sent only once, i.e. with no guarantee of receipt. • 1: The message will be sent at least once, i.e. the concentrator will send several times if necessary until the broker confirms that it has been sent. •2: Not managed by the Google Cloud IoT MQTTS server.

Command subfolder	MQTT2_ControlTopic	Name of the topic for the commands to be retrieved by the concentrator. The MQTT2_ResultTopic parameter must be populated for using commands. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTTS server.
Result subfolder	MQTT2_ResultTopic	Name of the topic for the results of the commands passed to the concentrator. The MQTT2_ControlTopic parameter must be populated for using commands. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTTS server.
Alarm subfolder	MQTT2_AlarmTopic	Name of the alarm topic to be published. If the field is empty, no alarm will be published to the broker. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTTS server.
Enable advanced data option	MQTT2_EnableAdvancedData	Publication of the number of complete readings over this acquisition period in the data topic. The possible values are: • 0: Disabled • 1: Enabled

The Google Cloud IoT MQTTS is only available on server 2 (backup).

3.2.2.5.8 MQTTS Azure IoT Hub

The Azure IoT Hub MQTTS server is a Microsoft secure server with certificate identification. It is necessary to import a certificate and a private key into the concentrator.



Server 2 settings are:

WEB INTERFACE	PARAMETER <uid>_config.ini configuration file	DESCRIPTION
Interface	SERVER2_Interface	Choice of the network interface to be used by the server: • Ethernet (see chapter 3.2.2.3: “Networks”) • Modem (see chapter 3.2.2.4: “Modem”)
Type	SERVER2_Type	Choice of server protocol: • MQTTS for Azure IoT Hub: MQTTS server on “Azure IoT Hub”
Address	SERVER2_Address	IP address or server name
Port	MQTT2_Port	MQTT server port (default 8883)

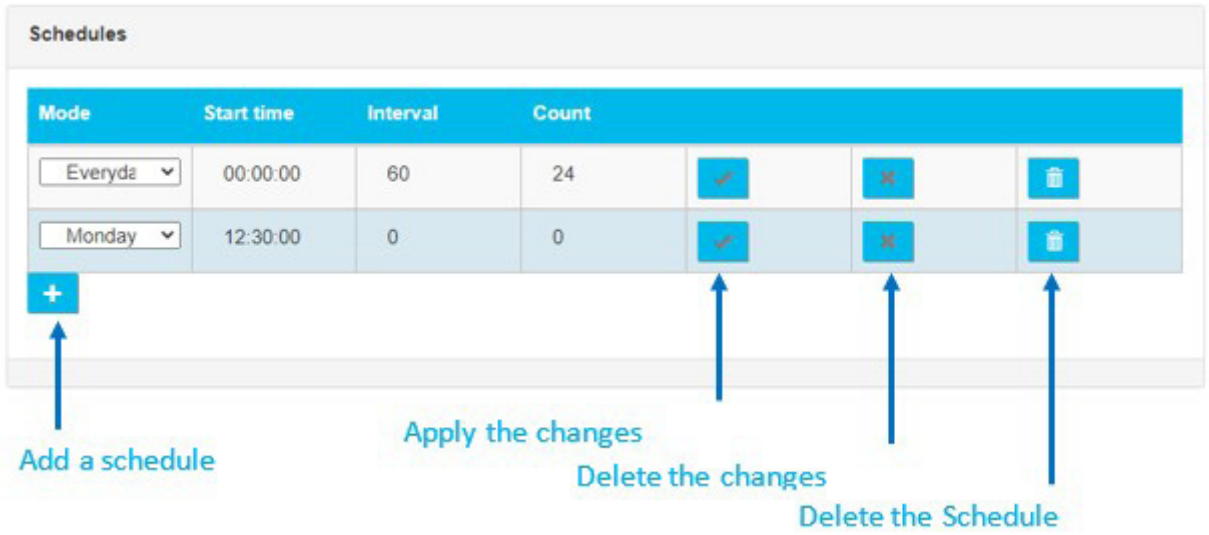
Timeout (s)	MQTT2_Timeout	Maximum wait time in seconds for the response from the MQTT server. If the server has not responded within the allotted time, the send is stopped and retried on the next schedule. Functional only in QoS 1 or QoS 2.
SunPM certificate	MQTT2_CertFile	Name of the hub-specific certificate used for the connection. The certificate is to be retrieved from your MQTTS server and must be imported to the concentrator by FTP or by the web interface.
Private or shared key	MQTT2_KeyFile	Name of the file containing the specific private key or shared key to the concentrator used for the connection. The file is to be retrieved from your MQTTS server and must be imported to the concentrator by FTP or by the web interface.
Azure root CA certificate	MQTT2_CaCertFile	Name of the certificate used to authenticate the specified MQTTS server. The certificate is to be retrieved from your MQTTS server and must be imported to the concentrator by FTP or by the web interface.
Cloud IoT Hub	MQTT2_CloudProjectId	Customizable unique identifier of the project defined on the MQTT server. This parameter is to be retrieved from your MQTT server and corresponds to “Iot Hub” on Azure IoT Hub.
Cloud device	MQTT2_CloudDevice	Customizable unique identifier of the equipment in a register defined on the MQTT server. This parameter is to be retrieved from your MQTT server and corresponds to “device_id” on Azure IoT Hub.

Keepalive (s)	MQTT2_KeepAlive	If no exchange made with the MQTT server for the time defined in seconds, the concentrator sends a ping to the MQTT server in order to verify the connection to it. If the value is “0”, KeepAlive is disabled. If the hub is in persistent connection mode with the MQTT server and a disconnection is detected after a KeepAlive, the hub will automatically reconnect to the MQTT server.
Data qos	MQTT2_QoS	Guaranteed service number for sending messages (Quality Of Service). The possible values are: • 0: The message will be delivered at most once, i.e. with no guarantee of reception. • 1: The message will be delivered at least once, i.e. the concentrator will transmit several times if necessary until the broker confirms that it has been transmitted. • 2: Not managed by Azure IoT Hub MQTTS Server
Enable invoke method	MQTT2_EnableInvokeMethod	Enable method calling. Allows the use of dedicated topics. The possible values are: • Unchecked: Disables method calls • Checked: Activated method call. The concentrator remains in permanent connection mode with the MQTTS server.
Publish alarms	MQTT2_EnableAlarmPost	Activate the publication of alarms on the dedicated topic. The possible values are: • Unchecked: Disables the publication of alarms • Checked: Activates the publication of alarms. The concentrator remains in permanent connection mode with the MQTTS server.
Enable advanced data option	MQTT2_EnableAdvancedData	Publication of the number of complete readings over this acquisition period in the data topic. The possible values are: • 0: Disabled • 1: Enabled

Azure IoT Hub MQTTS is only available on server 2 (backup).

3.2.2.5.9 Schedules:

It is possible to configure a set of connection “Schedules” for each server



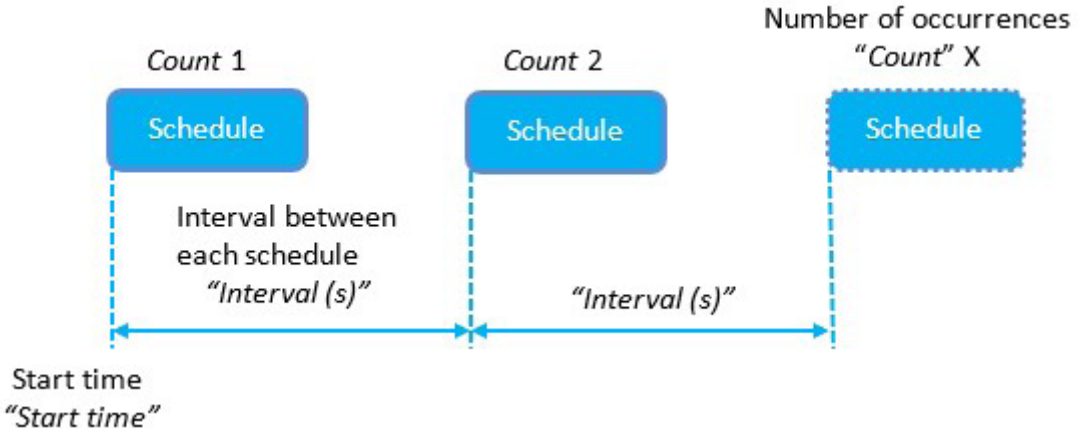
The “Schedule” parameters are:

SCHEDULE	DESCRIPTION
Mode	Schedule type selection: • Everyday: the schedule will be run every day • Monday: the schedule will be run every Monday • Tuesday: the schedule will be run every Tuesday • Wednesday: the schedule will be run every Wednesday • Thursday: the schedule will be run every Thursday • Friday: the schedule will be run every Friday • Saturday: the schedule will be run every Saturday • Sunday: the schedule will be run every Sunday • First day of each month: the schedule will be run on the 1st of every month • 15th of each month: the schedule will be run on the 15th of every month • Last day of each month: the schedule will be run on the last day of every month
Start time	Schedule start time in: “HH:MM:SS”
Interval	Schedule repeat interval in minutes. Special case: the value “0” is automatically switched to “1”

Count	Maximum number of schedules for the day 0 = no schedules

Schedules are only defined for one day, if the settings exceed one day, only the schedules that do not exceed 23H59 will be taken into account.

Every day, the first occurrence is given by the time entered in the “Start time” field. The number of events in the day is given by the “Count” field and the interval between each event by the “Interval” field.



Count: if the schedule is to be triggered all day at regular intervals, you can enter a value of more than “1440” in “Count”.

Example 1:

For a regular file upload every hour, the schedule should be configured as follows:

Mode	Start time	Interval	Count			
Everyday	00:00:00	60	24			

Example 2:

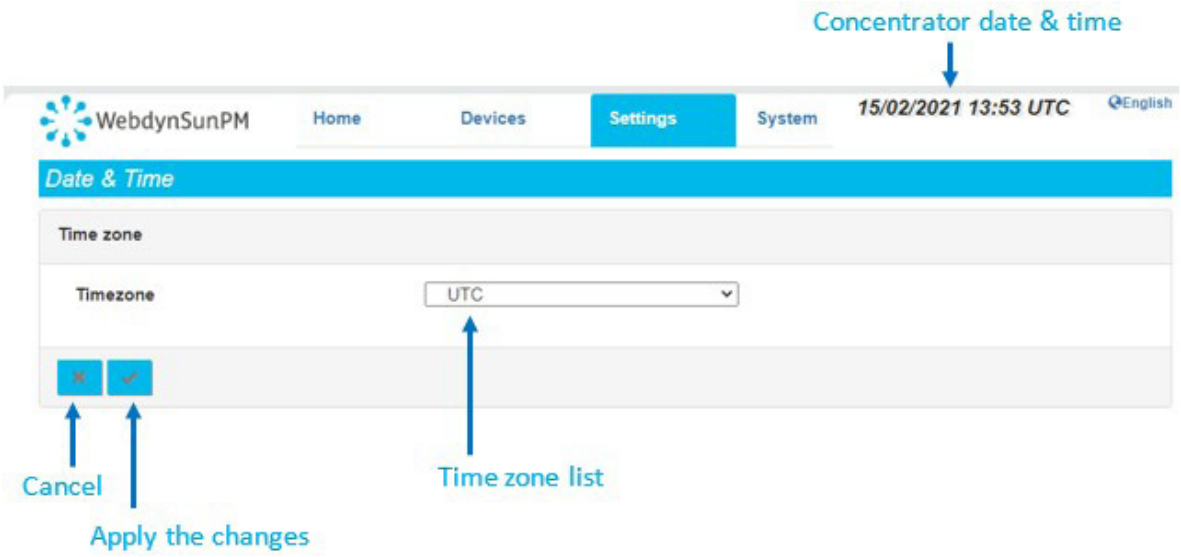
For a regular file upload every Sunday at midday, the schedule should be configured as follows:

Mode	Start time	Interval	Count			
Sunday	12:00:00	1	1			

3.2.2.6 Date & Time

The “Date & Time” part is used to configure the concentrator’s date and time and the NTP servers.

Time Zone:



The list of time zones is available in Appendix B: Time zone list.

Once a time zone is selected, click the “Apply” button to take the new time zone into account. The date and time are updated immediately and the time offset from UTC is indicated on the web interface.

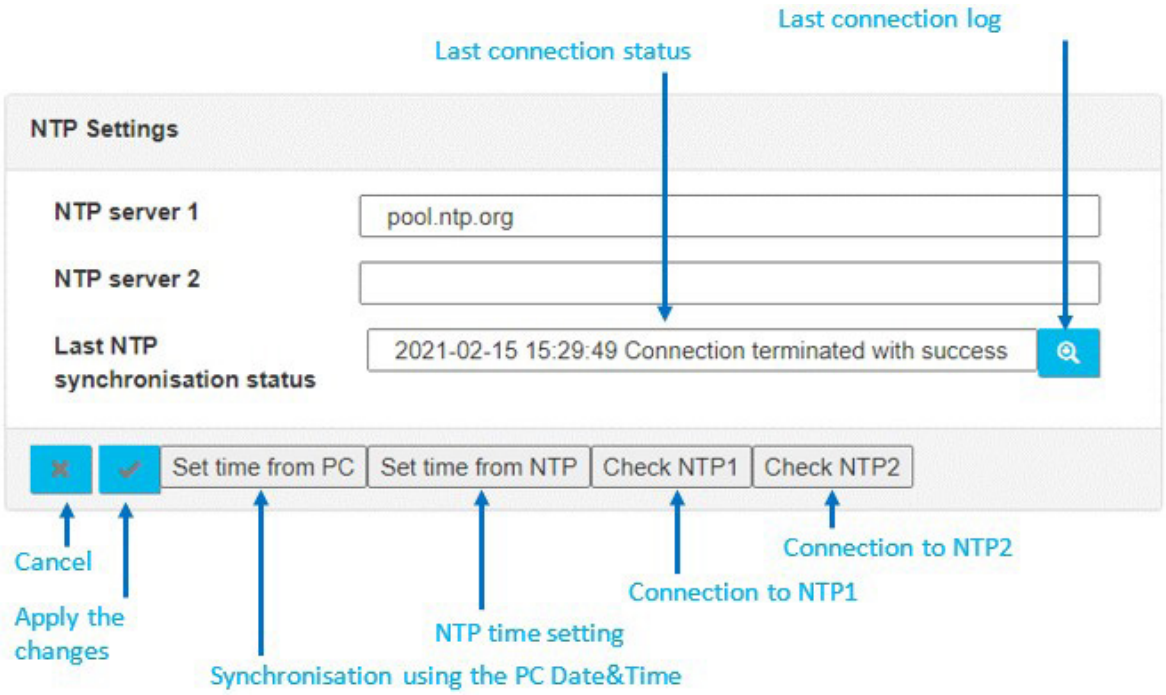
Time changes impact the names of generated files and recorded data which are then uploaded to the remote server by the concentrator. For example, if you select the “(GMT+01:00) Europe: Brussels, Copenhagen, Madrid, Paris” time zone, the concentrator will indicate: 15/02/2021 15:03 UTC+1.

The date and time setting is:

WEB INTERFACE	PARAMETER <uid>_config.ini configuration file	DESCRIPTION
Time zone	NTP_TimeZone	Time zone selection: (GMT-11:00) Midway Island, Samoa (GMT-10:00) Honolulu (GMT-10:00) Tahiti (GMT-09:30) Marquesas (GMT-09:00) Anchorage (GMT-08:00) Pacific Time (US and Canada) (GMT-08:00) Los angeles (GMT-07:00) Denver (GMT-07:00) Chihuahua, La Paz, Mazatlan (GMT-06:00) Guadalajara, Mexico City, Monterrey (GMT-06:00) Chicago, Central America (GMT-05:00) Bogota, Lima, Quito (GMT-05:00) New York (GMT-04:00) Atlantic Time (Canada) (GMT-04:00) Caracas (GMT-04:00) Martinique (GMT-04:00) Guadeloupe (GMT-03:30) Newfoundland, St Johns (GMT-03:00) Antarctica (GMT-03:00) Sao Paulo (GMT-02:00) Brazil (GMT-01:00) Azores UTC (GMT+01:00) Europe: Brussels, Copenhagen, Madrid, Paris (GMT+01:00) Algiers (GMT+02:00) Athens, Bucharest, Istanbul (GMT+02:00) Cairo (GMT+03:00) Moscow, St. Petersburg, Volgograd (GMT+03:00) Kuwait, Riyadh (GMT+04:00) Abu Dhabi, Dubai, Muscat (GMT+04:00) Baku, Tbilisi, Yerevan (GMT+04:30) Kabul (GMT+05:00) Karachi (GMT+05:00) Tashkent (GMT+05:30) Kolkata (GMT+05:45) Katmandu (GMT+06:00) Astana, Dhaka (GMT+06:00) Almaty, Novosibirsk (GMT+06:30) Rangoon, Yangon (GMT+06:30) Cocos (GMT+07:00) Bangkok, Hanoi, Jakarta (GMT+08:00) Beijing, Chongqing, Hong Kong, Urumqi, Shanghai (GMT+08:00) Taipei (GMT+09:00) Osaka, Sapporo, Tokyo (GMT+09:00) Seoul (GMT+09:30) Darwin (GMT+10:00) Brisbane, Sydney (GMT+10:00) Guam, Port Moresby (GMT+10:30) Adelaide (GMT+11:00) Noumea (GMT+11:00) Magadan, Solomon Islands (GMT+13:00) Auckland, Wellington

The time differences for countries are not taken into account by the concentrator.

NTP:



The NTP settings are:

WEB INTERFACE	PARAMETER <uid>_ config.ini configuration file	DESCRIPTION
NTP server 1	NTP_Server1	address for the NTP 1 server used to synchronise the concentrator clock
NTP server 2	NTP_Server2	NTP 2 server address used to synchronise the concentrator clock if server 1 did not respond

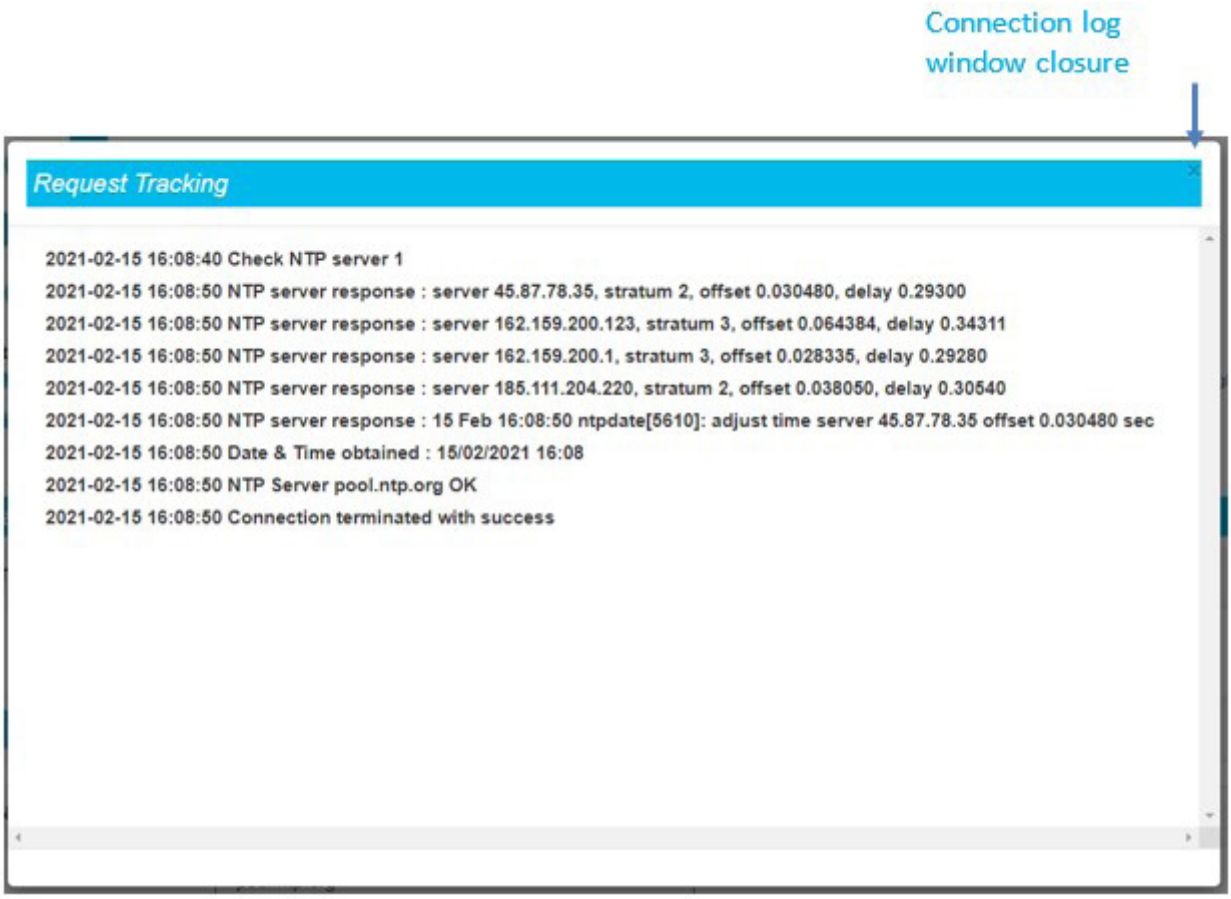
If the NTP1 and NTP2 server values are left blank, the concentrator will not use NTP synchronisation.

