webdyn
Flexitron group

WebdynSunPM

Application Note

Decoupling

www.webdyn.com

Introduction

This application note describes how to implement the "**Decoupling**" script
This script allows the internal relay of the inverters to be controlled on the digital inputs, whether by status reading or pulse reading.
It is also possible to call up the inverter relay control function directly via FTP or MQTT command file or post requests.

The use of this script requires the purchase of a license, please contact Webdyn's sales department (https://www.webdyn.com/contact) in order to obtain this license

This service makes it possible to meet various needs such as:

- The Simplified Decoupling Device (DDS) or the Simplified Information and Operations Exchange Device (DEIE) for which the inverters must be coupled/decoupled according to impulse commands received from a meter and the command acknowledged by dry contact via the relay.
- Management of negative rates.
Indeed, when spot electricity prices become negative, the producer must be able to turn off its inverters to avoid being billed by the electricity network operator. The goal is to set up an automated system that can record critical time slots and cut production in a timely manner.

Prerequisites

The WebdynSunPM must be updated to firmware version 4.6.5 or higher.
The script is present in the WebdynSunPM script library as of version 5.0.10.
However, it can be retrieved by following the link below and imported via the web interface or server
https://www.webdyn.com/download/Decouplage.zip

A "Decoupling" license specific to the WebdynSunPM used is required
Please contact the sales department (https://www.webdyn.com/contact) to obtain it, you will be asked for your gateway ID.

Knowledge of the basic principles of how WebdynSunPM works is highly recommended.
Refer to the WebdynSunPM user manual.

The settings described below on the UPS definition files are already made in most of the files integrated into the WebdynSunPM internal library.
In such cases, the use of the script does not require any additional specific settings on the definition files.

Inverter settings

In each definition file used by UPS connected to the hub, the following is required:

- ## Category (equipment identification)

In the header of the definition file, the category field (first row, 2nd column) must be defined with the name "Inverter". It is this name that identifies all the inverters to be checked.

- ## Tags (identification des variables)

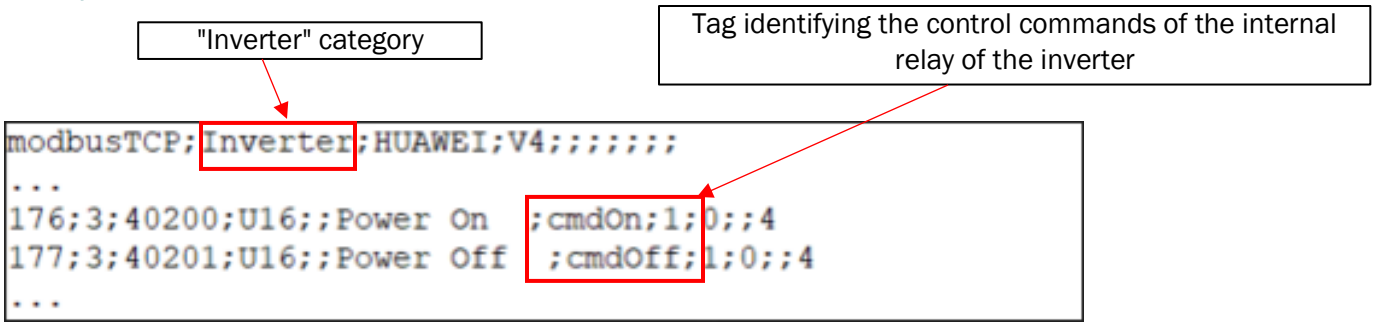
All equipment identified by the "Inverter" category must have the following tags:
Tags can be added to the appropriate definition file for each device directly from the "System/Definition File Library" page of the web interface
Otherwise, the "Tags" must be filled in column G (fields 7) of the definition file extracted from the WebdynSunPM

" **cmdOn** " tag: Identifies the variable that will receive a value of 1 during an inverter pairing command and 0 during a decoupling command.

" **CmdOff** " Tag: Identifies the variable that will receive a value of 0 during an inverter pairing command and 1 during a decoupling command.

It is not necessary for these two tags to be present. If the inverter only has a control register, it will be necessary to associate it with the appropriate tag:
If the logic is "direct": 1 to activate and 0 to deactivate then the "cmdOn" tag will be appropriate.
If the logic is "reverse": 0 to activate and 1 to deactivate then the "cmdOff" tag should be used.

Example:



In the event that the inverter to be controlled does not allow its internal relay to be controlled from the Modbus registers, it is possible to use the tags below to order the inverter's production to zero percent. To do this, you need to activate the "UsePwrPpct" option (see § script configuration)

Be careful, inverters above 100KW can retain a residual production of several hundred Watts, even when they receive the zero percent control.

The following tags allow:
Enabling order fulfillment:
WMaxLim_Ena Optional
Zeroing of control ramps
WMaxLimPct_RmpTms Optional
Cancellation of the original power restoration instruction in the absence of a command
WMaxLimPct_RvrtTms Optional

cmdPwrPercent Mandatory

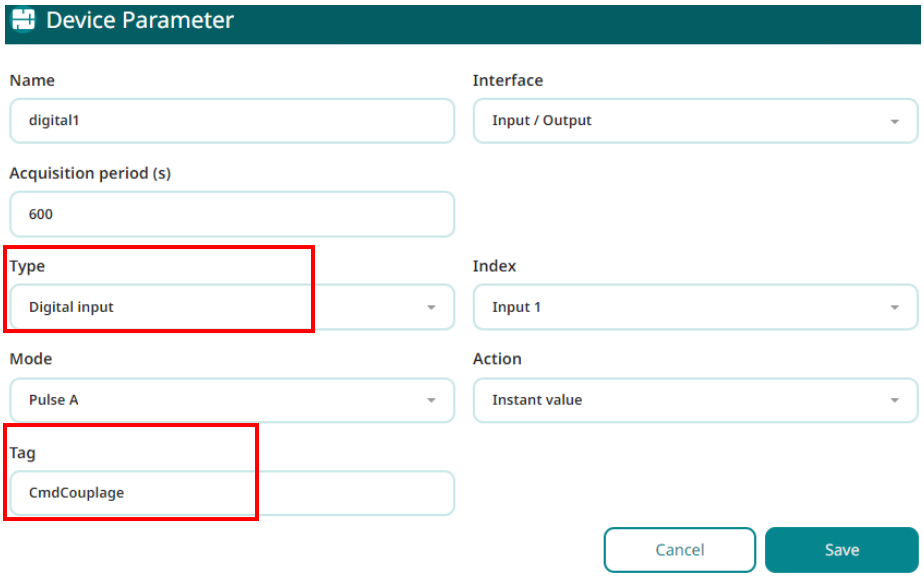
As a last resort, it is possible to control the internal relay of the WebdynSunPM in order to control a general circuit breaker and thus ensure decoupling in all circumstances (see § configuration of the IOs of the WebdynSunPM).

Setting up the IOs of the WebdynSunPM

Setting up these entries is not always necessary. Depending on the use case, the script in question may not need to react to analog commands, as is the case for negative pricing planning where the commands come directly from the monitoring server.

On the other hand, in the case of a Simplified Decoupling Device (DDS) or the Simplified DEIE, the commands come from 2 outputs of the meter. The commands are impulsive in the case of the DDS and static in the case of the simplified DEIE.

In order to monitor a DDS, it is therefore necessary to configure 2 of the WebdynSunPM's numeric inputs in impulse read mode and assign them the **CmdCoupling** and **CmdDecoupling tags**.



