webdyn

flexitron group

WebdynSunPM

User manual

www.webdyn.com

Index

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 cell phones 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 and sending it to subscribed customers. 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 https://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 https://www.modbus.org for more details).
MQTT	Message Queuing Telemetry Transport: publish-subscribe messaging protocol based on the TCP/IP protocol.
MQTTS	Secure MQTT messaging protocol.
MSISDN	Mobile Station International Subscriber Directory Number: the telephone number of a cell network user.
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.
SO	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	Télé-Information Client: 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 used by publishers 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...).
UID	Unique Identifier: Unique gateway identifier in "WPMxxxxxx" format with xxxxxx for hexadecimal digits.
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 remote file management. WebdynSunPM uses the WebDAV protocol with HTTPS. This is known as WebDAV-HTTPS.

Notes on this manual

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 has eight separate sections:

- Section 1: General presentation
- Section 2: Installation and maintenance
- Section 3: Configuration
- Section 4: Operation
- Section 5: Commands
- Section 6: Update
- Section 7: Tools & diagnostics
- Section 8: FAQ

Scope

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

Target group

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

We recommend calling on qualified and trained persons to install and commission the WebdynSunPM. Qualified persons must have the following knowledge:

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

Please get in touch with your sales contact (contact@webdyn.com) to obtain the list of partner portals.

Change history

Document version	Content	Applicable firmware version
V2.05	Manual creation	V2.2.0 to V2.2.6
V3.00	Micro SD card management Added WebDAV Added MQTT CRI and proprietary protocols placed in a specific appendix Updated device editing and creation Added the WebdynSunPM 4G version	V3.0.0 to V4.2.17
V4.02	Added variables to the SCL script file Added the WebService Added the option for the number of acquisitions in DATA files Updated the script HMI Added client-encrypted LUA scripts Added Webdyn scripts with licence Added forced write codes to the definition files Added action codes 6, 7 and 9 to the definition files Modified the MQTT data format (metadata) Added the "getData" command	V4.2.17
V4.03	ActivePowerControl script 1.02 connects in stop and flat mode and new script parameters (additional licence required) Added a new script function to send an alarm Added a new script function to check a device/tag in the configuration Correction of setKey by text message Correction to the diesel generator script (waiting time) Diesel generator scripts: remote configuration using FTP	V4.3.2
V4.6	Automatic detection of the following inverters: ABB, Cefem, Goodwe, Growatt, Huawei, Ingeteam, Kaco, Solis, Sungrow, Fronius, Kostal, SMA	V4.6.01 to V4.6.08
V5.00	New web interface Automatic detection of MaxConnect (SOLARMAX) devices and inverters: CyberPower, SAJ, SolarEdge Added Modbus slave Removed MQTTS Google Cloud IoT	V5.0.00 to V5.0.13
V5.05	Modifications on Schedules	V5.0.00 to

		V5.0.14
V5.06	Modification of the I/O management Mdofication Equipments Settings Added library for managing definition files Adding actions for the relay Added import and export of the Modbus Slave definition file	V5.1.0
V5.07	Added SMS encryption Added the 1st page of user and SMS password configuration Added log download from the web interface Added the "updateLibrary" command	V5.1.1
V5.08	Adding RAW values (raw modbus registers) to the definition file Adding a password for SMS encryption Added a web password setup page on first login.	V5.1.5

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.

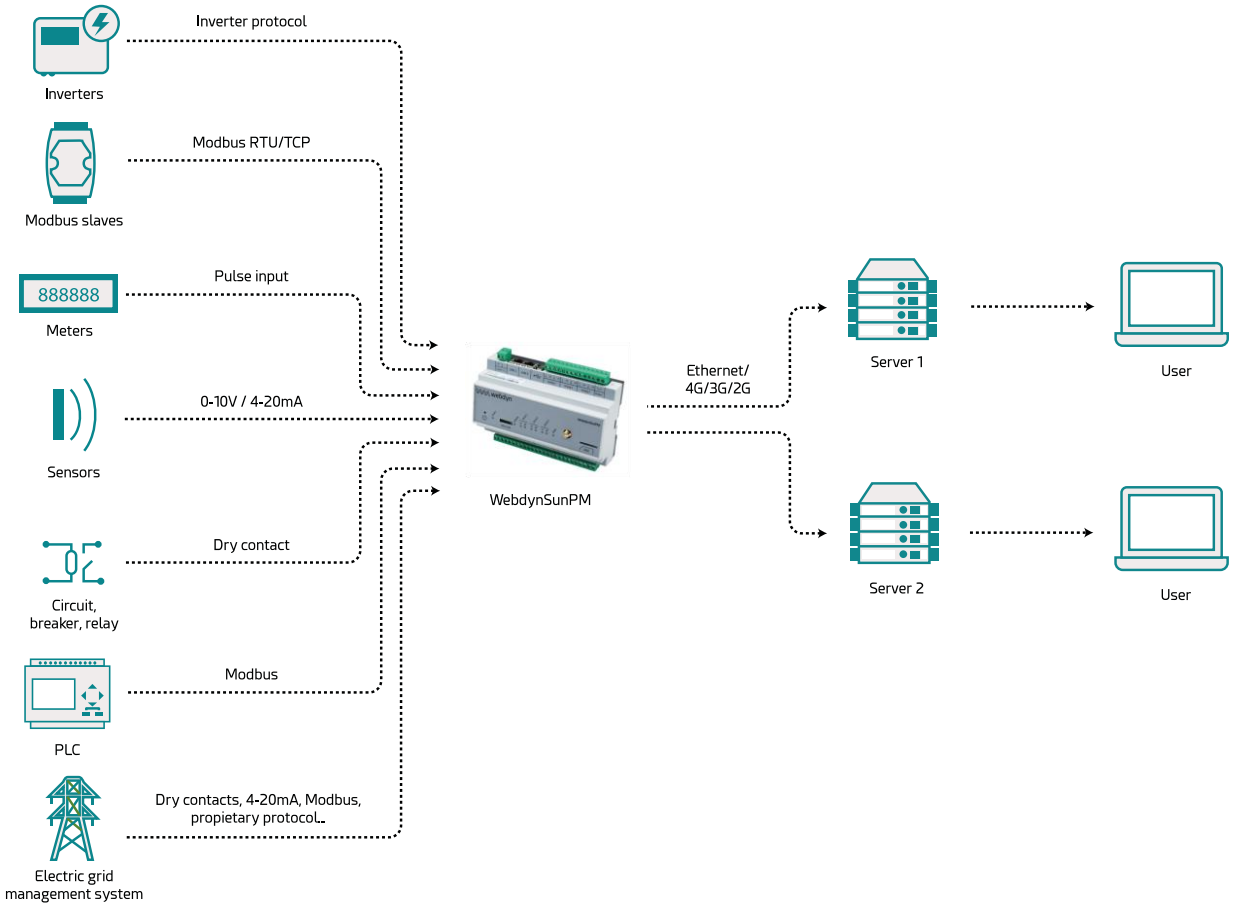
The collected information (data, parameters, etc.) is formatted before being sent to an Information System (IS). The concentrator provides the security and confidentiality of the exchanged information.

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

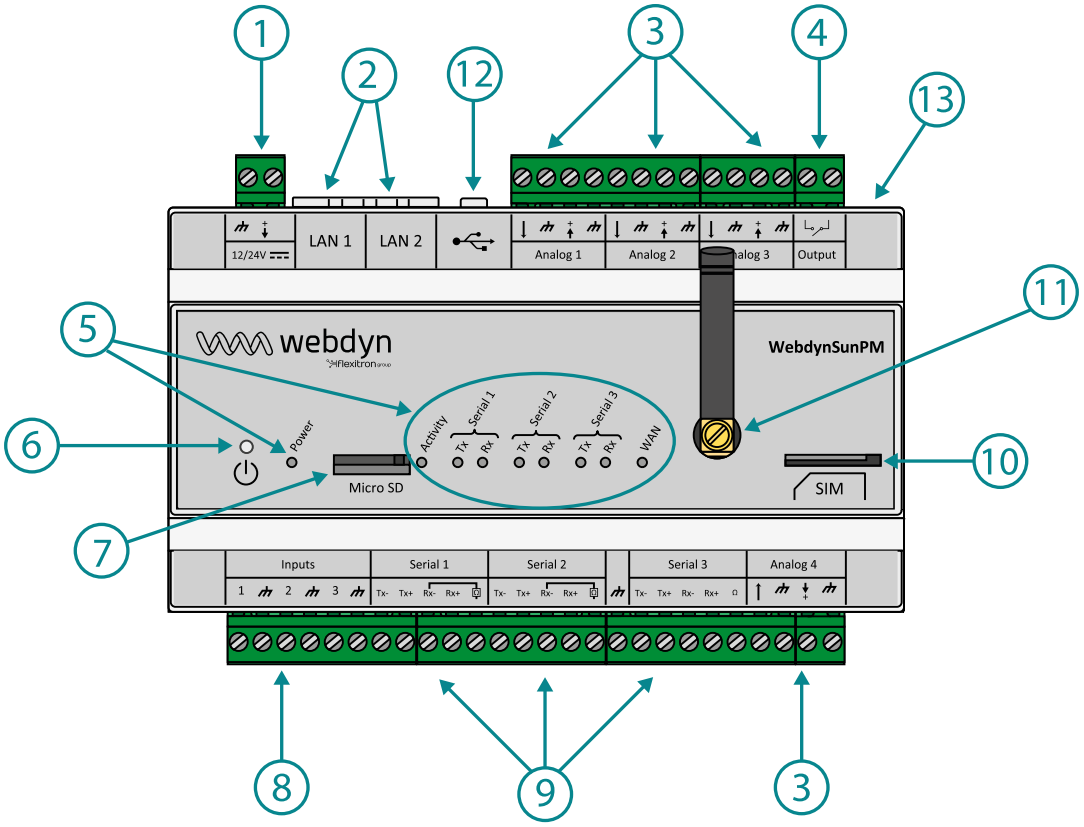
1.2 Operating principle

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 also manages the SFTP, WebDAV protocols using HTTPS and MQTTS to secure the 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



Number	Description
1	12-24 V power supply terminal block
2	2 Ethernet ports (LAN 1 and LAN 2)
3	4 0-10 V or 4-20 mA analog inputs
4	1 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	3 Digital inputs (ON-OFF or pulsed S0)
9	3 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
Solarmax	RS485/RS422 2 wires	100 max
CRI	USB (cable optional)	1 max
Delta-Solivia	RS485/RS422 2 wires	100 max
Kaco	RS485/RS422 2 wired or Ethernet	32 max by RS485 100 max by Ethernet

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

1.5 Product references

References	Descriptions
WG0517-A01	WebdynSunPM (Europe and India version)
WG0517-A02	WebdynSunPM (World Version)
WG0517-A03-DEIE	WebdynSunPM in a 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	+12Vdc 700mA (some functions are not supported on 12V) or 24Vdc 350mA
Battery	650mAh 3.7V, Lithium-polymer
Consumption	P: 5 W Pmax: 10 W
Dimensions	155x 106 x 58mm 9 DIN rail modules
Casing	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	0,330kg
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 "CE" marking created in the framework of European technical harmonisation legislation. It is mandatory for all products covered by one or more European regulations (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 AND 1800MHz • 3G (HSPA): B1 and B8 World version modem: • 2G (EDGE, GSM, GPRS): 850MHz, 900MHz, 1800MHz, 1900MHz • 3G (HSPA): B1, B2, B5, B6, B8 and B19 Europe and India 4G version modem: • 2G (EDGE, GSM, GPRS): 850MHz, 900MHz, 1800MHz, 1900MHz • 4G (LTE): B1, B3, B5, B7, B8, B20 and B28 Antenna: External SMA
SIM format	Standard SIM (mini SIM) 2FF format 1.8V and 3V compatible
Ethernet interface	2 10/100 Mbits/s ports available
USB interface	1 USB2.0 port
Serial interface	3 RS485/RS422 ports
Input/Output interface	4 analog 0/10V or 4/20mA inputs 3 digital ON-OFF or pulsed S0 inputs (class A or B) 1 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.

2G / 3G connectivity data for Europe and India:

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)

World 2G/3G connectivity data:

RF band	Emission frequencies	Max Power
UMTS B1	1922 MHz - 1978MHz	23 dBm(+-2dB)
UMTS B2	1852 MHz – 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	832.4 MHz – 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)

2G / 4G connectivity data for Europe and India:

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 and Azure 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.

Connection to the electricity supply:

- 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.

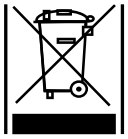
To clean the product, only use a slightly damp cloth to gently clean and wipe the surfaces. Never use aggressive chemical agents or solvents that could alter the plastic material or corrode the metal parts.

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

1.8 Regulation

The product complies with the European directives according to the EU Declaration of Conformity available from Webdyn or on website: 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 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:

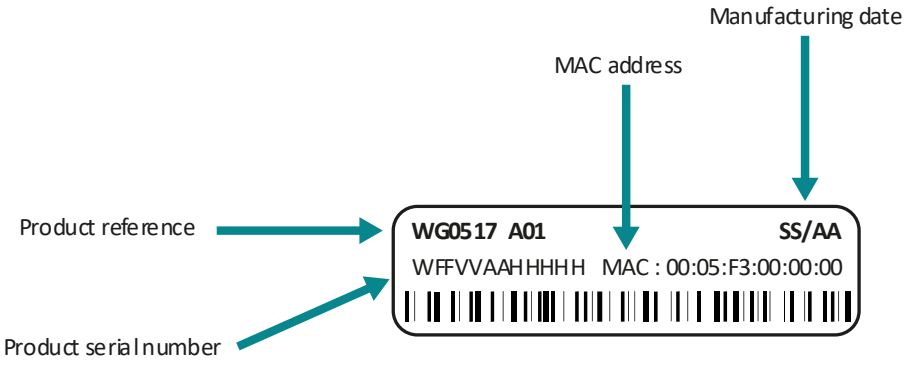
- An angled SMA antenna for the modem (taped to the back of the product).
- A battery (already in place in the product).

2.2.2 Identification

The commercial reference is composed as follows:

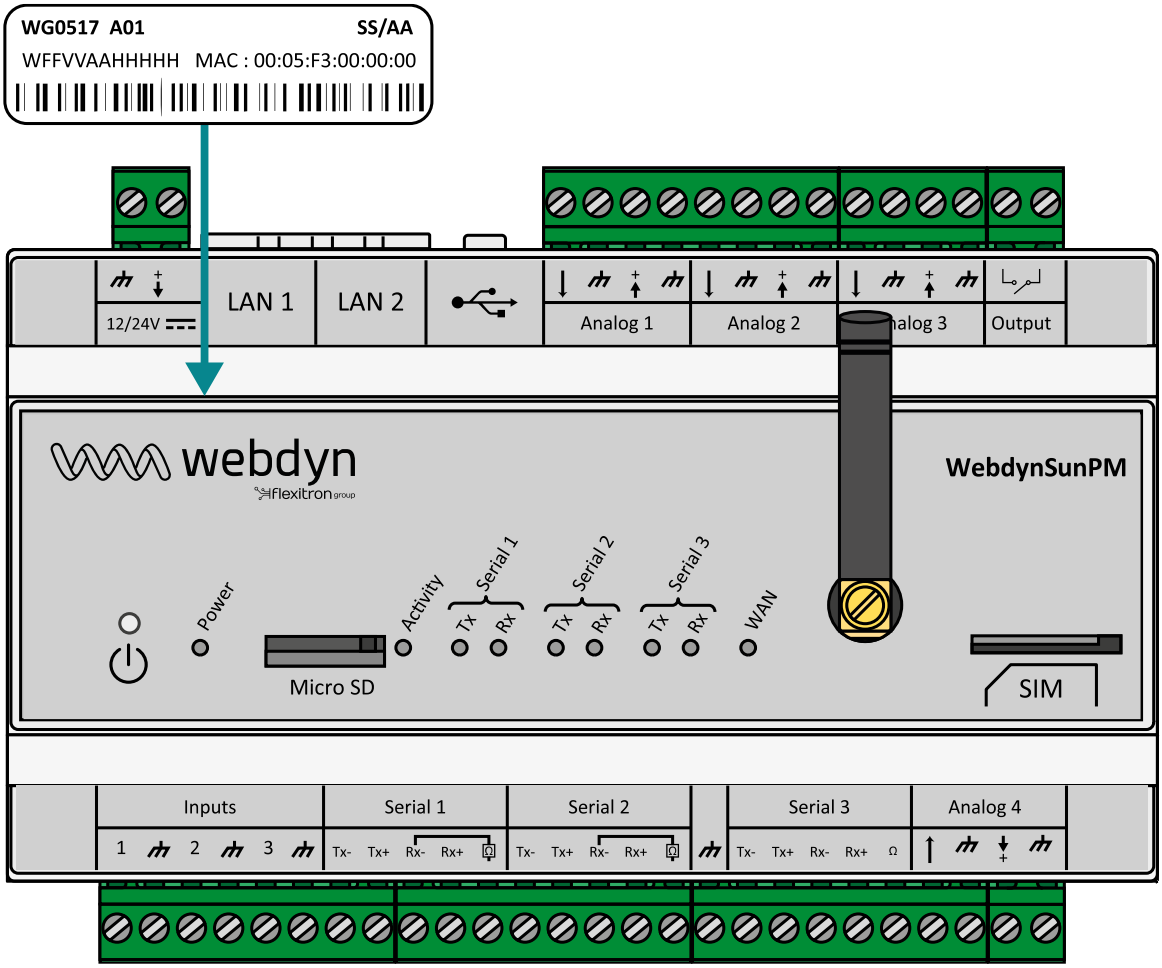
- **WG0517-A01**: WebdynSunPM Europe and India version
- **WG0517-A02**: WebdynSunPM World version
- **WG0517-A03-DEIE**: WebdynSunPM Europe and India in a DEIE box
- **WG0517-A04**: WebdynSunPM 4G Europe and India version

Each product is labelled with the following information:

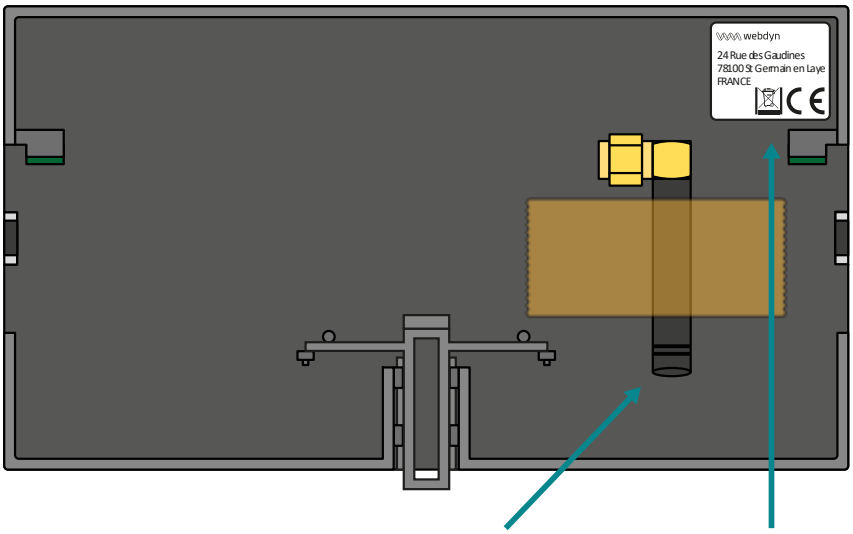


The content of the QR code is:

This label is accessible on the top of the product:
Serialnumber;IMEI;SMS_default_password



The product CE label is 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 "System" menu "About" section (*See section3.2.5.4: "About"*).

2.3 Assembly

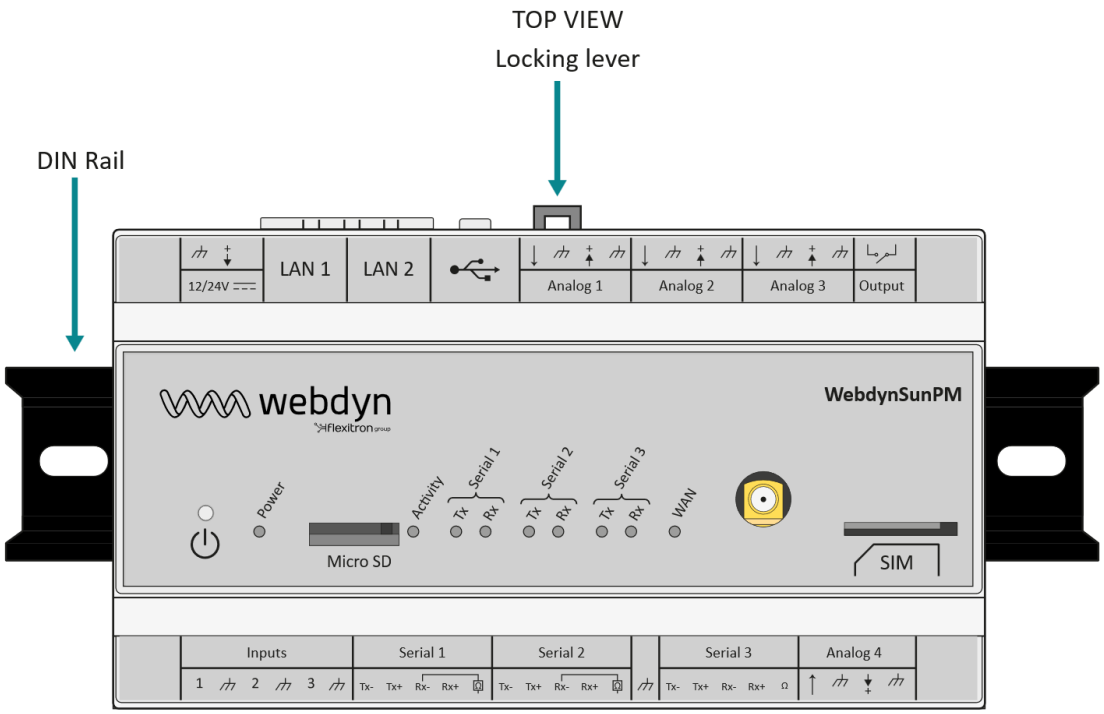
2.3.1 Opening/closing the enclosure

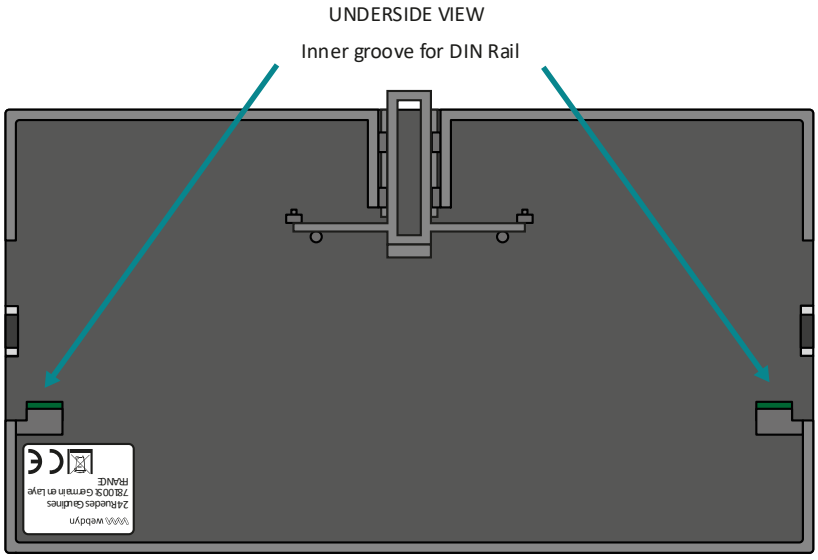
Users must not open the product.

The WebdynSunPM must be returned to after sales for all work on it (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.
- To remove the antenna taped to the back of the box.

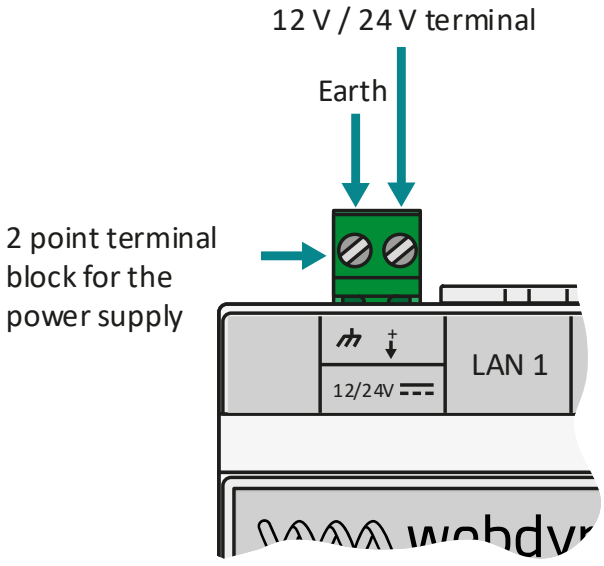
2.4 Interface description

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 uses 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

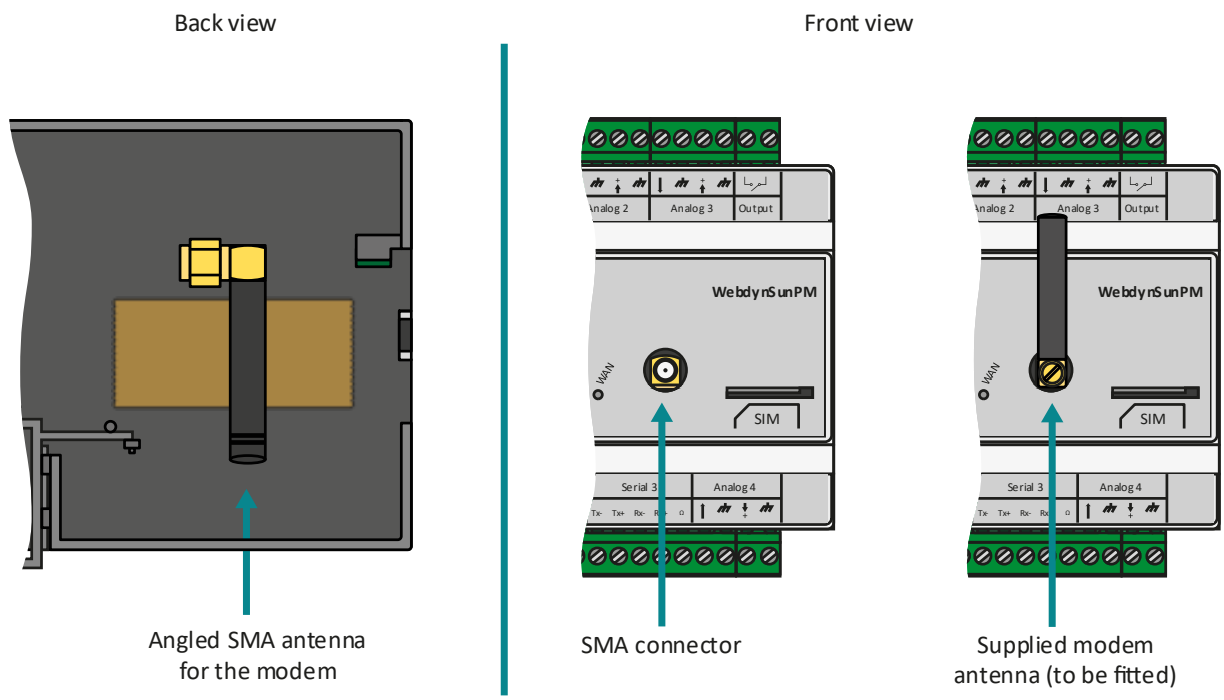
The WebdynSunPM concentrator has a built-in 2G/ 3G or 2G / 4G network compatible modem.

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.

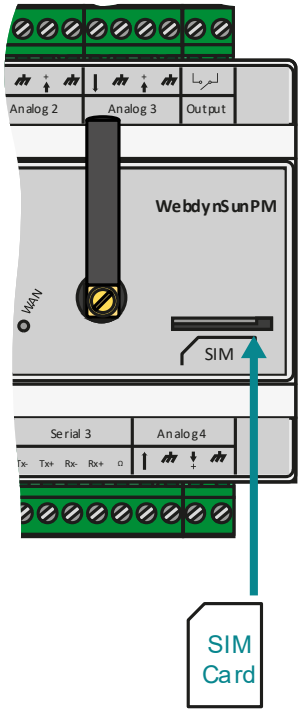
2.4.2.2 SIM card

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.

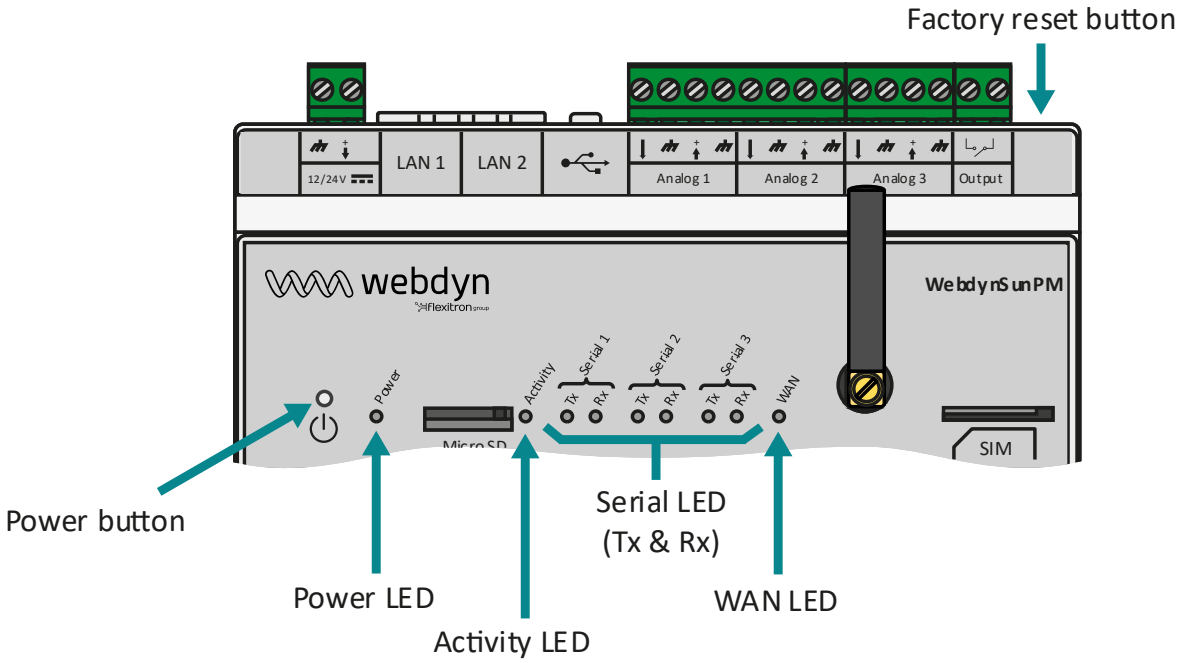
Webdyn does not supply any SIM cards. Contact an M2M operator that supports the 2G / 3G or 2G / 4G network or a partner portal.

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

Power button	Activity LED	Description
Long press	Start flashing every 1s	Trigger the power options reboot or shutdown
Short-press (<2sec)	2 pulses	reboot
Long-press(>5sec)	5 pulses	shutdown
Short-press	N/A	Reboot when shutdown

2.4.3.2 Factory Reset button

Factory Reset button	Activity LED	Description
Long press	Start flashing every 1s	Trigger the factory reset options
press (>4 and <14sec)	4 to 14 pulses	Reinitialise IP settings
Long-press(>15sec)	15 pulses	Reinitialise all parameters and all data

2.4.3.3 Activity LED

Activity LED	Description
Slow flashing	Normal operation
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.

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.

2.4.3.4 Serial LEDs

Serial LED	Description
Tx	No flashing: No transmission
	Flashing: Transmission
Rx	No flashing: No reception
	Flashing: reception

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

2.4.3.5 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). If server 1 is configured and activated, then it has priority for WAN indicator management. The WAN indicator only takes into account server 2 configured for MQTT if server 1 has no 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
	Off	No attempts to use the SD card
Green	Slow flashing	Indicates that the configuration is correct: the SD card has been detected
Red	Steady	SD card error. The SD card has not been detected. Check that it has been properly inserted
Red	Fast flashing	SD card error. A write error has been detected on the SD card. Check that it is properly formatted and that the directories have all been correctly created. If necessary, we recommend replacing the SD card

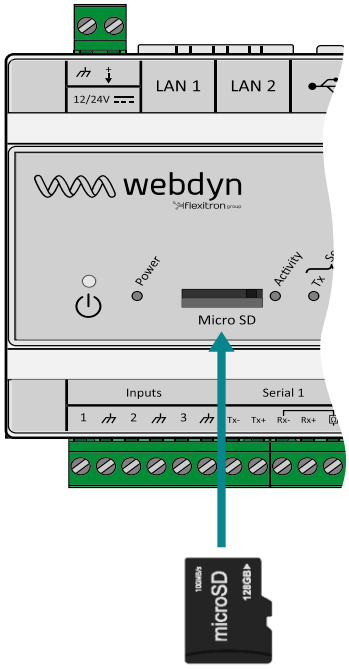
If an error occurs, the WAN LED stays on the last error until the next conclusive attempt or the product reboots.

2.4.4 Extension interface

2.4.4.1 External memory medium (MicroSD)

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

The SD card is used to store the configuration, carry out updates or memorise the data collected from the various devices locally, thus eliminating the need for a remote server.

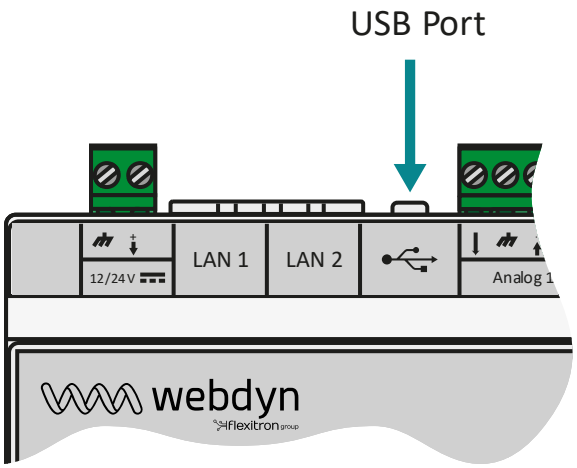


To insert the microSD card into the product, insert it into the slot on the front of the concentrator until it clicks into position.

Webdyn does not supply any SD cards. Contact a computer hardware retailer.

2.4.4.2 USB interface

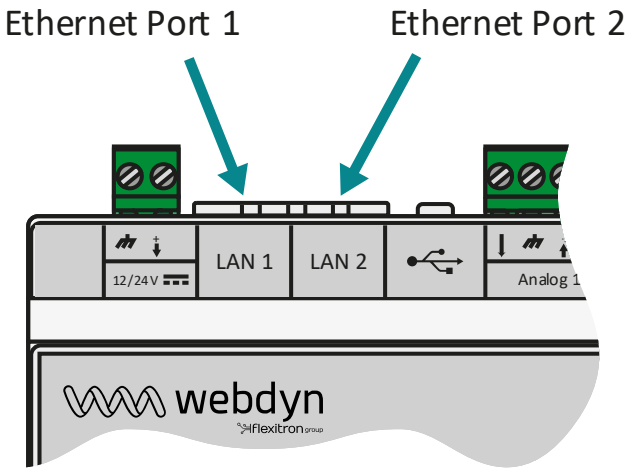
A USB port is available on top of the product next to the LAN connectors and analog inputs. The USB port is used to connect the TIC accessory to remotely read the information from electricity meters.



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:

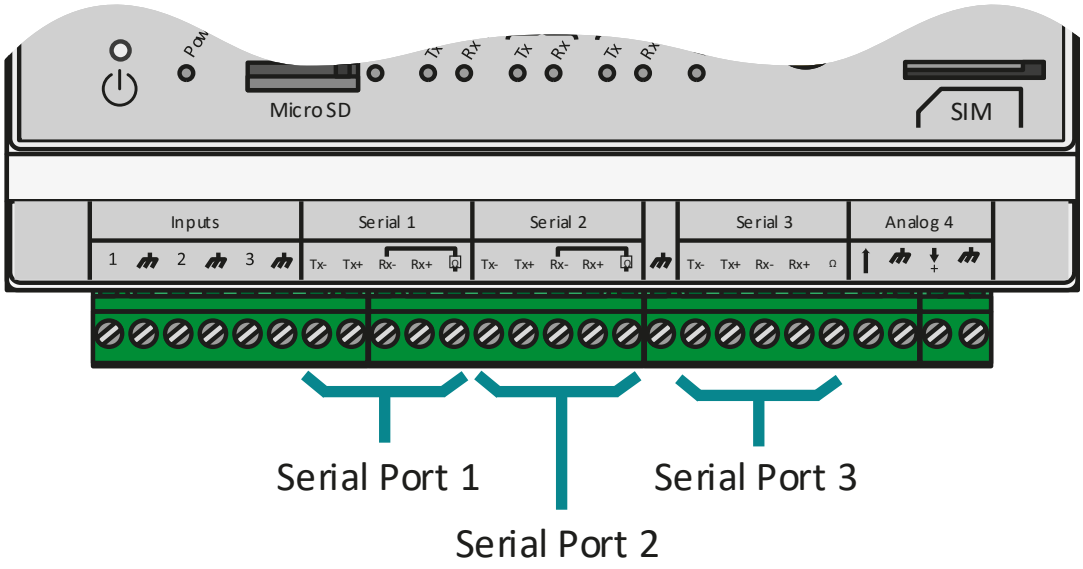
- “Link” (green): Used to check that the physical link with another network device is available.
- “Status” (orange): Used to view network traffic. It flashes depending on the traffic.

- 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 RS485/422 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, which are:

- 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 exceeds 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:

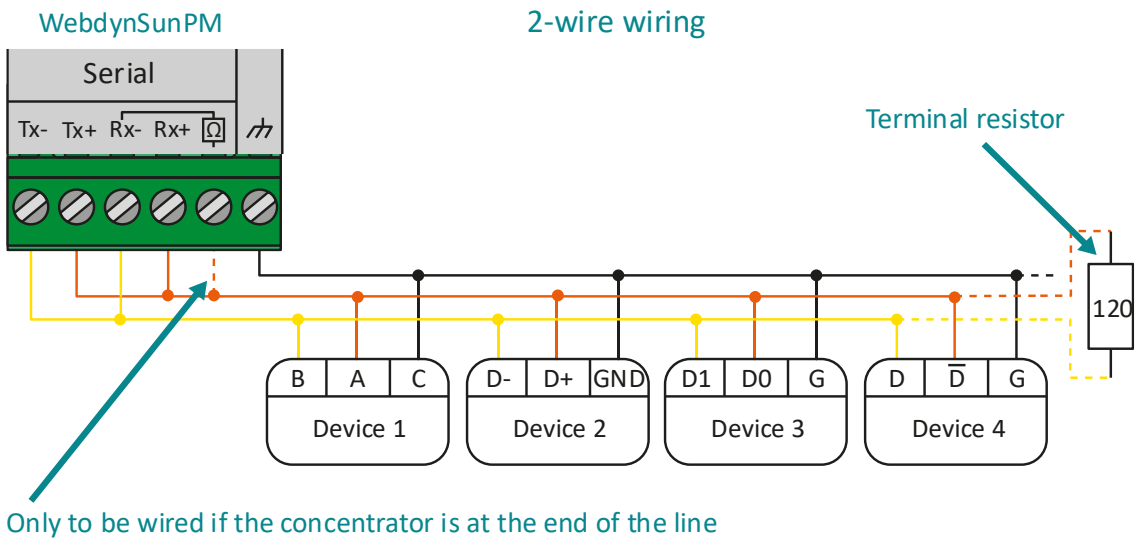
- The modules must be connected one after the other.
- Star connections are prohibited.
- The cables must either be screened or shielded, twisted pair per pair (see above: “cable type for RS485 bus connection”).
- The cable screen or shielding must be connected to the concentrator box earth and not to 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.

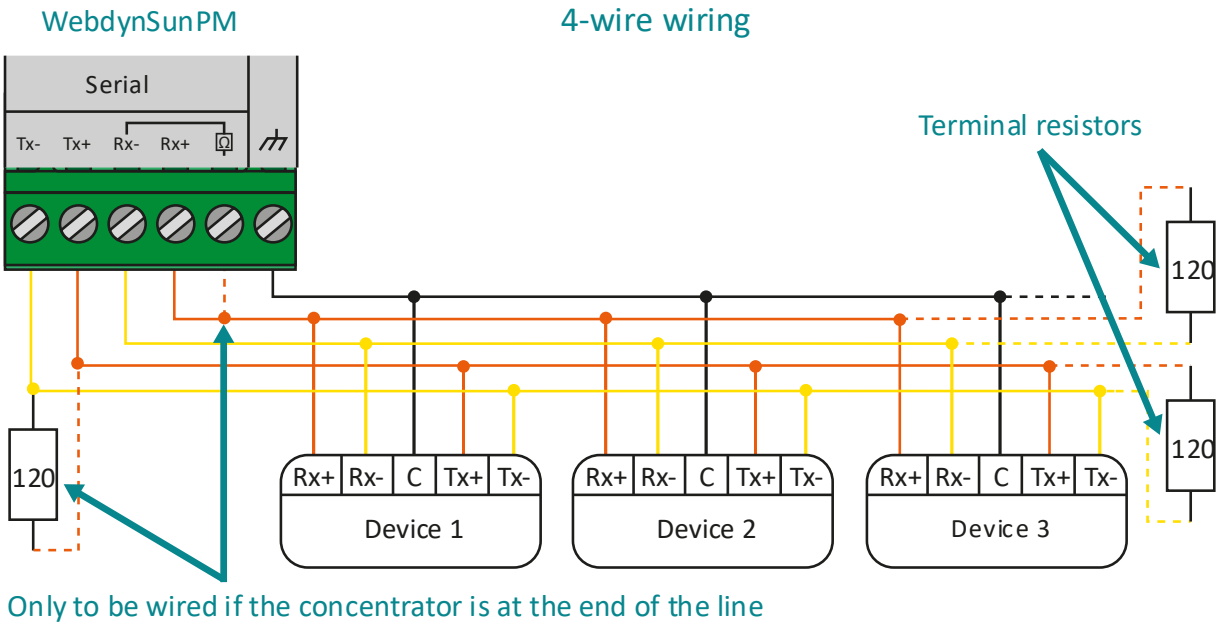
2-wire RS485 wiring (Half-Duplex):

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/ RS422 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.

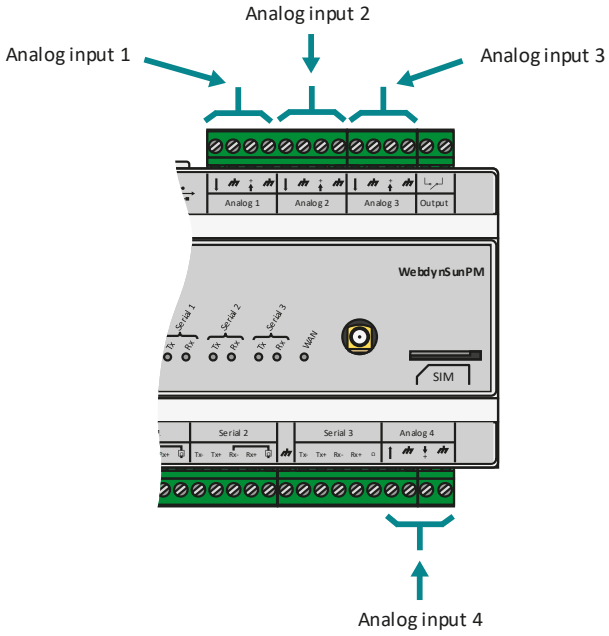
2.4.7 Input/Output interface

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 “Analog” inputs used to measure current of between 4 and 20 mA or a voltage of between 0 and 10 V. Each analogue 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.

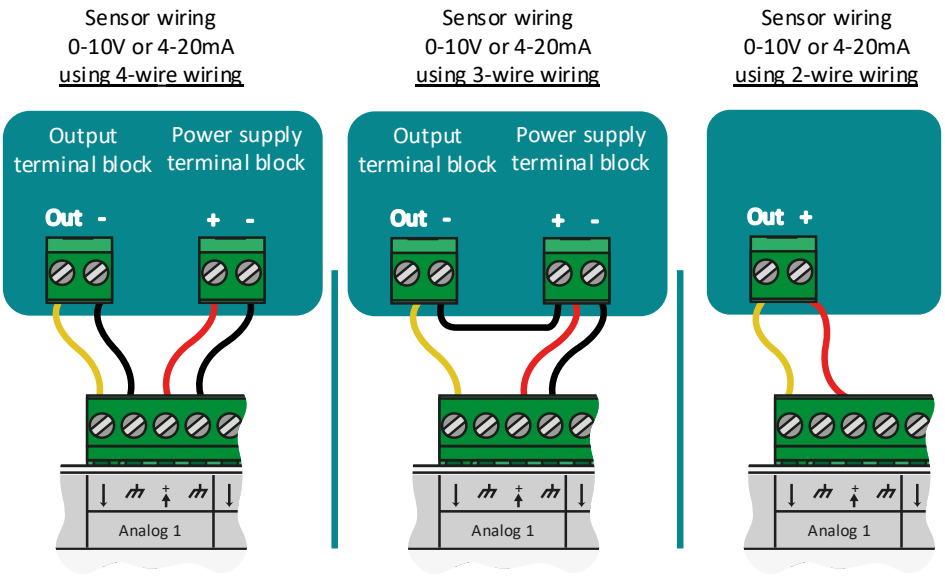


Each analog input can be software configured to 0-10V or 4-20mA.

The concentrator’s analogue/digital converters (CAN) have 12 bit resolution making it possible in:

- **0-10 V mode**: to have a 2.578 mV resolution
- **4-20 mA mode**: to have 5.578 µA 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.



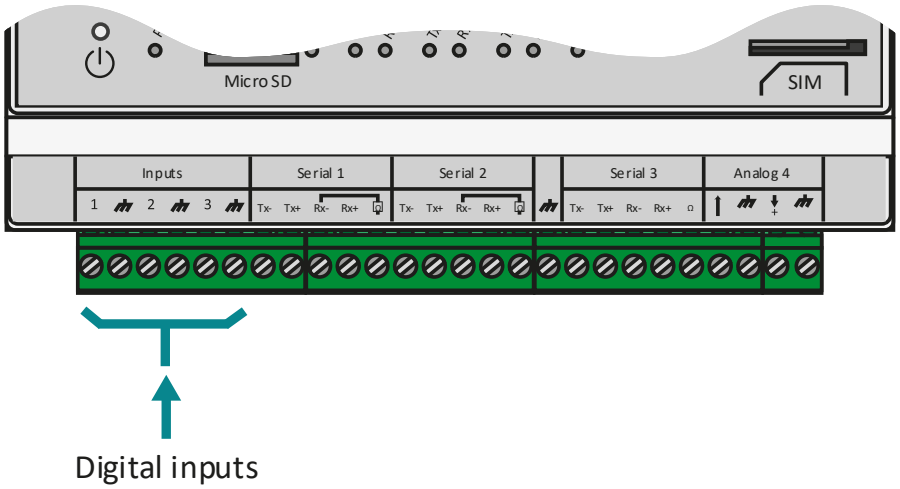
Do not apply a voltage higher than 12V or a current higher than 24mA to the analog inputs.

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.

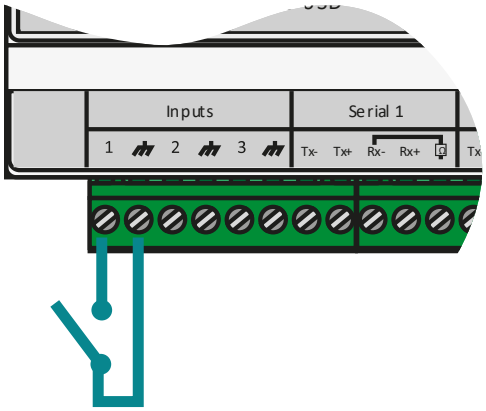
To prevent any damage to the concentrator, do not inject current or voltage onto the digital inputs.

In discrete mode:

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

Discrete 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	> 20ms

ON-OFF Input wiring



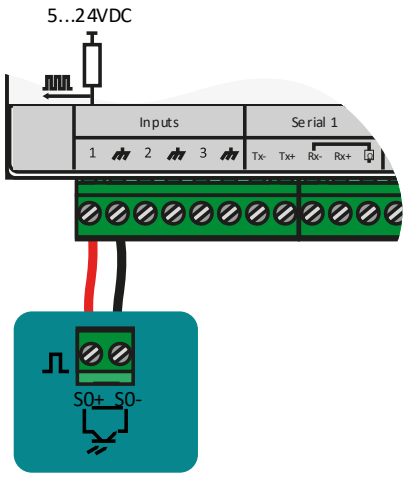
In S0 (pulse) mode:

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 the 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 (24V) Current pulses	Class B (5V) 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

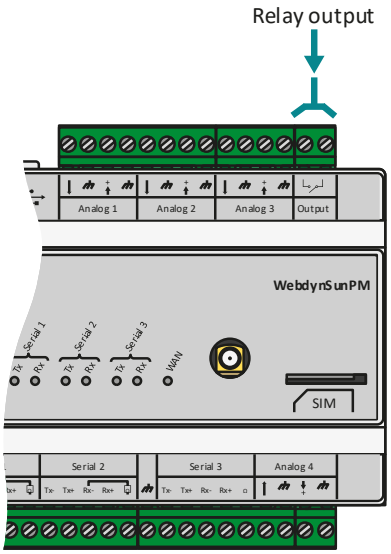


Meters with class A pulse outputs should be used for long distance transmissions. Meters with class B pulse outputs can only be used for short distances and make it possible to have reduced power consumption.

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 cannot be used to directly control high power. In that case, an external intermediate relay must be used.

The relay output can be controlled by:

- Command file (FTP, SFTP or WebDAV), MQTT/MQTTS message and text message. (*See section5.3.12: "*"setRelay": Relay status update*"*).
- LUA Scripts. (*See section3.2.4: "Control"*).

3 Configuration

The concentrator can be configured in several ways:

- Using configuration files uploaded to an FTP (unsecure), SFTP or WebDAV-HTTPS server
- Using RPC commands on an MQTT (unsecure) or MQTTS server.
- Using configuration files uploaded to a micro SD card.
- Using the concentrator's web interface
- Using encrypted text message commands

The use of an FTP/SFTP/WebDAV-HTTPS server is the preferred solution. 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.

Please note that on the first start-up, at least the following elements need to be configured using text messages (*see section 5.2.3: "Text* message *)*or the local web interface (see section 3.2.3.3: " Server "):

- SFTP server name.
- SFTP 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 SFTP server. They are never a substitute for full SFTP 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).

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 management 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 by using its last 6 characters prefixed by "WPM". Thus, for a product with a MAC address 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: these are the files that contain the variables accessible on a device. By default they are stored in the "/DEF" directory on the server.

Please note that the concentrator does not create the server tree structure. Therefore, all the directories must be created manually. Thus, by default, the following directories at least must be present on the server:

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

These directories must have read, write, delete and creation 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 on the server are modified, 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 to it.

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 interface configuration and the monitored device list:
  - 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 device:
    - Index
    - Name
    - Interface
    - Address
    - Definition file
- **<UID>_licence.ini**: this file contains the licences for the ".luaw" Lua Webdyn scripts

Where **< UID>** is the concentrator identifier.

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.

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 following configuration parameters must be added to the file:

For an FTP (unsecure) or SFTP server:

- SERVER_Address=< FTP/SFTP server address or name 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 (default)
- FTP_Login=<Login name 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 by Ethernet:
  - SERVER_Interface=ethernet
- If the connection is by modem:
  - SERVER_Interface=modem (default)
- HTTP_Login=<Login name to use for the server>
- HTTP_Password=<Password for the login>

The configuration parameter details for this file are in section 10.1: "Appendix A: "_config.ini" configuration file".

For an MQTT server:

WebdynSunPM can connect to an MQTT server to store its data and alarms. Commands can also be sent to the concentrator through the MQTT server. To do that, the topics subscribed to the concentrator in its settings must be indicated.

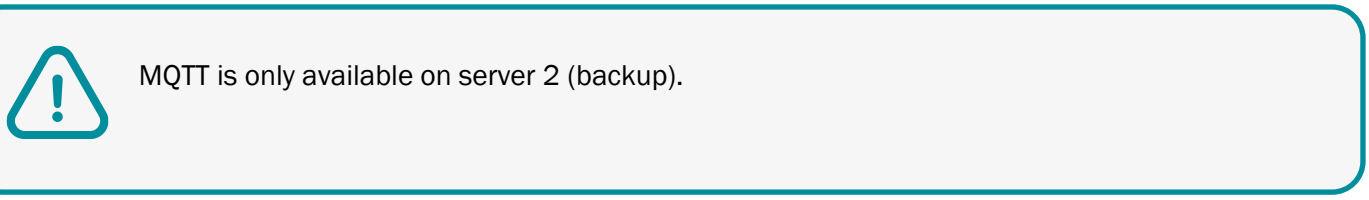
The concentrator cannot be configured using the MQTT server, either an FTP server or the embedded web interface must be used.

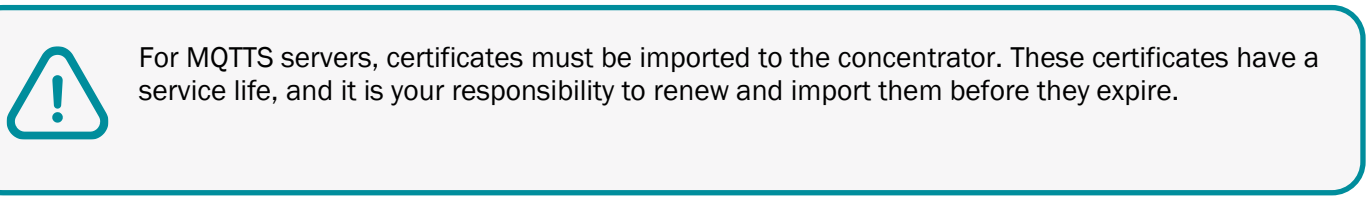
The concentrator supports 4 different MQTT server types, namely:

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

Contact the MQTT server manager to get the configuration to be applied and the certificates and secure keys to import to the concentrator.

The configuration parameter details for this file are in section 10.1: “Appendix A: “_config.ini” configuration file”.





3.1.2.1.2 “< UID>_var.ini” file

The “_var.ini” file contains the connection schedule list for connections to the servers 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 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[m]	Schedule parameters for the Server 1 connection	
SCHEDULE2_Params[n]	Schedule parameters for the Server 2 connection	

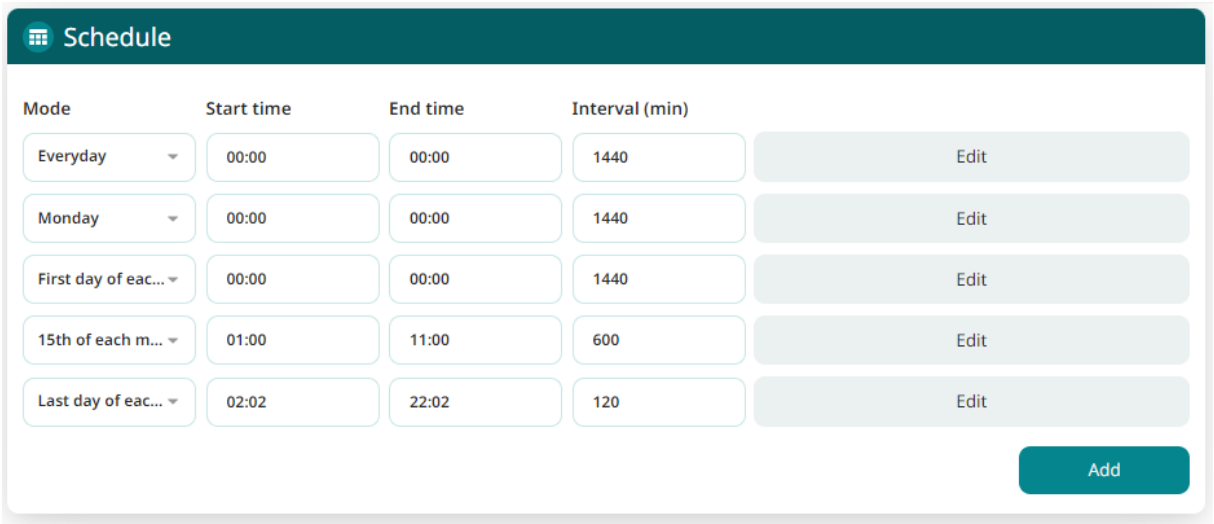
“” Where “m” or “n” are replaced by the schedule index number. These numbers start at 0.

The parameters for each row are in the following format:

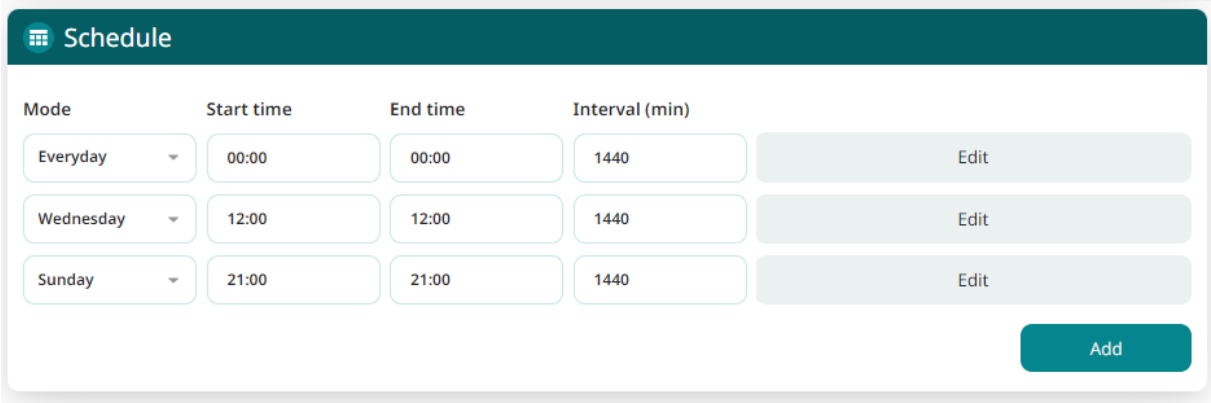
Each configured row therefore contains the following configuration information:

Parameter	Description	Default value
Id	Configuration line identifier. This 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 for the Schedule 2 connection:



The following configuration file is obtained:

```
S
S
S
S
S
S
SCHEDULE2_Params[1]=2|wednesday|12:00:00|1440|1
SCHEDULE2_Params[2]=3|sunday|21:00:00|1440|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 settings for all the interfaces and the list of devices 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.

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
- Modbus slave configuration
- Connected device configuration

3.1.2.1.3.1 Modem configuration

The modem configuration is based on the following parameters:

Parameter	Description	Default value
type	Device type. The only possible value here is “MODEM”	MODEM
pin	Is 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. When the field is filled in correctly, the IP connection to the mobile network is permanent.	
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.	
authentication	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. The above "login" and "password" must be entered. • **CHAP**: CHAP type authentication requested by the remote server. The above "login" and "password" must be entered. • **Both**: CHAP or PAP type authentication. The above "login" and "password" must be entered.	none
server	Text messaging centre. The text message centre is used to manage text messages from the concentrator. Enter the phone number for the text messaging centre you want to use instead of your mobile service provider. For example: “+33989004000”. This field can be empty.	
dns	DNS server. DNS (Domain Name System) servers translate explicit internet addresses (for example, www.webdyn.com) into their corresponding IP addresses. Enter the address of the DNS server you want to use instead of your mobile service provider. For example, you can use the google DNS: “8.8.8.8”. This field can be empty.	

type;pin;apn;login;password;authentication;server;dns
MODEM;;m2minternet;;;none;;

Below is a modem configuration example:

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 CSV format:

type	pin	apn	login	password	authentification	server	dns
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.	
DNS1	IP address for a DNS server the concentrator is to use to resolve names.	
DNS2	IP address for a DNS server the concentrator is to use to resolve names if the “DNS1” server fails to respond..	

```
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;;;
```

Below is an Ethernet interface configuration example:

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:

If these network interfaces are used to collect data from a device to be monitored, the requests may need to be spaced out to avoid saturating the device being queried. To do that, configure the Ethernet interframe parameter as follows:

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.

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	
LAN2	192.168.2.12	255.255.255.0			
tcpInterFrameMs	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

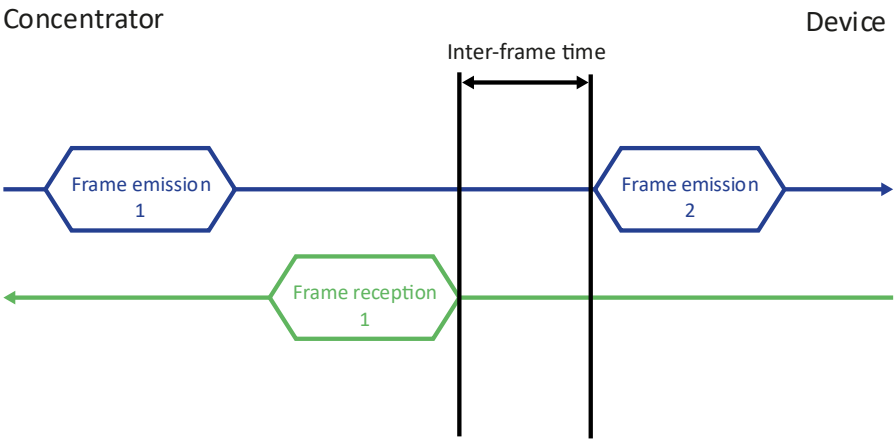
Serial port configuration is based on the following parameters:

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 first line SERIAL2 for the second line SERIAL3 for the third 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
data_bits	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
stop_bits	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 management • **proprietary protocol** (see specific appendix on proprietary protocols)	modbusRTU
interFrame(ms)	Waiting time between 2 frames exchanged on the serial port. This time is expressed in ms. See below for a detailed explanation of how this parameter operates.	0
forwarded_port	Forwarded s port.	

	If there is a value in this field, the concentrator opens the entered port. When ModbusTCP devices connect to this entered port, all sent requests are directly forwarded to the ModbusRTU bus then the ModbusRTU device response is returned to the connected ModbusTCP device using this same 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. Important: the port is only accessible on the LAN. No modem access is authorised even if your SIM card provides a public address.	

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.

type;baudrate;data_bits;parity;stop_bits;wire;protocol;interframe(ms);forwarded_port
SERIAL1;9600;8;N;1;2;Modbus;0;
SERIAL2;19200;8;N;1;2;Modbus;0;
SERIAL3;19200;8;N;1;2;PW1;0;

Below is a serial interface configuration example:

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 CSV format:

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

3.1.2.1.3.4 Modbus slave configuration

Modbus TCP slave configuration is based on the following parameters:

Parameter	Description	Default value
enabled	Modbus TCP slave status: • 0: disabled, • 1: Enabled	0
port	TCP port used by the Modbus server	502
mapping file	Name of the definition file describing the associations between registers and variables. The file must be present in the "/DEF" definition directory. If the "mapping file" field is empty, then only WebdynSunPM registers are available. (See section4.1.2: ""DEF", the definitions")	

3.1.2.1.3.5 Declaration of devices to be monitored

The configuration of the devices to be monitored 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.	
interface	The possible interface are: • 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 corresponds to the device address on the serial link. For Modbus, its value is therefore between 1 and 254 (UnitID).	
acqPeriod(s)	This field is used to indicate the data recording period in the file (device collection is continuous). If the field is set to "0", then no data is stored by the concentrator. It is expressed in seconds.	600
timeout(ms)	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.	1000
serialNumber	The device serial number. Fill in automatically if available during a detection.	
parameters	This field is only used in the following cases: • ModbusTCP device: If the field is set to "1", the connection is not closed after the device is queried thereby saving time for the next query. If the field is set to "0", the connection is systematically closed after the device is queried. The default value is 1. • Proprietary device: (see the specific appendix on proprietary protocols).	
tag / category	From V5.1.0, this field is used to assign a tag to the equipment Before V5.1.0, this field is used to assign a category to the equipment This field is automatically filled	
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 (See section 3.1.2.2: “Connected device definition”).	