By default, the concentrator uses the free “pool.ntp.org” NTP server. This server does not guarantee the exactness of the time synchronisation, nor its robustness. The use of a specific NTP server is strongly recommended. Contact an NTP server portal or supplier.

After having entered an NTP server, it can be tested by clicking one of the following buttons:

- “Set time from NTP”: launches NTP1 synchronisation and, if necessary NTP2 synchronisation and then applies it to the concentrator.
- “Check NTP1”: Tests NTP1 server synchronisation without applying it to the concentrator.
- “Check NTP2”: Tests NTP2 server synchronisation without applying it to the concentrator.

A window opens to display the connection log:

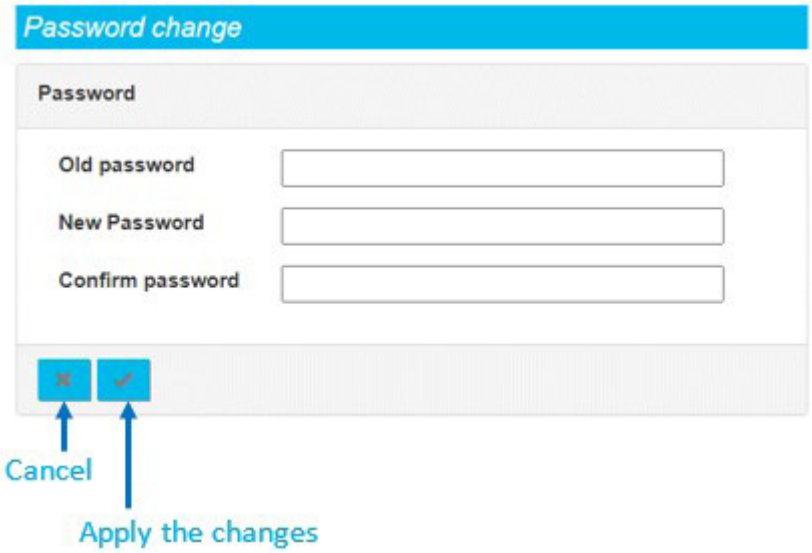


The last line in the log containing the test result is displayed and stored on the main web page. It is also possible to re-display the last connection log by clicking the “Magnifier” button.

The concentrator time can be synchronised with the PC time by clicking the “Set time from PC” button. The time zone configured in the concentrator is then applied.

3.2.2.7 Password

The “password” field is used to modify the password that authorizes access to the web interface.



Follow the steps below to change the password:

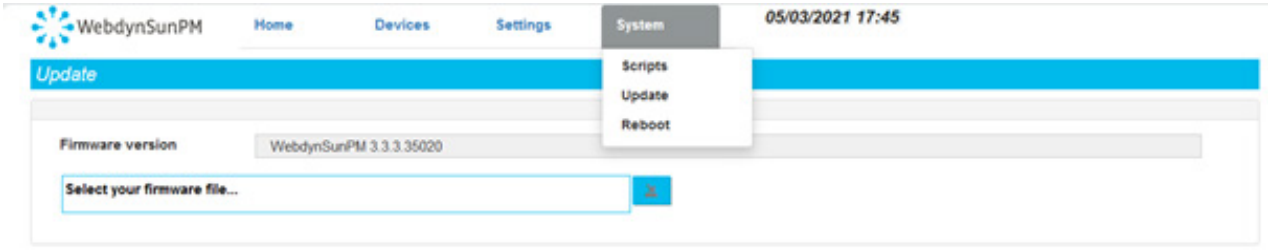
- Enter the current password in the “Old password” field.
- Enter the new password in the “New password” field.
- Enter the new password again in the “Confirm password” field.
- Validate by clicking “Apply”.

To secure access to the concentrator, we recommend changing the default passwords following the first configuration. The password can also be changed using the “WEB_Password” variable in the config file “<uid>_config.ini”.

If you lose the password and no server is configured, you will have to completely reset the concentrator by performing a factory reset via an SMS factory command (see chapter 5.3.3: “factory: Return to factory settings” Error! Return source not found.) or via the “Factory Reset” button (see chapter 2.4.3.2: “Factory Reset button”).

3.2.3 System

All the system settings are grouped together on the “System” tab.



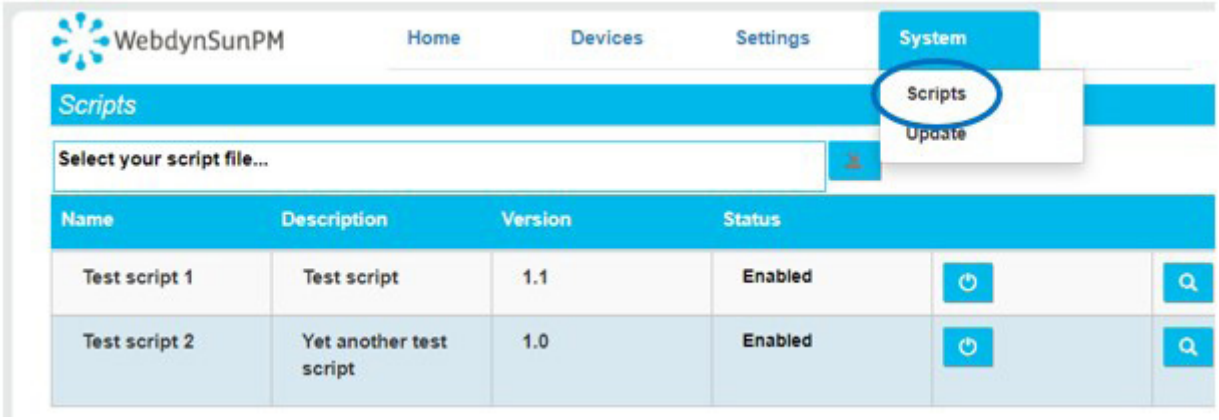
3.2.3.1 Scripts

The WebdynSunPM concentrator has a powerful script based device management and customisation tool.

The tool is based on a LUA command interpreter used to run tasks on the concentrator in the background.

A technical reference guide is available describing all the commands and possibilities of the supplied script language in detail. (“WebdynSunPM LUA User Guide.pdf”)

Script configuration and management are accessed from the local web site page:

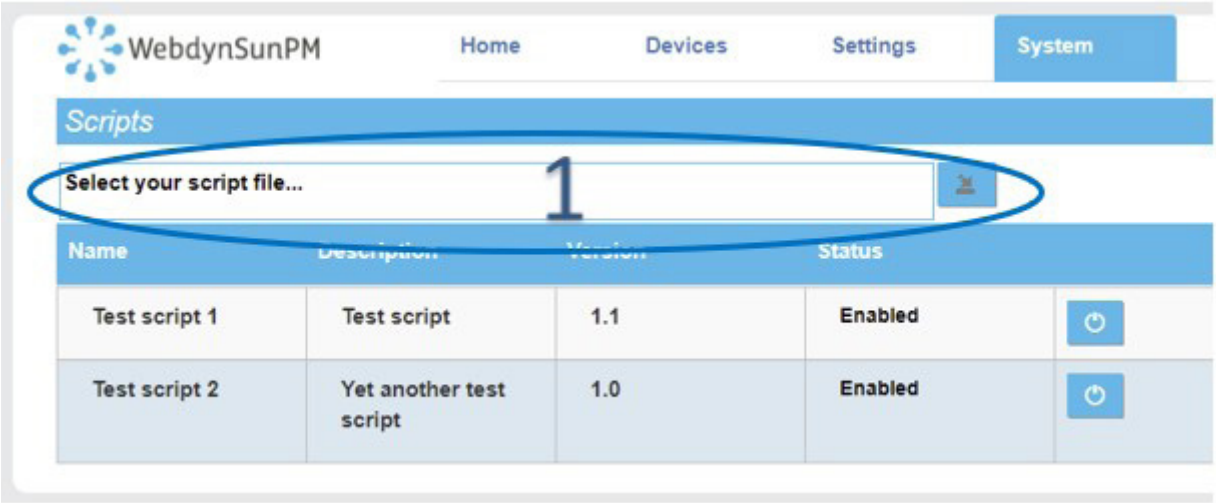


This page can be used to import new scripts, enable them, disable them or even delete them, or to view the run log.

3.2.3.1.1 Importing a Script

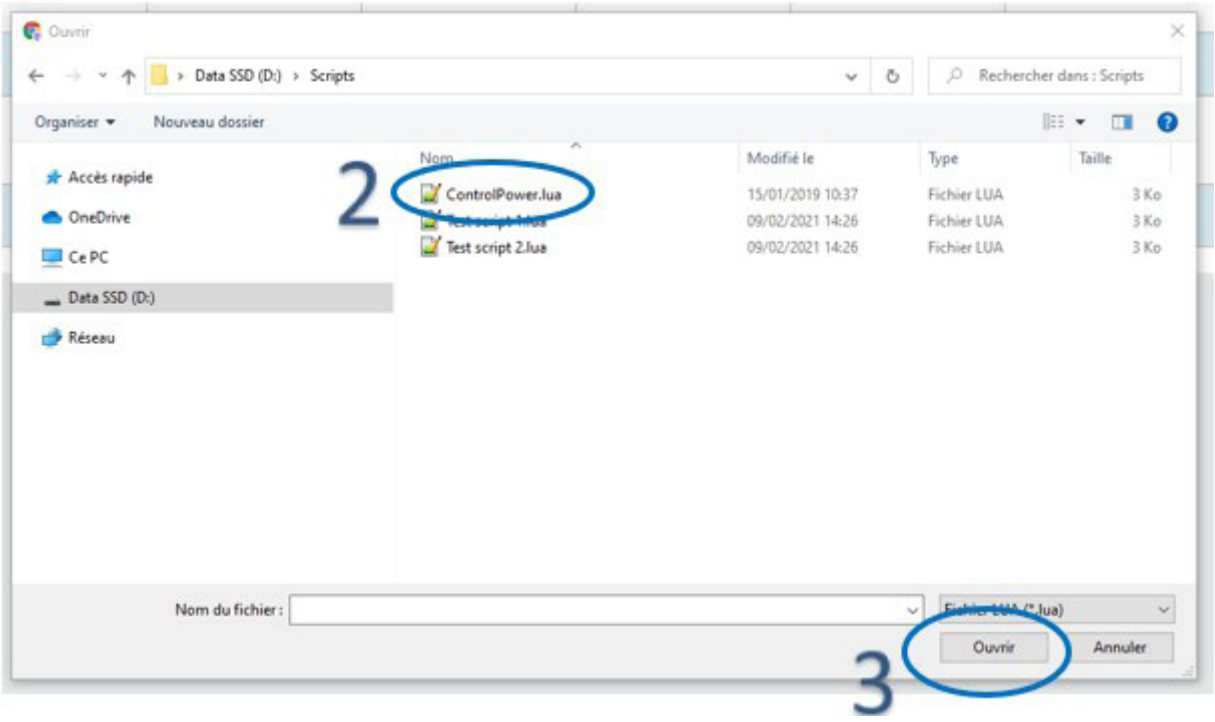
The only way to import a new script into the concentrator is using the local web site.

To do that, click the “Select your script file” zone as shown below.



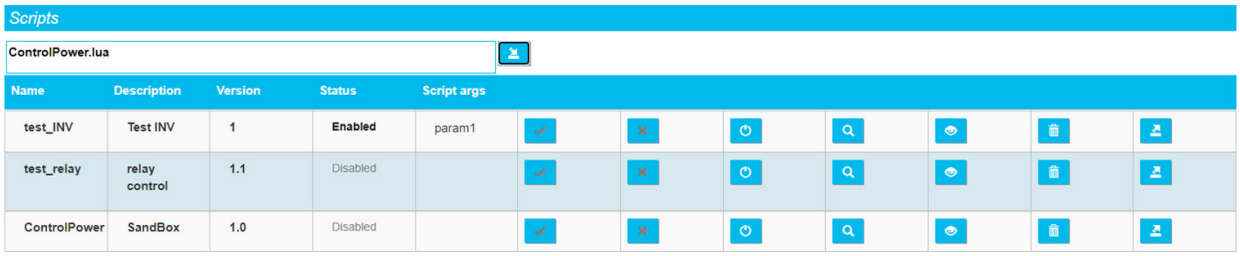
A dialogue window is displayed to select the file to import.

Then select the file to import and click the “Open” button.



Finally, click the transfer button to complete the operation.

Once this 4th step is complete; the script will appear in the script management window.



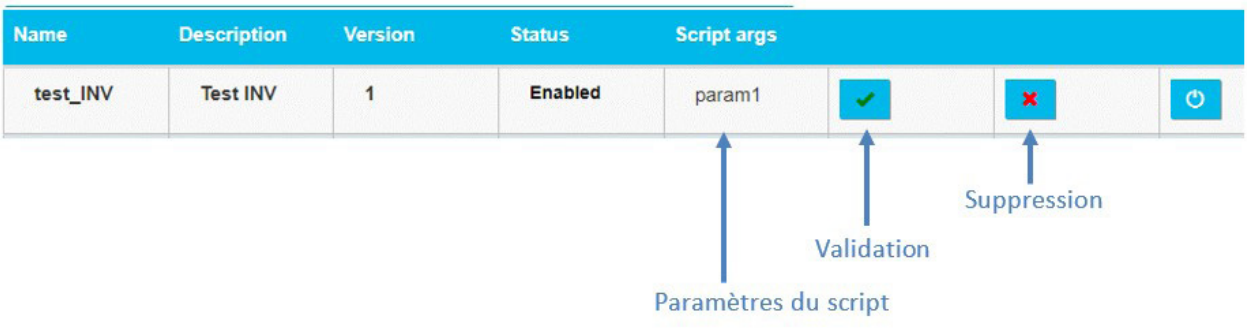
When importing a script in ".luax" format, if the following error message is displayed "error:Error deciphering test_script.luax: stoul", this means that the WebdynSunPM does not have the decryption keys . In this case, they must be sent using the "setKey" command. (See chapter 5.3.14: ""setKey": Adding keys for decrypting client scripts").

Note that the scripts are imported stopped, this means they are not started automatically.

The information displayed on the web page comes from the script, in particular the "header" section. Indeed the "ControlPower.lua" script starts with the following sequence:

```
header = {
      version = 1.1,
      label = "Demo control power"
}
```

We therefore find the description displayed, which comes from the "label" information as well as the version number which comes from the "version" information. See the "WebdynSunPM LUA User Guide.pdf" document for more details.



It is possible to pass parameters to the script, simply fill in the "Script args" field and validate by clicking on the green tick "Validation" so that the script takes them into account when it is activated. To delete parameters, you must click on the red tick "Delete".

3.2.3.1.2 Enabling/Disabling a Script

Enabling a script means starting to run it on the concentrator. In practice, the LUA script “wsinit()” function is run:

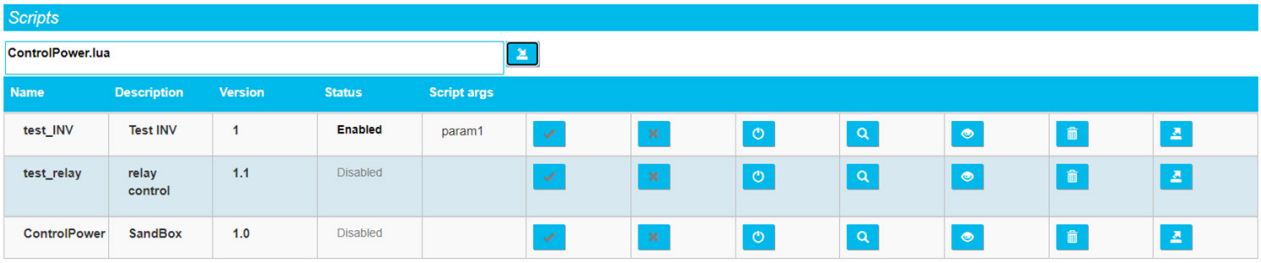
```
function wsInit()
      wd.log(“Control power initialized”)
end
```

See the “WebdynSunPM LUA User Guide.pdf” document for more details of what can be done.

Disabling a script meets stopping it from running.

When the WebdynSunPM is restarted, all scripts return to the same state as before. For example, if the script was started, it will be started.

A specific button is used to enable and disable:



When the script is disabled, its status is greyed out and “Disabled” is displayed.

When the script is enabled, its status is black and “Enabled” is displayed.

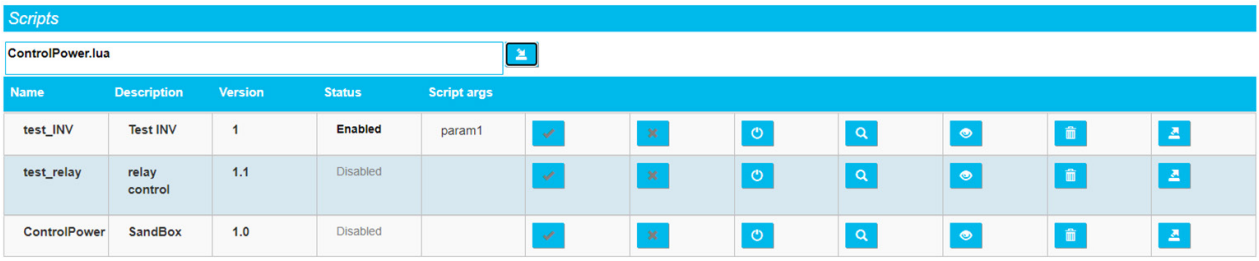
3.2.3.1.3 Viewing the Script Log

Scripts can report information to the end user using the wd.log()” function.

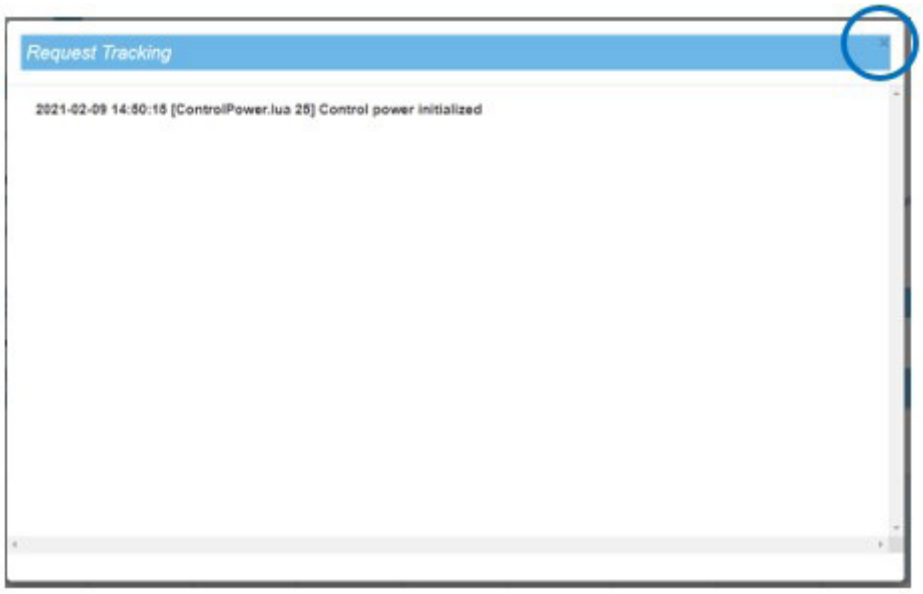
Thus, the following code will display the “Control power initialized” string in the start-up script log file:

```
function wsInit()
      wd.log(“Control power initialized”)
end
```

The log file is displayed by pressing the view button:



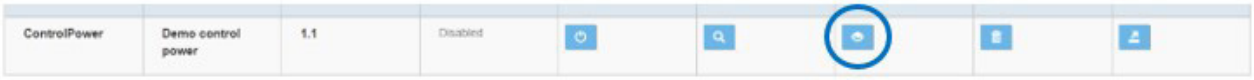
Pressing the button displays the following page:

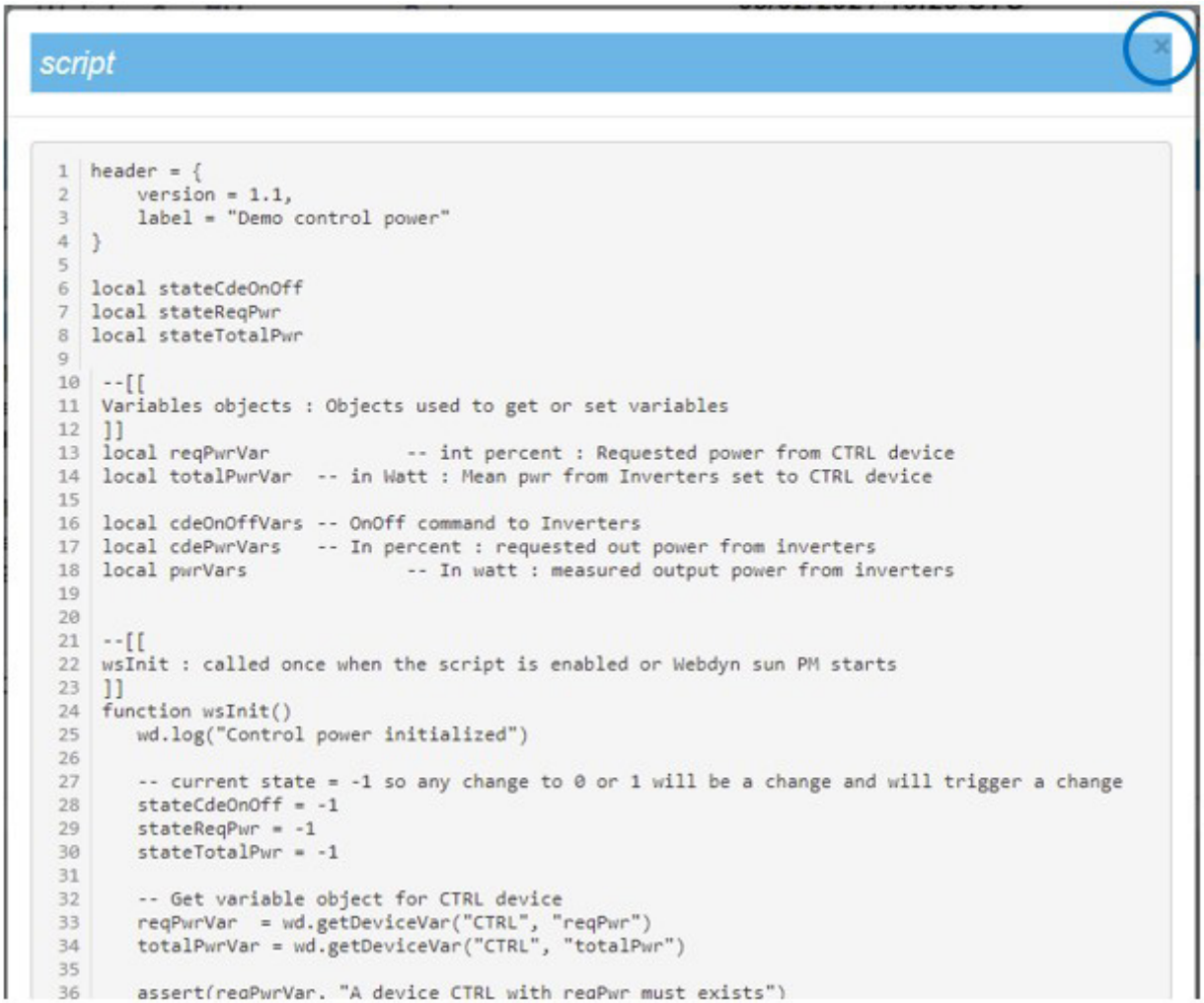


The page is closed by pressing the cross at the top right indicated above in the circle.

