

# OPMONis Manual

State of 30-AUG-2017

Version v1.4.0

<b>1</b>	<b>INSTALLATION.....</b>	<b>3</b>
<b>2</b>	<b>LICENSING.....</b>	<b>3</b>
<b>3</b>	<b>GUI.....</b>	<b>4</b>
<b>3.1</b>	<b>UPS-Monitoring.....</b>	<b>4</b>
3.1.1	Setting the threshold values .....	4
3.1.2	Adding and removing UPSs .....	4
3.1.3	Naming of UPSs for a clearer view .....	5
3.1.4	Refresh the display .....	5
<b>3.2</b>	<b>System Administration .....</b>	<b>5</b>
3.2.1	Add new System.....	5
3.2.1.1	Windows .....	6
3.2.1.2	Linux .....	6
3.2.1.3	ESXi (Host) .....	8
3.2.1.4	ESXi Client (VM).....	8
3.2.1.5	Free ESXi (Host).....	9
3.2.1.6	Free ESXi Client (VM) .....	9
3.2.2	Delete .....	9
3.2.3	Arrange Order .....	10
3.2.4	Refresh .....	10
<b>3.3</b>	<b>History Overview.....</b>	<b>10</b>
<b>3.4</b>	<b>Settings .....</b>	<b>10</b>
3.4.1	Language Settings .....	10
3.4.2	License Settings .....	10
<b>3.5</b>	<b>Information.....</b>	<b>10</b>
<b>4</b>	<b>SAVING THE SETTINGS.....</b>	<b>11</b>
<b>5</b>	<b>OPERATING MODE AUTOMATIC / MANUAL .....</b>	<b>11</b>
<b>6</b>	<b>COMMANDLINE EXECUTABLE .....</b>	<b>11</b>
<b>7</b>	<b>TECHNIQUE .....</b>	<b>13</b>
<b>7.1</b>	<b>Definitions.....</b>	<b>13</b>
7.1.1	Order, Task, Action and Protocols.....	13
7.1.2	In Detail .....	13
<b>7.2</b>	<b>The tasks in detail .....</b>	<b>13</b>
7.2.1	Task Request Status .....	13
7.2.2	Task Shutdown System.....	13
7.2.3	Task Power on .....	14
<b>7.3</b>	<b>Actions and used protocols .....</b>	<b>14</b>
7.3.1	Overview .....	14
7.3.2	The protocols in detail.....	14
7.3.2.1	Protocol "Ping" .....	14
7.3.2.2	Protocol "Wake on Lan" (WoL).....	15
7.3.2.3	Protocol "WMI" .....	15
7.3.2.4	Protocol "SSH" for Linux .....	15
7.3.2.5	Protocol "SSH" for free ESXi Host .....	15
7.3.2.6	Protocol "vShpere Web Services" .....	16
7.3.2.7	Protocol "Xen API" .....	16
<b>7.4</b>	<b>UPS Monitoring .....</b>	<b>16</b>
7.4.1	Monitoring by WMI.....	16
7.4.2	Monitoring by SNMP (v1).....	16
7.4.2.1	Polling of the UPS status .....	17
7.4.2.2	Requesting the remaining runtime and the charging status.....	17

## 1 Installation

After downloading the software at <https://opmonis.de/en/downloads> you can start the installation by running the downloaded .msi file. An installation wizard will guide you through all following steps. You'll have to accept our licensing agreement and choose where you want OPMONis to be installed. Clicking "Continue", OPMONis will start installing the GUI and the windows service which provides the needed functions for OPMONis.

## 2 Licensing

OPMONis provides different versions for purchase:

Version	Features
Trial	30 day full access
Free ESXi	Up to 10 systems under free VMware vSphere Hypervisor, serial/USB connected UPSs only
Small Business	Up to 10 systems, UPS connected by USB or serial cable
Standard	Unlimited Systems, UPS connected by USB or serial cable UPS connected by Network

While installing, OPMONis will automatically generate a trial license valid for 30 days. After purchasing a license you will along with the invoice receive a preliminary license which will extend the trial up to another 30 days. After receiving the money we will send the final license. You will find how to activate the licenses in Chapter 3.4.2 "License Settings".

## 3 GUI

On the left side you'll find five menu items. These provide the basic settings. In the following chapter these items will be explained first abstract and further in detail.