```
index;interface;name;address;acqPeriod(s);timeout(ms);serialNumber;parameters;category;model;defFile

IO;;Io;;36000;;;;WebdynSunPM;ioSunPM;WPM00C715_IO.csv
```

Below is a connected device interface configuration example:

```
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_Sol
ar_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 the 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 the specific appendix on proprietary protocols).
- 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 Modbus address 126
  - 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 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	SERIAL 1	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>_scl.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.

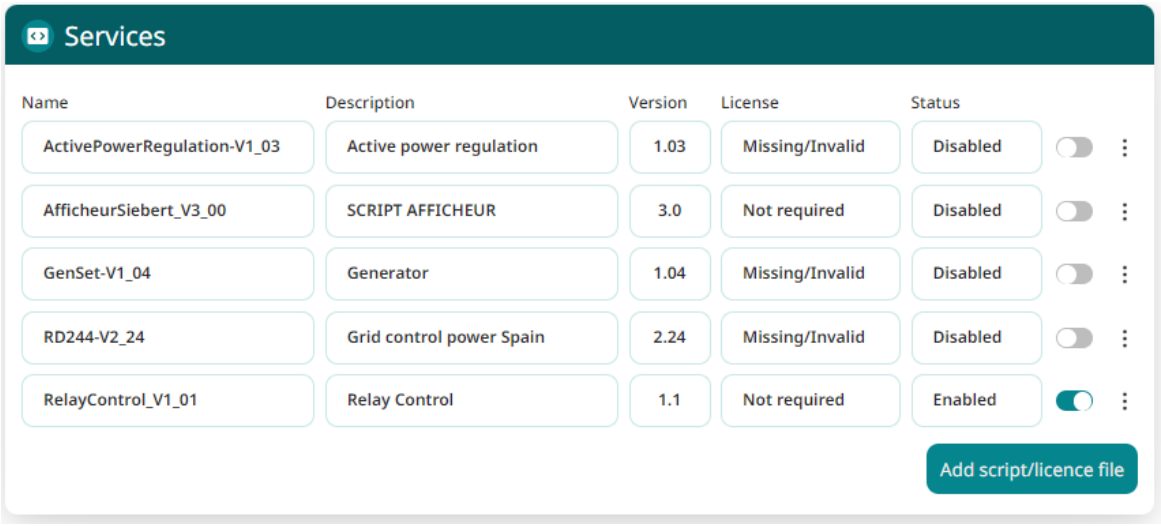
The file has three lines per configured script. Each line contains the script number.

The format is the following:

Variable	Definition	Default value
SCRIPT_File[n]	Scenario name	Script file name with the extension
SCRIPT_Enable[n]	Script activation	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:

```
S
o
S
S
S
S
S
S
S
S
S
0  'gridTargetKW":"60" "gridSafetyKW":"20" "errorAction":"stop" "dg":[{"dgEnabled":true "dgRatedPowerkW":"30" "dgTargetPercent":"1
0  },{"dgRatedPowerkW":"30","dgTargetPercent":"100"},{"dgRatedPowerkW":"30","dgTargetPercent":"100"}]}
S  :IPT_Enable[3]=0
S  IPT_File[3]=GenSet-V1_04.luaw
SCRIPT_Args[4]=
SCRIPT_Enable[4]=0
SCRIPT_File[4]=RD244-V2_24.luaw
SCRIPT_Args[5]=""
SCRIPT_Enable[5]=1
SCRIPT_File[5]=RelayControl_V1_01.lua
```

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 licences for the ".luaw" Webdyn Lua scripts. The licence file must be placed in the remote server CONFIG directory.

The licence file can be used to obtain specific scripts designed by Webdyn. In that case, contact the Webdyn sales department, which will be able to advise you and redirect you to the relevant contacts: contact@webdyn.com

If the file is detected when connecting to the remote server, it is uploaded and the configuration is applied immediately. WebdynSunPM will not download the licence file if it is deleted.

The licence file is specific to a WebdynSunPM. The same file cannot be used on several concentrators. The licence file contents must not be modified, otherwise the concentrator's licence management will be blocked.

3.1.2.2 Connected device definition

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

A definition file is needed for 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 (*See section 3.1.2.1.3.5: "Declaration of devices to be* monitored*")* 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 rest of the 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>_ <Manufacturer>_<Model>.csv

Where:

- 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.

3.1.2.2.1.3 Proprietary protocol

The proprietary protocol file name detailed in the proprietary protocol 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:

P

The fields are configured as follows:

Field	Description
Protocol	Protocol name used for this device. The possible values are: • Modbus: Communication via Modbus RTU or Modbus TCP ▪ modbusRTU: the device is accessed using the modbus RTU (backward compatibility from v5.1.0) ▪ modbusTCP: the device is accessed using TCP connection (backward compatibility from v5.1.0) • 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 write code	Indicates whether the Modbus function code used for writing should be forced to 0x10. The possible values are: • 0 (default value): a single register is written using function code 0x06, while several registers are written using function code 0x10. This is the classic behaviour for Modbus devices. • 1: Only function code 0x10 is used for writing, even in the case of a single register. This behaviour is required for certain devices. 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:

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 authorised formats 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_1 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) • RAW: The variable is a sequence of modbus registers. In this case, the notation "Adress_Size" must be used for the "Info2" field 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. (See table below) 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 will be exchanged. This is 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, the decimal separator is the ". ". 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 to be carried out by the operators who, if they want to convert the sent raw data, will have to apply the Ax+B operation to data x, in ” particular to obtain a value in the unit given 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. 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 as far as acquisition is concerned. But the variable will also be targeted by an instant collection using the "getData" command. • 7: the variable is equivalent to a variable defined with code 2 as far as acquisition is concerned. But the variable will also be targeted by an instant collection using 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 alarms file using a single field.. A connection is immediately triggered to send the file. • 9: the variable is equivalent to a variable defined with code 8, so it is an alarm. However, the alarm file containing the new value will not be uploaded until the next connection to the server (scheduled or manual). There is no automatic connection for this action code.

“Swap" modifiers can be used to adapt to any type of device that may manage data organisation in memory differently to make it compatible with the file format generated for the server.

List of authorised modifiers according to the value type:

Value type	"swap" modifier type	Conversion
• U8 • I8 • String • Bits • IP • IPV6 • MAC • RAW	No “swap” modifier	
• U16 • I16	_B	0xAABB→0xBBAA
• U32 • I32 • F32	_B	0xAABBCCDD→0xBBAADDCC
• U32 • I32 • F32	_W	0xAABBCCDD→0xCCDDAABB
• U32 • I32 • F32	_WB	0xAABBCCDD→0xDDCCBBAA
• U64 • I64 • F64	_B	0xAABBCCDDEEFF1122→0xBBAADDCCFFEE2211
• U64 • I64 • F64	_W	0xAABBCCDDEEFF1122→0x1122EEFFCCDDAABB
• U64 • I64 • F64	_WB	0xAABBCCDDEEFF1122→0x2211FFEEDDCCBBAA

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: analogue input (0-10V or 4-20mA) • 2: digital input • 3: switching relay output
Info2	Input/output number. The authorised values are: • 1 to 4: if “Info1” equal to 1 (analogue input) • 1 to 3: 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 (analogue input), the authorised values are: ▪ 1: analogue input of the 4- 20ma type ▪ 2: analogue input of the 0-10V type • If “Info1” equals 2 (digital input), the authorised values are: ▪ 1: Discrete 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 **0x02** This type is used to read inputs. 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 as follows and depend on the "Info3" variable type: • 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 the 2nd byte is to be read from modbus register 40000. **Register address, start bit and bit length.** This format is used to retrieves data of type bitfield ("Bits" of Info3 field) of a specific length. The format is the following: **Register address_1st bit_bit length**. Thus, for example, the value

	"40005_4_8" configures a bitfield variable stored on register 40005 at the 4th bit and is 8 bit long. In case the data is only 1 bit, there is no need to indicate a bit length. Example: value "40008_5" configures a 1-bit variable on the 5th bit of register 40008.
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_1 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) • RAW: The variable is a sequence of modbus registers. In this case, the notation "Adress_Size" must be used for the "Info2" field • 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 bytes, one by one • _WB: the words AND the bytes are exchanged. The 2 previous modifiers are applied. (See the table in the above explanation of the definition file content above) 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 will be exchanged. This is a "Little endian/Big endian" conversion.
Info4	Scale Factor When the variable was generated automatically by SunSpec detection, 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 * 10^{sf}" 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.

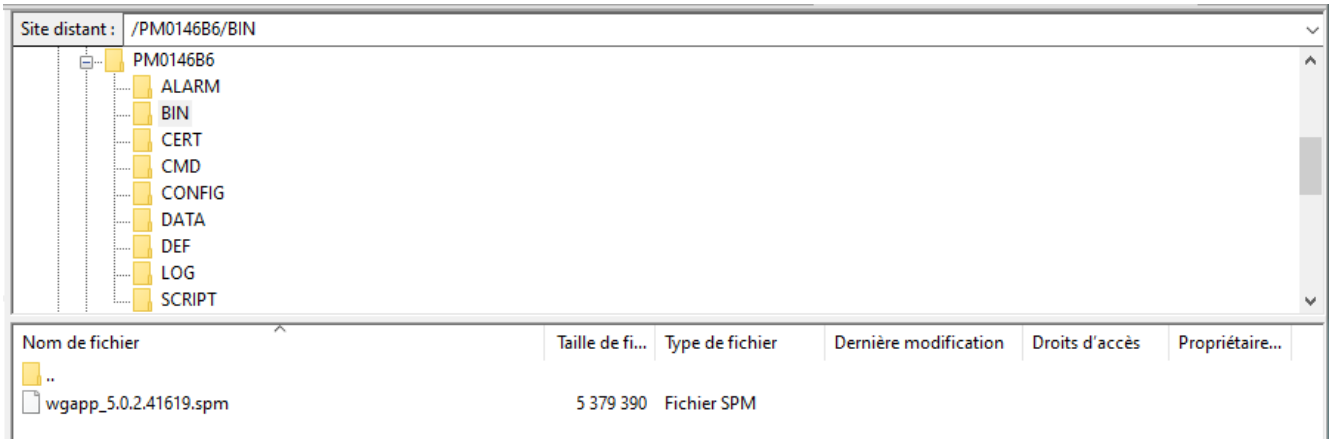
3.1.2.2.2.3 Proprietary protocol

Field	Description
Info1	(see specific proprietary protocol appendix)
Info2	(see specific proprietary protocol appendix)
Info3	(see specific proprietary protocol appendix)
Info4	(see specific proprietary protocol appendix)

3.1.3 Updates

To update using the server:

- Place the new file on the server in the "BIN" firmware directory, as follows:



- Modify the "_config.ini" file in the "CONFIG" configuration directory:
  - Put the binary name in the “BIN_FileName” field.
  - Put the binary validation checksum that was provided with the binary in the “BIN_Checksum “field.

If the example below, the modified lines will therefore be:

```
BIN_Checksum=ec97aab9a0f112918ce24a494d2fe141
BIN_FileName=wgapp_5.0.2.41619.spm
```

3.2 Embedded web interface

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

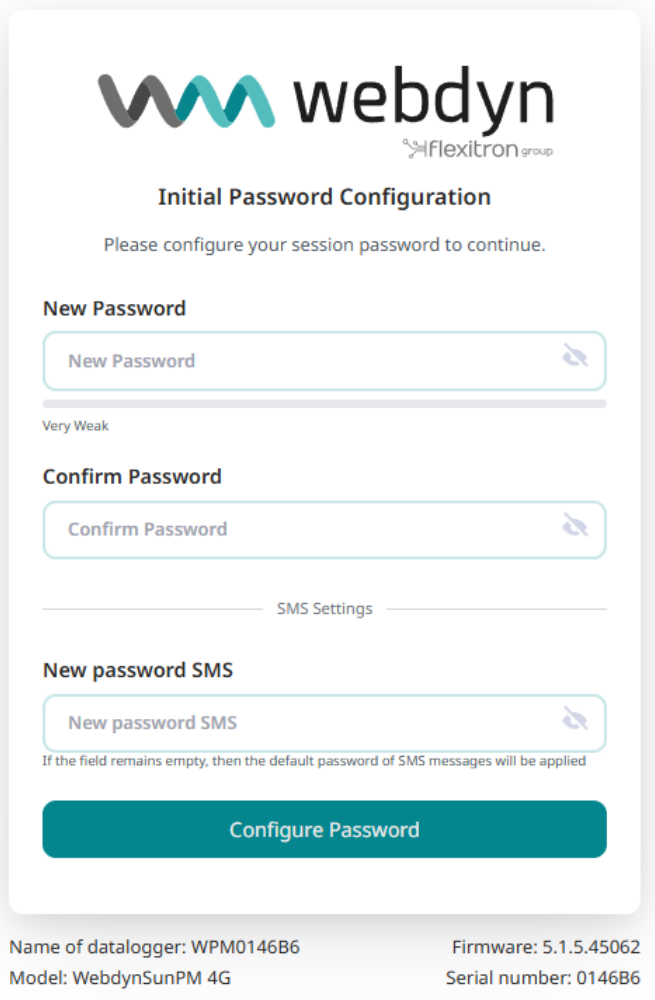
- Launch the web browser: the web interface is compatible with the latest versions of browsers: Firefox, Chrome and Edge. Older versions may work but they are not supported (IE 7 for example).
- Enter the concentrator IP address in your browser (the default address is: https://192.168.1.12 for LAN1 and https://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

When connecting to the hub via the HTTPS protocol, the browser may display a security warning. This message appears because the hub is using a self-signed certificate. On the first connection, it is necessary to access the browser's advanced options and manually validate the security exception in order to allow access to the interface.

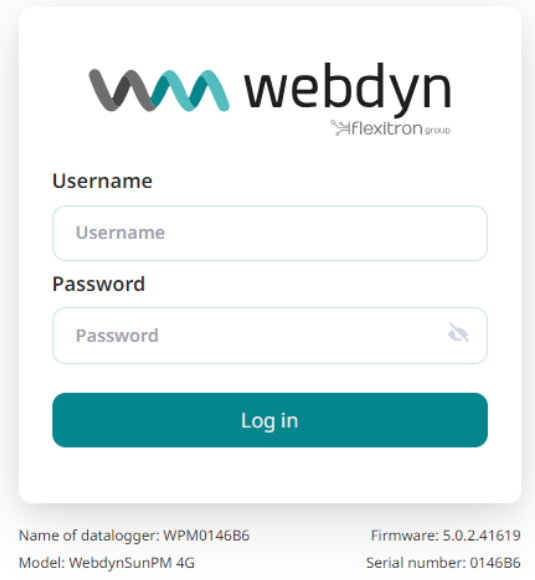
The hub remains locally insecurely accessible to addresses: http://192.168.1.12 for LAN1 and http://192.168.2.12 for LAN2 only if no gateway has been configured in the LAN settings.

- On the first connection, a window for configuring the password for web access and the password for SMS encryption is displayed:



- Enter the desired access password using the strength indicator to check that it complies with security requirements. Then specify a desired SMS password for the encryption of SMS exchanges with the hub.

- An authentication window is displayed:

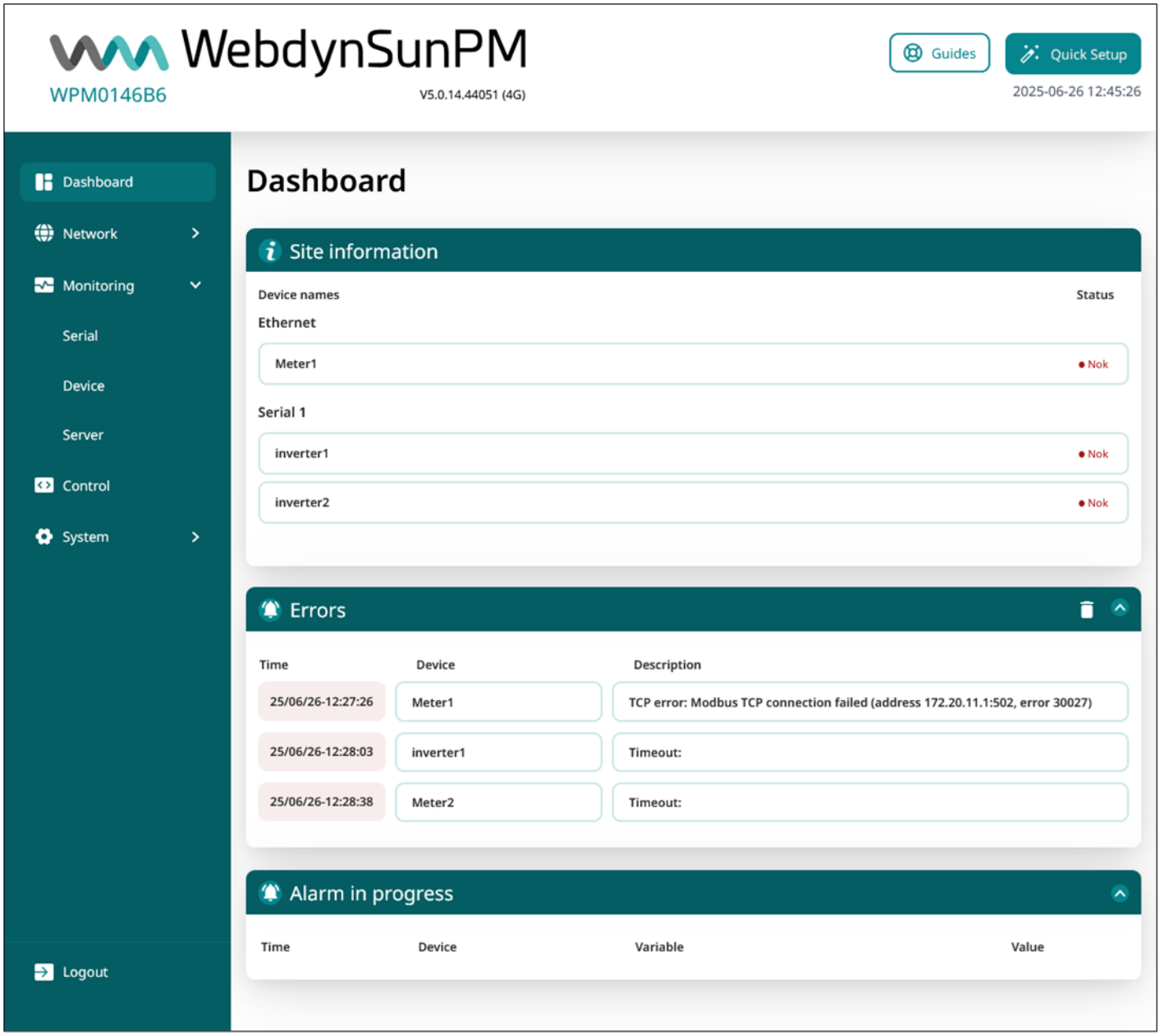


Enter your login and password:

Identifier
userhigh

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 3.2.5.1.2: "Password")* or using the configuration file (*see section 3.1.2.1.1: ""<UID>_config.ini" file"*).

- The Home page is displayed:



The "Dashboard" menu provides information about the site and the concentrator, as well as general device status, current alarms and communication errors.

Essential site information is at the top of the screen.

The identifier panel contains a list of configured devices and their status.

The possible device statuses are:

Status	Description
Ok	The device has been found or the current configuration is operational. (At least one request from the equipment has responded well, the other requests may be unanswered)
Nok	The device has not been found or the current configuration is not operational. (At least one request from the device is in error and the status of the other requests does not matter).

Errors	The device has been found but one or more variables in the definition file are not operational.
Unknown	Device status unknown.

Click on a device to go directly to its configuration page.

The "Errors" panel is used to display the list of errors detected on the different configured devices. Clicking the trash icon in the panel resets the list of detected errors. In that case, all new errors will be displayed.

Further down, the "Alarms in progress" panel contains the ongoing alarms.

There are 2 buttons on the web interface banner:

- "Guides: for help with installing and operating the concentrator.
- "Quick Setup": used to trigger a quick installation aid for the concentrator.

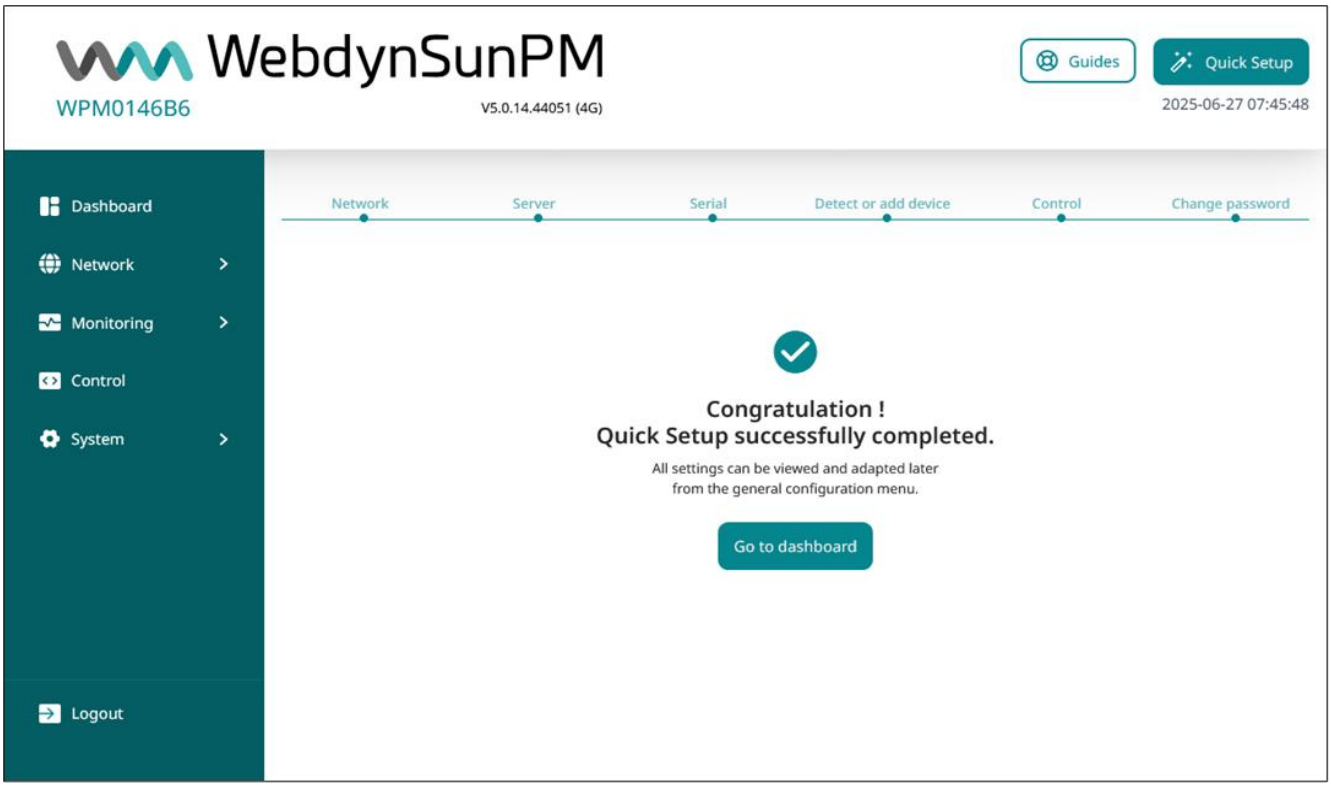
3.2.1 Quick Setup

To make it easier to configure the WebdynSunPM, the concentrator includes a function to configure the different network interfaces and device configurations with an automatic device discovery function and the option of activating services built into into the concentrator. “Quick Setup" simplifies concentrator configuration by guiding the installer through simple steps.

“Quick Setup” has 6 steps:

1. "Network”: Used to configure the Ethernet and Modem interface
2. "Server”: Used to configure the connection to the remote servers.
3. "Serial”: Used to configure the serial ports on the concentrator
4. "Detect or add device”: Used to add or detect the devices connected to the concentrator.
5. "Control”: Used to activate the services on the concentrator
6. "Change password”: Used to change the web interface password.

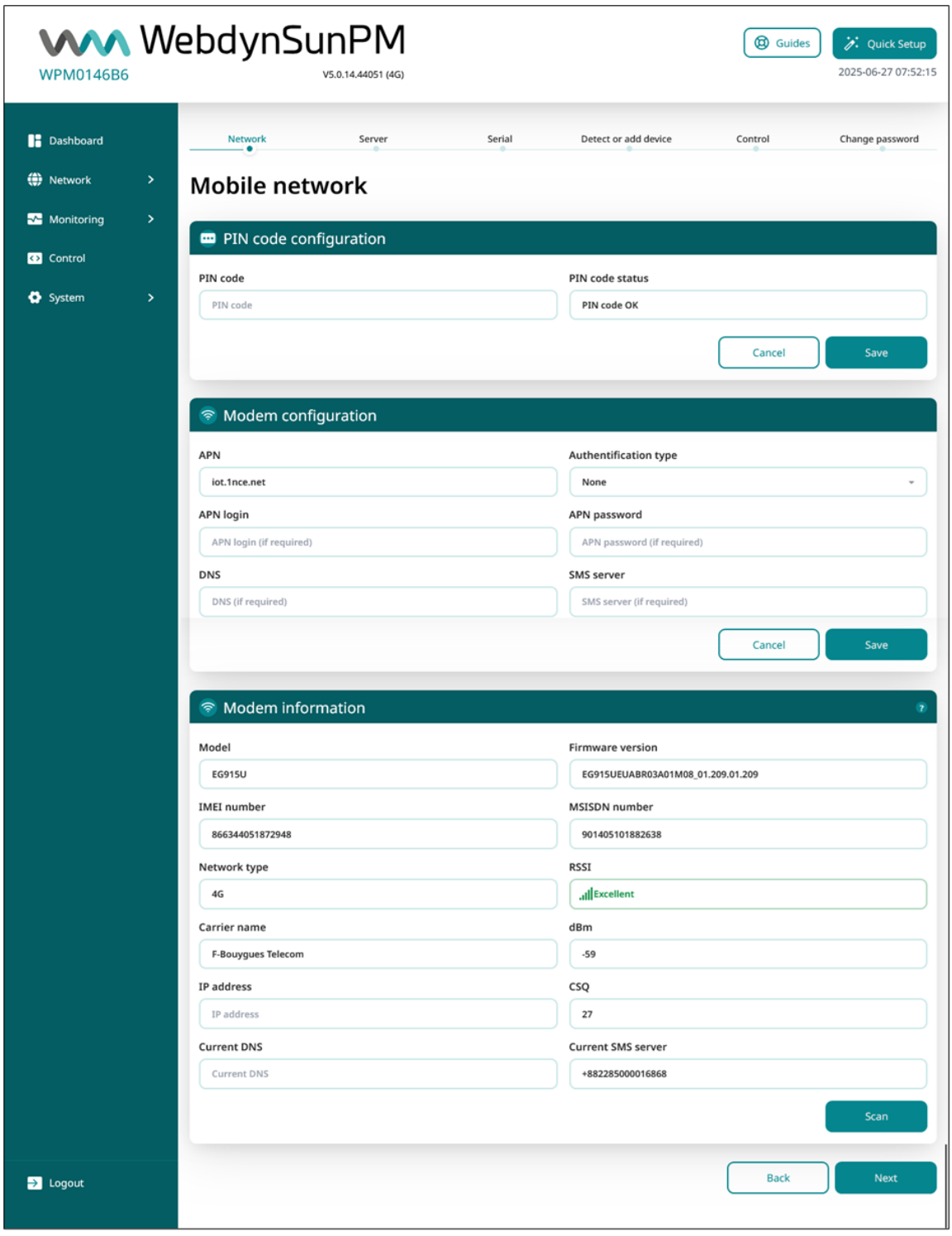
To move on to the next step, press the "Next" button on each page. In the final step, a message indicating that the Quick Setup is complete is displayed with a button to redirect to the concentrator's main Dashboard page.



3.2.1.1 "Network”

The “Local Network” 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 be part of 2 different Ethernet networks. (*See section3.2.2.1: “Ethernet (Local)”*)

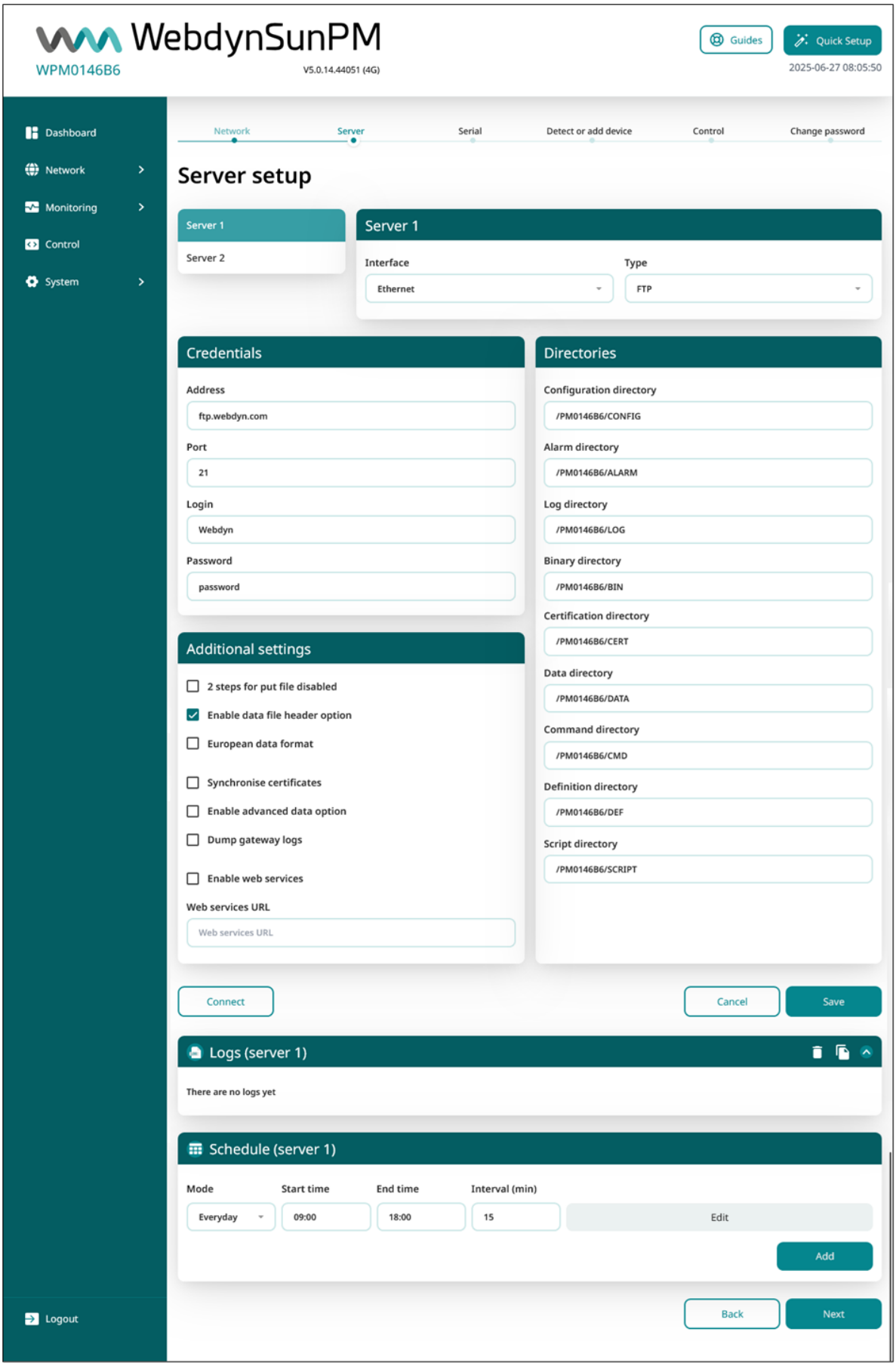
The “Mobile network” page is used to configure the modem and get network information. (*see section 3.2.2.2: “Modem (Mobile)”*) To use the modem, the SIM card must be inserted into the product (*see section 2.4.2.2: “SIM card”*).



3.2.1.2 "Server"

The “Server setup” page is used to configure the 2 servers available on the concentrator and to schedule the synchronisation times. This synchronisation can also be carried out locally on an SD card. (*See section3.2.3.3:*

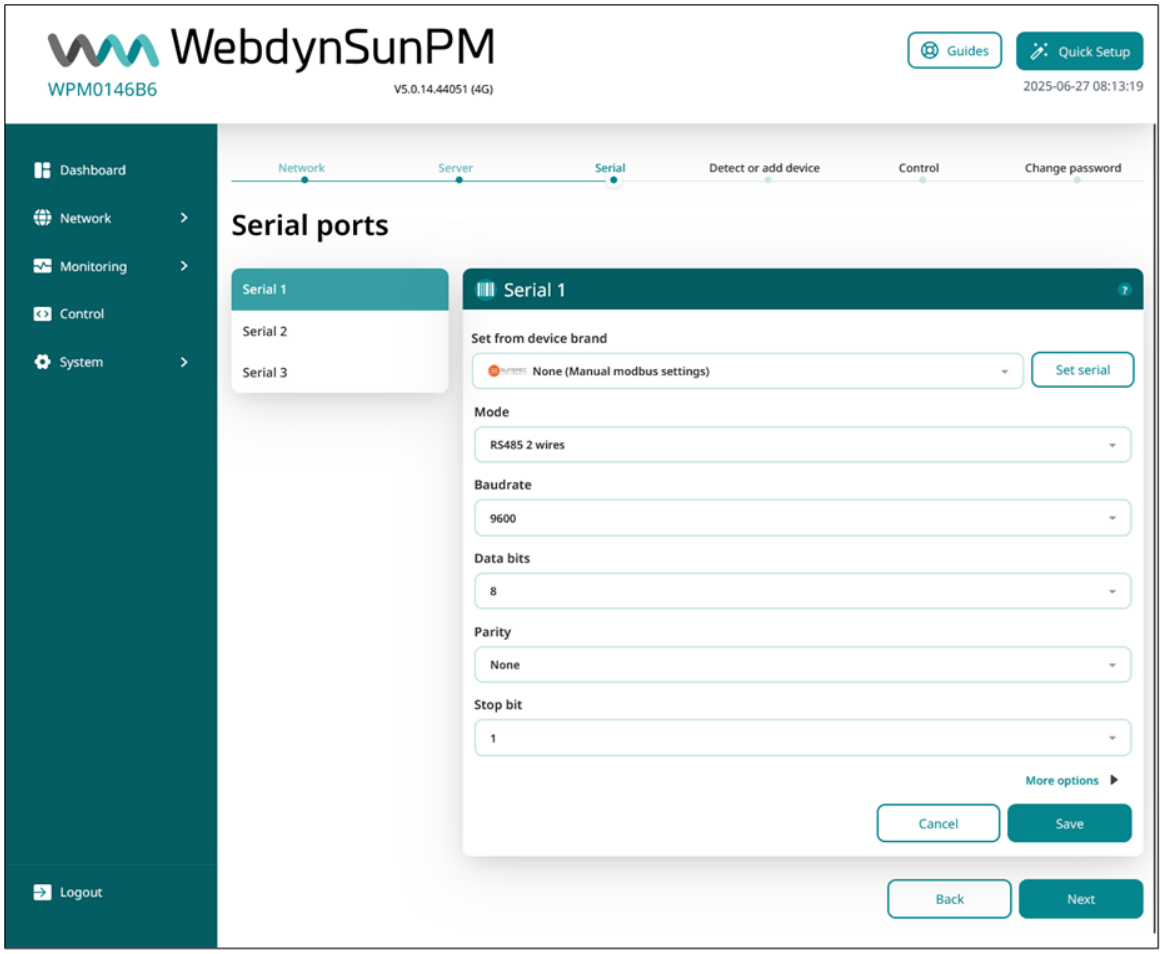
“Server”)


WebdynSunPM
WPM0146B6
V5.0.14.44051 (4G)
Guides
Quick Setup
2025-06-27 08:05:50
Dashboard
Network
Monitoring
Control
System
Logout
Network
Server
Serial
Detect or add device
Control
Change password
Server setup
Server 1
Server 2
Server 1
Interface
Ethernet
Type
FTP
Credentials
Address
ftp.webdyn.com
Port
21
Login
Webdyn
Password
password
Directories
Configuration directory
/PM0146B6/CONFIG
Alarm directory
/PM0146B6/ALARM
Log directory
/PM0146B6/LOG
Binary directory
/PM0146B6/BIN
Certification directory
/PM0146B6/CERT
Data directory
/PM0146B6/DATA
Command directory
/PM0146B6/CMD
Definition directory
/PM0146B6/DEF
Script directory
/PM0146B6/SCRIPT
Additional settings
2 steps for put file disabled
Enable data file header option
European data format
Synchronise certificates
Enable advanced data option
Dump gateway logs
Enable web services
Web services URL
Web services URL
Connect
Cancel
Save
Logs (server 1)
There are no logs yet
Schedule (server 1)
Mode
Start time
End time
Interval (min)
Everyday
09:00
18:00
15
Edit
Add
Back
Next

3.2.1.3 "Serial”

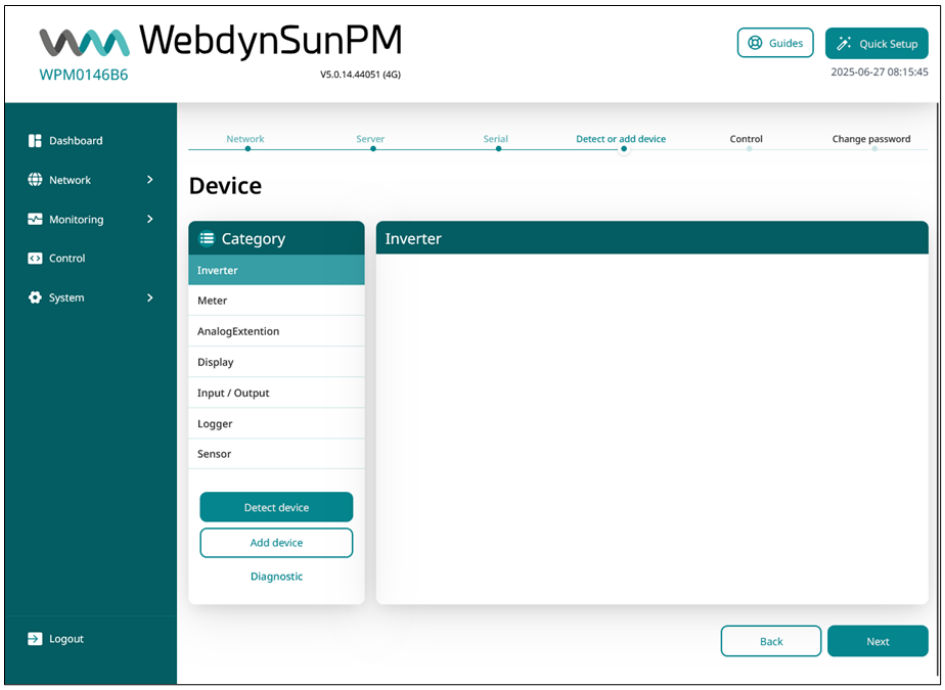
The “Serial ports” page is used to configure 3 RS485/422 serial ports which each have their own configuration and output. (see section *0*: “

Serial link”)



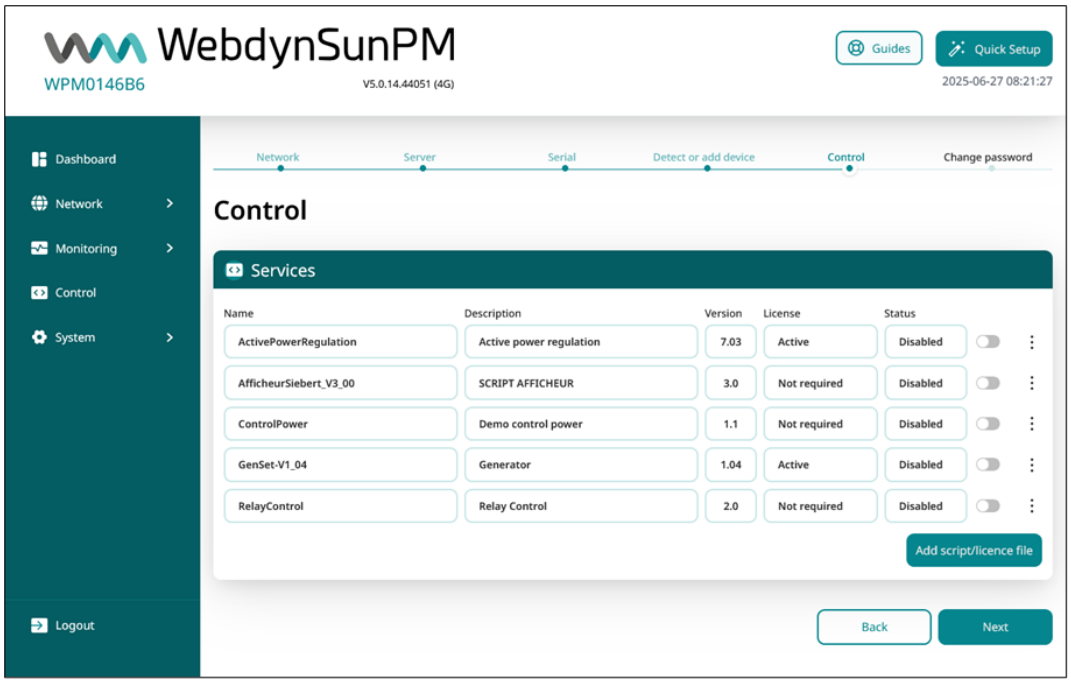
3.2.1.4 "Detect or add device”

The "Device" page is used to configure devices manually or by detecting them. *(See section 3.2.3.2: “Devices”)*



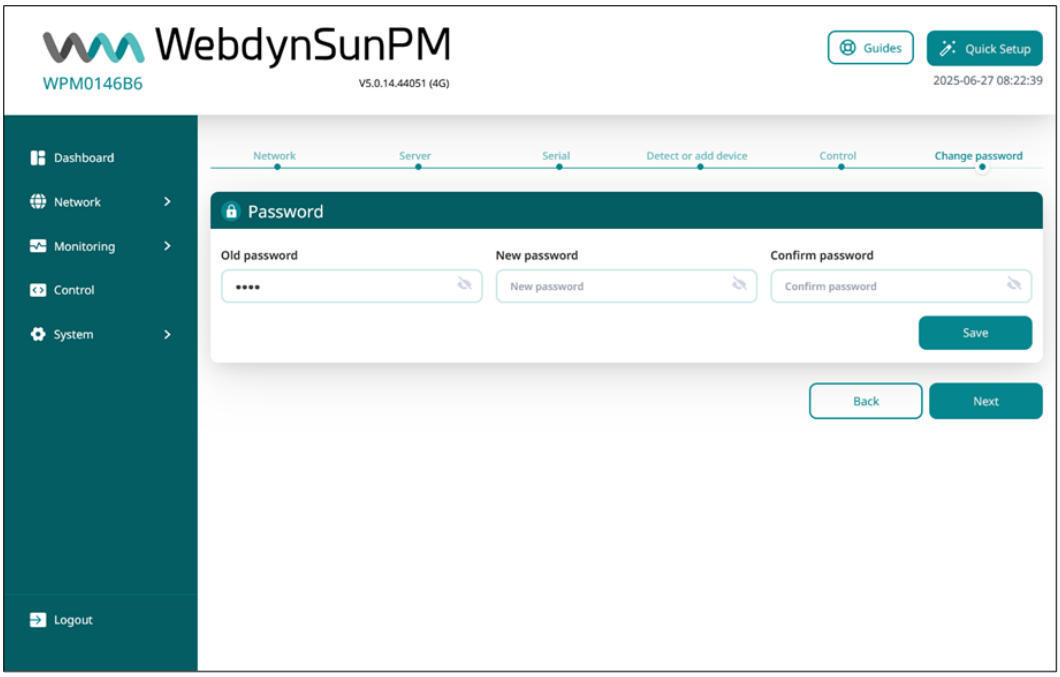
3.2.1.5 "Control"

The "Control" page is used to manage services (scripts). (*See section3.2.4: "Control"*)



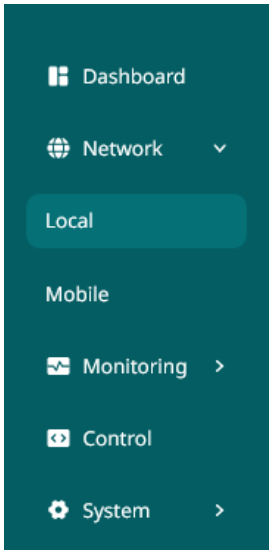
3.2.1.6 "Change Password"

The "Password" page is used to modify the password to access the concentrator web interface. (see section *3.2.5.1.2*: *"Password"*)



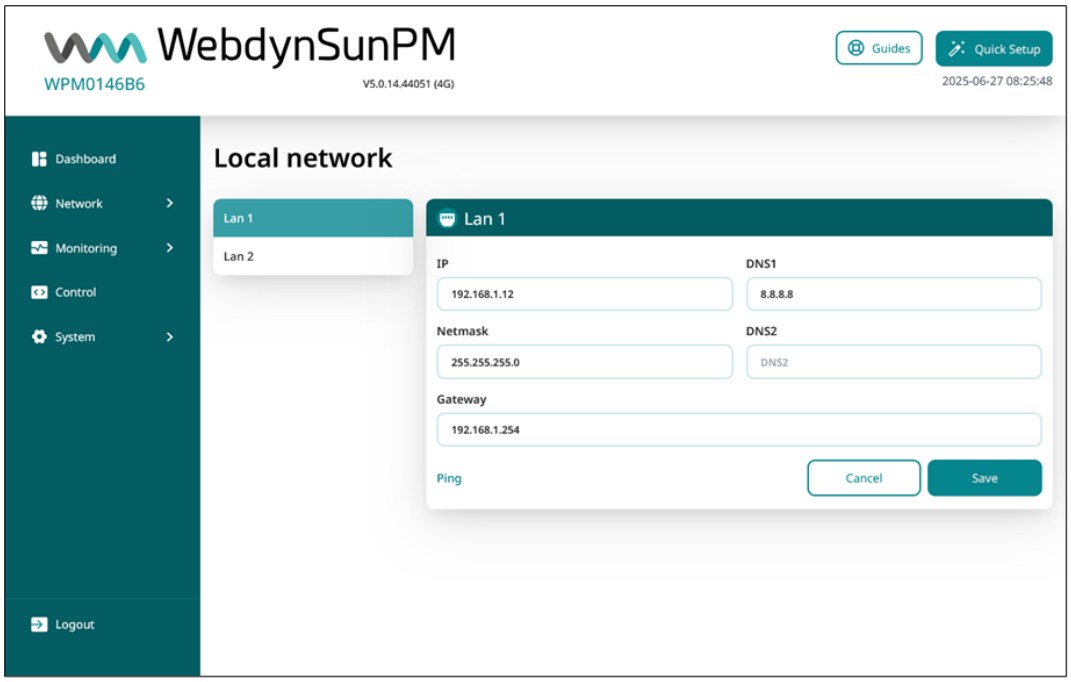
3.2.2 Network

All the concentrator network settings are grouped together in the “Network” menu. The network settings are split into two sub-parts on the menu.



3.2.2.1 Ethernet (Local)

The “Local” sub-menu 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 “Ping” button.

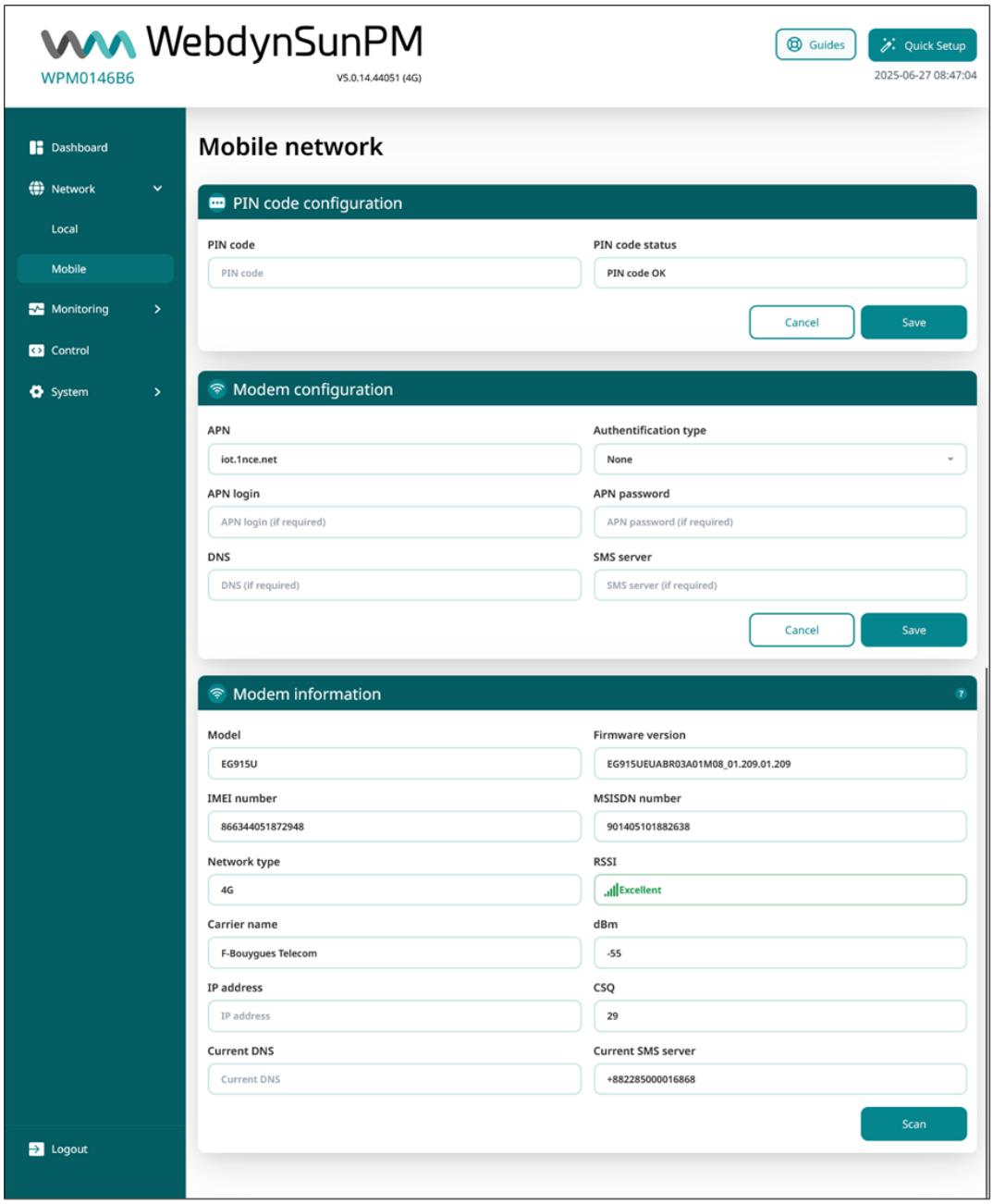
The 2 Ethernet interface settings are:

Web interface	Parameter in <UID>_daq.csv	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.

3.2.2.2 Modem (Mobile)

The “Mobile” sub-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”*).


WebdynSunPM
WPM0146B6
V5.0.14.44051 (4G)
Guides
Quick Setup
2025-06-27 08:47:04
Dashboard
Network
Local
Mobile
Monitoring
Control
System
Logout
Mobile network
PIN code configuration
PIN code
PIN code
PIN code status
PIN code OK
Cancel
Save
Modem configuration
APN
iot.1nce.net
Authentification type
None
APN login
APN login (if required)
APN password
APN password (if required)
DNS
DNS (if required)
SMS server
SMS server (if required)
Cancel
Save
Modem information
Model
EG915U
Firmware version
EG915UEUABR03A01M08_01.209.01.209
IMEI number
866344051872948
MSISDN number
901405101882638
Network type
4G
RSSI
Excellent
Carrier name
F-Bouygues Telecom
dBm
-55
IP address
IP address
CSQ
29
Current DNS
Current DNS
Current SMS server
+882285000016868
Scan

The modem parameters are:

Web interface	Parameter in <UID>_daq.csv	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. • **CHAP**: CHAP type authentication The login and password must be entered below. • **Both**: CHAP or PAP type authentication. The login and password must be entered below.
APN login	login	Your mobile operator's user name (optional depending on the operator)
APN password	password	Your mobile operator's password (optional depending on the operator)
SMS server	server	Text messaging centre. The text message centre is used to manage text messages from the concentrator. Enter the phone number for the text messaging centre you want to use instead of your mobile service provider. For example: "+33989004000"
DNS	dns	DNS server. DNS (Domain Name System) servers translate explicit internet addresses (for example, www.webdyn.com) into their corresponding IP addresses. Enter the address of the DNS server you want to use instead of your mobile service provider. For example, you can use the google DNS: "8.8.8.8"

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

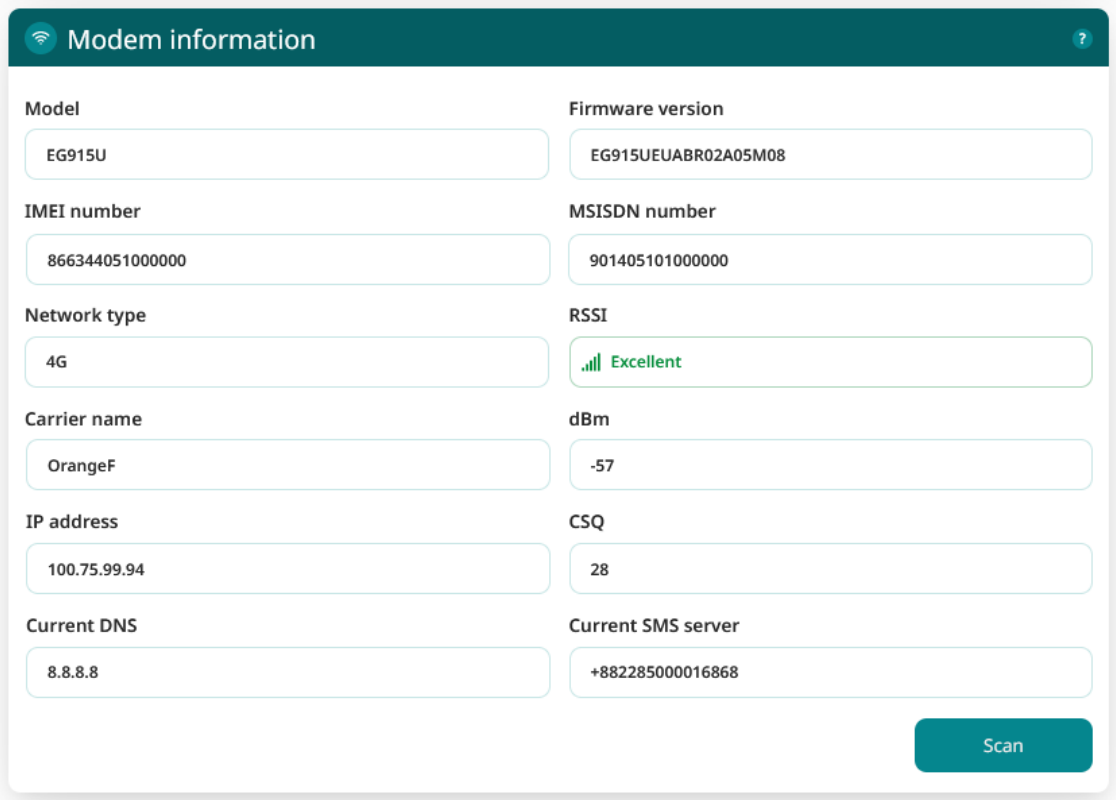
When an APN is correctly set, the IP connection to the mobile network is permanent.

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. The code 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:

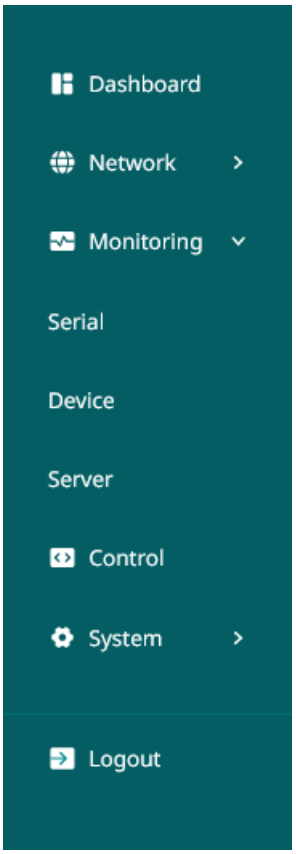


This information is:

Modem information	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
MSISDN number	Inserted SIM card MSISDN number used to uniquely identify a subscription on a mobile network. (subscriber's phone number)
Network Type	The type of network to which the modem is connected (2G/3G/4G).
RSSI	Indication of the modem's reception power level in number of bars
Carrier name	The name of the operator to which the modem is currently connected.
dBm	Signal level returned by the modem interpreted in dBm of -113 to -51
IP Address	IP address assigned automatically by the mobile operator
CSQ	Reception signal level returned by the modem of 0 to 31
Current DNS	DNS address currently used by the modem connection.
Current SMS server	Text messaging centre currently in use.

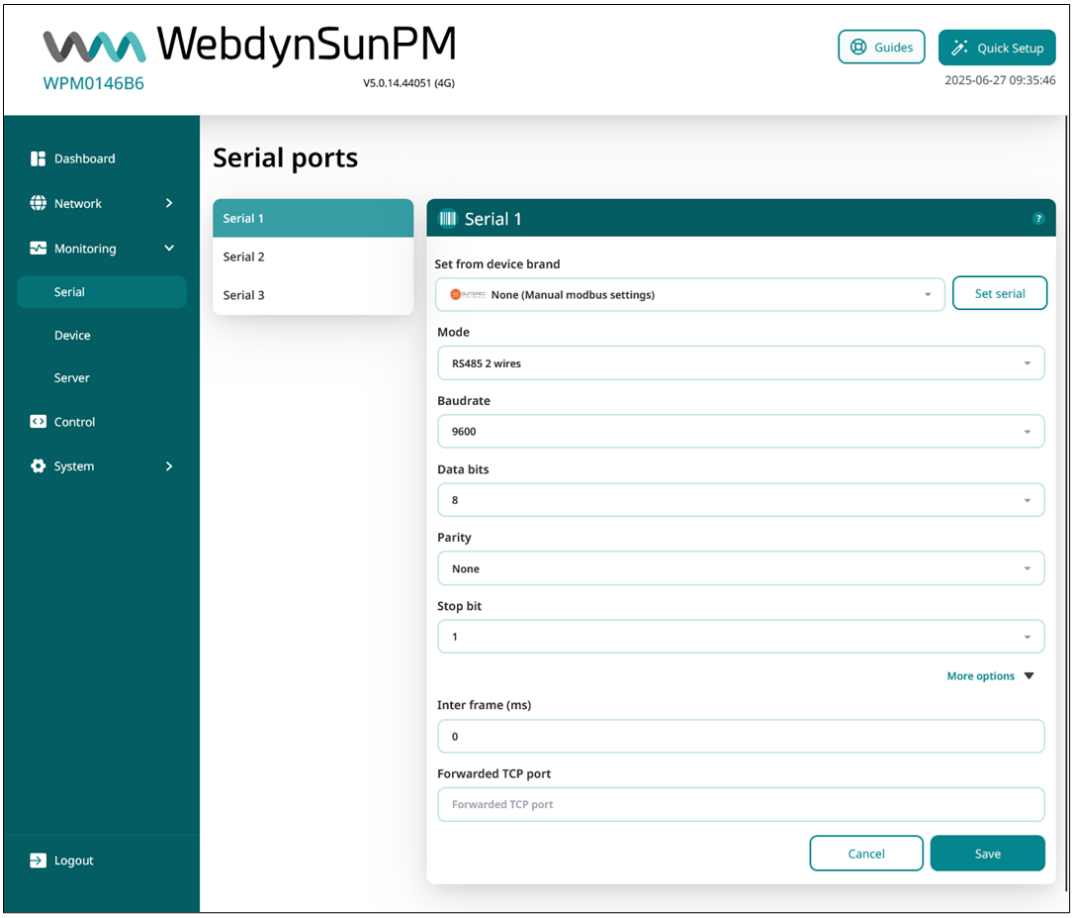
3.2.3 Monitoring

All the concentrator monitoring is grouped together in the "Monitoring" menu. The parameters are split into two sub-parts on the menu.



3.2.3.1 Serial link

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



The possible settings for each serial port are:

Web interface	Parameter in <uid>_daq.csv	Description

Set from device brand	protocol	The protocol type for this serial interface:
		• **None (Manual modbus settings):** serial port configured in modbus RTU mode
		• **Inverter brand**
		  ○ ABB
		  ○ Cefem
		  ○ CyberPower
		  ○ DELTA
		  ○ Fronius
		  ○ Goodwe
		  ○ Growatt
		  ○ Huawei
		  ○ Ingeteam
		  ○ KACO
		  ○ Kaco Modbus

		○ Kostal ○ SAJ ○ SMANET ○ SolarEdge ○ SOLARMAX ○ Solis ○ Sungrow *(see specific proprietary protocol appendix)*.
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
Inter frame (ms)	interframe	Waiting time between 2 frames exchanged on the serial port. This time is expressed in ms. *See 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.

The "Set serial" button automatically sets the default parameters for a device manufacturer. In that case, the parameters are pre-filled with the default parameters for the selected inverter brand.

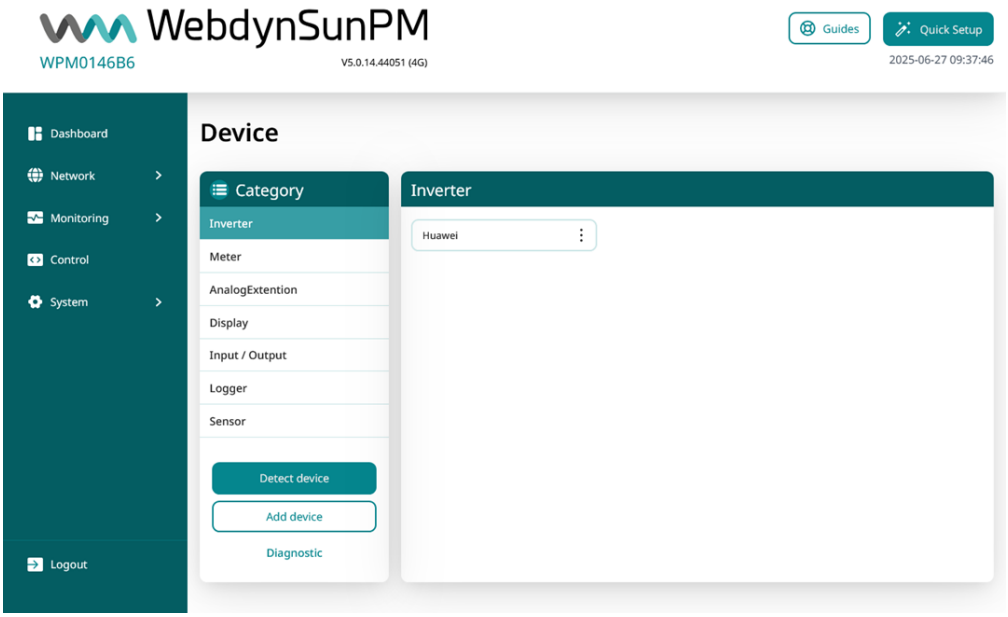
If the inverter default parameters have not been modified, it is then possible to validate the serial port configuration. Otherwise, its parameters should be adapted to those set in the inverter.

