

Qeedji

User manual

SBL10e m365_room

1.12.10 002A



Legal notice

SBL10e m365_room 1.12.10 (002A_en)

© 2024 Qeedji

Rights and Responsibilities

All rights reserved. No part of this manual may be reproduced in any form or by any means whatsoever, or by any means whatsoever without the written permission of the publisher. The products and services mentioned herein may be trademarks and/or service marks of the publisher, or trademarks of their respective owners. The publisher and the author do not claim any rights to these Marks.

Although every precaution has been taken in the preparation of this document, the publisher and the author assume no liability for errors or omissions, or for damages resulting from the use of the information contained in this document or the use of programs and source code that can go with it. Under no circumstances can the publisher and the author be held responsible for any loss of profits or any other commercial prejudice caused or alleged to have been caused directly or indirectly by this document.

Product information

Product design and specifications are subject to change at any time and 'Qeedji' reserves the right to modify them without notice. This includes the hardware, the embedded software and this manual, which should be considered as a general guide to the product. The accessories supplied with the product may differ slightly from those described in this manual, depending on the developments of the various suppliers.

Precautions for use

Please read and heed the following warnings before turning on the power: - installation and maintenance must be carried out by professionals. - do not use the device near water. - do not place anything on top of the device, including liquids (beverages) or flammable materials (fabrics, paper). - do not expose the device to direct sunlight, near a heat source, or in a place susceptible to dust, vibration or shock.

Warranty clauses

The 'Qeedji' device is guaranteed against material and manufacturing defects for a certain duration. Check the device warranty duration value at the end of the document. These warranty conditions do not apply if the failure is the result of improper use of the device, inappropriate maintenance, unauthorized modification, operation in an unspecified environment (see operating precautions at the beginning of the manual) or if the device has been damaged by shock or fall, incorrect operation, improper connection, lightning, insufficient protection against heat, humidity or frost.

WEEE Directive



This symbol means that your appliance at the end of its service life must not be disposed of with household waste, but must be taken to a collection point for waste electrical and electronic equipment or returned to your dealer. Your action will protect the environment. In this context, a collection and recycling system has been set up by the European Union.

Table of contents

Part I : Description and installation

Introduction	1.1
Device dimensions	1.1.1
Labelling	1.1.2
Installation	1.1.3
Uninstallation	1.1.4
Smart Busy Light applications	1.2

Part II : Applicative console Web user interface

Applicative console web user interface	2.1
--	-----

Part III : Administration console Web user interface

Device configuration console web user interface	3.1
Configuration > Administrator	3.1.1
Configuration > LAN	3.1.2
Configuration > Servers	3.1.3
Configuration > Date and time	3.1.4
Configuration > Tasks	3.1.5
Maintenance > Firmware	3.1.6
Maintenance > Preferences	3.1.7
Maintenance > Logs	3.1.8
Maintenance > Tools	3.1.9
Maintenance > Files	3.1.10
Information > Device	3.1.11
Information > Network	3.1.12

Part IV : Technical information

Technical specifications	4.1
Conformities	4.2

Part V : Contacts

Contacts	5.1
----------	-----

Part VI : Appendix

Appendix: web services	6.1
Appendix: Qether	6.2
Appendix: Device configuration with TFTP server (+ DHCP server code 66)	6.3
Appendix: Check Azure AD User Principal Name	6.4
Appendix: Create M365 application with Powershell	6.5
Appendix: Create M365 application with Azure AD portal	6.6
Appendix: Configuration using PowerShell for Microsoft 365 (M365)	6.7
Appendix: Devices configuration using Powershell	6.8

Part I

Description and installation

1.1 Introduction

This manual explains how to install and configure your device SBL10e device.

Recommendations and warnings

This device is designed for indoor use only.

To ensure better rendering of the SBL10e, the device should not be installed under direct sunlight.

The SBL10e device is designed to be illuminated 12 hours a day, 7 days a week.

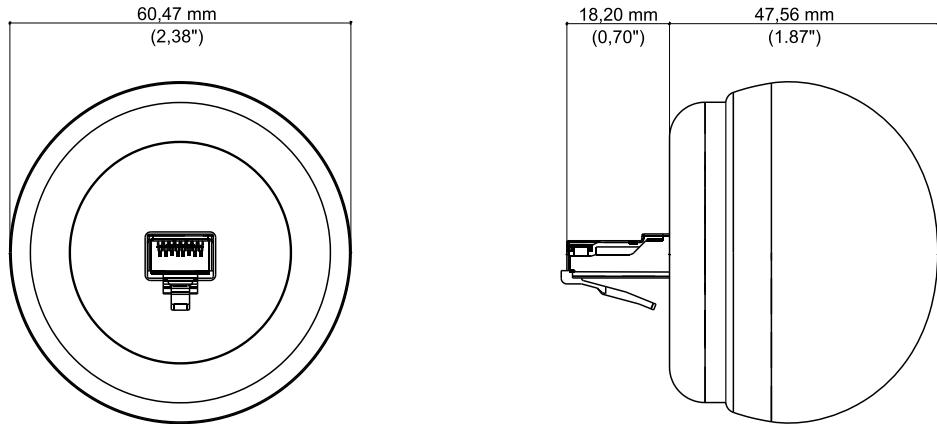
Package Contents

Articles	Description
Device	SBL10e device with the default <code>regular</code> ¹ application embedded.