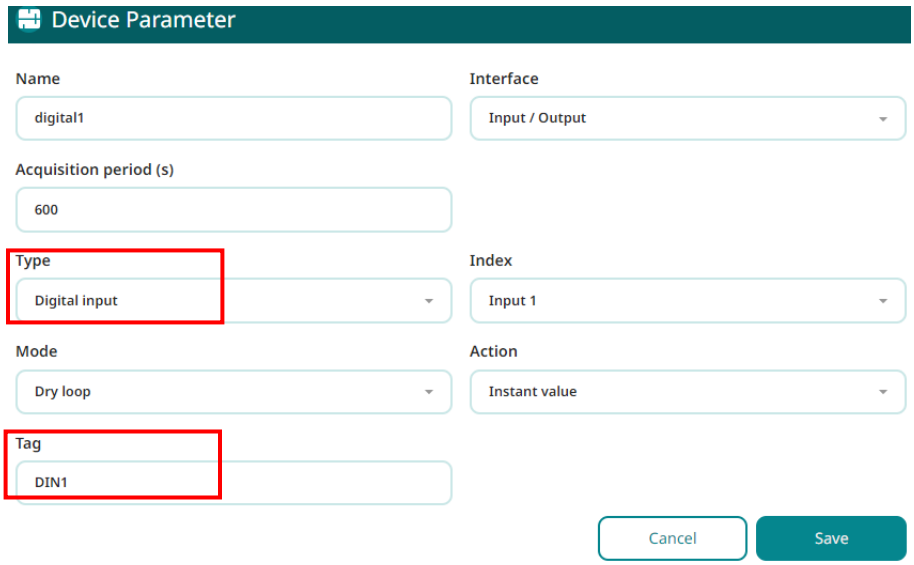

```
io;WebdynSunPM;Webdyn;ioSunPM
1;2;1;1;;digital1;CmdCouplage;1.000000;0.000000;;4
2;2;2;1;;digital2;CmdDecouplage;1.000000;0.000000;;4
3;2;3;1;;digital3;;1.000000;0.000000;;8
4;1;1;1;;analog1;;1.000000;0.000000;°C;4
5;1;2;1;;analog2;;1.000000;0.000000;None;4
6;1;3;1;;analog3;;1.000000;0.000000;°C;4
7;1;4;1;;analog4;;1.000000;0.000000;None;4
8;3;1;;;output;RelayOutput;1.000000;0.000000;;4
```

Detecting a pulse on the input associated with the coupling command tag will cause a 1 to be written on the variables associated with the **cmdOn tag** and a 0 on the variables associated with the **cmdOff tag.**

Detecting a pulse on the input associated with the decoupling command tag will cause a 0 to be written on the variables associated with the **cmdOn tag** and a 1 on the variables associated with the **cmdOff tag**

Alternatively, it is possible to use only one digital input configured as a state readout (dry contact) to which the "**DIN1**" tag will be associated .
This configuration makes it possible, for example, to meet the need to interrupt the inverters in the event of a generator starting signaled by closing a dry contact, or to reduce the number of digital inputs required to manage the DDS, by adding a contactor with a hook-up coil in order to maintain the impulse signal received.

In order to supervise a simplified DEIE, it is necessary to configure 2 of the WebdynSunPM's digital inputs in "Dry loop" mode and assign them the **DIN1** and **DIN2 tags**.



By default, closing the **DIN1 contact** (passing to one) will cause a 1 to be written on the variables associated with the **cmdOn tag** and a 0 on the variables associated with the **cmdOff tag**
Conversely, opening the contact (going to zero) will cause a 0 to be written on the variables associated with the **cmdOn tag** and a 1 on the variables associated with the **cmdOff tag**
However, it is possible to reverse the logic of the digital input by setting the digital input with a gain of -1 and an offset of 1.
The **DIN2** contact works exactly the opposite, the script only takes into account the **DIN1** tag for the execution of commands, the **DIN2** tag is only used to check the consistency of the **DIN1 information.**

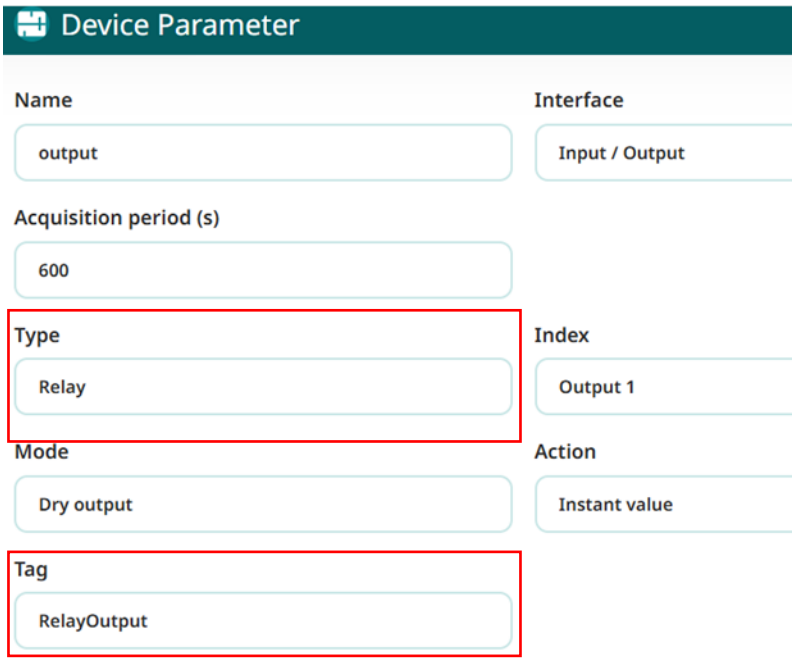
Using the latter mode has an effect as soon as the script starts.
Unlike the pulse mode, which only acts when a pulse is received, the dry contact mode applies the corresponding command to the input state as soon as the script starts. Most often, if nothing is connected, the input state is zero, which causes the inverters to shut down.

In any case, the inverters may not be switched off immediately and depends on the inverter's settings, which can apply discharge ramps in the event of an outage.

A **60s delay** corresponding to the start-up time of the inverters is applied as a result of a coupling command after decoupling. The last order received during this period will be applied at the end of the period.

In order to force the non-injection of inverters that do not integrate the necessary commands to control their internal relay or the power reduction control, it is possible to control the internal relay of the WebdynSunPM so that it activates a Schneider TeSys LC1D type general circuit breaker.

To do this, you need to declare the **RelayOutput tag**, so the internal relay of the WebdynSunPM will respect the same commands as those issued for the **cmdOn tag** of the inverters.



For the connection of the internal relay to the site's main circuit breaker, it may be necessary to use an intermediate relay type Finder 55.32.9.024.0000 in order to comply with the cutting powers of each relay.