3.2.3.1.4 Viewing the Script

It is possible to display the source code for the scripts loaded onto the concentrator by clicking the display button:





When a script source code is displayed, click the cross at the top right of the window to make it disappear.

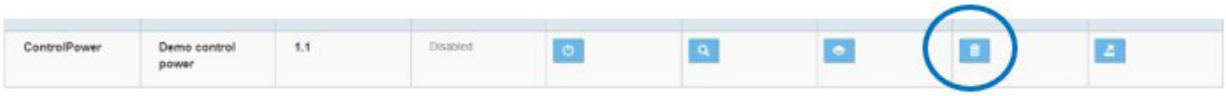
3.2.3.1.5 Exporting a Script

Pressing the script export button launches an immediate local script load by the browser.

When the user clicks this button, the script is loaded directly into the browser downloads directory using the name in the “Name” field followed by the “.lua” suffix. There is no confirmation message.

3.2.3.1.6 Deleting a Script

The delete script button is used to delete loaded scripts.

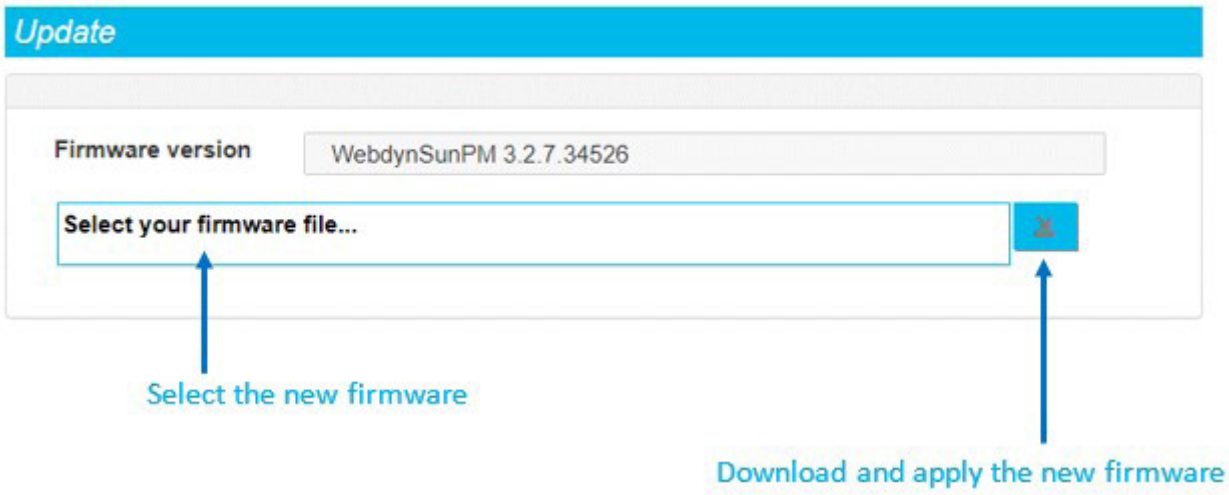


The scripts do not need to be disabled before deleting them.

Note that there is no confirmation message when a script is deleted.

3.2.3.2 Update

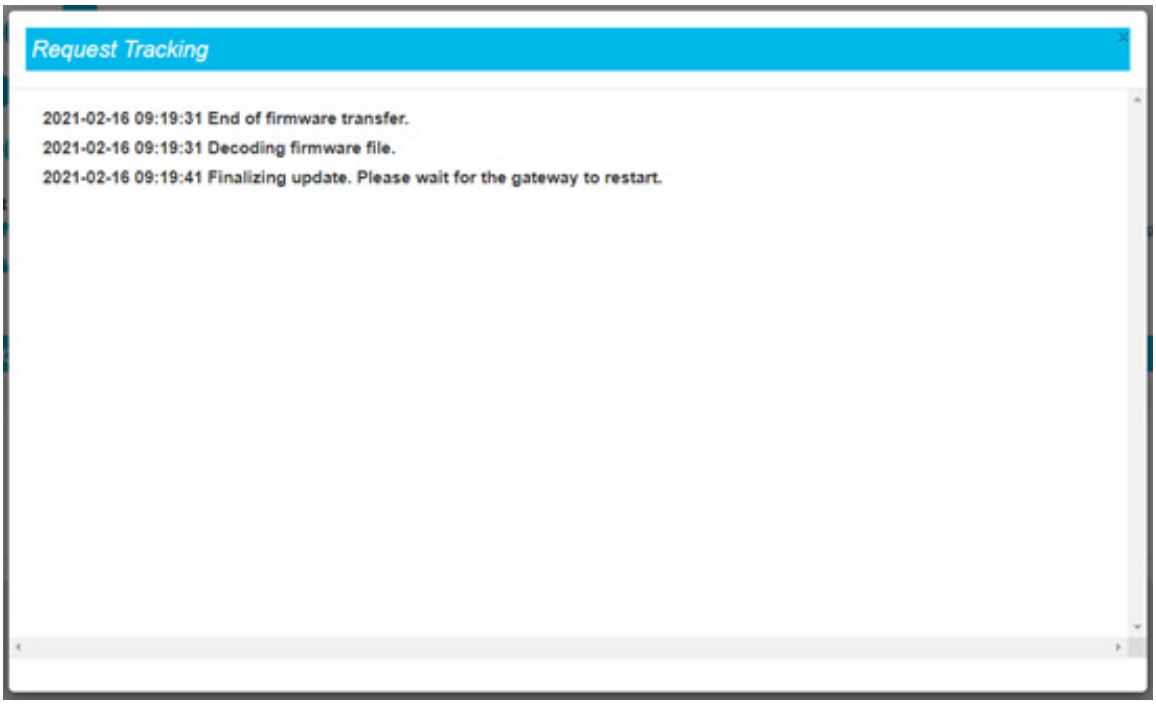
The web interface “Update” menu is used to update the concentrator.



Follow the steps below to update the the concentrator:

- Retrieve the firmware from the web site (see section “6. Update”): https://www.webdyn.com/support/WebdynSunPM/.
- Unzip the retrieved file.
- Click the “Select your firmware file” field. A window opens used to select the new firmware.
- Select the “wgapp_x.x.x.xxxxx.spm” firmware that has a “.spm “ extension.

- Press the “Download and apply the new firmware” button.
- Follow the update progress in the window that opens:



- Please wait while the update is applied and the concentrator restarts.
- Refresh the web page (F5 key on the keyboard).
- Connect back to the concentrator (see section 3.2: “Embedded web interface”).
- Go back to the concentrator’s “Upgrade” page.
- Check that the new version is shown in the Firmware version field.

- The concentrator has been updated.

Do not disconnect the concentrator and avoid operations on it during its update.

If an error occurs during the update, the concentrator will keep its previous operational firmware. In that case, repeat the update procedure exactly as indicated.

3.3.3.3 Rebooting the WebdynSunPM

The “Reboot” menu of the web interface allows you to simply restart the concentrator.

To restart the concentrator, follow these steps:

1. Click the OK button in the new window.

2. Please wait while the hub restarts. The screen is then greyed out, during the restart then once the concentrator has restarted, it will display the following screen:

3. Refresh the web page (F5 key on the keyboard)

4. Connect again to the concentrator (see chapter 3.2: “”)

5. The concentrator is ready again.

3.3 Carte Micro SD

Configuration by Micro SD card follows the same operation as previously described with the FTP/SFTP/ WebDAV server (See chapter 3.1: “FTP/SFTP/WebDAV”).

The only difference is that the hub does not need a connection to the remote server since all files will be accessible directly on the inserted SD card.

Also, the directories on the SD card not being configurable, the tree structure must respect the following format:

- /CONFIG
- /ALARM
- /LOG
- /BIN
- /CERT

  /DATA

- /CMD
- /DEF
- /SCRIPT

If the directories do not exist, they are created on the SD card at the next “connection” request.

If the hub is configured to use the SD card and the user makes a test connection, the device will search for any configuration files on the card and use them.

The Micro SD card is seen and treated as an FTP server by the hub.

It should be noted that the command files (CMD) present on the SD card are not processed on the SD card by the concentrator.

Webdyn does not provide any SD card. Please contact a computer hardware retailer.

4. Operation

The concentrator communicates with one or more remote servers using the FTP/SFTP/WebDAV-HTTPS protocol and/or the MQTT/MQTTS protocol. These servers can be used to manage the concentrator remotely.

Remote servers have several roles:

- Store the data and alarms collected locally by the concentrator: at each connection to the server, whether following a manual request, the triggering of an alarm (case of an action code 8) or the triggering of the connection schedule, the concentrator deposits its memorized data.
- Configure the concentrator: at each connection, the concentrator synchronizes its configuration with the dedicated file present on the server. In the absence of this file, the concentrator creates it from its current configuration.
- Trigger actions on concentrator: the command files must be placed on the server in a directory associated with the concentrator.
- Monitor the concentrator and assist in diagnosis: the concentrator can upload log files for diagnosis purposes.

If only one of the servers is an FTP/SFTP /WebDAV-HTTPS server, the client must choose whether to define it as server 1 or server 2 depending on the required behaviour:

- Server 1: it will be considered a main server.
- Server 2: tit will be considered a backup server.

The main server is used to create or modify the configuration and send commands, as well as to receive alarms. The backup server is only used as a copy of the files uploaded to the main server. It cannot create or modify the configuration or carry out any actions.

4.1 The FTP/SFTP/WebDAV Server

For the WebdynSunPM, the operation of an FTP, SFTP or WebDAV-HTTPS server is identical. But it is preferable to use an SFTP or WebDAV-HTTPS server which integrates security layers unlike a classic FTP. The description in this chapter applies to the various types of servers.

Parameters:

The server is defined by the following parameters:

- An address: This can be an IP address or a domain name.
- A connection port (by default 21in FTP, 22 in SFTP, 443 in WebDAV-HTTPS).
- A login and a password: The parameters are used to define the account to be used.
- A root directory: The root directory can be "/", the server root directory, or a series of sub-directories (for example: "/WebdynSunPM/00CF4/").

You can configure your concentrator remotely from your server. This is only possible if your WebdynSunPM is properly configured to upload and synchronise its configuration on this one.

Server Tree Structure:

The server must have the tree structure specific to the WebdynSunPM product. The concentrator proposes one by default but it can be customised. This architecture must exist on the server before the first connection because the concentrator does not create the directories.

Below the root directory, the server must have the following directories:

NAME	RIGHTS	DESCRIPTION
/CONFIG	Read/Write	Contains config files. The concentrator configuration uploaded by the concentrator is in the following format: <uid>_config.ini The connection interface configuration uploaded by the concentrator is in the following format: <uid>_daq.csv The connection schedule configuration uploaded by the concentrator is in the following format: <uid>_var.ini The connection schedule configuration uploaded by the concentrator is in the following format: <uid>_scl.ini
/DEF	Read/Write	Contains the definition files. The definition file has the following format: <uid>_<interface>_<comment>.csv
/SCRIPT	Read/Write	Contains the script files. The script file has the following format: <comment>.lua
/CERT	Read/Write	Contains the certificates. The certificate has the following format: <comment>.pem

/DATA	Write	Contains the collected data. The data file name is in the following format: <uid>_<interface>_<timestamp>.csv.gz
/CMD	Read/Write/ Delete	Contains the commands. The command file has the following format: <uid>_cmd.csv
/ALARM	Write	Contains the alarms. The alarm file name is in the following format: <uid>_AL_<timestamp>.csv.gz
/LOG	Write	Contains the log and debug files. The log file has the following format: <uid>_LOG_<timestamp>.log.gz The debug file has the following format: <uid>_ SYSTEME_<timestamp>.tar.gz
/BIN	Read	Contains the update files. The concentrator update has the following format: wgapp_<version>.spm

Where:

- <uid>: Concentrator identifier (site).
- <timestamp>: The timestamp format is “YYMMDD_HHMMSS” so that an alphabetical sort of the directory gives the chronological order.
- <interface>: the interface name from a defined list (see section 3.1.2.1.3.4: “Declaration of equipment to be supervised”).
- <comment>: free user field.
- <version>: update version number.

The data, alarm and log files are compressed to the Gzip format “.gz”.

The minimum access rights to the different directories must be defined as specified in the table above.

If the directories are not created at the concentrator connection, or if the rights are not sufficient to upload or download files, contact the server administrator.

All files exchanged between the concentrator and the server must have standard UTF-8 encoding.

Operation:

In FTP or SFTP, if the “FTP_TwoStepsSendingDisabled” or “FTP2_TwoStepsSendingDisabled” parameter equals “1”, the concentrator uploads the files to the server using a 2 step process:

- At the start of the transfer the file has an additional “.tmp” extension.
- When the file transfer is complete, it is renamed by removing the “.tmp” extension.

This process allows the remote server to easily differentiate between files being uploaded and files that are completely uploaded.

For a WebDAV-HTTPS server, this mechanism is useless.

File formats:

The concentrator manages different formats depending on the file type. They can be grouped by extension:

EXTENSION	FILE TYPE	DESCRIPTION
.ini	• Concentrator configuration file • Connection schedule file • Data file (compressed)	Configuration file in a data format
.csv	• Connection interface file • Device definition files • Alarm file	Delimited data file in the form of semi-colon delimited values. (easy to use with Excel type spreadsheet software)
.json	• Command file	Text file containing JSON
.lua	• Script file	Script file
.pem	• Certificate file	Certificate file
.log	• Log file (compressed)	Log file (compressed)
.spm	• Update file	Update file

4.1.1 The Configuration “CONFIG”

The concentrator can receive remote configurations in configuration files or from text messages.

Configuration File:

The WebdynSunPM concentrator needs 4 types of configuration file in text and CSV format. The files names are the following:

<uid>_config.ini

<uid>_daq.csv

<uid>_var.ini

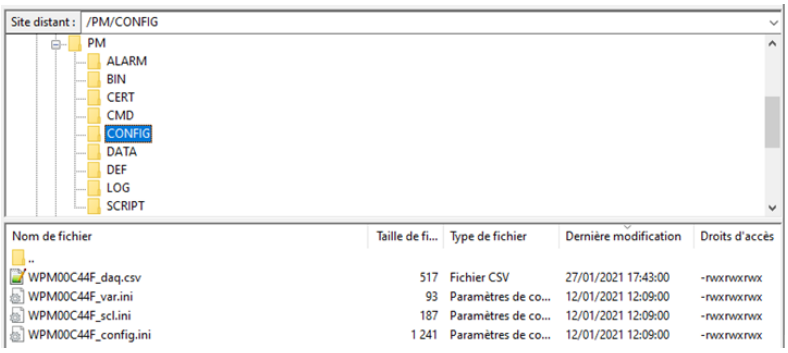
<uid>_scl.ini

Where <uid> is the concentrator identifier.

The current configuration is available on the remote server in the “CONFIG” directory. Whether after a local or a remote configuration update, the concentrator sends its new configuration to the remote server at the next connection.

Configuration files can be sent remotely using the “CONFIG” directory. Configuration files must be uploaded or modified in the directory. On the next connection to the server, the concentrator will carry out 2 steps:

- Download the configuration file available on the server.
- Apply the new configuration.



The configuration file names indicated above must be respected.

Once the new configuration has been applied, the result is indicated in the concentrator’s log file.

If there is an error in the configuration file such as an incorrect value, the concentrator will not take it into account and will use its default value if one exists, otherwise the file will be rejected. The LOG file will report the error and the applied default value.

Refer to section “1. Concentrator Operation” or to “Appendix A: “_config.ini” file” to see the list of variables and their possible values.

4.1.2 “DEF”, the Definitions

The devices declared in the “<uid>_daq.csv” file use a definition file describing all the available variables on the device. The devices available on the concentrator are:

- Inputs/outputs: IO
- Remote customer information: TIC
- RTU/TCP Modbus
- Proprietary inverter protocols

The definition files the WebdynSunPM can generate automatically are:

- The IO file
- The SunSpec files
- The proprietary inverter protocol files

To create modbus definition files, see section 3.1.2.2.2.2: “Modbus”.

Launching a device scan makes it possible to automatically generate its definition file and to upload it to the “DEF” directory on the server. It is also possible to build your own definition file or to modify the automatically generated one.

A new definition file or a modification to one of the definition files is automatically retrieved by the concentrator at its next connection to the server.

The definition file name can be customised, by default it has the following format:

<uid>_<interface>_<comment>.csv

Where:

- <uid>: Concentrator identifier
- <interface>: the interface name from a defined list (see section 3.1.2.1.3.4: “Declaration of equipment to be supervised”)
- <comment>: free user field

Examples:

WPM00C44F_SunSpec_inverter_SMA_Solar_Inverter_9301_modbusTCP.csv

WPM00C44F_IO.csv

custom.csv

Refer to section “2. Connected Device Definition” for the definition file structures.

4.1.3 "DATA"

Data is uploaded to the "DATA" directory on the FTP server in the form of CSV format files compressed to Gzip ".gz" format.

Below is the data file name format:

<uid>_<interface>_<timestamp>.csv.gz

Where:

- <uid>: Concentrator identifier
- <interface>: the interface name from the following list:
  - TIC
  - IO
  - MODBUS
- <timestamp>: The timestamp format is "YYMMDD_HHMMSS" so that an alphabetical sort of the directory gives the chronological order

Examples:

WPM00C44F_MODBUS_210112_105947.csv.gz

WPM00C44F_IO_210202_084443.csv.gz

WPM00C44F_TIC_210202_095243.csv.gz

Every declared and configured device acquires its data over a defined period (see section 3.2.1: "Devices") and regularly uploads it to a server (see section 3.2.2.5: "Servers") in the "DATA" directory.

The concentrator stores the data until it has been uploaded to the server. This makes it possible to resend it if the transfer fails.

When the concentrator memory is full, new data is not stored until the memory has been emptied by uploading files to a server. The concentrator can store up to 50Mb of uncompressed data per defined device.

The WebdynSunPM permanently collects the device and interface data and saves it. The reported values are always raw and must be connected to the device definition file. The contents of a data file are in 2 parts which are:

- A header: which is different depending on the device or interface.
- Data: which is formatted identically for all devices and interfaces.

4.1.3.1 Input/Output (IO) Header

The IO data file header is the following:

```
TypeIO;fileDefinitionName
```

Colour code:

- Black: fixed text.
- Blue: device-specific information or data.

Where:

- fileDefinitionName: definition file name for the Inputs/Outputs

4.1.3.2 Device Header (Modbus, inverters)

The device data file header is the following:

```
DEVICEINDEX;NumDevice_1
Protocol_1;fileDefinitionName_1
...
...
DEVICEINDEX;NumDevice_N
Protocol_N;fileDefinitionName_N
```

Colour code:

- Black: fixed text.
- Blue: device-specific information or data.

Where:

- NumDevice_N: the device “index” in the connection interface configuration file for device N (see section 3.1.2.2.2: “Definition file content”).
- fileDefinitionName_N: definition file name for device N.

4.1.3.3 Data

The formatting of the data is identical, regardless of the equipment or the interface.

The reported values are raw and must be linked to the settings made on the device or interface in its definition file. The SI must interpret the data using the raw data file and the definition file in order to be able to apply the A and B coefficients as well as the unit for each variable. (see chapter 3.1.2.2.2: “Content of the definition file”).

The “action” field associated with each variable in the definition file allows you to select a type of value:

“ACTION” CODE	DESCRIPTION
0	No values are reported.
1	The parameter value is reported.
2	The min, max and mean values are reported.
4 or 6	The instant value is reported.
7	The min, max and average values are uploaded in the acquisition file; the instantaneous value is uploaded in the file created by the getData command.
8 or 9	The instant value is reported and and alarm is generated every time the value changes.