¹ It is possible to easily update the device with the `m365_room` application afterwards.

 In this documentation, the unit of measurement for dimensions is done in millimeters followed by its equivalent value in inches.

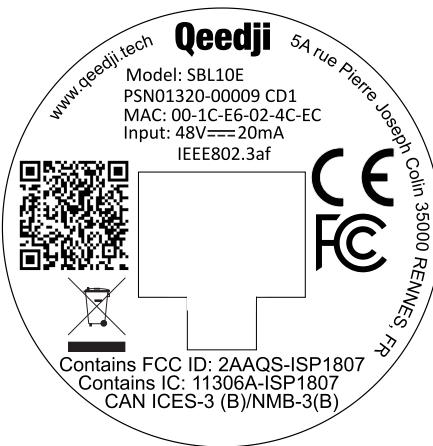
1.1.1 Device dimensions



1.1.2 Labelling

Product label

The model of the device, the power supply characteristics, the serial number (PSN) and the MAC address are written on a label stuck on the case.



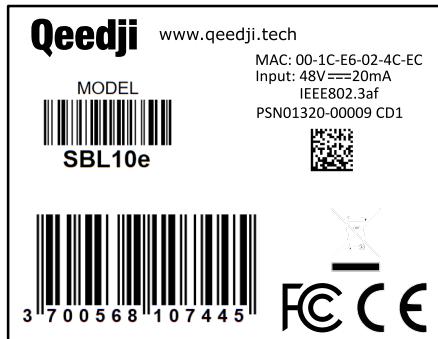
☞ This is an example of registration QR code URL:

i.qeedji.tech?model=SBL10e&sn=01320-00009&mac.Lan1=00-1C-E6-02-4C-EC&mac.wpan1=DF-27-83-3C-8A-90.

Packingbox label

This is the label stuck also on the packingbox. It is showing:

- the device model,
- the product serial number (PSN) (embedded also in the QR code),
- the manufacturer Website.



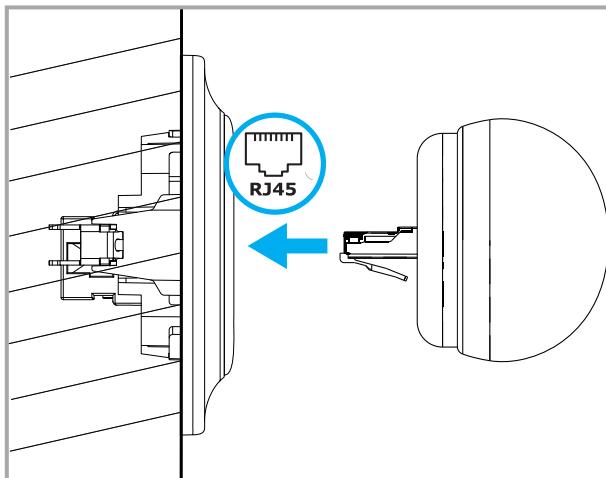
☞ The QR code on the packingbox label is corresponding to the product PSN, for example:
PSN01320-00009 CD1.

☞ The serial number of the device may be requested in case of technical support.

1.1.3 Installation

☞ Install the SBL10e device on the Ethernet wall plugs of the buildings following the installation map given by your IT department.

The SBL10e device has to be plugged to an Ethernet wall plug supporting PoE IEEE802.3af.



Given the device footprint, it is preconised to use Ethernet wall plug plastron with a right insertion.