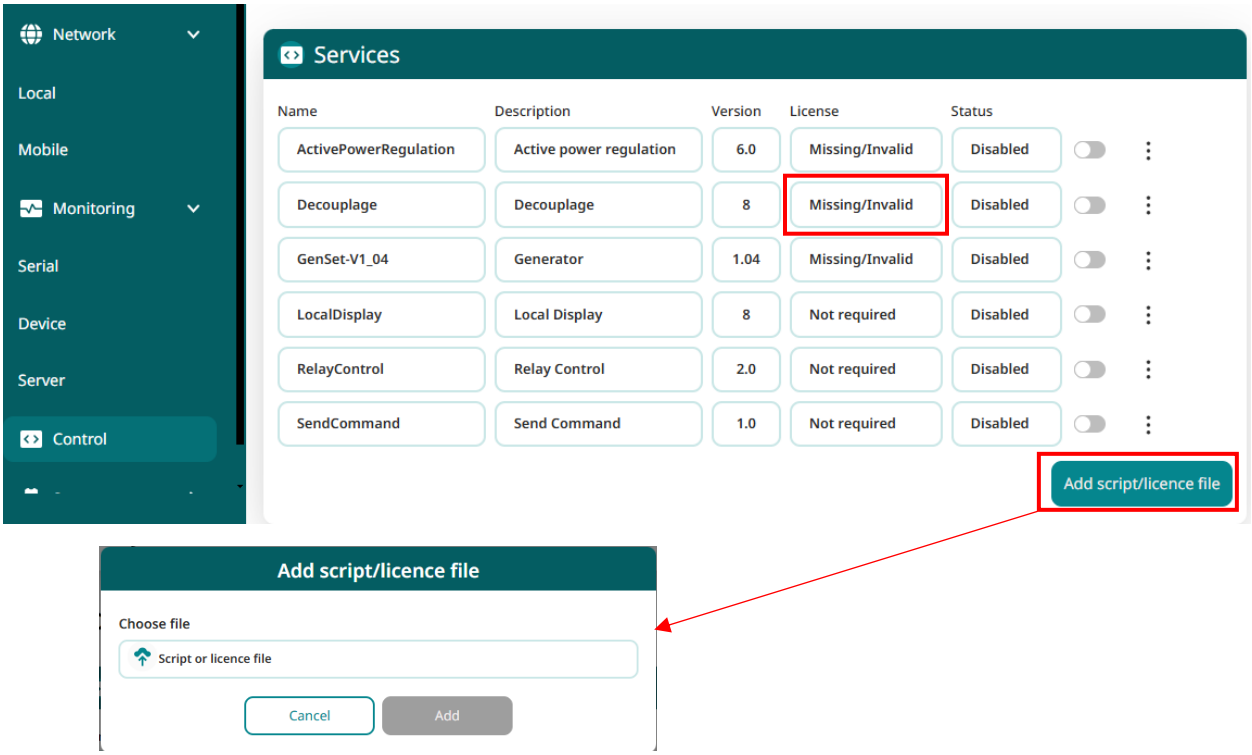
The declaration of the tag associated with the relay is also necessary to indicate the coupling/decoupling of the control unit to a simplified DEIE type device (see § alternative uses)

Script

Loading the script and license

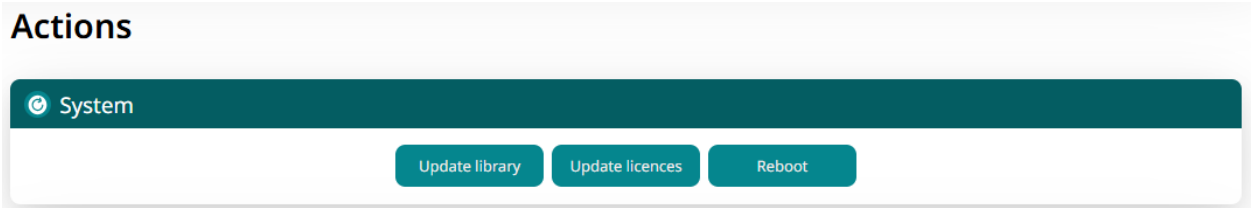
The latest up-to-date version should be retrieved via the following link:
https://www.webdyn.com/download/Decouplage.zip

From the ***control*** page you can load the script by clicking on the "*Add script/licence file*" button



License Integration Verification:

If the license is not loaded into the product, the message "Missing/Invalid" appears in the License column.
The license can be automatically imported by clicking on the "Update licenses" button on the /system/Actions page

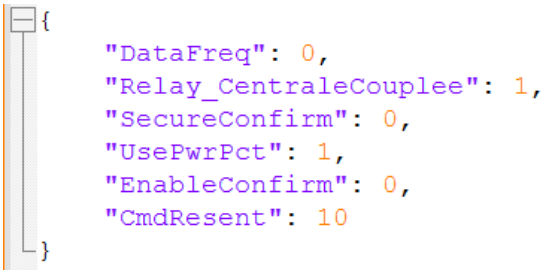


Or by importing the license file that was sent to you by clicking on the "*Add script/licence file*" button.

The "License" field must indicate "Active"

Script Setup

It is possible to modify some parameters of the script to adjust its behavior, the following variables can be adjusted using the "json" structure described below:



```
{
    "DataFreq": 0,
    "Relay_CentraleCouplee": 1,
    "SecureConfirm": 0,
    "UsePwrPct": 1,
    "EnableConfirm": 0,
    "CmdResent": 10
}
```

{"DataFreq":0,"Relay_CentraleCouplee":1,"SecureConfirm":1,"UsePwrPct":1,"EnableConfirm":0,"
CmdResent":1000}

DataFreq : allows you to set the interval (in seconds) for writing the variables of the script (virtual device) in the data file transmitted to the server (0 means no writing in the data file).
Relay_CentraleCouplee : Necessary for the use of the script in the context of the implementation of a simplified DEIE, the option allows the use of the relay to be dedicated to the control of the Coupled Central input of the simplified DEIE, the relay is activated after the commands sent to the inverters.
SecureConfirm: This option is in addition to the **Relay_CentraleCouplee option**, it allows you to condition the action of the relay to indicate the decoupling of the control unit to the output reading of each inverter, which must indicate zero.
UsePwrPpct : The use of this option allows the production control register of each inverter to be controlled in addition to or in the absence of a register to control its internal relay. The applied commands are 100% in case of coupling or 0% in case of decoupling.
EnableConfirm: Enables the sending of an acknowledgment command of the emitted throttling command if the **UsePwrPct** option is used. This behavior is sometimes expected by some inverters.
CmdResent : generally the power of inverters is provided in Watt by inverters, this value must be converted to KW to allow regulation, so this coefficient is set to 1000 by default.

Setting up and starting the script from the remote server

From the remote server, the "<uid>_scl.ini" file allows the configuration and activation of scripts, it is present in the /Config directory

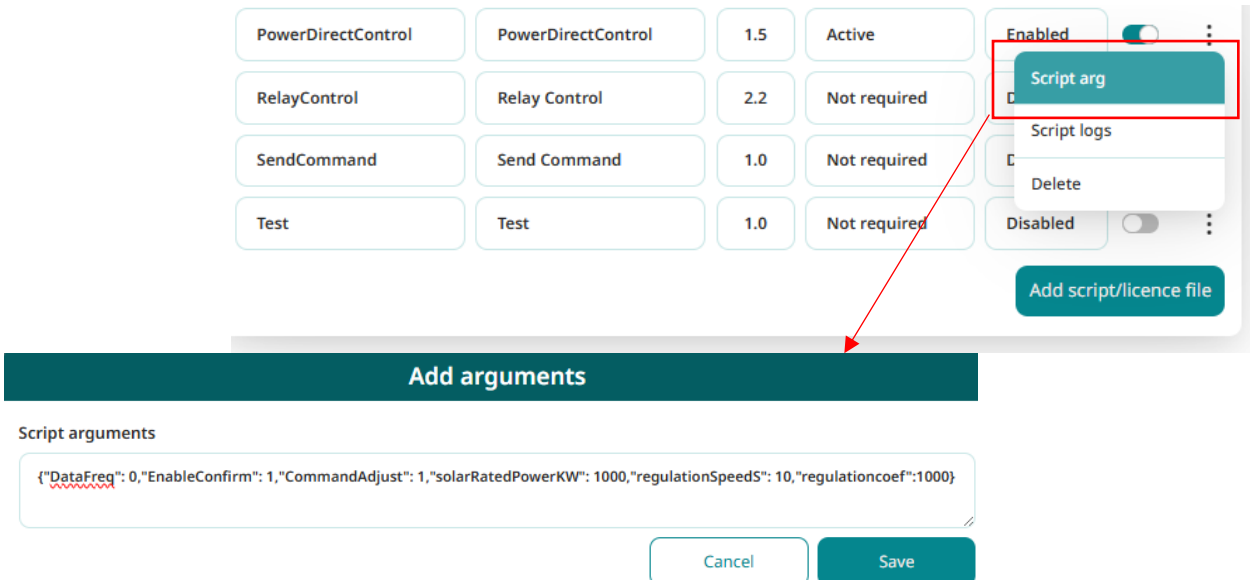
The SCRIPT_Enable[n] **parameter** indicates the working state and allows the activation (=1) and deactivation (=0) of the script identified by the **SCRIPT_File[n] parameter,** which is set to Decoupling.luaw in this case.