### 3.1 UPS-Monitoring

The UPS-Monitoring is taking care of all UPS that are allocated in your system. Depending on your license you can use any UPS that is recognized by Windows and which status can be queried by "WMI" or - additional in a higher license - all UPSs connected by network (Or UPS that provide their data via SNMPv1). Like this it is possible to use the much cheaper UPSs that are lacking a network interface card, too.

OPMONis is monitoring two different values of the UPSs. First is the remaining runtime (in minutes) and second is the battery capacity (in percent). Both can be chosen via a checkbox simultaneous or separately.

The main advantage of monitoring the remaining time is a better measurability and thus a better adjustability. Capacity is only a relative value which can change over time e.g. when the battery is worn out.

#### 3.1.1 Setting the threshold values

You can monitor the percentage of the battery capacity or the remaining battery runtime. Both values can be monitored simultaneous, too. Just select the checkboxes convenient for your system. You can write the threshold value directly into the input field or adjust it using the small arrow keys. The shutdown of your systems will be triggered as soon as one of the chosen threshold values is reached.

Using only the remaining battery runtime is sufficient for most systems and is recommended. Here you should choose a value that represents the real duration of the time needed to shut down the whole system with an additional extra time (e.g. two minutes).

TIPP: You can find the ideal value by making a test shutdown and measuring the time needed until the shutdown is finished. Use this value with some extra time as threshold value.

Monitoring the percentage of the battery capacity can be useful if the manufacturer's recommendation is not to discharge deeply for better durability of the battery. Most manufacturers recommend not to discharge their UPS below 30%. This could be an eligible threshold value.

#### 3.1.2 Adding and removing UPSs

The button "plus" allows you to add your UPS to the monitoring by OPMONis. You can choose between a cable connected UPS (serial or USB) and a network UPS (SNMP). You'll find further information in the chapter "Technique". You will be asked which one of these two possibilities you'll have to choose.

Cable connected devices will be identified automatically after clicking on the small arrow button next to the "Device" field and be displayed in a list from which you can choose. This list always contains all available devices. Already monitored devices will be shown in a lighter grey and can't be selected any more. Not yet monitored devices can be added one by one.

Network UPSs (or SNMP using devices) have to be added manually. You will need either the devices DNS-name or its IP-address. All other settings are preset to the

actual standards (defined in the MIB “RFC 1628”). It is possible to adjust these settings manually if your UPS is not supporting the standard. (Further information in the chapter “Technique”).

Generally it’s possible to check if you settings work with the “Test” button. This allows you to see in which format the needed data for OPMONis is delivered by the UPS.

The button “recycle bin” will remove unneeded UPSs from the monitored list.

**Important:** If you use multiple UPSs, the first time a threshold is reached by an UPS will trigger the shutdown, even if there is enough capacity left in another UPS.

### 3.1.3 Naming of UPSs for a clearer view

Each UPSs can be given a name for a better overview. You can set the name in the input field “Display Name”. OPMONis will generally select and insert the official name reported by Windows. The chosen name has no further effect and only will be used and saved within OPMONis.

### 3.1.4 Refresh the display

The button “refresh” will rescan the charge levels of the monitored UPSs. An actual level will be loaded starting the GUI and as well automatically on a regular base.

## 3.2 System Administration

In the system administration you can add all virtual and physical systems of your network, which you want to be controlled by OPMONis. At the moment OPMONis can handle Windows, Linux and ESXi (Hosts and virtual machines).

The added systems will be shut down in the same order as they are displayed. The standard behavior of OPMONis is to monitor the status of the systems and to trigger the next systems shutdown not before the actual system has finished their action (shutdown). Only after that the next shutdown is triggered. Selecting the checkbox “Await Execution” will allow you to change this behavior.

If it is not possible for any reason to shut down a system (e.g. an execution error or just the system not reacting to the shutdown command), automatically the next system will be triggered.

Following chapter will explain every system type with their specifics. You can find technical details in the last chapter “Technique”.

### 3.2.1 Add new System

The button “plus” will allow you to add your systems. After clicking this button a new window will open asking to choose between:

- Windows
- Linux
- ESXi host
- ESXi Client (VM)
- free ESXi
- free ESXi host (VM)
- XenServer Host
- XenServer Client (VM)
- Hyper-V Host
- Hyper-V Client (VM)

If you are using an OPMONis license for free ESXi this is limited to the free ESXi versions. Also you will be asked to give a name to the system. The device is now added, but still needs some more information to allow OPMONis the communication needed.