The device or interface data file data is the following:

```
nbVariableDevice_N;indexVariable_1_Device_N;indexVariable_A_
Device_N;val_of_EnAdvData(=1)

datetime_1;variable_1_value_1_Device_N;variable_x_value_1_
Device_N;nb_refreshes_during_1

datetime_2;variable_1_value_2_Device_N;variable_x_value_2_
Device_N;nb_refreshes_during_2

...

datetime_Y;variable_1_value_B_Device_N;variable_A_value_B_
Device_N;nb_refreshes_during_B
```

Colour code:

- Green: optional header that can be enabled or disabled using the “FTP_HeaderOption” parameter for server 1 and “FTP2-HeaderOption” for server 2 in the <uid>_config.ini configuration file.
- Blue: device-specific information or data.

Where:

- nbVariableDevice_N: Total number of collected variables for device N.
- indexVariable_X_Device_N: Index X of collected variables for device N.
- datetime_Y: data timestamp at acquisition point Y. For the format, see variables FTP_EuroDateFormat, FTP2_EuroDateFormat, HTTP_EuroDateFormat and HTTP2_EuroDateFormat in section 3.2.2.5: “Servers”.
- variable_A_value_B_Device_N: Value A of Variable B corresponding to Index A collected at acquisition point Y and the action defined in the definition file for device N.
- nb_refreshes_during_B: number of complete readings over this acquisition period of variable B. This information is displayed only if the “FTP_EnableAdvancedData” or “HTTP_EnableAdvancedData” parameter of server 1 or “FTP2_EnableAdvancedData” or “HTTP2_EnableAdvancedData” of server 2 is at 1. This data may only be useful for Modbus and Inverter devices. For IOs, TICs, virtual devices and for the parameter collection file, the number of refreshes is always 0.

To avoid needlessly sending data to the server and thereby optimise the connection, it is recommended to only enable the variables that need to be reported.

Input/Output (IO):

Example of an IO data file with an acquisition frequency of every 10 seconds:

- Input/Output Configuration:

INPUT/OUTPUT	“ACTION” CODE	DISPLAY
1	4	Instant value
2	0	None
3	4	Instant value
4	4	Instant value
5	4	Instant value
6	2	Min, max and average values
7	8	Instant value
8	4	Instant value

- CSV Data File (edited using Excel):

TypeIO	WPM00C44F_IO.csv									
9	1	3	4	5	6(min)	6 (max)	6 (avg)	7	8	9
21/02/02-15:41:10	0	0	5	1	130	170	150	5	0	0
21/02/02-15:41:20	0	1	2	2	130	170	120	3	0	0
21/02/02-15:41:30	1	0	3	1	120	160	140	2	0	0
21/02/02-15:41:40	1	0	6	5	120	170	140	3	1	0
21/02/02-15:41:50	0	0	6	4	130	180	150	5	0	0
21/02/02-15:42:00	0	0	6	5	130	200	160	6	0	0

Devices (Modbus, inverters):

Example of a device data file with an acquisition frequency of every 10 minutes:

- Device 1 index configuration:

INDEX	"ACTION" CODE	DISPLAY
1	1	Parameter value
2	0	None
3-11	4	Instant value
12	8	Instant value

- Device 2 index configuration:

INDEX	"ACTION" CODE	DISPLAY
1-2	2	Min, max and average values

- CSV data file (edited using Excel):

DEVICEINDEX	1											
modbusTCP	WPM00C44F_SunSpec_inverter_SMA_Solar_Inverter_9301_modbusTCP.csv											
11	1	3	4	5	6	7	8	9	10	11	12	1
21/02/05-09:50:00	32	52	5	102	1	0	1	0	0	0	0	5
21/02/05-10:00:00	35	57	5	108	1	10	0	0	0	0	1	6

DEVICEINDEX	2						
SMANET	WPM00C44F_SMA_Inverter_SMA_WR21TL09.csv						
7	1 (min)	1 (max)	1 (avg)	2 (min)	2 (max)	2 (avg)	1
21/02/05-09:50:00	16	32	26.00	52	58	51.00	12
21/02/05-10:00:00	4	6	05:50	102	105	103.00	12

4.1.4 "ALARM" Alarms

Alarms are uploaded in the form of CSV format files compressed to Gzip ".gz" format. They are uploaded to the "ALARM" directory on remote servers. No files other than alarm files are uploaded to the servers and the concentrator will not trigger NTP synchronisation. The list of alarms that can be generated is:

ALARM SOURCE	INFO	DESCRIPTION
GATEWAY	Power ON	Concentrator boot
	Power OFF	Concentrator shut down
	TIC accessory loss	TIC accessory removed
	TIC accessory return	TIC accessory reconnected
IO	Definition file name + Index + Value	The value of an alarm type input has changed
MODBUS	Definition file name + Index + Value	The value of an alarm type index has changed

A "Power OFF" alarm is sent following a power cut of at least 10 seconds and a "Power ON" alarm is sent once the power supply has returned for at least 1 minute. The other alarms have no timers and are sent as soon as the concentrator detects them.

The alarm file on the server has the following format:

<uid>_AL_<timestamp>.csv.gz

Where:

- <uid>: Concentrator identifier.
- <timestamp>: The timestamp format is "YYMMDD_HHMMSS" so that an alphabetical sort of the directory gives the chronological order.

The GATEWAY type alarm file format is the following:

```
datetime_1;GATEWAY;info_1
datetime_2;GATEWAY;info_2
...
datetime_Y;GATEWAY;info_X
```

Colour code:

- Black: fixed text.
- Blue: device-specific information or data.

Where:

- datetime_Y: alarm timestamp at trigger point Y. For the format, see variables FTP_EuroDateFormat, FTP2_EuroDateFormat, HTTP_EuroDateFormat and HTTP2_EuroDateFormat in section 3.2.2.5: “Servers”.
- info_X: Information on alarm X.

The IO and MODBUS type alarm file format is the following:

```
datetime_1;AlarmSource_1;fileDefinitionName_1;nameEquipment_1;
indexVariable_1;value_1
datetime_2;AlarmSource_2;fileDefinitionName_2;nameEquipment_2
indexVariable_2;
value_2
...
datetime_Y;AlarmSource_X;fileDefinitionName_N;nameEquipment_N;
indexVariable_A;variable_A_value_B
```

Colour code:

- Blue: device-specific information or data.

Where:

- datetime_Y: alarm timestamp at trigger point Y. For the format, see variables FTP_EuroDateFormat, FTP2_EuroDateFormat, HTTP_EuroDateFormat and HTTP2_EuroDateFormat

in section 3.2.2.5: “Servers”.

- AlarmSource_X: Source that triggered the alarm (IO, MODBUS).
- fileDefinitionName_N: Definition file name for the device.
- nameEquipment_N: Device name. “Name” field in the “<uid>_daq.csv” file (see section “3.1.2.1.3.4 Declaration of equipment”).
- indexVariable_A: Index A for the alarm variable.
- variable_A_value_B: Value B of Variable A corresponding to alarm index A.

The reported values for IO and MODBUS alarms are raw and must be linked to the index configuration in the device definition file. The IS must interpret the data using the raw data file and the definition file to be able to apply the defined factors A and B and the unit for each input or index. (see section “2.2. Definition File Content”).

Example of a MODBUS alarm file:

21/02/12-07:00:19	MODBUS	MODBUS_DELTA_M88H-COM1.20-Sunspec.csv	OndA	48	0.000000
21/02/12-07:00:49	MODBUS	MODBUS_DELTA_M88H-COM1.20-Sunspec.csv	OndA	48	3.000000

4.1.5 “CMD” Commands

Command files are used to perform tasks remotely on the hub. It is thus possible to ask the concentrator to launch a search for equipment, to obtain the configuration, and so on. The list of possibilities is limitless.

Command files are not supported when the SD card is used.

The operation of the command file (format and processing) is described in chapter 5: “Commands”. The commands available are detailed in chapter 5.3: “List of Commands”.

4.1.6 “SCRIPTS”

The “SCRIPT” directory on the server is used to supply or retrieve Lua scripts to and from the concentrator.

Lua script files can have the following extension:

- “.lua”: an unencrypted LUA script
- “.luax”: an LUA script encrypted with customer keys
- “.luaw”: an encrypted LUA script with a Webdyn license

The script file format is below:

<comment>_.lua

<commentaire>_.luax

<scriptwebdyn>_.luaw

Where:

- <comment>: free user field
- <scriptwebdyn> : Webdyn proprietary script

Examples:

ControlPower.lua

Injection.luax

Deie.luaw

For more details on script use, see the “WebdynSunPM LUA User Guide.pdf” document available at: https://www.webdyn.com/support/WebdynSunPM/

Proprietary Webdyn “.luaw” scripts are not redeposited by the hub if deleted on the remote server.

4.1.7 “BIN” Update

The “BIN” directory on the server is used to store the firmware used to update the concentrator.

The update file format is below:

<uid>_wgapp_x.x.x.xxxxx.spm

Where:

- <uid>: Concentrator identifier.
- x.x.x.xxxxx: is the firmware version number.

Examples: wgapp_3.2.9.34734.spm

To perform the update, follow the procedure described in chapter 6.2: “By FTP/SFTP/WebDAV”). After applying the update, it is possible to delete the firmware file.

4.1.8 "LOG"

Log files are the files used to monitor the concentrator actions and analyse when things go wrong.

When contacting support it is essential to be able to provide the log files for the problem encountered.

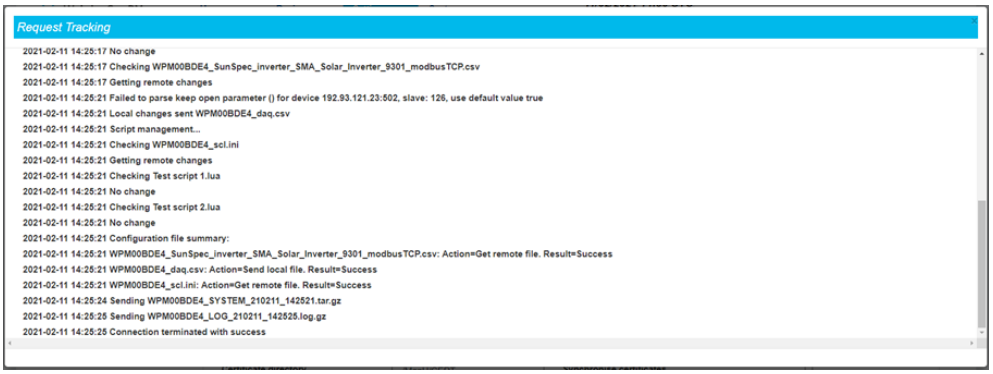
There are 4 types of log file:

- Connection logs: "uid"_LOG_"date".log.gz.
- LUA script run logs: "uid"_LUA_"script name"_"date".log.gz.
- SunSpec detection logs: "uid"_SUNSPEC_"date".log.gz.
- System logs: "uid"_SYSTEM_"date".log.gz.

4.1.8.1. Connection Logs

Every time the concentrator connects to a server, either by modem or Ethernet, all operations are saved for future viewing.

When the connection is forced by the local web interface, a window is displayed showing all ongoing operations as shown below:



For the same connection, the log file uploaded to the server contains the following data:

```
2021-02-11 14:25:03:Firmware version : WebdynSunPM 3.2.11.34874
2021-02-11 14:25:03:Connection 1 wait for exclusive...
2021-02-11 14:25:03:Connection requested
2021-02-11 14:25:16:Ntp synchronisation success
2021-02-11 14:25:16:Connection to 192.168.2.13:test server in
progress …
2021-02-11 14:25:16:Alarms management…
2021-02-11 14:25:16:Data acquisition management…
2021-02-11 14:25:16:Sending WPM00BDE4_MODBUS_210211_142503.csv.gz
2021-02-11 14:25:16:Data acquisition management…
2021-02-11 14:25:16:Sending WPM00BDE4_IO_210211_142507.csv.gz
2021-02-11 14:25:16:Sending WPM00BDE4_SUNSPEC_210211_142507.log.gz
2021-02-11 14:25:16:Configuration management…
2021-02-11 14:25:16:Checking WPM00BDE4_config.ini
2021-02-11 14:25:16:No change
2021-02-11 14:25:16:Checking WPM00BDE4_daq.csv
2021-02-11 14:25:16:Sending local changes
2021-02-11 14:25:16:Remove file
2021-02-11 14:25:17:Checking WPM00BDE4_var.ini
2021-02-11 14:25:17:No change
2021-02-11 14:25:17:Checking WPM00BDE4_IO.csv
2021-02-11 14:25:17:No change
2021-02-11 14:25:17:Checking WPM00BDE4_SunSpec_inverter_SMA_Solar_
Inverter_9301_modbusTCP.csv
2021-02-11 14:25:17:Getting remote changes
2021-02-11 14:25:21:Failed to parse keep open parameter () for device
192.93.121.23:502, slave: 126, use default value true
2021-02-11 14:25:21:Local changes sent WPM00BDE4_daq.csv
2021-02-11 14:25:21:Script management...
2021-02-11 14:25:21:Checking WPM00BDE4_scl.ini
2021-02-11 14:25:21:Getting remote changes
2021-02-11 14:25:21:Checking Test script 1.lua
2021-02-11 14:25:21:No change
2021-02-11 14:25:21:Checking Test script 2.lua
2021-02-11 14:25:21:No change
2021-02-11 14:25:21:Configuration file summary:
2021-02-11 14:25:21:WPM00BDE4_SunSpec_inverter_SMA_Solar_
Inverter_9301_modbusTCP.csv: Action=Get remote file. Result=Success
2021-02-11 14:25:21:WPM00BDE4_daq.csv: Action=Send local file.
Result=Success
2021-02-11 14:25:21:WPM00BDE4_scl.ini: Action=Get remote file.
Result=Success
2021-02-11 14:25:24:Sending WPM00BDE4_SYSTEM_210211_142521.tar.gz
```

First features the information on the concentrator software version, then the connection to the NTP server for time synchronisation.

Then comes the actual connection to the server starting with alarm synchronisation where applicable:

```
2021-02-11 14:25:16:Alarms management…
```

Once this step is complete, the data files are transferred to the server:

```
2021-02-11 14:25:16:Data acquisition management…
2021-02-11 14:25:16:Sending WPM00BDE4_MODBUS_210211_142503.csv.gz
2021-02-11 14:25:16:Data acquisition management…
2021-02-11 14:25:16:Sending WPM00BDE4_IO_210211_142507.csv.gz
2021-02-11 14:25:16:Sending WPM00BDE4_SUNSPEC_210211_142507.log.gz
```

Then comes the processing of the configuration files to be sent or received:

```
2021-02-11 14:25:16:Configuration management…
2021-02-11 14:25:16:Checking WPM00BDE4_config.ini
2021-02-11 14:25:16:No change
2021-02-11 14:25:16:Checking WPM00BDE4_daq.csv
2021-02-11 14:25:16:Sending local changes
2021-02-11 14:25:16:Remove file
2021-02-11 14:25:17:Checking WPM00BDE4_var.ini
2021-02-11 14:25:17:No change
2021-02-11 14:25:17:Checking WPM00BDE4_IO.csv
2021-02-11 14:25:17:No change
2021-02-11 14:25:17:Checking WPM00BDE4_SunSpec_inverter_SMA_Solar_
Inverter_9301_modbusTCP.csv
2021-02-11 14:25:17:Getting remote changes
2021-02-11 14:25:21:Failed to parse keep open parameter () for device
192.93.121.23:502, slave: 126, use default value true
2021-02-11 14:25:21:Local changes sent WPM00BDE4_daq.csv
```

The log file indicates that a certain number of files were checked and that no modifications were detected.

It then indicates that file “WPM00BDE4_SunSpec_inverter_SMA_Solar_Inverter_9301_modbusTCP.csv” was modified on the remote server. It is then retrieved, read and imported locally:

```
2021-02-11 14:25:17:Checking WPM00BDE4_SunSpec_inverter_SMA_Solar_
Inverter_9301_modbusTCP.csv
2021-02-11 14:25:17:Getting remote changes
```

Modifications were also detected on the devices and the "_daq.csv" file is sent to the server:

```
2021-02-11 14:25:21:Local changes sent WPM00BDE4_daq.csv
```

The processing completes with script file management:

```
2021-02-11 14:25:21:Script management...
2021-02-11 14:25:21:Checking WPM00BDE4_scl.ini
2021-02-11 14:25:21:Getting remote changes
2021-02-11 14:25:21:Checking Test script 1.lua
2021-02-11 14:25:21:No change
2021-02-11 14:25:21:Checking Test script 2.lua
2021-02-11 14:25:21:No change
```

All the scripts are checked on the server. Here, the log indicates that there were no changes.

Once all the processing is complete, a summary recaps all the completed processing:

```
2021-02-11 14:25:21:Configuration file summary:
2021-02-11 14:25:21:WPM00BDE4_SunSpec_inverter_SMA_Solar_
Inverter_9301_modbusTCP.csv: Action=Get remote file. Result=Success
2021-02-11 14:25:21:WPM00BDE4_daq.csv: Action=Send local file.
Result=Success
2021-02-11 14:25:21:WPM00BDE4_scl.ini: Action=Get remote file.
Result=Success
```

It shows that files "WPM00BDE4_SunSpec_inverter_SMA_Solar_Inverter_9301_modbusTCP.csv" and "WPM00BDE4_scl.ini" were successfully retrieved from the server.

It also indicates that file "WPM00BDE4_daq.csv" was sent to the remote server.

The log file ends by indicating that the system log files were sent:

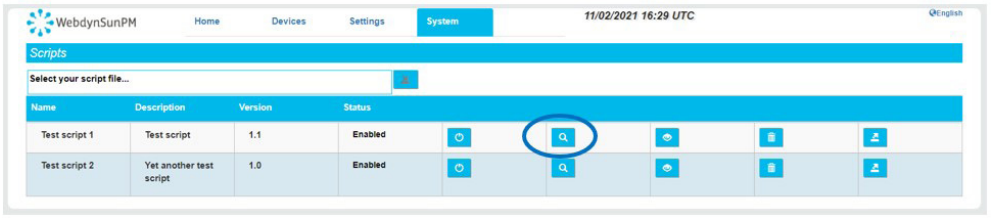
```
2021-02-11 14:25:24:Sending WPM00BDE4_SYSTEM_210211_142521.tar.gz
```

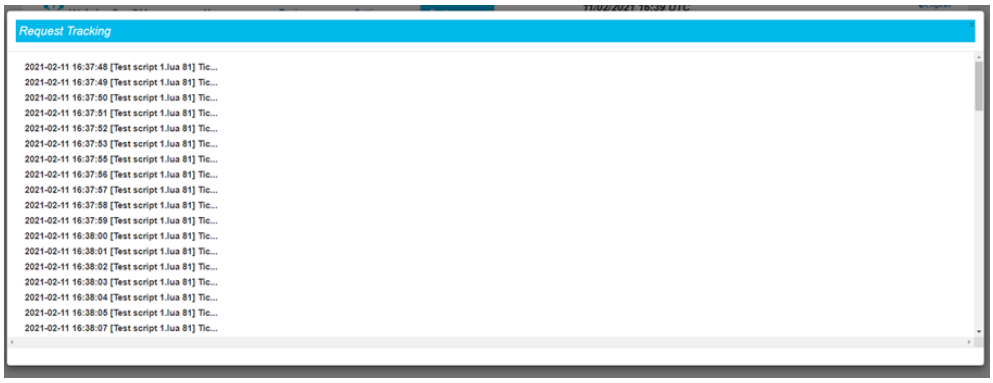
4.1.8.2 Script Logs

Each script runs in a separate environment. As a result, they each have an automatically generated log file when they run.

The log file can be viewed in 2 ways:

- Either directly on the local web site on the script page by clicking the view log icon (see section "3.2.3.1 Scripts"):



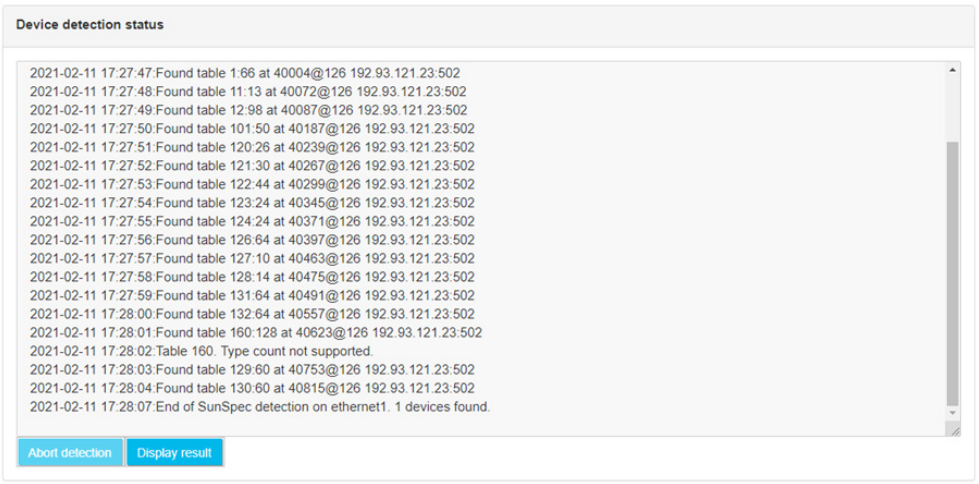


- Or on the server in the log files. The logs displayed below are also available on the server. They contain the "_LUA_" character string followed by the script name. For the log shown above, the file name will be "WPM00BDE4_LUA_Test script 1_210211_165930.log.gz" and will contain exactly what is displayed on the screen.