The SCRIPT_Args[n] **parameter** of the "<uid>_scl.ini" file allows you to modify

```
SCRIPT_Args[1]={"DataFreq": 0,"Relay_CentraleCouplee": 1,"SecureConfirm": 0,"UsePwrPct": 1,"EnableConfirm": 0,"CmdResent": 1000}
SCRIPT_Enable[1]=1
SCRIPT_File[1]=Decouplage.lua
```

Setting up and starting the script from web pages.

The previously described json structure should be copied directly into the parameter input field if a change to the defaults is needed. If no configuration is required, the default settings are applied.



The script is activated by clicking on the button at the end of the line:

Working principle

The script provides a series of variables that are described in the "**WPMXXXXXX_Script_Decouplage.csv" file** deposited in the /DEF directory of the server.

none	Script	Script	Decouplage							
1			U16		CentraleCouplee	CentraleCouplee	1.000000	0.000000		4
2			U32		TotInvPower	TotInvPower	1.000000	0.000000		4

It describes the variables in the script in the same way that the other files in this directory describe the variables that are accessible on a physical device. In the case of the script, we speak of virtual equipment.
By default, the values associated with these variables are not transmitted to the monitoring servers, the **DataFreq** option described above allows them to be transmitted at the specified frequency.

In the case of the decoupling script, two variables are available, they respectively allow the sum of the active powers of each **TotInvPower inverter to be reported** and the status of the coupling/decoupling setpoint: **CentralCoupled**.

Remote control of the control panel

The script contains a function called **inverterOnOff(state)** used to drive all UPSs.

When this function is called parameter 1, it commands the sending of the value 1 to the variables **tagged cmdOn** and the value 0 to the variables tagged **cmdOff**.

When called with the 0 parameter, it commands the sending of the value 0 to the variables with the **cmdOn tag** and the value 1 to the variables with the cmdOff tag.

This function can be called up by any means made available on the WebdynSunPM

- Command file via FTP server /CMD directory
- Requests via the MQTT server's topic command
- HTTP requests

In the event that the script is used for remote control of the UPS decoupling feature (without using digital inputs), it is recommended to remove the tags associated with IOs. This is because the use of the dry contact mode (DIN1 tag) in particular will have the effect of regularly applying a command reflecting the state of the corresponding digital input that may be in contradiction with the last command issued.

In any case, the inverters may not be switched off immediately and depends on the inverter's settings, which can apply discharge ramps in the event of an outage.

A **60s delay** corresponding to the start-up time of the inverters is applied following a coupling command after decoupling. The last order received during this period will be applied at the end of the period.

Control by sending orders via an FTP server

It is possible to send the command described above by dropping a command file on the FTP server in the /CMD directory. The command will then be executed after the command file is retrieved during the next connection.

The command file must have the following format: **<uid>_cmd.json**

```
[
    {
        "rpcName": "Decouplage.inverterOnOff",
        "parameters": 0,
        "callerId": "1"
    }
]
```

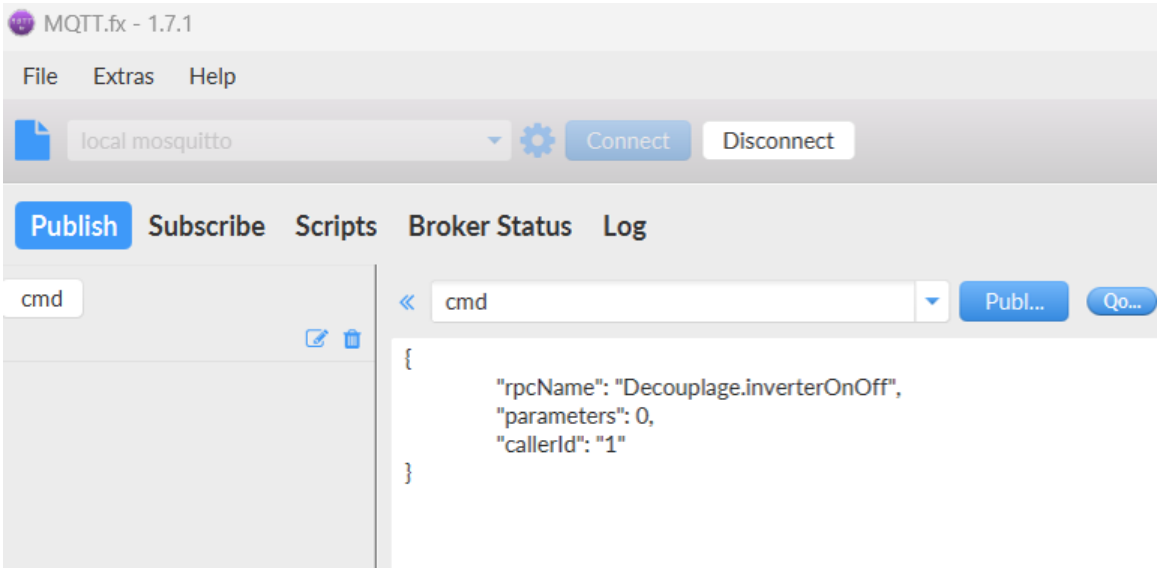
An acknowledgment file (<UID>_ACK_240930_130945.json) deposited at the next connection will indicate the result of the execution of the command.

Control by sending a command via an MQTT server

To configure the connection of the WebdynSunPM to an MQTT server, refer to the user manual §3.2.3.3.4 MQTT page 130. As the MQTT connection does not allow the monitoring of the datalogger configuration, only server 2 can be configured with this type of connection.