### 3.2.1.1 Windows

#### Title

The title is the systems logic name and will only be used within OPMONis. The title is arbitrary.

#### Host Name (DNS) / IP

The DNS name or the IP address of the system. OPMONis will use this value to communicate by network with the system.

#### MAC Address

The MAC address can be set automatically by clicking on the search button in the MAC address line. This is only possible if the system is located in the same subnet as OPMONis, else it has to be inserted manually. OPMONis needs this value to trigger the boot process via “Wake on Lan”.

#### Login

The login name will be used to login by network and to trigger the action “shutdown”. For this action the user of whose login data is used for OPMONis has to have set the needed authorizations. **IMPORTANT:** If you use Active Directory and the remote system has UAC enabled (User Account Control), you have to adjust the authorizations for DCOM. If you don’t know how to do this you’ll find a step by step explanation on our forum: <http://forum.opmonis.de/thread/preset-windows-systems/>

#### Password

The password is needed to log into the system by network. It will be encrypted by OPMONis (details about the encrypting process you can find in the chapter “Technique”)

#### Domain

If the system is member of a Windows domain and you want to use a login from this domain, you have to add the domain name here. If you are using a local user on the system, this field has to be left empty.

#### Await Execution

If this checkbox is selected, OPMONis will monitor the system via Ping until the active action is finished. Only then OPMONis will continue with the next action. If this option is not selected, OPMONis will continue with the next device immediately. You can find details about this in the chapter “Technique”

### 3.2.1.2 Linux

#### Title

The title is the logic name of the system and will only be used within OPMONis. The title is arbitrary.

#### Host Name (DNS) / IP

The DNS name or the IP address of the system. OPMONis will use this value to communicate by network with the system.

### MAC Address

The MAC address can be set automatically by clicking on the search button in the MAC address line. This is only possible if the system is located in the same subnet as OPMONis, else it has to be inserted manually. OPMONis needs this value to trigger the boot process via “Wake on Lan”.

### SSH Port

The TCP port used to establish the SSH connection to the specified host. The default port number is 22.

### Login

The login name will be used to login by network and to trigger the action “shutdown”. For this action the user whose login data is used for OPMONis has to have set the needed authorizations.

### Password

The password needed to log into the system by network. It will be encrypted by OPMONis (details about the encrypting process you can find in the chapter “Technique”).

### SSH Private Key

If your setup requires a SSH login using public/private key pair, you must copy the content of the private key file into this text box. The private key must be in DER encoding (first line must look like: -----BEGIN RSA PRIVATE KEY-----).

The private key will be encrypted by OPMONis (details about the encrypting process you can find in the chapter “Technique”).

### SSH Private Key Passphrase

The passphrase by which the SSH private key is protected. If the private key is not protected by a passphrase, this field has to be empty.

The passphrase will be encrypted by OPMONis (details about the encrypting process you can find in the chapter “Technique”).

### SSH-Command

OPMONis is using an ssh-command to trigger the shutdown. This command is predefined in this field and should work for almost every Linux system. Still it can be edited if needed. You´ll find a detailed explanation about this in the chapter “Technique”

**IMPORTANT:** Basically it is always possible to change the type of a system at any time. But the correct ssh-command will only be filled in this field if the initial system type was Linux.

The SSH command will be encrypted by OPMONis (details about the encrypting process you can find in the chapter “Technique”).

### Await Execution

If this checkbox is selected, OPMONis will monitor the system via Ping until the active action is finished. Only then OPMONis will continue with the next action. If this option is not selected, OPMONis will trigger directly the next action (e.g. shutdown). You can find details about this in the chapter “Technique”

### 3.2.1.3 ESXi (Host)

#### Title

The title is the logic name of the system and will only be used within OPMONis. The title is arbitrary.

#### Host Name (DNS) / IP

The DNS name or the IP address of the system. OPMONis will use this value to communicate by network with the system.

#### ESXi Host Login

The login name will be used to login to the ESXi host via network. For this to be possible the chosen user has to have set the required authorizations (admin rights)

#### ESXi Host Password

The password is required to log in to the system. The password will be encrypted by OPMONis (You'll find technical details about this in the chapter "Technique")

#### Await Execution

If this checkbox is selected, OPMONis will monitor the system via Ping until the active action is finished. Only then OPMONis will continue with the next action. If this option is not selected, OPMONis will trigger directly the next action (e.g. shutdown). You can find details about this in the chapter "Technique"

### 3.2.1.4 ESXi Client (VM)

#### Title

The title is the logic name of the system and will only be used within OPMONis. The title is arbitrary.