3.2.3.2 Devices

Devices can be added 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 using a command text message
- By manually editing the configuration using the web interface

Device configuration using the web interface and with access using the "Device" part will be detailed in this section.



By default, devices are split into 3 categories:

- **Inverter**: Includes all the inverters managed by the concentrator.
- **Meter**: Includes energy and other meter types.
- **Input/Output**: Includes the devices connected to the concentrator inputs and outputs.

New categories can be created from the device definition file. (*See section3.1.2.2: "Connected device definition"*)

> To add a device, you must first update the definition file library on the hub *(see chapter* ***Erreur ! Source du renvoi introuvable.*** *: "****Erreur ! Source du renvoi introuvable.****")* or manually add the device definition file *(see chapter* ***Erreur ! Source du renvoi introuvable.****: "****Erreur ! Source du renvoi introuvable.****").*

Inverters are the following:

- *Supported Modbus inverters:*
  - *ABB*
  - *Cefem*
  - *CyberPower*
  - *Fronius*
  - *Goodwe*
  - *Growatt*
  - *Huawei*
  - *Ingeteam*
  - *Kaco*

- *Kostal*
- *SAJ*
- *SMA*
- *SolarEdge*
- *Solis*
- *Sungrow*

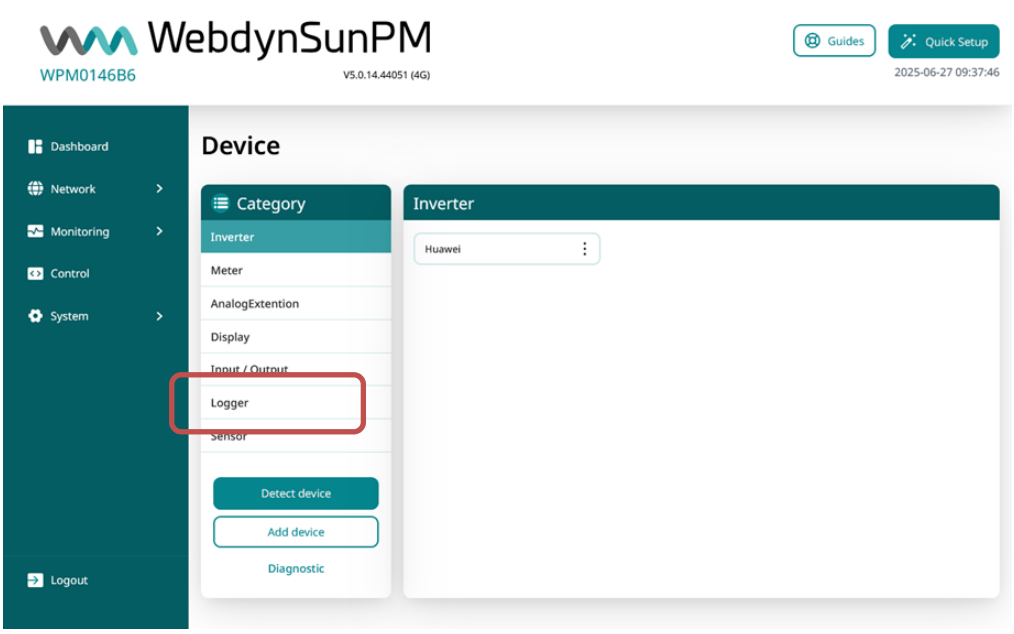
- *SunSpec*: The modbus Inverters in the tables meeting the SunSpec standard specification (http://www.sunspec.org/) can be detected and configured automatically using the Ethernet (Modbus TCP) or serial (Modbus RTU) connection.
- *Proprietary protocol: (see specific proprietary protocol application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

3.2.3.2.1.1 Detection of a supported or Sunspec-compatible inverter

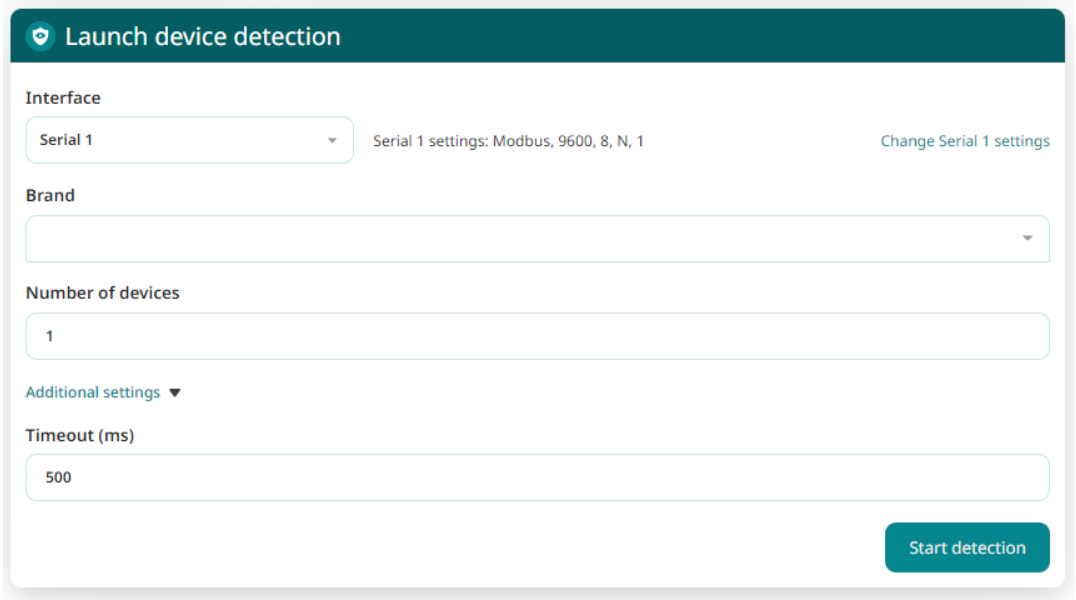
For SunSpec detection to work, the definition file library must be present on the hub. *(See chapter **Erreur ! Source du renvoi introuvable.** : "**Erreur ! Source du renvoi introuvable.**")*

The automatic detection of a SunSpec inverter requires the following steps:

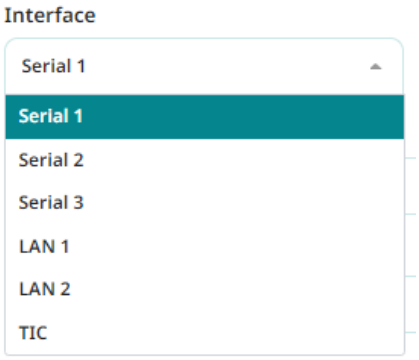
- Connect the inverter to the concentrator on one of the serial connections or on the Ethernet network
- Check the device configuration:
  - For serial connections, check that the configuration is the same on the concentrator and the device.
  - For an Ethernet connection: check that the network configuration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



- The detection page is displayed:



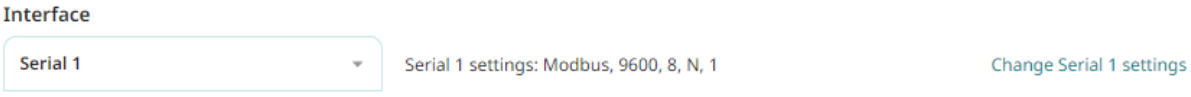
- Select the interface you want to use to detect



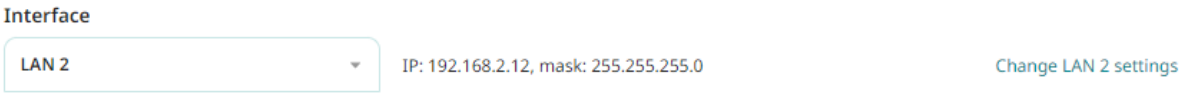
The interfaces are of the "Serial", "LAN" or "TIC" (USB accessory) type

For each interface type, the parameters are displayed along with a link to the WEB page on which they can be changed.

Serial:



LAN:



A list of device brands is proposed depending on the interface and configured protocol.

For example:

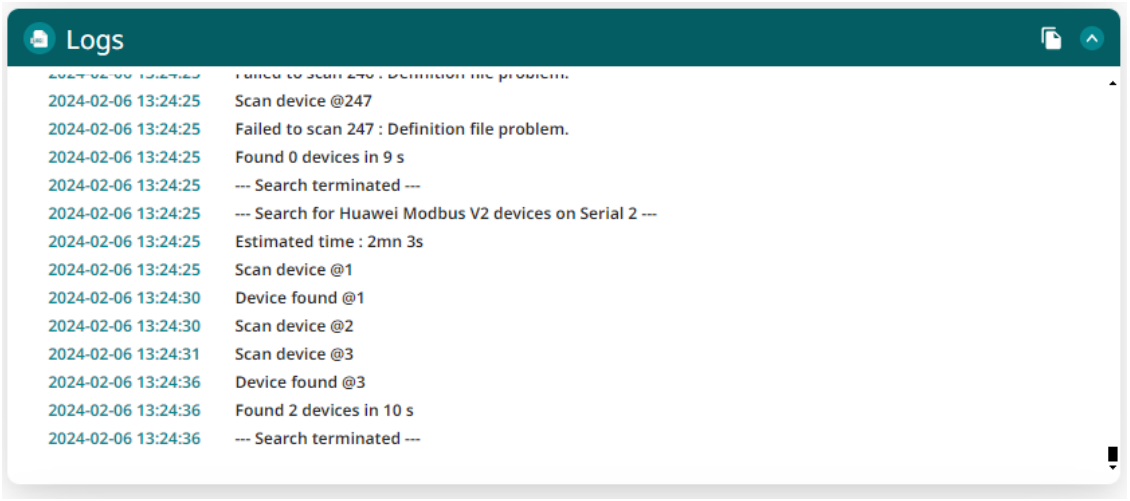
The inverters can be detected by the SunSpec protocol.

When the inverter brand is taken into account, it is advisable to use detection by brand so that all inverters of the brand can be detected, whether they are Sunspec or not.

- Enter the number of devices to detect in the “Number of devices” field. The default value is “1”.

The “Timeout” parameter can be used to modify the concentrator waiting time (in ms) to wait for a response sent late by a device. However, it is advisable not to set too high a value, as it could considerably slow down the automatic detection.

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



The first progress window lines show the detection start date and time and an estimate of the detection time.

The displayed messages are used to monitor progress. The example above is for SunSpec detection.

2023-02-03 9:56:10:NON SUNSPEC device found at @1 192.93.121.23:502

SUNSPEC detection:

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:

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:

2023-02-03 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:

2023-02-03 09:56:21:Found table 1:66 at 40004 on 126 192.93.121.23:502

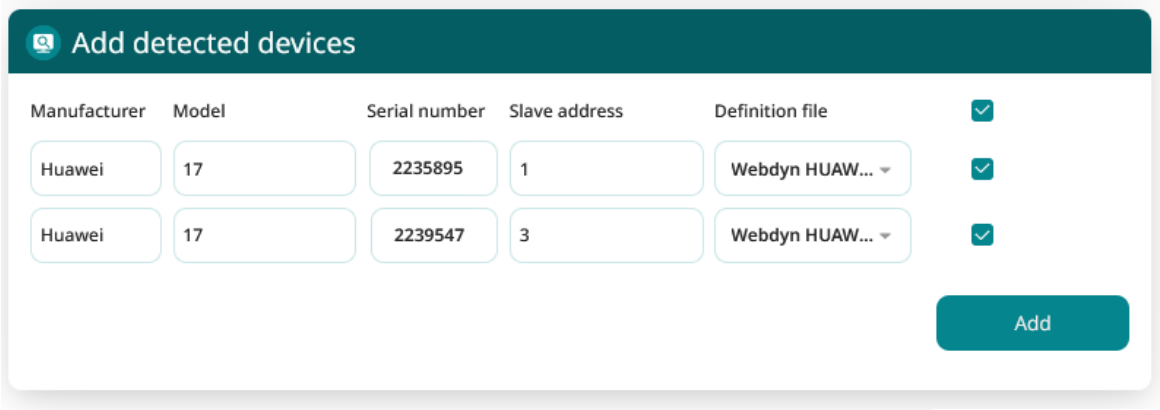
Means that table identifier “1” of size “66” registers was detected on the modbus TCP device at IP address “192.93.121.23” at register “40004” and modbus address “126”

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

2023-02-03 09:56:33:End of SunSpec detection on ethernet1: 1 devices found.

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

On completion of detection, the last detection page is used to view all the detected devices and, eventually, to add them to the configuration.



This screen therefore displays all the detected devices as well as a certain number of data read from the Sunspec device tables (model, serial number, address, manufacturer) as well as the associated definition file name.

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.

When the detection is SUNSPEC, it is a dynamically generated file based on the tables described by the SUNSPEC consortium and implemented by the inverter manufacturer.

The variables for this new device will be generated according to the detected SunSpec tables. For each detected SunSpec table, the following variables will be created:

- *<idTable>_tableId*: this variable will contain the table identifier in 16-bit integer numeric format
- *<idTable>_tableSize*: this variable contains the table size in number of registers, in 16-bit integer format
- *<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 be composed of the table identifier followed by the variable name
- *<idTable>_<repeatBlock>_<variableName>*: for variables that come from a repeating block, the variables are created using the table identifier, the repeat number and the variable name so that the generated name is unique.

By default, the generated variables are of the “Parameter” type and will therefore have an “Action” code equal to **1**, except for variables for tables **101**, **102**, **103**, **111**, **112, 113**, **123**, **160** and **401** which will be created using the “Immediate” type, i.e. code **4**.

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

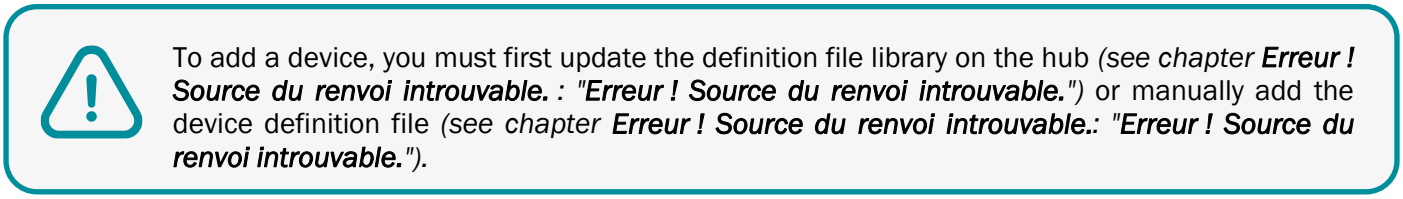
- *WMaxLimPct* in table 123 has the “**cmdPwrPercent**” tag
- *WMaxLimPct_RmpTms* in table 123 has the “**WMaxLimPct_RmpTms**” tag
- *WMaxLimPct_Ena* in table 123 has the “**WMaxLimPct_Ena**” tag
- *VarPct_Mod* in table 123 has the “**VArPct_Mod**” tag

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 name is different.

If the user does not add any devices, this list is deleted when a new detection is run or the web page is changed.

3.2.3.2.1.2 Proprietary protocol device detection

(see specific proprietary protocol application note)



To add a device, you must first update the definition file library on the hub *(see chapter **Erreur ! Source du renvoi introuvable.** : "**Erreur ! Source du renvoi introuvable.**")* or manually add the device definition file *(see chapter **Erreur ! Source du renvoi introuvable.**: "**Erreur ! Source du renvoi introuvable.**").*

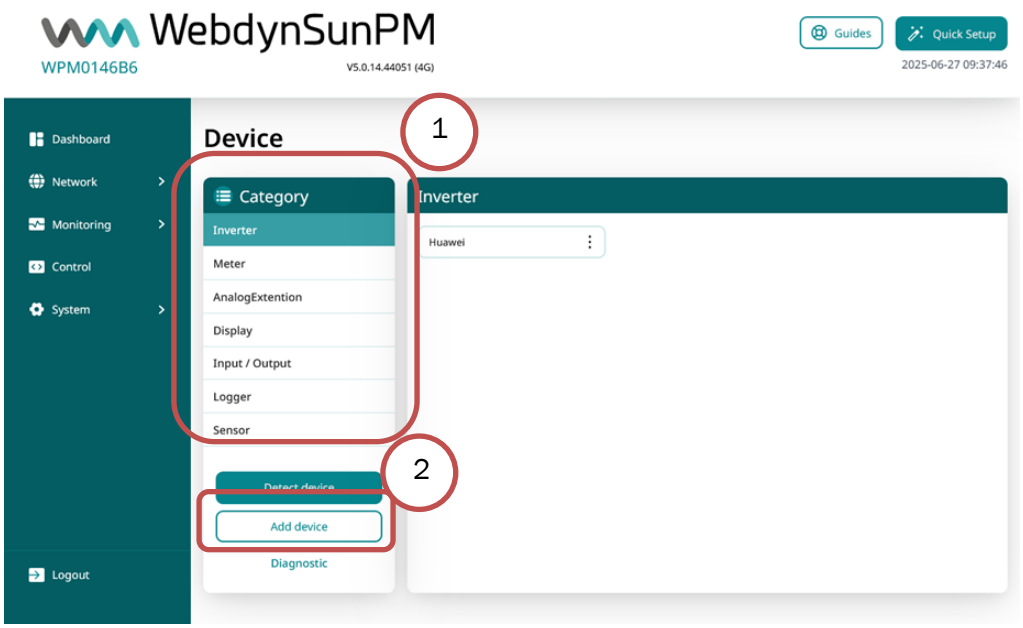
3.2.3.2.2 Manual device management

Devices can be managed manually using the web interface.

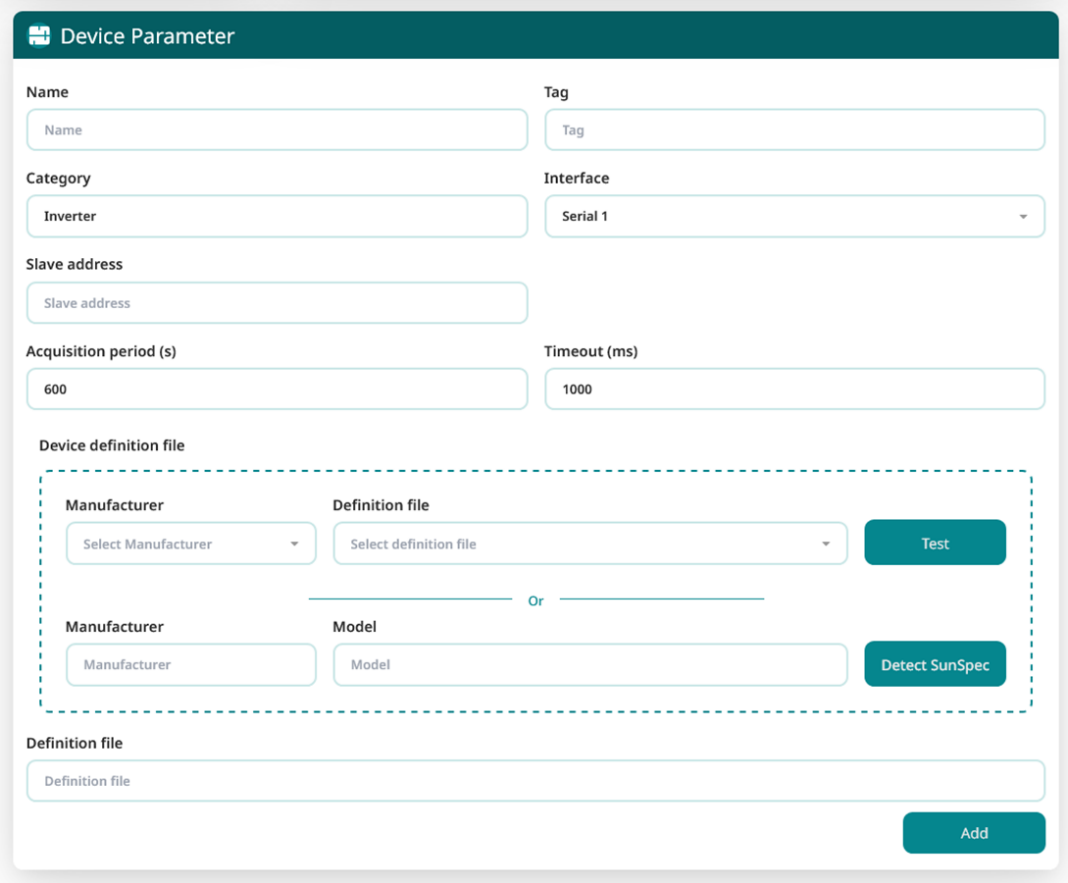
Everything is managed from the “Monitoring menu and then the “Device” menu described previously (*see section 3.2.3.2: “Devices”*).

3.2.3.2.2.1 Add a device

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



The next page is displayed:



On this form:

- Enter a device name in the "*Name*" field. This value will be used on the web pages and on the MQTT servers. The name must be unique.
- Give a tag to the device in the "*Tag*" field (optional). This field will be used in scripts.
- Provide the device slave address in the "*Slave address*" field
- If necessary, adjust the device acquisition period in the "*Acquisition period*" field. The acquisition period is the period during which data is recorded in the file (device collection is continuous). It is expressed in seconds.

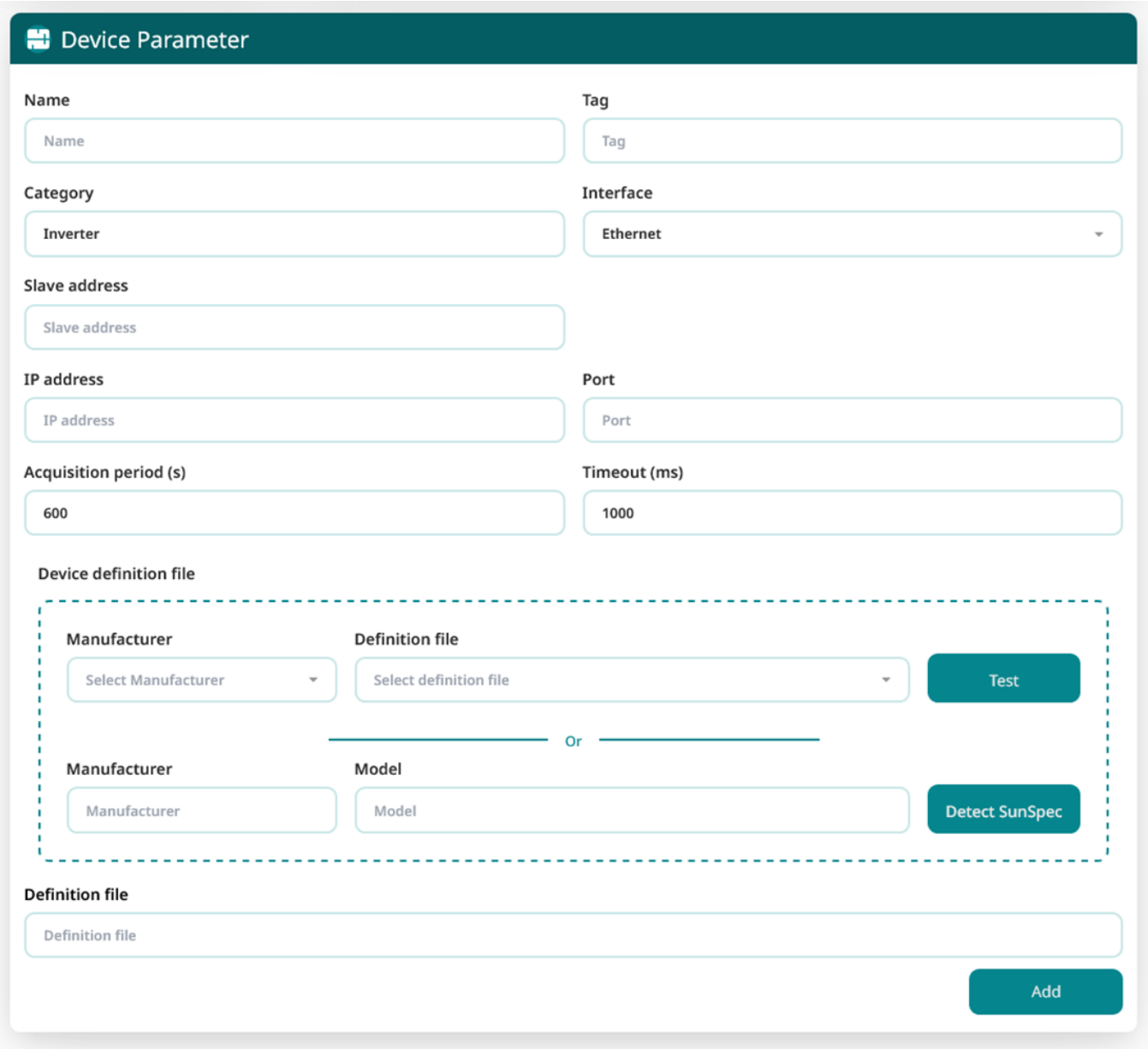
When acquisition period is set to 0, the device is only queried and no data file is generated

- If necessary, adjust the timeout delay in the "*Timeout (msec)*" field. If the device doesn't answer, the concentrator assumes the query has failed.
- Select the interface where the device is connected:

- **Ethernet**: if the device is connected to the Ethernet network. For example, for a “Modbus TCP” device
- **Serial1**, **Serial2** or **Serial3**: if the device is connected to one of WebdynSunPM's serial ports. For example, for a "modbus RTU” device.
- **Input/Output**: if you want to configure an input or output available on the concentrator.

The following parameters differ depending on the selected interface.

Ethernet:



For Ethernet type devices, the following fields are also displayed:

- *“IP address”*: enter the device’s network IP address
- *“Port”*: enter the port number to access the device. Usually port: 502

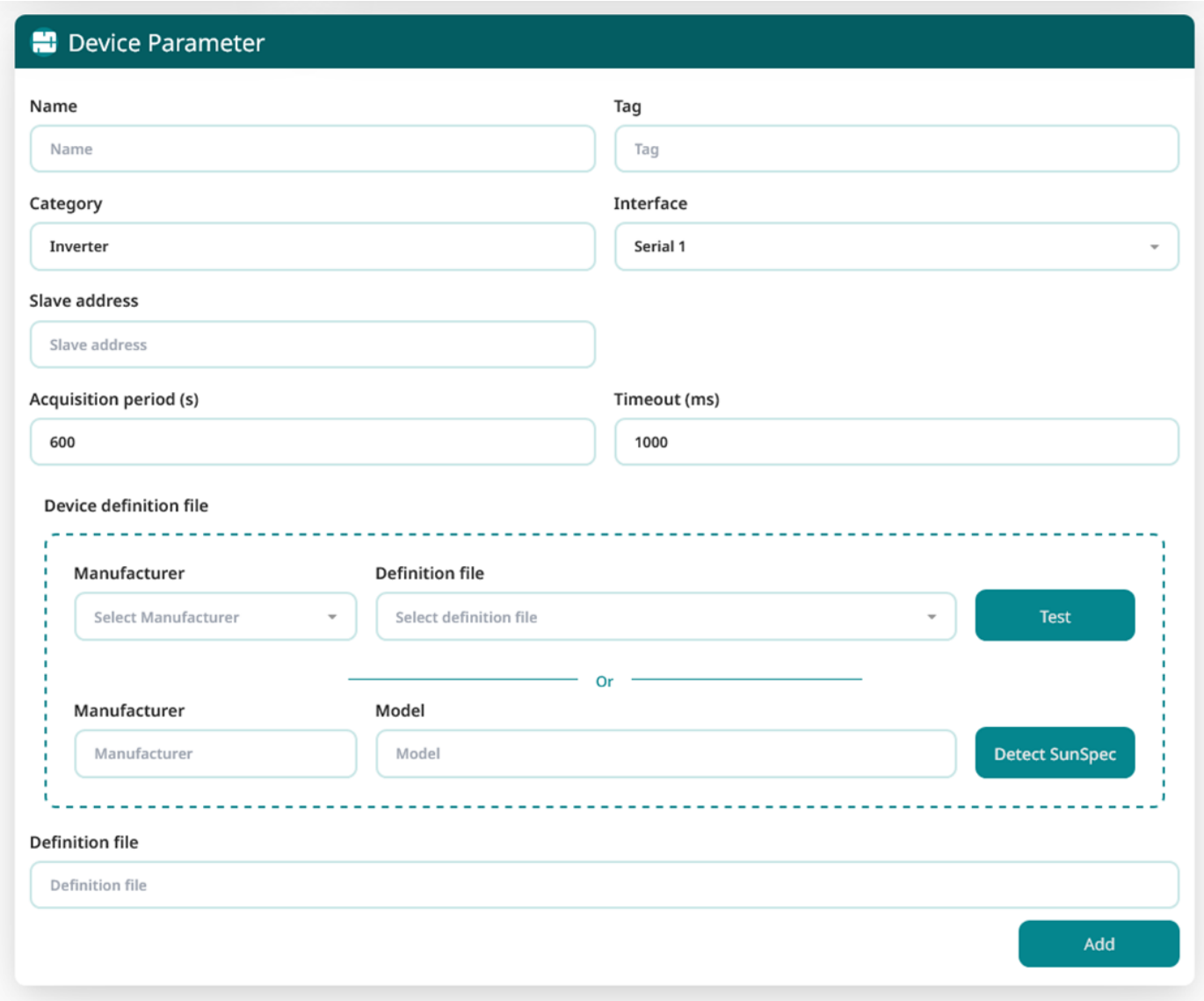
Select the device file to be added. There are possible 3 options:

- Using a “*From list*”: list of all the definition files available on the concentrator for the selected interface.

- Using a *From Brand*" list: list of manufacturer protocols available on the concentrator for automatic detection on the selected interface. If the device is found, its definition file will be generated automatically.

To finalise the addition of the Ethernet device to the concentrator, simply click the "*Add* " button.

Serial1, Serial2 or Serial3:



Select the device file to be added. There are possible 2 options:

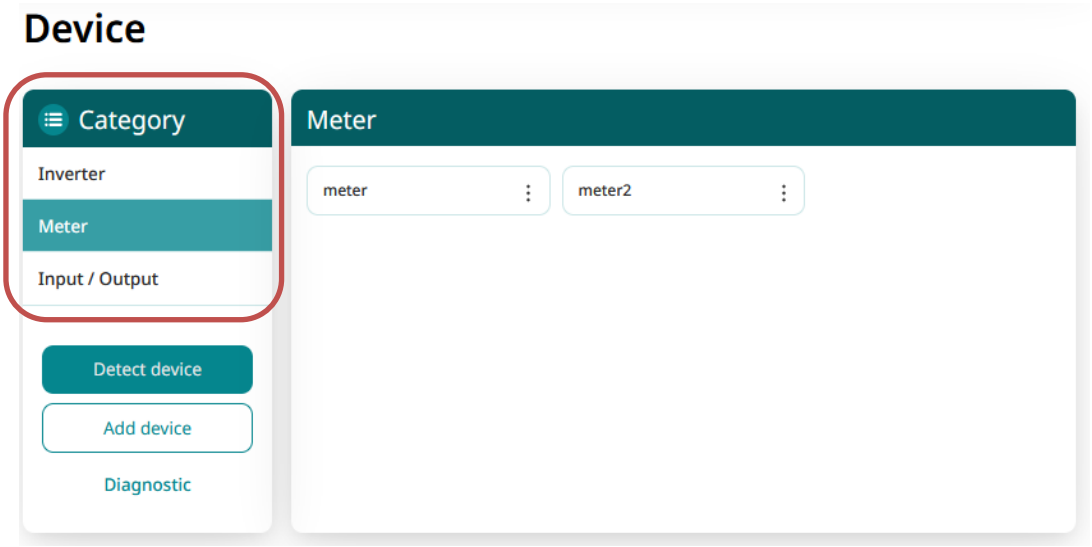
- Using a *From Manufacturer*" list: list of all related definition file available on the concentrator for automatic detection on the selected interface. Test button allow to verify the communication with the selected device.
- For a SunSpec automatic detection: when clicked, the device will be searched. The new definition file will be available in the list of the concentrator

To complete the addition of the Ethernet equipment to the hub, simply click on the "*Add*" button.

To add a device, you must first update the definition file library on the hub *(see chapter **Erreur ! Source du renvoi introuvable.** : "**Erreur ! Source du renvoi introuvable.**")* or manually add the device definition file *(see chapter **Erreur ! Source du renvoi introuvable.**: "**Erreur ! Source du renvoi introuvable.**")*.

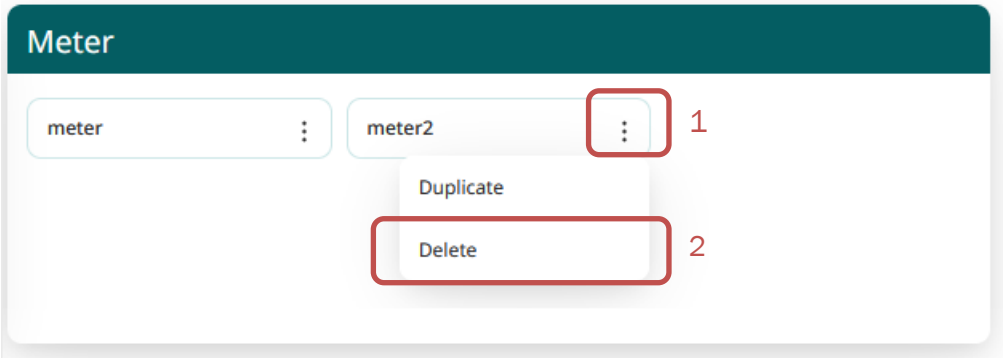
3.2.3.2.2.2 Deleting a device

To delete a device, first select the device category to display the device to be deleted:



There are 2 delete options.

Deletion from the device sub-menu:



Click the device sub-menu icon for the device you want to delete, then select "*Delete*".

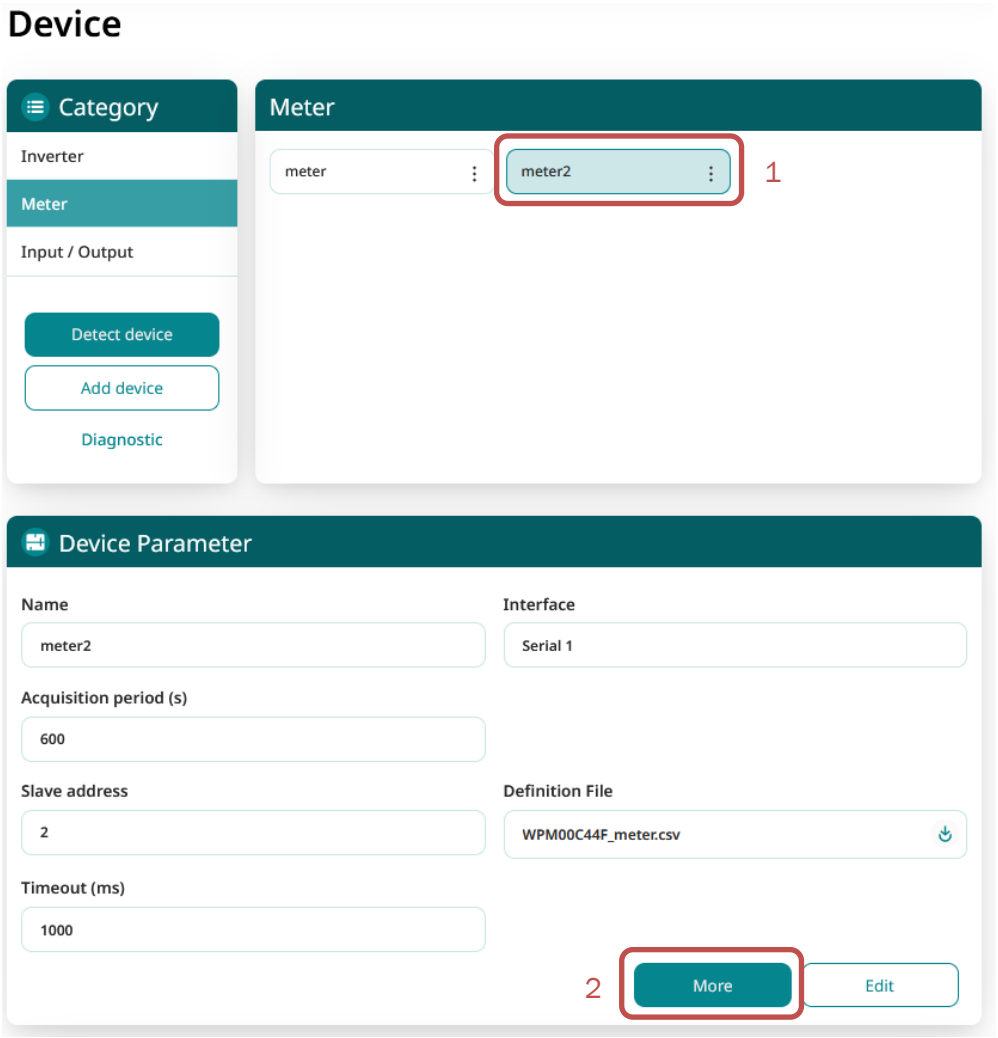
A dialogue box is displayed requesting confirmation.

After confirmation, the device is deleted from the concentrator.

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

Deletion from the device parameters:

Click the device to be deleted.



Then click the "*More*" button in the device parameters, and then on the trash icon under the device description:

A dialogue box is displayed requesting confirmation.

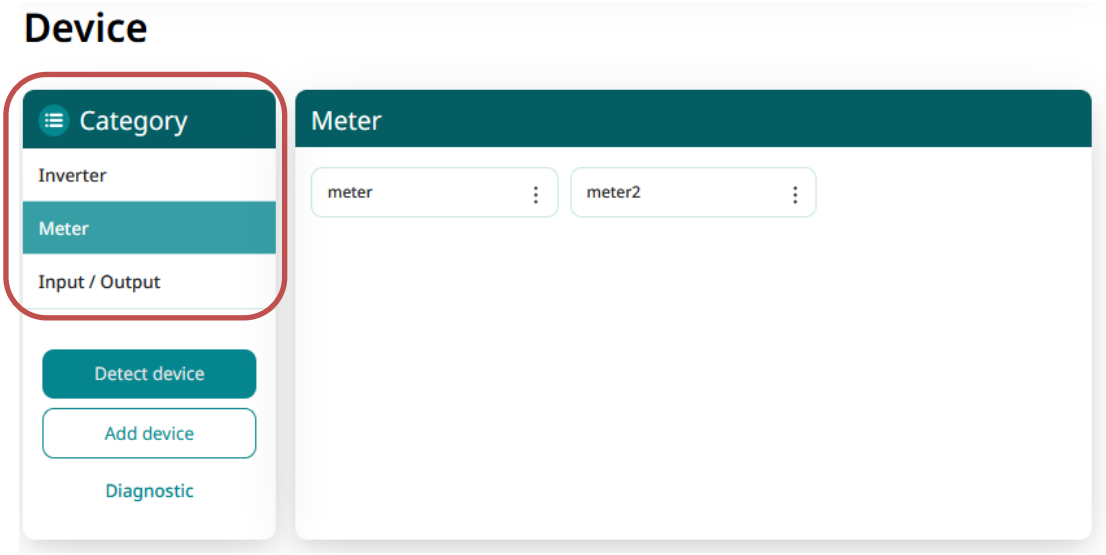
After confirmation, the device is deleted from the concentrator.

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

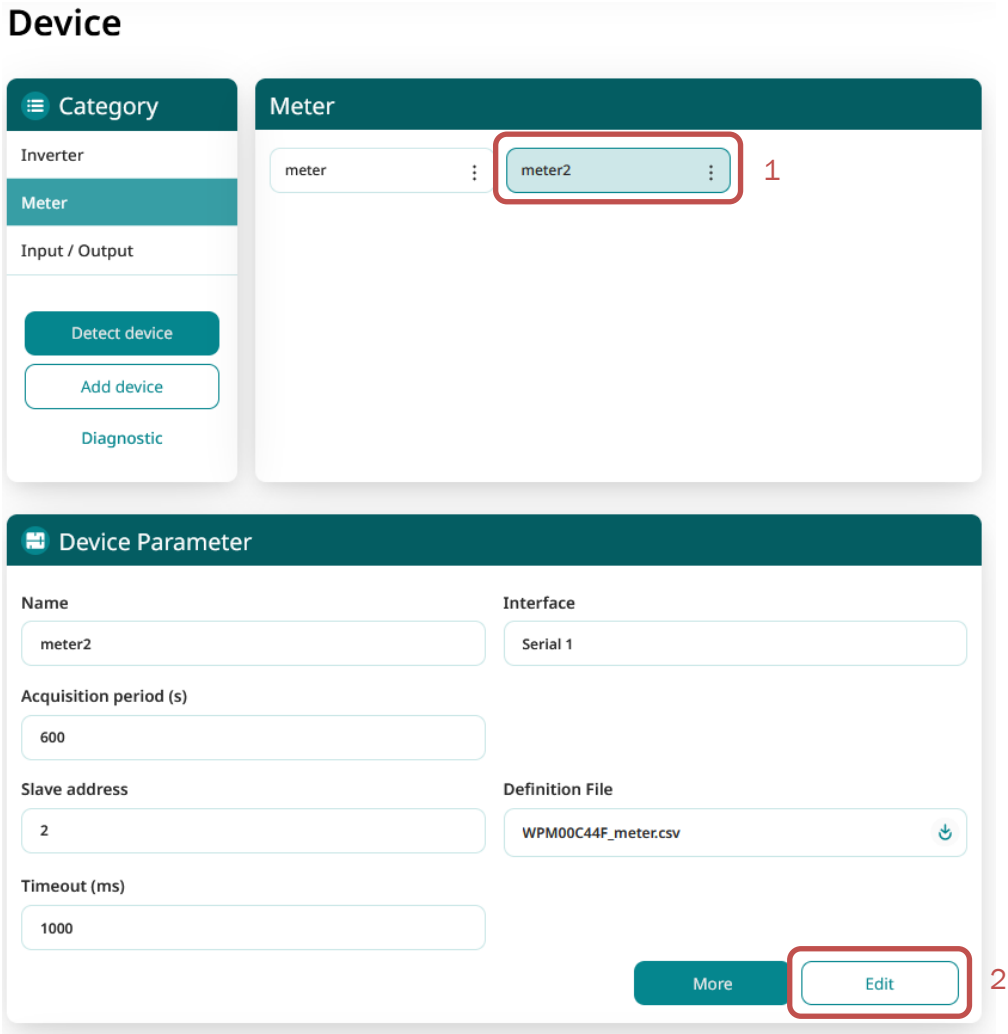
The device WebdynSunPM which defines input/output of the concentrator can't be deleted

3.2.3.2.2.3 Editing a device

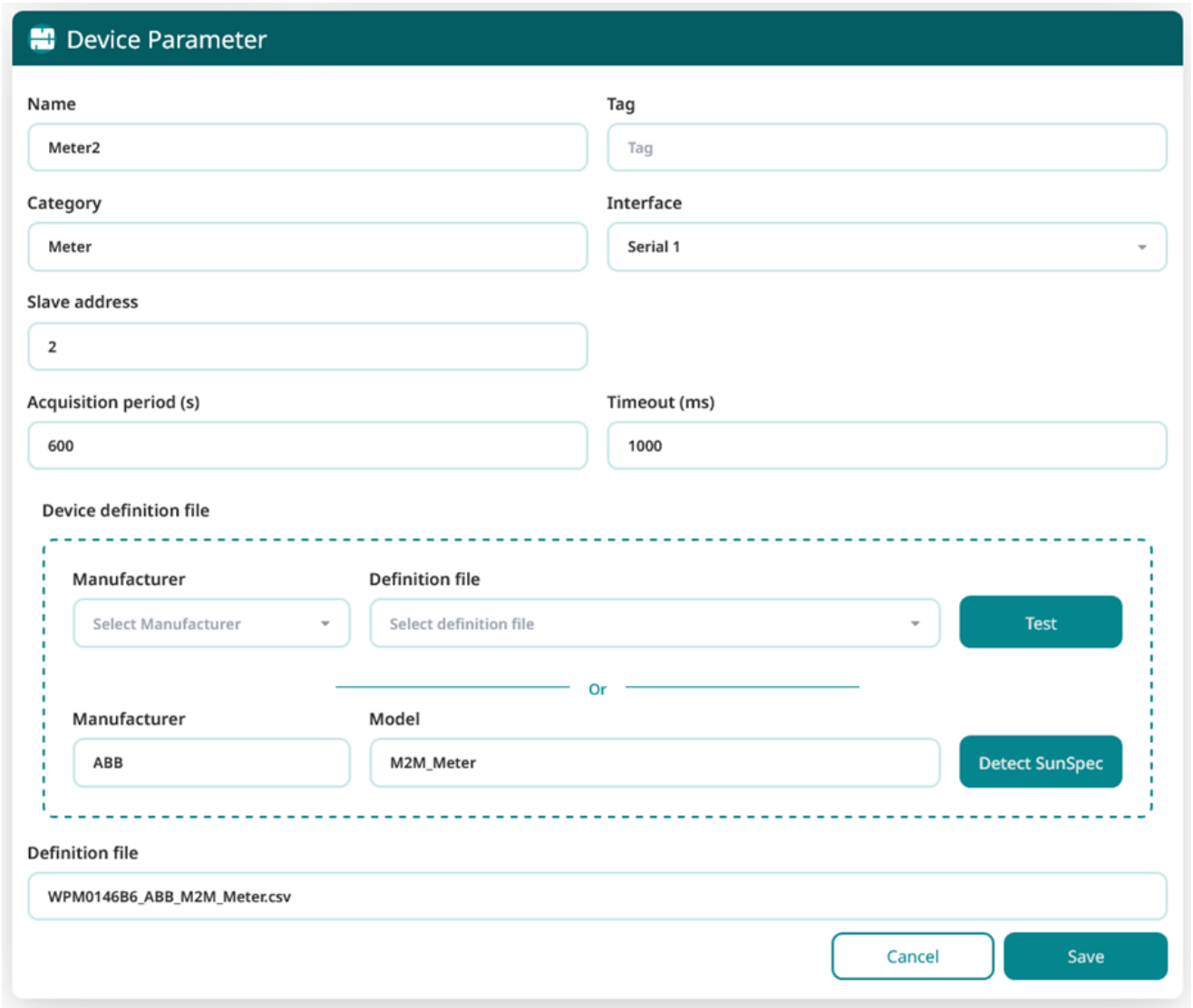
To edit a device, first select the device category to display the device to be edited:



Click the device to be edited.



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

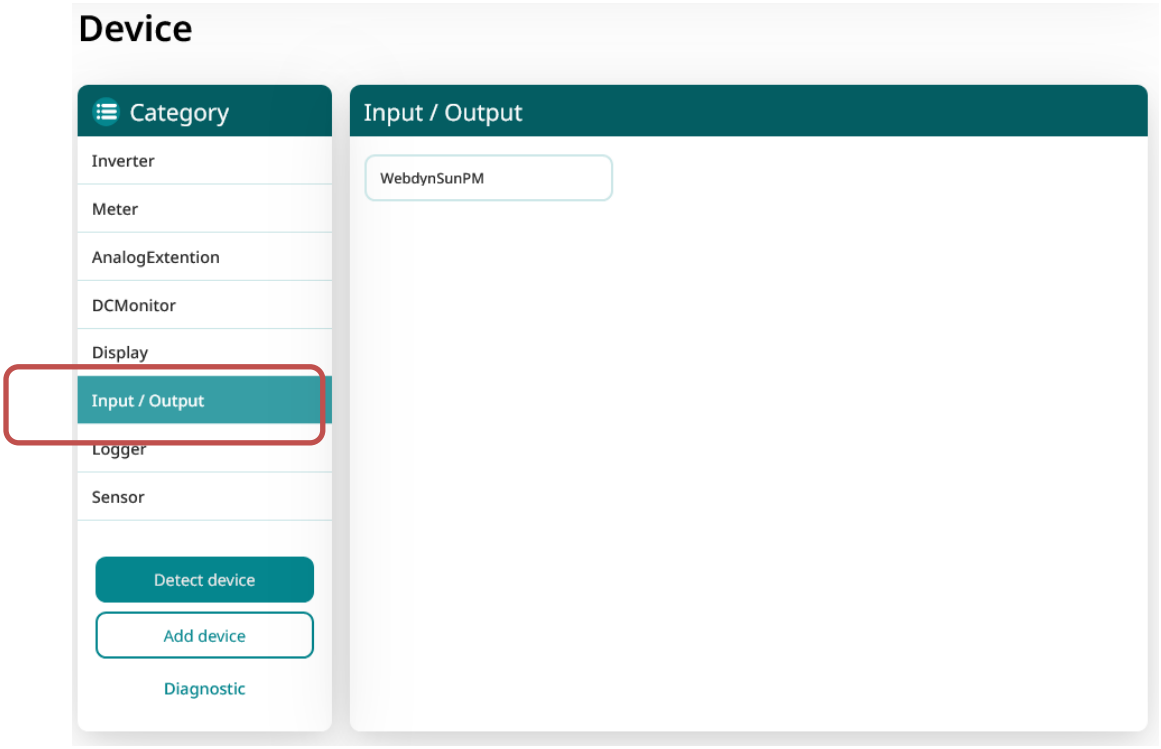


It is then possible to modify the different device fields and thus change the name, the interface, acquisition period, the model and the specific protocol parameters: IP address and port number for IP devices, slave address for Modbus devices, acquisition period, etc.

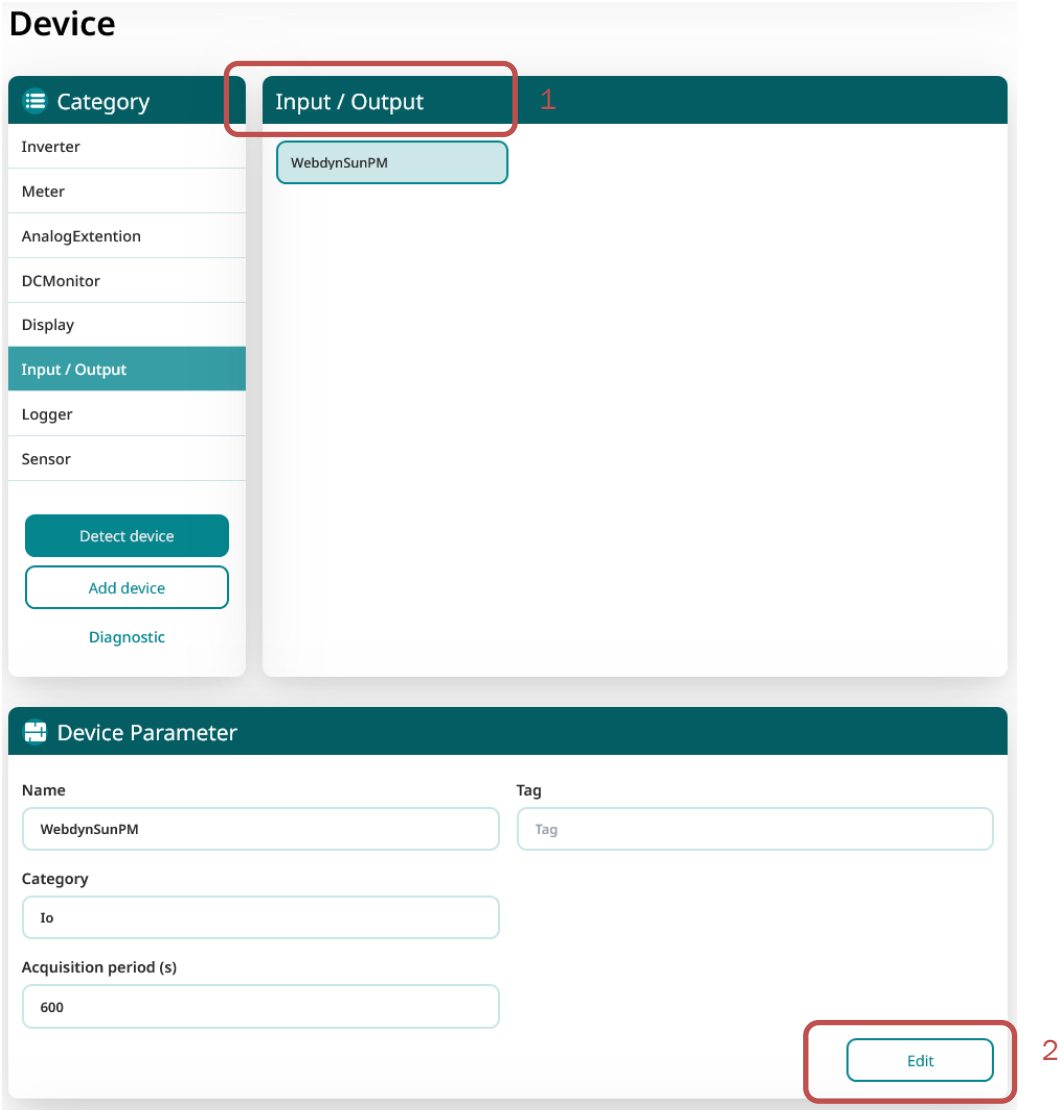
Pressing the “Save” button validates the new entry. Pressing the “Cancel” button ignores all changes made on the data entry form.

3.2.3.2.2.4 Edit WebdynSunPM I/O's

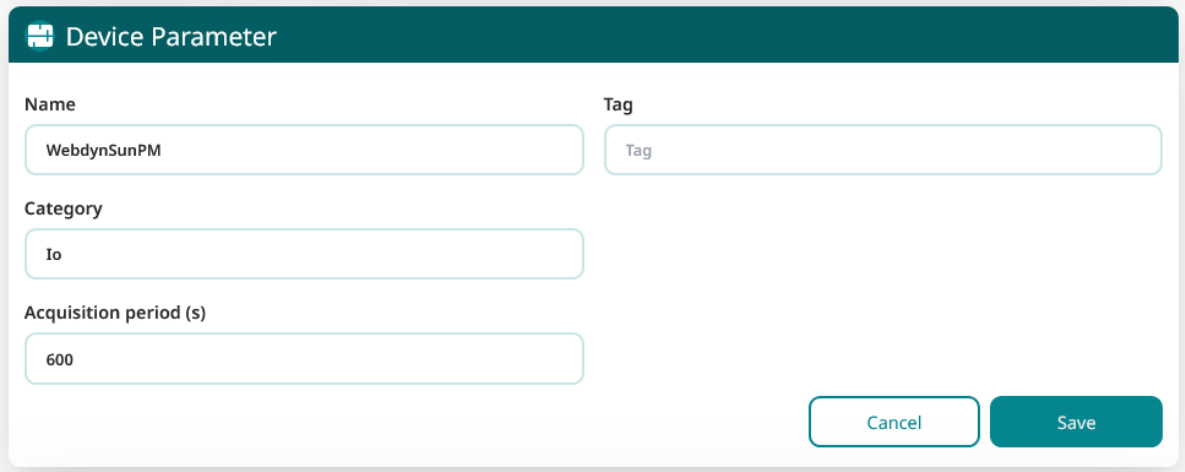
Select Input/Output category in device:



Click on the device to edit it.



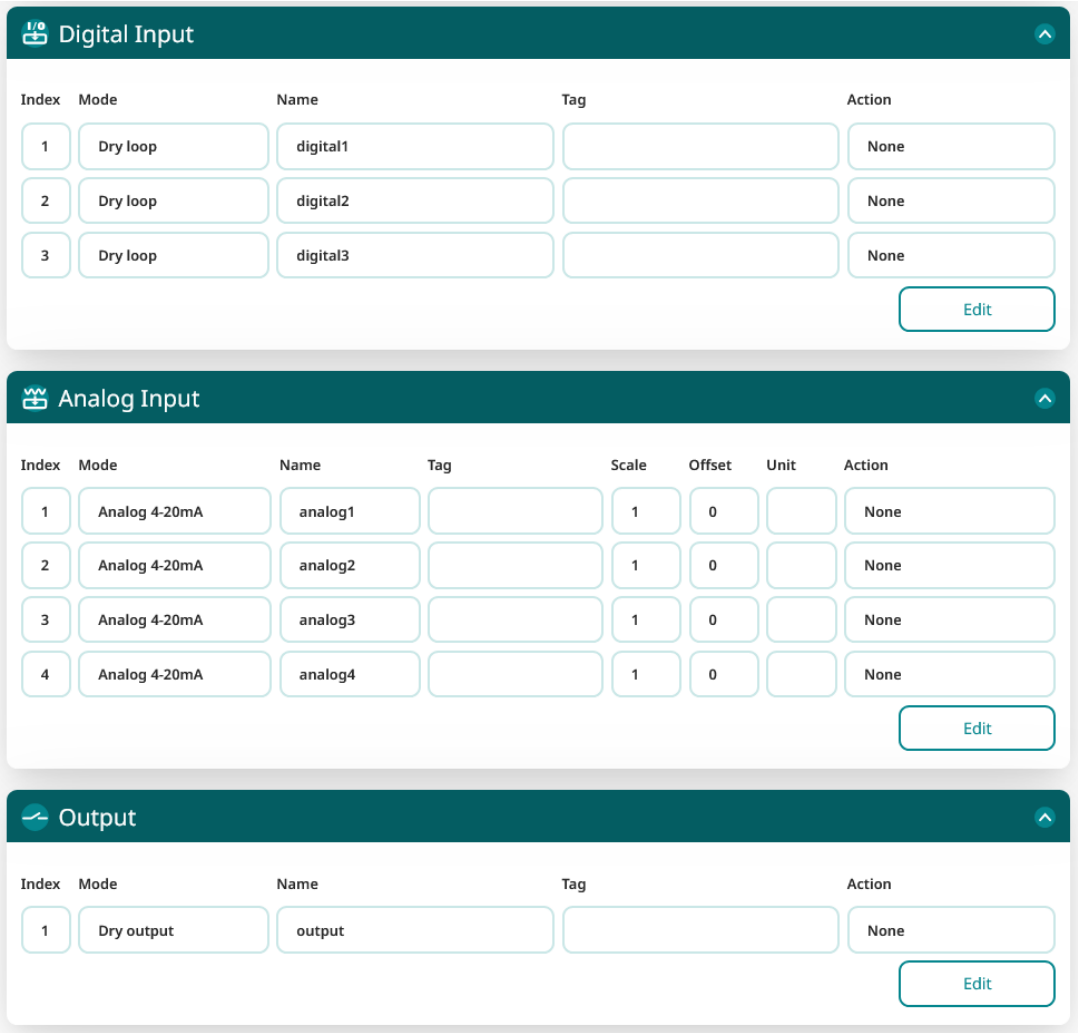
The fields can now be edited such as category and acquisition period. To finish editing, click on save



Name and tag can’t be modified

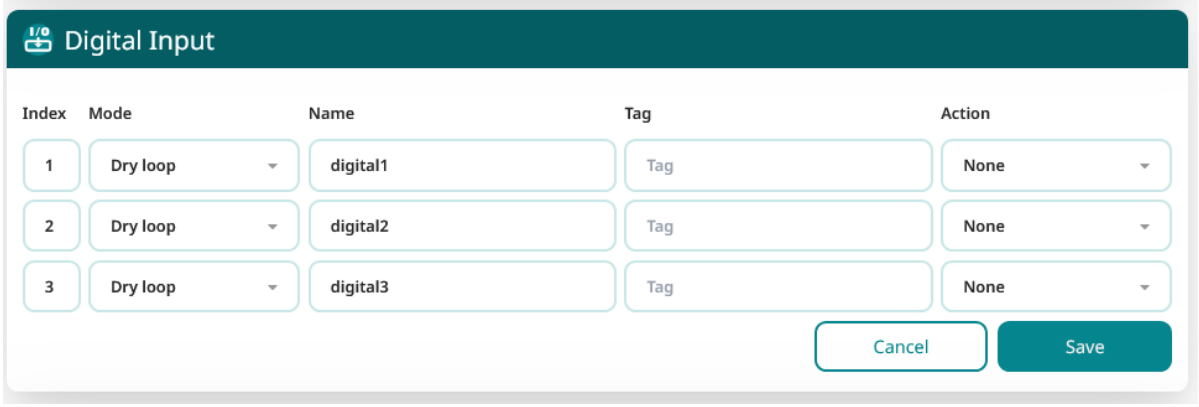
I/O's can be configured and are grouped as following :

- **Digit Input**
- **Analog Input**
- **Output**



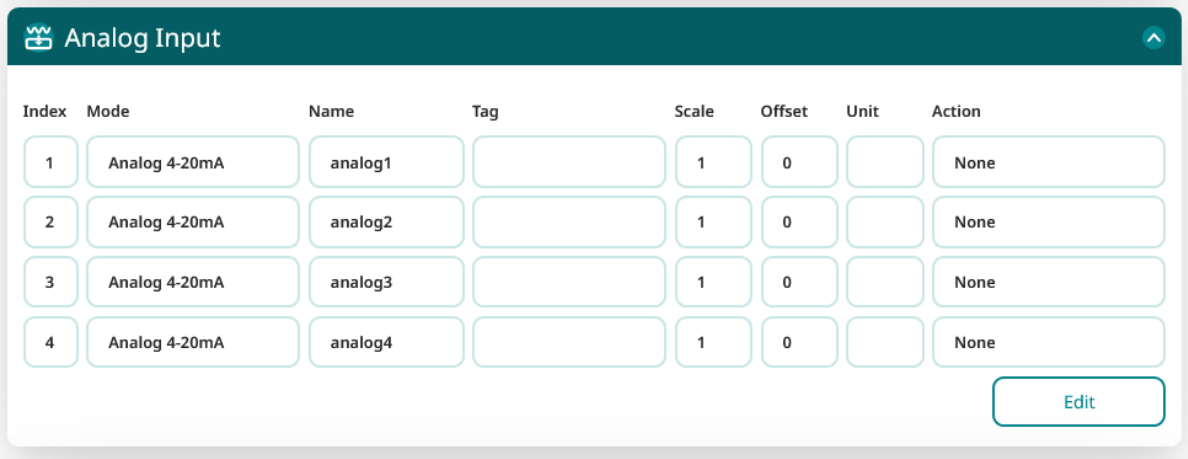
Digital Input:

The concentrator has 3 digital inputs. Click on edit to assign them.



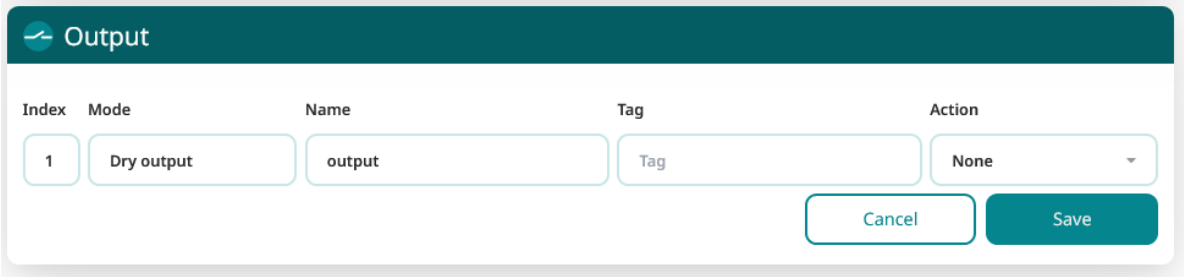
Analog input:

The concentrator has 4 analog inputs. Click on edit to assign them.