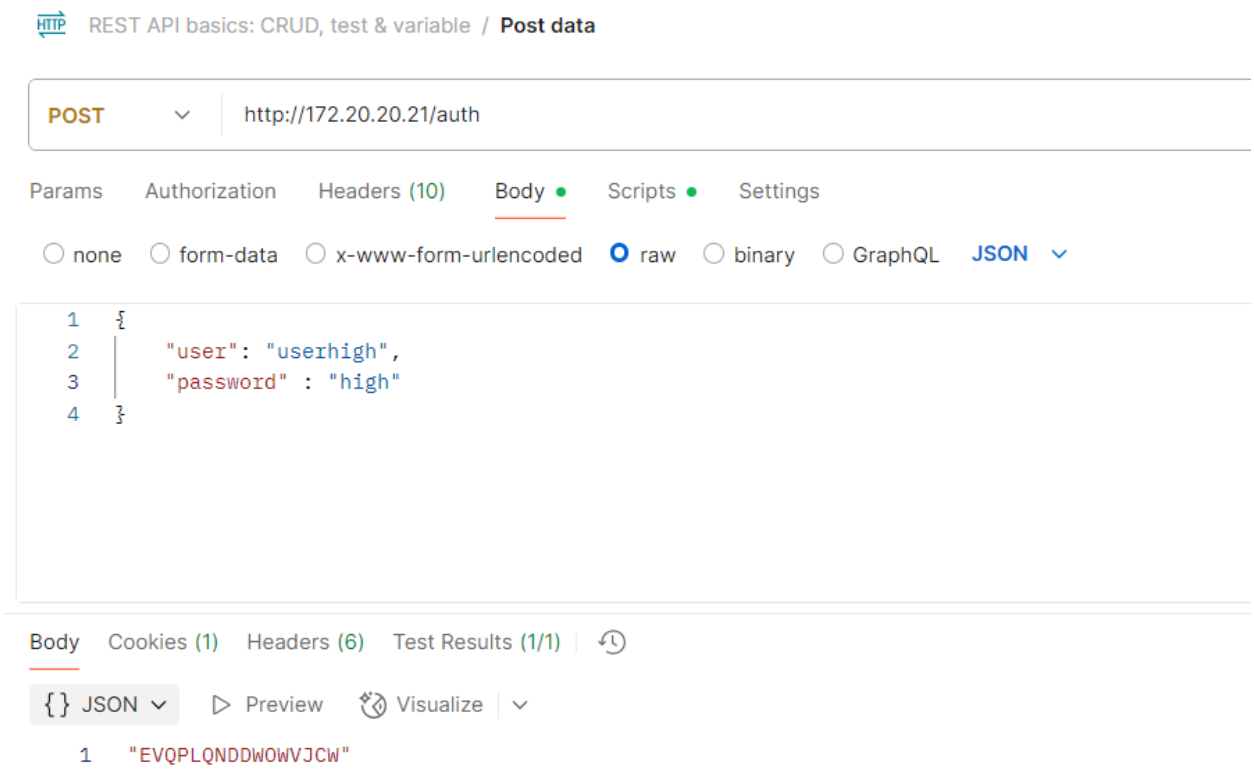
Make sure you have defined a **Command topic** and a **Result topic** in order to allow the receipt of commands and the sending of the result.

The control can be transmitted via software allowing you to subscribe to an MQTT broker such as MQTTfx (https://www.softblade.de/download/).

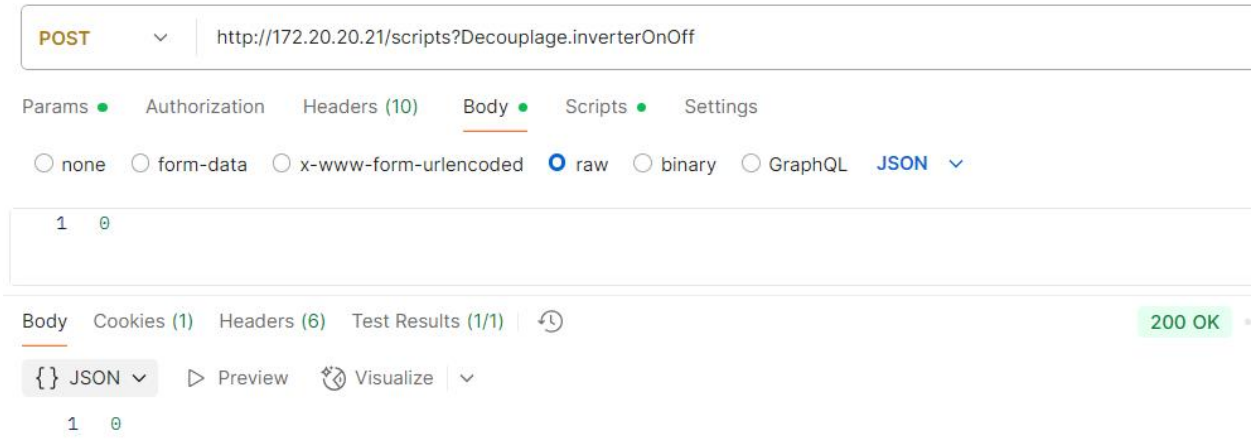


Sending a POST request

Before sending a post command to the WebdynSunPM, it is necessary to log on by sending an authentication request as follows using the Postman API for example (https://web.postman.com/).
Below is the logon command:



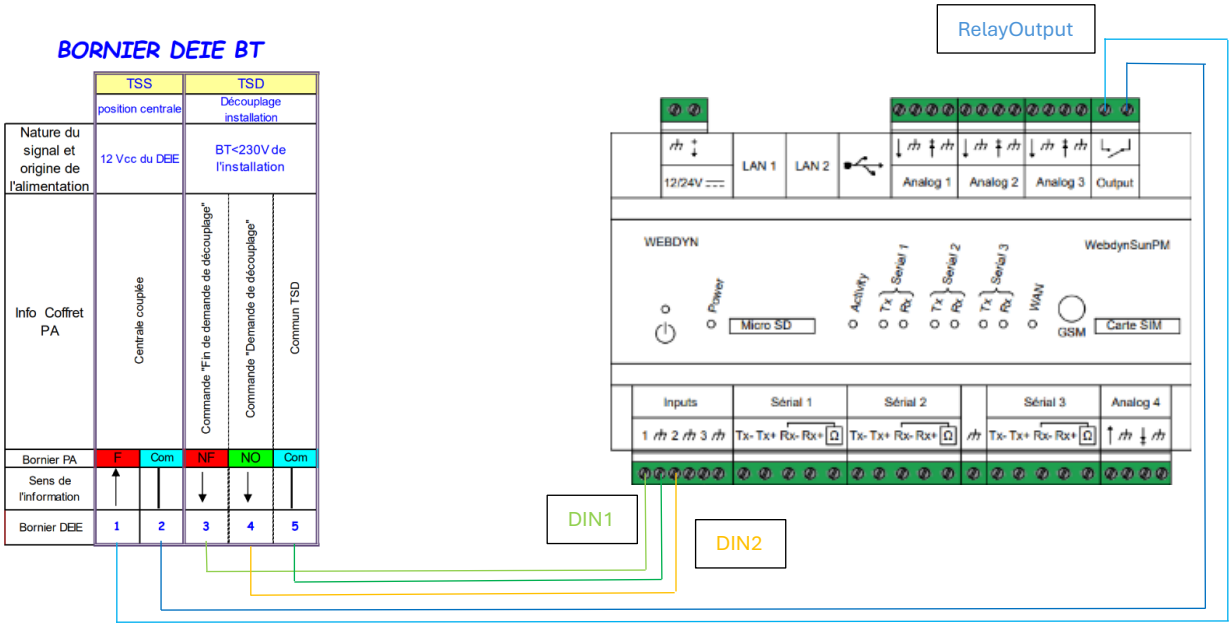
After authentication, you can then call up the command described above.
In the example below, a decoupling command:



Alternative use: DEIE Simplified.

In order to supervise a simplified DEIE, it is necessary to configure 2 of the WebdynSunPM's digital inputs in "Dry loop" mode and assign them the **DIN1** and **DIN2 tags**. (cf **Configuring the IOs of the WebdynSunPM** page 6/7)

By default, closing the **DIN1 contact** (passing to one) will cause a 1 to be written on the variables associated with the **cmdOn tag** and a 0 on the variables associated with the **cmdOff tag**
Conversely, opening the contact (going to zero) will cause a 0 to be written on the variables associated with the **cmdOn tag** and a 1 on the variables associated with the **cmdOff tag**
The **DIN2** contact works exactly the opposite, the script only takes into account the **DIN1** tag for the execution of commands, the **DIN2** tag is only used to check the consistency of the **DIN1 information.**



The following options are available starting with version 13 of the Decoupling script. This use case requires the Relay_CentraleCouplée parameter to be enabled, refer to the Script **Setup paragraph** on page 10 to enable the option.