Note that script log files are in "CSV" format so that they can be imported to spreadsheet software.

4.1.8.3 SunSpec Detection Logs

During a SunSpec detection, the detection result is displayed on the local web page as it progresses:



This log can also be found in the "LOGS" directory at the next connection to the server. The file name is composed of the concentrator's uid, followed by SUNSPEC and the date and time the file was uploaded to the server: "WPM00BDE4_SUNSPEC_210211_181543.log.gz".

The log file contains exactly the same thing as what was displayed on the screen:

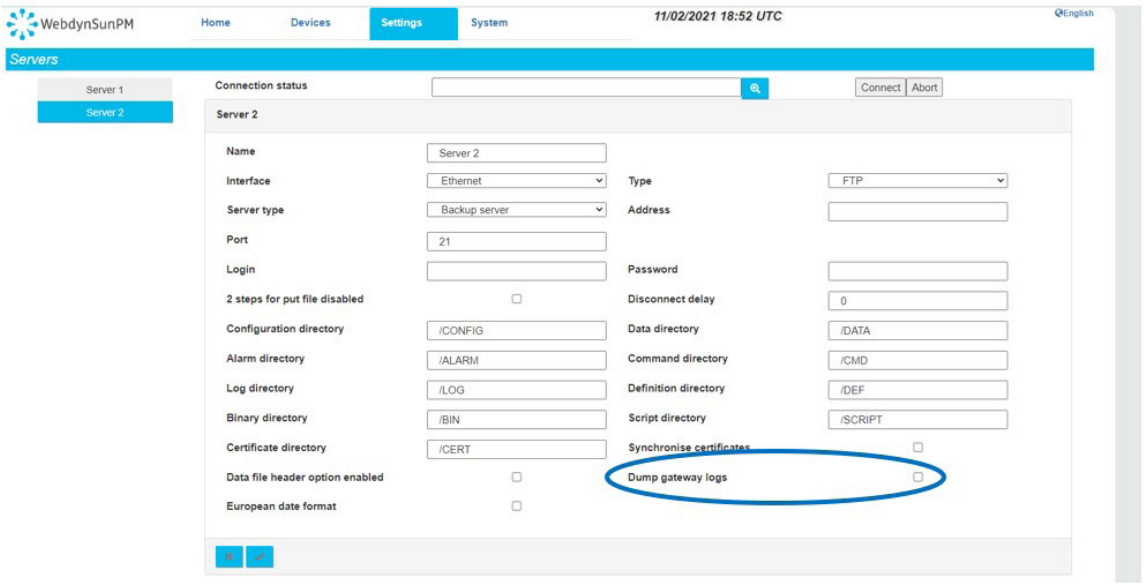
```
21/02/11-17:27:28;SunSpec detection start on ethernet1. Estimated
time : 621 seconds.
21/02/11-17:27:30;NON SUNSPEC device found at @1 192.93.121.23:502.
21/02/11-17:27:31;NON SUNSPEC device found at @2 192.93.121.23:502.
21/02/11-17:27:32;NON SUNSPEC device found at @3 192.93.121.23:502.
21/02/11-17:27:47;SunSpec device found at @126 192.93.121.23:502.
21/02/11-17:27:48;Found table 1:66 at 40004@126 192.93.121.23:502
21/02/11-17:27:49;Found table 11:13 at 40072@126 192.93.121.23:502
21/02/11-17:27:50;Found table 12:98 at 40087@126 192.93.121.23:502
21/02/11-17:27:51;Found table 101:50 at 40187@126 192.93.121.23:502
21/02/11-17:27:52;Found table 120:26 at 40239@126 192.93.121.23:502
21/02/11-17:27:53;Found table 121:30 at 40267@126 192.93.121.23:502
21/02/11-17:27:54;Found table 122:44 at 40299@126 192.93.121.23:502
21/02/11-17:27:55;Found table 123:24 at 40345@126 192.93.121.23:502
21/02/11-17:27:56;Found table 124:24 at 40371@126 192.93.121.23:502
21/02/11-17:27:57;Found table 126:64 at 40397@126 192.93.121.23:502
21/02/11-17:27:58;Found table 127:10 at 40463@126 192.93.121.23:502
21/02/11-17:27:59;Found table 128:14 at 40475@126 192.93.121.23:502
21/02/11-17:28:00;Found table 131:64 at 40491@126 192.93.121.23:502
21/02/11-17:28:01;Found table 132:64 at 40557@126 192.93.121.23:502
21/02/11-17:28:02;Found table 160:128 at 40623@126 192.93.121.23:502
21/02/11-17:28:03;Table 160. Type count not supported.
21/02/11-17:28:04;Found table 129:60 at 40753@126 192.93.121.23:502
21/02/11-17:28:05;Found table 130:60 at 40815@126 192.93.121.23:502
21/02/11-17:28:07;End of SunSpec detection on ethernet1. 1 devices
found.
```

Note that the SunSpec log files are in “CSV” format so that they can be imported to spreadsheet software.

4.1.8.4 System Logs

System logs are internal concentrator operation logs. They contain internal debugging information that may be requested by support.

By default, the system logs are not transferred to the FTP server. To do so, the “Dump gateway logs” option must be enabled on the server configuration page:



System log files are also transferred automatically when a connection is forced via the server configuration page, by clicking on the “Connect” button or via the SMS command “connect”.

The file name is composed of the concentrator’s uid, followed by “SYSTEM” and the date and time the file was uploaded to the server: “WPM00BDE4_SYSTEM_210211_181543.log.gz”.

4.1.9 Web Services

There are 2 parameters to configure the web services:

- WebServiceEnable: 0 to disable and 1 to enable web services.
- WebServiceUrl: when web services are enabled, this parameter contains the network address that will be called under certain conditions described in this section.

The following actions can trigger the call to a web service:

- Starting the product
- Deposit of data files on the FTP server
- Modification of the configuration (loading or depositing configuration)
- Running a batch file
- Loading a new binary file

Calls to web services are made in the following format:

```
URL/confirm.php;NSITE=IDsite&ACTION=action&ACTION-COMP=donneés
complémentaires&RC=0&RC-COMP=
```

The “confirm.php” page is called at the configured URL. This page will receive the different information of the actions and will have to process them.

The information is:

- NSITE: concentrator identifier. This is the identifier that was configured during commissioning
- Action: identifier of the action that triggered the call to the web service. The list of actions is described below
- ACTION-COMP: for certain actions, additional information is specified. The complete list is described below
- RC: always 0
- RC-COMP: always empty

Action	Web service data
First FTP connection since hub startup. When the gateway connects for the first time since it was started (power up, software update, etc.), the software version number is sent.	NSITE=IDsite&ACTION=VERSION&ACTION-COMP=version firmware&RC=0&RC-COMP=
Sending data from inverters or IOs to the FTP server. When the gateway connects to the FTP server to upload the data files of the different devices, this web service is called.	NSITE=IDsite&ACTION=UPLOADDATA&RC=0&RC-COMP=
Sending alarms to the FTP server. When an alarm has been detected, an alarm file is uploaded to the FTP server and this web service is called.	NSITE=IDsite&ACTION=UPLOADALARM&RC=0&RC-COMP=
Sending configuration files from the gateway to the FTP server. When a configuration change is made locally on the gateway, on the next connection, the impacted configuration files are transferred to the FTP server and this web service is called	NSITE=IDsite&ACTION=UPLOADGLOBAL&ACTION-COMP=liste de fichiers de configuration&RC=0&RC-COMP= The names of the configuration files are separated by the character “; “. The affected files are “_config.ini”, “_var.ini” and “_daq.ini”

Uploading configuration files from the FTP server to the gateway. When changes to remote configuration files are made, the gateway loads these files on the next connection and this web service is called.	NSITE=IDsite&ACTION=CONFIGGLOBAL&ACTION-COMP=liste de fichiers de configuration&RC=0&RC-COMP= The names of the configuration files are separated by the character “; “. The affected files are “_config.ini”, “_var.ini” and “_daq.ini”
Sending one or more definition files. When the gateway modifies definition files (creation by detection for example), these definition files are sent to the remote FTP server on the next connection and this web service is called.	NSITE=IDsite&ACTION=UPLOADDEF&ACTION-COMP=liste de fichiers de définition&RC=0&RC-COMP= The names of the definition files are separated by the character “; “.
Receive one or more definition files. When definition files are modified on the remote FTP server, the gateway loads them on the next connection. This web service is then called.	NSITE=IDsite&ACTION=CONFIGDEF&ACTION-COMP=liste de fichiers de définition&RC=0&RC-COMP= The names of the definition files are separated by the character “; “.
Executing a command file. When the gateway detects a command file to load and execute on the FTP server, it calls this web service.	NSITE=IDsite&ACTION=CMD&RC=0&RC-COMP=
Loading new firmware. When the gateway detects and uploads a valid firmware file (correct CRC) to the FTP server, it calls this web service.	NSITE=IDsite&ACTION=CONFIGBIN&RC=0&RC-COMP=

The web service must return one of the following values:

Code	Description
00	OK
10	Unknown site ID
11	Unknown action code
12	RC received unknown
13	Missing MAC address
-1	Internal server error

Examples of web service requests:

Loading a new firmware version:

```
URL/confirm.php;NSITE=IDsite&ACTION=CONFIGBIN&RC=0&RC=
```

First login since last startup:

```
URL/confirm.php;NSITE=IDsite&ACTION=VERSION&ACTION-COMP=WebdynSunPM
4.2.3.38295&RC=0&RC-COMP=
```

Loading the IDSite_config.ini and IDsite_var.ini configuration files:

```
URL/confirm.php;NSITE=IDsite&ACTION=CONFIGGLOBAL&ACTION-COMP=IDsite_
config.ini;IDsite_var.ini&RC=0&RC-COMP=
```

4.2 The MQTT/MQTTS server

The operation of an MQTT or MQTTS server is identical. But it is preferable to use an MQTTS server which integrates layers of security unlike an MQTT server. The description in this chapter is equally valid for all types of MQTT servers.

Settings:

The MQTT server is defined by the following parameters:

- An address: This address can be an IP address or a domain name.
- An MQTT connection port (default 1883 for MQTT and 8883 for MQTTS).
- A server identification: depends on the type of MQTT server chosen. Identification can be done either by a simple username and password or by certificates and a key to be imported into the concentrator.
- An application identification: Unique identifier of the server allowing to have its own application space.
- Topics: Name of information channels on the server for storing data and alarms.

From an MQTT/MQTTS server, the concentrator supports the following actions:

- Data repository,
- Filing of alarms,
- Order receipt,
- Update hub.

From an MQTT/MQTTS server, you cannot:

- Configure the hub,
- Create, add or modify a definition file,
- Create, add or modify a script,
- Add or replace a certificate,
- Drop the logs.

To perform these actions, you must configure the hub's second server with an FTP/SFTP server.

Server tree:

An MQTT/MQTTS server has information channels called topics. Personalization of topic names depending on the type of server. You must enter the same topic between the MQTT/MQTTS server and the concentrator.

A topic name must be entered for the data so that the concentrator can deposit this data.

If you want to send the alarms to the MQTT/MQTTS server, you must in this case enter a topic for the alarms.

If you want to receive commands from the MQTT/MQTTS server, you must fill in a topic name for the commands and another topic for the command result.

Working:

The concentrator connects and deposits the data on the MQTT/MQTTS server at each planned schedule. (See chapter 3.2.2.5.9: “Schedules”)

If an Alarm and/or Command topic name is entered, then the concentrator remains in permanent connection with the MQTT/MQTTS server in order to carry out the action immediately.

Unlike the FTP/SFTP server, the data deposited on the MQTT/MQTTS server is formatted and takes into account the coefficients A and B defined as well as the unit defined for each variable. (see chapter 3.1.2.2.2: “Content of the definition file”).

4.2.1 Data format

The formatting of the data is identical, regardless of the equipment or the interface.

The reported values are interpreted and use the A and B coefficients defined for each variable in the equipment or interface definition file. (see chapter 3.1.2.2.2: “Content of the definition file”).

The value of the “action” field of the equipment or the interface defined in its definition file allows the following action:

Code « Action »	Description
0	no value is returned.
1	the value of the parameter is raised.
2	the min, max and average values are uploaded.
4 or 6	the instantaneous value is increased.
7	the min, max and average values are uploaded in the acquisition file; the instantaneous value is uploaded in the file created by the getData command.
8 or 9	the instantaneous value is raised and an alarm will be generated each time the value changes.

The data in the device or interface data file is in the following JSON format:

```
{
      "__MetaData__":{
            "id":"concentrator_id"
        },
      "DAQ_Name_eqp_1":[
      {
            "DEF_Name_var_1_eqp_1":var_1_value_1_eqp_1,
            "DEF_Name_var_2_eqp_1":var_2_value_1_eqp_1,
            "DEF_Name_var_X_eqp_1":var_X_value_1_eqp_1,
            "date":"YY/MM/DD-hh:mm:ss",
            "timestamp":value_timestamp_1_eqp_1,
            "nbOfRefreshes":value_nbOfRefreshes_1_eqp_1
      },
      {
            "DEF_Name_var_1_eqp_1":var_1_value_Z_eqp_1,
            "DEF_Name_var_2_eqp_1":var_2_value_Z_eqp_1,
            "DEF_Name_var_X_eqp_1":var_X_value_Z_eqp_1,
            "date":"YY/MM/DD-hh:mm:ss",
            "timestamp":value_timestamp_Y_eqp_1,
            "nbOfRefreshes":value_nbOfRefreshes_Y_eqp_1
      }
      ],
      "DAQ_Name_eqp_N":[
      {
            "DEF_Name_var_1_eqp_N":var_1_value_1_eqp_N,
            "DEF_Name_var_2_eqp_N":var_2_value_1_eqp_N,
            "DEF_Name_var_X_eqp_N":var_X_value_1_eqp_N,
            "date":"YY/MM/DD-hh:mm:ss",
            "timestamp":value_timestamp_1_eqp_N,
            "nbOfRefreshes":value_nbOfRefreshes_1_eqp_N
      },
      {
            "DEF_Name_var_1_eqp_N":var_1_value_Z_eqp_N,
            "DEF_Name_var_2_eqp_N":var_2_value_Z_eqp_N,
            "DEF_Name_var_X_eqp_N":var_X_value_Z_eqp_N,
            "date":"YY/MM/DD-hh:mm:ss",
            "timestamp":value_timestamp_Y_eqp_N,
            "nbOfRefreshes":value_nbOfRefreshes_Y_eqp_N
      }
      ]
}
```

Color code:

- In green: Name or variable of the device or interface.
- In blue: Data that depends on the equipment or the interface.
- In black: Fixed text.

With:

- DAQ_Name_eqp_N: Name of device N, “Name” field of the device in the <uid>_daq.csv file (see chapter 3.1.2.1.3.4: “Declaration of devices”)
- DEF_Name_var_X_eqp_N: Name of variable X of equipment N, “Name” field of the variable in the definition file (see chapter 3.1.2.2.2: “Content of the definition file”)
- var_X_value_Z_eqp_N: Z value of variable X of equipment N collected at acquisition point Y and at the action defined in the equipment N definition file.
- date: Time stamp of data at acquisition point Y in UTC+timezone (see the “NTP_TimeZone”

parameter in appendix A). In the Format: “YY/MM/DD-hh:mm:ss” for “Year/Month/Days-Hours:Minutes:Seconds”

- timestamp: Timestamp of data at acquisition point Y in UTC+0. Number of milliseconds elapsed since January 1, 1970.
- nbOfRefreshes: Number of complete readings over this acquisition period. This information is displayed only if the “MQTT_EnableAdvancedData” parameter is at 1. This data can only be useful for Modbus and Inverter devices. For IOs, TICs, virtual devices and for the parameter collection file, the number of refreshes is always 0.
- __MetaData__: Metadata on the context of the equipment present in the file. Currently the only information transmitted is the identifier of the WebdynSunPM in the “id” item.

In order to avoid sending unnecessary data to the server and thus optimizing the connection, it is advisable to activate only the variables that you wish to upload.

Equipment (Modbus, inverters):

Example of an equipment data file with an acquisition period every 10 minutes:

- Parameterization of the indexes of device 1 (modbusTCP):

Index	Code « Action »	Display
1	1	Parameter value
2	0	No
3-11	4	Instant value
12	8	Instant value

- Parameterization of the indexes of device 2 (SMANET):

Index	Code « Action »	Display
1-2	2	Min, max and average values

- MQTT data file in JSON format:


```
{
	"__MetaData__":{
		"id":"WPxxxxxx"
	}
	"modbusTCP":[
	{
		"var_1":32,
		"var_3":52,
		"var_4":5,
		"var_5":102,
		"var_6":1,
		"var_7":0,
		"var_8":1,
		"var_9":0,
		"var_10":0,
		"var_11":0,
		"var_12":0,
		"date":"21/02/05-09:50:00",
		"timestamp":1612515000000,
		"nbOfRefreshes":12
	},
	{
		"var_1":35,
		"var_3":57,
		"var_4":5,
		"var_5":108,
		"var_6":1,
		"var_7":10,
		"var_8":0,
		"var_9":0,
		"var_10":0,
		"var_11":0,
		"var_12":1,
		"date":"21/02/05-10:00:00",
		"timestamp":1612515600000,
		"nbOfRefreshes":12
	}
	],
	"SMANET":[
	{
		"var_1":[16,32,26.00],
		"var_2":[52,58,51.00],
		"date":"21/02/05-09:50:00",
		"timestamp":1612515000000,
		"nbOfRefreshes":2
	},
	{
		"var_1":[4,6,5.50],
		"var_2":[102,105,103.00],
		"date":"21/02/05-10:00:00",
		"timestamp":1612515600000,
		"nbOfRefreshes":2
	}
	]
}
```

Inputs/Output (IO):

Example of an IO data file with an acquisition period every 10 seconds:

- Input/output settings:

Entrées/Sortie	Code « Action »	Display
1	2	Instant value
2	0	No
3	2	Instant value
4	2	Instant value
5	2	Instant value
6	4	Min, max and average values
7	8	Instant value
8	2	Instant value

- MQTT data file in JSON format:


```
{
        "__MetaData__":{
                "id":"WPxxxxxx"
        },
        "IO":[
        {
                "var_1":0,
                "var_3":0,
                "var_4":5,
                "var_5":1,
                "var_6":[130,170,150],
                "var_7":5,
                "var_8":0,
                "date":"21/02/05-09:50:00",
                "timestamp":1612515000000,
                "nbOfRefreshes":0
        },
        {
                "var_1":0,
                "var_3":1,
                "var_4":2,
                "var_5":2,
                "var_6":[120,160,140],
                "var_7":3,
                "var_8":0,
                "date":"21/02/05-10:00:00",
                "timestamp":1612515600000,
                "nbOfRefreshes":0
        }
}
```

4.2.2 Alarms

For the concentrator to deposit alarms on the MQTT/MQTTS server, the Alarm topic must be filled in. The concentrator remains in permanent connection with the MQTT/MQTTS server in order to carry out the action immediately.

At the time of an alarm, no other data apart from that of the alarm is deposited on the server. The list of alarms that can be generated is:

Source Alarm	Info	Description
GATEWAY	Power ON Concentrator start	Démarrage du concentrateur
	Power OFF Switching off the concentrator	Extinction du concentrateur
	Loss of ICT accessory ICT accessory removed	Accessoire TIC retiré
	Return ICT accessory ICT accessory reconnected	Accessoire TIC reconnecté
IO	Definition File Name + Index + Value Value of an alarm type input has changed	Valeur d'une entrée de type alarme a changé
MODBUS	Definition file name + Index + Value Value of an alarm type index to change	Valeur d'un index de type alarme à changer

A "Power OFF" alarm is sent after a power outage of at least 10 seconds and a "Power ON" alarm is sent after the power has been restored for at least 1 minute. The other alarms are not delayed and are sent as soon as they are detected by the concentrator.

The format of alarm data in JSON format is as follows:

```
{
        "__MetaData__":{
            "id":"concentrator_id"
         },
         "alarms":[
        {
            "defName":defName_alarms,
            "deviceName":deviceName_alarms,
            "source":source_alarms,
            "value":value_alarms,
            "variableIndex":variableIndex_alarms,
            "date":"YY/MM/DD-hh:mm:ss",
            "timestamp":value_timestamp_alarms
        }],
        "alarmsDevice":[
        {
            "type":type_alarmsDevice,
            "info":info_alarmsDevice,
            "date":"YY/MM/DD-hh:mm:ss",
            "timestamp":value_timestamp_alarmsDevice
        }
}
```

Color code:

- In blue: Data that depends on the source of the alarm.
- In black: Fixed text.