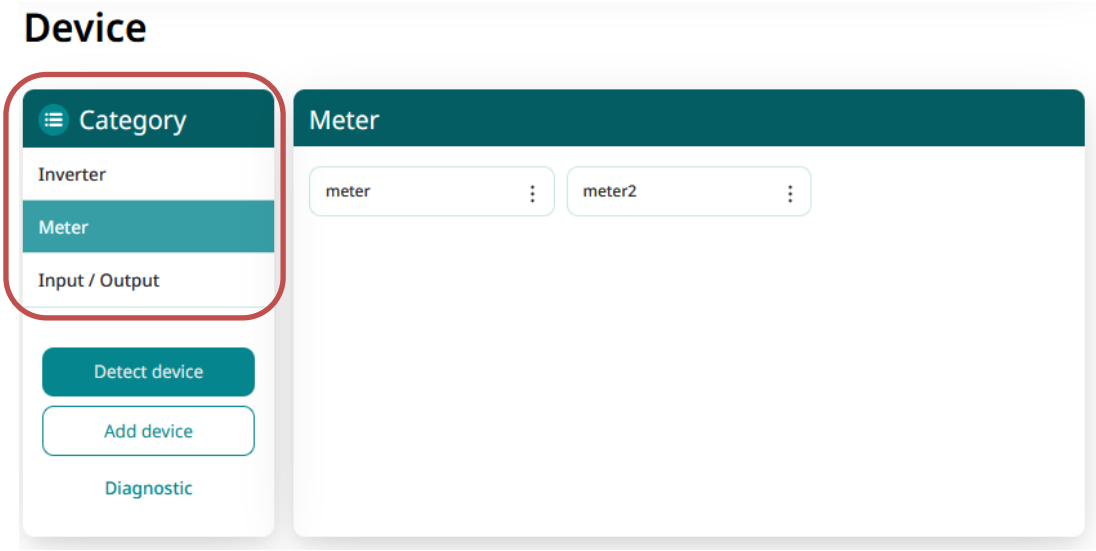
Output:

The concentrator has one relay. Click on edit to assign it.



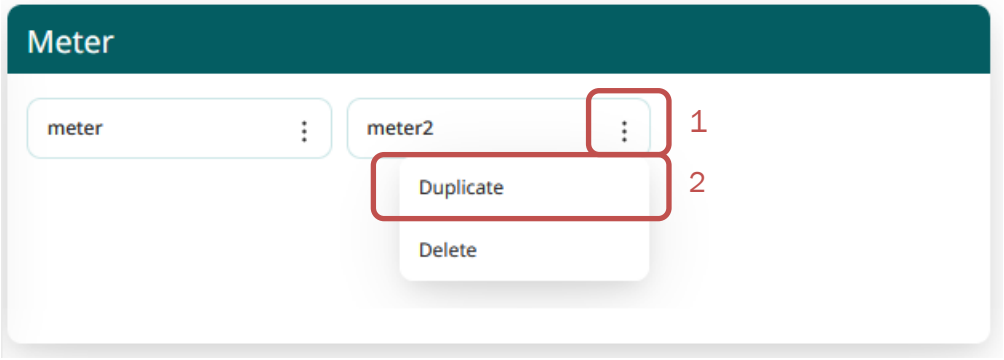
3.2.3.2.2.5 Duplicating a device

To duplicate a device, first select the device category to display the device to be duplicated:



There are 2 duplication methods.

Duplication from the device sub-menu:



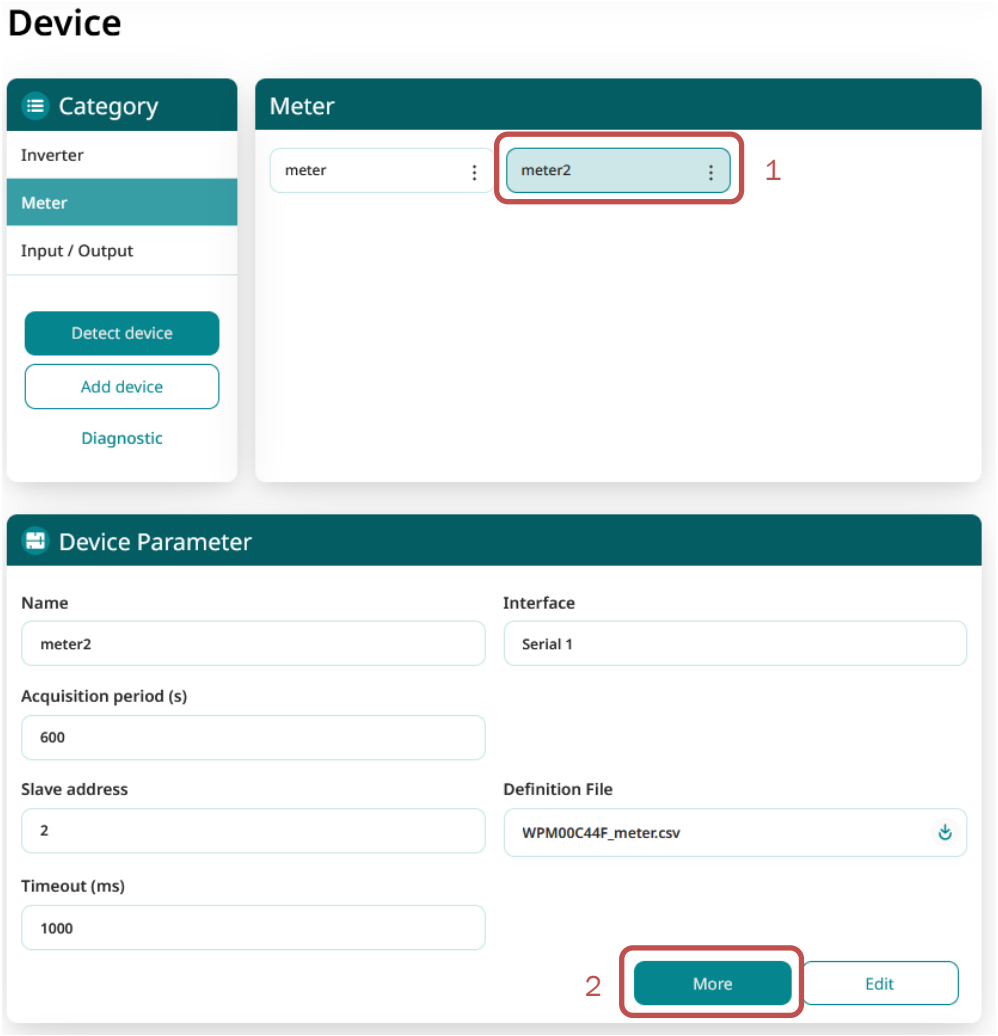
Click the sub-menu icon for the device to be duplicated, then select "*Duplicate*".

When the button is pressed, the device is duplicated. The new device has the same name as the original device, plus "(copy)".

The new instance uses the same definition file as the original device. It is then possible to rename and change the definition file by editing the device (*see section 3.2.3.2.2.3: "Editing a device"*).

Duplication from the device parameters:

Click the device to be duplicated.



Then click the " *More* " button in the device parameters, and then the duplication icon under the device description:

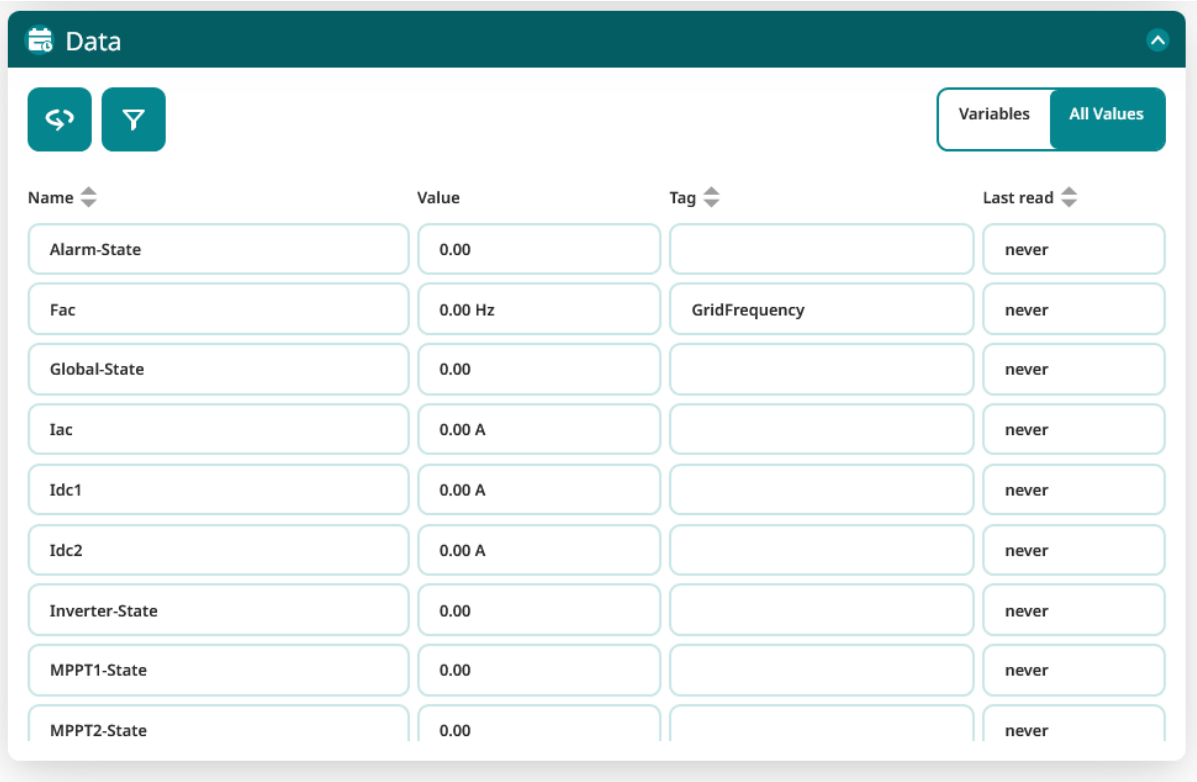
When the button is pressed, the device is duplicated. The new device has the same name as the original device, plus "(copy)".

The new instance uses the same definition file as the original device. It is then possible to rename and change the definition file by editing the device (*see section 3.2.3.2.2.3: "Editing a device"*).

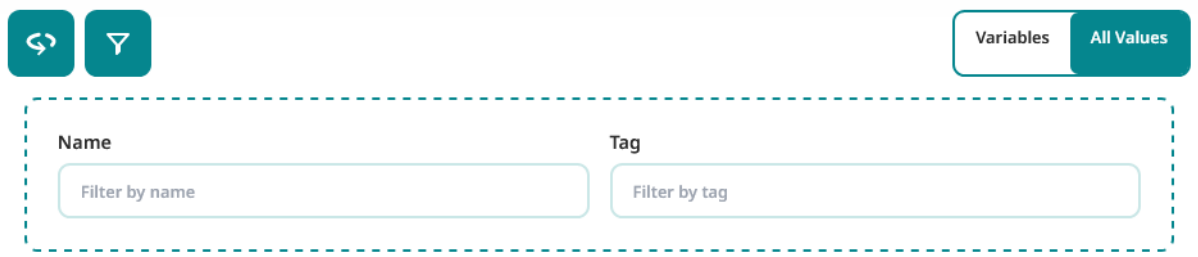
The device WebdynSunPM which defines input/output of the concentrator can't be deleted

3.2.3.2.2.6 Data

Under device settings, a « Data » tab can be used to see all parameters collected by the concentrator. For each parameters : name, value, tag and last read is present. Two buttons on the top left for refreshing or filtering.



By clicking on filter button, it is possible to search for name or tag. Another click on filter button cancels filter.



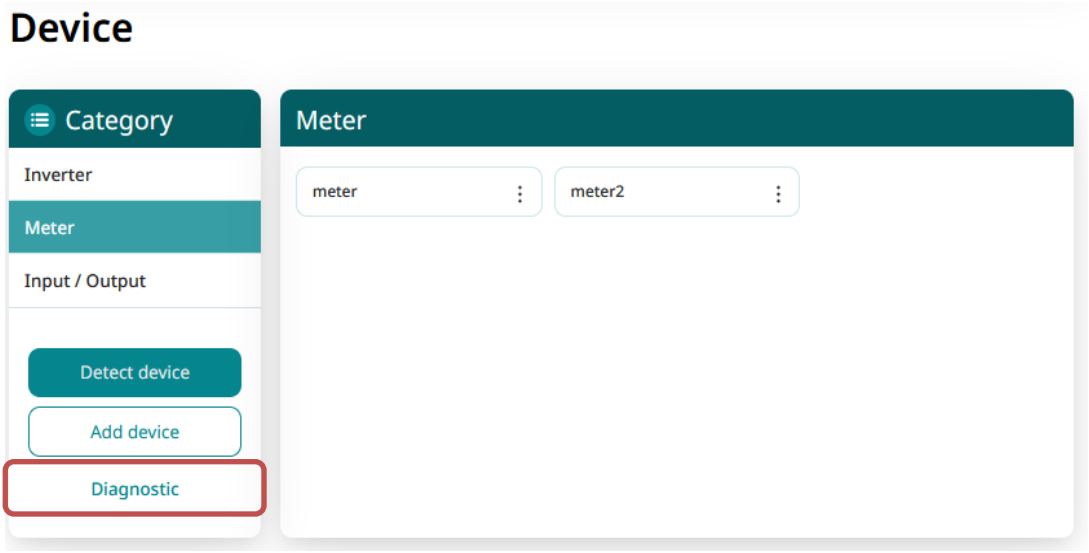
It is also possible to filter data with top right button “Variables” or “All values”.

- Variables: Only variables are displayed. Data set as parameters or deactivated (“non”) will not be shown.
- All Values: all data are shown including parameters and deactivated data.

3.2.3.2.3 Device troubleshooting tools

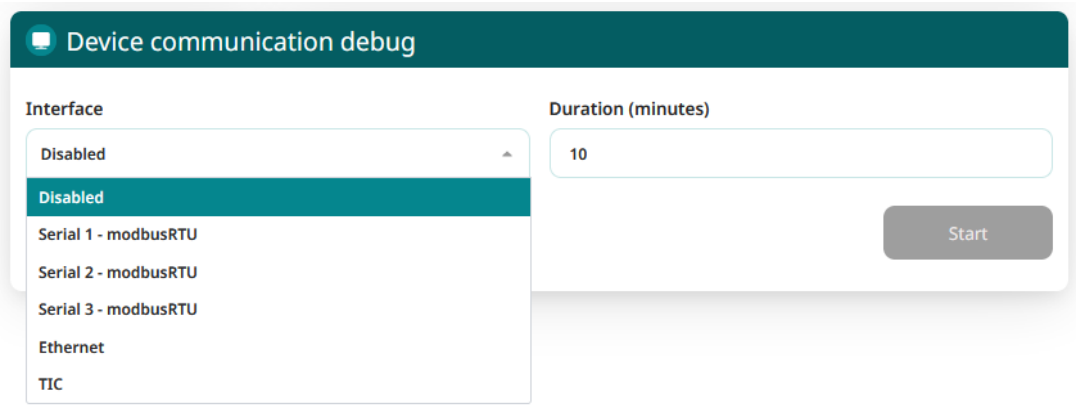
Troubleshooting tools are available to analyse the frames sent to devices as well as received frames. These tools help to understand what is happening when there are configuration problems.

3.2.3.2.3.1 Communication log activation



To access the device communications troubleshooting tool page, click the "*Diagnostic*" button on the device page.

The following page is displayed:



The first parameter is for device communications logs using the serial ports, Ethernet, TIC or IO.

When communication traces are selected for a given interface, all the communications on that interface will be logged and sent to the server in the form of a log time-stamped to the nearest millisecond.

Considering the amount of sent and received data, these logs are only activated for a set period, which can be configured in the "Duration" field. The period is expressed in minutes.

Once the time limit has expired, log activation automatically switches back to "Disabled".

3.2.3.2.3.2 Using logs

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

Serial and Ethernet interface logs:

The device communication, log file names are built using the same principle as the other gateway log files, i.e.: “uid”_”interface”_”date”.log.gz.

The interface corresponds to the one selected for the logs, i.e.:

- *Serial1*
- *Serial2*
- *Serial3*
- *Ethernet*

The logs contain the following information:

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

"datetime" is in *DD/MM/YYYY hh:mm:ss.mmm* format, with DD being the day of the month, MM the number of the month, YYYY the year, hh the time of the communication, mm the minute, ss the seconds, and mmm the milliseconds.
The "data" field contains the sent data and its meaning. For outgoing communications, the direction is "=>". For incoming communications, the direction is "<=".
If the communication interface is of the "serial" type, the data is then supplied in hexadecimal format.
If the communication interface is of the "ethernet" type, the IP address of the device to be monitored is logged, then the data is supplied in hexadecimal format in the same way as for the serial protocol.
Note that for modbus, the complete modbus TCP frame is supplied if the connection is of the Ethernet type. Otherwise it is the modbus RTU frame.

For modbus errors, the frame may be prefixed by 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 indicate a problem with the installation. The frame is ignored.
- *“*** BAD SLAVE ***”*: A slave with an incorrect number responded to the request. This may be due to a device configuration error that can disrupt the smooth running of the installation. The frame is ignored.
- "**** EXCEPTION ****": The slave has responded to the request with an exception. This means that the request that was sent is incorrect for the device in question.
- "**** INVALID ID ****": A slave has responded to a ModbusTCP request with an incorrect identifier. The frame was ignored.
- "**** INVALID FCT ****": The response contains an incorrect function code. The frame is ignored.

Input/output logs:

The input/output log file names are based on the same principle as the other gateway log files, i.e.: “uid”_IO_”date”.log.gz.

The logs contain the following information:

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

"datetime" is in *DD/MM/YYYY hh:mm:ss.mmm* format, with DD being the day of the month, MM the number of the month, YYYY the year, hh the time of the communication, mm the minute, ss the seconds, and mmm the milliseconds.
The "data" field contains the input/output data.
Firstly, the information type is shown:

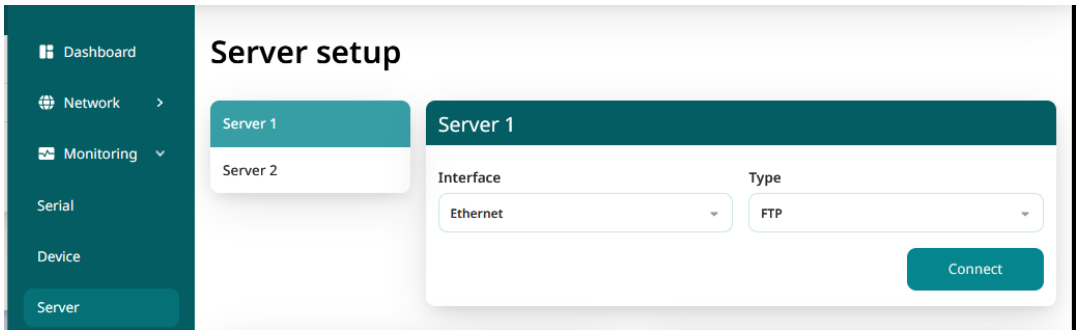
- "In": indicates an input
- "Out": indicates an output

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

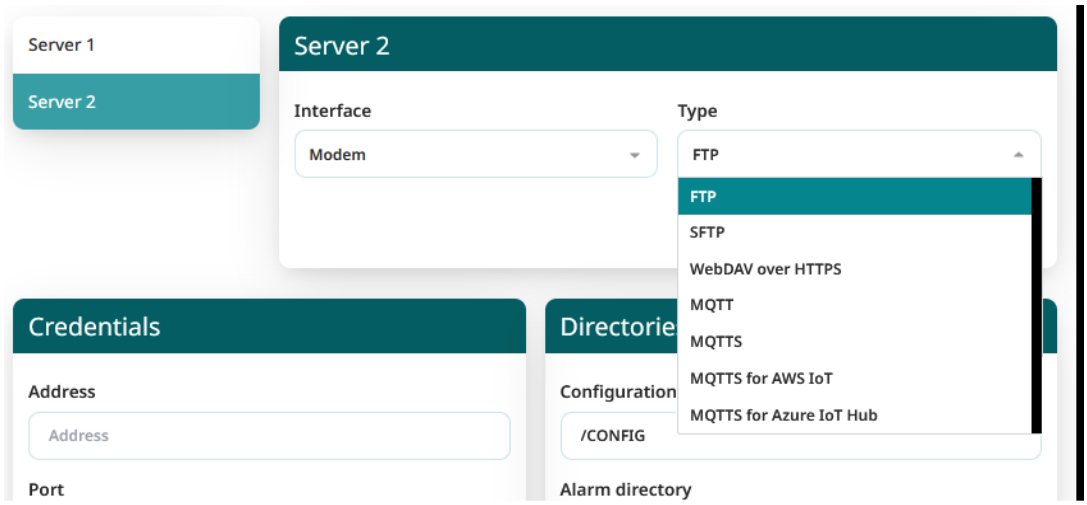
- 0: the input/output is closed
- 1: the input/output is open
- Any other value contains the value of an analogue input

3.2.3.3 Server

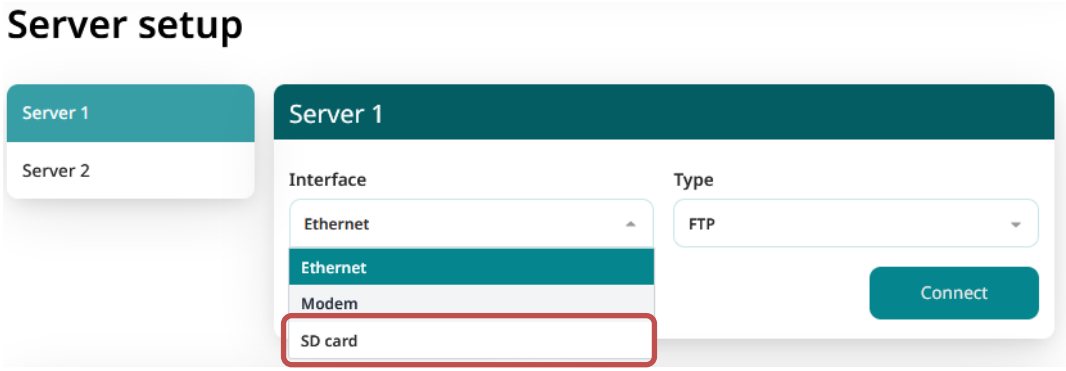
The "server" part is used to configure the 2 servers available on the concentrator and to schedule the synchronisation times. This synchronisation can also be carried out locally on an SD card.



The concentrator supports 7 different server types:



The concentrator can also store data locally on an SD card. To do that, select the "SD card" interface:

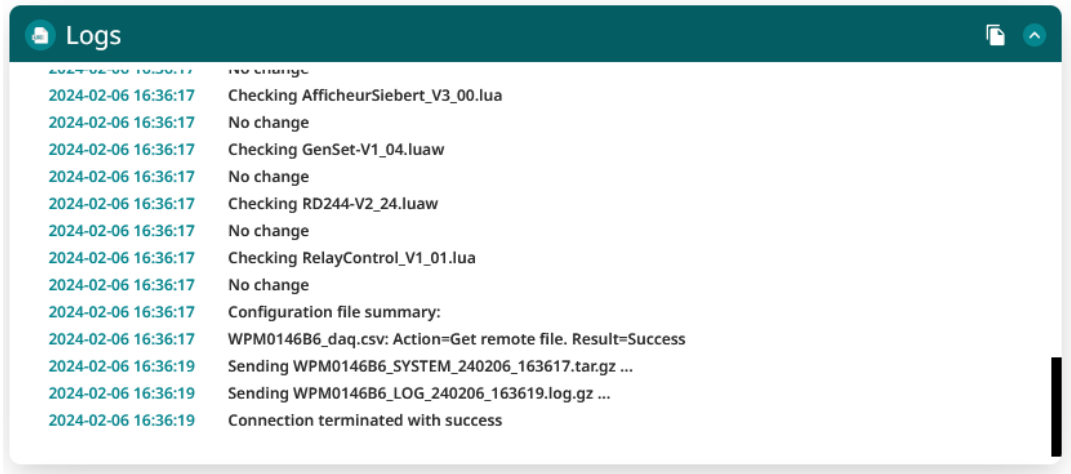


The choice of the server type or the SD Card interface modifies the parameters to be entered. The server parameters remain unchanged.

Server 2 is only used to backup the configurations. Only server 1 manages the configuration file synchronisation.
The MQTT/MQTTS/MQTTS AWS IOT/MQTTS Azure IoT Hub is only available on server 2 (backup).

To check that the server configuration is correct, clicking the "*Connect*" button is recommended. A window showing the connection log is displayed showing all the files exchanged between the concentrator and the remote server. Looking at the last line is a quick way to see if the connection was successful or not.

Ending an ongoing connection using the "*Cancel connect*" button is not instant. Indeed, if an action is in progress, it must complete first. Every line in the connection log represents an action.

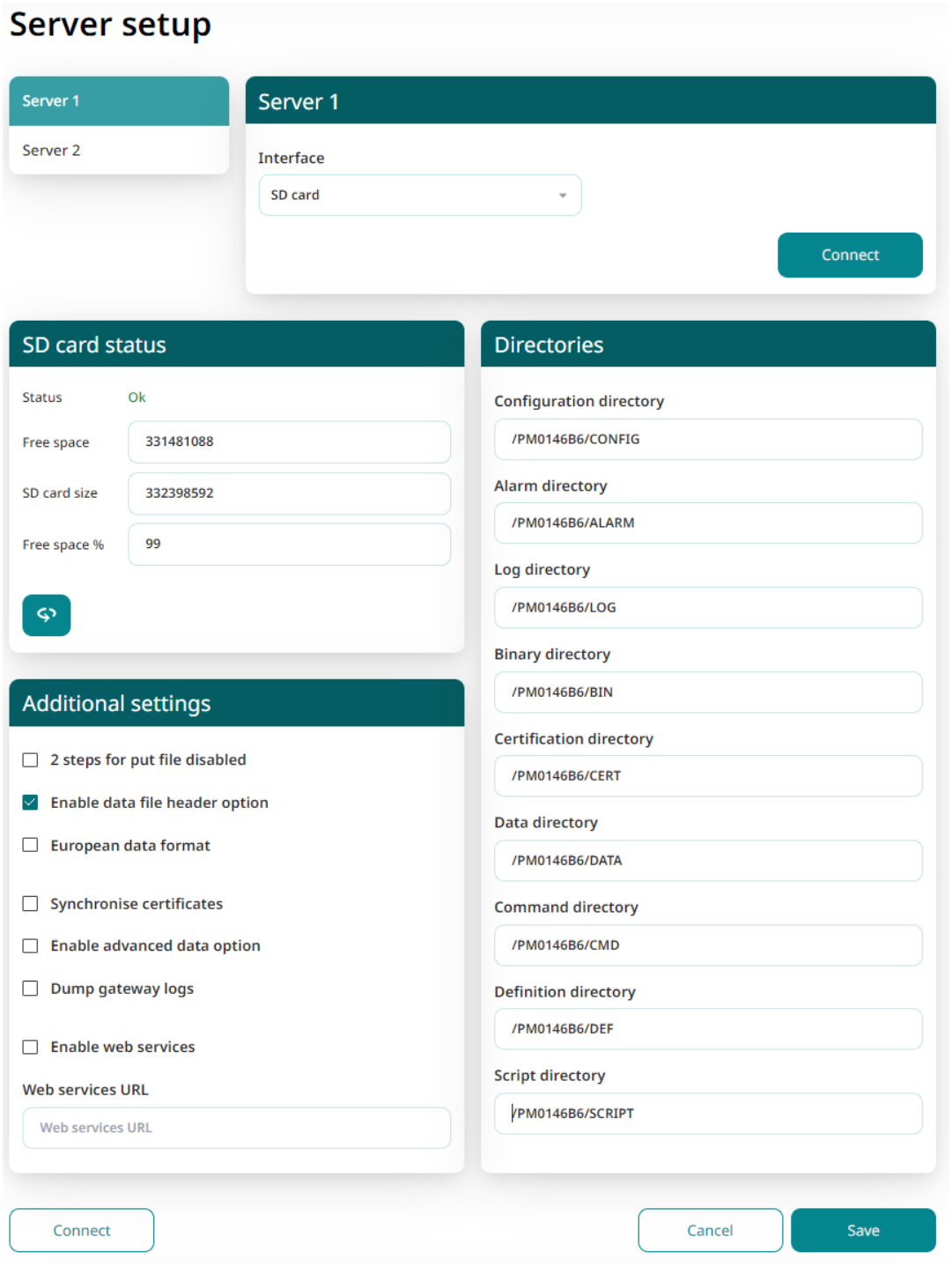


If there are errors, check all the server settings.

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

3.2.3.3.1 SD Card

When the "SD card" interface is selected, the display removes the unnecessary fields and a box appears with the current information about the SD card:



La carte SD est exclusivement réservée à l'usage du serveur 1.

The parameters on server 1:

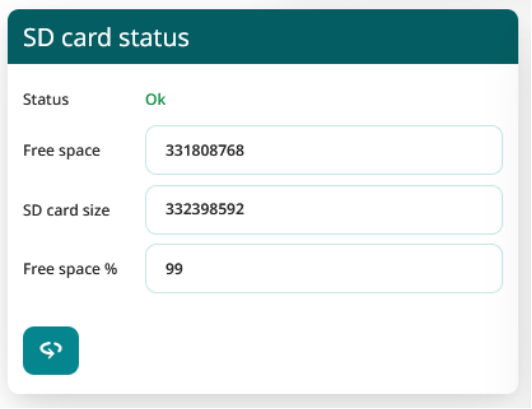
Web interface	Parameter in <uid>_config.ini	Description
Interface	SERVER_Interface	Selection of the network interface to use for the server: • **sdcard**

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*: SD card status, the possible statuses are:
  - *Unknown*: the status is unknown because it has not been accessed yet
  - *Ok*: the last attempt to write to the SD card was successful
  - *Failed to mount SD Card*: the SD card was not detected. Check that the card is correctly inserted and formatted (FAT32 or exFAT)
  - *SD Card read error*: the SD card was detected but there were errors reading the information on it. Check that the card is correctly formatted (FAT32 or exFAT)
- *Free space*: memory size available on the SD card (in bytes)
- *SD card size*: SD card capacity (in bytes)
- *Free space (%)*: SD card remaining space percentage. Used to quickly and easily monitor how full the SD card is.

At each connection, the concentrator will save data to and retrieve data from the SD card. The "SD card status" box is used to monitor how much space is occupied by the data stored on the card:

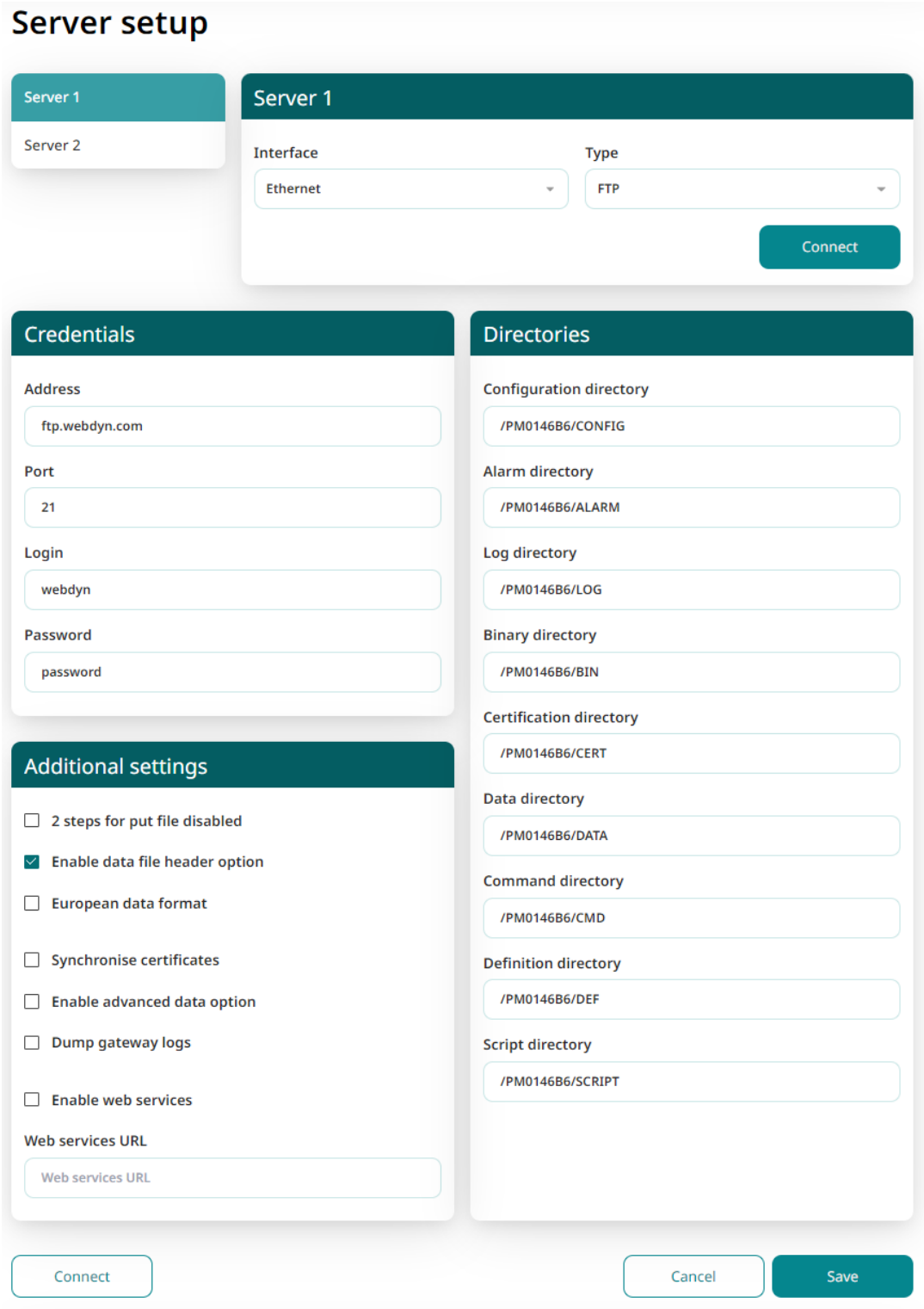


It is of course possible to force a write to the SD card by clicking the 'Connect' button on the page.

It is possible to load a new configuration to the concentrator by saving the configuration files directly to the SD card before inserting it into the concentrator.

3.2.3.3.2 FTP/SFTP

FTP (unsecure) and SFTP servers have the same parameters.



The parameters on the 2 servers are:

Server 1 or Server 2:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Interface	SERVER_Interface SERVER2_Interface	Selection of the network interface to use for the server: • **Ethernet** (*see section 3.2.2.1: "Ethernet (Local)"*) • **Modem** (*see section 3.2.2.2: "Modem (Mobile)"*)
Type	SERVER_Type SERVER2_Type	Server protocol selection: • **FTP**: FTP server (unsecure) • **SFTP**: SFTP server

Credentials:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Address	SERVER_Address SERVER2_Address	IP address or server name
Port	FTP_Port FTP2_Port	FTP/SFTP server port
Login	FTP_Login FTP2_Login	User name used by the concentrator for the connection to the remote FTP/SFTP server
Password	FTP_Password FTP2_Password	Password used by the concentrator for the connection to the remote FTP/SFTP server

Directories:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Configuration directory	FTP_DirConfig FTP2_DirConfig	Configuration file directory on the FTP/SFTP server.
Alarm directory	FTP_DirAlarm FTP2_DirAlarm	Alarm file directory on the FTP/SFTP server.
Log directory	FTP_DirLog FTP2_DirLog	Log file directory on the FTP/SFTP server.
Binary directory	FTP_DirBin FTP2_DirBin	Update file directory on the FTP/SFTP server.
Certificate directory	FTP_DirCert FTP2_DirCert	Certificate file directory on the FTP/SFTP server.
Data directory	FTP_DirData FTP2_DirData	Data file directory on the FTP/SFTP server.
Command directory	FTP_DirCmd FTP2_DirCmd	Command file directory on the FTP/SFTP server.

Definition directory	FTP_DirDef FTP2_DirDef	Definition file directory on the FTP/SFTP server.
Script directory	FTP_DirScript FTP2_DirScript	Script file directory on the FTP/SFTP server.

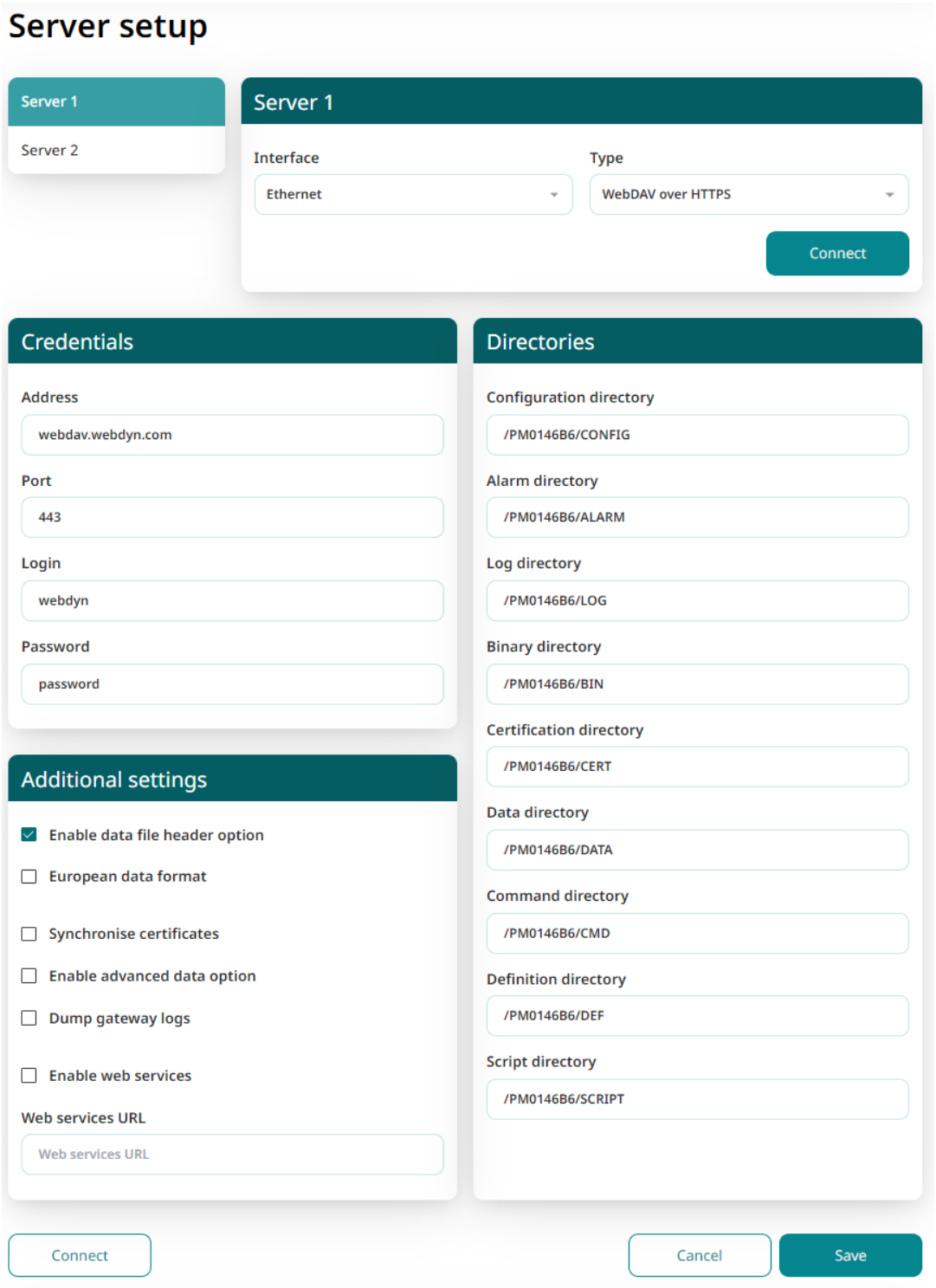
Additional settings:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
2 steps for put file disabled	FTP_TwoStepsSendingDisabled FTP2_TwoStepsSendingDisabled	Choice of file transfers in 2 steps using a temporary file while the file is not complete on the remote server: • **Checked:** disabled • **Unchecked**: enabled
Enable data file header option	FTP_HeaderOption FTP2_HeaderOption	Optional header selection in the data files uploaded to the FTP/SFTP server: • **Checked:** With optional headers • **Unchecked:** Without optional headers
European date format	FTP_EuroDateFormat FTP2_EuroDateFormat	Timestamp type for data uploaded on the FTP/SFTP server: • **Checked:** European format (DD/MM/YY-HH:MM:SS) • Unchecked: ISO format (YY/MM/DD-HH:MM:SS)
Synchronise certificates	FTP_SynchroniseCertificates FTP2_SynchroniseCertificates	Certificate synchronisation selection: • **Checked:** Enables certificate synchronisation • **Unchecked:** No certificate synchronisation
Enable advanced data option	FTP_EnableAdvancedData FTP2_EnableAdvancedData	Addition of the number of complete reads during this acquisition period to the data files placed on the FTP/SFTP server: • **Checked:** Addition of the number of complete reads • **Unchecked:** Number of complete reads not added
Dump gateway logs	FTP_UploadLog FTP2_UploadLog	System log file upload selection on the FTP/SFTP server: • **Checked:** System log files uploaded on a schedule. • **Unchecked:** No system log files uploaded on a schedule. The system log files are systematically uploaded on manual action.
Enable Web Services	FTP_WebServicesEnable FTP2_WebServicesEnable	Activation of the web services associated with the FTP actions: • **Checked**: the web services are enabled • **Unchecked**: the web services are disabled
Web Services URL	FTP_WebServicesUrl FTP2_WebServicesUrl	URL to call when the FTP actions have been completed and the web services are enabled.

The directory tree structure on the remote FTP/SFTP server must be created before any connections. *(see section 4.1: "The FTP/SFTP/WebDAV server"")*

3.2.3.3.3 WebDAV over HTTPS

The WebDAV over HTTPS server is a secure server with an identification using a login and password.



The parameters on the 2 servers are:

Server 1 or Server 2:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Interface	SERVER_Interface SERVER2_Interface	Selection of the network interface to use for the server: • **Ethernet** (*see section 3.2.2.1: "Ethernet (Local)"*) • **Modem** (*see section 3.2.2.2: "Modem (Mobile)"*)
Type	SERVER_Type SERVER2_Type	Server protocol selection: • **WebDAV over HTTPS**

Credentials:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Address	SERVER_Address SERVER2_Address	IP address or server name
Port	HTTP_Port HTTP2_Port	WebDAV server port
Login	HTTP_Login HTTP2_Login	The login used by the concentrator to connect to the remote WebDAV server
Password	HTTP_Password HTTP2_Password	The password used by the concentrator to connect to the remote WebDAV server

Directories:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Configuration directory	HTTP_DirConfig HTTP2_DirConfig	Configuration file directory on the WebDAV server
Alarm directory	HTTP_DirAlarm HTTP2_DirAlarm	Alarm file directory on the WebDAV server
Log directory	HTTP_DirLog HTTP2_DirLog	Log file directory on the WebDAV server
Binary directory	HTTP_DirBin HTTP2_DirBin	Update file directory on the WebDAV server
Certificate directory	HTTP_DirCert HTTP2_DirCert	Certificate file directory on the WebDAV server
Data directory	HTTP_DirData HTTP2_DirData	Data file directory on the WebDAV server

Command directory	HTTP_DirCmd HTTP2_DirCmd	Command file directory on the WebDAV server
Definition directory	HTTP_DirDef HTTP2_DirDef	Definition file directory on the WebDAV server
Script directory	HTTP_DirScript HTTP2_DirScript	Script file directory on the WebDAV server

Additional settings:

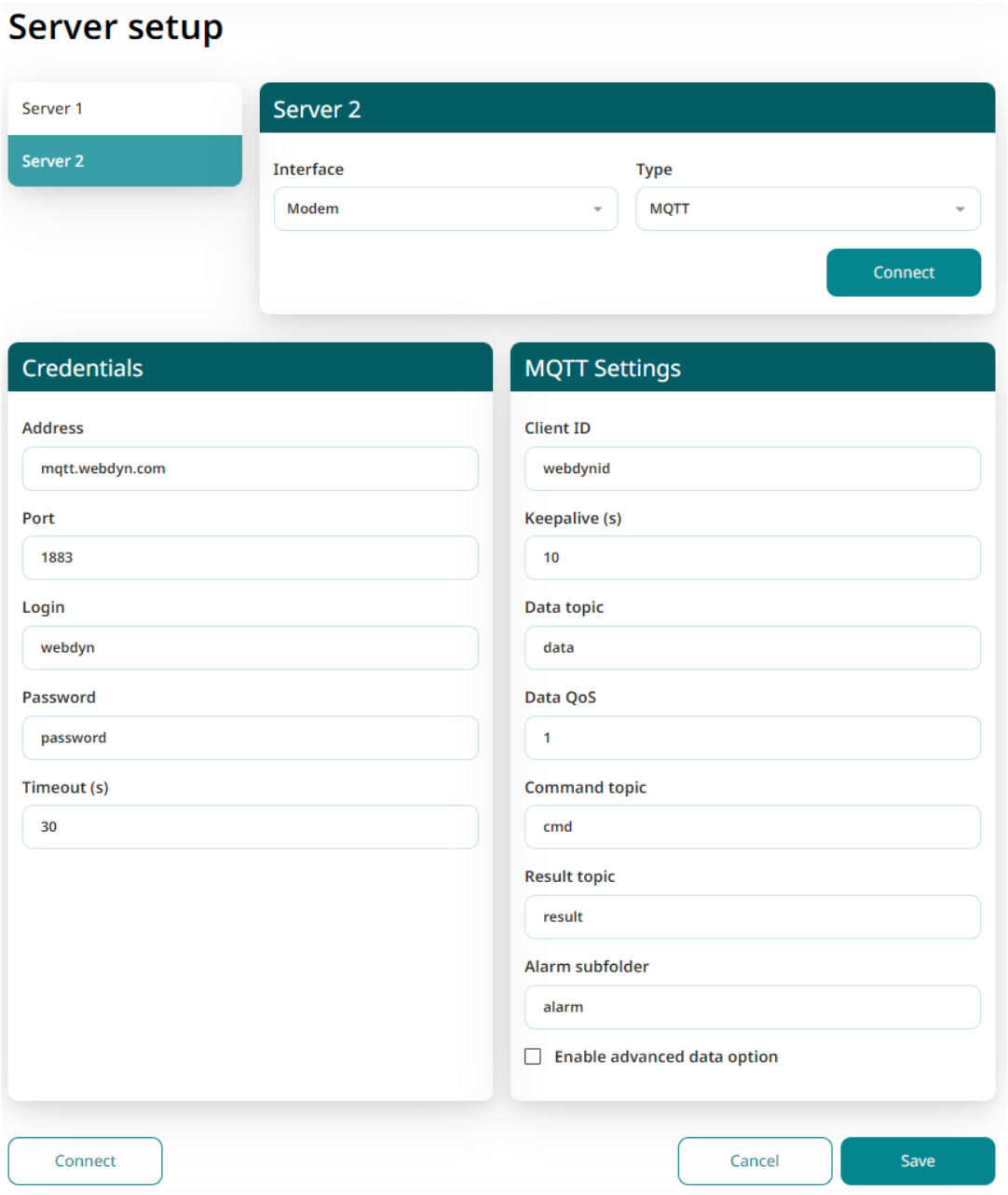
Web interface	Parameter in the <uid>_config.ini configuration file	Description
2 steps for put file disabled	HTTP_TwoStepsSendingDisabled HTTP2_TwoStepsSendingDisabled	Choice of file transfers in 2 steps using a temporary file while the file is not complete on the remote server: • **Checked:** disabled • **Unchecked**: enabled
Enable data file header option	HTTP_HeaderOption HTTP2_HeaderOption	Optional header selection in the data files uploaded to the WebDAV server: • **Checked:** With optional headers • **Unchecked:** Without optional headers
European date format	HTTP_EuroDateFormat HTTP2_EuroDateFormat	Timestamp type selection for the data uploaded to the WebDAV server: • **Checked:** European format (DD/MM/YY-HH:MM:SS) • Unchecked: ISO format (YY/MM/DD-HH:MM:SS)
Synchronise certificates	HTTP_SynchroniseCertificates HTTP2_SynchroniseCertificates	Certificate synchronisation selection: • **Checked:** Enables certificate synchronisation • **Unchecked:** No certificate synchronisation
Enable advanced data option	HTTP_EnableAdvancedData HTTP2_EnableAdvancedData	Addition of the number of complete reads during this acquisition period in the data files uploaded to the WebDAV server: • **Checked:** Addition of the number of complete reads • **Unchecked:** Number of complete reads not added
Dump gateway logs	HTTP_UploadLog HTTP2_UploadLog	System log file upload selection to the WebDAV server: • **Checked:** System log files uploaded on a schedule. • **Unchecked:** No system log files uploaded on a schedule. The system log files are systematically uploaded on manual action.
Enable Web Services	HTTP_WebServicesEnable HTTP2_WebServicesEnable	Activation of the web services associated with the WebDAV actions: • **Checked**: the web services are enabled • **Unchecked**: the web services are disabled

Web Services URL	HTTP _WebServicesUrl HTTP 2_WebServicesUrl	URL to call when the WebDAV actions have been completed and the web services are enabled.

The directory tree structure on the remote WebDAV-HTTPS server must be created before any connections. *(see section 4.1; "The FTP/SFTP/WebDAV server")*

3.2.3.3.4 MQTT

The MQTT server is an unsecured server with a login and a password.



The server 2 parameters are:

Server 2:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Interface	SERVER2_Interface	Selection of the network interface to use for the server: • **Ethernet** (*see section 3.2.2.1: "Ethernet (Local)"*) • **Modem** (*see section 3.2.2.2: "Modem (Mobile)"*)
Type	SERVER2_Type	Server protocol selection: • **MQTT**: MQTT server

Credentials:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Address	SERVER2_Address	IP address or server name
Port	MQTT2_Port	MQTT server port (the default is 1883)
Login	MQTT2_Login	The login used by the concentrator to connect to the MQTT server
Password	MQTT2_Password	The password used by the concentrator to connect to the MQTT server
Timeout (s)	MQTT2_Timeout	Maximum waiting time in seconds for the MQTT server response. If the server has not responded within the allotted time, the transmission is stopped and reattempted during the next schedule. Only works with QoS 1 or QoS 2.

MQTT Settings:

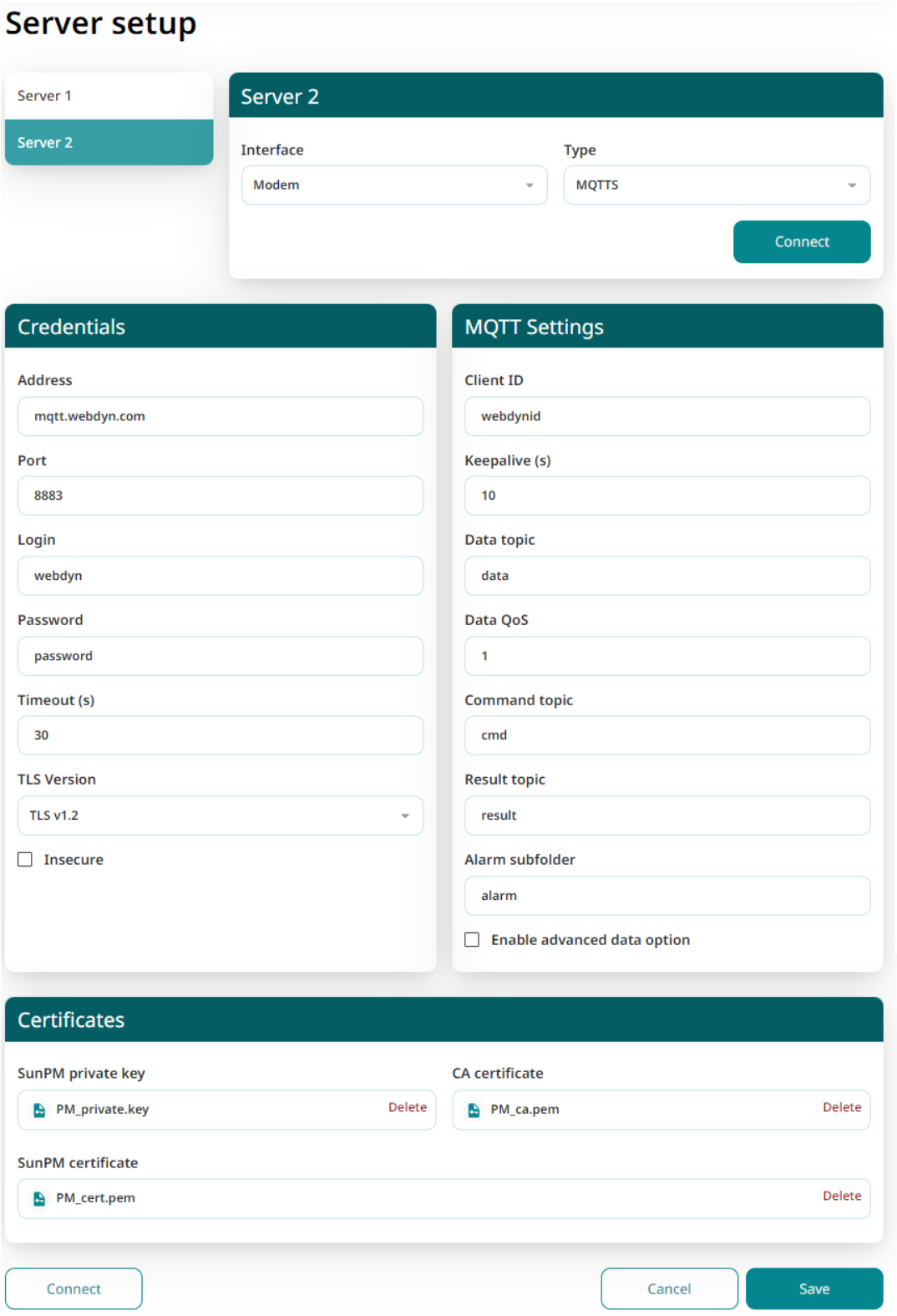
Web interface	Parameter in <uid>_config.ini	Description
Client ID	MQTT2_ClientId	Customisable device identifier on the MQTT server. This parameter is retrieved from your MQTT server.
Keepalive (s)	MQTT2_KeepAlive	If there is no exchange with the MQTT server for the time defined in seconds, the concentrator sends a ping to the MQTT server to check the connection. If the value is "0", KeepAlive is disabled. If the concentrator is in permanent connection mode with the MQTT server and a disconnection is detected after a KeepAlive, the concentrator will automatically reconnect to the MQTT server.
Data topic	MQTT2_Topic	Topic name for the data uploaded by the concentrator.

Data qos	MQTT2_QoS	Guaranteed service number for message sending (Quality Of Service). The possible values are: • 0: The message will only be sent once, i.e. with no guaranteed 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 be always be saved by the concentrator and will continue to be sent as long as the broker does not confirm it has been sent. (avoids message duplication)
Command topic	MQTT2_ControlTopic	Name of the topic for commands to be retrieved by the concentrator. The MQTT2_ResultTopic parameter must be set to use the commands. If a topic name is entered, the concentrator stays in permanent connection mode with the MQTT server.
Result topic	MQTT2_ResultTopic	Name of the topic for the results of commands sent to the concentrator. The MQTT2_ControlTopic parameter must be set to use the commands. If a topic name is entered, the concentrator stays in permanent connection mode with the MQTT server.
Alarm topic	MQTT2_AlarmTopic	Name of the alarm topic to be published. If the field is empty, no alarms will be published to the broker. If a topic name is entered, the concentrator stays in permanent connection mode with the MQTT server.
Enable advanced data option	MQTT2_EnableAdvancedData	Publication of the number of complete reads over this acquisition period in the data topic. The possible values are: • 0: Deactivated • 1: Enabled

MQTT is only available on server 2 (backup).
If the command topic is filled in, then the connection to the MQTT server will be permanent to be able to receive the sent commands.

3.2.3.3.5 MQTTS

The MQTTS server is a secure server with a login and password. Certificates and a private key must be imported to secure the connection between the concentrator and the MQTTS server.



The server 2 parameters are:

Server 2:

Web interface	Parameter in <uid>_config.ini	Description
Interface	SERVER2_Interface	Selection of the network interface to use for the server:

		• **Ethernet** (*see section 3.2.2.1: "Ethernet (Local)"*) • **Modem** (*see section 3.2.2.2: "Modem (Mobile)"*)
Type	SERVER2_Type	Server protocol selection: • **MQTTS**: secure MQTT server

Credentials:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Address	SERVER2_Address	IP address or server name
Port	MQTT2_Port	MQTTS server port (the default is 8883)
Login	MQTT2_Login	The login used by the concentrator to connect to the MQTTS server
Password	MQTT2_Password	The password used by the concentrator to connect to the MQTTS server
Timeout (s)	MQTT2_Timeout	Maximum waiting time in seconds for the MQTTS server response. If the server has not responded within the allotted time, the transmission is stopped and reattempted during the next schedule. Only works with QoS 1 or QoS 2.
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 the certificates. The possible values are: • Unchecked: Verification enabled • Checked: Verification disabled.

MQTT Settings:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Client ID	MQTT2_ClientId	Customisable device identifier on the MQTTS server. This parameter is retrieved from your MQTTS server.
Keepalive (s)	MQTT2_KeepAlive	If there is no exchange with the MQTTS server for the time defined in seconds, the concentrator sends a ping to the MQTTS server to check the connection. If the value is "0", KeepAlive is disabled. If the concentrator is in permanent connection mode with the MQTTS server and a disconnection is detected after a

		KeepAlive, the concentrator will automatically reconnect to the MQTTS server.
Data topic	MQTT2_Topic	Topic name for the data uploaded by the concentrator.
Data qos	MQTT2_QoS	Guaranteed service number for message sending (Quality Of Service). The possible values are: • 0: The message will only be sent once, i.e. with no guaranteed 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 be always be saved by the concentrator and will continue to be sent as long as the broker does not confirm it has been sent. (avoids message duplication)
Command topic	MQTT2_ControlTopic	Name of the topic for commands to be retrieved by the concentrator. The MQTT2_ResultTopic parameter must be set to use the commands. If a topic name is entered, the concentrator stays in permanent connection mode with the MQTTS server.
Result topic	MQTT2_ResultTopic	Name of the topic for the results of commands sent to the concentrator. The MQTT2_ControlTopic parameter must be set to use the commands. If a topic name is entered, the concentrator stays 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 alarms will be published to the broker. If a topic name is entered, the concentrator stays in permanent connection mode with the MQTTS server.
Enable advanced data option	MQTT2_EnableAdvancedData	Publication of the number of complete reads over this acquisition period in the data topic. The possible values are: • 0: Deactivated • 1: Enabled

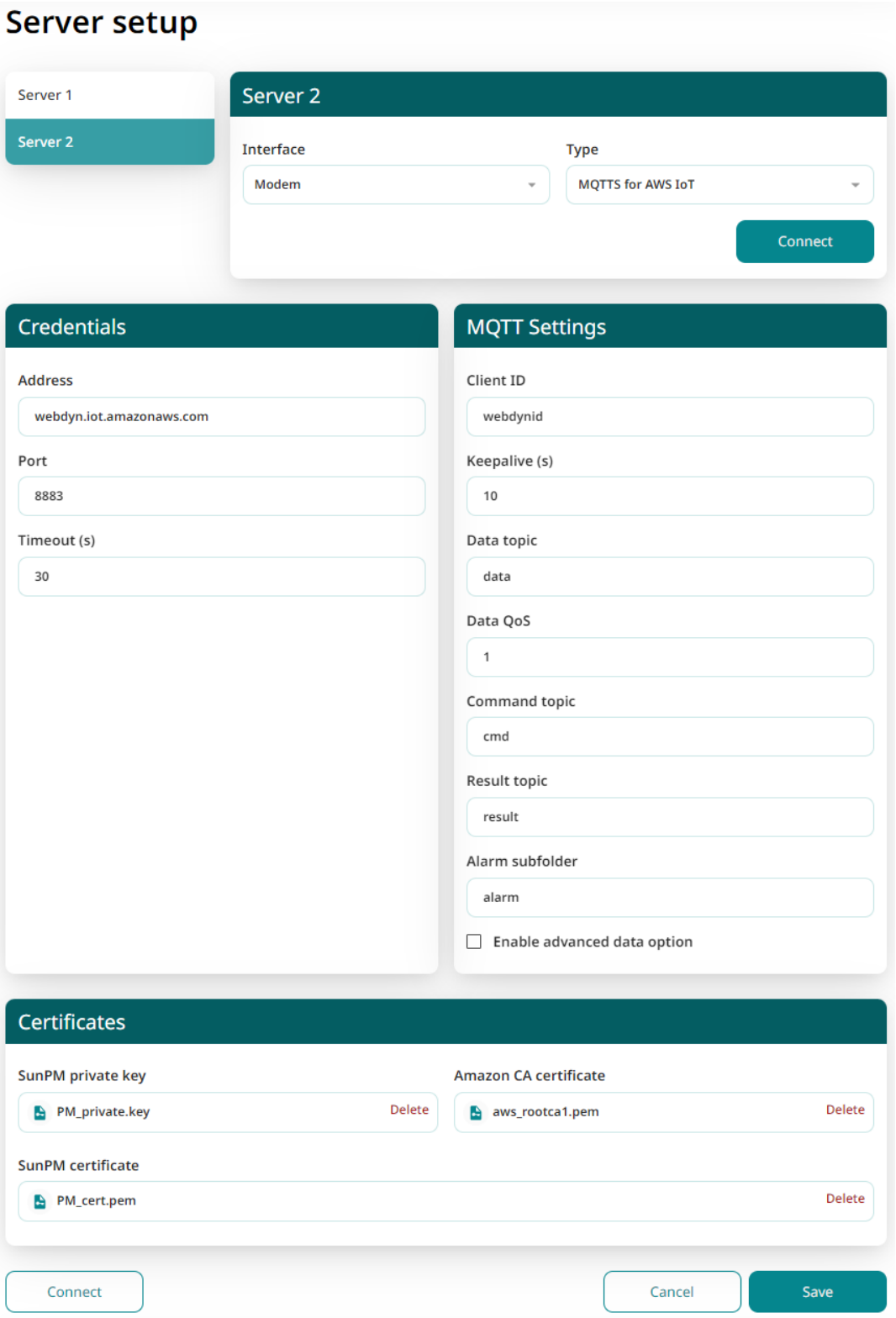
Certificates:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
SunPM Private Key	MQTT2_KeyFile	Name of the file containing the private key specific to the concentrator used for the connection. The file must be retrieved from your MQTTS server and imported to the concentrator using FTP or the web interface.
CA certificate	MQTT2_CaCertFile	Name of the certificate used to authenticate the entered MQTTS server. The certificate is to be retrieved from your MQTTS server and imported to the concentrator using FTP or the web interface.
SunPM certificate	MQTT2_CertFile	Name of the certificate specific to the concentrator used for the connection. The certificate is to be retrieved from your MQTTS server and imported to the concentrator using FTP or the web interface.

MQTTS is only available on server 2 (backup).
If the command topic is filled in, then the connection to the MQTT server will be permanent to be able to receive the sent commands.

3.2.3.3.6 MQTTS AWS IoT

The AWS IoT MQTTS server is a secure Amazon server with certificate identification. Certificates and a private key must be imported to the concentrator.