Relay_CentraleCouplee : the option allows the use of the relay to be dedicated to the control of the Coupled Central input of the simplified DEIE, the relay is activated after all the commands have been sent to the inverters.

SecureConfirm: (optional) This option allows you to condition the relay action to indicate the decoupling of the control unit to the output reading of each inverter, which must indicate zero. The use of this option may result in a longer acknowledgment time since the issuance of the command becomes highly dependent on the data refresh time of each UPS.

Order planning

Starting with version 9 of the script, a series of functions allow you to schedule the execution of the previously described coupling and decoupling commands automatically for the next 48 hours with hourly granularity.

Starting with version 12 of the script, new functions allow quarter-hour planning for the next 72 hours.

Each of the functions described can be controlled via the same mechanisms as those described above:

-FTP command
-MQTT command
-post http command

The functions that enable scheduling are:

Time Configuration:

- **ActivationSetDay** allows you to configure the pairing or decoupling commands that must be issued on the same day.
- **ActivationSetNextDay** allows you to configure the pairing or decoupling commands that should be issued from the next day.
- **ActivationSetDay3** allows you to configure the pairing or decoupling commands that must be issued from the next day.

Quarter-hour configuration:

- **ActivationSetDetailedDay** is used to configure the pairing or decoupling commands that must be issued on the same day.
- **ActivationSetDetailedNextDay** is used to configure the pairing or decoupling commands that should be issued from the next day.
- **ActivationSetDetailedDay3** allows you to configure the pairing or decoupling commands that should be issued from the next day.

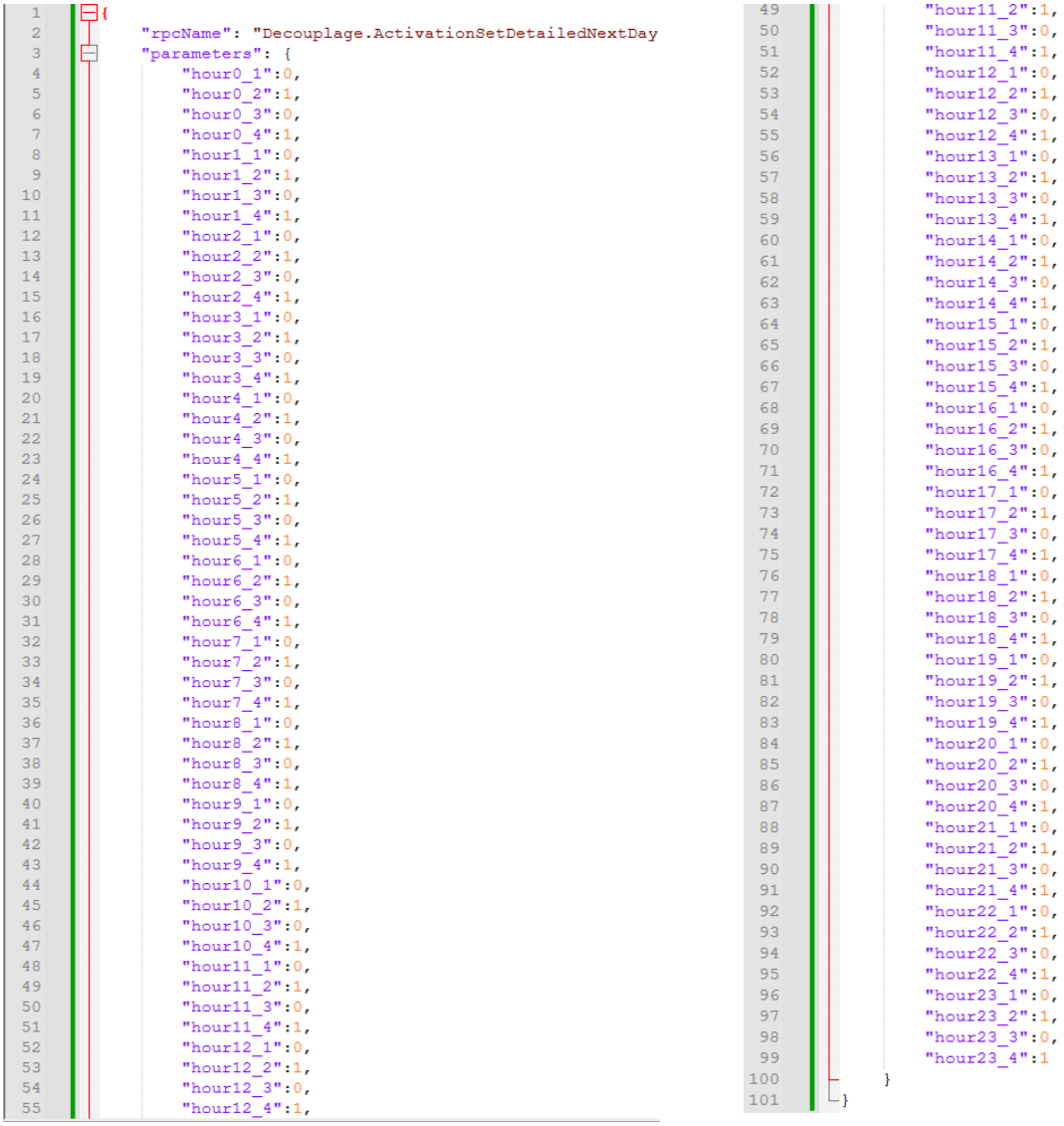
The principle is as follows: once the order has been sent, the order schedule transmitted is applied from the next quarter of an hour, by default a day that has not received a configuration will apply a permanent coupling command.

At the chosen switchover time, the "next day" configuration will replace the "day" configuration and the "day3" configuration will replace the "next day" configuration, the "day3" configuration will receive the default configuration (permanent pairing).
By default, the switch is at midnight.

Thus, the order file below describes the orders that will be sent every hour of the day following that of the order being sent. (next day)
A pairing command will be sent for odd hours and a decoupling command for even hours.


```
{
    "rpcName": "Decouplage.ActivationSetNextDay",
    "parameters": {
        "hour0": 0
        "hour1": 1,
        "hour2": 0,
        "hour3": 1,
        "hour4": 0,
        "hour5": 1,
        "hour6": 0,
        "hour7": 1,
        "hour8": 0,
        "hour9": 1,
        "hour10": 0,
        "hour11": 1,
        "hour12": 0,
        "hour13": 1,
        "hour14": 0,
        "hour15": 1,
        "hour16": 0,
        "hour17": 1,
        "hour18": 0,
        "hour19": 1,
        "hour20": 0,
        "hour21": 1,
        "hour22": 0,
        "hour23": 1
    },
    "callerId": "1"
}
```