With:

- alarms: Alarm with IO or MODBUS source.
- alarmsDevice: Alarm whose source is GATEWAY.
- defName_alarms: Name of the Definition file of the device or interface which triggered the alarm. (See chapter 4.1.4: "The "ALARM" alarms")
- deviceName_alarms: Name of the device or interface that triggered the alarm, "Name" field of the device in the <uid>_daq.csv file (See chapter 3.1.2.1.3.4: "Declaration of equipment")
- source_alarms: Alarm source (IO or MODBUS)
- value_alarms: Value of the variable that triggered the alarm.
- variableIndex_alarms: Index of the variable that triggered the alarm.
- type_alarmsDevice: Type of alarm (GATEWAY).
- info_alarmsDevice: Information on the alarm (see the "Info" column of the table above)
- date: Timestamp of the alarm in UTC+timezone (see the "NTP_TimeZone" parameter in appendix A). In the Format: "YY/MM/DD-hh:mm:ss" for "Year/Month/Days-Hours:Minutes:Seconds"
- timestamp: Timestamp of the alarm in UTC+0. Number of milliseconds elapsed since January 1, 1970.
- __MetaData__: Metadata on the context of the devices present in the file. Currently the only information transmitted is the identifier of the WebdynSunPM in the "id" item.

Example of an alarm on a device:

```
{
        "__MetaData__":{

            "id":"WPxxxxxx"
    },

    "alarms":[
      {
            "defName":"WPM00C44F_SunSpec_inverter_SMA_Solar_Inverter_9301_
modbusTCP.csv",
            "deviceName":"modbusTCP",
            "source":"MODBUS",
            "value":"106",
            "variableIndex":3,
            "date":"21/02/05-09:50:00",
            "timestamp":1612515000000
      }],
      "alarmsDevice":null
}
```

Example of an internal concentrator alarm:

```
{
        "__MetaData__":{

            "id":"WPxxxxxx"

      },
      "alarms":null,
      "alarmsDevice":[
      {
            "type":"POWER OFF",
            "info":"GATEWAY",
            "date":"21/02/05-09:50:00",
            "timestamp":1612515000000
      }
}
```

4.2.3 Commands

It is possible to send commands by the MQTT/MQTTS server to the WebdynSunPM concentrator.

To do this, the “command” and “Result” topics must be entered in the concentrator configuration (See chapter 3.2.2.5.4: “MQTT”).

When a command is published on the “Command” topic on the MQTT/MQTTS server, it is retrieved by the concentrator. The command is executed by the concentrator and the result of the command is published on the “Result” topic.

4.2.3.1 Update Command

The Update command is used to retrieve firmware from an HTTP/HTTPS or FTP/SFTP server.

The Update command file is in the following JSON format:

```
{
	"url":url_value,
	"login":login_value,
	"password":password_value,
	"checksum":checksum_value,
	"interface":interface_value,
}
```

Color code:

- In blue: Firmware or server information.
- In black: Fixed text.

With:

- url_value: IP address or domain name of the HTTP/HTTPS or FTP/SFTP server.
- login_value: Identifier for the FTP/SFTP server. The field must be "null" for an FTP/SFTP server.
- password_value: Password for the FTP/SFTP server. The field must be "null" for an FTP/SFTP server.
- checksum_value: Checksum of the new firmware in MD5 format to verify its integrity.
- interface_value: name of the network interface to use to access the remote server to retrieve the new firmware. Possible values are "ethernet" or "modem".

Example of updating against an HTTPS server:

```
{
	"rpcName":"sunpm.updateFirmware",
	"parameters":
	{
		"url":"https://www.webdyn.com/download/wgapp_3.3.666.35005.
spm",
		"checksum":"78a37fa7f6714876be7d08d0c39a067b",
		"interface":"modem",
	},
	"callerId":"667"
}
```

Example of updating against an SFTP server:

```
{
	"rpcName":"sunpm.updateFirmware",
	"parameters":
	{
		"url":"sftp://192.168.1.66:8081/download/wgapp_3.3.666.35005.
spm",
		"login":"login",
		"password":"password",
		"checksum":"78a37fa7f6714876be7d08d0c39a067b",
		"interface":"ethernet",
	},
	"callerId":"668"
}
```

4.3 microSD card

The use of the SD card follows the same operating rules as for the FTP server.

The only differences are:

- Directories not configurable. Default directories are used and these are the same on all installations.

The SD card is only read when the hub needs to access it. This means that all the rest of the time it can be removed without any consequences. The information is written as it is securely written to allow safe withdrawal of the card at any time.

The level of use of the SD card can be consulted at any time on the server configuration page in the insert dedicated to the SD card:

Webdyn does not provide any SD card. Please contact a computer hardware retailer.

5. Commands

5.1 Principle

It is possible to send commands to the WebdynSunPM. They allow you to perform tasks remotely for configuration, control or monitoring purposes. For example: launching a search for equipment, obtaining the current configuration, triggering a connection to the IS, etc. The same mechanism also makes it possible to invoke a function of a script installed on the concentrator.

5.2 How it Works

An order can be sent via three distinct methods:

- A command file deposited on the server (FTP, SFTP or WebDAV) which will be retrieved by the concentrator during the SI connection.
- An MQTT message posted on the WebdynSunPM control topic.
- An SMS sent to the WebdynSunPM SIM card.

5.2.1 Command file

During an FTP, SFTP or WebDAV connection, the WebdynSunPM checks for the presence of a command file in the directory configured for this purpose (/CMD by default). This file must be named <uid>_cmd.json where <uid> is the gateway identifier. The commands included in it are in the JSON format described below and are executed in order. Invocation of script functions is also supported. The file is deleted after import so as not to process the same order twice.

The results of commands are also written in files which will be deposited during the next SI connection. A file can contain one or more results. These are named according to the <uid>_ACK_<timestamp>.json pattern.

For an FTP, SFTP or WebDAV connection, the commands work only on server 1; the commands placed in the “/CMD” directory on a server 2 (backup) are not taken into account by the concentrator.

5.2.1.1 Command file JSON format

The format used for commands and invocations of script functions is as follows:

```
[{
    “rpcName”:”<nom du script>.<nom de la fonction>”,
    “parameters”:{ <paramètres de la fonction au format json> },
    “callerId”:”<identifiant commande 1>”
},
{
    “rpcName”:”<nom du script>.<nom de la fonction>”,
    “parameters”:{ <paramètres de la fonction au format json> },
    “callerId”:”<identifiant commande 2>”
},
…
]
```

Properties:

- rpcName: Name of the script and the function to be executed in the form <script name>.<function name>. For a command, we will use for the script name sunpm which is reserved for the internal commands of the WebdynSunPM.
- parameters: Some functions and commands require additional parameters. When not, this field is optional.
- callerId: An identifier associated with this request.

Each entry in the table corresponds to a different command. Note that if the file contains only one command, the square brackets are optional. The format of the results file is as follows:

```
[{
    "result":{ <éléments retournés par la fonction au format JSON> },
    "error":"<description de l'erreur en cas d'échec>",
    "callerId":"<identifiant commande 1>"
},
{
    "result":{ <éléments retournés par la fonction au format JSON> },
    "error":"<description de l'erreur en cas d'échec>",
    "callerId":"<identifiant commande 2>"
},
…
]
```

Properties:

- result: If the function or command is successful, it returns a result in JSON format contained in this field. In case of error this field is absent.
- error: In the event of an error, this field contains a description of the problem encountered. If the function or command is successful, this field is absent.
- callerId: The same identifier as in the request. So you can associate this answer to its original request.

Each entry in the array corresponds to a different command result.

5.2.1.2 Example

Let's start with the following script to illustrate the function invocation:

```
header = {
    version = 1.1,
    label = "Test",
    name = "test"
}
--[[
    Test function
]]
function testFunction(parameters)
    wd.log("question is " .. parameters.text)
    local result = {
        value = 42 = 42
    }
    return result
end
```

Command file for invoking the function:

```
[{
    "rpcName":"test.testFunction",
    "parameters": {
        "text":"what is the answer ?"is the answer ?"
    },
    "callerId":"3d9311ed-0076-4f28-ac59-a2debfa35b86"
}]
```

Result file obtained:

```
[{
    "result": {
        "value":42
    },
    "callerId":"3d9311ed-0076-4f28-ac59-a2debfa35b86"
}]
```

5.2.2 MQTT command message

The WebdynSunPM can receive command messages in MQTT. Of course, for this an MQTT server must be configured (See chapter 3.2.2.5.4: "MQTT"). In particular the topics of commands and results must be informed. The WebdynSunPM subscribes to the command topic so that any message sent on it is received and executed. The result of a command or function invocation is posted to the result topic.

The JSON format used for these messages is the JSON format described in chapter 5.2.1.1: "JSON format of the command file".

An MQTT message can contain only one command or result. Consequently, opening and closing square brackets are not allowed.

5.2.2.1 Utilisation with Azure IoT

Azure has its own function invocation mechanism. The JSON format used is therefore the following:

```
{
    "methodName":"<nom du script>.<nom de la fonction>",
    "responseTimeoutInSeconds":<délai avant timeout>,
    "payload":{
        "parameters":{ <paramètres de la fonction au format json> }
    }
}
```

Properties:

- methodName replaces rpcName.
- parameters is contained in the payload field of the Azure message.
- callerId is ignored.

5.2.3 SMS

It is possible to send SMS commands to the WebdynSunPM modem. To do this, check that the modem is correctly configured (See chapter 3.2.2.4: "Modem").

SMS commands do not use JSON format. Instead the accepted format is:

```
<commande 1>=<paramètre 1>:<paramètre 2>:<paramètre 3>… ;
<commande 2>=<paramètre 1>:<paramètre 2>:<paramètre 3>… ;
…
```

5.3 List of Commands

List of commands available on the hub:

Commands	Descriptions	SMS	MQTT/ MQTTS	Command file
connect	Trigger a connection	X	X	
status	Hub Status Recovery	X	X	X
factory	Return to factory settings	X		
reboot	Restarting the hub	X	X	X
updateFirmware	Hub software update	X	X	

apn	Modem Setup	X		
ftp	FTP/SFTP Server Setup	X		
log	Enabling Device Communications Logs	X	X	X
setRelay	Changing the state of the relay	X	X	X
discoverDevices	Discovery of equipment	X	X	X
getParameters	Collection of parameters, i.e. variables defined with action code 1	X	X	X
getData	Collection of variables defined with action code 6 or 7	X	X	X
writeVariable	Writing a variable on a device	X	X	X
setKey	Added keys for decrypting client scripts	X	X	X
deleteKey	Removing keys for decrypting client scripts	X	X	X

In the case of sending several simultaneous commands, the “factory” and “update” commands can cause the following commands to be lost. In case of error on a previous command, the following ones will be executed.

The available commands are described below with expected parameters and returned results.

5.3.1 "connect": Trigger a connection

Asks the WebdynSunPM to start a connection to the server. This then forces a synchronization of the settings with the server as well as the deposit of all log files. The connection is launched immediately.

Available via SMS only.

```
connect=<connexion>
```

Settings:

- connection: Number of the connection to establish. If there is no parameter, the connection will be made to server 1. The possible values are:
  - 1: Using server 1.
  - 2: Using server 2.

Return:

- If successful for an SMS command: no return.
- If an error is encountered: an explanatory message.

Examples SMS :

```
connect
```

or

```
connect=2
```

Example MQTT command:

```
{
    "rpcName":"sunpm.connect",
    "callerId":"1"
}
```

And answer:

```
{
    "callerId":"1",
    "result":"OK"
}
```

5.3.2 "status": Retrieval of the status of the concentrator

Returns information about the current configuration.

Available via command file (FTP, SFTP or WebDAV), MQTT/MQTTS message and SMS.

```
status
```

No SMS settings.

Example of SMS

```
status
```

By command file, the following parameter is taken into account:

- number: Telephone number that will receive the status SMS sent by the gateway. It is best to fill in the number with the international code.

A command acknowledgment file is sent to the server (FTP, SFTP, WebDAV, MQTT/MQTTS) that sent the command. The order information is sent to the phone number provided.

The "status" command sent by command file (FTP, SFTP, WebDAV) or by MQTT/MQTTS makes it possible to retrieve the concentrator's SIM card number.

Sample command file (FTP/SFTP/WebDav):

```
{
  "rpcName":"sunpm.status",
  "parameters":{
      "number": "+33700000000"
  },
  "callerId":"1"
}
```

The order information is returned in the form of an SMS, the details of which are as follows: type=WebdynSunPM3G

```
type=WebdynSunPM 3G
idSite=WPM01194D
FW=WebdynSunPM 4.2.12.39124
ModeServ1=modem
apn=tm
csq=26,99
network=3G
ip=192.168.1.12
mask=255.255.255.0
gw=
dns=
srv1:
type=ftp
addr=user@ftpserver.com:21
last=12/10 12:53 success
```

- *type:* Product name
- *idSite:* Corresponds to the name of the site, i.e. the identifier of the WebdynSunPM.

- *FW:* Version number of the firmware currently loaded on the concentrator.
- *ModeServ1*: Server connection mode (ethernet or modem).
- *apn*: Name of the apn configured for the modem.
- *csq:* Modem signal strength.
- *network*: Type of network used.
- *ip*: IP address configured on the Ethernet 1 interface.
- *mask*: Subnet mask configured on the Ethernet 1 interface.
- *gateway:* Address of the gateway allowing the concentrator to connect to an external network on the Ethernet 1 interface.
- *dns*: List of DNS servers used for name resolution for the Ethernet 1 interface. If several DNS servers are configured, they are separated by the '/' character.
- *type*: Type of server 1. The possible values are:
- *ftp*: The server is of the FTP type.
- *sftp*: The server is of the SFTP type.
- *webdav*: The server is WebDAV-HTTPS type.
- *mqtt:* The server is of MQTT type.
- *mqtts*: The server is of MQTTS type.
- *addr:* Server address in login@server-address:server-port format.
- *last:* Last date and time at which the WebdynSunPM connected to server 1 as well as the result of this connection in the following format: "DD/MM HH:MM <connection result>".

5.3.3 "factory": Return to factory settings

Resetting the WebdynSunPM. The configuration files as well as the equipment acquisition data are deleted and the product is restarted immediately.

Available via SMS only.

```
factory
```

No parameters.

Return:

- If successful for an SMS command: no return.
- If an error is encountered: an explanatory message.

Example SMS :

```
factory
```

5.3.4 "reboot": Rebooting the concentrator

Restarting the WebdynSunPM. This command runs immediately.

Available via command file (FTP, SFTP or WebDAV), MQTT/MQTTS message and SMS.

```
reboot
```

No parameters.

Return:

- If successful for an SMS command: no return.
- If an error is encountered: an explanatory message.

  Example SMS :

```
reboot
```

5.3.5 "updateFirmware": Concentrator software update

Retrieves firmware from a specified URL, validates it with a checksum and installs it. The command returns a success result just before proceeding with the installation. The result of the actual installation should be checked by the usual means of version control.

Available via MQTT/MQTTS message and SMS.

```
updatefirmware=<url>:<login>:<password>:<checksum>:<interface>
```

Settings:

- url: URL of the file to retrieve. Accepted protocols are HTTP, HTTPS, FTP, SFTP. The port can be specified using the address:port format.
- login: Server identifier.
- password: Server password.
- checksum: MD5 checksum of the file for validity verification.
- interface: Interface used for the connection: ethernet or modem.

Return:

- If successful: the message “Firmware downloaded successfully. System will restart...”.
- If an error is encountered: an explanatory message.

Example, command MQTT :

```
{
    “rpcName”:”sunpm.updateFirmware”,
    “parameters”:{
        “url”:”https://www.webdyn.com/download/wgapp_new_fw.spm”,
        “checksum”:”78a37fa7f6714876be7d08d0c39a067b”,
        “interface”:”modem”
    },
    “callerId”:”674”
}
```

And answer:

```
{
    “callerId”:”674”,
    “result”:”Firmware downloaded successfully. System will restart...”
}
```

Example SMS :

```
updatefirmware=ftp://ftp3.webdyn.com/wgapp_4.1.0.37427.
spm:login:webdyn:70a0eeeae295a7e16d3811b66bee9b66:modem
```

5.3.6 “apn”: Modem configuration

Modem APN configuration. This APN is required to establish a 2G/3G mobile connection. See modem configuration in chapter 3.2.2.4: “Modem” for more details.

Available via SMS only.

```
apn=<apn>:<login>:<password>
```

Settings:

- aapn: Name of the APN to use for the modem connection.
- login: User required for authentication on some APNs.
- password: Password required for authentication on some APNs.

Return:

- If successful for an SMS command: no return.
- If an error is encountered: an explanatory message.

Examples SMS :

```
apn=m
```

or

```
apn=m2minternet
```

5.3.7 "ftp": Configuration of the FTP/SFTP server

Configuration of the FTP/SFTP server "Server 1".

Available via SMS only.

```
ftp=<server>:<login>:<password>:<port>:<interface>
```

Settings:

- server: Name of the FTP server to connect to. This parameter can be a name or an IP address.
- login: User name on the specified FTP server.
- password: Password associated with the above user.
- port: FTP server port number. By default FTP servers use port 21.
- interface: Type of connection to use. The allowed values are:
  - ethernet: Uses the RJ45 Ethernet link to establish the connection with the FTP server. The Ethernet interface must have been configured beforehand for the connection to work (See chapter 3.2.2.3: "Networks").
  - modem: Use the modem to establish the connection with the FTP server. The modem must have been configured beforehand. Otherwise, it is possible to use the apn command to configure it by SMS (See chapter 5.3.6: "apn: Modem configuration").

Return:

- If successful for an SMS command: no return.
- If an error is encountered: an explanatory message.

Example SMS :

```
ftp=ftp3.webdyn.com:login:webdyn:70a0eeeae295a7e16d3811b66bee9b6621
```

5.3.8 "log": Activation of equipment communication logs

This command activates the device communications log system.

Available via command file (FTP, SFTP or WebDAV), MQTT/MQTTS message and SMS.

```
log=<interface>:<duration>
```

Settings:

- interface: The name of the interface on which to start the logs: ethernet, serial1, serial2 or serial3.
- duration: Duration in minutes during which the logs will be activated.

Return:

- If successful for a JSON command: "OK".
- If successful for an SMS command: no return.
- If an error is encountered: an explanatory message.

Example, command file:

```
[{
    "rpcName":"sunpm.log",
    "parameters":{
        "interface":"ethernet",
        "duration":2
    },
    "callerId":"672"
},
{
    "rpcName":"sunpm.log",
    "parameters":{
        "interface":"ethernet",
        "duration":1
    },
    "callerId":"673"
}]
```

And answer:

```
[
    {"callerId":"672", "result":"OK"},
    {"callerId":"673", "error":"Invalid interface: ethernet"}
]
```

Example SMS :

```
log=serial1:5
```

5.3.9 "setRelay": Changing the state of the relay

Modifies the state of the relay: open, close or 1s pulse.

Available via command file (FTP, SFTP or WebDAV), MQTT/MQTTS message and SMS.

```
setrelay=<action>
```

Settings:

- action: Action to execute: open, close or pulse.

Return:

- If successful for a JSON command: "OK".
- If successful for an SMS command: no return.
- If an error is encountered: an explanatory message.

Example SMS :

```
setrelay=pulse
```

5.3.10 "discoverDevices": Discovery of equipment

Triggers a device discovery.

Available via command file (FTP, SFTP or WebDAV), MQTT/MQTTS message and SMS.

```
discoverdevices=<protocol>:<maxDevices>:<interface>:<timeout>:<port>
```

Settings:

- protocol: Discovery protocol: sunspec, tic, (see specific appendix for proprietary protocols).
- maxDevices: Maximum number of devices to discover. When this number is reached, discovery ceases. This parameter is ignored in the case of the TIC protocol.
- interface: Interface used for discovery: serial1, serial2, serial3, ethernet1 or ethernet2. This parameter is ignored in the case of the TIC protocol. Only the SunSpec protocol is compatible with ethernet1 and ethernet2 values.
- timeout: Maximum duration (in milliseconds) for the discovery of a device. This parameter is ignored in the case of the TIC protocol.
- port: Port used for network discovery. This setting is only meaningful for SunSpec discovery over Ethernet. It is ignored otherwise.

Return:

- If successful: the number of equipment discovered.
- If an error is encountered: an explanatory message.

In SMS, the detect command is also available but it is limited to SunSpec detection (see above).

Example SMS :

```
discoverdevices
```

5.3.11 "getParameters": Collection of parameters

When this command is received by the WebdynSunPM, the variables defined with an action code 1 (ACTION_PARAMETER) are collected. At the next SI connection, the data will be dropped into a file named <uid>_<interface>_P_<timestamp>.csv.gz.

Available via command file (FTP, SFTP or WebDAV), MQTT/MQTTS message and SMS.

```
getparameters
```

No parameters.

Return:

- If successful for a JSON command: “OK”.
- If successful for an SMS command: no return.
- If an error is encountered: an explanatory message.

Example SMS :

```
getparameters
```

5.3.12 “getData”: Collection of action code variables 6 or 7

When this command is received by the WebdynSunPM, the variables defined with an action code 6 or 7 are collected. The value read here for each variable is the last read, even for an action code 7. The min, max, average type values are only calculated in the acquisition file. On the next SI connection, the snapshot data will be dumped into a file named <uid>_<interface>_I_<timestamp>.csv.gz.

In addition, if an MQTT server is configured, the connection to it is made automatically so that this data is deposited immediately.

If data is available in addition to instant data, it will also be deposited on the remote server with the usual naming.

Available via command file (FTP, SFTP or WebDAV), MQTT/MQTTS message and SMS.

```
getdata
```

No parameters.

Return:

- If successful for a JSON command: “OK”.
- If successful for an SMS command: no return.
- If an error is encountered: an explanatory message.

Example, command file:

```
[{
    “rpcName”:”sunpm.getData”,
    “callerId”:”1”
}]
```

And answer:

```
[
    {"callerId":"1", "result":"OK"}
]
```

5.3.13 "writeVariable": Writing a variable on a device

Writing a variable declared in a definition file.