Server setup
Server 1
Server 2
Server 2
Interface
Modem
Type
MQTTS for AWS IoT
Connect
Credentials
Address
webdyn.iot.amazonaws.com
Port
8883
Timeout (s)
30
MQTT Settings
Client ID
webdynid
Keepalive (s)
10
Data topic
data
Data QoS
1
Command topic
cmd
Result topic
result
Alarm subfolder
alarm
Enable advanced data option
Certificates
SunPM private key
PM_private.key
Delete
Amazon CA certificate
aws_rootca1.pem
Delete
SunPM certificate
PM_cert.pem
Delete
Connect
Cancel
Save

The server 2 parameters are:

Server 2:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Interface	SERVER2_Interface	Selection of the network interface to use for the server: • **Ethernet** (*see section 3.2.2.1: "Ethernet (Local)"*) • **Modem** (*see section 3.2.2.2: "Modem (Mobile)"*)
Type	SERVER2_Type	Server protocol selection: • **MQTTS for AWS IoT**: MQTTS server on "AWS IoT"

Credentials:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Address	SERVER2_Address	IP address or server name
Port	MQTT2_Port	MQTTS server port (the default is 8883)
Timeout (s)	MQTT2_Timeout	Maximum waiting time in seconds for the MQTTS server response. If the server has not responded within the allotted time, the transmission is stopped and reattempted during the next schedule. Only works with QoS 1 or QoS 2.

MQTT Settings:

Web interface	Parameter in <uid>_config.ini	Description
Client ID	MQTT2_ClientId	Customisable device identifier on the MQTTS server. This parameter is retrieved from your MQTTS server.
Keepalive (s)	MQTT2_KeepAlive	If there is no exchange with the MQTTS server for the time defined in seconds, the concentrator sends a ping to the MQTTS server to check the connection. If the value is "0", KeepAlive is disabled.

		If the concentrator is in permanent connection mode with the MQTTS server and a disconnection is detected after a KeepAlive, the concentrator will automatically reconnect to the MQTTS server.
Data topic	MQTT2_Topic	Topic name for the data uploaded by the concentrator.
Data qos	MQTT2_QoS	Guaranteed service number for message sending (Quality Of Service). The possible values are: • 0: The message will only be sent once, i.e. with no guaranteed 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 commands to be retrieved by the concentrator. The MQTT2_ResultTopic parameter must be set to use the commands. If a topic name is entered, the concentrator stays in permanent connection mode with the MQTTS server.
Result topic	MQTT2_ResultTopic	Name of the topic for the results of commands sent to the concentrator. The MQTT2_ControlTopic parameter must be set to use the commands. If a topic name is entered, the concentrator stays 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 alarms will be published to the broker. If a topic name is entered, the concentrator stays in permanent connection mode with the MQTTS server.
Enable advanced data option	MQTT2_EnableAdvancedData	Publication of the number of complete reads over this acquisition period in the data topic. The possible values are: • 0: Deactivated • 1: Enabled

Certificates:

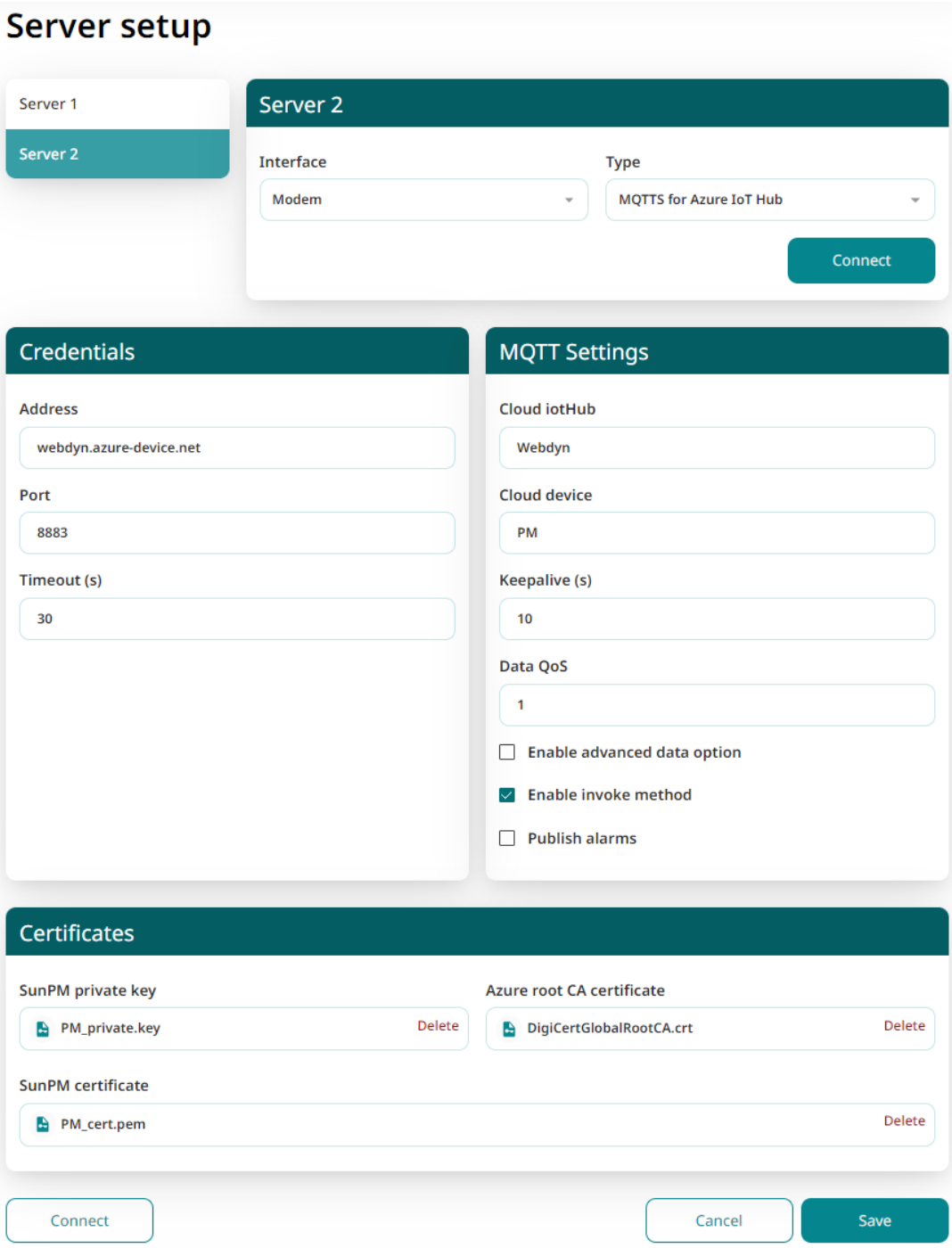
Web interface	Parameter in <uid>_config.ini	Description
SunPM Private Key	MQTT2_KeyFile	Name of the file containing the private key specific to the concentrator used for the connection. The file must be retrieved from your MQTTS server and imported to the concentrator using FTP or the web interface.

Amazon root CA certificate	MQTT2_CaCertFile	Name of the certificate used to authenticate the entered MQTTS server. The certificate is to be retrieved from your MQTTS server and imported to the concentrator using FTP or the web interface.
SunPM certificate	MQTT2_CertFile	Name of the certificate specific to the concentrator used for the connection. The certificate is to be retrieved from your MQTTS server and imported to the concentrator using FTP or the web interface.

AWS IoT MQTTS is only available on server 2 (backup).
If the command topic is filled in, then the connection to the MQTT server will be permanent to be able to receive the sent commands.

3.2.3.3.7 MQTTS Azure IoT Hub

The Azure IoT Hub MQTTS server is a secure Microsoft server with certificate-based identification. A certificate and private key must be imported to the concentrator.



The server 2 parameters are:

Server 2:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Interface	SERVER2_Interface	Selection of the network interface to use for the server: • **Ethernet** (*see section 3.2.2.1: "Ethernet (Local)"*)

		• **Modem** (*see section 3.2.2.2: "Modem (Mobile)"*)
Type	SERVER2_Type	Server protocol selection: • **MQTTS for Azure IoT Hub**: MQTTS server on "Azure IoT Hub"

Credentials:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Address	SERVER2_Address	IP address or server name
Port	MQTT2_Port	MQTTS server port (the default is 8883)
Timeout (s)	MQTT2_Timeout	Maximum waiting time in seconds for the MQTTS server response. If the server has not responded within the allotted time, the transmission is stopped and reattempted during the next schedule. Only works with QoS 1 or QoS 2.

MQTT Settings:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
Cloud IoT Hub	MQTT2_CloudProjectId	Unique, customisable identifier for 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	Unique, customisable device identifier 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 there is no exchange with the MQTT server for the time defined in seconds, the concentrator sends a ping to the MQTT server to check the connection. If the value is "0", KeepAlive is disabled. If the concentrator is in permanent connection mode with the MQTT server and a disconnection is detected after a KeepAlive, the concentrator will automatically reconnect to the MQTT server.
Data qos	MQTT2_QoS	Guaranteed service number for message sending (Quality Of Service). The possible values are: • 0: The message will only be sent once, i.e. with no guaranteed 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 Azure IoT Hub MQTTS server
Enable advanced data option	MQTT2_EnableAdvancedData	Publication of the number of complete reads over this acquisition period in the data topic. The possible values are: • 0: Deactivated • 1: Enabled
Enable invoke method	MQTT2_EnableInvokeMethod	Enable method calls. Allows the use of dedicated topics. The possible values are: • Unchecked: Disables method calls • Checked: Enables method calls. The concentrator stays in permanent connection mode with the MQTTS server.
Publish alarms	MQTT2_EnableAlarmPost	Enable the publication of alarms on the dedicated topic. The possible values are: • Unchecked: Disables alarm publication

		• Checked: Enables alarm publication. The concentrator stays in permanent connection mode with the MQTTS server.

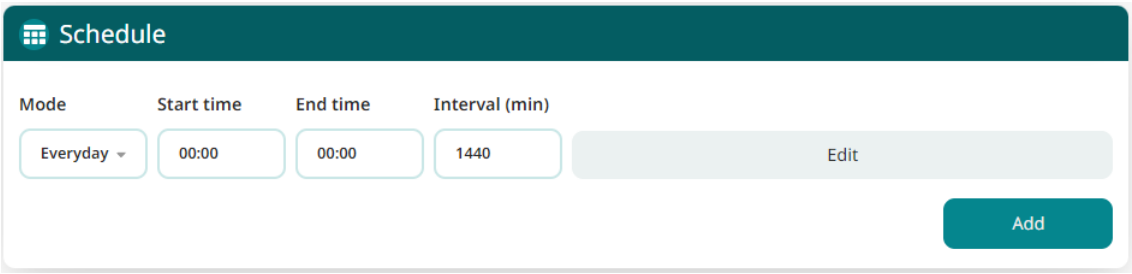
Certificates:

Web interface	Parameter in the <uid>_config.ini configuration file	Description
SunPM private key	MQTT2_KeyFile	Name of the file containing the specific private key or the key shared with the concentrator used for the connection. The file must be retrieved from your MQTTS server and imported to the concentrator using FTP or the web interface.
Azure root CA certificate	MQTT2_CaCertFile	Name of the certificate used to authenticate the entered MQTTS server. The certificate is to be retrieved from your MQTTS server and imported to the concentrator using FTP or the web interface.
SunPM certificate	MQTT2_CertFile	Name of the certificate specific to the concentrator used for the connection. The certificate is to be retrieved from your MQTTS server and imported to the concentrator using FTP or the web interface.

The MQTTS Azure IoT Hub is only available on server 2 (backup).
If the command topic is filled in, then the connection to the MQTT server will be permanent to be able to receive the sent commands.

3.2.3.3.8 Schedule

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



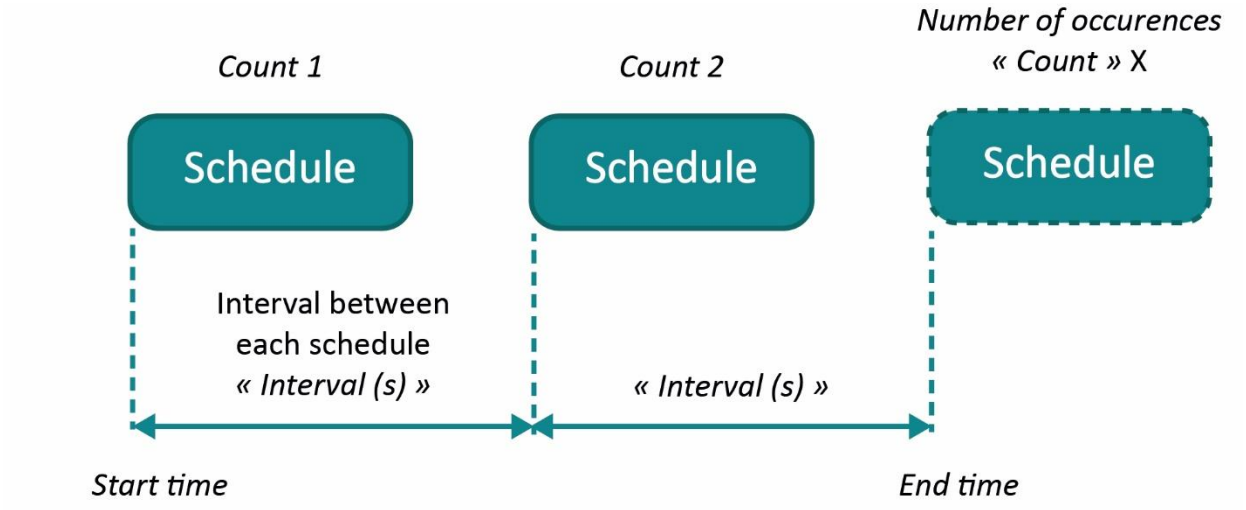
The “Schedule” parameters are:

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”.
End Time	Schedule end time in: “HH:MM”. The “End Time” does not appear in the “<UID>_var.ini” file, but allows you to automatically calculate the “Count” variable, which is only present in the “<UID>_var.ini” file.
Interval(min)	Schedule repeat interval in minutes. Special case: if the value “0” is filled in, then it is automatically switched to “1”.

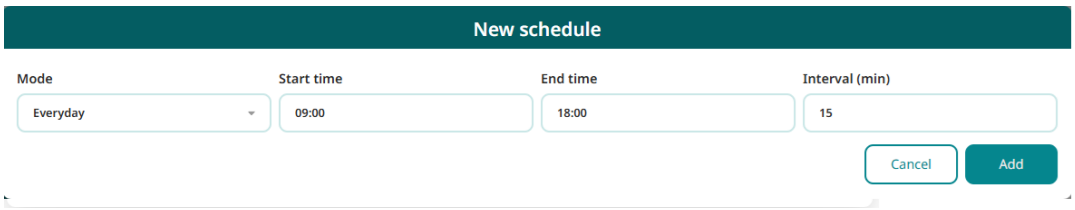
Schedules are only defined for one day. If the value of “End Time” is ‘00:00’, then the end of the schedule will be midnight.

The "End Time" value is automatically adjusted after the schedule has been validated, depending on the number of, possible intervals, and will indicate the time of the last occurrence of the day.

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*” variable and the interval between each event by the “*Interval*” field The "*End time"* field indicates the time of the day’s last occurrence. (*see chapter 3.1.2.1.2 “< UID>_var.ini” file*)



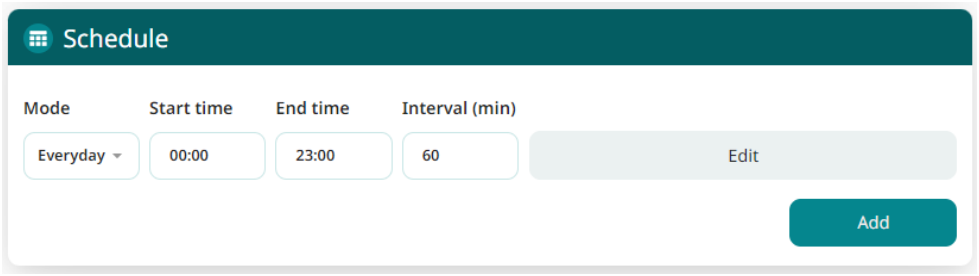
The "Add" button is used to add a new schedule.



An ongoing schedule can always be modified. To do that, simply click the "Edit" button to display the schedule editor.

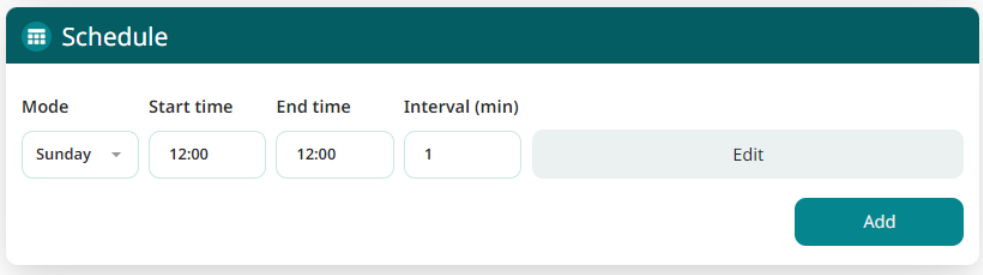
Example 1:

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



Example 2:

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



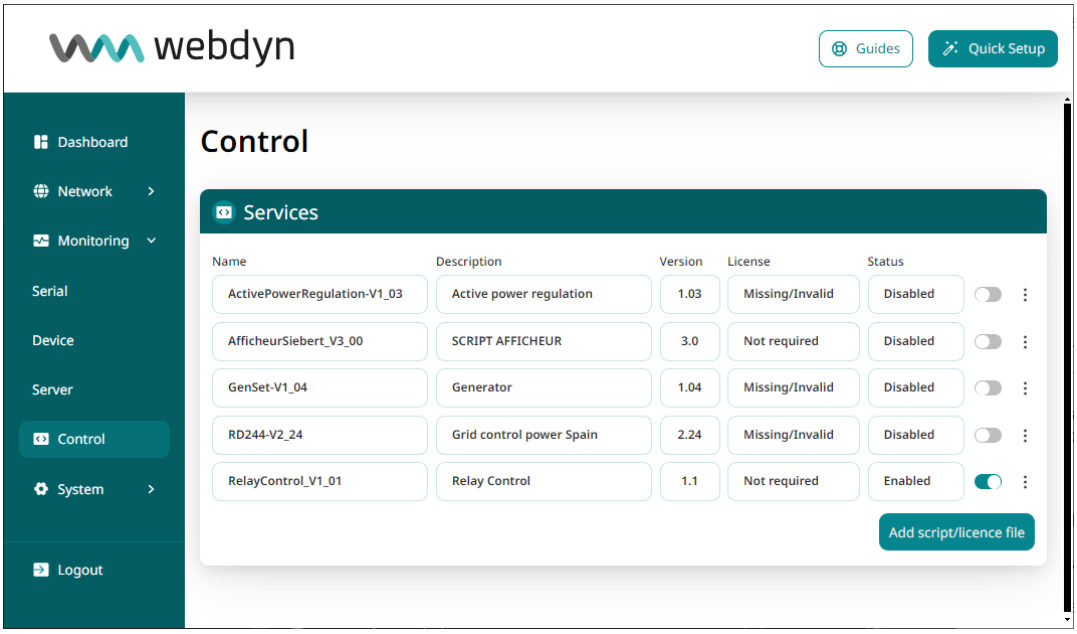
3.2.4 Control

The WebdynSunPM concentrator has a powerful script-based device management and customisation tool also known as services.

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”)

Access to service configuration and management is from the “Control” page on the local web site:



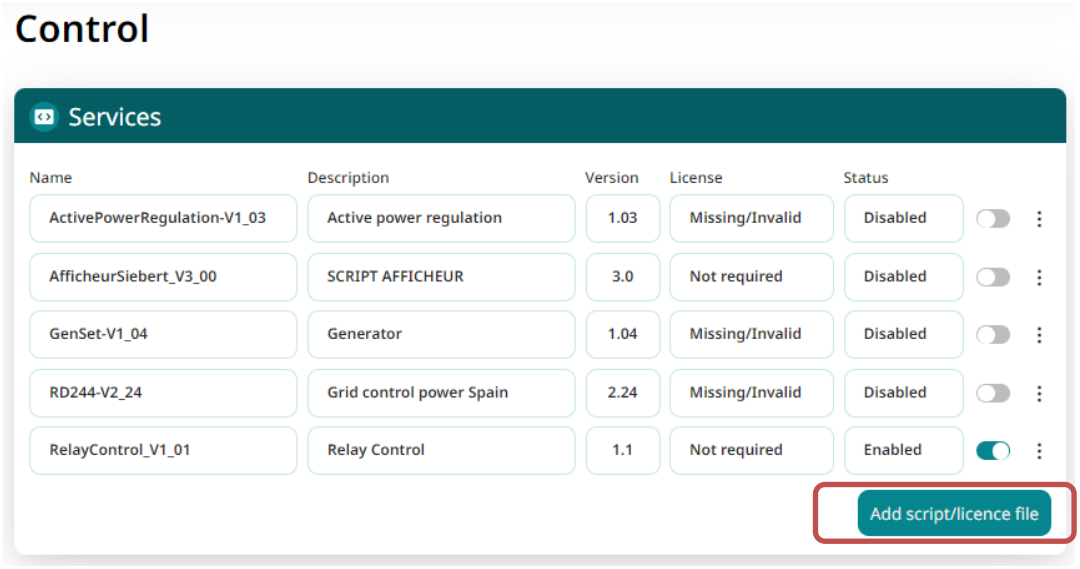
This page can be used to import new scripts, enable them, disable them or even delete them, or to view the run log.

A set of services are built into the concentrator by default, some of which require a licence to be paid for to be able to use them.

3.2.4.1 Import a service or a licence

One of the ways to import a new service into the concentrator is by using the local web site.

To do that, click the “Add script/licence file” button as shown below.

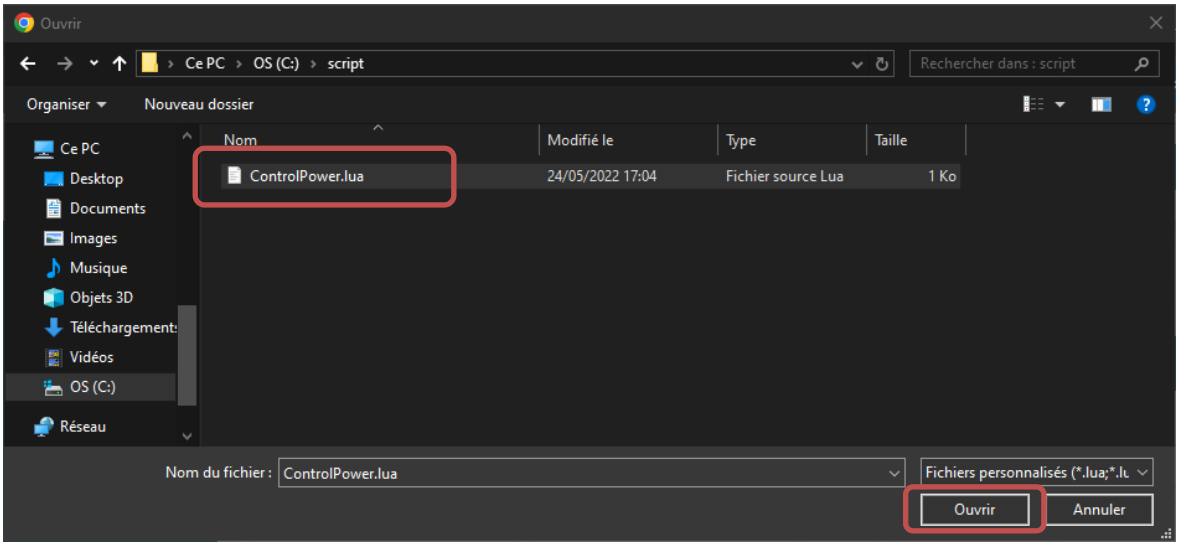


A dialogue window is displayed to select the file to import.

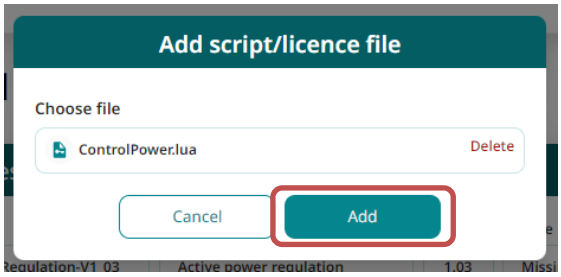
The window displays the “LUA” type files, i.e. with a file extension of:

- ".lua": an unencrypted LUA script
- ".luax": an encrypted LUA script with client keys
- ".luaw": an encrypted LUA script with a Webdyn licence

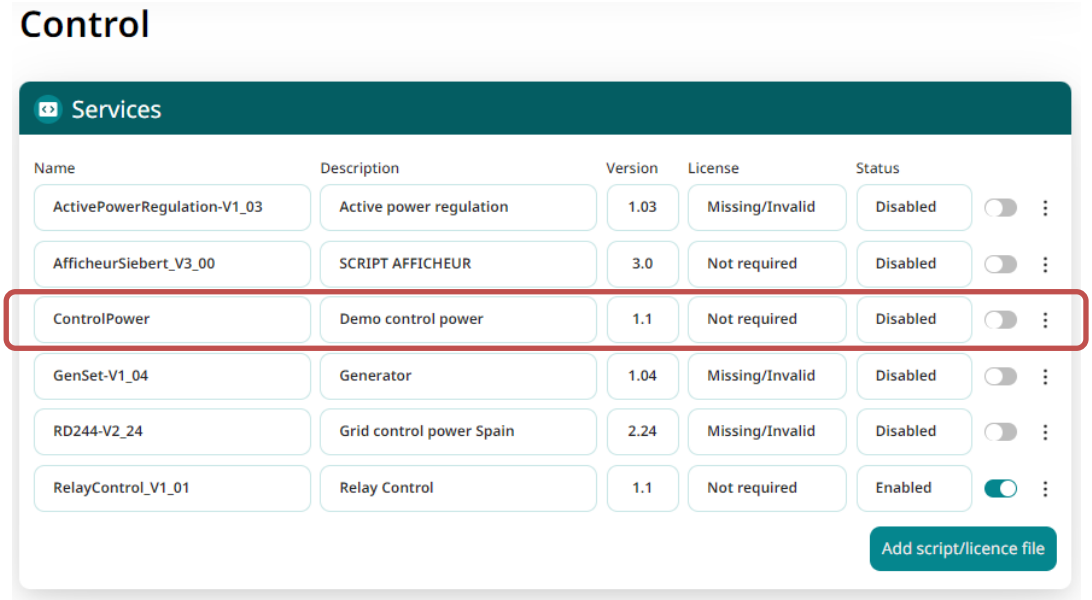
Then select the file to import and click the “Open” button.



Finally, click the “Add” button to complete the operation.



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



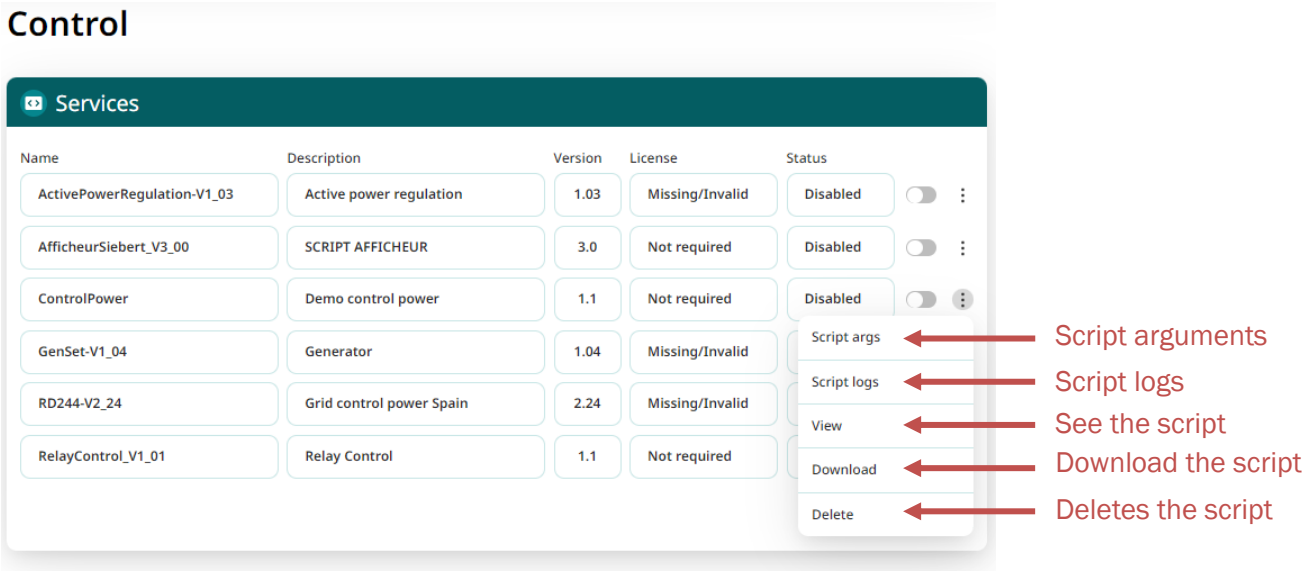
When importing a ".luax" format script, if the following error message is displayed "error:Error deciphering test_script.luax: stoul", it means that the webdynSunPM does not have the decryption keys. In that case, they must be sent using the "setKey" command. (See section5.3.17: ""setKey": Addition of client script decryption keys")

Note that the services are imported stopped, meaning they are not started automatically.

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

```
h
        version = 1.1,
        label = "Demo control power"
}
```

The description is 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.



Parameters can be passed to the script, just enter the "Script args" field in the “Scripts args” available in the service option menu and validate by clicking the “Save” button for the service to take them into account when it is enabled. To delete parameters, delete the arguments and click the "Save" button.

3.2.4.2 Enabling/Disabling a service

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

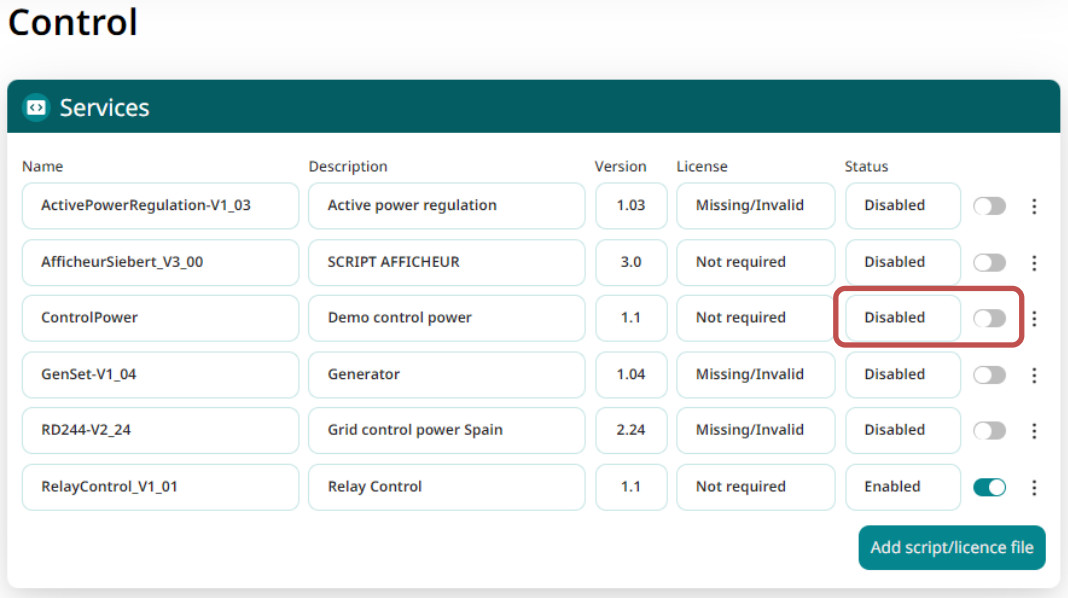
```
f  ction wsInit()
      wd.log("Control power initialized")
e  l
```

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

Disabling a service meets stopping it running.

When webdynSunPM is restarted, all the services return to the same state as before. For example, if the service was started, it will be started.

A specific button is used to enable and disable:



When the service is disabled, its “Disabled” status is displayed.

When the service is enabled, its status is “ Enabled”.

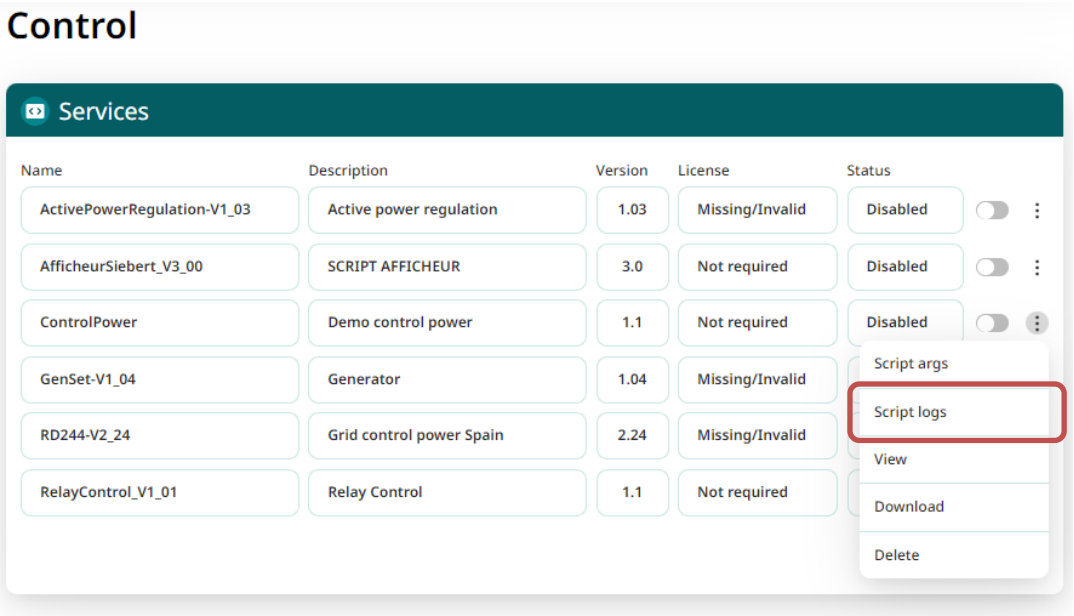
3.2.4.3 Viewing the service log

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

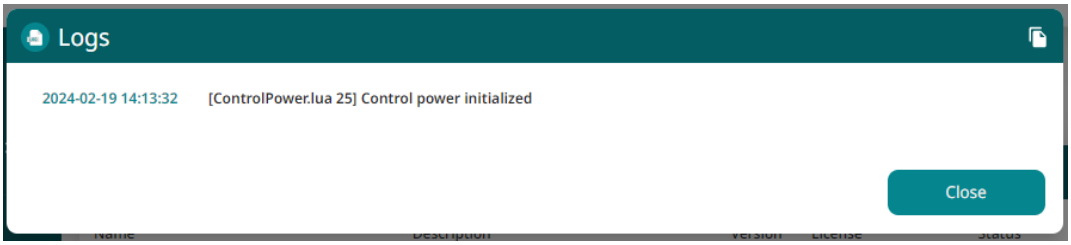
Thus, the following code will display the “*Control power initialized*” string in the start-up service log file:

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

The log file is displayed by going to the "Script logs" menu available in the service options menu.



Pressing the “Script logs” menu displays the following page:



The page is closed by pressing the "Close" button.

3.2.4.4 View the service

The source code for the services loaded onto the concentrator can be displayed by clicking the “View” menu available from the service option menu.

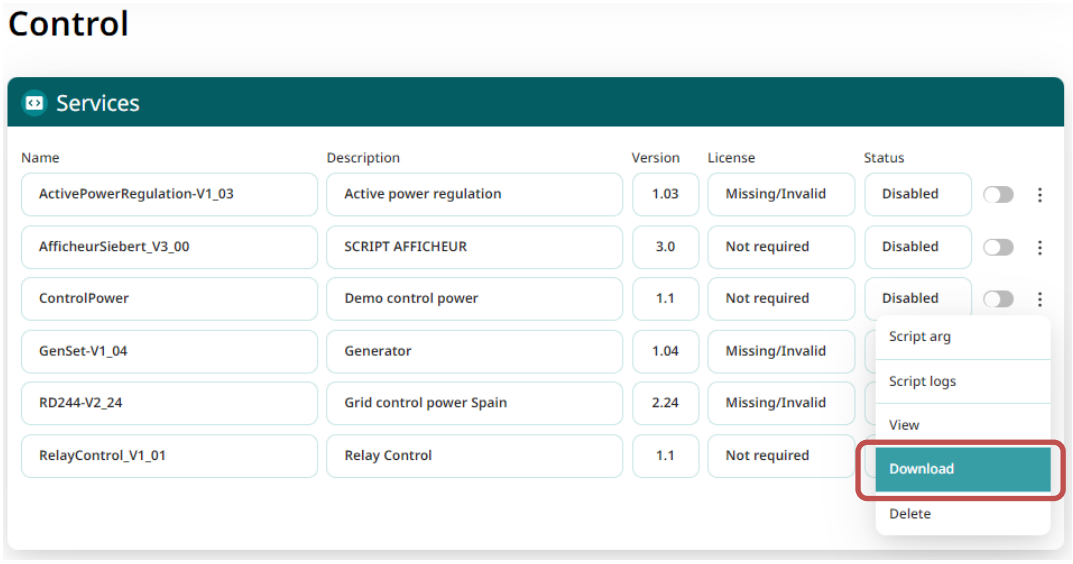
```
header = {
    version = 1.1,
    label = "Demo control power"
}


function wsInit(param)
    wd.log("Initialized")

    if param ~= nil then
        wd.log("parametre = "..param)
    end
end

function wsStop()
    wd.log("Uninitialized")
end


function wsTick()
    wd.log("tick")
end
```

When script source code is displayed, click outside the window to make it disappear.

Only unencrypted LUA scripts can be viewed.

3.2.4.5 Export a service

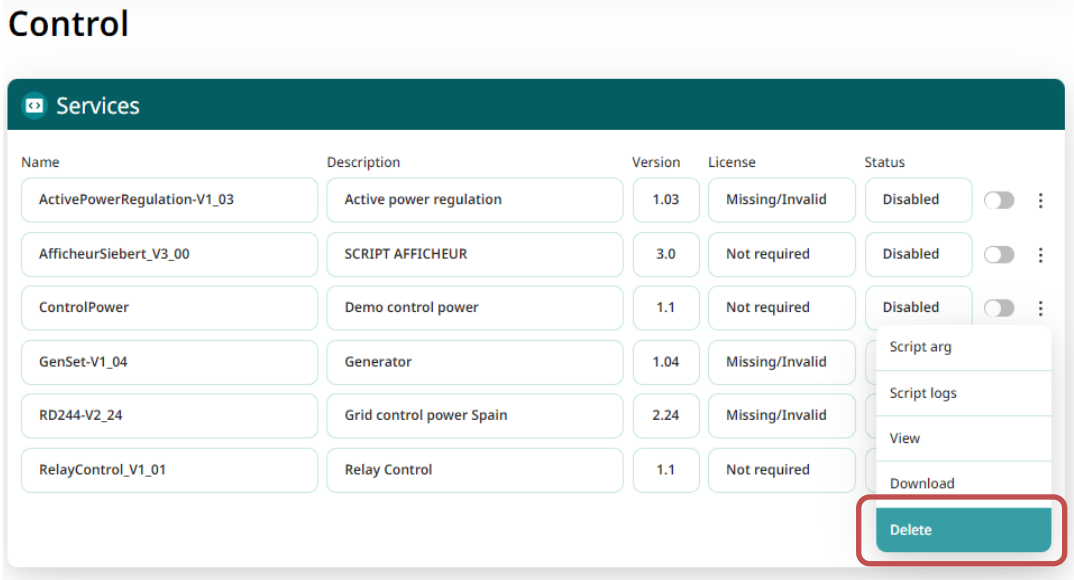
You can export the source code of the services loaded into the concentrator by clicking the "Download" menu available in the service options menu. Pressing the “Download” menu launches an immediate local script load by the browser.



Only unencrypted LUA scripts can be exported.

3.2.4.6 Deleting a service

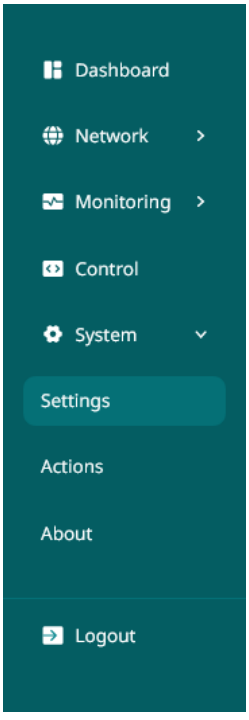
A service loaded in the concentrator can be deleted by clicking the "Delete" menu available in the service options menu. A confirmation message will ask you if you really want to remove this service from the concentrator.



Deleting a running service will, of course, cause it to be stopped and deleted.

3.2.5 System

All the system settings are grouped together in the “System” menu.



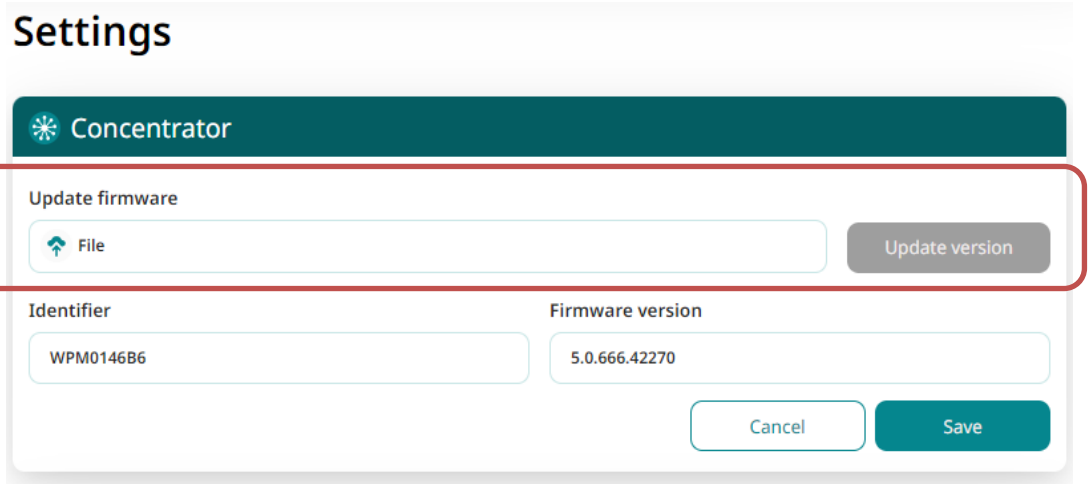
3.2.5.1 Settings

The system parameter “Settings” sub-menu shows all the parameters specific to the concentrator.

3.2.5.1.1 Updating and identifier

The "Concentrator" part is used to update and modify the concentrator identifier.

Update:



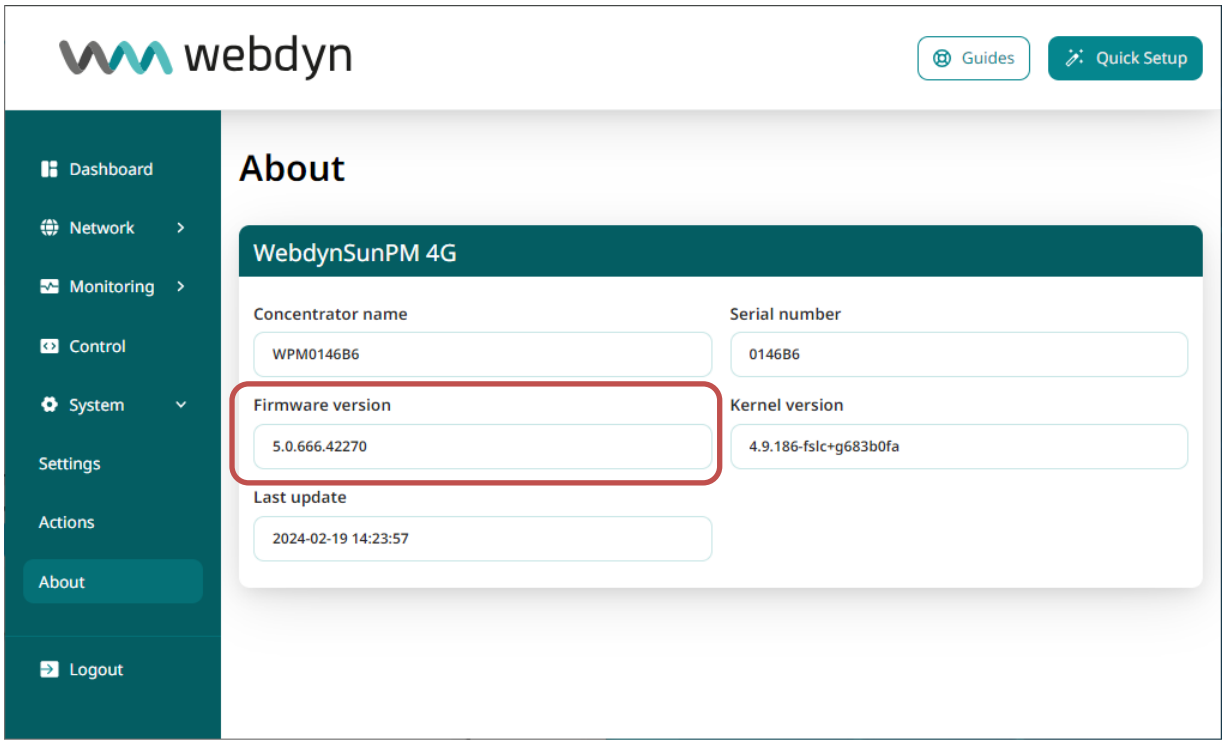
Follow the steps below to update the the concentrator:

1. Retrieve the firmware from the web site (*see section 6: “Update”*):
   https://www.webdyn.com/support/webdynsunpm/
2. Unzip the retrieved file,
3. Click the "File" field in the "Update firmware" parameter. A window opens used to select the new firmware,
4. Select the “wgapp_x.x.x.xxxxx.spm” firmware that has a “.spm “ extension and click the “Open” button.

5. Press the "Update version" button.
6. Monitor the update progress next to the “Update version” button:

7. Please wait while the update is applied and the concentrator restarts.
8. Refresh the web page (F5 key on the keyboard)
9. Connect back to the concentrator (*see section 3.2: “Embedded web interface”*)
10. Go to the "System" menu, then to the concentrator "About" sub-menu

11. Check that the new version is shown in the “Firmware version field on the “About” field.



The concentrator has been updated.

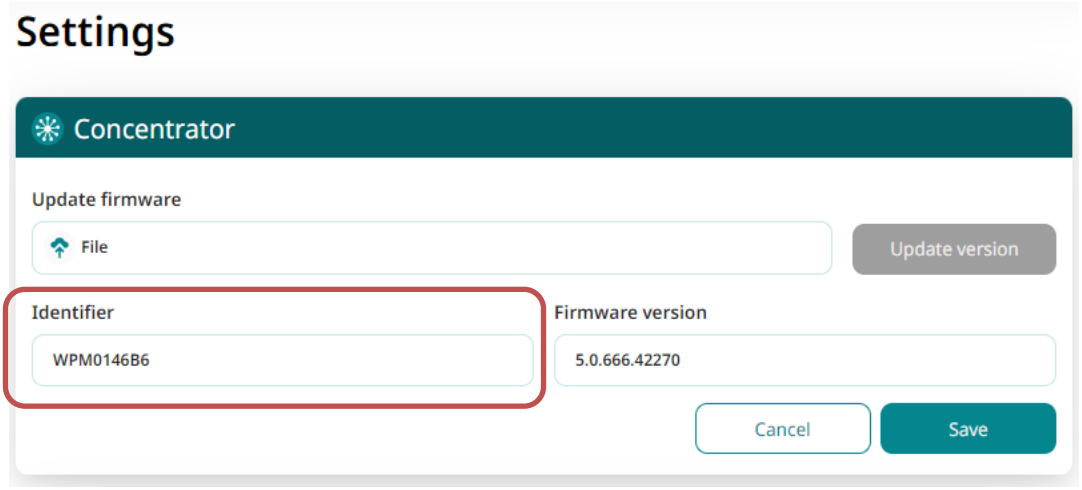
Do not disconnect the concentrator and avoid operations on it during its update.

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.

After each update, it is advisable to delete the configuration files present on the server. This operation allows the hub to automatically generate new configuration files integrating any changes during the next connection.

Identifier:

The "Identifier name" field in the "Concentrator" part of the "Settings" page is used to change the concentrator identifier.



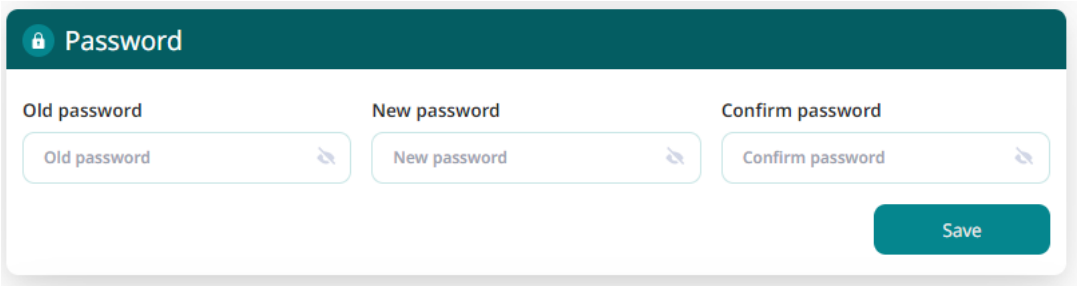
Web interface	Parameter in the <uid>_config.ini configuration file	Description
Identifier	Concentrator_Identifier	Concentrator identifier

By default, a unique “WPMxxxxxx” identifier is filled in. The “xxxxxx” corresponds to the last 6 characters of the concentrator MAC address which is provided on the product label (*see section 2.2.2: “Identification”*). To validate any change of identifier, press the "Save" button.

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>.

3.2.5.1.2 Password

The “Password” part is used to modify the password to access the concentrator web interface.



Follow the steps below to change the password:

1. Enter the current password in the “Old password” field,

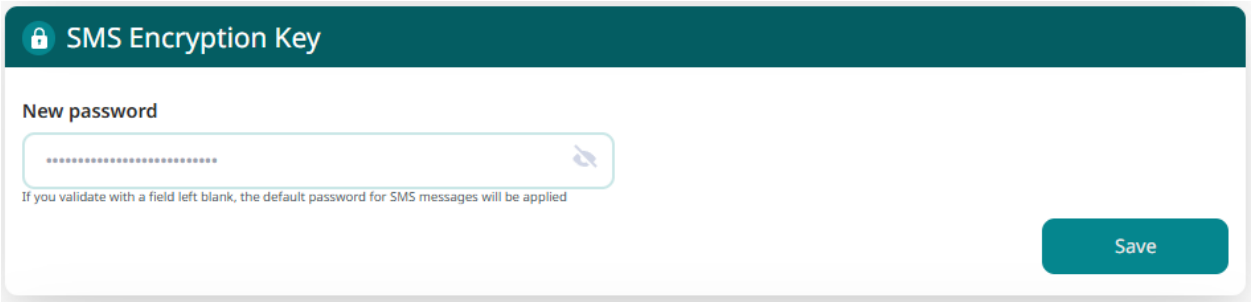
Enter the new password in the "New Password" field and use the strength indicator to make sure it meets the security requirements,
2. Enter the new password again in the “Confirm password” field
3. Validate by clicking the “Save” button.

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 servers are configured, you must completely reinitialise the concentrator by applying the factory settings using a *factory* text message command (*see section 5.3.3: ““factory”: Back to factory settings”*) or using the “Factory return” button (see section 2.4.3.2: “Factory Reset button”)

3.2.5.1.3 SMS Encryption Key

The "SMS Encryption Key" part allows you to change the SMS encryption password. The purpose of this encryption is to secure SMS exchanges with the hub. (*See chapter 5.2.3: "Text* message")



Pour changer le mot de passe SMS, il faut suivre les étapes suivantes :

1. Enter the new password in the "New Password" field,
2. Validate by clicking on the "Save" button.

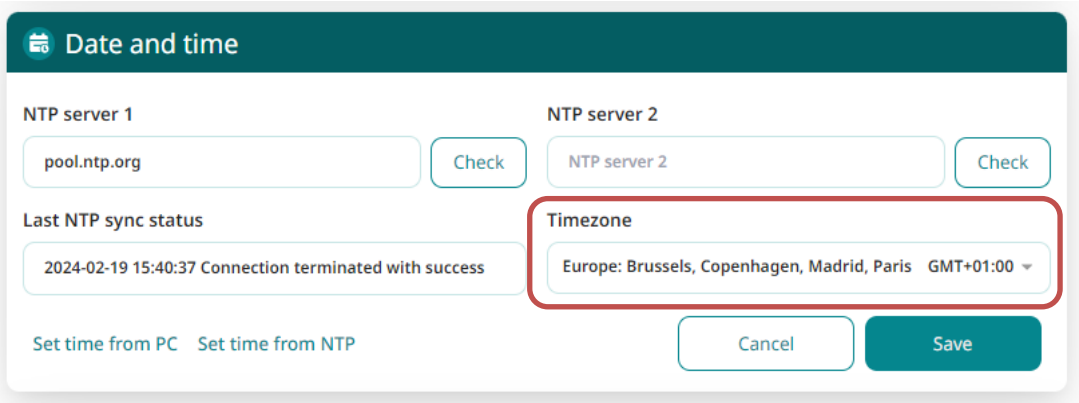
To secure access to the hub by SMS, it is strongly recommended to enter your own password. The password can also be changed by the "SMS_Password" variable in the "<uid>_config_.ini" config file . The password is not readable from the web interface.

If an empty field is validated for the SMS password, then the hub will use its own Factory password. Valid only for concentrators manufactured after the 03/26 date

3.2.5.1.4 Date and 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 has been selected, click the “Save” 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.

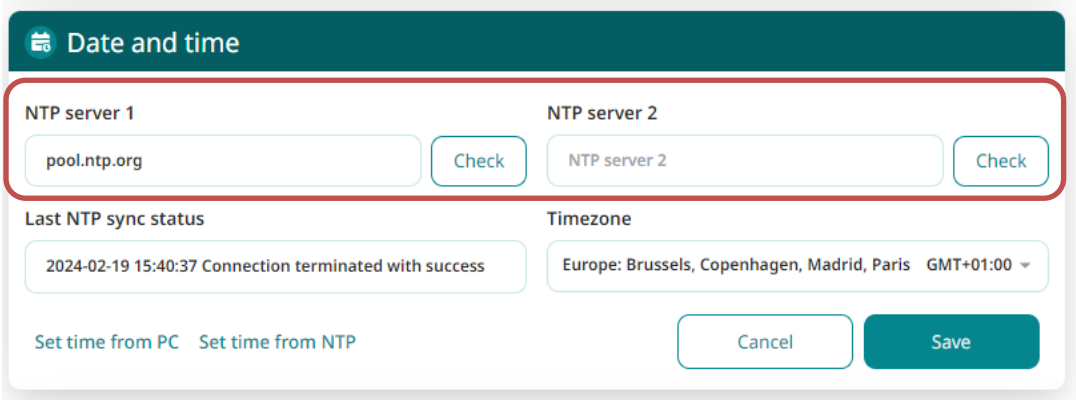
The date and time setting is:

Web interface	Parameter in the <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 in the <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.

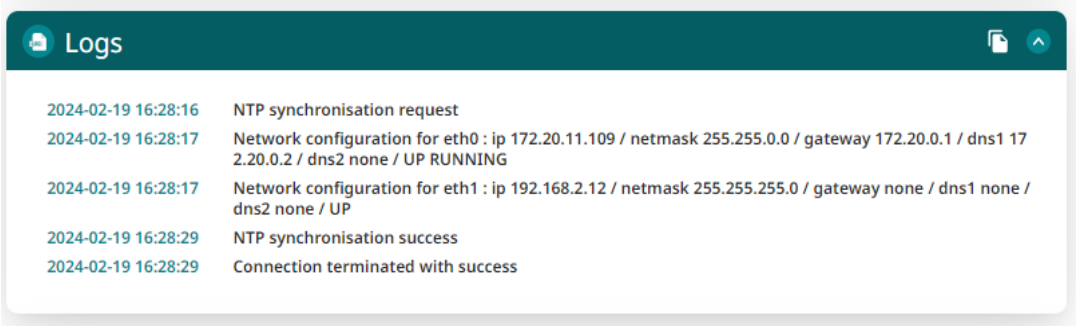
If the NTP1 and NTP2 server values are left blank, the concentrator will not use NTP synchronisation.

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”** of the NTP 1 server: Tests the NTP1 server synchronisation without applying it to the concentrator.
- **“Check”** of the NTP 2 server: Tests the NTP2 server synchronisation without applying it to the concentrator.

Observe the result on the log shown below:

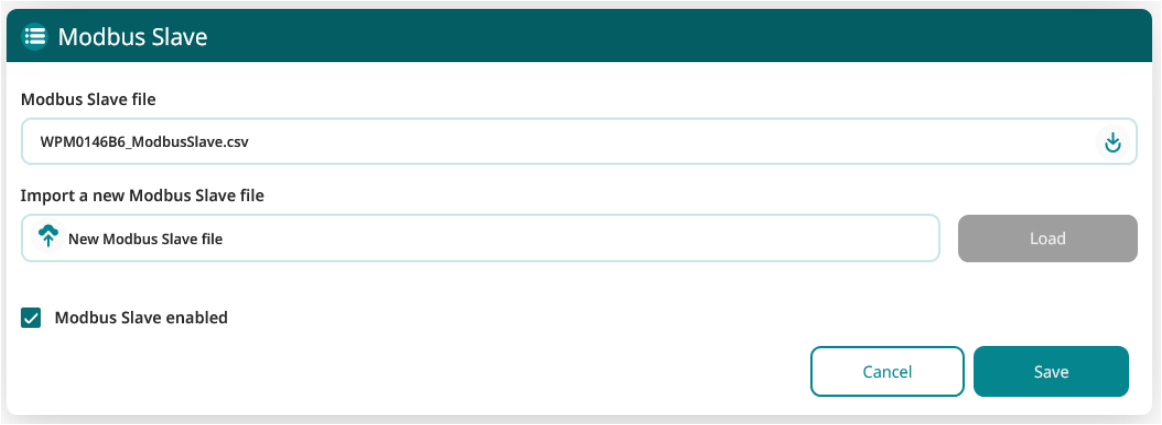


The last line of the log containing the test result is displayed and stored in the "Last NTP sync status" field in the "Date and time" part.

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.

3.2.5.1.5 Modbus Slave

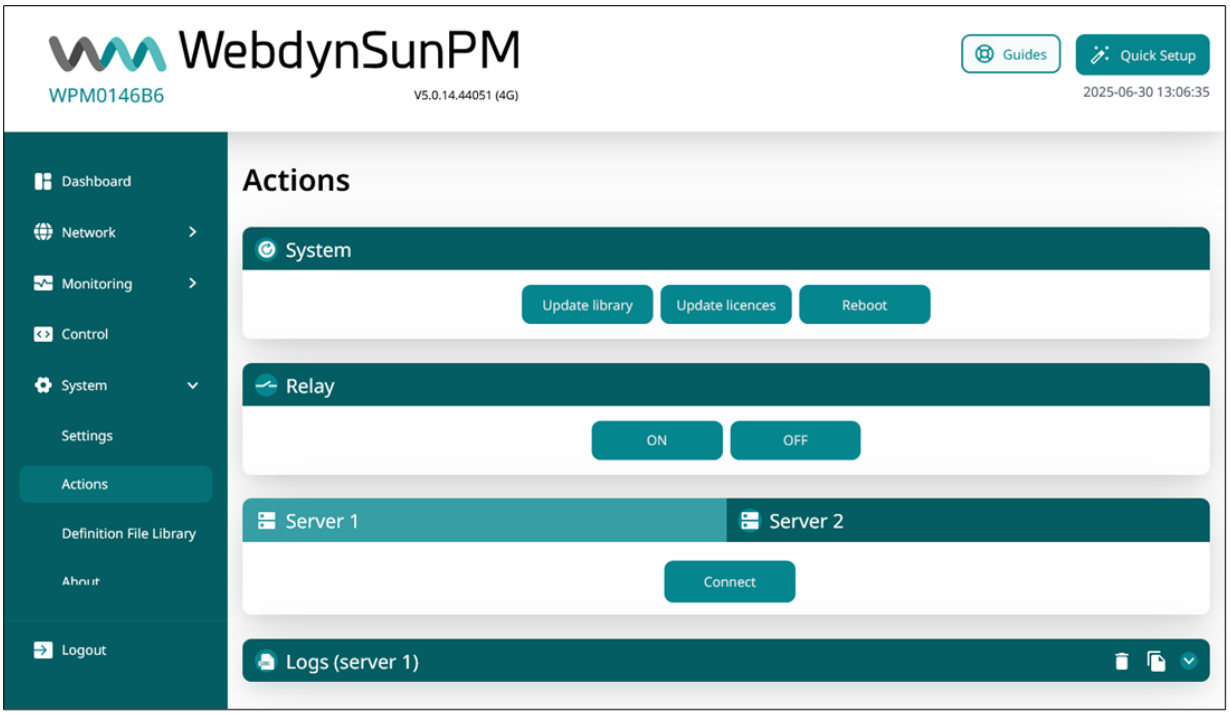
The Modbus Slave section is used to enable or disable the Modbus slave *(see section 3.4: “Modbus slave TCP”)*.



When the Modbus slave is enabled using the web interface, only the WebdynSunPM registers are accessible. The user registers require a specific definition file and can only be configured from a server.

3.2.5.2 Actions

The system's 'Actions' sub-menu can be used to trigger various actions such as a reboot or a connection to a server.



3.2.5.2.1 Reboot

The "Reboot" on the web interface is in the "Action" menu, then in the "Actions" sub-menu and finally in the "System" part. It is used to simply reboot the concentrator.

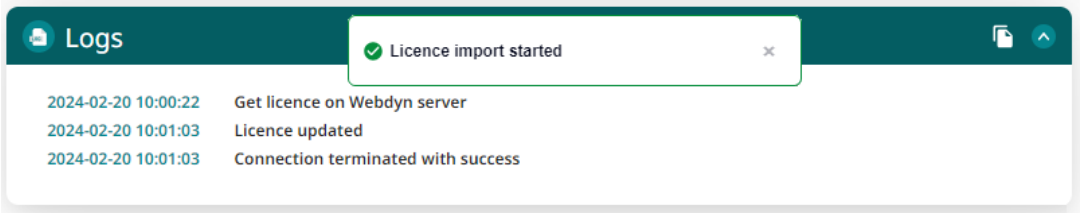
Follow the steps below to reboot the concentrator:

1. Click the "Reboot" button.
2. A notification will indicate that the restart has been taken into account. Wait while the concentrator restarts.
3. Once the concentrator has restarted, the identification page will be displayed.
4. Connect back to the concentrator (*see section 3.2: "Embedded web interface"*)
5. The concentrator is ready once again.

3.2.5.2.2 Licence updates

The web interface license update can be found in the "Action" menu, then the "Actions" sub-menu and finally in the "System" part. It is used to trigger a connection to the Webdyn license server to automatically retrieve the paying Webdyn ".luaw" licenses purchased from the Webdyn sales department (contact@webdyn.com).

When the "Update licences" button is pressed, a notification will indicate the start of the connection to the Webdyn licence server, then the licence retrieval will be tracked in the log located below.