#### Host Name (DNS) / IP

The DNS name or the IP address of the system. OPMONis will use this value to communicate by network with the system.

#### ESXi Host Login

The login name will be used to login to the ESXi host via network. For this to be possible the chosen user has to have set the required authorizations (admin rights)

#### ESXi Host Password

The password is required to log in to the system. The password will be encrypted by OPMONis (You'll find technical details about this in the chapter "Technique")

#### Name of Virtual Machine

The name of the virtual machine controlled by OPMONis. You can look up this name by using the vSphere Client.

#### Await Execution

If this checkbox is selected, OPMONis will monitor the system via Ping until the active action is finished. Only then OPMONis will continue with the next action. If this option is not selected, OPMONis will trigger directly the next action (e.g. shutdown). You can find details about this in the chapter "Technique"

### 3.2.1.5 Free ESXi (Host)

#### Title

The title is the logic name of the system and will only be used within OPMONis. The title is arbitrary.

#### Host Name (DNS) / IP

The DNS name or the IP address of the system. OPMONis will use this value to communicate by network with the system.

#### ESXi Host Login

The login name will be used to login to the ESXi host via network. For this to be possible the chosen user has to have set the required authorizations (admin rights)

#### ESXi Host Password

The password is required to log in to the system. The password will be encrypted by OPMONis (You'll find technical details about this in the chapter "Technique")

#### Await Execution

If this checkbox is selected, OPMONis will monitor the system via Ping until the active action is finished. Only then OPMONis will continue with the next action. If this option is not selected, OPMONis will trigger directly the next action (e.g. shutdown). You can find details about this in the chapter "Technique"

### 3.2.1.6 Free ESXi Client (VM)

#### Title

The title is the logic name of the system and will only be used within OPMONis. The title is arbitrary.

#### Host Name (DNS) / IP

The DNS name or the IP address of the system. OPMONis will use this value to communicate by network with the system.

#### ESXi Host Login

The login name will be used to login to the ESXi host via network. For this to be possible the chosen user has to have set the required authorizations (admin rights)

#### ESXi Host Password

The password is required to log in to the system. The password will be encrypted by OPMONis (You'll find technical details about this in the chapter "Technique")

#### Name of Virtual Machine

The name of the virtual machine controlled by OPMONis. You can look up this name by using the vSphere Client.

#### Await Execution

If this checkbox is selected, OPMONis will monitor the system via Ping until the active action is finished. Only then OPMONis will continue with the next action. If this option is not selected, OPMONis will trigger directly the next action (e.g. shutdown). You can find details about this in the chapter "Technique"

### 3.2.2 Delete

With the button "recycle bin" you can remove a system from the monitoring

### 3.2.3 Arrange Order

If you are controlling more than one system with OPMONis you can arrange the order your systems should be handled with the arrow buttons.

### 3.2.4 Refresh

The button “refresh” is reloading the actual state (online, offline, request) of the controlled systems. By starting the GUI an actual state is loaded. After the loading at start a new status will only loaded by clicking this button.

## 3.3 History Overview

Using the third menu item “History Overview” will allow you to get an overview of all events and incidents that happened concerning your UPSs: What happened, which systems or which UPSs where involved and much more. You can choose start and end date by directly entering the relevant dates or you can use the little arrow buttons. Also it is possible to use a full text search over the parameters to limit the amount of displayed events.

IMPORTANT: Only parameters are effected by the full text search, not the events.

## 3.4 Settings

### 3.4.1 Language Settings

In the language settings you can choose your preferred language. At the moment only English and German are available. It is not needed to save the change, it will apply immediately.

### 3.4.2 License Settings

In the licensing section you can either see the information about your license and also you can enter your purchased license key. If you want to activate your license, just click on “new license”. You will have to select your license file which was provided by independis.

If you want to buy a new license or if you want to upgrade an existing license, we have to invite you to our homepage: [www.opmonis.com](http://www.opmonis.com)

## 3.5 Information

Checking the Information you will find the version number of the OPMONis version you actually are using. Also you see a time stamp when the application was started. As well you can see the paths where on your system is saving the needed data. At last you see the address of the interface which is in use by OPMONis for communication between GUI and the Windows service.