Available via command file (FTP, SFTP or WebDAV), MQTT/MQTTS message and SMS.

```
setkey=<url>:<login>:<password>:<interface>
```

Settings:

- deviceName: Name of the targeted device.
- tagName: Target tag.
- value: Value to write. It can be a number or a character string. Note that in SMS the value will be interpreted as a number if possible. To force the interpretation in character string, it is possible to frame the value between "".

Return:

- If successful for a JSON command: "OK".
- If successful for an SMS command: no return.
- If an error is encountered: an explanatory message.

Example SMS :

```
writevariable=INV1:
```

or

```
writevariable=INV1:setLimit:30
```

5.3.14 "setKey": Added keys for decrypting client scripts

This command adds the keys for decrypting client Lua scripts. The keys must be in a specific format and contained in a file. The file containing the keys This must be made available on a server in order to be downloaded by the command. For details on the key format for Lua scripts, see the "WebdynSunPM LUA User Guide.pdf" document.

Available via command file (FTP, SFTP or WebDAV), MQTT/MQTTS message and SMS.

```
setkey=<url>:<login>:<password>:<interface>
```

Settings:

- url: URL of the file to retrieve. Accepted protocols are HTTP, HTTPS, FTP, SFTP. The port can be specified via the format address:port
- interface: Interface used for the connection: ethernet or modem
- login: Server identifier
- password: Server password

Return:

- If successful for a JSON command: “OK”.
- If successful for an SMS command: no return.
- If an error is encountered: an explanatory message.

Example, command file:

```
[{
    “rpcName”:”sunpm.setKey”,
    “parameters”:{
        “url”:”ftp://ftp.webdyn.com/script_key.json”,
        “interface”:”ethernet”,
        “login”:”login”,
        “password”:”pwd”
    },
    “callerId”:”203”
}]
```

OK response:

```
[
    {“callerId”:”203”, “result”:”OK”}
]
```

Or reply with an error:

```
[
    {“callerId”:”203”, “error”:”Invalid interface: ethernet”}
]
```

SMS example:

```
setkey=ftp://ftp.webdyn.com/script_key.json:ethernet:login:pwd5
```

5.3.15 "deleteKey": Removing keys for decrypting client scripts

This command removes keys for decrypting client Lua scripts.

If scripts in ".luax" format are present after the decryption keys are deleted, they will continue to work as long as the script remains active and the hub is not restarted. It is strongly recommended to delete the ".luax" scripts after deleting the keys.

Available via command file (FTP, SFTP or WebDAV), MQTT/MQTTS message and SMS.

```
deletekey
```

Return:

- If successful for a JSON command: "OK".
- If successful for an SMS command: no return.
- If an error is encountered: an explanatory message.

Example, command file:

```
[{
    "rpcName":"sunpm.deleteKey",
    "callerId":"117"
}]
```

OK response:

```
[
    {"callerId":"117", "result":"OK"}
]
```

SMS example:

```
deletekey
```

6. Update

The WebdynSunPM concentrator can be updated locally using the web interface or remotely by FTP, SFTP or WebDAV-HTTPS. The latest firmware version (“WebdynSunPM_x.x.x.zip”) is available for downloading from our web site at the following address: https://www.webdyn.com/support/WebdynSunPM/

Once the download is complete, unzip the file with contains 2 files:

- “wgapp_x.x.x.xxxxx.spm” which is the concentrator firmware
- “CheckSumx.x.x.txt” which contains the firmware checksum.

6.1 Using the Web Interface

To update the concentrator locally, go to the “System” tab on its web interface, then “Update” and follow the update procedure using the web interface (see section 3.2.3.2: “Update”).

6.2 Using FTP/SFTP/WebDAV

For remote updates, follow the steps below:

- Place the “wgapp_x.x.x.xxxxx.spm” file containing the updates in the “BIN” directory on the remote server.
- Edit the “<uid>_config_.ini” file (<uid>: Concentrator identifier) which is in the “CONFIG” directory on the server. Put the file name that has just been uploaded to the “BIN” directory in the “BIN_FileName” variable and enter the checksum indicated in the “CheckSumx.x.x.txt” file in the “BIN_Checksum” variable.

The concentrator will retrieve its configuration file and its new firmware at the next connection to the server.

The update using can only be carried out by the primary FTP, SFTP or WebDAV-HTTPS server.

The application of the update can be seen in the LOG files uploaded to the “LOG” directory on the server.

Once the update has been applied, the firmware file can be deleted and the “BIN_FleName” and “BIN_Checksum” variables in the “<uid>_config.ini” file can be emptied.

If there is an error during the update, it will not be re-attempted.

The failure to follow the order of the previously described steps will lead to the failure of the concentrator update.

6.3 By SMS or MQTT/MQTTS command

The “updateFirmware” command allows the update of the WebdynSunPM by specifying a URL where to find the new firmware. For details of the procedure, see chapter 5.3.5: “”updateFirmware”: Updating concentrator software”.

6.4 By micro SD card

For an update via the SD card, follow these steps:

-
- Put the “wgapp_x.x.x.xxxxx.spm” file containing the update in the “\BIN” directory of the SD card.
- Edit the “<uid>_config_.ini” file (<uid>: Concentrator identifier) which is located in the “CONFIG” directory of the SD card.

  Put the name of the file that has just been deposited in the “BIN” directory in the “BIN_FileName” variable and fill in the checksum indicated in the “CheckSumx.x.x.txt” file in the “BIN_Checksum” variable.

The concentrator will recover its configuration file then its new firmware at the next “connection” on the SD card.

The update is only possible in this way if the SD card is configured as the main medium

The application of the update can be seen in the LOG files deposited in the “LOG” directory of the SD card.

After applying the update, it is possible to delete the firmware file and put the empty field of the “BIN_FileName” and “BIN_Checksum” variables of the configuration file “<uid>_config_.ini”.

If an error occurs during the update, it will not be retried.

Failure to follow the order of the steps described in the procedure will lead to a failure of the concentrator update.

Webdyn does not provide any SD card. Please contact a computer hardware retailer.

7. Tools & Diagnostics

7.1 Diagnostics

If the product malfunctions, there are several ways to troubleshoot the faults.

- First of all, if the concentrator is connected by the modem with a SIM card, it is possible to request a status from the product by sending it the “status” command. (See chapter 5.3.5: “”updateFirmware”: Concentrator software update”)
- The log files described in chapter 4.1.8: “The “LOG” logs allow you to view the various errors and understand their causes. There are log files for each hub feature to better isolate issues.
- It is also possible to diagnose Ethernet or serial communication errors using the built-in tools. These tools can be activated by FTP (See chapter 5.3.8: “”log”: Activation of equipment communication logs”), by SMS (See chapter 5.3.8: “”log”: Activation of equipment communication logs ”) or via the WEB interface (See chapter 3.2.1.3: “Equipment diagnostic tools”).
- On the “Home” page of the WEB interface, the “Device info” section allows you to see all the equipment and quickly detect those with anomalies. (see chapter 3.2: “Embedded web interface”)

If necessary, the WebdynSunPM can be restarted remotely using the “reboot” SMS command (See chapter 5.3.4: ““reboot”: Rebooting the concentrator”).

Finally, it is possible to carry out a total reset of the product by sending the “factory” SMS command (See chapter 5.3.3: ““factory”: Return to factory settings”).

7.2 Tools

Definition File Converter

As the file format has changed between these two products, a tool is available to convert the WebdynSun definition files to WebdynSunPM definition files.

This tool can be downloaded from the Webdyn web server at the following address: www.webdyn.com/download/DefFileConverter.zip

For any questions, contact support: support@webdyn.com

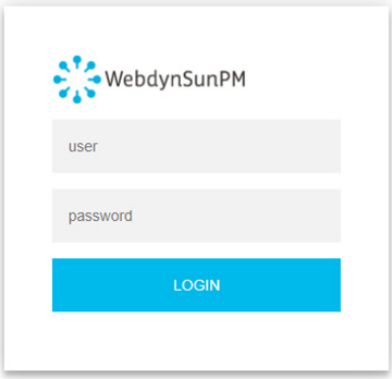
8. FAQ

Gateway configuration:

How do you access the gateway configuration?

Start by checking that the computer's IP parameters are compatible with the WebdynSunPM IP address (by default 192.168.1.12)

Launch a web browser (Chrome, Firefox, Edge, Safari, etc.) and enter the WebdynSunPM concentrator IP address in the address bar. An authentication page is displayed:



The default accesses are:

LOGIN	PASSWORD
userhigh	high

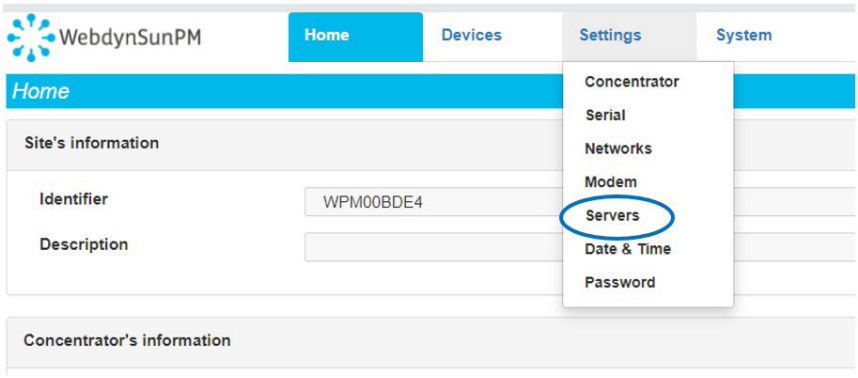
Click "Login"

How do you configure the WebdynSunPM concentrator to access the remote server?

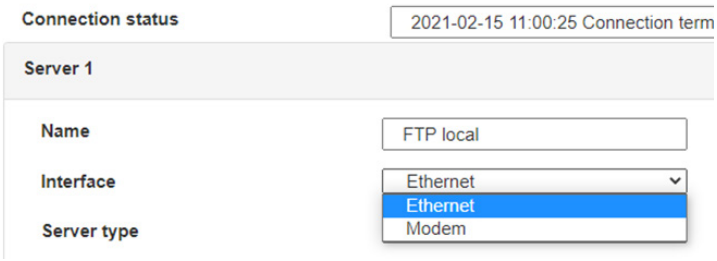
There are two configuration solutions, using the web interface and using text messages:

- Configuration using the web interface:

Start by establishing a connection to the concentrator by connecting to it to access the server configuration:

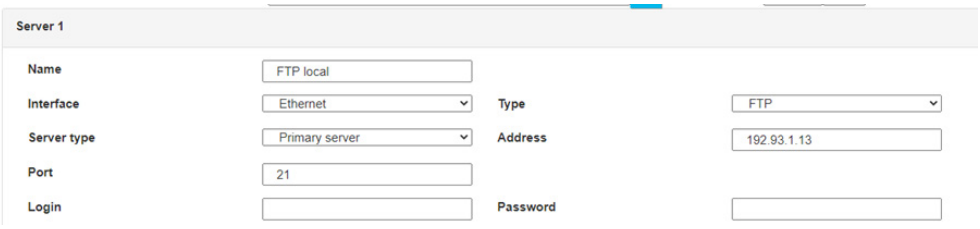


Enter the “ethernet” or “modem” connection type:



For an ethernet configuration, make sure the IP parameters are compatible with server access according to the concentrator local network configuration. For an ethernet connection, the configuration must be compatible with the concentrator’s local network topology so that it can access the servers. This configuration is done from the “Networks” configuration page (see section 3.2.2.3: “Networks”).

For a modem connection, the modem configuration must be correct before a connection can be set up. This configuration is done from the “Modem” configuration page (see section 3.2.2.4: “Modem”). The parameters for the servers to be configured are at least the following:



Therefore the following fields need to be configured: “Interface”, “Type”, “Server type”, “Address”, “Port”, “Login” and “Password”. The other fields can be left at the default values subject to the directories having been properly created beforehand. See section 3.1.2: “Configuration files” for more details.

- Text message configuration: text message configuration requires sending the following commands:
  - apn: to configure the SIM card APN. (see section 5.3.6: ““apn” modem configuration command. Mistake! Referral source not found).

```
apn <apn> <login> <password>
```

  - ftp: to configure the FTP server that will contain the concentration configuration (see section 5.3.7: ““ftp” FTP/SFTP configuration command”. Mistake! Referral source not found).

```
ftp <server> <login> <password> <port> <interface>
```

  - connect: to launch the connection to the FTP server and load the configuration (see section 5.3.1: ““connect” connection command”).

```
connect <connection>
```

For configuration by SMS, you must not add spaces between the parameters. The syntax must be strictly identical.

What are the FTP server access identifiers?

Access to the FTP server depends on the selected solution.

If you have chosen a portal, it will give you the FTP server access identifiers.

If you want to use your own FTP server, contact your network administrator.

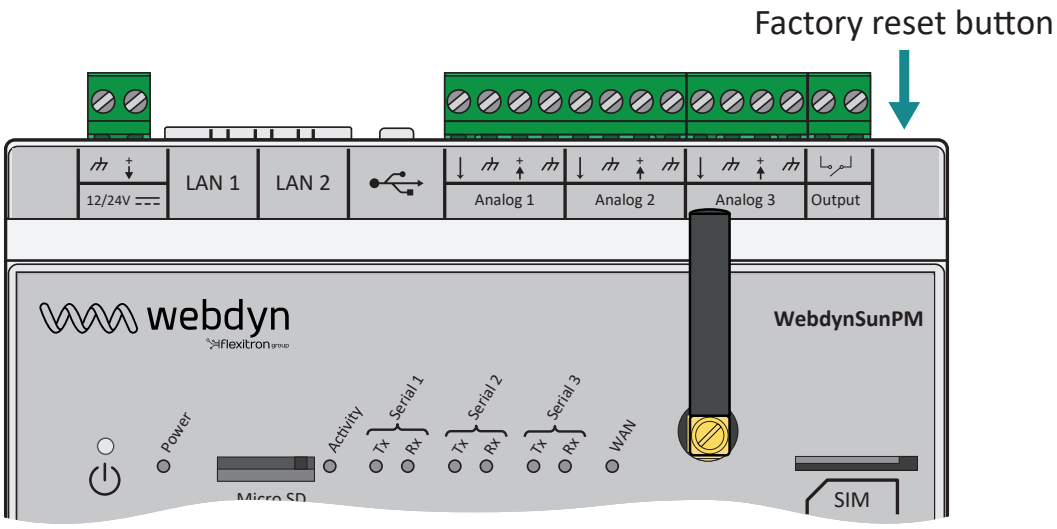
For all other configurations, and to determine the best solution, contact the Webdyn sales department which will advise you and direct you to the relevant contacts: contact@webdyn.com.

General gateway use

How do you reinitialise the concentrator?

There are 2 methods to force a concentrator factory reset:

- Press the Factory Reset button on the concentrator for 20 seconds:



  Wait. The concentrator will reboot using its factory configuration.

- If a SIM card is installed and configured, a “factory” SMS also enables a factory reset. Simply send it to the phone number of the SIM card (see chapter 5.3.3: “Factory return command”. Mistake! Referral source not found)

The factory reset restores the original configuration. The data is not kept.

Can the gateway send commands to connected devices?

It is possible to send commands to connected devices if they accept them.

How long can data be stored for?

The WebdynSunPM can store up to 50Mb of uncompressed data per declared device.

If there is no access to the remote server, the WebdynSunPM concentrator can store the data for several months.

The maximum data storage time varies depending on the amount of data to be collected and the configured collection frequency.

The average storage time varies from 3 to 4 months.

What is the battery life?

The average service life of the battery is 5 years.

It may vary depending on the installation environment.

What are the warranty conditions?

All our products are guaranteed for 2 years.

For more information, read the general terms and conditions of sale.

What is the volume of data exchanged by the modem?

The data volume depends on the exchanged files.

The average is about 5 MB per month but this varies from one installation to another.

Inverter compatibility

Which inverters are compatible?

See section 1.4: “Supported devices”.

Modbus device compatibility:

Can I connect different Modbus devices to the same serial port?

Yes, different Modbus devices can be connected to the same serial port.

Device compatibility:

- Same type of RS485 or 4 wire connection.
- All devices should be able to be configured using identical bus specifications. Same speed, same parity, same number of stop bits and data bits on all devices and on the WebdynSunPM.
- Each device must be assigned a unique Modbus address (between 1 and 247) on the bus. (UnitID).

9. Appendices

9.1 Appendix A: "_config.ini" file

The authorised configuration parameters in the "<uid>_config.ini" file are the following:

PARAMETER	DESCRIPTION	DEFAULT VALUE
BIN_Checksum	Indicates the firmware name to use to update the gateway software. This parameter cannot be empty if BIN_FileName contains a value. The firmware thus named must be in the configured binary directory. See the section on update configuration for how to use updates.	
BIN_FileName	Indicates the checksum for the firmware indicated in the BIN_FileName field. This parameter cannot be empty if BIN_Checksum contains a value. See the section on update configuration for how to use updates.	
Concentrator_Identifier	Gateway identifier (uid). If this field is left empty, the default identifier is used.	WPMABCDEF (where ABCDEF are the last characters of the serial number).
FTP_DirAlarm	FTP 1 server directory in which the alarms from the concentrator are stored. Note that the directory MUST exist. The concentrator will not create it when uploading files.	/ALARM
FTP_DirBin	FTP 1 server directory in which the concentrator will get the update files when requested. See the section on update configuration for how to use updates.	/BIN
FTP_DirCert	FTP 1 server directory in which the concentrator will get the certificates to use for MQTT connections. See the MQTT configuration section for more details.	/CERT

FTP_DirCmd	FTP 1 server directory in which the concentrator will get the command files used later. See the command file configuration use section for more details.	/CMD
FTP_DirConfig	FTP 1 server directory to which the concentrator uploads its configuration files. Note that the directory MUST exist. The concentrator will not create it when uploading files. The concentrator will also reread the configuration files to detect the updates to download and apply. See the section on concentrator configuration using concentrator files for the operating principle (see section 3.1.2: “Concentrator operation”).	/CONFIG
FTP_DirData	FTP 1 server directory to which the concentrator uploads the data files collected during operation Note that the directory MUST exist. The concentrator will not create it when uploading files.	/DATA
FTP_DirDef	FTP 1 server directory to which the concentrator uploads the definition files it creates. Note that the directory MUST exist. The concentrator will not create it when uploading files. The concentrator will also reread the configuration files to detect the updates to download and apply. See the section on definition file configuration for the operating principle (see section 3.1.2.2: “Connected device definition”).	/DEF
FTP_DirLog	FTP 1 server directory to which the concentrator uploads the generated log files. Note that the directory MUST exist. The concentrator will not create it when uploading files. See the section on logs for the different available files (see section “4.1.8 “LOG””).	/LOG

FTP_DirScript	FTP 1 server directory to which the concentrator uploads and rereads the script files currently in use. If a file is loaded onto the concentrator, it will be transferred at the next connection to the FTP server. If a file is modified on the FTP server, it is loaded on the concentrator at the next connection. If a file is added to the FTP server, it is ignored.	/SCRIPT
FTP_DisconnectDelay	Time the connection to the FTP 1 server is kept alive. This value is expressed in seconds. A value of 0 indicates that the connection is ended as soon as processing completes.	0
FTP_EnableAdvancedData	Addition of the number of complete readings over this acquisition period in the data files deposited on the FTP/SFTP server: •0: No addition of the number of complete readings •1: Addition of the number of complete readings	0
FTP_EuroDateFormat	Indicates the format used to timestamp the data sent to the FTP 1 server. The possible values are: •0: the ISO format is used (YY/MM/D-HH:MM:SS) •1: the European format is used (DD/MM/YY-HH:MM:SS)	0
FTP_HeaderOption	Indicates whether the gateway must add the optional headers in the data files uploaded to the FTP 1 server. The possible values are: •0: no optional header •1 : optional headers generated	0
FTP_Login	The login to use to connect to the FTP 1 server. This value is mandatory.	
FTP_Password	The password configured for the login configured in the “FTP_Login” parameter to connect to server FTP 1.	
FTP_Port	The network port to use to connect to the FTP 1 server.	21
FTP_ SynchroniseCertificates	(Future upgrade)	0

FTP_ TwoStepsSendingDisabled	Used for file transfers in 2 steps using a temporary file while the file is not complete on the FTP 1 server. The possible values are: • 0: the temporary file is used • 1: the temporary file is not used	0
FTP_UploadLog	Indicates whether the gateway must also upload the internal programmed connection operating logs in the FTP 1 server log upload directory. The possible values are: • 0: no file upload • 1: file upload	0
FTP_WebServicesEnable	Indicates whether web services are enabled on the FTP interface. See the section dedicated to web services for more information on the use of this feature. The possible values are: • 0: web services disabled • 1: web services activated	0
FTP_WebServicesUrl	URL that will be called in response to certain FTP actions if web services are enabled. See the section dedicated to web services for more information on the use of this feature. The URL must have the following format: • http://address/page/	
FTP2_DirAlarm	FTP 2 server directory in which the alarms from the concentrator are stored. Note that the directory MUST exist. The concentrator will not create it when uploading files.	/ALARM
FTP2_DirBin	FTP 2 server directory in which the concentrator will get the update files when requested. See the section on update configuration for how to use updates.	/BIN
FTP2_DirCert	FTP 2 server directory in which the concentrator will get the certificates to use for MQTT connections. See the MQTT configuration section for more details.	/CERT
FTP2_DirCmd	FTP 2 server directory in which the concentrator will get the command files used later. See the command file configuration use section for more details.	/CMD