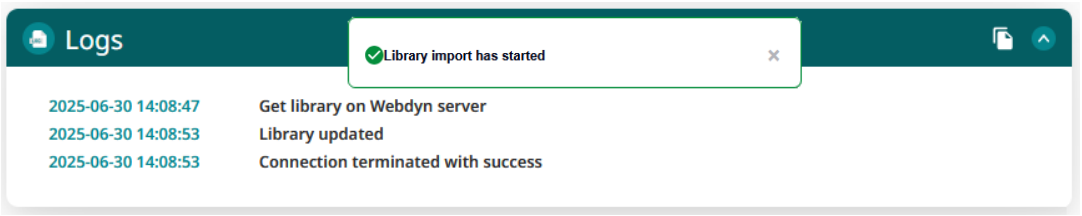
The licence file is specific to a WebdynSunPM. The same file cannot be used on several concentrators. The licence file contents must not be modified, otherwise the concentrator's licence management will be blocked.

The downloaded licence is automatically synchronised with the configured remote server. If the product is returned to the factory and the licence is deleted on the remote server, the licences remain available on the Webdyn licence server.

3.2.5.2.3 Update Library

Definition file library can be updated via a connection to Webdyn server

When clicked, a pop up notify the beginning of the update. Status will be available in the log below

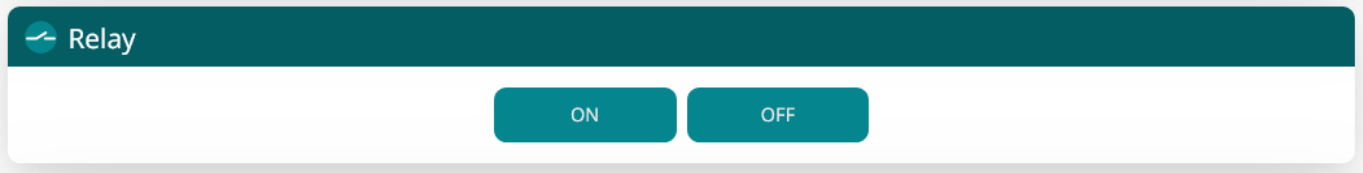


Definition file library is distributed by Webdyn. The library is encrypted to avoid any modification

The downloaded library is automatically synchronized with the remote server. In case of a factory setting and removal of the library on the remote server, the library is always available on Webdyn's

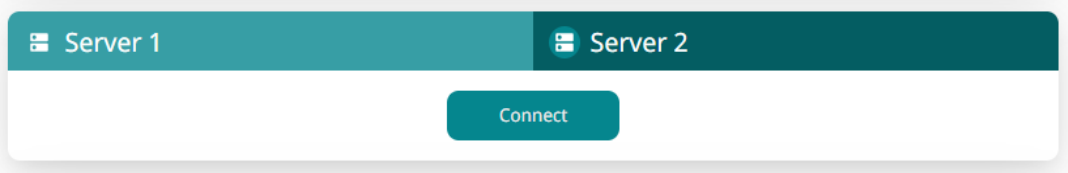
3.2.5.2.4 Relay

Relay can be switched via the webserver

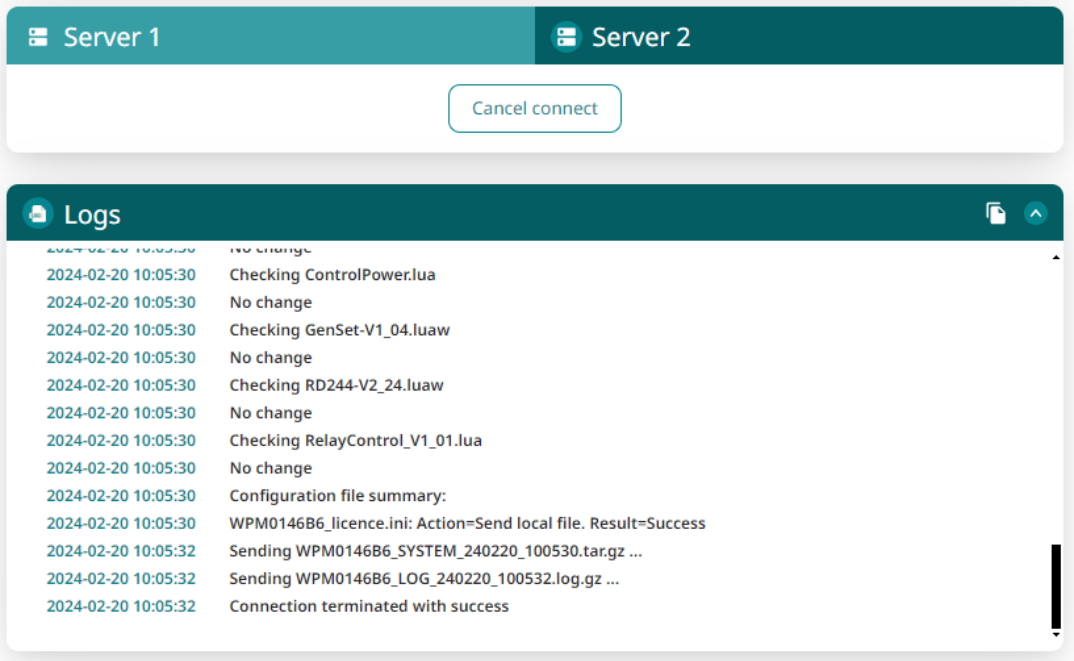


3.2.5.2.5 Connection to the servers

To trigger a connection to a server, select the server (server 1 or server 2) you want and click the 'Connect' button. A "Logs" part underneath is used to monitor the connection log and to view all the files exchanged between the concentrator and the remote server. Looking at the last line is a quick way to see if the connection was successful or not.

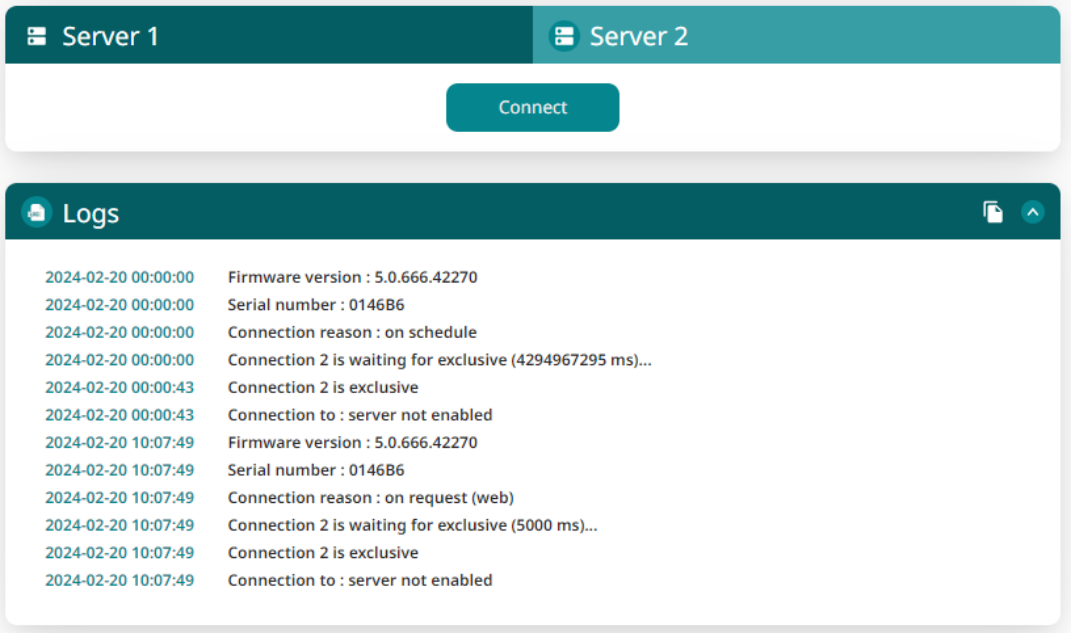


Ending an ongoing connection using the "Cancel connect" button is not instant. Indeed, if an action is in progress, it must complete first. Every line in the connection log represents an action.



3.2.5.2.6 Logs

The connection log shows the various events on server 1 and server 2.

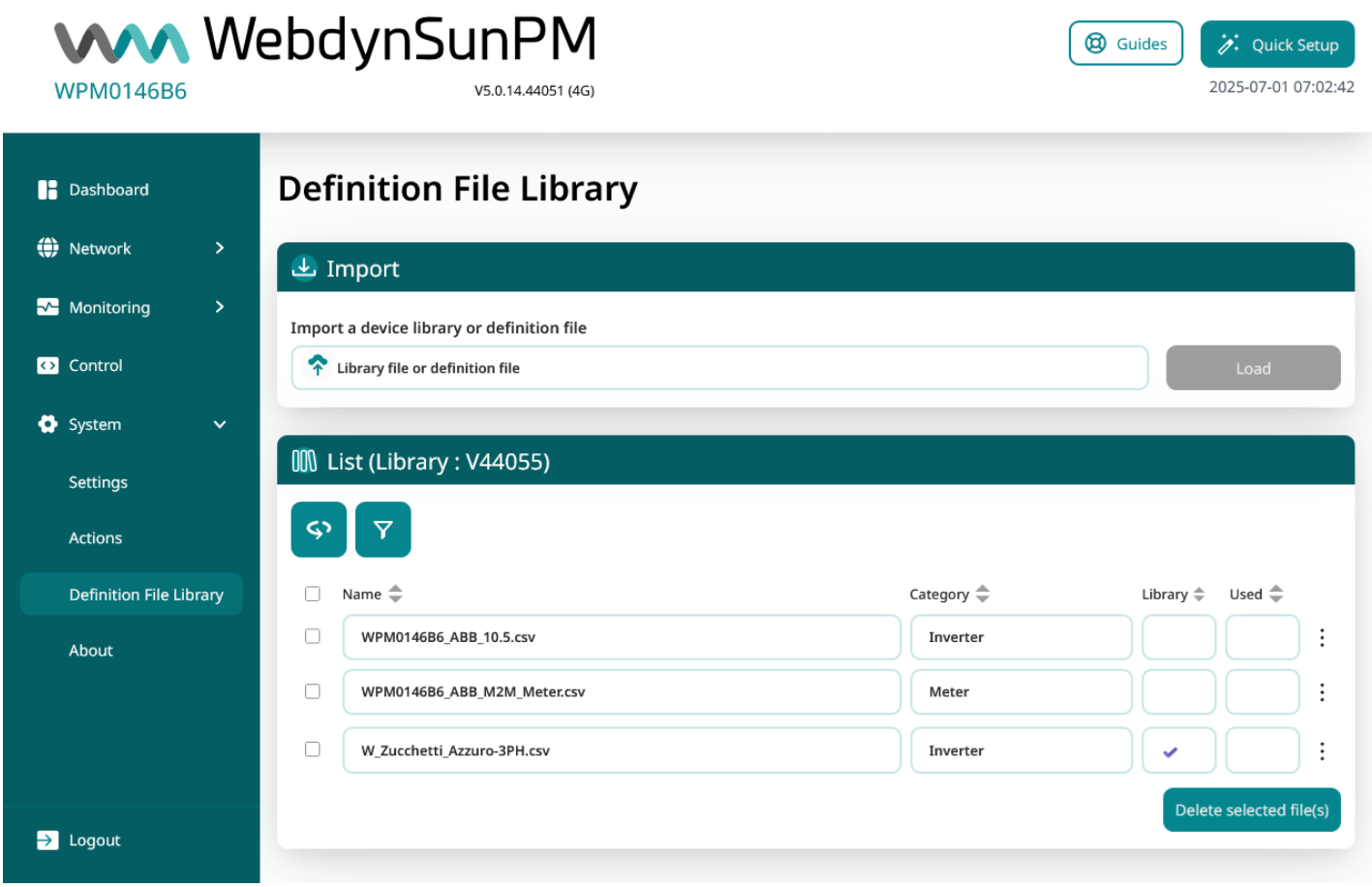


To select a server's connection log, first select server 1 or server 2 in the “server” section just above. By default, the connection log displays server 1 events.

If the concentrator is restarted, the connection log will be erased from the web pages and previous events will no longer be viewable on the web interface. The previous events remain stored and are uploaded to the configured remote server.

3.2.5.3 Definition File Library

The definition file library is used to manage all the definition file stored in the device.



3.2.5.3.1 Import

“Import” section is used to download manually a library or definition files. Supported file formats are the following:

- « .libw » for the library,
- « .csv » for definition files

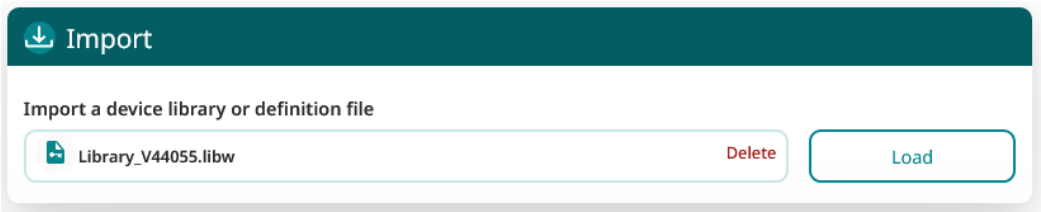
> The latest version of the library can be downloaded from Webdyn server via a simple action on the webserver (see chapter 3.2.5.3)

Follow the steps below:

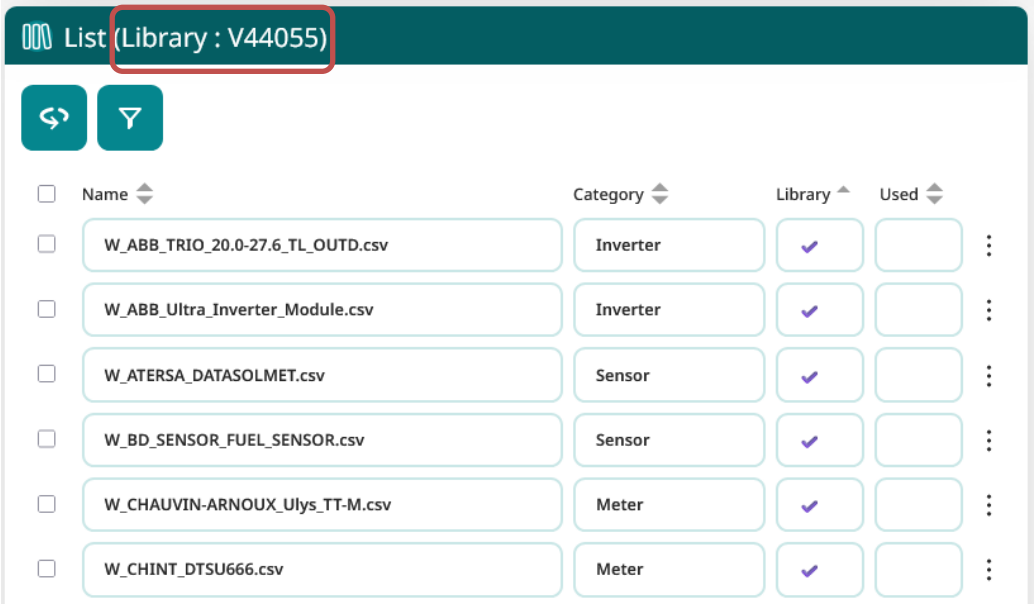
Download from :

1. https://repository.webdyn.com/products/WebdynSunPM/defLibrary/defLibrary.zip

2. Click on « Library file or definition file » from « Import a device library or definition file ».
3. Select the library file with file extension « .libw » and click on load



4. Verify the version. The list now all the definition files from the

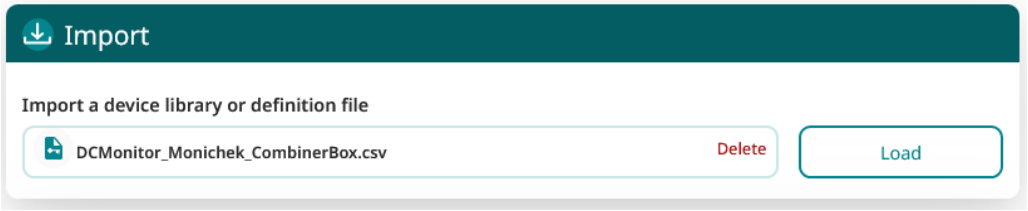


If no library is stored in the concentrator, no version will be displayed

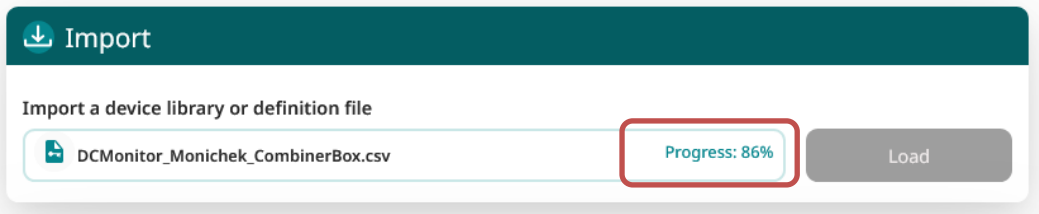
If a library is already stored in the concentrator, an import will replace the existing library. If a definition file was used by an equipment, the definition file is kept.

To load a definition file in the concentrator, follow the steps below:

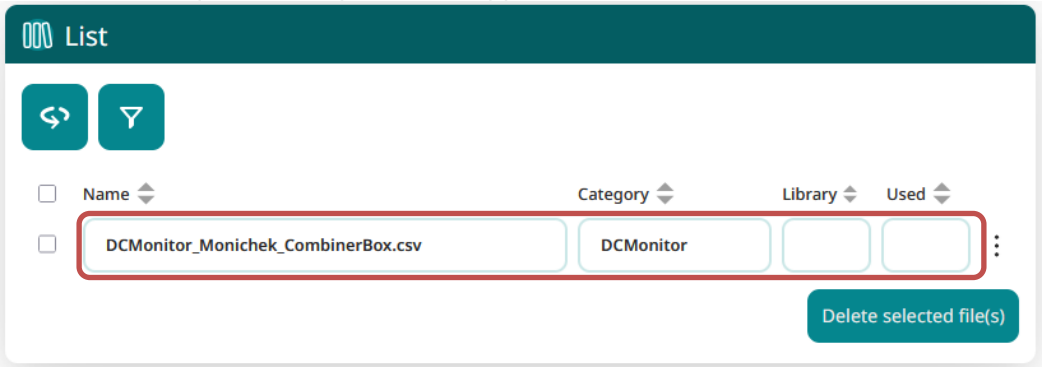
1. Click on the "Library file or definition file" field of the "Import a device library or definition file" parameter. A window will open allowing you to select the file to be imported,
2. Select the definition file that includes a ".csv" extension and click the "Open" button.



3. Press the "Load" button.
4. Follow the progress of the update next to the "Load" button:



5. Please wait while the definition file is imported.
6. At the end of the import, the imported file appears in the "List" section.



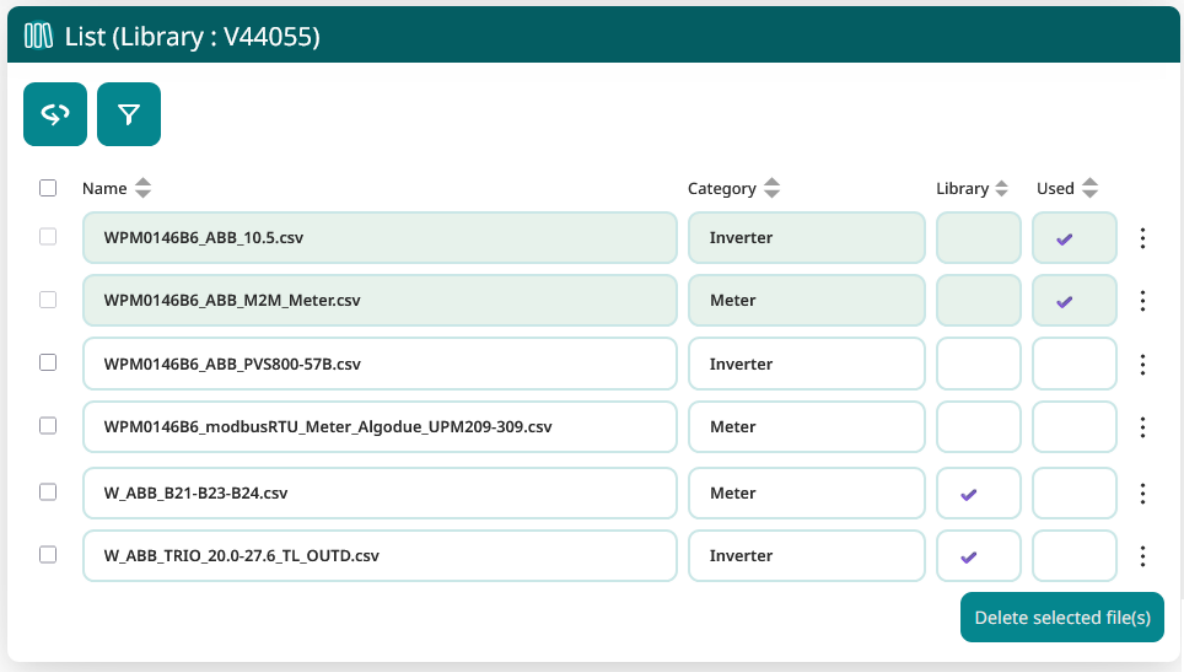
When you import a definition file, if it already has the same name and/or header (first line of the file) as a file in the concentrator:

- If this file is in use, the import is denied;
- Otherwise, the existing file will be replaced with the new one.

3.2.5.3.2 List

The "List" part allows you to manage all the definition files present in the concentrator. It includes two buttons, one to refresh the list, the other, a filter to facilitate the search for definition files, and then the display of the definition list, which consists of six columns which are:

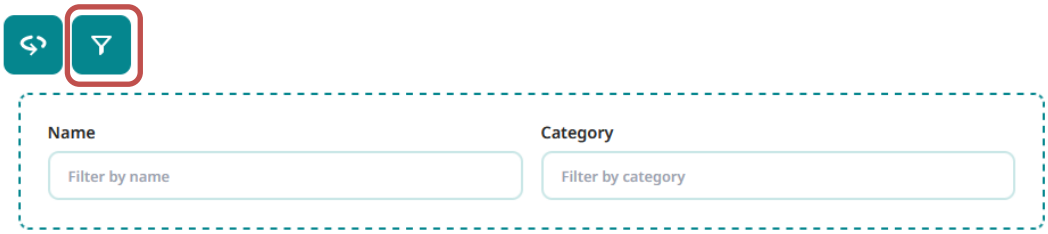
- The selection of the file allows multiple deletion with a simple press on the "Delete selected file(s)" button at the bottom of the page after selection.
- The name of the definition file.
- The category of which the definition file is a part.
- Information about whether the file is part of the library.
- Information about whether the file is currently in use by a device. In order to increase visibility, a green highlight is also made on the definition files used by a device.
- The last column allows you to perform specific actions on the definition file.



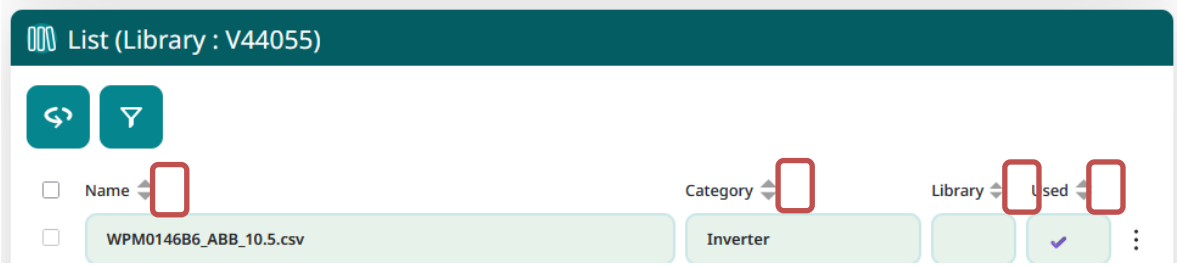
The WebdynSunPM device that contains all the I/O of the WebdynSunPM is not part of the list of definition files on the "Definition File Library" page. This equipment cannot be deleted and can only be modified by its specific page in "Monitoring" and then "Device".

When a device uses a library definition file, it is automatically renamed. The library file "W_<Manufacturer>_<Model>.csv" then becomes "<UID>_<Manufacturer>_<Model>.csv".
If the device is deleted, then the definition file takes over the one from the library.

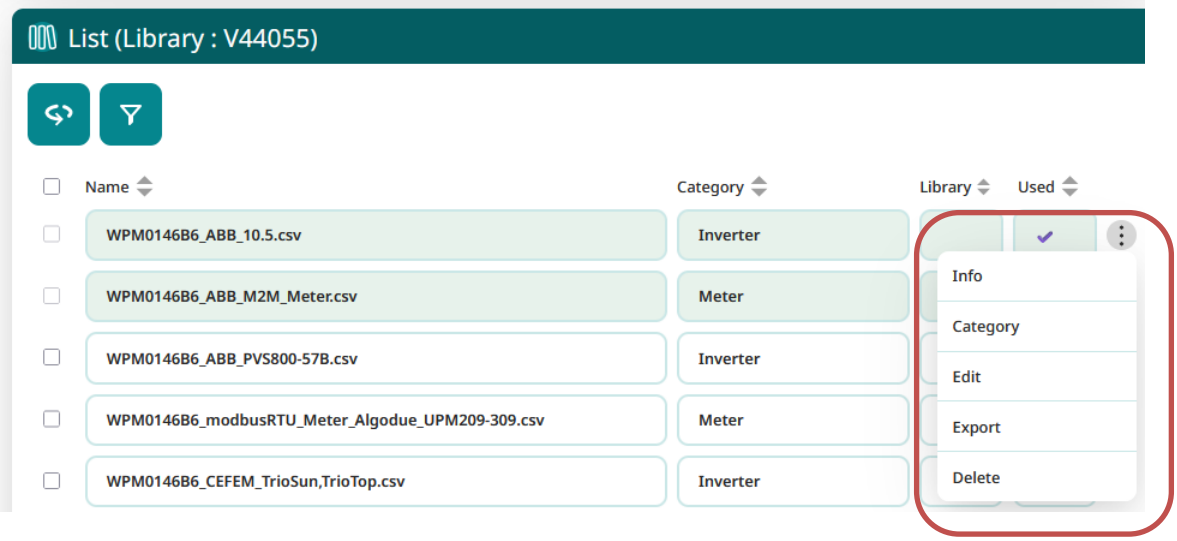
Definition files can be searched in the list by name and/or category. Pressing the Filter icon again disables filtering.



A sorting is also available, just click on the sorting icon to the left of the name of the column you want to sort.



Specific action on the definition file:



All possible actions on definition files are:

- **Info**: Lists all devices that use this definition file. (Only if the file is not part of the library)
- **Category**: Allows you to change the category of the definition file. (Only if the file is not part of the library)
- **Edit**: Allows you to edit part of the content of the definition file. (Only if the file is not part of the library)
- **Export**: Allows you to download the definition file.
- **Delete**: Allows you to delete the hub definition file.

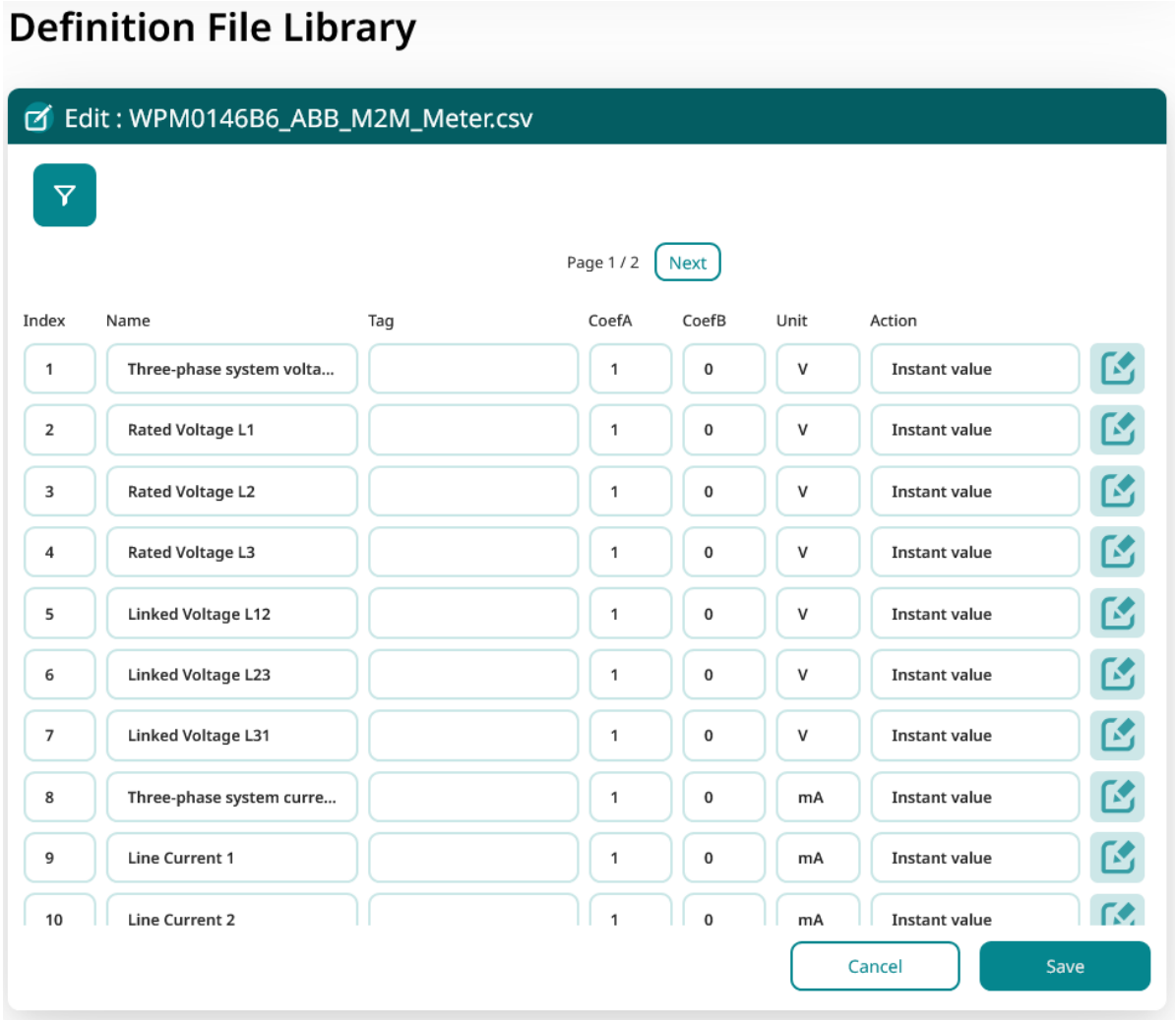
Editing the definition file:

The definition file is represented in a table format and consists of eight columns:

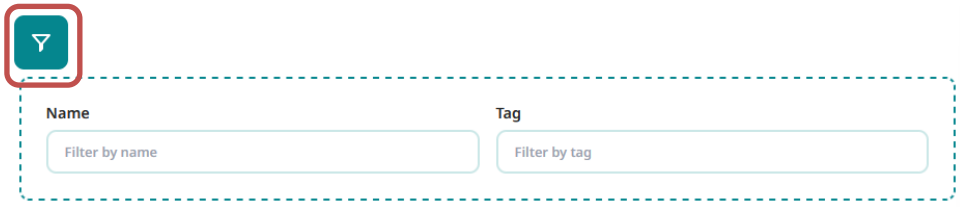
- **Index**: Data index.
- **Name**: The name of the data.
- **Tag**: A tag of the data used to easily locate a variable or parameter and for use in Lua scripts.
- **CoefA**: Coefficient A of the data (See chapter 3.1.2.2.2 "Definition file content")
- **CoefB**: Coefficient B of the data (See chapter 3.1.2.2.2 "Definition file content")
- **Unit**: Unit of data

- **Action**: Action defined for the data (See chapter 3.1.2.2.2 "Definition file content")

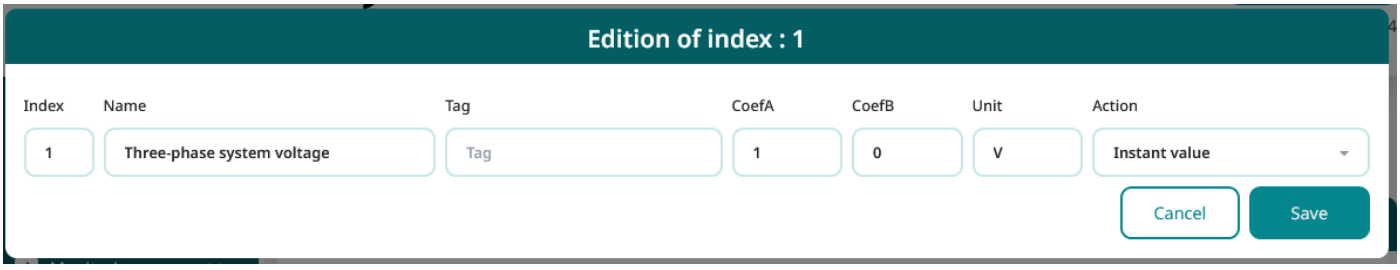
"Edit" button allows editing of the data row.



It is possible to search data by name and/or tag by clicking on the Filter icon. Pressing the Filter icon again disables filtering.



Pressing the "Edit" button opens the following page:

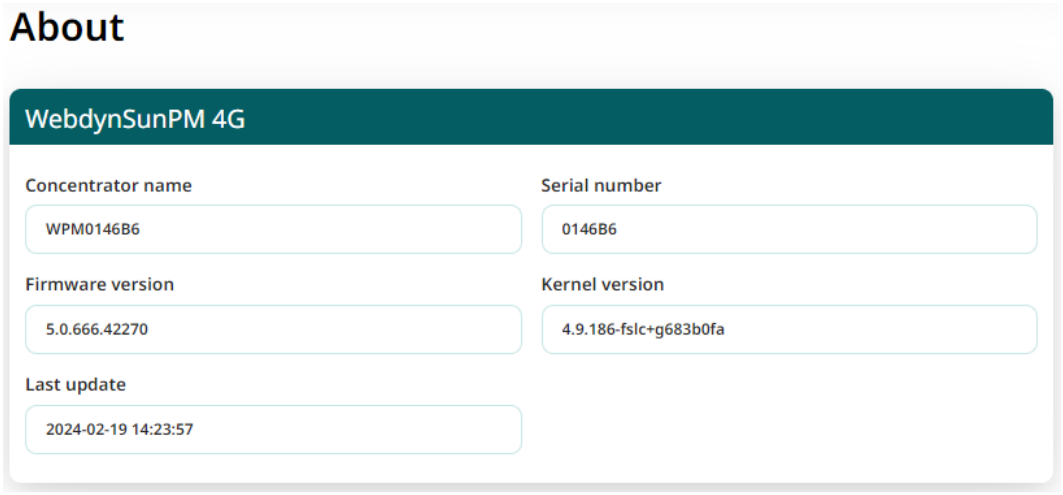


Each column displayed in the table representing the definition file can be modified here.

3.2.5.4 About

The “About” page provides the following concentrator data:

- "Concentrator name”: Unique identifier for the concentrator (*see section 3.2.5.1.1): “Updating and identifier”*)
- “Serial number”: Concentrator serial number corresponding to the last 6 characters of the concentrator MAC address.
- “Firmware version”: Concentrator software version
- "Kernel version”: Version of the Linux kernel used by the concentrator
- "Last update”: Date and time of the last update applied to the concentrator.
  (Format: YYYY- MM-DD hh:mm:ss)



3.3 Micro SD card

Configuration using a Micro SD card works in the same way as described previously for the FTP/SFTP/WebDAV server (see section 3.1: "FTP/SFTP/WebDAV”).

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

As the SD card directories are not configurable, the tree structure must use the following format:

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

- /CERT
- /DATA
- /CMD
- /DEF
- /SCRIPT

If the directories do not exist, they will be created on the SD card the next time there is a "connection" request.

If the concentrator 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 by the concentrator as an FTP server.

Note that the command files (CMD) on the SD card are not processed on the SD card by the concentrator.

Webdyn does not supply any SD cards. Contact a computer hardware retailer.
WebdynSunPM is compatible with micro SDXC cards (15 x 11 mm) of a capacity of up to 32 Gb.

3.4 Modbus slave TCP

The WebdynSunPM has a Modbus TCP slave function which is used to:

- access certain predefined information specific to WebdynSunPM (serial number, firmware version, etc.),
- access variables for the devices managed by the WebdynSunPM,
- access virtual device variables created by the scripts,
- run concentrator commands.

3.4.1 General operation

The Modbus slave can only be accessed using TCP at slave address "1" and is available on both Ethernet interfaces (LAN1 and LAN2). All the Modbus registers are in 16-bit “holding register” format. The Modbus slave registers are split into two parts:

- **The user registers**: these registers can be addressed from 0x0000 to 0x7FFF (0 to 32767). Their use is free and they can be configured using a definition file.
- **WebdynSunPM registers**: these registers are addressed in the 0x8000 to 0xFFFF (32768 to 65535) range. They are reserved and cannot be configured. They provide access to the WebdynSunPM information.

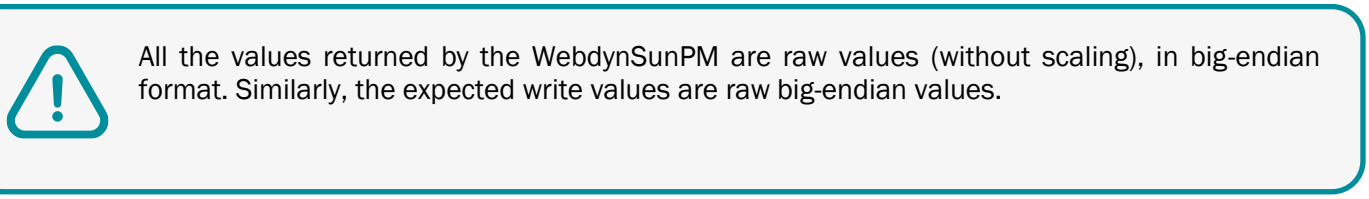
Each register can be read using the following Modbus functions:

- 0x03 (Read Holding Registers): read
- 0x06 (Write Single Register): single write
- 0x10 (Write Multiple Registers): multiple writes

A variable can be associated with a group of one or more adjacent registers.

When a variable is accessed, the behaviour is as follows:

- Variable read: the WebdynSunPM returns the last read value for the variable. This may include:
  - WebdynSunPM information
  - A variable linked to a connected physical device (for example: an inverter, an energy meter, etc.).
  - A virtual device variable created by a running Lua script. *(See Lua technical guide: "WebdynSunPM LUA User Guide.pdf")*
- Variable write: the WebdynSunPM modifies the variable value. This may include:
  - The WebdynSunPM relay output. This changes the relay output status.
  - Sending a command. (*See section5: "Commands"*)
  - A variable linked to a connected physical device (for example: an inverter, an energy meter, etc.). This action requires a request to the physical device.
  - A virtual device variable created by a running Lua script. *(See Lua technical guide: "WebdynSunPM LUA User Guide.pdf")*

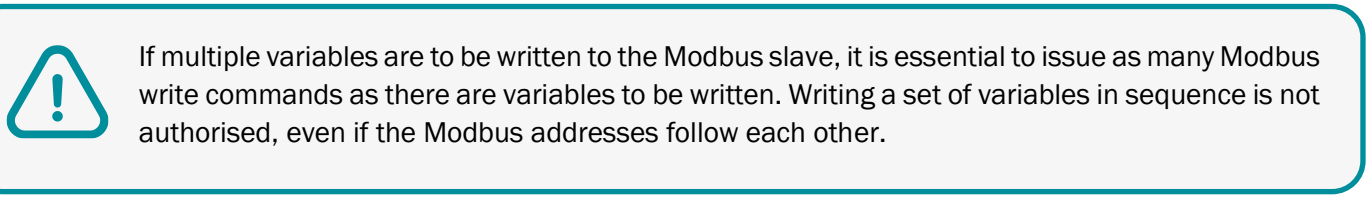


All the values returned by the WebdynSunPM are raw values (without scaling), in big-endian format. Similarly, the expected write values are raw big-endian values.

3.4.1.1 Reading or writing a variable

When a group of registers associated with a variable is read, the last known value of the variable is returned in a standard Modbus response.

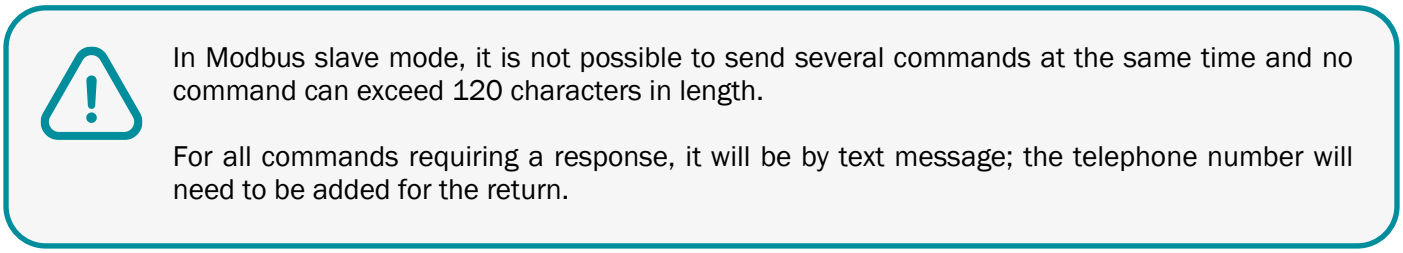
When a group of registers associated with a variable is written, the WebdynSunPM writes the new value to the device as soon as possible. The response is only returned to the client when the operation has been completed.



If multiple variables are to be written to the Modbus slave, it is essential to issue as many Modbus write commands as there are variables to be written. Writing a set of variables in sequence is not authorised, even if the Modbus addresses follow each other.

3.4.1.2 Running a command

To run a Modbus slave command, simply write a character string to register address 34000 (*see section 3.4.2.1: "Predefined Webdyn variables")*using the same syntax as for a text message command*(see section 5.2.3: "Text* message *")* and a maximum length of 120 characters.



In Modbus slave mode, it is not possible to send several commands at the same time and no command can exceed 120 characters in length.

For all commands requiring a response, it will be by text message; the telephone number will need to be added for the return.

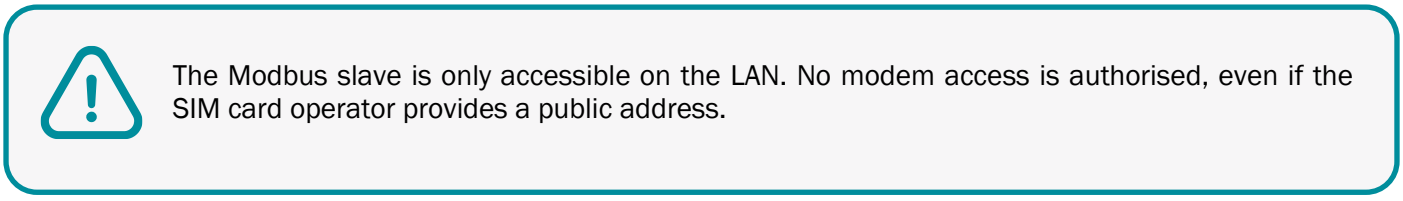
3.4.2 Configuration

The Modbus slave is configured using an entry in the "<UID>_daq.csv" file in the "/CONF" directory on server 1 (main server).

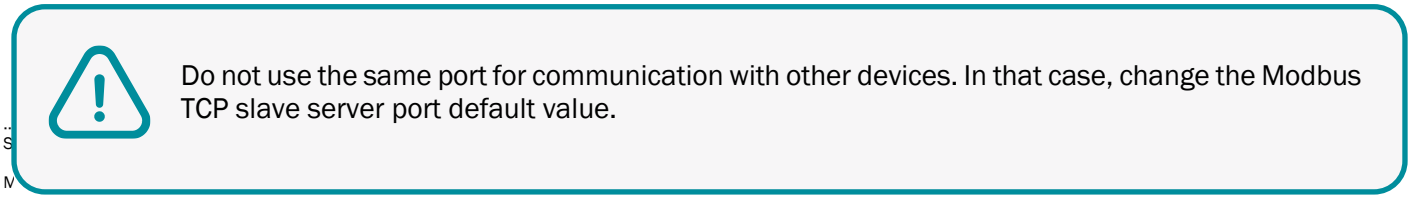
ModbusSlave:<enabled>;<port>;<mapping file>

Where:

- **enabled**: Modbus slave status:
  - 0⟶ Disabled,
  - 1⟶ Enabled.
- **port**: the TCP port used by the server. By default, port 502 is used.
- **mapping file**: name of the definition file describing the associations between the user registers and the device variables or Lua scripts. The file must be present in the "/DEF" definition directory. If the "mapping file" field is empty, then only the WebdynSunPM variables are accessible.



The Modbus slave is only accessible on the LAN. No modem access is authorised, even if the SIM card operator provides a public address.



Do not use the same port for communication with other devices. In that case, change the Modbus TCP slave server port default value.

S :

M

index;interface;name;address;acqPeriod(s);timeout(ms);serialNumber;parameters;category;model;defFile

Example of a file extract from "<UID>_daq.csv":

3.4.2.1 Predefined Webdyn variables

The predefined Webdyn variables are stored in the part of the WebdynSunPM registers addressed between 0x8000 and 0xFFFF (32,768 to 65,535). The WebdynSunPM variables cannot be configured and are used to access certain WebdynSunPM data.

The Webdyn variables are available at the following Modbus register addresses:

Address and number of registers (16-bit word)	Data type	Access	Description
33000 / 15	String	Read	Firmware version
33015 / 15	String	Read	Kernel version
33030 / 15	String	Read	Serial number
33045 / 15	String	Read	Identifier
33060 / 2	"_W" swap time_t(*See section 3.1.2.2.2: "Definition file content*")	Read	Last update date
33200 / 2	U32	Read	Digital input 1
33202 / 2	U32	Read	Digital input 2
33204 / 2	U32	Read	Digital input 3
33206 / 2	U32	Read	Analog input 1
33208 / 2	U32	Read	Analog input 2
33210 / 2	U32	Read	Analog input 3

33212 / 2	U32	Read	Analog input 4
33214 / 2	U32	Read/Write	Relay output
34000 / 120	String	Write	Commands This register group is special. It is not associated with a variable or item of data, but is used to run commands. See section 5.3: "Command list" for more details on running a command with Modbus.

3.4.2.2 User variables

The user variables are stored in the user register section, which can be addressed from 0x0000 to 0x7FFF (0 to 32767). The user registers must be configured using their own definition file in the "/DEF" definition directory configured on server 1 (main server). The definition file describes the associations between the user registers and the device variables or the Lua script variables.

For the definition file to be taken into account by the concentrator, it must be referenced in the "mapping file" field in the "ModbusSlave" part of the "<UID>_daq.csv" file located in the "/CONFIG" configuration directory. (*See section3.1.2.1.3.4: "Modbus slave configuration"*)

Only one definition file is possible for all the registers to be declared. The file name is free and can be modified by the client at will, the gateway will use the name provided in the "< UID> daq.csv file.".

The Modbus definition file is in csv format, it is composed of text rows each composed of ";" delimited fields.

ModbusSlave

The first row in the file contains the following information:

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

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

Each row fully describes a variable.

Each row will have the following format:

The field meanings are the following:

Field	Description
Index	Contains the unique associated identifier in the file. It is free form for the client as long as it remains unique.
Info1	Not used.
Info2	Address of the first register for this association and size in bytes. The address of the first register and the size in bytes are separated by an underscore "_". (by default, the size is two bytes, which corresponds to one register) The size in bytes must be a multiple of 2.
Info3	Not used.
Info4	Name of the device targeted by this association, i.e. either: • a device declared with a targeted variable • a Lua script with a virtual device variable created and targeted.
Name	Name of the variable targeted by this association. This field must not be used if the "Tag" field is used.
Tag	Tag for the variable targeted by this association. The "Tag" field must be identical to that of the variable in the target device definition file. This field must not be used if the "Name" field is used.
CoefA	Not used.
CoefB	Not used.
Unit	Not used.
Action	Contains the code describing the possible action types which are: • 0: variable disabled. The variable is not used. • 1: the targeted variable is read only. Write access is not authorised. • 4: the targeted variable is read-write.

	By default, if no value is entered, the action code is set to 1.

The definition file may be rejected if:

- the "Name" and "Tag" fields are both present,
- the groups of registers are then superimposed on the file,
- the entered register size is not a multiple of 2,
- the format is not respected.

As with the other definition files, the Modbus slave definition file is retrieved during an IS connection. It is then checked for consistency and may be rejected if an error is found.

In the example below, the variables for 2 different inverters and a variable created by a Lua script are retrieved. According to the following list:

1) Read of the temperature variable with the name "Temperature" in U16 format from inverter 1 (1 register i.e. 2 bytes).
2) Read of the power variable with the name "E-Total" in F32 format from inverter 1 (2 registers i.e. 4 bytes).
3) Read of the firmware version variable with the name "Firmware Version" in String format and a maximum length of 20 bytes from inverter 2 (i.e. 10 registers).
4) Read of the serial number variable with tag "DisplaySerialNumber" in String format and a maximum length of 24 bytes from inverter 2 (i.e. 12 registers).
5) Read and write of the "ValueRegulation" variable for the virtual device in F32 format created by the "regulation.lua" Lua script *(See the Lua technical guide: "WebdynSunPM LUA User Guide.pdf")* (4 registers or 8 bytes).

Definition file example:

```
ModbusSlave
1;;1_2;;INVERTER1;Temperature;;;1
2;;1_4;;INVERTER1;E-Total;;;;;1
3;;3_20;;INVERTER2;Firmware Version;;;;;1
4;;13_24;;INVERTER2;;DisplaySerialNumber;;;;1
5;;25_8;;regulation;ValueRegulation;;;;;4
```

3.4.3 Modbus error management

If an error occurs, the WebdynSunPM returns a Modbus exception code. The error cases are the following:

Code	Name	Description

0x01	Illegal Function	Function code not supported, i.e. other than 0x03, 0x06 or 0x10.
0x02	Illegal Data Address	Consistency problem between the request and the accessed registers: • A register is not configured. • A register is ignored (action code 0). • A register group is incomplete, i.e. fewer registers are requested than the group contains. • Attempt to write on more than one variable. • Attempt to write to a read-only variable (action code 1). • Attempt to read the command register.
0x03	Illegal Data Value	The entered value is illegal.
0x04	Server Device Failure	Invalid Modbus address, i.e. not equal to 1.
0x0A	Gateway Path Unavailable	When writing, unable to connect to a TCP device. This may be due to the network configuration.
0x0B	Gateway Target Device Failed to Respond	When writing: • A device has not responded (timeout). • A device has responded with an invalid frame (CRC error, incorrectly formed, etc.). • A device has unexpectedly closed its TCP socket.
0x70	Invalid Mapping	A variable of which the size is incompatible with the number of registers is read or written.
0x71	No Such a Variable	A device or variable is not known to the WebdynSunPM.
0x72	Invalid Action	Writing to the target variable is not supported (Input Register, U8, etc.).
> 0x80		During a write, if a device responds with a Modbus exception, it is transferred to the client by setting the most significant bit to 1, i.e. by performing the 0x80|code conversion.

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:

- **Storing the data and alarms collected locally by the concentrator:**

  Each time a connection is made to the server, whether by manual request, the triggering of an alarm (action code 8) or the triggering of the connection schedule, the concentrator uploads its stored data.

- **Configuring the concentrator:**

  At each connection, the concentrator synchronises its configuration with the specific file on the server. If the file does not exist, the concentrator creates it from its current configuration.

- **Triggering actions on the concentrator:**

  The command files must be uploaded to the server in an directory associated with the concentrator.

- **Monitoring the concentrator and assisting in troubleshooting:**

  The concentrator can upload log files for diagnosis purposes.

If only one of the two, servers is an FTPv/ 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: it 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 FTP (unsecure), SFTP and WebDAV-HTTPS servers operate identically. However, the use of an SFTP or WebDAV-HTTPS server is preferable because it has built-in security layers which are not present on a classic FTP server. The description in this section is equally valid for the different types of server.

Configuration:

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 21 for FTP, 22 for SFTP, 443 for 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 it.

Server tree structure:

The server must have a 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.

The server must have the following directories under the root directory:

Name	Rights	Description
/CONFIG	Read/ write	Contains the configuration files. The concentrator configuration uploaded by the concentrator is in the following format: <UID>_config.ini The connection interface configuration downloaded by the concentrator is in the following format: <UID>_daq.csv The connection schedule configuration downloaded by the concentrator is in the following format: <UID>_var.ini The connection schedule configuration downloaded 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

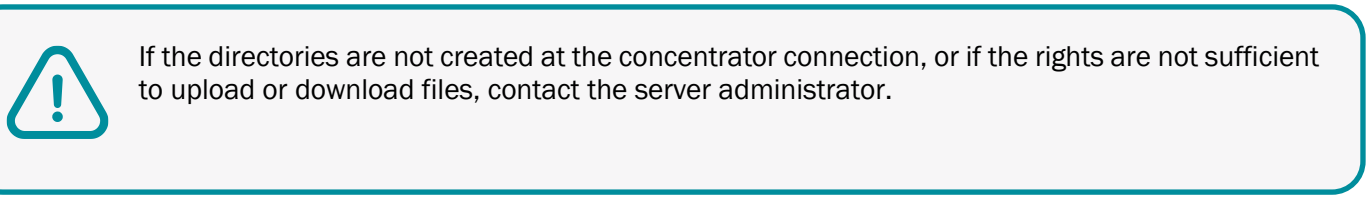
/LOG	Write	Contains the log and debug files. The log file has the following format: < UID>_LOG_<timestamp>.log.gz The debug file has the following format: < UID>_SYSTEM_<timestamp>.tar.gz
/BIN	Read	Contains the update files. The concentrator update has the following format: wgapp_<version>.spm

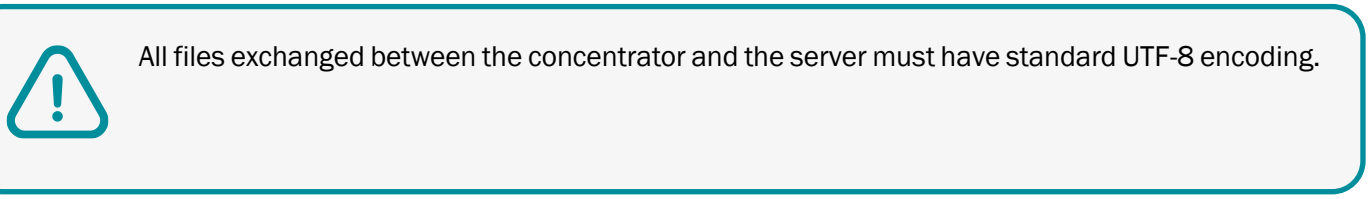
Where:

- <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 3.1.2.1.3.5: “Declaration of devices to be monitored”*)
- <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.





Operation:

in FTP or SFTP, if the “FTP_TwoStepsSendingDisabled” or “FTP2_TwoStepsSendingDisabled” parameter equals “0”, the concentrator uploads the files to the server using a 2 step process:

1. At the start of the transfer the file has an additional ".tmp” extension.
2. 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 unnecessary.

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	Lua language script
.pem	• Certificate file	PEM format certificate allowing connection to a secure server.
.log	• Log file (compressed)	Text file
.spm	• Update file	Package containing all the update files

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:

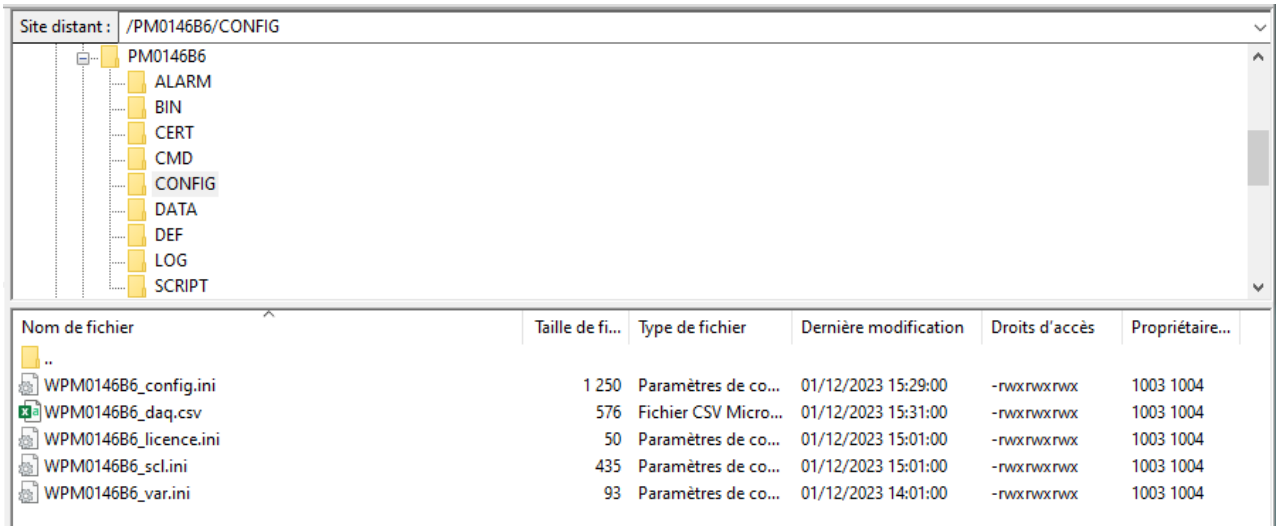
- <UID>_config.ini
- <UID>_daq.csv
- <UID>_licence.ini
- <UID>_var.ini
- <UID>_scl.ini

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. The configuration files must be uploaded or modified in this 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 shown in the concentrator’s LOG file.

If there 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 3.1.2.1: “Concentrator operation” or to “Appendix A: “_config.ini” configuration 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:

- The inputs/outputs: IO
- Customer remote information: CRI
- RTU/TCP Modbus
- Proprietary inverter protocols
- TCP Modbus slave

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”.

To create Modbus TCP slave definition files, see section 3.4: "Modbus slave TCP” or the application note.

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 3.1.2.1.3.5: “Declaration of devices to be monitored”*)
- <comment>: free user field

Examples:

WPM00C44F_SunSpec_inverter_SMA_Solar_Inverter_9301_ModbusTCP.csv

WPM00C44F_IO.csv

custom.csv

Refer to section 3.1.2.1.3.5: “Declaration of devices to be monitored” 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:
  - CRI
  - 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 3.2.3.2: “Devices”*) and regularly uploads it to a server (*see section 3.2.3.3: “Server”*) 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.

4.1.8.1 Input/Output (IO) header

The IO data file header is the following:

Colour code:

- Black: Static 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: Static 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

Data formatting is identical regardless of the device or interface.

The reported values are raw and must be linked to the configuration on the device or the interface in its definition file. The IS must interpret the data using the raw data file and the definition file to be able to apply factors A and B and the unit for each variable. (see section *3.1.2.2.2*: “*Definition file content*”)

The "action" field associated with each variable in the definition file is used 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	The instant value is reported.
6	The instant value is copied to the acquisition file and to the file created by the "getData" command.

7	The min, max and mean values are transferred to the acquisition file; the instant value is transferred to the file created by the "getData" command.
8	The instant value is reported and and alarm is generated every time the value changes.
9	The instant value is reported and and alarm is generated every time the value changes but is uploaded at the next connection.

The device or interface data file data is the following:

n

d

d

..

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 variable FTP_EuroDateFormat, FTP2_EuroDateFormat; HTTP_EuroDateFormat and HTTP2_EuroDateFormat in section 3.2.3.3: “Server”.
- 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 reads over this acquisition period of variable B. This information is only displayed if the "FTP_EnableAdvancedData" or "HTTP_EnableAdvancedData" parameter on server 1 or "FTP2_EnableAdvancedData" or "HTTP2_EnableAdvancedData" on server 2 is set to 1. This data can only be useful for Modbus and Inverter devices. For IO, CRI, virtual devices and 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 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 + alarm on change in 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	150	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	1											
ModbusTCP	WPM00C44F_SunSpec_inverter_SMA_Solar_Inverter_9301_ModbusTCP.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	05:50:00	102	105	103.00	12					

4.1.4 "ALARM" alarms

The alarms are uploaded in the form of CSV format files compressed to Gzip ".gz" format. They are uploaded to the "ALARM/" directory of the 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 that has changed
MODBUS	Definition file name + Index + Value	The value of an alarm type index that 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:

```
d
datetime_2;GATEWAY;info_2
...
datetime_Y;GATEWAY;info_X
```

Colour code:

- Black: Static text.
- Blue: Device-specific information or data.

datetime_Y: Alarm timestamp at trigger point Y. For the format, see variables FTP_EuroDateFormat, FTP2_EuroDateFormat; HTTP_EuroDateFormat and HTTP2_EuroDateFormat in section 3.2.3.3: “Server”.

Where:

```
datetime_1;AlarmSource_1;fileDefinitionName_1;nameEquipment_1;indexVariable_1;value_1
```

- info_X: Information on alarm X.

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

Colour code:

- Blue: Device-specific information or data.

datetime_2;AlarmSource_2;fileDefinitionName_2;nameEquipment_2indexVariable_2;value_2
...
datetime_Y;AlarmSource_X;fileDefinitionName_N;nameEquipment_N;indexVariable_A;variable_A_value_B

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 3.2.3.3: “Server”.
- AlarmSource: 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 3.1.2.1.3.5: “Declaration of devices to be monitored”).
- 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 3.1.2.2.2: “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 can be used to run remote tasks on the concentrator. It is therefore possible to ask the concentrator to launch a device search, obtain the configuration.

Command files are not managed when the SD card is in use.

The command file operation (format and processing) is described in section 5: “Commands”. The available commands are detailed in 5.2.1: “Command file”.

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 client keys
- ".luaw": an encrypted LUA script with a Webdyn licence

The script file format is as follows:

<comment>_.lua

<comment>_.luax

<scriptwebdyn>_.luaw

Where:

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

Examples:

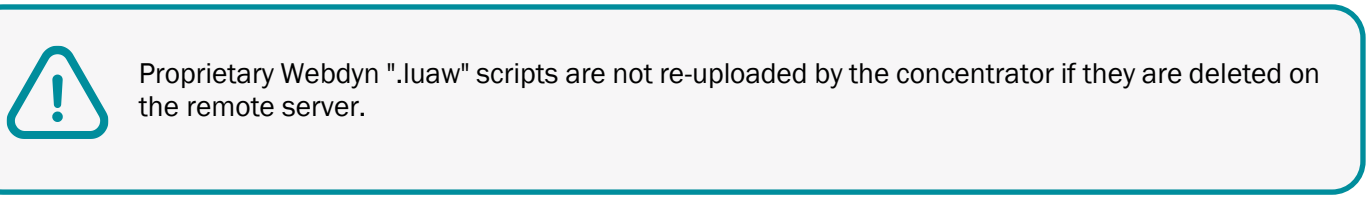
ControlPower.lua

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/



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 name format is as follows:

<UID>_wgapp_x.x.x.xxxxx.spm

Where:

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

Example:

wgapp_3.2.9.34734.spm

To, apply the update, use the procedure described in section 6.2: “Using FTP/SFTP/WebDAV”. Once the update has been applied, the firmware file can be deleted.

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.