The order file below describes the orders that will be sent every quarter of an hour of the day following that the order is sent, with an alternating status every quarter of an hour:



```
{
    "rpcName": "Decouplage.ActivationSetDetailedNextDay
    "parameters": {
        "hour0_1":0,
        "hour0_2":1,
        "hour0_3":0,
        "hour0_4":1,
        "hour1_1":0,
        "hour1_2":1,
        "hour1_3":0,
        "hour1_4":1,
        "hour2_1":0,
        "hour2_2":1,
        "hour2_3":0,
        "hour2_4":1,
        "hour3_1":0,
        "hour3_2":1,
        "hour3_3":0,
        "hour3_4":1,
        "hour4_1":0,
        "hour4_2":1,
        "hour4_3":0,
        "hour4_4":1,
        "hour5_1":0,
        "hour5_2":1,
        "hour5_3":0,
        "hour5_4":1,
        "hour6_1":0,
        "hour6_2":1,
        "hour6_3":0,
        "hour6_4":1,
        "hour7_1":0,
        "hour7_2":1,
        "hour7_3":0,
        "hour7_4":1,
        "hour8_1":0,
        "hour8_2":1,
        "hour8_3":0,
        "hour8_4":1,
        "hour9_1":0,
        "hour9_2":1,
        "hour9_3":0,
        "hour9_4":1,
        "hour10_1":0,
        "hour10_2":1,
        "hour10_3":0,
        "hour10_4":1,
        "hour11_1":0,
        "hour11_2":1,
        "hour11_3":0,
        "hour11_4":1,
        "hour12_1":0,
        "hour12_2":1,
        "hour12_3":0,
        "hour12_4":1,
        "hour11_2":1,
        "hour11_3":0,
        "hour11_4":1,
        "hour12_1":0,
        "hour12_2":1,
        "hour12_3":0,
        "hour12_4":1,
        "hour13_1":0,
        "hour13_2":1,
        "hour13_3":0,
        "hour13_4":1,
        "hour14_1":0,
        "hour14_2":1,
        "hour14_3":0,
        "hour14_4":1,
        "hour15_1":0,
        "hour15_2":1,
        "hour15_3":0,
        "hour15_4":1,
        "hour16_1":0,
        "hour16_2":1,
        "hour16_3":0,
        "hour16_4":1,
        "hour17_1":0,
        "hour17_2":1,
        "hour17_3":0,
        "hour17_4":1,
        "hour18_1":0,
        "hour18_2":1,
        "hour18_3":0,
        "hour18_4":1,
        "hour19_1":0,
        "hour19_2":1,
        "hour19_3":0,
        "hour19_4":1,
        "hour20_1":0,
        "hour20_2":1,
        "hour20_3":0,
        "hour20_4":1,
        "hour21_1":0,
        "hour21_2":1,
        "hour21_3":0,
        "hour21_4":1,
        "hour22_1":0,
        "hour22_2":1,
        "hour22_3":0,
        "hour22_4":1,
        "hour23_1":0,
        "hour23_2":1,
        "hour23_3":0,
        "hour23_4":1
    }
}
```

The ActivationSetDay **and** ActivationSetDay3 **commands** have exactly the same structure as the **ActivationSetNextDay command** above, except for the name.

The ActivationSetDetailedDay **and** ActivationSetDetailedDay3 **commands** have exactly the same structure as the **ActivationSetDetailedNextDay** command above in name.

It is possible to transmit only the times of the plant state switchover; the missing parameters will retain the last state previously described.

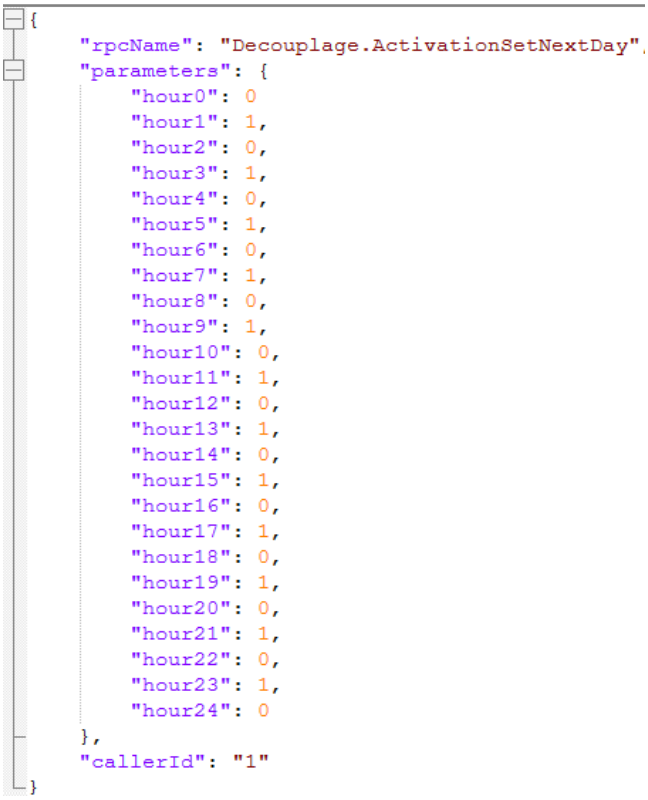
Thus, the next order will cause a decoupling of the control unit from 11:30 a.m. to 1:45 p.m. and then from 2:15 p.m. to 3 p.m. during the day following the sending of the order. The plant will be coupled the rest of the time.

```
{
    "rpcName": "Decouplage.ActivationSetDetailedNextDay",
    "parameters": {
        "hour0_1":1,
        "hour11_3":0,
        "hour13_4":1,
        "hour14_2":0,
        "hour15_1":1
    }
}
```

Similarly, the following control will decouple the control unit from 12 noon to 2 pm:

```
{
    "rpcName": "Decouplage.ActivationSetNextDay",
    "parameters": {
        "hour0": 1,
        "hour12": 0,
        "hour14": 1
    },
    "callerId": "1"
}
```

It is possible to use the "hour24" parameter which completely replaces the "hour0" parameter, if the 2 parameters are present, they must have the same value, otherwise an error message will be sent back from the order and the order will be rejected.



```
{
    "rpcName": "Decouplage.ActivationSetNextDay",
    "parameters": {
        "hour0": 0
        "hour1": 1,
        "hour2": 0,
        "hour3": 1,
        "hour4": 0,
        "hour5": 1,
        "hour6": 0,
        "hour7": 1,
        "hour8": 0,
        "hour9": 1,
        "hour10": 0,
        "hour11": 1,
        "hour12": 0,
        "hour13": 1,
        "hour14": 0,
        "hour15": 1,
        "hour16": 0,
        "hour17": 1,
        "hour18": 0,
        "hour19": 1,
        "hour20": 0,
        "hour21": 1,
        "hour22": 0,
        "hour23": 1,
        "hour24": 0
    },
    "callerId": "1"
}
```

The same applies to the parameters "hour0_1", "hour0_2", "hour0_3" and "hour0_4" with the respective parameters "hour24_1", "hour24_2", "hour24_3" and "hour24_4" respectively

By default the schedule is active from midnight the next day, but it is possible to choose the time to switch the orders from the current day to the next day using the next **SetSwitchTime** command .

```
{
    "rpcName": "Decouplage.SetSwitchTime",
    "parameters": {
        "hour": 10,
        "min": 50
    },
    "callerId": "1"
}
```

Thus, the schedule described in the **ActivationSetNextDay** command will be applied from the next passage at 10:50 am. The same day if the **ActivationSetNextDay order** is sent before 10:50 AM or the next day if the order is sent after 10:50 AM. The first command applied will be the 11 a.m. command corresponding to the "hour11" parameter and the last order applied will be the 10 a.m. command the next day corresponding to the "hour10" parameter before making a new switch at 10:50 a.m.