FTP2_DirConfig	FTP 2 server directory to which the concentrator uploads its configuration files. Note that the directory MUST exist. The concentrator will not create it when uploading files. See the section on concentrator configuration using concentrator files for the operating principle (see section 3.1.2: “Concentrator operation”).	/CONFIG
FTP2_DirData	FTP 2 server directory to which the concentrator uploads the data files collected during operation. Note that the directory MUST exist. The concentrator will not create it when uploading files.	/DATA
FTP2_DirDef	FTP 2 server directory to which the concentrator uploads the definition files it creates. Note that the directory MUST exist. The concentrator will not create it when uploading files. See the section on definition file configuration for the operating principle (see section “2. Connected device definition”).	/DEF
FTP2_DirLog	FTP 2 server directory to which the concentrator uploads the generated log files. Note that the directory MUST exist. The concentrator will not create it when uploading files. See the section on logs for the different available files (see section “4.1.8 “LOG””).	/LOG
FTP2_DirScript	FTP 2 server directory to which the concentrator uploads and rereads the script files currently in use. If a file is loaded onto the concentrator, it will be transferred at the next connection to the FTP server. If a file is modified on the FTP server, it is loaded on the concentrator at the next connection. If a file is added to the FTP server, it is ignored.	/SCRIPT
FTP2_EuroDateFormat	Indicates the format used to timestamp the data sent to the FTP 2 server. The possible values are: •0: the ISO format is used (YY/MM/D-HH:MM:SS) •1: the European format is used (DD/MM/YY-HH:MM:SS)	0

FTP2_HeaderOption	Indicates whether the gateway must add the optional headers in the data files uploaded to the FTP 2 server. The possible values are: •0: no optional header •1 : optional headers generated	0
FTP2_ EnableAdvancedData	Addition of the number of complete readings over this acquisition period in the data files deposited on the FTP/SFTP server: •0: No addition of the number of complete readings •1: addition of the number of complete readings	0
FTP2_Login	The login to use to connect to the FTP 2 server. This value is mandatory.	
FTP2_Password	The password configured for the login configured in the “FTP2_Login” parameter to connect to server FTP 2.	
FTP2_Port	The network port to use to connect to the FTP 2 server.	21
FTP2_ SynchroniseCertificates	(Future upgrade)	0
FTP2_ TwoStepsSendingDisabled	Used for file transfers in 2 steps using a temporary file while the file is not complete on the FTP 2 server. The possible values are: •0: the temporary file is used •1: the temporary file is not used	0
FTP2_UploadLog	Indicates whether the gateway must also upload the internal programmed connection operating logs in the FTP 2 server log upload directory. The possible values are: •0: no file upload •1: file upload	0
FTP2_WebServicesEnable	Indicates whether web services are enabled on the FTP 2 interface. See the section dedicated to web services for more information on using this feature. The possible values are: •0: web services disabled •1: web services activated	0

FTP2_WebServicesUrl	URL that will be called in response to certain FTP 2 actions if web services are enabled. See the section dedicated to web services for more information on the use of this feature. The URL must have the following format: • http://address/page/
MQTT2_AlarmTopic	Name of the alarm topic to be published. If the field is empty, no alarm will be published to the broker. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTT server. Works for all MQTT types except **“mqtts_azure”**
MQTT2_CaCertFile	Name of the certificate used to authenticate the specified MQTTS server. The certificate is to be retrieved from your MQTT server and must be imported to the concentrator by FTP or by the web interface. Works for all MQTT types except **“mqtt”**
MQTT2_CertFile	Name of the hub-specific certificate used for the connection. The certificate is to be retrieved from your MQTT server and must be imported to the concentrator by FTP or by the web interface. Functional for all MQTT types except **“mqtt”** and **“mqtts_gcloud”**
MQTT2_CloudDevice	Customizable unique identifier of the equipment in a register defined on the MQTT server. This parameter is to be retrieved from your MQTT server and corresponds to: • “deviceId” on Google IoT Cloud. • “device_id” on Azure IoT Hub. Functional only for **“mqtts_gcloud”** and **“mqtts_azure**” types
MQTT2_CloudProjectId	Customizable unique identifier of the project defined on the MQTT server. This parameter is to be retrieved from your MQTT server and corresponds to: • “projectId” on Google IoT Cloud. • The name of the “Iot Hub” on Azure IoT Hub. Functional only for **“mqtts_gcloud”** and **“mqtts_azure”** types

MQTT2_CloudRegion	Device registry MQTT server region. This parameter is to be retrieved from your MQTT server and corresponds to: • "deviceRegistryLocation" on Google IoT Cloud. For example: "europe-west1" Functional only for **"mqtts_gcloud"** type	
MQTT2_CloudRegistry	Customizable registry name defined on the MQTT server. This parameter is to be retrieved from your MQTT server and corresponds to: • "deviceRegistryId" on Google IoT Cloud. Functional only for **"mqtts_gcloud"** type	
MQTT2_CloudSigningAlgo	Type of key used to verify the signature of the MQTT server certificate. This parameter is to be retrieved from your Google IoT Cloud server. The possible values are: • "RS256" for the RSA key • "ES256" for the elliptical curve key Functional only for **"mqtts_gcloud"** type	
MQTT2_ClientId	Customizable identifier of the equipment on the MQTT server. This parameter is to be retrieved from your MQTT server. Functional only for **"mqtt"** and **"mqtts"** types	
MQTT2_ControlTopic	Name of the topic for the commands to be retrieved by the concentrator. The MQTT2_ResultTopic parameter must be populated for using commands. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTT server. Works for all MQTT types except **"mqtts_azure"**	
MQTT2_ EnableAdvancedData	Publication of the number of complete readings over this acquisition period in the data topic. The possible values are: • 0: Disabled • 1: Enabled	0

MQTT2_EnableAlarmPost	Activate the publication of alarms on the dedicated topic. The possible values are: •0: Publication of alarms is disabled •1: The publication of alarms is activated and the concentrator remains in permanent connection mode with the MQTT server. Functional only for **“mqtts_azure”** type	
MQTT2_ EnableInvokeMethod	Enable method calling. Allows the use of dedicated topics. The possible values are: •0: Method call disabled •1: Method call activated and the concentrator remains in permanent connection mode with the MQTT server. Functional only for **“mqtts_azure”** type	
MQTT2_Insecure	Disable verification of the host name specified in certificates. The possible values are: •0: Verification enabled •1: Verification disabled. Functional only for **“mqtts”** type	
MQTT2_KeepAlive	If no exchange made with the MQTT server for the time defined in seconds, the concentrator sends a ping to the MQTT server in order to check the connection to it. If the value is “0”, KeepAlive is disabled. If the hub is in persistent connection mode with the MQTT server and a disconnection is detected after a KeepAlive, the hub will automatically reconnect to the MQTT server.	10
MQTT2_KeyFile	Name of the file including the private key specific to the concentrator used for the connection. The file is to be retrieved from your MQTT server and must be imported to the concentrator by FTP or by the web interface. Works for all MQTT types except **“mqtt”**	
MQTT2_Login	Username to access the MQTT server. Functional only for **“mqtt”** and **“mqtts”** types	

MQTT2_Password	Password to access the MQTT server. Functional only for **“mqtt”** and **“mqtts”** types	
MQTT2_Port	MQTT server port. For an MQTT server, the default port is 1883. For an MQTTS secure server, the default port is 8883.	8883
MQTT2_QoS	Guaranteed service number for sending messages (Quality Of Service). The possible values are: • 0: The message will be sent only once, i.e. with no guarantee of receipt. • 1: The message will be sent at least once, i.e. the concentrator will send several times if necessary until the broker confirms that it has been sent. •2: The message will necessarily be saved by the concentrator and will always send it as long as the broker does not confirm its sending. (avoids duplication of messages) For **“mqtts_gcloud”, “mqtts_azure”**, and **“mqtts_aws”** types, QoS 2 is not supported.	1
MQTT2_ResultTopic	Name of the topic for the results of the commands passed to the concentrator. The MQTT2_ControlTopic parameter must be populated for using commands. If a topic name is entered, the concentrator remains in permanent connection mode with the MQTT server. Works for all MQTT types except **“mqtts_azure”**	
MQTT2_Timeout	Maximum wait time in seconds for the response from the MQTT server. If the server has not responded within the allotted time, the send is stopped and retried on the next schedule. Functional only in QoS 1 or QoS 2.	30
MQTT2_TlsVersion	TLS version supported by the MQTT server. The possible values are: •tlsv1.1: TLS in V1.1 •tlsv1.2: TLS in V1.2 Functional only for **“mqtts”** type	tlsv1.2

MQTT2_Topic	Name of the topic for the data deposited by the concentrator. Works for all MQTT types except “mqtts_azure”	
HTTP_DirAlarm	Directory of the WebDAV-HTTPS 1 server in which the alarms coming from the concentrator will be stored. Note that the directory MUST exist. It will not be created by the hub during file uploads.	/ALARM
HTTP_DirBin	WebDAV-HTTPS 1 server directory where the hub will look for update files when requested. See the section on configuring updates for how to use updates. (3.1.2.1)	/BIN
HTTP_DirCert	WebDAV-HTTPS 1 server directory where the hub will look for certificates to use for MQTT connections. See the section on WebDAV server configuration for more details.	/CERT
HTTP_DirCmd	Directory of the WebDAV-HTTPS 1 server in which the concentrator will search for the command files which will be used later. See the section on using command files for more details	/CMD
HTTP_DirConfig	Directory of the WebDAV-HTTPS 1 server in which the concentrator places its configuration files. Note that the directory MUST exist. It will not be created by the hub during file uploads. The hub will also replay configuration files to detect updates to download and apply. See the section devoted to the configuration of the concentrator by the configuration files for the principle of operation (see chapter 3.1.2: “Operation of the concentrator”)	/CONFIG
HTTP_DirData	Directory of the WebDAV-HTTPS 1 server in which the concentrator will store the data files collected during operation. Note that the directory MUST exist. It will not be created by the hub during file uploads.	/DATA

HTTP_DirDef	Directory of the WebDAV-HTTPS 1 server in which the concentrator will deposit the definition files created by itself. Note that the directory MUST exist. It will not be created by the hub during file uploads. The hub will also replay configuration files to detect updates to download and apply. See the section dedicated to the configuration of definition files for the operating principle (see chapter 3.1.2.2: “Definition of connected devices”)	/DEF
HTTP_DirLog	Directory of the WebDAV-HTTPS 1 server in which the concentrator will store the generated log files. Note that the directory MUST exist. It will not be created by the hub during file uploads. See the section dedicated to logs for the different files available (see chapter 4.1.8: “Logs “LOG””)	/LOG
HTTP_DirScript	Directory of the WebDAV-HTTPS 1 server in which the concentrator will deposit and reread the script files currently in use If a file is uploaded to the hub, it will be transferred the next time you connect to the server. If a file is modified on the server, it is uploaded to the hub on the next connection. If a file is added to the server, it is ignored.	/SCRIPT
FTP_DisconnectDelay	FTP server connection hold time 1. This value is expressed in seconds. A value of 0 indicates that the connection is cut as soon as the processing ends.	0
HTTP_EuroDateFormat	Specifies the format to use for the timestamp of data sent to the WebDAV-HTTPS server 1. The possible values are: •0: ISO format is used (YY/MM/DD-HH:MM:SS) •1: the European format is used (DD/MM/YY-HH:MM:SS)	0

HTTP_HeaderOption	Indicates whether the gateway must add optional headers in the data files uploaded to the WebDAV-HTTPS 1 server The possible values are: •0: no optional header •1: optional headers generated	0
HTTP_Login	The login to use to connect to the WebDAV-HTTPS server 1. This value is mandatory.	
HTTP_Password	The password associated with the login configured in the “HTTP_Login” parameter to connect to the WebDAV-HTTPS server 1.	
HTTP_Port	Network port to use to connect to the WebDAV-HTTPS server 1	443
HTTP_ SynchroniseCertificates	(future development)	0
HTTP_UploadLog	Indicates whether the gateway must also deposit the internal operating logs for the programmed connections in the directory for storing the logs of the WebDAV-HTTPS 1 server. The possible values are: •0: no file upload •1: deposit of files	0
HTTP2_DirAlarm	Directory of the WebDAV-HTTPS 2 server in which the alarms coming from the concentrator will be stored. Note that the directory MUST exist. It will not be created by the hub during file uploads.	/ALARM
HTTP2_DirBin	WebDAV-HTTPS 2 server directory where the hub will look for update files when requested. See the section on configuring updates for how to use updates. (3.1.2.1)	/BIN
HTTP2_DirCert	WebDAV-HTTPS 2 server directory where the hub will look for certificates to use for MQTT connections. See the section on WebDAV server configuration for more details.	/CERT

HTTP2_DirCmd	Directory of the WebDAV-HTTPS 2 server in which the concentrator will search for the command files which will be used later. See the section on using command files for more details	/CMD
HTTP2_DirConfig	Directory of the WebDAV-HTTPS 2 server in which the concentrator places its configuration files. Note that the directory MUST exist. It will not be created by the hub during file uploads. See the section devoted to the configuration of the concentrator by the configuration files for the principle of operation (see chapter 3.1.2: “Operation of the concentrator”)	/CONFIG
HTTP2_DirData	Directory of the WebDAV-HTTPS 2 server in which the concentrator will store the data files collected during operation. Note that the directory MUST exist. It will not be created by the hub during file uploads.	/DATA
HTTP2_DirDef	Directory of the WebDAV-HTTPS 2 server in which the concentrator will deposit the definition files created by itself. Note that the directory MUST exist. It will not be created by the hub during file uploads. See the section dedicated to the configuration of definition files for the operating principle (see chapter 3.1.2.2: “Definition of connected devices”)	/DEF
HTTP2_DirLog	WebDAV-HTTPS 2 server directory in which the concentrator will store the generated log files. Note that the directory MUST exist. It will not be created by the hub during file uploads. See the section dedicated to logs for the different files available (see chapter 4.1.8: “Logs “LOG””)	/LOG
HTTP2_DirScript	Directory of the WebDAV-HTTPS 2 server in which the concentrator will deposit and reread the script files during operation If a file is uploaded to the hub, it will be transferred the next time you connect to the server. If a file is modified on the server, it is uploaded to the hub on the next connection. If a file is added to the server, it is ignored.	/SCRIPT

HTTP2_EuroDateFormat	Specifies the format to use for the timestamp of data sent to the WebDAV-HTTPS 2 server. The possible values are: •0: ISO format is used (YY/MM/DD-HH:MM:SS) •1: the European format is used (DD/MM/YY-HH:MM:SS)	0
HTTP2_HeaderOption	Indicates whether the gateway must add optional headers in the data files uploaded to the WebDAV-HTTPS 2 server. The possible values are: •0: no optional header •1: optional headers generated	0
HTTP2_Login	The login to use to connect to the WebDAV-HTTPS server 2. This value is mandatory.	
HTTP2_Password	The password associated with the login configured in the “HTTP2_Login” parameter to connect to the WebDAV-HTTPS server 2.	
HTTP2_Port	Network port to use to connect to the WebDAV-HTTPS 2 server	443
HTTP2_ SynchroniseCertificates	(future development)	0
HTTP2_UploadLog	Indicates whether the gateway must also deposit the internal operating logs for the connections programmed in the directory for depositing the logs of the WebDAV-HTTPS 2 server. The possible values are: •0: no file upload •1: deposit of files	0
LOG_Level	The concentrator log level in the system files. Used for the box debug mode. The parameter impacts the system log detail level (see section “4. System logs”). The values will be given by Webdyn support if needed.	3
NTP_Server1	1st server to query to set the date/time. (factory value: pool.ntp.org).	

NTP_Server2	2nd server to query to set the date/time. This server is used if the 1st server does not respond.	
NTP_TimeZone	Time zone to apply. This value is used jointly with the time data returned by the configured NTP servers. See appendix “Appendix B: Time zone list” for this list of authorised values.	UTC
SERVER_Address	The remote server address to use to connect to Server 1. This address can be a name if the DNSs are properly configured or an IP address. This value is mandatory.	
SERVER_Interface	Network interface to use to access the remote server for the Server 1 connection. The possible values are: • ethernet: uses the ethernet connection • modem: uses the 2G/3G/4G interface with the embedded SIM card • sdcard : utilise la carte microSD insérée dans le concentrateur This parameter cannot be empty	ethernet
SERVER_Type	The type of server to which the concentrator is to connect for the Server 1 connection. There are several different server types. It is therefore important to select the right server type. The possible values are: • ftp: FTP server • sftp: SFTP server • webdav : serveur WebDAV-HTTPS This parameter cannot be empty	ftp
SERVER2_Address	The remote server address to use to connect to Server 2. This address can be a name if the DNSs are properly configured or an IP address. This value is mandatory.	

SERVER2_Interface	Network interface to use to access the remote server for server 2 connection. The possible values are: • ethernet: uses the ethernet connection • modem: uses the 2G/3G/4G interface with the embedded SIM card • sdcard : utilise la carte microSD insérée dans le concentrateur This parameter cannot be empty	modem
SERVER2_Type	The type of server to which the concentrator is to connect for the Server 2 connection. There are several different server types. It is therefore important to select the right server type. The possible values are: • ftp: FTP server • sftp: SFTP server • webdav : serveur WebDAV-HTTPS • mqtt: MQTT server • mqtts: secure MQTT server • mqtts_aws: MQTTS server on “AWS IoT” • mqtts_azure : serveur MQTTS sur « Azure IoT » • mqtts_gcloud: MQTTS server on “Google Cloud IoT” This parameter cannot be empty	ftp
WEB_Password	Password to access the configuration using the concentrator web site.	high

Note that unless otherwise indicated, parameters can be omitted from the configuration file. In that case the default value will be used by the concentrator as indicated by the file import.

When the concentrator writes the file and sends it to the server, all the parameters are re-established.

9.2 Appendix B: Time zone list

The list of authorised values for the “NTP_TimeZone” parameter is the following:

(GMT-11:00) Midway Island, Samoa

(GMT-10:00) Honolulu

(GMT-10:00) Tahiti

(GMT-09:30) Marquesas

(GMT-09:00) Anchorage

(GMT-08:00) Pacific Time (US and Canada)

(GMT-08:00) Los angeles

(GMT-07:00) Denver

(GMT-07:00) Chihuahua, La Paz, Mazatlan

(GMT-06:00) Guadalajara, Mexico City, Monterrey

(GMT-06:00) Chicago, Central America

(GMT-05:00) Bogota, Lima, Quito

(GMT-05:00) New York

(GMT-04:00) Atlantic Time (Canada)

(GMT-04:00) Caracas

(GMT-04:00) Martinique

(GMT-04:00) Guadeloupe

(GMT-03:30) Newfoundland, St Johns

(GMT-03:00) Antarctica

(GMT-03:00) Sao Paulo

(GMT-02:00) Brazil

(GMT-01:00) Azores

UTC

(GMT+01:00) Europe: Brussels, Copenhagen, Madrid, Paris

(GMT+01:00) Algiers

(GMT+02:00) Athens, Bucharest, Istanbul

(GMT+02:00) Cairo

(GMT+03:00) Moscow, St. Petersburg, Volgograd

(GMT+03:00) Kuwait, Riyadh

(GMT+04:00) Abu Dhabi, Dubai, Muscat

(GMT+04:00) Baku, Tbilisi, Yerevan

(GMT+04:30) Kabul

(GMT+05:00) Karachi

(GMT+05:00) Tashkent

(GMT+05:30) Kolkata

(GMT+05:45) Katmandu

(GMT+06:00) Astana, Dhaka

(GMT+06:00) Almaty, Novosibirsk

(GMT+06:30) Rangoon, Yangon

(GMT+06:30) Cocos

(GMT+07:00) Bangkok, Hanoi, Jakarta

(GMT+08:00) Beijing, Chongqing, Hong Kong, Urumqi, Shanghai

(GMT+08:00) Taipei

(GMT+09:00) Osaka, Sapporo, Tokyo

(GMT+09:00) Seoul

(GMT+09:30) Darwin

(GMT+10:00) Brisbane, Sydney

(GMT+10:00) Guam, Port Moresby

(GMT+10:30) Adelaide

(GMT+11:00) Noumea

(GMT+11:00) Magadan, Solomon Islands

(GMT+13:00) Auckland, Wellington

Offices & Support Contact

SPAIN

C/ Alejandro Sánchez 109
28019 Madrid

Phone: +34.915602737
Email: contact@webdyn.com

FRANCE

26 Rue des Gaudines
78100 Saint-Germain-en-Laye

Phone: +33.139042940
Email: contact@webdyn.com

INDIA

803-804 8th floor, Vishwadeep Building
District Centre, Janakpurt, 110058 Delhi

Phone: +91.1141519011
Email: contact@webdyn.com

PORTUGAL

Av. Coronel Eduardo Galhardo 7-1ºC
1170-105 Lisbon

Phone: +351.218162625
Email: comercial@lusomatrix.pt

TAIWAN

5F, No. 4, Sec. 3 Yanping N. Rd.
Datong Dist. Taipei City, 103027

Phone: +886.965333367
Email: contact@webdyn.com

SUPPORT

Madrid Offices

Phone: +34.915602737
Email: iotsupport@mtxm2m.com

Saint-Germain-en-Laye Offices

Phone: +33.139042940
Email: support@webdyn.com

Delhi Offices

Phone: +91.1141519011
Email: support-india@webdyn.com

Taipei City Offices

Phone: +886.905655535
Email: iotsupport@mtxm2m.com