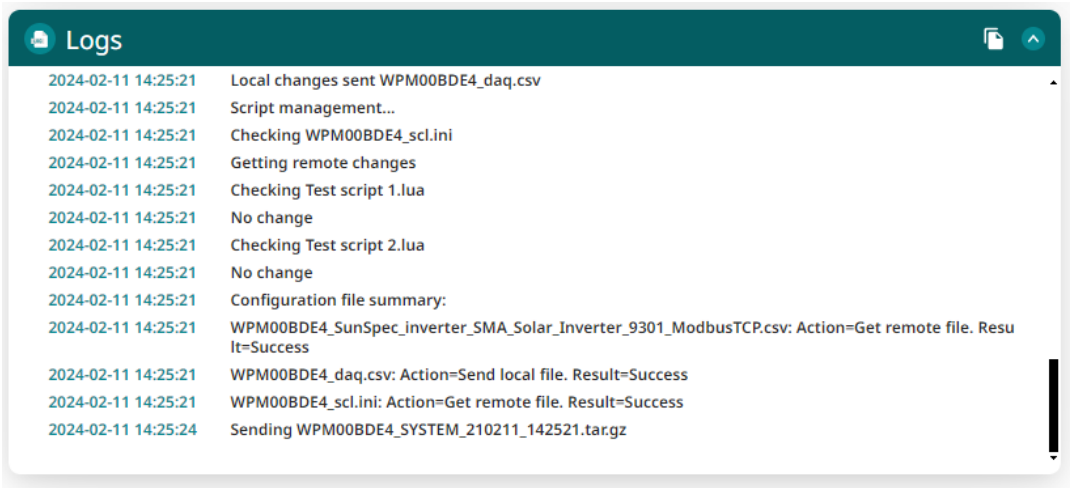
There are 4 types of log file:

- 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.
- Troubleshooting logs: “< UID> _”interface”_”date”.log.gz.

4.1.8.1 Connection logs

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

When the connection is forced using the local web interface, the connection sequence is shown in the "Logs" part.



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

```
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
1
2
2
2
2
2
2
2
2 4-02-11 14:25:21:No change
2 4-02-11 14:25:21:Configuration file summary:
2 4-02-11 14:25:21:WPM00BDE4_SunSpec_inverter_SMA_Solar_Inverter_9301_ModbusTCP.csv: Action=Get remote file. Result=Success
2024-02-11 14:25:21:WPM00BDE4_daq.csv: Action=Send local file.
Result=Success
2024-02-11 14:25:21:WPM00BDE4_scl.ini: Action=Get remote file.
Result=Success
2024-02-11 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.

```
2024-02-11 14:25:16:Alarms management...
```

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

```
2024-02-11 14:25:16:Data acquisition management...
2024-02-11 14:25:16:Sending WPM00BDE4_Modbus_210211_142503.csv.gz
2024-02-11 14:25:16:Data acquisition management...
2024-02-11 14:25:16:Sending WPM00BDE4_IO_210211_142507.csv.gz
2024-02-11 14:25:16:Sending WPM00BDE4_SUNSPEC_210211_142507.log.gz
```

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

```
2024-02-11 14:25:16:Configuration management...
```

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

```
2
2
2
2
2
2
2
2024-02-11 14:25:17:Checking WPM00BDE4_IO.csv
2024-02-11 14:25:17:No change
2024-02-11 14:25:17:Checking WPM00BDE4_SunSpec_inverter_SMA_Solar_Inverter_9301_ModbusTCP.csv
2024-02-11 14:25:17:Getting remote changes
2024-02-11 14:25:21:Failed to parse keep open parameter () for device
192.93.123.50, invalid value default value
2024-02-11 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:

```
2
2024-02-11 14:25:17:Checking WPM00BDE4_SunSpec_inverter_SMA_Solar_
Inverter_9301_ModbusTCP.csv
2024-02-11 14:25:17:Getting remote changes
```

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

```
2
```

The processing completes with script file management:

```
2   :4-02-11 14:25:21:Script management...
2024-02-11 14:25:21:Checking WPM00BDE4_scl.ini
2024-02-11 14:25:21:Getting remote changes
2024-02-11 14:25:21:Checking Test script 1.lua
2024-02-11 14:25:21:No change
2024-02-11 14:25:21:Checking Test script 2.lua
2024-02-11 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:

```
2024-02-11 14:25:21:Configuration file summary:
2024-02-11 14:25:21:WPM00BDE4_SunSpec_inverter_SMA_Solar_Inverter_9301_ModbusTCP.csv: Action=Get remote file. Result=Success
2024-02-11 14:25:21:WPM00BDE4_daq.csv: Action=Send local file.
Result=Success
2024-02-11 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 server.

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

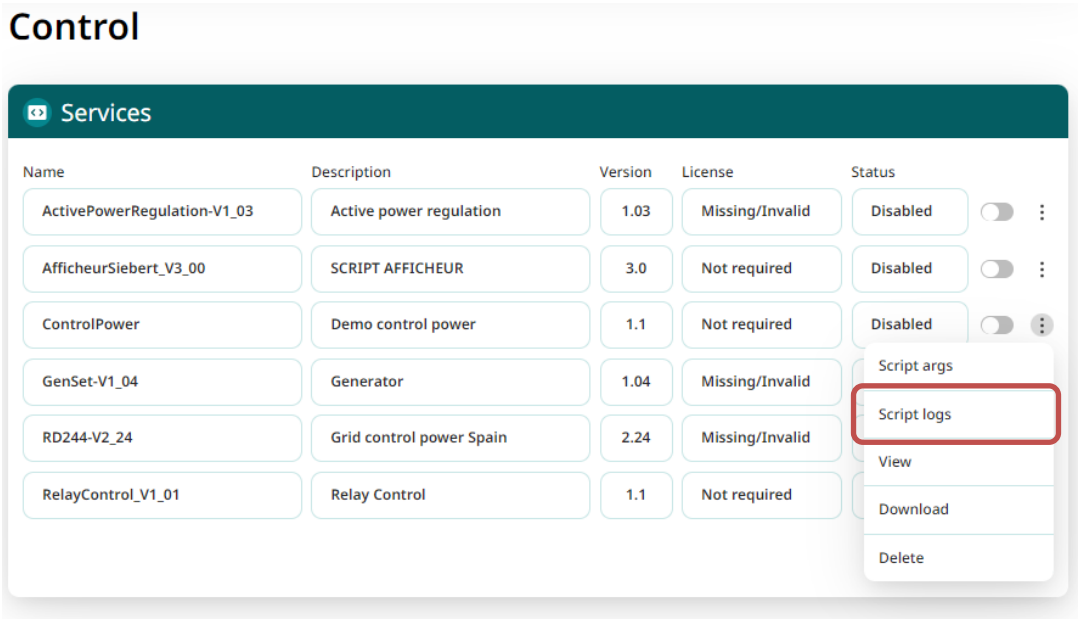
4.1.8.2 Script logs

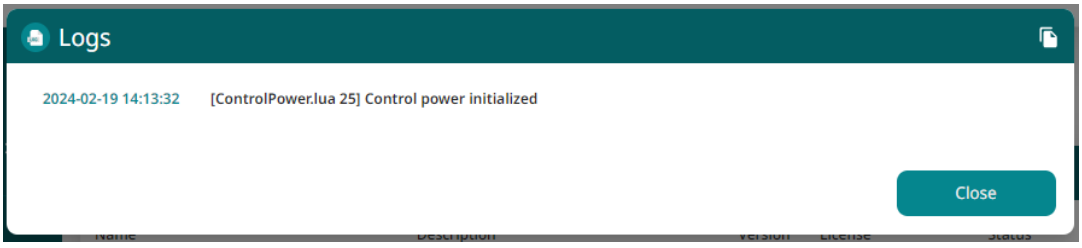
Each script runs in a separate environment. As a result, they each have an automatically generated log file when they run.

2024-02-11 14:25:24:Sending WPM00BDE4_SYSTEM_210211_142521.tar.gz

The log file can be viewed in 2 ways:

- Either directly on the local web site on the "Control page by clicking “Script logs” menu on the view service menu (see section 3.2.4.4: " View the service "):



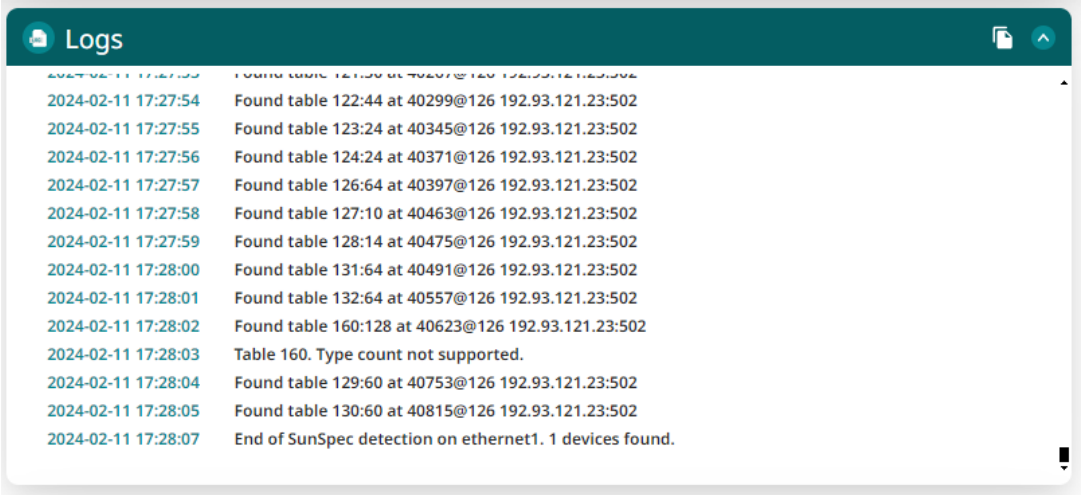


- Or on the server in the log files. The logs displayed below are also available on the server. They contain the “_LUA_” character string followed by the script name. For the log shown above, the file name will be “WPM00BDE4_LUA_Test script 1_210211_165930.log.gz” and will contain exactly what is displayed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



This log can also be found in the “LOG” 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:

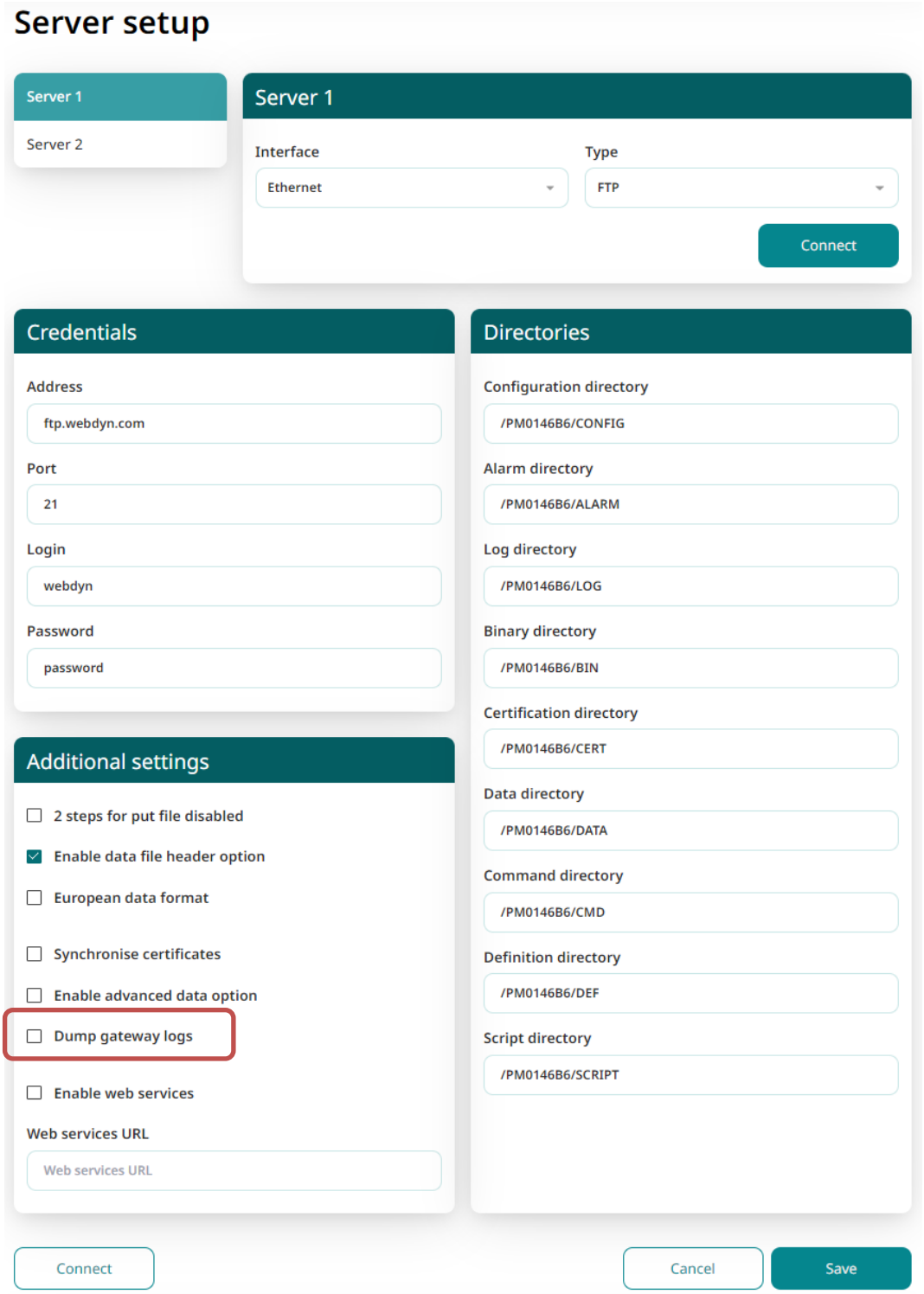
```
2
2
2
2
2
2
2
2   02/11-17:27:50;Found table 12:98 at 40087@126 192.93.121.23:502
2   02/11-17:27:51;Found table 101:50 at 40187@126 192.93.121.23:502
2   02/11-17:27:52;Found table 120:26 at 40239@126 192.93.121.23:502
2   02/11-17:27:53;Found table 121:30 at 40267@126 192.93.121.23:502
24/02/11-17:27:54;Found table 122:44 at 40299@126 192.93.121.23:502
24/02/11-17:27:55;Found table 123:24 at 40345@126 192.93.121.23:502
24/02/11-17:27:56;Found table 124:24 at 40371@126 192.93.121.23:502
24/02/11-17:27:57;Found table 126:64 at 40397@126 192.93.121.23:502
24/02/11-17:27:58;Found table 127:10 at 40463@126 192.93.121.23:502
24/02/11-17:27:59;Found table 128:14 at 40475@126 192.93.121.23:502
24/02/11-17:28:00;Found table 131:64 at 40491@126 192.93.121.23:502
24/02/11-17:28:01;Found table 132:64 at 40557@126 192.93.121.23:502
24/02/11-17:28:02;Found table 160:128 at 40623@126 192.93.121.23:502
24/02/11-17:28:03;Table 160. Type count not supported.
24/02/11-17:28:04;Found table 129:60 at 40753@126 192.93.121.23:502
24/02/11-17:28:05;Found table 130:60 at 40815@126 192.93.121.23:502
24/02/11-17:28:07;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 server. To do so, the “Dump gateway logs” option must be enabled on the server configuration page:



The system log files are also transferred automatically when a connection is forced from the server configuration page by clicking the “Connect” button or using a “connect” text message command.

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 used to configure 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 a call to a web service:

- Starting the product
- Data file upload to the FTP server
- Modification of the configuration (configuration load or upload)
- Running a command 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=additional data&RC=0&RC-COMP=
```

The "confirm.php" page is called at the configured URL. This page will receive the various action information and must process it.

The information is:

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

Action	Web service data
First FTP connection since the concentrator was started. When the gateway connects for the first time since its start-up (power-up, software update, etc.), the software version number is sent.	NSITE=IDsite&ACTION=VERSION&ACTION-COMP=firmware version&RC=0&RC-COMP=
Data sent from inverters or IOs to the FTP server. When the gateway connects to the FTP server to upload data files from the various devices, this web service is called.	NSITE=IDsite&ACTION=UPLOADDATA&RC=0&RC-COMP=
Alarms sent to the FTP server.	NSITE=IDsite&ACTION=UPLOADALARM&RC=0&RC-COMP=

When an alarm has been detected, an alarm file is uploaded to the FTP server and this web service is called.	
Gateway configuration files sent 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=list of configuration files&RC=0&RC-COMP= The configuration file names are separated by the ";" character. The files concerned are "_config.ini", "_var.ini" and "_daq.ini"
Configuration files loaded from the FTP server to the gateway. When changes are made to remote configuration files, the gateway loads these files on the next connection and this web service is called.	NSITE=IDsite&ACTION=CONFIGGLOBAL&ACTION-COMP=list of configuration files&RC=0&RC-COMP= The configuration file names are separated by the ";" character. The files concerned are "_config.ini", "_var.ini" and "_daq.ini"
Sending of one or more definition files. When the gateway modifies definition files (e.g. creation by detection), 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=list of definition files&RC=0&RC-COMP= The definition file names are separated by the ";" character.
Receipt of one or more definition files. When definition files are modified on the remote FTP server, the gateway loads them the next time it connects. This web service is then called.	NSITE=IDsite&ACTION=CONFIGDEF&ACTION-COMP=list of definition files&RC=0&RC-COMP= The definition file names are separated by the ";" character.
Running a command file. When the gateway detects a command file to be loaded and run 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 loads a valid firmware file (correct CRC) on 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	Action code unknown
12	Unknown RC received
13	Missing MAC address
-1	Internal server error

Examples of web service requests:

Load a new firmware version:

First connection since last start-up:

Load IDSite_config.ini and IDsite_var.ini configuration files:

URL/confirm.php;NSITE=IDsite&ACTION=VERSION&ACTION-COMP=WebdynSunPM4.2.3.38295&RC=0&RC-COMP=

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

MQTT and MQTTS servers operate identically. The use of an MQTTS server is preferable because of the extra security layers compared to a MQTT server. The description in this section is equally valid for both types of MQTT server.

Configuration:

The MQTT server is defined by the following parameters:

- **An address**: This 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 selected MQTT server type. Identification can be either by a simple identifier and password or by certificates and a key to be imported into the concentrator.
- **An application identification**: Unique server identifier allowing to have your own application space.
- **Topics**: Name of the information channels on the server to upload data and alarms.

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

- Data upload,
- Alarm uploads,
- Command reception,
- Concentrator update.

From an MQTT/MQTTS server, you cannot:

- Configure the concentrator,
- Create, add or modify a definition file,
- Create, add or modify a script,

- Add or replace a certificate,
- Upload the logs.

To carry out those actions, you must configure the second server on the concentrator using an FTP/SFTP server.

Server tree structure:

An MQTT/MQTTS server has information channels called topics. You must enter the same topic name between the MQTT/MQTTS server and the concentrator.

A topic must be entered the data so that the concentrator can upload it.

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

If you want to receive commands from the MQTT/MQTTS server, you must enter a topic for the commands and another for the command results.

Operation:

The concentrator connects and uploads data to the MQTT/MQTTS server at each scheduled time. (*See section3.2.3.3.8: "Schedule"*)

If Alarm and/or Command topics are entered, the concentrator is permanently connected to the MQTT/MQTTS server so that action can be taken immediately.

Unlike the FTP/SFTP server, the data uploaded to the MQTT/MQTTS server is formatted and takes into account the defined A and B factors and the unit defined for each variable (*see section 3.1.2.2.2: "Definition file content"*).

4.2.1 Data format

Data formatting is identical regardless of the device or interface.

The reported values are interpreted and use the A and B factors defined for each variable in the device or interface definition file. (see section *3.1.2.2.2*: "*Definition file content*").

The "action" field value for the device or the interface defined in its definition file is used for the following action:

"Action" code	Description
0	No values are reported.
1	The parameter value is reported.
2	The min, max and mean values are reported.
4	The instant value is reported.

6	The instant value is copied to the acquisition file and to the file created by the "getData" command.
7	The min, max and mean values are transferred to the acquisition file; the instant value is transferred to the file created by the "getData" command.
8	The instant value is reported and and alarm is generated every time the value changes.
9	The instant value is reported and and alarm is generated every time the value changes but is uploaded at the next connection.

The device or interface data file data is in JSON format as per:

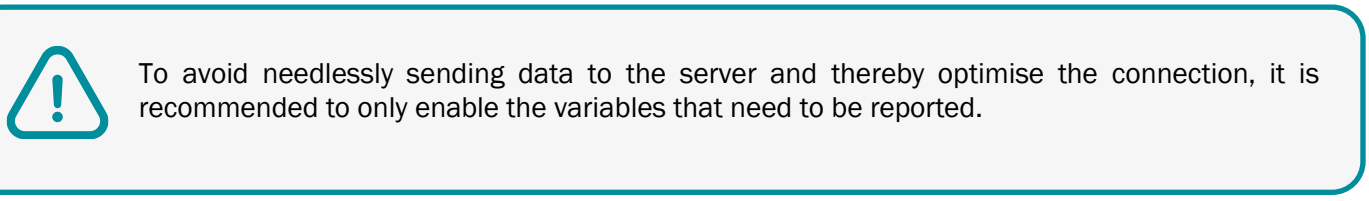

```
{
                "timestamp":value_timestamp_1_eqp_1,
                "nbOfRefreshes":value_nbOfRefreshes_1_eqp_1
        },
        {
                "DEF_Name_var_1_eqp_1":var_1_value_Z_eqp_1,
                "DEF_Name_var_2_eqp_1":var_2_value_Z_eqp_1,
                "DEF_Name_var_X_eqp_1":var_X_value_Z_eqp_1,
                "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
        }
        ]
}
```

Colour code:

- Green: Device or interface name or variable.
- Blue: Device or interface-specific data.
- Black: Static text.
- Orange: Concentrator Information

Where:

- DAQ_Name_eqp_N: Device name N, device "Name" field in the<UID>_daq.csv file (*see section 3.1.2.1.3.5: "Declaration of devices to be* monitored*")*
- DEF_Name_var_X_eqp_N: X variable name X on device N, variable "Name" field in the definition file (*(see section 3.1.2.2.2: "Definition file content"*)
- var_X_value_Z_eqp_N: Value Z of variable X of device N collected at acquisition point Y and the action defined in the device N definition file.
- **date**: Time-stamping of data at acquisition point Y in UTC+timezone (see the "NTP_TimeZone" parameter in Appendix A). In format: "YY/MM/DD-hh:mm:ss" for "Year/Month/Day/Hour:Minutes:Seconds"
- **timestamp**: Data timestamp at acquisition point Y in UTC+0 format. Number of milliseconds elapsed since 1st January 1970.
- **nbOfRefreshes**: Number of complete reads during this acquisition period. This information is only displayed if the "MQTT_EnableAdvancedData" parameter is set to 1. This data can only be useful for Modbus and Inverter devices. For IO, CRI, virtual devices and the parameter collection file, the number of refreshes is always 0.
- __MetaData__: Metadata on the context of the device in the file. Currently, the only information sent is the WebdynSunPM identifier in the "id" item.



Devices (Modbus, inverters):

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

- Device 1 index configuration (ModbusTCP):

Index	"Action" code	Display

1	1	Parameter value
2	0	None
3-11	4	Instant value
12	8	Instant value + alarm on change in value

- Device 2 index configuration (SMANET):

Index	"Action" code	Display
1-2	2	Min, max and average values

- MQTT data file in JSON format:


```
{
    "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
    }
```

```
                "timestamp":1612515600000,
                "nbOfRefreshes":2
        }
        ]
}
```


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 + alarm on change in value
8	4	Instant value


```
{
        "__MetaData__":{
                "id":"WPxxxxxx"
```


- MQTT data file in JSON format:

```
    }
}
```

4.2.2 Alarms

For the concentrator to upload alarms to the MQTT/MQTTS server, the Alarm topic must have been entered. The concentrator remains permanently connected to the MQTT/MQTTS server so that the action can be taken immediately.

When an alarm occurs, no data other than the alarm data is stored on the server. 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 that has changed

MODBUS	Definition file name + Index + Value	The value of an alarm type index that 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 in JSON format alarm Data format is the following:

```
{
                "timestamp":value_timestamp_alarmsDevice
        }
}
```

Colour code:

- Blue: Data depending on the alarm source.
- Black: Static text.
- Orange: Concentrator Information

Where:

- **alarms**: Alarm from an IO or Modbus source.
- **alarmsDevice**: Alarm from GATEWAY.
- **defName_alarms**: The data file name for the device or interface that triggered the alarm (*See section4.1.4: “ “ALARM” alarms*”)
- **deviceName_alarms**: Name of the device or interface that triggered the alarm, device "Name" field in the<UID>_daq.csv file *(see section 3.1.2.1.3.5: “Declaration of devices to be* monitored*”)*
- **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: Alarm type (GATEWAY).
- info_alarmsDevice: Alarm information (see "Info" column in the above table)
- **date**: Alarm timestamp in UTC+timezone format (see the "NTP_TimeZone" parameter in Appendix A). In format: "YY/MM/DD-hh:mm:ss" for "Year/Month/Day/Hour:Minutes:Seconds"
- **timestamp**: Alarm time stamp in UTC+0 format. Number of milliseconds elapsed since 1st January 1970.
- __MetaData__: Metadata on the context of the device in the file. Currently, the only information sent is the WebdynSunPM identifier in the "id" item.

Example of an alarm on a device:


```
{
                "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

Commands can be sent to the WebdynSunPM concentrator by the MQTT/MQTTS server.

To do that, the "Command" and "Result" topics must have been entered in the concentrator configuration *(see section 3.2.3.3.4: "MQTT")*.

When a command is published in the "Command" topic on the MQTT/MQTTS server, it is retrieved by the concentrator. The command is run by the concentrator and the command result is published in 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:

```
{
        "interface":interface_value,
}
```

Colour code:

- Blue: Information about the firmware or server.
- Black: Static text.

Where:

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

Example of an update for an HTTPS server:

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

Example of an update for an SFTP server:

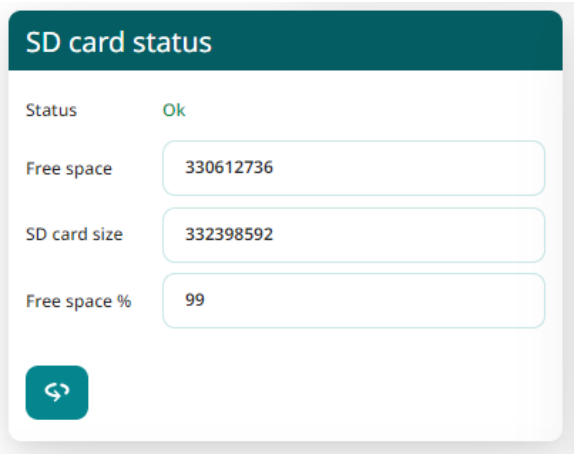

```
}
{
                checksum : 78a37fa7fb714876be7d08d0c39a067b ,
                "interface":"ethernet",
        },
        "callerId":"668"
}
```


4.3 MicroSD card

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

The SD card is only read when the concentrator needs to access it. That means that the rest of the time it can be removed without any consequences. The information is written securely as it is received, so that the card can be safely removed at all times.

The SD card use level can be viewed at all times on the server configuration page in the box for the SD card:



Webdyn does not supply any SD cards. Contact a computer hardware retailer.

5 Commands

5.1 Principle

Commands can be sent to the WebdynSunPM. They make it possible to run remote tasks for configuration, control or monitoring purposes. For example: launch a device search, obtain the current configuration, trigger a connection to the IS, etc. The same mechanism can also be used to invoke a function in a script installed on the concentrator.

5.2 Operation

A command can be sent using three different methods:

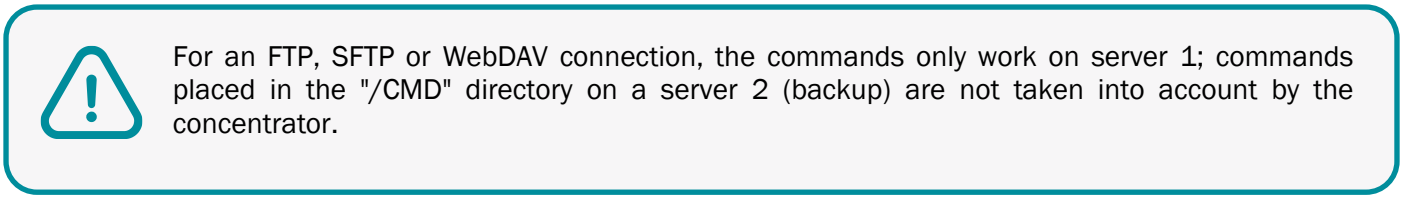
- A command file uploaded to the server (FTP, SFTP or WebDAV) which will be retrieved by the concentrator during the IS connection.
- An MQTT message posted on the WebdynSunPM control topic.
- A text message sent to the WebdynSunPM SIM card.

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 that purpose ("/CMD" by default). The file name must be <UID>_cmd.json where <UID> is the gateway identifier.

The commands included in it are in the JSON format described below and are run in order. Calling script functions is also supported. The file is deleted after import so that the same command is not processed twice.

Command results are also written to files that will be saved at the next connection to the IS. A file can contain one or more results. These are named according to the <UID>_ACK_<timestamp>.json template.



For an FTP, SFTP or WebDAV connection, the commands only work on server 1; 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 script function calls is as follows:

```
[{
        "rpcName":"<script name>.<function name>",
        "parameters":{ <function parameters in json format> },
        "callerId":"<command identifier 1>"
},
{
        "rpcName":"<script name>.<function name>",
        "parameters":{ <function parameters in json format> },
        "callerId":"<command identifier 2>"
},
...
```

Properties:

- *rpcName*: Script name and function to be run in <script name>.<function name> format. For a command, use the script name sunpm, which is reserved for internal WebdynSunPM commands.
- *parameters*: Some functions and commands require additional parameters. Where that is not the case, 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 only contains one command, the square brackets are optional.

The result file format is the following:

```
[
},
{
},
..
]
```

Properties:

- *result*: If the function or command is successful, it returns a result in the JSON format contained in this field. If there is an error, this field is absent.
- *error*: If an error occurs, 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. You can thus associate this response with its original request.

Each table entry if for a different command result.

5.2.1.2 Example

Using the following script to illustrate function calls:

```
header = {
        version = 1.0,
        label = “Test”,
        name = “test”
}
--[[
        Test function
]]
function testFunction(parameters)
        wd.log(“question is “ .. parameters.text)
        local result = {
```

Command file used to call the function:

```
                value = 42

e

[

    “parameters”: {
            “text”:”what is the answer ?”
    },
    “callerId”:”3d9311ed-0076-4f28-ac59-a2debfa35b86”
}]
```

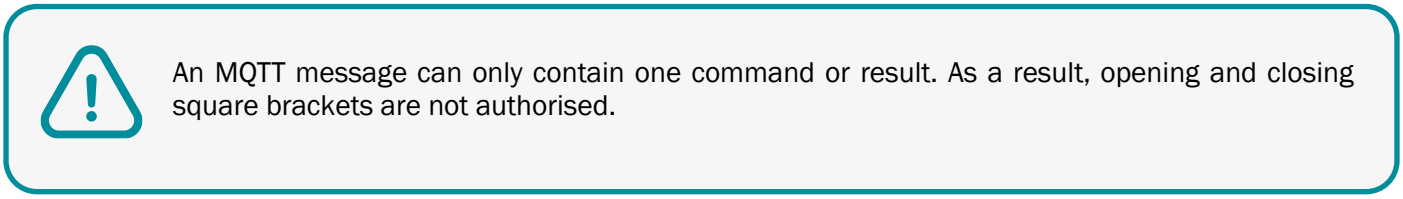
Result file obtained:

```
[

}
```

5.2.2 MQTT command message

WebdynSunPM can receive MQTT format command messages. Of course, that requires a configured MQTT server (see section 3.2.3.3.4: “MQTT”). The command and result topics must have been entered. WebdynSunPM subscribes to the command topic so that all messages sent to it are received and run. The result of a command or function call is published in the result topic.

The JSON format used for those messages is the JSON format described in section 5.2.1.1: “Command file JSON format”.




```
{
    "methodName":"<script name>.<function name>",
    "responseTimeoutInSeconds":<delay before timeout>,
    "payload":{
```

5.2.2.1 Use with Azure IoT

Azure has its own function call mechanism. The format JSON be used is therefore as follows:

Properties:

- *methodeName* replaces rpcName.
- *parameters* is contained in the Azure message payload field.
- *callerId* is ignored.

5.2.3 Text message

Text message commands can be sent to the WebdynSunPM modem. To do that, check that the modem is correctly configured (*see section 3.2.2.2: “Modem (Mobile)”*).

Text message commands do not use JSON format. Instead, the accepted format is as follows:

Several commands can be sequenced together by adding the ";" separator between each command.

If a command error occurs, a text message will be returned containing an explicit error message.

SMS encryption:

For security reasons, the SMS commands of the hub are always encrypted with the SMS password (See chapter **Erreur ! Source du renvoi introuvable.**: "**Erreur ! Source du renvoi introuvable.**"). It is necessary to encrypt SMS messages before sending an order to the hub. Responses sent by the hub are not encrypted and therefore do not require decoding.

To achieve encryption, the hub uses OpenSSL's AES-256-GCM+PBKDF2.

It is possible to easily carry out encrypted SMS commands using this web page available here:

WedbynSunPM - SMS Commands - Webdyn

For software integration, a python script designed by Webdyn for SMS encryption is also available. In this case, please contact Webdyn's sales or technical department who can provide you with it.

5.2.4 Modbus slave

Modbus slave commands can be run by writing a character string to register address 34000 (*see section 3.4.2.1: “Predefined Webdyn variables”*).

Modbus slave commands do not use JSON format and are the same as for text message commands. The format is the following:

In Modbus slave mode, it is not possible to send several commands at the same time and no command can exceed 120 characters in length.

For the "status" command, which requires a return, the return will be in a text message; the telephone number for the return must be added as a parameter.

If there is an error on a command sent by Modbus slave, no error messages will be returned.

5.3 Command list

List of the commands available on the concentrator:

Commands	Descriptions	Text message/ Modbus slave	MQTT/ MQTTS	Command file
connect	Connection trigger	✓	✓	
status	Hub status retrieval	✓	✓	✓
factory	Back to factory settings	✓	✓	✓
reboot	Concentrator reboot	✓	✓	✓
updateFirmware	Hub software update	✓	✓	✓
updateLibrary	Updating the library	✓	✓	✓
apn	Modem configuration	✓	✓	✓
ftp	FTP server configuration	✓	✓	✓
sftp	SFTP server configuration	✓	✓	✓
https	Webdav/HTTPS server configuration	✓	✓	✓
log	Activation of device communication logs	✓	✓	✓

setRelay	Relay status update	✓	✓	✓
discoverDevices	Device discovery	✓	✓	✓
getParameters	Collection of parameters, i.e. of variables defined with action code 1	✓	✓	✓
getData	Collection of variables defined with a 6 or 7 action code	✓	✓	✓
writeVariable	Write of a variable to a device	✓	✓	✓
setKey	Addition of client script decryption keys	✓	✓	✓
deleteKey	Removal of client script decryption keys	✓	✓	✓
startScript	Start a script	✓	✓	✓
stopScript	Stop a script	✓	✓	✓

If several command files are sent at the same time, the "factory" and "update" commands can result in the commands following them being lost. If there is an error on a previous command, the following commands will be run.

The available commands are described below with the expected parameters and the returned results.

5.3.1 "connect": Connection trigger

Asks the WebdynSunPM to start a connection to the server. That forces the settings to be synchronised with the server and all the log files to be uploaded.

connect=<connection>

The connection is launched immediately.

Parameters:

- connection: Number of the connection to be established. If no parameter is specified, the connection will be to server 1. The possible values are:
  - 1: Use server 1.
  - 2: Use server 2.

Returns:

- If successful:
  - For a text message/Modbus slave command: no return.
  - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.

Examples of text message/Modbus slave commands:

```
connect
```

or

```
connect=2
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{

}
```

Response:

```
{
        "callerId": "1",
        “result”:”OK”
}
```

5.3.2 “status”: Concentrator status retrieval

status Returns information about the current configuration.

By text message:

```
status;<number phone>
```

By Modbus slave:

```

```

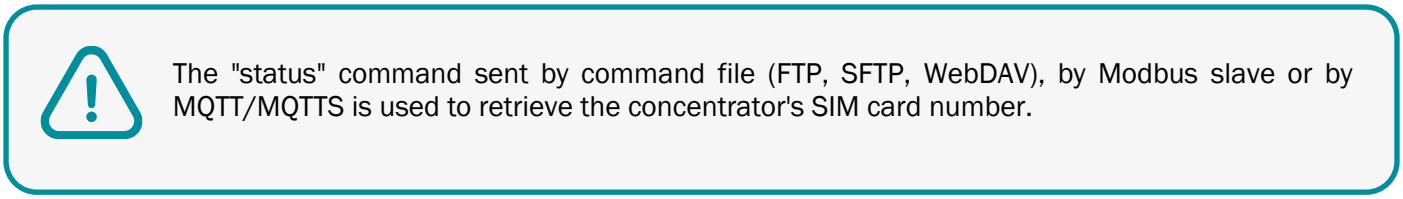
Parameters:

- By text message: no parameters.
- By FTP/SFTP/MQTT/MQTTS/WebDav and Modbus slave:
    - number: Telephone number that will receive the status text messages sent by the gateway. It is preferable to enter the number with the international dialling code.

Returns:

- If successful:
    - For a text message/Modbus slave command: no return.
    - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return

      A command acknowledgement file is sent to the server (FTP, SFTP, WebDAV, MQTT/MQTTS) that sent the command. The command information is sent to the entered telephone number.
- If an error is encountered: an explanatory message.



The "status" command sent by command file (FTP, SFTP, WebDAV), by Modbus slave or by MQTT/MQTTS is used to retrieve the concentrator's SIM card number.

Example text message command:

```
status
```

Example Modbus slave command with response to number "+33700000000":

```
status=+33700000000
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav) with response to number "+33700000000":

```
{
	"rpcName": "sunpm.status",
	“parameters”:{
		“number”: “+33700000000”
	"callerId": "1"
}
```

Response:

Text message response:

{
"callerId": "1",
"result": "OK"
}

The command information is returned in the form of a text message to the entered number of which the detail is the following:

```
type=WebdynSunPM 4G
idSite=WPM01194D
FW=5.0.2.41619
ModeServ1=modem
apn=tm
ipMobile=10.6.157.235
csq=26,99
network=4G
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: Web site name, i.e. the WebdynSunPM identifier
- FW: The firmware version currently loaded on the concentrator.
- ModeServ1: Server connection mode (Ethernet or modem).
- apn: The APN name configured for the modem.
- ipMobile: The modem IP address.
- csq: The modem signal strength.
- network: Network type used.
- ip: The IP address configured on the Ethernet 1 interface.
- mask: The subnet mask configured on the Ethernet 1 interface.
- gw: The gateway address used by the concentrator to connect to an external network on the Ethernet 1 interface.
- dns: The list of DNS servers used to resolve names for the Ethernet 1 interface. If several DNS servers are configured, they are delimited by the "/" character.
- type: Server 1 type. The possible values are:
  - ftp: The server is of the FTP type.
  - sftp: The server is of the SFTP type.
  - webdav: The server is of the WebDAV-HTTPS type.
  - mqtt: The server is of the MQTT type.

        - mqtts: The server is of the MQTTS type.
    - addr: Server address in login@server-address:server-port format.
    - last: The date and time the WebdynSunPM last connected to server 1 and the connection result in the following format: “DD/MM HH:MM <connection result>”.

5.3.3 “factory”: Back to factory settings

Reset the WebdynSunPM. The configuration files and the device acquisition data are deleted and the product is rebooted immediately.

Parameters:

factory

- No parameters.

Returns:

- If successful:
    - For a text message/Modbus slave/FTP/SFTP/MQTT/MQTTS/WebDav command: no return.
- If an error is encountered: an explanatory message.

Example text message/Modbus slave command:

```
factory
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{
    “rpcName”:”sunpm.factory”,
    "callerId": "1"
}
```

Response:

```
{
    "callerId": "1",
    “result”:”OK”
}
```

5.3.4 “reboot”: Concentrator reboot

WebdynSunPM reboot. The command runs immediately.

Parameters:

- No parameters.

reboot

Returns:

- If successful:
  - For a text message/Modbus slave/FTP/SFTP/MQTT/MQTTS/WebDav command: no return.
- If an error is encountered: an explanatory message.

Example text message/Modbus slave command:

```
reboot
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{
        "callerId": "1"
}
```

Response:

```
{
        "callerId": "1",
        “result”:”OK”
}
```

5.3.5 "updateFirmware": Concentrator software update

updatefirmware=<url>:<login>:<password>:<checksum>:<interface>

Retrieves firmware from a specified URL, validates it using a checksum and installs it. The command returns a successful result just before proceeding with the installation. The result of the installation itself must be checked by the usual version control methods.

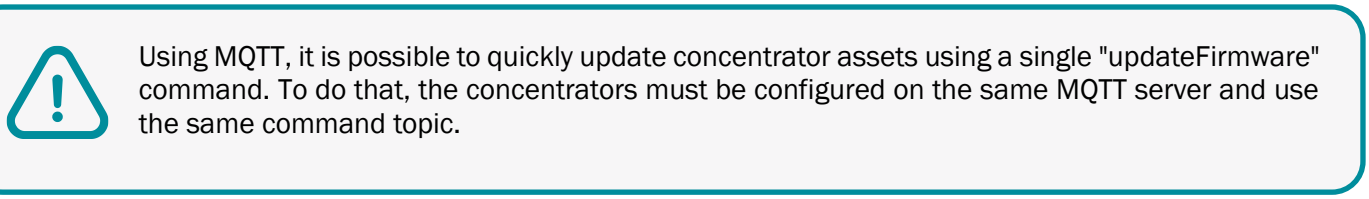
Parameters:

- url: URL of the file to be retrieved. The accepted protocols are HTTP, HTTPS, FTP and SFTP. The port can be indicated in address:port format.

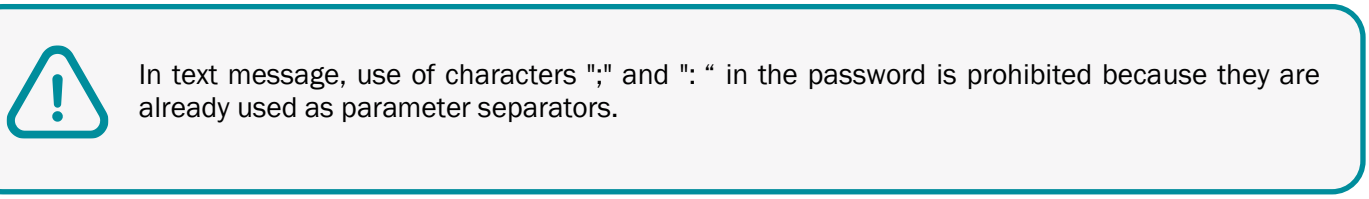
- login: Server login.
- password: Server password.
- checksum: MD5 checksum of the file to check its validity.
- Interface: Interface used for the connection: Ethernet or modem.

Returns:

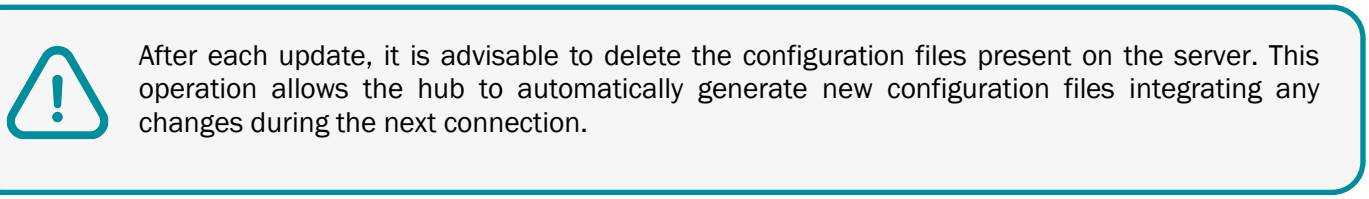
- If successful:
  - For a text message/Modbus slave command: no return.
  - For an FTP/SFTP/MQTT/MQTTS/WebDav command: the following message: "Firmware downloaded successfully. System will restart...".
- If an error is encountered: an explanatory message.



Using MQTT, it is possible to quickly update concentrator assets using a single "updateFirmware" command. To do that, the concentrators must be configured on the same MQTT server and use the same command topic.



In text message, use of characters ";" and ": “ in the password is prohibited because they are already used as parameter separators.



After each update, it is advisable to delete the configuration files present on the server. This operation allows the hub to automatically generate new configuration files integrating any changes during the next connection.

Example text message/Modbus slave command:

```
updatefirmware=ftp://ftp3.webdyn.com/wgapp_4.1.0.37427.spm:identifiant:webdyn:70a0eeeae295a7e16d3811b66bee9b66:modem
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):


```
{
	"rpcName":"sunpm.updateFirmware",
	"parameters":{
		"url":"https://www.webdyn.com/download/wgapp_new_fw.spm",
		"login":"identifier",
		"password":"webdyn",
		"checksum":"78a37fa7f671487c6be7d08dd6c39a067b",
```

Response:

```
}
```

5.3.6 "updateLibrary": Updating the library

```
{
    "callerId":"674",
    "status":"Webdyn SunPM library restart in progress"
}
```

Update of the WebdynSunPM library. This command executes immediately.

Parameters:

updateLibrary

No parameters.

Returns:

If successful:

For a slave SMS/Modbus order: no return.

For an FTP/SFTP/MQTT/MQTTS/WebDav command: the following message: "OK".

If an error is encountered: an explanatory message.

In MQTT, it is possible to quickly update a fleet of hubs with a single "updateLibrary" command. To do this, the hubs must be configured on the same MQTT server and use the same command topic.

Example SMS/Modbus slave command:

```
updateLibrary
```

```
{
    "rpcName":"sunpm.updateLibrary",
    "callerId":"674"
}
```

Example command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{
```

Answer:

```
    "callerId": "1",
    "result":"OK"
}
```

5.3.7. "apn": Modem configuration

Modem APN configuration. The APN is required to establish a 2G/3G mobile connection. For modem configuration, see section 3.2.2.2: "Modem (Mobile)" for more details.

Parameters:

```
apn=<apn>:<login>:<password>
```

- apn: The APN name to use for the modem connection.
- login: User required for authentication on certain APNs. (optional)
- password: Password required for authentication on certain APNs. (optional)

Returns:

- If successful:
  - For a text message/Modbus slave command: no return.
  - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.

Examples of text message/Modbus slave commands:

```
apn=m2mapn:login:webdyn
```

Or

```
apn=m2minternet
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{
    "rpcName": "sunpm.apn",
    "parameters":{
        "apn": "iot.1nce.net",
        "login":"",
        "password":""
    },
    "callerId": "1"
}
```

Response:

{
"callerId":"1",
"result":"OK"
}

5.3.3. "ftp": FTP/SFTP server configuration

FTP server configuration for "Server 1".

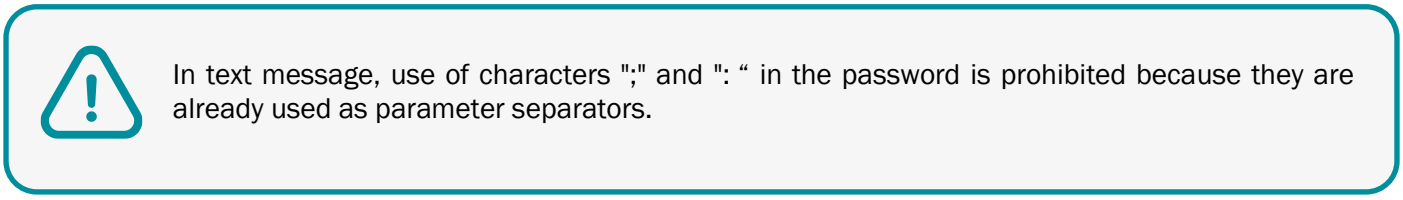
Parameters:

ftp=<server>:<login>:<password>:<port>:<interface>

- *server*: FTP server name to connect to. This parameter can be a name or an IP address.
- *login*: Login for the indicated FTP server.
- *password*: Password for the above login.
- *port*: FTP server port number. By default, FTP servers use port 21.
- *interface*: Connection type to use. The authorised values are:
  - *ethernet*: Uses the ethernet RJ45 connection to connect to the FTP server. The Ethernet interface must first have been configured for the connection to work (*See section 3.2.2.1: "Ethernet (Local)"*)
  - *modem*: Uses the modem to connect to the FTP server. The modem must have been configured first. Otherwise the *apn* command can be used to configure it by text message (*See section 5.3.7: ""apn": Modem configuration"*).

Returns:

- If successful:
  - For a text message/Modbus slave command: no return.
  - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.



In text message, use of characters ";" and ": " in the password is prohibited because they are already used as parameter separators.

Example text message/Modbus slave command:

```
ftp=ftp3.webdyn.com:login:pass
word:21:modem
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{
        "server": "ftp3.webdyn.com",
        "login":"login",
        "password":"password
        "port":21,
        "interface":"modem"
}
```

Response:

```
{
        "callerId": "1",
        “result”:”OK”
}
```

5.3.9 "sftp": SFTP server configuration

SFTP server configuration for "Server 1".

```
s  =<server>:<login>:<password>:<port>:<interface>
```

Parameters:

- *server*: SFTP server name to connect to. This parameter can be a name or an IP address.
- *login*: Login for the indicated SFTP server.
- *password*: Password for the above login.
- *port*: SFTP server port number. By default, SFTP servers use port 22.
- *interface*: Connection type to use. The authorised values are:
  - *ethernet*: Uses the ethernet RJ45 connection to connect to the SFTP server. The Ethernet interface must first have been configured for the connection to work (*See section 3.2.2.1: “Ethernet (Local)”*)
  - *modem*: Uses the modem to connect to the SFTP server. The modem must have been configured first. Otherwise the *apn* command can be used to configure it by text message (*See section 5.3.7: ““apn”: Modem configuration”*).

Returns:

- If successful:
  - For a text message/Modbus slave command: no return.
  - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.

In text message, use of characters ";" and ": “ in the password is prohibited because they are already used as parameter separators.

Example text message/Modbus slave command:

```
sftp=sftp.webdyn.com:login:pas
sword:22:modem
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{

}
```

Response:

```
{
    "callerId": "1",
    “result”:”OK”
}
```

https=<server>:<login>:<password>:<port>:<interface>

5.3.10 "https": Webdav/HTTPS server configuration

Configuration of the Webdav/HTTPS server for "Server 1".

Parameters:

- *server*: WebDAV/HTTPS server name to connect to. This parameter can be a name or an IP address.
- *login*: Login for the indicated WebDAV/HTTPS server.
- *password*: Password for the above login.

- *port*: WebDAV/HTTPS server port number. By default, WebDAV/HTTPS servers use port 443.
- *interface*: Connection type to use. The authorised values are:
  - *ethernet*: Uses the ethernet RJ45 connection to connect to the WebDAV/HTTPS server. The Ethernet interface must first have been configured for the connection to work (*See section 3.2.2.1: "Ethernet (Local)"*)
  - *modem*: Uses the modem to connect to the FTP server. The modem must have been configured first. Otherwise the *apn* command can be used to configure it by text message (*See section 5.3.7: ""apn": Modem configuration"*).

Returns:

- If successful:
  - For a text message/Modbus slave command: no return.
  - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.

In text message, use of characters ";" and ": " in the password is prohibited because they are already used as parameter separators.

Example text message/Modbus slave command:

```
https=webdav.webdyn.com:login:
password:443:modem
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{
	"rpcName":"sunpm.https",
	"parameters":{
		"server": "webdav.webdyn.com",
		"login":"login",
		"password":"password",
		"port":443,
		"interface":"ethernet"
	},
	"callerId": "1"
}
```

Response:

```
{
	"callerId": "1",
	"result":"OK"
}
```

5.3.11 "log": Activation of device communication logs

This command enables the device communications log system.

Parameters:

- *interface*: The name of the interface on which to start the logs: *ethernet*, *cri*, *serial1*, *serial2* or *serial3*.
- *duration*: Duration in minutes during which the logs will be enabled.

```
log=<interface>:<duration>
```

Returns:

- If successful:
  - For a text message/Modbus slave command: no return.
  - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.

Example text message/Modbus slave command:

```
log=serial1:5
```

Example of a functional command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{
	"rpcName":"sunpm.log",
	"parameters":{
		"interface":"serial1",
		"duration":2
	},
	"callerId":"672"
}
```

Response:

```
{
	"callerId": "672",
	"result": "OK"
}
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav) that generates an error:

```
{
	"rpcName":"sunpm.log",
```

Response:

```
"parameters":{
}
```

```
{
	"callerId":"673",
	"error":"Invalid interface: ethernet"
}
```

5.3.12 "setRelay": Relay status update

Changes the relay status: opening, closure or 1 second pulse.

```
setrelay=<action>
```

Parameters:

- *action*: Action to be completed: *open*, *close* or *pulse*.

Returns:

- If successful:
  - For a text message/Modbus slave command: no return.
  - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.

Example text message/Modbus slave command:

```
setrelay=pulse
```

```
{
	“rpcName”:”sunpm.setrelay”,
	"parameters":{
		"action":"pulse",
		"duration":2
	},
	"callerId":"1"
}
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

Response:

```
}
```

```
{
        "callerId": "1",
        “result”:”OK”
}
```

5.3.13 “discoverDevices”: Device discovery

Triggers a device discovery.

```
discoverdevices=<protocol>:<maxDevices>:<interface>:<timeout>:<port>
```

Parameters:

- *protocol*: The codes below are used to start discovering different device types.
  - *sunspec*: Sunspec
  - *tic*: CRI
  - *abbpvi*: ABB PVI
  - *abb2*: ABB other inverters
  - *cefem*: Cefem
  - *cyberpower*: CyberPower
  - *fronius*: Fronius
  - *goodwe*: Goodwe
  - *growatt*: Growatt
  - *huawei*: Huawei
  - *3play*: Ingeteam 3Play
  - *1playhf*: Ingeteam 1Play HF
  - *1playtlm*: Ingeteam 1Play TLM
  - *powerblock*: Ingeteam Power Block
  - *powermax*: Ingeteam Powe Max
  - *kacomodbus*: Kaco Modbus
  - *kostal*: Kostal
  - *smamodbus*: SMA Modbus
  - *saj*: SAJ
  - *solaredge*: SolarEdge
  - *solis*: Solis
  - *sungrow*: Sungrow
  - Some devices that use a proprietary protocol can also be detected using this command (see specific proprietary protocol appendix).
- *maxDevices*: Maximum number of devices to discover. When this number is reached, discovery stops. This parameter is ignored for the TIC protocol.
- *interface*: Interface used for the discovery: *serial1*, *serial2*, *serial3*, *ethernet1* or *ethernet2*. This parameter is ignored for the TIC protocol. Only the SunSpec protocol is compatible with the *ethernet1* and *ethernet2* values.
- *timeout*: Maximum time (in milliseconds) for a device to be discovered. This parameter is ignored for the TIC protocol.
- *port*: Port used for network discovery. This parameter only makes sense for SunSpec discovery on Ethernet. It is ignored in all other cases.

Returns:

- If successful: the number of discovered devices.
- If an error is encountered: an explanatory message.

Example text message/Modbus slave command:

```
discoverdevices=sunspec:10:ser
ial1:10000
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{

}
```

Response:

```
{
    "callerId": "1",
    “result”:”OK”
}
```

5.3.14 "getParameters": Parameter collection

getparameters

When this command is received by the WebdynSun PM, the variables defined with an action code 1 (ACTION_PARAMETER) are collected. On the next IS connection, the data will be uploaded to a file named *<UID>_<interface>_P_<timestamp>.csv.gz*.

Parameters:

- No parameters.

Returns:

- If successful:
    - For a text message/Modbus slave command: no return.
    - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.

Example text message/Modbus slave command:

```
getparameters
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{
    "rpcName":"sunpm.getParameters",
    "calleeId":"1"
}
```

Response:

```
{

}
```

5.3.15 "getData": Collection of action code 6 or 7 variables

When this command is received by the WebdynSun PM, the variables defined with 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 and mean type values are only calculated in the acquisition file. On the next IS connection, the instant data will be uploaded to a file named *<UID>_<interface>_I_<timestamp>.csv.gz*.

Furthermore, if an MQTT server is configured, the connection to it is automatic so that the data is uploaded immediately.

```
getdata
```

If data is available in addition to the instant data, it will also be uploaded to the remote server with the usual naming.

Parameters:

- No parameters.

Returns:

- If successful:

- For a text message/Modbus slave command: no return.
- For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return

- If an error is encountered: an explanatory message.

Example text message/Modbus slave command:

```
getdata
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
[{
    "rpcName":"sunpm.getData",
    "params":"1"
}]
```

Response:

```
{

}
```

5.3.16 "writeVariable": Write of a variable to a device

Writing a variable declared in a definition file.

```
writeVariable <deviceName>:<tagName>:<value>
```

Parameters:

- *deviceName*: Target device name.
- *tagName*: Target tag.
- *value*: Value to be written. It can be a number or a character string. Note that for text messages, the value will be interpreted as a number if possible. To force interpretation as a character string, the value can be placed between "".

Returns:

- If successful:
  - For a text message/Modbus slave command: no return.
  - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.

Example: Text message/Modbus slave command with a character string value:

```
writevariable=INV1:name:
“inverter1”
```

Example Text message/Modbus slave command with a value that is a number:

```
writevariable=INV1:setLimit:30
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{
}
```

Response:

```
{
    "callerId": "1",
    “result”:”OK”
}
```

5.3.17 "setKey": Addition of client script decryption keys

This command is used to add client Lua script decryption keys. The keys must be in a specific format and contained in a file. The file containing the keys must be made available on a server so that it can be downloaded by the command. For more details on the key format for Lua scripts, see document "WebdynSunPM LUA User Guide.pdf".

```
setkey=<url>:<login>:<password>:<interface>
```

Parameters:

- *url*: URL of the file to be retrieved. The accepted protocols are HTTP, HTTPS, FTP and SFTP. The port can be indicated using in address:port format
- *login*: Server login
- *password*: Server password
- *interface*: Interface used for the connection: *ethernet* or *modem*.

Returns:

- If successful:
    - For a text message/Modbus slave command: no return.
    - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.

In text message, use of characters ";" and ": “ in the password is prohibited because they are already used as parameter separators.

Example text message command:

```
setkey=ftp://ftp.webdyn.com/sc
ript_key.json:modem:login:pwd
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{
        "url":"ftp://ftp.webdyn.com/script_key.json"
        "interface":"ethernet",
        "login":"login",
        "password":"pwd"
    },
    "callerId":"203"
}
```

Response:

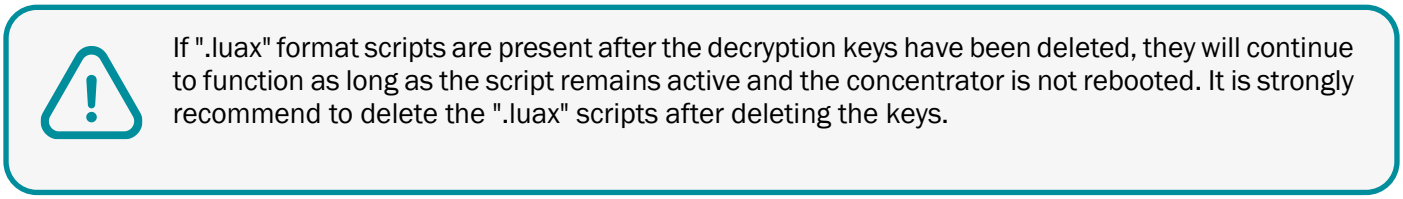
```
{
    "callerId": "203",
    “result”:”OK”
}
```

```
{
    "callerId":"203",
    "error":"Invalid interface: ethernet"
}
```

Or response with an error:

5.3.18 "deleteKey": Removal of client script decryption keys

This command is used to client Lua script decryption keys.



If ".luax" format scripts are present after the decryption keys have been deleted, they will continue to function as long as the script remains active and the concentrator is not rebooted. It is strongly recommend to delete the ".luax" scripts after deleting the keys.

Parameters:

- No parameters.

deletekey

Returns:

- If successful:
  - For a text message/Modbus slave command: no return.
  - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.

Example text message/Modbus slave command:

```
deletekey
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
[{
        "rpcName":"sunpm.deleteKey",
        "callerId":"117"
}]
```

Response:

```
{
        "callerId": "117",
        "result":"OK"
}
```

5.3.19 "startScript": Starting a script

Starts the indicated script.

Parameters:

- *name*: Name of the script to be started.

Returns:

startscript=<name>

- If successful:
  - For a text message/Modbus slave command: no return.
  - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.

Example text message/Modbus slave command:

```
startscript=ActivePowerRegulat
ion-V1_03
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{
	“rpcName”:”sunpm startScript"
	"parameters": {
		"name": "ActivePowerRegulation-V1_03"
	},
	"callerId": "1"
}
```

Response:

```
{
	"callerId": "1",
	“result”:”OK”
}
```

stopscript=<name>

5.3.20 "stopScript”: Stopping a script

Stops the indicated script.

Parameters:

- name: Name of the script to be stopped.

Returns:

- If successful:
  - For a text message/Modbus slave command: no return.
  - For an FTP/SFTP/MQTT/MQTTS/WebDav command: "OK" return
- If an error is encountered: an explanatory message.

Example text message/Modbus slave command:

```
stopscript=ActivePowerRegulati
on-V1_03
```

Example of a command file (FTP/SFTP/MQTT/MQTTS/WebDav):

```
{

}
```

Response:

```
{
    "callerId": "1",
    “result”:”OK”
}
```

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.

After each update, it is advisable to delete the configuration files present on the server. This operation allows the hub to automatically generate new configuration files integrating any changes during the next connection.

6.1 Using the web interface

To update the concentrator locally, use its interface and go to the "System" menu, then the "Settings" sub-menu and in the "Concentrator", then follow the web interface update procedure *(see section 3.2.5.1.1: "Updating and identifier").*

6.2 Using FTP/SFTP/WebDAV

For remote updates, follow the steps below:

1. Place the "wgapp_x.x.x.xxxxx.spm" file containing the updates in the "BIN" directory on the remote server.
2. 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.

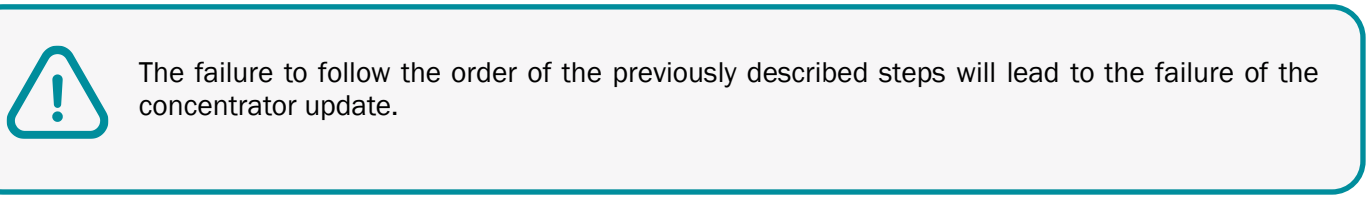
The concentrator will retrieve its configuration file and its new firmware at the next connection to the server.

The update using FTP, SFTP or WebDAV-HTTPS can only be carried out by the primary server (server 1).

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.



6.3 By text message or MQTT/MQTTS command

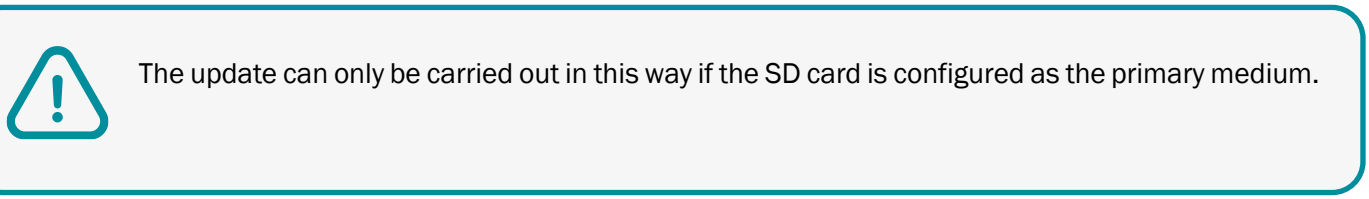
The "*updateFirmware"* command is used to update the WebdynSun PM by indicating a URL at which the new firmware is to be found. For details of the procedure, see section 5.3.5: “"updateFirmware": Concentrator software update”.