## 4 Saving the Settings

Every change that is made in OPMONis has to be saved before it takes effect. If you have made any changes OPMONis will remind you to save if you want to exit without saving. You find the button for saving on the righter bottom. This button will appear in red for saving. Next to it you find another button in Orange for revoking the changes if needed.

## 5 Operating Mode Automatic / Manual

By default the configured mode on OPMONis is “Automatic”. In this mode OPMONis is monitoring the state of the registered UPSs and will trigger a shutdown of all systems as soon as the configured threshold is reached (e. g. as result of a blackout).

On the upper left, right next to the name of the actual operating mode you´ll find the button “change mode”. After clicking you will be asked if you are really sure to switch to manual mode. In this mode the automatic shutdown triggering is deactivated. While you are in manual mode you can manually trigger shutdown or startup of your systems. This can be useful for testing the configurations or for planned shutdowns for maintenance or hardware changings.

You should also set the manual operating mode if you want to calibrate the runtime of your UPSs, which mostly is done by a forced discharge. Else the calibration will result in a shutdown because most (cheaper) UPSs don´t deliver different data for calibration and blackout, so by reaching the set thresholds OPMONis will shut your systems down.

**IMPORTANT:** While in manual mode, no automatic shutdown will happen when thresholds are reached. It is required to set OPMONis back to “automatic mode” as soon as you have finished testing or maintenance.

## 6 Commandline Executable

OPMONis is providing a commandline executable, which allows easy access and monitoring:

- Triggering a shutdown or power on of all systems
- Check the actual state
- Switch between the modes automatic and manual
- Check the actual mode

You can find the commandline executable `independis.OPMONis.Command.exe` in the installation folder of the client components. The standard path for this is `%ProgramFiles(x86)%\independis\OPMONis\Client`

If you start the commandline tool without parameters, it will show you the following screen:

```
C:\Program Files (x86)\independis\OPMONis\Client>independis.OPMONis.Command.exe
OPMONis Commandline Tool 1.2.2.0
Commandline tool for controlling the OPMONis service.

Usage:
  independis.OPMONis.Command [command]

The following commands are supported:
  TriggerShutdown  Initiates a shutdown job to be executed by the OPMONis service.
  TriggerStartup   Initiates a startup job to be executed by the OPMONis service.
  ManualMode       Switch the OPMONis service in manual operation mode.
  AutomaticMode    Switch the OPMONis service in automatic operation mode.
  QueryMode        Queries the current OPMONis service operation mode (Manual oder Automatic).
  QueryJob         Queries the job currently executed by the OPMONis service (Idle, QueryState,
  PowerOn, PowerOff).
  Pause           Wait until the user presses the enter key.

Exit Codes:
  If the command was executed successfully, the exit code is 0. If any error occurred, the exit
  code is -1.

Examples:

  Initiates a PowerOff job to be executed by the OPMONis service:
  independis.OPMONis.Command TriggerShutdown

  Initiates a PowerOn job to be executed by the OPMONis service:
  independis.OPMONis.Command TriggerStartup

  Switches the OPMONis service in manual operation mode:
  independis.OPMONis.Command ManualMode
```

## 7 Technique

In this chapter you will find the techniques and structures behind OPMONis. We define our used vocabulary and also show in detail our used command sets. This way you know what happens in your systems while OPMONis is working for you.

### 7.1 Definitions

#### 7.1.1 Order, Task, Action and Protocols

OPMONis is working with orders (e.g. shutdown the systems). An order is containing different tasks (each system represents a task). Each task is grouped into different actions (e.g. shutdown command, query state). For all this OPMONis is using different protocols to be able to perform the required actions on your systems.

#### 7.1.2 In Detail

OPMONis is using different orders depending on the situation, triggered automatically or manually by the user.

Situation	Order
USV discharging and threshold reached	Shutdown
USV charging and threshold exceeded	Power On
User triggering shutdown (manual mode)	Shutdown
User triggering Power On (manual mode)	Power On
User triggered refresh	Request Status

An order is proceeded by creating the different tasks on every configured system. The systems will be processed in the order displayed in the configuration.

A cancellation of an order is possible after completion of a task. As example an order can be cancelled, when the power supply has been reestablished and OPMONis is starting a new task and the charge level of the UPS is exceeds the set thresholds.