By default, the commands are applied every quarter of an hour (0min, 15min, 30min, 45min), but it is possible to configure a delay to allow all inverters to be in the desired state at the expected time. Thus, the use of the **SetAnticipation function** makes it possible to define the advance of execution of the order in minutes compared to the expected time. In the example below, we define an anticipation of 12min.

```
{
    "rpcName": "Decouplage.SetAnticipation",
    "parameters": 12,
    "callerId": "2"
}
```

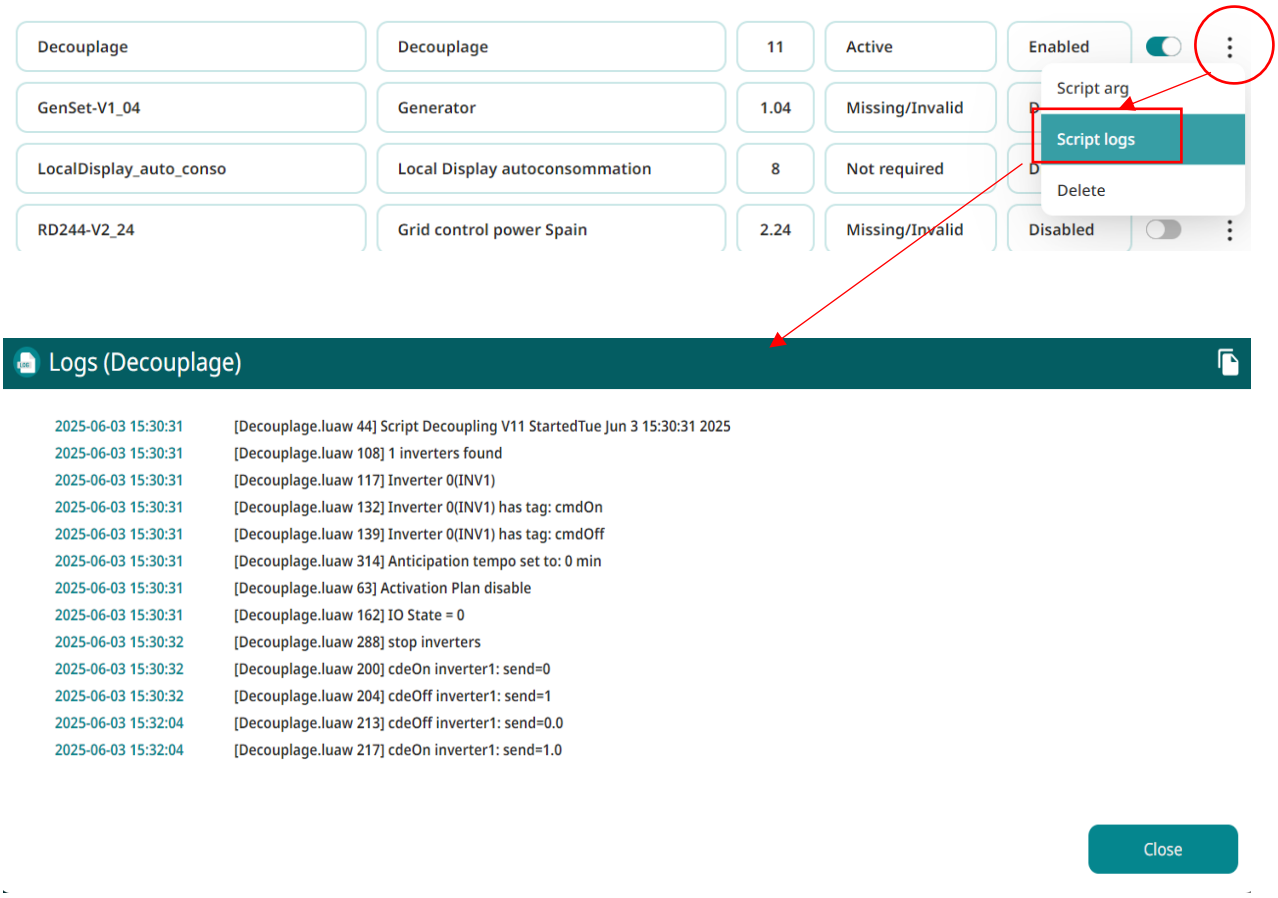
Thus, the 1 p.m. order will be executed at 12:48 p.m.

The script respects the time set on the WebdynSunPM.
It considers the selected time zone. The difference due to daylight saving time is not taken into account. In summer, it is therefore advisable to expect that the command executed will be the one defined in the previous hour's parameter. Similarly, the changeover will be one hour later than the actual time.

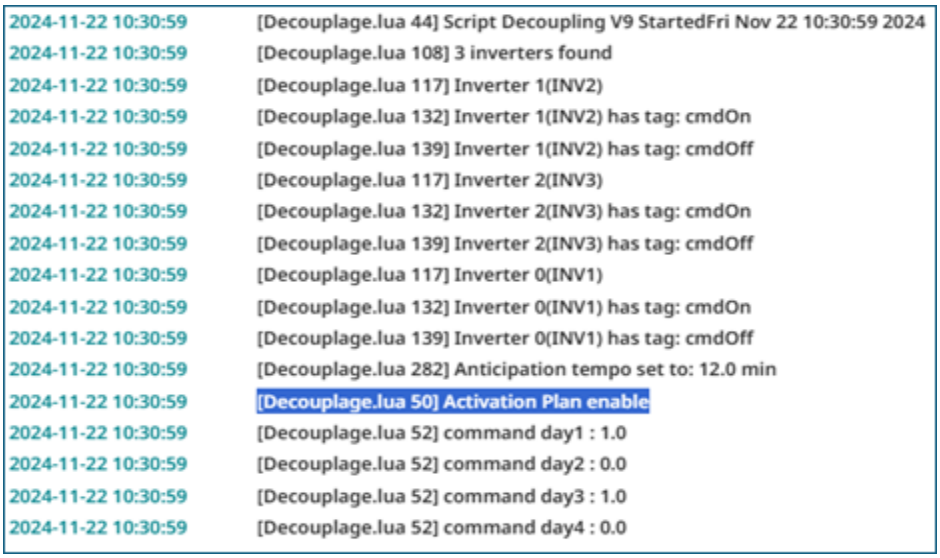
Once the functionality has been initialized, the schedules for the 3 days as well as the changeover time and the advance time of orders, are kept permanently, even in the event of an interruption of operation or restart of the product. Only deleting the script allows the feature to be stopped and schedules to be deleted. However, voluntarily stopping the script or passing commands describing a permanent coupling inhibits the scheduling system.

Log exploitation

Script logs contain no trace until a command is sent to the UPS, so that it is easy to trace the different commands received.



Starting with version 9 of the script, the activation status of the scheduling feature is displayed when the script starts.



```
2024-11-22 10:30:59 [Decouplage.lua 50] Activation Plan enable
2024-11-22 10:30:59 [Decouplage.lua 52] command day1 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day2 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day3 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day4 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day5 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day6 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day7 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day8 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day9 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day10 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day11 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day12 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day13 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day14 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day15 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day16 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day17 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day18 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day19 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day20 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day21 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day22 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day23 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 52] command day24 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day1 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day2 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day3 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day4 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day5 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day6 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day7 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day8 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day9 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day10 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day11 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day12 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day13 : 1.0
```

The changeover time and the time to anticipate orders are then displayed:

```
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day23 : 1.0
2024-11-22 10:30:59 [Decouplage.lua 55] command Next day24 : 0.0
2024-11-22 10:30:59 [Decouplage.lua 57] switch time10.0H22.0
2024-11-22 10:30:59 [Decouplage.lua 58] Anticipation12.0
```

Compatible equipment.

Not all inverters allow the control of their internal relay via a Modbus register.

With these inverters, it is advisable to use an external circuit breaker driven by the internal relay of the WebdynSunPM or to check if they have the power reduction control in order to allow the script to use the "UsePwrPct" function described in this manual.

The UPS units that we have identified as compatible with the control of their internal relay are the following:

-EFACEC
-GOODWE
-HUAWEI
-KEHUA
-SAJ
-SOFAR
-SUNGROW
-As well as all Sunspec inverters that have implemented table 123 (immediate command)

To confirm the operation with other inverters, you can contact us at support@webdyn.com or inquire with the inverter manufacturer.