6.4 By micro SD card

For updates using the micro SD card, follow the steps below:

1. Place the “wgapp_x.x.x.xxxxx.spm” file containing the update in the “\BIN” directory on the SD card.
2. Edit the “<uid>_config_.ini” file (< UID>: Concentrator identifier) which is in the “CONFIG” directory on the SD card.
   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.

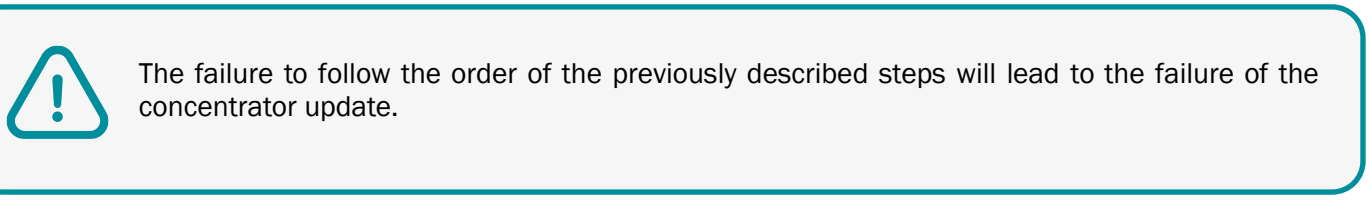
The concentrator will retrieve its configuration file and its new firmware at the next “connection” to the SD card.



The application of the update can be seen in the LOG files uploaded to the “LOG” directory on the SD card.

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.



Webdyn does not supply any SD cards. Contact a computer hardware retailer.

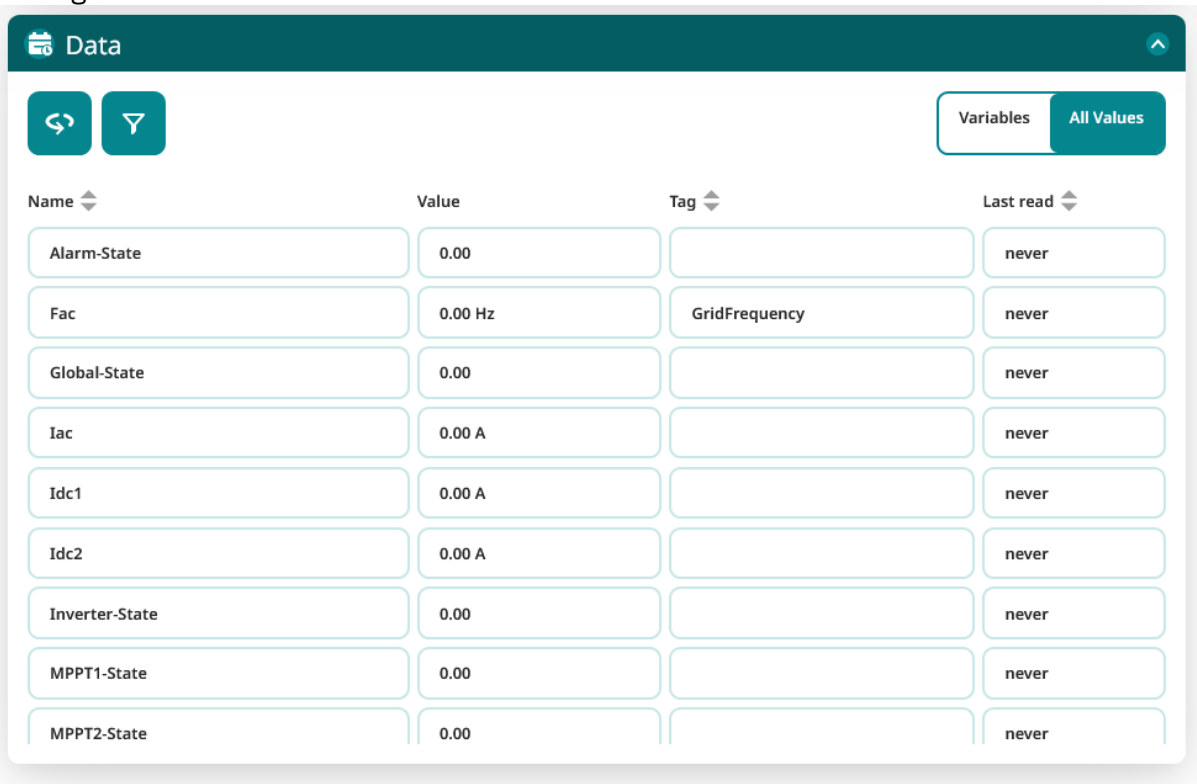
7 Tools & diagnostics

7.1 Diagnostics

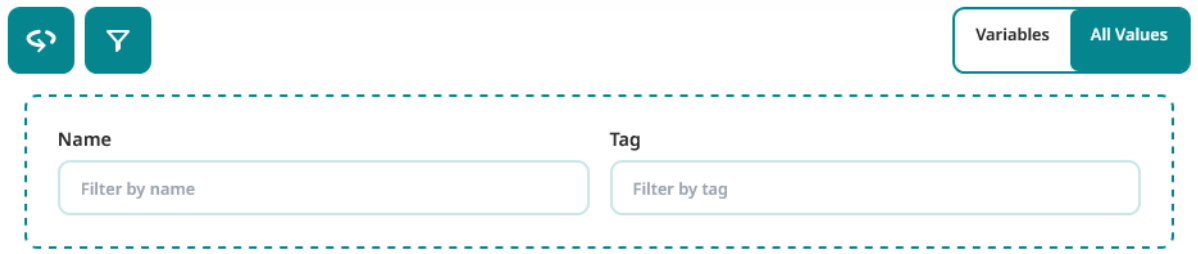
If the product malfunctions, there are several ways to troubleshoot the faults.

- Firstly, if the concentrator is connected using a modem with a SIM card, it is possible to request the product status by sending it the "*status*" command.(*See section 5.3.2: ""status": Concentrator status retrieval"*)
- The log files described in section 4.1.8: ""LOG"" can be used to view the different errors and understand their causes. There are log files for each concentrator function in order to better isolate problems.
- It is also possible to diagnose Ethernet or serial communication errors using the built-in tools. These tools can be enabled using FTP *(See section 5.3.11: ""log": Activation of device communication logs")*, by text message*(See section 5.3.11: ""log": Activation of device communication logs"*) or using the WEB interface (*See section 0: "*

- *Data*
- *Under* device settings, a « Data » tab can be used to see all parameters collected by the concentrator. For each parameters : name, value, tag and last read is present. Two buttons on the top left for refreshing or filtering.



By clicking on filter button, it is possible to search for name or tag. Another click on filter button cancels filter.



It is also possible to filter data with top right button “Variables” or “All values”.

- Variables: Only variables are displayed. Data set as parameters or deactivated (“non”) will not be shown.
- All Values: all data are shown including parameters and deactivated data.

- Device troubleshooting tools").
- On the WEB interface "Dashboard" page, the "Site information" part shows all the devices and is used to quickly detect those that have anomalies. (*See section3.2: "Embedded web interface*")

If necessary, the WebdynSunPM can be rebooted remotely using the "*reboot »*" text message command (*See section 5.3.4: ""reboot": Concentrator reboot*").

Finally, the product can be fully reinitialised by sending the *factory*" text message (*See section 5.3.3: ""factory": Back 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

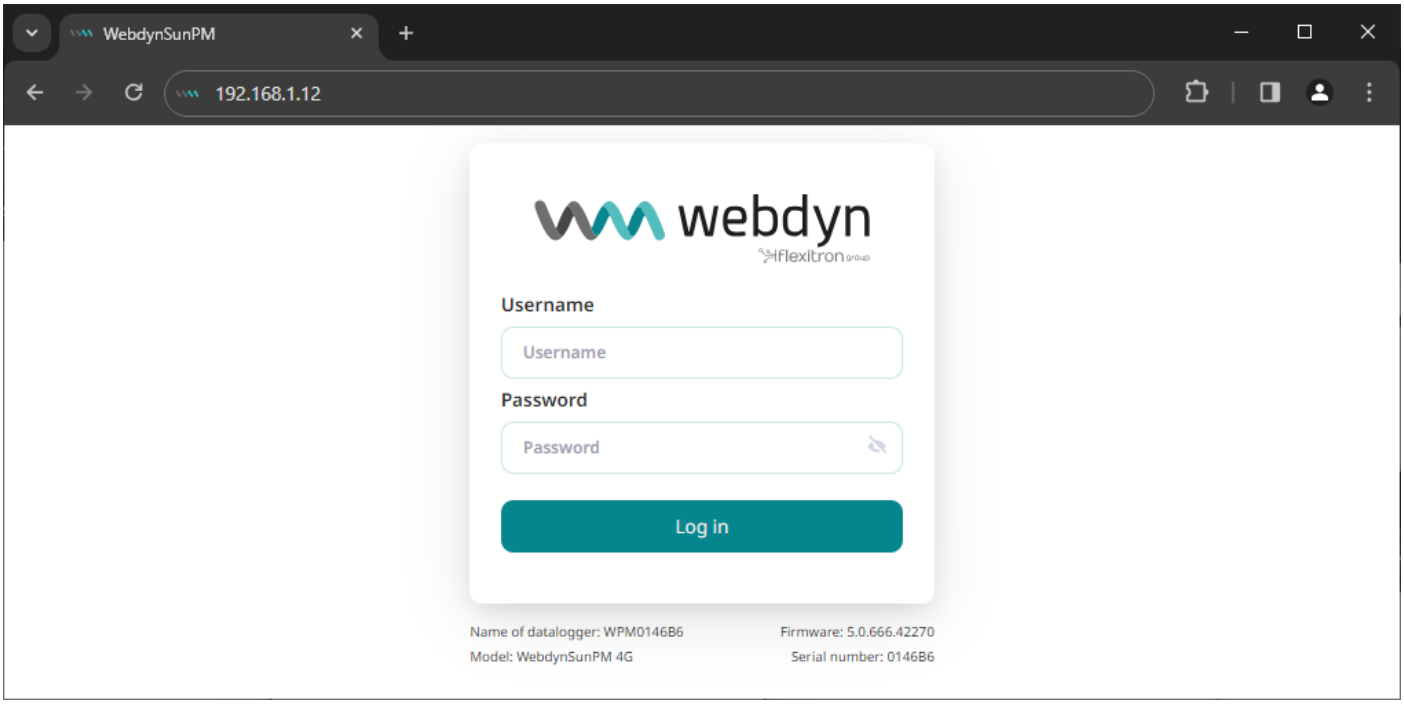
8 FAQ

Gateway configuration:

How do I 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 will appear:



The default LOGIN is:

Login
userhigh

Password is device specific

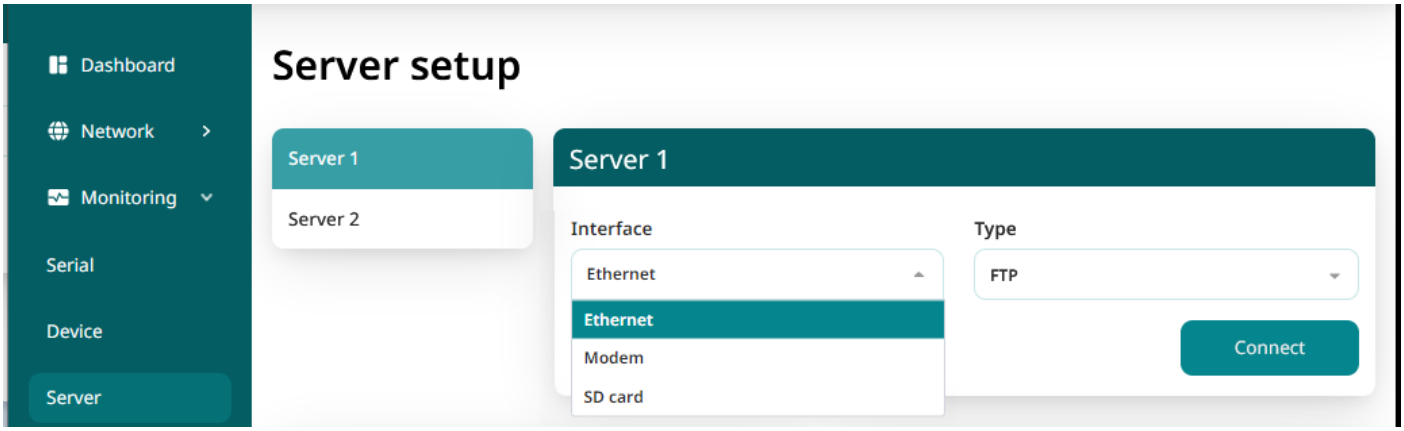
Click "Log in".

How do you configure the WebdynSunPM concentrator to access the remote FTP server?

There are two configuration solutions, using the web interface and using text messages:

Configuration using the web interface:

Start by setting up a connection to the concentrator by connecting to it to access the server configuration:

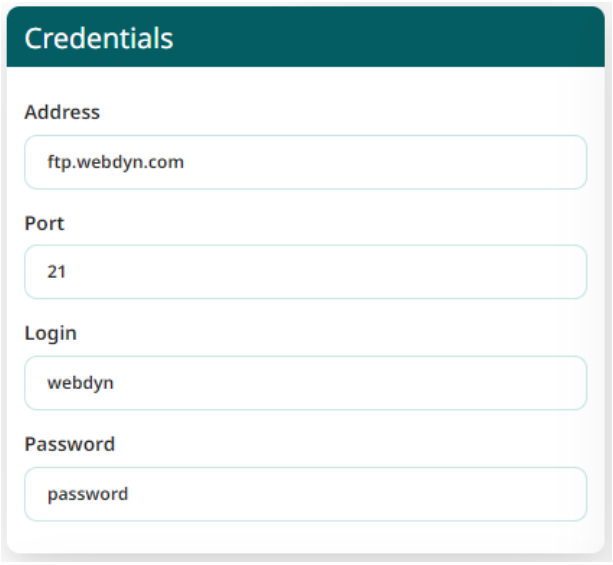


Enter the “Ethernet” or “modem” connection type.

For an Ethernet configuration, make sure the IP parameters are compatible with server access according to the concentrator local network configuration. 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 local network configuration in “Network” and then “Local” page (See section 3.2.2.1: “Ethernet (Local)”*).

For a modem connection, the modem configuration must be correct before a connection can be set up. This configuration is done using the mobile network configuration page in “Network” then “Mobile” (*See section 3.2.2.2: “Modem (Mobile)”*).

The parameters for the servers to be configured are at least the following:



Therefore the following fields need to be configured: “Interface”, “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 section3.1.2: “Configuration files”*)

apn=<apn>:<login>:<password>

Text message configuration:

Text message configuration requires sending the following commands:

- apn: to configure the SIM card APN (*See section 5.3.7: ““apn”: Modem configuration”*).

- ftp: to configure the FTP server that will contain the concentrator configuration (*See section 5.3.8: "“ftp”: FTP/SFTP server configuration”*).

- connect: to launch the connection to the FTP server and load the configuration (*See section 5.3.1: "“connect”: Connection trigger”*).

```
ftp=<server>:<login>:<password>:<port>:<interface>
```

For text message configuration, do 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.

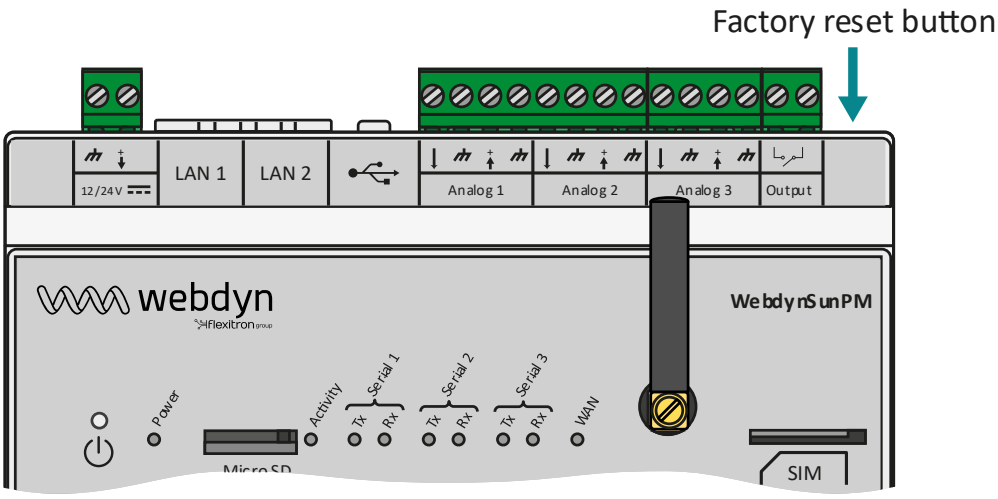
For all other configurations, and to determine the best solution, contact the Webdyn sales department which will advise you and direct you to the relevant contacts: contact@webdyn.com.

General gateway use

How do you reinitialise the concentrator?

There are 3 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” text message can also be used for a factory reset. Just the sent it to the SIM card phone number (*See section 5.3.3: ““factory”: Back to factory settings*”).
- By command file for an FTP, SFTP or WebDAV server, or by RPC command using an MQTT or MQTTS server.

The factory reset restores the original configuration. The data and licences will not be kept.
The licences can be retrieved from the remote server or directly from the Webdyn licence server from the web interface.

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 files exchanged.
The average is about 5 MB per month but this varies from one installation to another.

Inverter compatibility

Which inverters are compatible?

See section 0: “

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 2 or 4 wire connection.
- All the devices must be configured with identical bus characteristics: same speed, same parity, same number of stop bits and data bits on all the devices and in the WebdynSunPM.
- Each device must be assigned a unique Modbus address (between 1 and 247) on the bus. (UnitID)

9 Support

If there are technical problems related to our products, contact WEBDYN support:

Webdyn SA

26 Rue des Gaudines

78100 Saint-Germain-en-Laye

FRANCE

Phone: +33 1 39 04 29 40

email support@webdyn.com

https://www.webdyn.com

Please have the following elements to hand:

- Product type
- Product serial number.
- Product hardware and software version.
- Concentrator logs
- Concentrator configuration

The user manual and firmware are available at this web address: https://www.webdyn.com/support/webdynsunpm/

10 APPENDICES

10.1 Appendix A: "_config.ini" configuration 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 section6.2: "Using FTP/SFTP/WebDAV"*)	
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 section6.2: "Using FTP/SFTP/WebDAV"*)	
Concentrator_Identifier	Gateway identifier (UID) If this field is left empty, the default identifier is used.	WPMxxxxxx (where xxxxxx are the last characters of the serial number).
FTP_DirAlarm	FTP/ SFTP server 1 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/ SFTP server 1 directory in which the concentrator will get the update files when requested. (*See section6.2: "Using FTP/SFTP/WebDAV"*)	/BIN
FTP_DirCert	FTP/ SFTP server 1 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/ SFTP server 1 directory in which the concentrator will get the command files used later. See the command file use section for more details. (*See section5.2.1: "Command file"*)	/CMD
FTP_DirConfig	FTP/ SFTP server 1 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.	/CONFIG

	See the section on concentrator configuration using configuration files for the operating principle (*See section 3.1.2: "Configuration files"*)	
FTP_DirData	FTP/ SFTP server 1 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/ SFTP server 1 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/ SFTP server 1 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/ SFTP server 1 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 server. If a file is modified on the server, it is loaded on the concentrator at the next connection. If a file is added to the server, it is ignored.	/SCRIPT
FTP_EnableAdvancedData	Addition of the number of complete reads during this acquisition period to the data files placed on the FTP/SFTP server: • **0:** Number of complete reads not added • **1:** Addition of the number of complete reads	0
FTP_EuroDateFormat	Indicates the format used to timestamp the data sent to the FTP/ SFTP server 1. The possible values are: • **0**: the ISO format is used (YY/MM/DD-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/ SFTP server 1 The possible values are: • **0**: no optional header • **1**: optional headers generated	0
FTP_Login	The login to use to connect to the FTP/ SFTP server 1. This value is mandatory.	

FTP_Password	The password for the “FTP_Login” to connect to FTP/ SFTP server 1.	
FTP_Port	The network port to use to connect to the FTP/ SFTP server 1.	21
FTP_SynchroniseCertificates	Choice to synchronise certificates on the FTP/SFTP server 1: • **0:** No certificate synchronisation • **1:** Enables certificate synchronisation	0
FTP_TwoStepsSendingDisabled	Used to transfer the files in 2 steps using a temporary file while the file is not complete on the FTP/SFTP server 1: 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/ SFTP server 1 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 on web services for more information on using this feature. The possible values are: • **0**: web services disabled • **1**: web services enabled	0
FTP_WebServicesUrl	URL that will be called in response to certain FTP actions if web services are enabled. See the section on web services for more information on using this feature. The URL must have the following format: • http://adresse/page/	
FTP2_DirAlarm	FTP/ SFTP server 2 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/ SFTP server 2 directory in which the concentrator will get the update files when requested. (*See section6.2: “Using FTP/SFTP/WebDAV”*)	/BIN
FTP2_DirCert	FTP/ SFTP server 2 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/ SFTP server 2 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/ SFTP server 2 directory to which the concentrator uploads its configuration files. Note that the directory MUST exist. The concentrator will not create it when uploading files.	/CONFIG

	See the section on concentrator configuration using configuration files for the operating principle (*See section 3.1.2: "Configuration files"*)	
FTP2_DirData	FTP/ SFTP server 2 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/ SFTP server 2 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 3.1.2.2: "Connected device definition"*)	/DEF
FTP2_DirLog	FTP/ SFTP server 2 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/ SFTP server 2 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 server. If a file is modified on the server, it is loaded on the concentrator at the next connection. If a file is added to the server, it is ignored.	/SCRIPT
FTP2_EuroDateFormat	Indicates the format used to timestamp the data sent to the FTP/ SFTP server 2. The possible values are: • **0**: the ISO format is used (YY/MM/DD-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/ SFTP server 2 The possible values are: • **0**: no optional header • **1**: optional headers generated	0
FTP2_EnableAdvancedData	Addition of the number of complete reads during this acquisition period to the data files placed on the FTP/SFTP server: • **0:** Number of complete reads not added • **1:** number of complete reads added	0
FTP2_Login	The login to use to connect to the FTP/ SFTP server 2. This value is mandatory.	
FTP2_Password	The password for the "FTP2_Login" to connect to FTP/ SFTP server 2.	
FTP2_Port	The network port to use to connect to the FTP/ SFTP server 2.	21

FTP2_SynchroniseCertificates	Choice to synchronise certificates on the FTP/SFTP server 2: • **0:** No certificate synchronisation • **1:** Enables certificate synchronisation	0
FTP2_TwoStepsSendingDisabled	Used to transfer the files in 2 steps using a temporary file while the file is not complete on the FTP/SFTP server 2: 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/ SFTP server 2 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 on web services for more information on using this feature. The possible values are: • **0**: web services disabled • **1**: web services enabled	0
FTP2_WebServicesUrl	URL that will be called in response to certain FTP 2 actions if web services are enabled. See the section on web services for more information on using this feature. The URL must have the following format: • http://adresse/page/	
MQTT2_AlarmTopic	Name of the alarm topic to be published. If the field is empty, no alarms will be published to the broker. If a topic name is entered, the concentrator stays 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 entered MQTTS server. The certificate is to be retrieved from your MQTT server and imported to the concentrator using FTP or the web interface. Works for all MQTT types except "**mqtt**"	
MQTT2_CertFile	Name of the certificate specific to the concentrator used for the connection. The certificate is to be retrieved from your MQTT server and imported to the concentrator using FTP or the web interface. Works for all MQTT types except "**mqtt**" and "**mqtts_gcloud**"	
MQTT2_CloudDevice	Unique, customisable device identifier in a register defined on the MQTT server. This parameter must be retrieved from your MQTT server and corresponds to: • "deviceId" on Google IoT Cloud. • "device_id" on Azure IoT Hub. Works only for the "**mqtts_gcloud**" and "**mqtts_azure**" types	
MQTT2_CloudProjectId	Unique, customisable identifier for the project defined on the MQTT server.	

	This parameter must be retrieved from your MQTT server and corresponds to: • "projectId" on Google IoT Cloud. • The "Iot Hub" name on Azure IoT Hub. Works only for the "**mqtts_gcloud**" and "**mqtts_azure**" types	
MQTT2_CloudRegion	Region of the MQTT server in the device register. This parameter must be retrieved from your MQTT server and corresponds to: • "deviceRegistryLocation" on Google IoT Cloud. *Example: "europe-west1"* Works only for the "**mqtts_gcloud**" type	
MQTT2_CloudRegistry	Name of the customisable register defined on the MQTT server. This parameter must be retrieved from your MQTT server and corresponds to: • "deviceRegistryId" on Google IoT Cloud. Works only for the "**mqtts_gcloud**" type	
MQTT2_CloudSigningAlgo	Type of key used to verify the MQTT server certificate signature. This parameter must be retrieved from your Google IoT Cloud server. The possible values are: • "RS256" for the RSA key • "ES256" for the elliptical curve key Works only for the "**mqtts_gcloud**" type	
MQTT2_ClientId	Customisable device identifier on the MQTT server. This parameter is retrieved from your MQTT server. Works only for the "**mqtt**" and "**mqtts**" types	
MQTT2_ControlTopic	Name of the topic for commands to be retrieved by the concentrator. The MQTT2_ResultTopic parameter must be set to use the commands. If a topic name is entered, the concentrator stays in permanent connection mode with the MQTT server. Works for all MQTT types except "**mqtts_azure**"	
MQTT2_EnableAdvancedData	Publication of the number of complete reads over this acquisition period in the data topic. The possible values are: • **0**: Deactivated • **1**: Enabled	0
MQTT2_EnableAlarmPost	Enable the publication of alarms on the dedicated topic. The possible values are: • **0**: Alarm publication is disabled • **1**: Alarm publication of alarms is enabled and the concentrator remains in permanent connection mode with the MQTT server. Works only for the "**mqtts_azure**" type	
MQTT2_EnableInvokeMethod	Enable method calls. Makes it possible to use dedicated topics. The possible values are: • **0**: Method calls disabled	

	• **1**: Method calls enabled and the concentrator remains in permanent connection mode with the MQTT server. Works only for the "**mqtts_azure**" type	
MQTT2_Insecure	Disable verification of the host name specified in the certificates. The possible values are: • **0**: Verification enabled • **1**: Verification disabled. Works only for the "**mqtts**" type	
MQTT2_KeepAlive	If there is no exchange with the MQTT server for the time defined in seconds, the concentrator sends a ping to the MQTT server to check the connection. If the value is "0", KeepAlive is disabled. If the concentrator is in permanent connection mode with the MQTT server and a disconnection is detected after a KeepAlive, the concentrator will automatically reconnect to the MQTT server.	10
MQTT2_KeyFile	Name of the file containing the private key specific to the concentrator used for the connection. The file must be retrieved from your MQTT server and imported to the concentrator using FTP or the web interface. Works for all MQTT types except "**mqtt**"	
MQTT2_Login	User name to access the MQTT server. Works only for the "**mqtt**" and "**mqtts**" types	
MQTT2_Password	MQTT server access password. Works only for the "**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 message sending (Quality Of Service). The possible values are: • **0**: The message will only be sent once, i.e. with no guaranteed 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 be always be saved by the concentrator and will continue to be sent as long as the broker does not confirm it has been sent. (avoids message duplication) For the "**mqtts_gcloud**", "**mqtts_azure**" and "**mqtts_aws**" types, QoS 2 is not supported.	1
MQTT2_ResultTopic	Name of the topic for the results of commands sent to the concentrator. The MQTT2_ControlTopic parameter must be set to use the commands. If a topic name is entered, the concentrator stays in permanent connection mode with the MQTT server. Works for all MQTT types except "**mqtts_azure**"	

MQTT2_Timeout	Maximum waiting time in seconds for the MQTT server response. If the server has not responded within the allotted time, the transmission is stopped and reattempted during the next schedule. Only works with 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 Works only for the "**mqtts**" type	tlsv1.2
MQTT2_Topic	Topic name for the data uploaded by the concentrator. Works for all MQTT types except "**mqtts_azure**"	
HTTP_DirAlarm	WebDAV-HTTPS server 1 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
HTTP_DirBin	WebDAV-HTTPS server 1 directory in which the concentrator will get the update files when requested. See the section on update configuration for how to use updates. (*See section6.2: "Using FTP/SFTP/WebDAV"*)	/BIN
HTTP_DirCert	WebDAV-HTTPS server 1 directory in which the concentrator will get the certificates to use for MQTT connections. See the WebDAV server configuration section for more details.	/CERT
HTTP_DirCmd	WebDAV-HTTPS server 1 directory in which the concentrator will get the command files used later. See the command file configuration use section for more details	/CMD
HTTP_DirConfig	WebDAV-HTTPS server 1 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 configuration files for the operating principle (*See section 3.1.2: "Configuration files"*)	/CONFIG
HTTP_DirData	WabDAV-HTTPS server 1 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
HTTP_DirDef	WebDAV-HTTPS server 1 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.	/DEF

	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"*)	
HTTP_DirLog	WebDAV-HTTPS server 1 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
HTTP_DirScript	WebDAV-HTTPS server 1 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 server. If a file is modified on the server, it is loaded on the concentrator at the next connection. If a file is added to the server, it is ignored.	/SCRIPT
HTTP_EuroDateFormat	Indicates the format used to timestamp the data sent to the WebDAV-HTTPS server 1. The possible values are: • **0**: the 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 the optional headers in the data files uploaded to the WebDAV-HTTPS server 1 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 configured for the login configured in the "HTTP_Login" parameter to connect to the WebDAV-HTTPS server 1.	
HTTP_Port	The network port to use to connect to the WebDAV-HTTPS server 1.	443
HTTP_SynchroniseCertificates	Certificate synchronisation selection: • **0:** No certificate synchronisation • **1:** Enables certificate synchronisation	0
HTTP_UploadLog	Indicates whether the gateway must also upload the internal programmed connection operating logs in the WebDAV-HTTPS server 1 log upload directory. The possible values are: • **0**: no file upload • **1**: file upload	0
HTTP2_DirAlarm	WebDAV-HTTPS server 2 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

HTTP2_DirBin	WebDAV-HTTPS server 2 directory in which the concentrator will get the update files when requested. See the section on update configuration for how to use updates. (*See section6.2: "Using FTP/SFTP/WebDAV"*)	/BIN
HTTP2_DirCert	WebDAV-HTTPS server 2 directory in which the concentrator will get the certificates to use for MQTT connections. See the WebDAV server configuration section for more details.	/CERT
HTTP2_DirCmd	WebDAV-HTTPS server 2 directory in which the concentrator will get the command files used later. See the command file use section for more details. (*See section5.2.1: "Command file"*)	/CMD
HTTP2_DirConfig	WebDAV-HTTPS server 2 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 configuration files for the operating principle (*See section 3.1.2: "Configuration files"*)	/CONFIG
HTTP2_DirData	WebDAV-HTTPS server 2 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
HTTP2_DirDef	WebDAV-HTTPS server 2 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 3.1.2.2: "Connected device definition"*)	/DEF
HTTP2_DirLog	WebDAV-HTTPS server 2 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
HTTP2_DirScript	WebDAV-HTTPS server 2 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 server. If a file is modified on the server, it is loaded on the concentrator at the next connection. If a file is added to the server, it is ignored.	/SCRIPT
HTTP2_EuroDateFormat	Indicates the format used to timestamp the data sent to the WebDAV-HTTPS server 2. The possible values are:	0

	• **0**: the ISO format is used (YY/MM/DD-HH:MM:SS) • **1**: the European format is used (DD/MM/YY-HH:MM:SS)	
HTTP2_HeaderOption	Indicates whether the gateway must add the optional headers in the data files uploaded to the WebDAV-HTTPS server 2 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 for the "HTTP2_Login" to connect to the WebDAV-HTTPS server 2.	
HTTP2_Port	The network port to use to connect to the WebDAV-HTTPS server 2.	443
HTTP2_SynchroniseCertificates	Certificate synchronisation selection: • **0:** No certificate synchronisation • **1:** Enables certificate synchronisation	0
HTTP2_UploadLog	Indicates whether the gateway must also upload the internal programmed connection operating logs in the WebDAV-HTTPS server 2 log upload directory. The possible values are: • **0**: no file upload • **1**: file upload	0
LOG_Level	The concentrator log level in the system files. Used for the box debug mode. The parameter impacts the system log file detail level (*See section 4.1.8.4: "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)	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"	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**: uses the microSD card inserted in the concentrator **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**: WebDAV-HTTPS server **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 the 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**: uses the microSD card inserted in the concentrator **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**: WebDAV-HTTPS server • **mqtt**: MQTT server • **mqtts**: secure MQTT server • **mqtts_aws**: MQTTS server on “AWS IoT” • **mqtts_azure**: MQTTS server on “Azure IoT” • **mqtts_gcloud**: MQTTS server on “Google Cloud IoT” (obsolete) **This parameter cannot be empty**	ftp
SMS_Password	Password for SMS encryption	
WEB_Password	Password to access the configuration using the concentrator web site.	high

Note that unless otherwise indicated, the 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.

10.2 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

10.3 Appendix C: Compatible inverters

The WebdynSunPM is compatible with all MODBUS and SUNSPEC MODBUS inverters as well as inverters compatible with the following proprietary protocols:

- DELTA
- KACO
- SMANET
- SOLARMAX

It also allows the automatic detection of the following inverters:

10.3.1 ABB

- PVI-10.0-TL-OUTD
- PVI-12.5-TL-OUTD
- TRIO-20.0-TL-OUTD
- TRIO-27.6-TL-OUTD
- ULTRA 700.0-TL
- ULTRA 750-TL-OUTD-X-US-690
- ULTRA 1050.0-TL
- ULTRA 1100-TL-OUTD-X-US-690
- ULTRA 1400.0-TL
- ULTRA 1500-TL-OUTD-X-US-690

10.3.2 CEFEM

TRIO-TOP series:

- 9 TR RO (HF)
- 10K TR (HF)
- 12K TR (HF)
- 15K TR (HF)
- 18K TR (HF)

TRIO-SUN series:

- 18 kVA
- 20 kVA
- 25 kVA
- 30 kVA
- 33 kVA
- 36 kVA

10.3.3 Fronius

IG series:

- IG 15
- IG 20
- IG 30
- IG 40
- IG 60
- IG 60 HV
- IG 300
- IG 400
- IG 500
- IG 2000
- IG 3000
- IG 4000
- IG 5100

IG LV series:

- IG 2500-LV
- IG 4500-LV

IG TL series:

- IG-TL 3.0
- IG-TL 3.6
- IG-TL 4.0
- IG-TL 5.0

IG PLUS series:

- IG Plus 35-1
- IG Plus 35V-1
- IG Plus 50-1
- IG Plus 50V-1
- IG Plus 70-1
- IG Plus 70-2
- IG Plus 70V-1
- IG Plus 70V-2
- IG Plus 100-1
- IG Plus 100-2
- IG Plus 100V-1
- IG Plus 100V-2
- IG Plus 120-3
- IG Plus 120V-3
- IG Plus 150-3
- IG Plus 150V-3
- IG Plus 3.0-1 UNI
- IG Plus 3.8-1 UNI
- IG Plus 5.0-1 UNI
- IG Plus 6.0-1 UNI
- IG Plus 7.5-1 UNI
- IG Plus 10.0-1 UNI
- IG Plus 11.4-1 UNI
- IG Plus 11.4-3-Delta
- IG Plus 12.0-3 WYE277

CL series:

- CL 36.0
- CL 48.0
- CL 60.0
- CL 36.0 WYE277
- CL 48.0 WYE277
- CL 60.0 WYE277
- CL 33.3 Delta
- CL 44.4 Delta
- CL 55.5 Delta

GALVO series:

- GALVO 1.5-1
- GALVO 2.0-1
- GALVO 2.5-1
- GALVO 3.0-1
- GALVO 3.1-1

SYMO series:

- SYMO 3.0-3-S
- SYMO 3.7-3-S
- SYMO 4.5-3-S
- SYMO 3.0-3-M
- SYMO 3.7-3-M
- SYMO 4.5-3-M
- SYMO 5.0-3-M
- SYMO 6.0-3-M
- SYMO 7.0-3-M
- SYMO 8.2-3-M
- SYMO 10.0-3-M
- SYMO 12.5-3-M
- SYMO 15.0-3-M
- SYMO 17.5-3-M
- SYMO 20.0-3-M

PRIMO series:

- PRIMO 3.0-1
- PRIMO 3.5-1
- PRIMO 3.6-1
- PRIMO 4.0-1
- PRIMO 4.6-1
- PRIMO 5.0-1
- PRIMO 6.0-1
- PRIMO 8.2-1

PRIMO GEN24 Series:

- PRIMO GEN24 3.0 PLUS
- PRIMO GEN24 3.6 PLUS
- PRIMO GEN24 4.0 PLUS
- PRIMO GEN24 4.6 PLUS
- PRIMO GEN24 5.0 PLUS
- PRIMO GEN24 6.0 PLUS

TAURO series:

- TAURO 50-3-D
- TAURO 50-3-P

TAURO ECO series:

- TAURO ECO 50-3-D
- TAURO ECO 99-3-D
- TAURO ECO 100-3-D
- TAURO ECO 50-3-P
- TAURO ECO 99-3-P
- TAURO ECO 100-3-P

ECO series:

- FRONIUS ECO 25.0-3-S
- FRONIUS ECO 27.0-3-S

10.3.4 GOODWE

ES series:

- GW3648D-ES
- GW5048D-ES

ES G2 Series:

- GW3000-ES-20
- GW3600-ES-20
- GW3600M-ES-20
- GW5000-ES-20
- GW5000M-ES-20
- GW6000-ES-20
- GW6000M-ES-20

ESA series:

- GW5048-ESA

EM series:

- GW3048-EM
- GW3648-EM
- GW5048-EM

BP series:

- GW2500-BP
- GW3600S-BP
- GW5000S-BP

SBP G2 Series:

- GW3600-SBP-20
- GW5000-SBP-20

- GW6000-SBP-20

ET series:

- GW5KL-ET
- GW6KL-ET
- GW8KL-ET
- GW10KL-ET
- GW5K-ET
- GW6.5K-ET
- GW8K-ET
- GW10K-ET
- GW15K-ET
- GW20K-ET
- GW25K-ET
- GW29.9K-ET
- GW30K-ET
- GW5KN-ET
- GW6.5KN-ET
- GW8KN-ET
- GW10KN-ET
- GW50K-ETC

BT series:

- GW5K-BT
- GW6K-BT
- GW8K-BT
- GW10K-BT
- GW50K-BTC

EH series:

- GW3600-EH
- GW5000-EH
- GW6000-EH

EH+ series:

- GW3600N-EH
- GW5000N-EH
- GW6000N-EH

BH series:

- GW3K-BH
- GW3600-BH
- GW5000-BH
- GW6000-BH

XS series:

- GW700-XS
- GW1000-XS
- GW1500-XS
- GW2000-XS
- GW2500-XS

- GW3000-XS
- GW2500N-XS
- GW3000N-XS

XS+ series:

- GW700-XS-11
- GW1000-XS-11
- GW1500-XS-11
- GW2000-XS-11
- GW2500-XS-11
- GW3000-XS-11

DNS series:

- GW2900D-NS
- GW3000D-NS
- GW3600D-NS
- GW4200D-NS
- GW5000D-NS
- GW6000D-NS
- GW3000T-DS
- GW3600T-DS
- GW4200T-DS
- GW5000T-DS
- GW6000T-DS

DNS G3 Series:

- GW3000-DNS-30
- GW3600-DNS-30
- GW4200-DNS-30
- GW5000-DNS-30
- GW6000-DNS-30

SDT-G2 series:

- GW17KT-DT
- GW20KT-DT
- GW25KT-DT
- GW4K-DT
- GW5K-DT
- GW6K-DT
- GW8K-DT
- GW10KT-DT
- GW12KT-DT
- GW15KT-DT
- GW17KT-DT
- GW20KT-DT
- GW25KT-DT

SDT-G2+ series:

- GW4000-SDT-20
- GW5000-SDT-20
- GW6000-SDT-20

- GW8000-SDT-20
- GW10K-SDT-20
- GW12K-SDT-20
- GW12KLV-SDT-20
- GW15K-SDT-20
- GW17K-SDT-20
- GW20K-SDT-20

SDT/LV SDT series:

- GW12KN-DT
- GW15KN-DT
- GW17KN-DT
- GW20KN-DT
- GW12KLN-DT

SMT series:

- GW25K
- GW29.9K-MT
- GW30K-MT
- GW36K-MT
- GW50KS-MT
- GW60KS-MT

LV SMT series:

- GW12KLV-MT
- GW15KLV-MT
- GW20KLV-MT
- GW30KLS-MT
- GW35KLS-MT

MT series:

- GW50KN-MT
- GW60KN-MT
- GW50KBF-MT
- GW60KBF-MT
- GW75KBF-MT
- GW80KBF-MT
- GW70KHV-MT
- GW80KHV-MT
- GW75K-MT
- GW80K-MT

LV-MT series:

- GW30KLV-MT
- GW35KLV-MT
- GW50KLV-MT

HT series:

- GW73KLV-HT
- GW75K-HT
- GW80K-HT

- GW100K-HT
- GW110K-HT
- GW120K-HT
- GW136K-HTH
- GW225K-HT
- GW250K-HT
- GW225KN-HT
- GW250KN-HT

MS series:

- GW5000-MS
- GW6000-MS
- GW7000-MS
- GW8500-MS
- GW9000-MS
- GW10K-MS

10.3.5 GROWATT

MAC series:

- MAC 15KTL3-XL
- MAC 20KTL3-XL
- MAC 22KTL3-XL
- MAC 25KTL3-XL
- MAC 30KTL3-XL
- MAC 36KTL3-XL
- MAC 30KTL3-X-LV
- MAC 40KTL3-X-LV
- MAC 50KTL3-X-LV
- MAC 60KTL3-X-LV
- MAC 50KTL3-X-MV
- MAC 60KTL3-X-MV
- MAC 70KTL3-X-MV

MAX series:

- MAX 50KTL3-LV
- MAX 60KTL3-LV
- MAX 70KTL3-LV
- MAX 80KTL3-LV
- MAX 100KTL3-X-LV
- MAX 110KTL3-X-LV
- MAX 120KTL3-X-LV
- MAX 125KTL3-X-LV
- MAX 185KTL3-X-HV
- MAX 216KTL3-X-HV
- MAX 250KTL3-X-HV
- MAX 253KTL3-X-HV

MIC series:

- MIC-750TL-X

- MIC-1000TL-X
- MIC-1500TL-X
- MIC-2000TL-X
- MIC-2500TL-X
- MIC-3000TL-X
- MIC-3300TL-X

MID series:

- MID-15KTL3-X
- MID-17KTL3-X
- MID-20KTL3-X
- MID-22KTL3-X
- MID-25KTL3-X
- MID-25KTL3-X1
- MID-30KTL3-X
- MID-33KTL3-X
- MID-36KTL3-X
- MID-40KTL3-X

MIN series:

- MIN 2500TL-X
- MIN 3000TL-X
- MIN 3600TL-X
- MIN 4200TL-X
- MIN 4600TL-X
- MIN 5000TL-X
- MIN 6000TL-X

S series:

- 750-S
- 1000-S
- 1500-S
- 2000-S
- 3000-S

SPA TL-BL series:

- SPA 1000TL-BL
- SPA 2000TL-BL
- SPA 3000TL-BL

SPA TL3-BH series:

- SPA 4000TL3-BH
- SPA 5000TL3-BH
- SPA 6000TL3-BH
- SPA 7000TL3-BH
- SPA 8000TL3-BH
- SPA 10000TL3-BH

SPH TL3 BH-UP series:

- SPH 4000TL3 BH-UP
- SPH 5000TL3 BH-UP

- SPH 6000TL3 BH-UP
- SPH7000TL3 BH-UP
- SPH8000TL3 BH-UP
- SPH10000TL3 BH-UP

SPH series:

- SPH3000
- SPH3600
- SPH4000
- SPH4600
- SPH5000
- SPH6000

TL3-S series:

- 3000TL3-S
- 4000TL3-S
- 5000TL3-S
- 6000TL3-S
- 7000TL3-S
- 8000TL3-S
- 9000TL3-S
- 10000TL3-S
- 11000TL3-S
- 12000TL3-S
- 13000TL3-S
- 15000TL3-S
- 17000TL3-S
- 20000TL3-S
- 25000TL3-S
- 30000TL3-S
- 33000TL3-S
- 50000TL3-S

TL3-NS series:

- 40000TL3-NS

MLT-S series:

- 2500MTL-S
- 3000MTL-S
- 3600MTL-S
- 4200MTL-S
- 5000MTL-S
- 5500MTL-S

MTLP-S series:

- 8000MTLP-S
- 9000MTLP-S
- 10500MTLP-S

UE series:

- 4000UE

- 5000UE
- 6000UE
- 7000UE
- 8000UE
- 9000UE
- 10000UE
- 12000UE
- 18000UE
- 20000UE

10.3.6 Huawei

- SUN2000-100KTL-H0
- SUN2000-100KTL-H1
- SUN2000-100KTL-H2
- SUN2000-100KTL-INM0
- SUN2000-100KTL-M0
- SUN2000-100KTL-M1
- SUN2000-100KTL-USH0
- SUN2000-105KTL-H1
- SUN2000-10KTL
- SUN2000-10KTL-M0
- SUN2000-10KTL-M1
- SUN2000-10KTL-M2
- SUN2000-10KTL-USL0
- SUN2000-11.4KTL-USL0
- SUN2000-110KTL-M0
- SUN2000-111KTL-NHM0
- SUN2000-125KTL-JPH0
- SUN2000-125KTL-M0
- SUN2000-12KTL
- SUN2000-12KTL-M0
- SUN2000-12KTL-M1
- SUN2000-12KTL-M2
- SUN2000-15KTL-M0
- SUN2000-15KTL-M2
- SUN2000-15KTL-M3
- SUN2000-168KTL-H1
- SUN2000-175KTL-H0
- SUN2000-17KTL-M0
- SUN2000-17KTL-M2
- SUN2000-17KTL-M3
- SUN2000-185KTL-H1
- SUN2000-185KTL-INH0
- SUN2000-193KTL-H0
- SUN2000-196KTL-H0
- SUN2000-196KTL-H3
- SUN2000-200KTL-H2
- SUN2000-200KTL-H3
- SUN2000-20KTL-M0
- SUN2000-20KTL-M2
- SUN2000-20KTL-M3
- SUN2000-215KTL-H0

- SUN2000-215KTL-H3
- SUN2000-22KTLUS
- SUN2000-23KTL
- SUN2000-23KTL-M3
- SUN2000-24.5KTL
- SUN2000-24.7KTL-JP
- SUN2000-25KTL-NAM3
- SUN2000-25KTLUS
- SUN2000-28KTL
- SUN2000-28KTL-M3
- SUN2000-29.9KTL
- SUN2000-29.9KTL-M3
- SUN2000-2KTL-L0
- SUN2000-3.8KTL-USL0
- SUN2000-30KTLA
- SUN2000-30KTL-M3
- SUN2000-30KTL-NAM3
- SUN2000-30KTLUS
- SUN2000-33KTL
- SUN2000-33KTLA
- SUN2000-33KTLE001
- SUN2000-33KTL-JP
- SUN2000-33KTL-NAM3
- SUN2000-33KTLUS
- SUN2000-36KTL
- SUN2000-36KTL-M3
- SUN2000-36KTL-NAM3
- SUN2000-36KTLUS
- SUN2000-3KTL-CNL0
- SUN2000-3KTL-L0
- SUN2000-3KTL-M0
- SUN2000-3KTL-M1
- SUN2000-4.95KTL-JPL0
- SUN2000-40KTL
- SUN2000-40KTL-JP
- SUN2000-40KTL-M3
- SUN2000-40KTL-NAM3
- SUN2000-40KTLUS
- SUN2000-42KTL
- SUN2000-42KTL-M3
- SUN2000-43KTL-IN-C1
- SUN2000-43KTL-INM3
- SUN2000-44KTL-M3
- SUN2000-45KTLUS-HV-D0
- SUN2000-4KTL-CNL0
- SUN2000-4KTL-L0
- SUN2000-4KTL-M0
- SUN2000-4KTL-M1
- SUN2000-50KTL
- SUN2000-50KTLC1
- SUN2000-50KTL-JPM0
- SUN2000-50KTL-JPM1
- SUN2000-50KTL-M0
- SUN2000-50KTL-M3

- SUN2000-55KTLHV-D1
- SUN2000-55KTLHV-D1-001
- SUN2000-55KTLIN-HV-D1
- SUN2000-5KTL-CNL0
- SUN2000-5KTL-L0
- SUN2000-5KTL-M0
- SUN2000-5KTL-M1
- SUN2000-5KTL-USL0
- SUN2000-60KTLHV-D1
- SUN2000-60KTLHV-D1-001
- SUN2000-60KTL-M0
- SUN2000-63KTL-JPH0
- SUN2000-63KTL-JPM0
- SUN2000-65KTL-M0
- SUN2000-6KTL-CNL0
- SUN2000-6KTL-M0
- SUN2000-6KTL-M1
- SUN2000-7.6KTL-USL0
- SUN2000-70KTL-C1
- SUN2000-70KTL-INM0
- SUN2000-75KTL-C1
- SUN2000-8KTL
- SUN2000-8KTL-M0
- SUN2000-8KTL-M1
- SUN2000-8KTL-M2
- SUN2000-90KTL-H0
- SUN2000-90KTL-H1
- SUN2000-90KTL-H2
- SUN2000-95KTL-INH0
- SUN2000-95KTL-INH1
- SUN2000-9KTL-USL0
- SUN2000L-2KTL
- SUN2000L-3.68KTL
- SUN2000L-3KTL
- SUN2000L-3KTL-CN
- SUN2000L-3KTL-CN-4G
- SUN2000L-4.125KTL-JP
- SUN2000L-4.6KTL
- SUN2000L-4.95KTL-JP
- SUN2000L-4KTL
- SUN2000L-4KTL-CN
- SUN2000L-4KTL-CN-4G
- SUN2000L-5KTL
- SUN2000L-5KTL-CN
- SUN2000L-5KTL-CN-4G

10.3.7 INGETEAM

1Play HF series:

- SUN 1Play 2.5HF
- SUN 1Play 2.7HF
- SUN 1Play 3HF

- SUN 1Play 3.3HF
- SUN 1Play 3.68HF
- SUN 1Play 4.6HF
- SUN 1Play 5HF
- SUN 1Play 5.5HF
- SUN 1Play 6HF

1Play TL-M series:

- SUN 1Play 2.5TL M
- SUN 1Play 2.7TL M
- SUN 1Play 3TL M
- SUN 1Play 3.3TL M
- SUN 1Play 3.68TL M
- SUN 1Play 4.6TL M
- SUN 1Play 5TL M
- SUN 1Play 5.5TL M
- SUN 1Play 6TL M

3Play series:

- SUN 20TL
- SUN 33TL
- SUN 20TL M
- SUN 33TL M
- SUN 40TL M480
- SUN 100TL
- SUN 160TL

Power Block series:

- SUN 830TL B300
- SUN 1000TL B360
- SUN 1070TL B385
- SUN 1110TL B400
- SUN 1140TL B410
- SUN 1165TL B420
- SUN 1190TL B430
- SUN 1220TL B440
- SUN 1250TL B450
- SUN 1275TL B460
- SUN 1170TL B450
- SUN 1400TL B540
- SUN 1500TL B578
- SUN 1560TL B600
- SUN 1600TL B615
- SUN 1640TL B630
- SUN 1665TL B640
- SUN 1690TL B650
- SUN 1740TL B670
- SUN 1800TL B690

Power Max series:

- SUN 250 TL
- SUN 315HE TL

- SUN 365HE TL
- SUN 375 TL
- SUN 400HE TL
- SUN 420HE TL
- SUN 500 TL
- SUN 500HE TL
- SUN 550HE TL
- SUN 600HE TL
- SUN 625HE TL
- SUN 630HE TL
- SUN 730HE TL
- SUN 800HE TL
- SUN 840HE TL

10.3.8 KACO

Powador series:

- Powador 2500XI
- Powador 3600XI
- Powador 4000XI
- Powador 4500XI
- Powador 5000XI
- Powador 4000 Supreme
- Powador 2002
- Powador 3002
- Powador 4202
- Powador 5002
- Powador 6002
- Powador 6400XI
- Powador 6650XI
- Powador 7200XI
- Powador 8000XI
- Powador 6400
- Powador 6650
- Powador 7200
- Powador 8000 Supreme
- Powador 25000XI
- Powador 30000XI
- Powador 33000XI
- Powador Park
- Powador 10.0 TL3
- Powador 12.0 TL3
- Powador 14.0 TL3
- Powador 30.0 TL3
- Powador 37.5 TL3
- Powador 39.0 TL3
- Powador 72.0 TL3
- Powador 12.0 TR3
- Powador 14.0 TR3
- Powador 18.0 TR3
- Powador XP100-HV
- Powador XP200-HV

- Powador XP 250-HV
- Powador XP200-HV-TL
- Powador XP250-HV-TL
- Powador XP350-HV-TL

BluePlanet TL3 series:

- Blueplanet 3.0 TL3
- Blueplanet 4.0 TL3
- Blueplanet 5.0 TL3
- Blueplanet 6.5 TL3
- Blueplanet 7.5 TL3
- Blueplanet 8.6 TL3
- Blueplanet 9.0 TL3
- Blueplanet 10.0 TL3
- Blueplanet 15.0 TL3
- Blueplanet 20.0 TL3
- Blueplanet 50.0 TL3
- Blueplanet 50.0 TL3 RP only
- Blueplanet 87.0 TL3
- Blueplanet 92.0 TL3
- Blueplanet 105 TL3
- Blueplanet 110TL3-US
- Blueplanet 125 TL3
- Blueplanet 125TL3-US
- Blueplanet 137 TL3
- Blueplanet 150 TL3
- Blueplanet 155 TL3
- Blueplanet 165 TL3

BluePlanet Gridsave series:

- Gridsave 50.0 TL3-S
- Gridsave 92.0 TL3-S
- Gridsave 110 TL3-S
- Gridsave 137 TL3-S

NX1 series:

- 3.0 NX1 M2
- 3.7 NX1 M2
- 4.0 NX1 M2
- 5.0 NX1 M2

NX3 series:

- 3.0 NX3 M2
- 5.0 NX3 M2
- 8.0 NX3 M2
- 10.0 NX3 M2
- 15.0 NX3 M2
- 20.0 NX3 M2
- 25 NX3 M3
- 30 NX3 M3
- 33 NX3 M3
- 100 NX3 M8

- 125 NX3 M8

10.3.9 Kostal

PIKO 3.0-20 series:

- PIKO 3.0
- PIKO 4.2
- PIKO 4.6
- PIKO 5.5
- PIKO 7
- PIKO 8.5
- PIKO 10
- PIKO 12
- PIKO 15
- PIKO 17
- PIKO 20

PIKO MP PLUS series:

- PIKO MP PLUS 1.5-1
- PIKO MP PLUS 2.0-1
- PIKO MP PLUS 2.5-1
- PIKO MP PLUS 3.0-1
- PIKO MP PLUS 3.0-2
- PIKO MP PLUS 3.6-1
- PIKO MP PLUS 3.6-2
- PIKO MP PLUS 4.6-2
- PIKO MP PLUS 5.0-2

PIKO CI series:

- PIKO CI 30 kW
- PIKO CI 50 kW
- PIKO CI 60 kW

PIKO IQ series:

- PIKO IQ 3.0 kW
- PIKO IQ 4.2 kW
- PIKO IQ 5.5 kW
- PIKO IQ 7.0 kW
- PIKO IQ 8.5 kW
- PIKO IQ 10.0 kW

PIKO PLENTICORE PLUS series:

- PIKO PLENTICORE PLUS 3.0
- PIKO PLENTICORE PLUS 4.2
- PIKO PLENTICORE PLUS 5.5
- PIKO PLENTICORE PLUS 7.0
- PIKO PLENTICORE PLUS 8.5
- PIKO PLENTICORE PLUS 10

10.3.10 SMA

SUNNY BOY series

SUNNY TRI POWER series

SUNNY MINI central series

10.3.11 Solis

Single phase series:

- S6-GR1P2.5K
- S6-GR1P3K
- S6-GR1P3.6K
- S6-GR1P4K
- S6-GR1P4.6K
- S6-GR1P5K
- S6-GR1P6K

Three phases series:

- Solis-15K-LV
- Solis-20K-LV
- Solis-40K
- Solis-50K
- Solis-50K-HV
- Solis-60K-HV

Energy Storage Inverters RHI / RAI series:

- RHI-3K-48ES
- RHI-3.6K-48ES
- RHI-4.6K-48ES
- RHI-5K-48ES
- RAI-3K-48ES-5G
- RHI-3K-48ES-5G
- RHI-3.6K-48ES-5G
- RHI-4.6K-48ES-5G
- RHI-5K-48ES-5G
- RHI-6K-48ES-5G

Single phase US 4G series:

- Solis-1P6K-4G-US
- Solis-1P7K-4G-US
- Solis-1P7.6K-4G-US
- Solis-1P8K-4G-US
- Solis-1P8.6K-4G-US
- Solis-1P9K-4G-US
- Solis-1P10K-4G-US

Three phases US 5G series:

- Solis-75K-5G-US

- Solis-80K-5G-US
- Solis-90K-5G-US
- Solis-100K-5G-US

Mini Inverter 4G series:

- Solis-mini-700-4G
- Solis-mini-1000-4G
- Solis-mini-1500-4G
- Solis-mini-2000-4G
- Solis-mini-2500-4G
- Solis-mini-3000-4G
- Solis-mini-3600-4G

Single phase K-4G series:

- Solis-1P2.5K-4G
- Solis-1P3K-4G
- Solis-1P3.6K-4G
- Solis-1P4K-4G
- Solis-1P4.6K-4G
- Solis-1P5K-4G
- Solis-1P6K-4G
- Solis-1P9K-4G
- Solis-1P10K-4G

Three phases K-4G series:

- Solis-60K-4G
- Solis-3P3K-4G
- Solis-3P4K-4G
- Solis-3P5K-4G
- Solis-3P6K-4G
- Solis-3P8K-4G
- Solis-3P9K-4G
- Solis-3P10K-4G
- Solis-3P12K-4G
- Solis-3P15K-4G
- Solis-3P17K-4G
- Solis-3P20K-4G
- Solis-70K-HV-4G

Single phase K-5G series:

- Solis-1P7K-5G
- Solis-1P8K-5G

Three phases 5G series:

- Solis-25K-5G
- Solis-30K-5G
- Solis-33K-5G
- Solis-36K-5G
- Solis-40K-5G
- Solis-80K-5G
- Solis-100K-5G

- Solis-40K-HV-5G
- Solis-50K-HV-5G
- Solis-100K-HV-5G
- Solis-110K-HV-5
- Solis-125K-EHV-5G
- Solis-136K-EHV-5G
- Solis-250K-EHV-5G

US series:

- Solis-25K-US
- Solis-30K-US
- Solis-36K-US
- Solis-40K-US
- Solis-50K-US
- Solis-36K-US-F
- Solis-40K-US-F
- Solis-50K-US-F
- Solis-60K-US-F
- Solis-66K-US-F
- Solis-25K-US-SW
- Solis-25K-US-F-SW
- Solis-30K-US-SW
- Solis-30K-US-F-SW
- Solis-36K-US-SW
- Solis-36K-US-F-SW
- Solis-40K-US-SW
- Solis-40K-US-F-SW
- Solis-50K-US-SW
- Solis-50K-US-F-SW
- Solis-60K-US-F-SW
- Solis-66K-US-F-SW

10.3.12 SUNGROW

- SG5.5RS-JP
- SG0.7RS-S
- SG1.0RS-S
- SG1.5RS-S
- SG2.0RS-S
- SG2.5RS-S
- SG3.0RS-S
- SG3.0RS
- SG3.6RS
- SG4.0RS
- SG5.0RS
- SG6.0RS
- SG8.0RS
- SG9.0RS
- SG10RS
- SG5.0RS-ADA
- SG3.0RT
- SG4.0RT

- SG5.0RT
- SG6.0RT
- SG7.0RT
- SG8.0RT
- SG10RT
- SG12RT
- SG15RT
- SG17RT
- SG20RT
- SG3.0 RT-P2
- SG4.0 RT-P2
- SG5.0 RT-P2
- SG6.0 RT-P2
- SG7.0 RT-P2
- SG8.0 RT-P2
- SG10 RT-P2
- SG12 RT-P2
- SG15 RT-P2
- SG17 RT-P2
- SG20RT-P2
- SG 25CX-P2
- SG 30CX-P2
- SG 33CX-P2
- SG 36CX-P2
- SG 40CX-P2
- SG 50CX-P2
- SG 75CX-P2
- SG 110CX-P2
- SG 125CX-P2
- SG30KTL-M
- SG30KTL-M-V31
- SG33KTL-M
- SG36KTL-M
- SG33K3J
- SG49K5J
- SG34KJ
- LP_P34KSG
- SG49.5CX-JP
- SG50KTL-M-20
- SG60KTL
- SG80KTL
- SG80KTL-20
- SG60KU-M
- SG5KTL-MT
- SG6KTL-MT
- SG8KTL-M
- SG10KTL-M
- SG10KTL-MT
- SG12KTL-M
- SG15KTL-M
- SG17KTL-M
- SG20KTL-M
- SG80KTL-M
- SG85BF

- SG80HV
- SG80BF
- SG110HV-M
- SG111HV
- SG125HV
- SG125HV-20
- SG25CX-SA
- SG30CX
- SG33CX
- SG40CX
- SG50CX
- SG36CX-US
- SG60CX-US
- SG75CX
- SG100CX
- SG100CX-JP
- SG110CX
- SG136TX
- SG225HX
- SG250HX
- SG250HX-IN
- SG250HX-US
- SG285HX
- SG350HX
- SG125HX
- SG125HX-JP
- SG333HX
- SG320HX
- SG30KTL
- SG10KTL
- SG12KTL
- SG15KTL
- SG20KTL
- SG30KU
- SG36KTL
- SG36KU
- SG40KTL
- SG40KTL-M
- SG50KTL-M
- SG60KTL-M
- SG60KU

11 Offices and support

SPAIN

C/ Alejandro Sánchez 109
28019 Madrid

Telephone: +34.915602737
E-mail: contact@webdyn.com

FRANCE

26 Rue des Gaudines
78100 Saint-Germain-en-Laye

Telephone: +33.139042940
E-mail: contact@webdyn.com

INDIA

803-804 8th floor, Vishwadeep Building
District Centre, Janakpurt, 110058 Delhi

Telephone: +91.1141519011
E-mail: contact@webdyn.com

PORTUGAL

Av. Coronel Eduardo Galhardo 7-1ºC
1170-105 Lisbon

Telephone: +351.218162625
E-mail: comercial@lusomatrix.pt

SUPPORT

Madrid

Telephone: +34.915602737
E-mail: iotsupport@matrix.es

Saint-Germain-en-Laye

Telephone: +33.139042940
E-mail: support@webdyn.com

Delhi

Telephone: +91.1141519011
E-mail: support-india@webdyn.com