OPMONis distinguishes three different tasks:

1. Status Request
2. System Shutdown
3. System Power On

The different tasks use several different actions to reach the desired goal of a task.

### 7.2 The tasks in detail

#### 7.2.1 Task Request Status

The task "request status" is using the action "request status" to get the information about the online status of a system. The result is either "online" or "offline"

#### 7.2.2 Task Shutdown System

The task "shutdown system" is using three different actions on the configured systems:

1. Action: "request status"

2. Action: "shutdown system"
3. Action: "check status"

If the action "request status" shows that the system is already "offline", further actions won't be executed.

If the option "await execution" is deactivated, OPMONis will skip the action "check status"

### 7.2.3 Task Power on

The task "power on" is using three different actions on the configured systems:

1. Action: "request status"
2. Action: "system power on"
3. Action: "check status"

If the action "request status" shows that the system is already "online", further actions won't be executed.

If the option "await execution" is deactivated, OPMONis will skip the action "check status"

## 7.3 Actions and used protocols

### 7.3.1 Overview

OPMONis is communicating over network with the systems to be controlled. For the different actions following protocols are used:

System Type	Aktion Request Status	Action Shutdown	Action Power On	Action Check Status
Windows	Ping	WMI	WOL	Ping
Linux	Ping	SSH	WOL	Ping
ESXi Host	Ping	vSphere Web Services	WOL	Ping
ESXi Client (VM)	vSphere Web Services	vSphere Web Services	vSphere Web Services	vSpere Web Services
Free ESXi Host	Ping	SSH	WOL	Ping
Free ESXi Client (VM)	SSH	SSH	SSH	SSH
XenServer Host	Ping	Xen API	WOL	Ping
XenServer Client (VM)	Xen API	Xen API	Xen API	Xen API
Hyper-V Host	Ping	WMI	WOL	Ping
Hyper-V Client (VM)	WMI	WMI	WMI	WMI

### 7.3.2 The protocols in detail

#### 7.3.2.1 Protocol "Ping"

The "Ping" protocol is used by OPMONis to request the status of a system. For this the Ping protocol is sending an "Echo-Request" package to the network address of the system and awaits the receiving of the "Echo-Reply". A missing "Echo-Reply" or a dedicated error message in an "Echo-Reply" will be interpreted as if the system is offline.

In case of a missing “Echo-Reply” the request will be repeated up to three times, with a delay of a maximum of 5 seconds in order to avoid short-dated errors in the network.

### 7.3.2.2 Protocol “Wake on Lan” (WoL)

The “Wake on Lan” protocol is used by OPMONis to power on an offline system through network. For this a “magic packet” is sent directly to the network address of the target system. The network card of the target system is defined by the MAC address.

Unfortunately the target system will not send a confirmation of receiving or a successful processing of the “WoL” order. That is why OPMONis will send several “magic packets” during the action “Request Status”.

### 7.3.2.3 Protocol “WMI”

The “WMI” protocol (Windows Management Instrumentation) is used by OPMONis to shut down Windows systems via network. For this OPMONis will execute a shutdown command on the System (Win32\_OperationSystem) via WMI. Windows is requested to shut down the system even if an application is not reacting to the shutdown trigger by setting the flag “ForcedShutdown”

### 7.3.2.4 Protocol “SSH” for Linux

The “SSH” (Secure Shell) protocol is used by OPMONis to shut down a Linux system via network. For this a shell command is executed on the Linux system. As a standard following command line is executed:

```
sudo --reset-timestamp --shell --stdin --background sh -c \"sleep 1 && shutdown -P +0\" <<< {Password}
```

This shell command in detail:

- **sudo**  
starting a command as „root“ user with the password of the actual user
- Parameter **reset-timestamp**  
deleting a possibly cached password from an earlier „sudo“ command, forcing a password request
- Parameter **shell**  
starting a shell conform to the user configuration on the Linux system.
- Parameter **stdin**  
reading the password from standard input (making input redirect possible)
- Parameter **background**  
starting for the following command in a new process and allowing the execution in the background.
- Befehl **sh -c "sleep 1 && shutdown -P +0"**  
starting a new shell, waiting for one second and triggering immediate shutdown
- **<<< {Password}**  
entering the password by input redirect to the sudo command

### 7.3.2.5 Protocol “SSH” for free ESXi Host

The “SSH” (Secure Shell) protocol is used by OPMONis to manage virtual machines running on a free ESXi host via network.

OPMONis uses the command “`vim-cmd vmsvc/getallvms`” to query registered virtual machines. By using the “`vim-cmd vmsvc/power`” commands OPMONis is triggering the shutdown and startup of virtual machines and queries the current power state.

#### 7.3.2.6 Protocol “vSphere Web Services”

The “vSphere Web Services” protocol is used by OPMONis to control an ESXi Host and the virtual machines that are running there. The “vSphere Web Services” is allowing access to the “vSphere Management APIs” via network.

With this management API it is possible to control the virtual machines as well as the ESXi host systems (Power On, Shutdown, Shutdown guest systems).

#### 7.3.2.7 Protocol “Xen API”

The Citrix XenServer Management API is provided by citrix and used a network connection to control a Xen Host and virtual machines running on a Xen Host.

## 7.4 UPS Monitoring

OPMONis is monitoring the UPS by polling the status of the UPS on a regular base. The status of each added UPS is polled every 20 seconds and compared to the last polled status. If there is a change, then this is saved as an event. (E.g. a change from charging to discharging). If this event is recognized as the status “discharging” and the set threshold values are met, then OPMONis will trigger the Shutdown of the systems (if OPMONis is in automatic mode)

### 7.4.1 Monitoring by WMI

Most UPSs that are connected by cable (USB or serial) will be recognized automatically as a battery by Windows (sometimes the drivers have to be added or removed, especially using APC). Here, the actual state of the UPS can be polled via “WMI” (Windows Management Instrumentation). With this information OPMONis is gathering the information if the state of the UPS is charging or discharging, as well as the remaining time in minutes and the charging level as a percentage.

### 7.4.2 Monitoring by SNMP (v1)

Network UPSs are monitored by OPMONis via SNMPv1 (Simple Network Management Protocol). With this protocol it is possible to poll the actual state of the UPS by network (GET data package).

To be able to poll a data value from an UPS, OPMONis needs to know two information:

1. Community name:  
This is used to display access permissions. Usually, for read-only access the used community name is “public”.
2. Object Identifier (OID)  
The OID is an explicit identifier. The OID is defined in dot notation (e.g. 1.2.6.1.2.33.1.2.3.0 for the remaining time in minutes)

The general difficulty about SNMP is to find the matching OID value. For a representation that also can be read by humans usually a notation is used that is

called “Management Information Base” (MIB) is used. For UPSs there exists a standard MIB which was defined in the “RFC 1628”. Each company can choose to use an own definition. These companies usually make their MIBs available in the internet.

OPMONis will poll three data values by SNMPv1:

1. UPS status (charging or discharging)
2. Remaining runtime in minutes
3. Charging level in percent

#### 7.4.2.1 *Polling of the UPS status*

Most UPS provide in case of a power failure the “runtime on battery”. This value is 0, while the UPS is running supported by a power supply. Beginning with the power outage this value increases and mostly is displayed as runtime on battery in milliseconds. But each company can choose their own way and e.g. provide the value as runtime on battery in seconds.

This is why we choose an approach that makes this unimportant for OPMONis. For OPMONis it’s only important if the UPS is on status “charging” or on status “discharging”. This is why the polled SNMP value is rated by a regular expression (Regex). In case the Regex is rated as a match, OPMONis will interpret this as “charging”, in any other case as “discharging”.

For OPMONis this means: In the already mentioned standard MIB “RFC 1628” it is defined, that by OID „1.3.6.1.2.1.33.1.2.2.0” it is possible to poll the UPS runtime on battery in seconds. By the regular expression „^[1-9][0-9]\*\$” it will be rated if the runtime is bigger than 0, which would mean that there is in fact a power outage occurring.

#### 7.4.2.2 *Requesting the remaining runtime and the charging status*

To be able to recognize the reach of the thresholds, determined in the UPS settings, OPMONis needs two more values: The remaining runtime and the remaining capacity in percent.

Again we have to take a look in the standard MIB “RFC 1628” where the needed OIDs are defined:

- 1.3.6.1.2.1.33.1.2.3.0 for remaining runtime in minutes
- 1.3.6.1.2.1.33.1.2.4.0 for capacity of the battery in percent

It seems that there are differences for the different companies in which format this data is provided (e.g. minutes, seconds and even milliseconds). Therefore we added a multiplier that allows to adjust the value to the needed format.