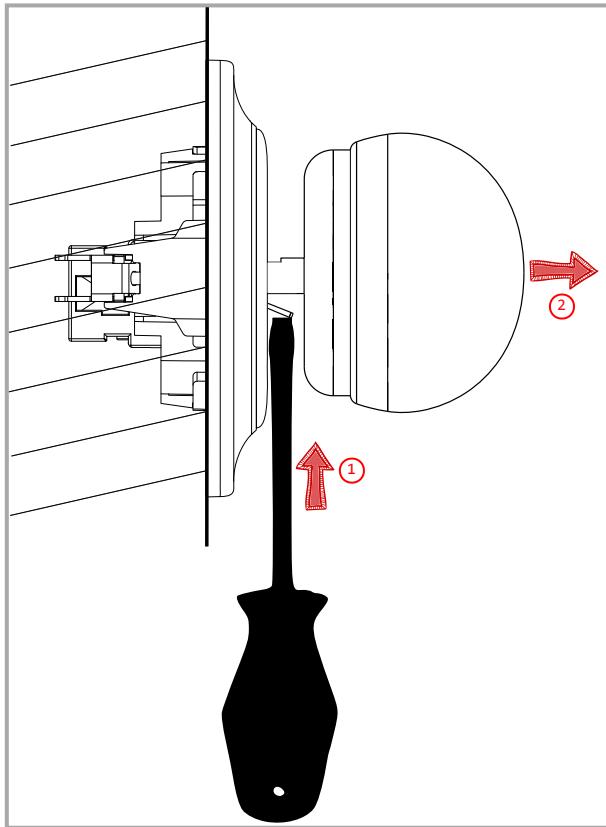
Consequently, the Ethernet wall plug whose plastron is angled is not supported.



☞ Thanks to the lock pin of its Ethernet connector, the SBL10e device can be installed on vertical surfaces, like walls as well as horizontal surfaces, like ceilings.

1.1.4 Uninstallation

With a screw driver, hold down the pin of the Ethernet connector ① of the SBL10e device at the same time you are releasing with the hand ② the SBL10e device from the Ethernet wall plug.



1.2 Smart Busy Light applications

The `m365_room` application periodically connects to your `M365` (Microsoft 365) solution and to get information about the Office calendar of a specific `resource id`. If an event has started or happening in the day, the event's title, the event's start time and the event's end time can be watched in the console web user interface of the device. Depending on the event has started, happens soon or will happen later in the day, the appropriate light state/color is displayed.

Light states and colors

The device can support the states and colors values showed below.

Color	State
	OFF
Red	ON steady OR ON flashing
Green	ON steady OR ON flashing
Blue	ON steady OR ON flashing
Orange	ON steady OR ON flashing
Yellow	ON steady OR ON flashing

☞ The `ON flashing` state is flashing with this sequence: `on` for 0,5 seconds then `off` for 0,5 seconds every one second.

☞ Depending on the application running on it, some color/state may be never used.

☞ The light color and state values are stored in the volatile memory (RAM). That means that in case the SBL10e device is unplugged from the Ethernet wall plug then plugged back again, the light comes back to its default state: `OFF` until its state is then modified by the App or by the user.

Configuration

The Smart Busy Light application supports the configuration update:

- by connecting to the device configuration console web user interface `http://<device-ip-addr>/` and changing parameters,
- by pushing, from a WebDAV client or with the device console web user interface, a `prefs.json` configuration file on the device WebDAV directory `http://<device-ip-addr>/.conf/`,
- by pushing, from a WebDAV client or with the device console web user interface, a `.js` configuration script on the device WebDAV directory `http://<device-ip-addr>/.conf/`,
- by receiving a `configure` command with an appropriate `.js` configuration script from the `Qether` tool (Qether V1.12.10 or above).

Firmware upgrade

The Smart Busy Light application supports the firmware upgrade:

- by connecting to the device configuration console web user interface `http://<device-ip-addr>/` and loading an appropriate `bm0032_m365_room-sbl10e-xx.yy.zz.bin`¹ firmware file,
- by pushing a new `bm0032_m365_room-sbl10e-xx.yy.zz.bin`¹ firmware file at the root of the device WebDAV directory `http://<device-ip-addr>/`, pushed with a WebDAV client,
- by receiving an `install` command with an appropriate `bm0032_m365_room-sbl10e-xx.yy.zz.bin`¹ firmware file from the `Qether` tool.

¹ Can work also with any other `bm0032_<custom>-sbl10e-xx.yy.zz` compatible firmware.

☞ After a firmware upgrade, the device is rebooting once.

☞ When the `configuration` command or the `install` command has been processed, the last Smart Busy Light state and color are restored.

Preprogrammed flashing sequence

The SBL10e device has two modes:

- Nominal mode : the Smart Busy Light application runs properly and sets the light state and color as expected. When a configuration or a firmware upgrade is in progress, the light illumination can be temporarily inconsistent and follows the light flashing sequence shown in the table hereafter.
- Recovery mode : the Smart Busy Light application can not be executed. The light state or color can not be modified anymore. It is required to update the firmware to return to the nominal mode .

Depending on these modes, the Smart Busy Light applications can fall into one of these preprogrammed flashing sequences:

Mode	Smart Busy light behaviour	Information
Recovery	2 very short and consecutive blue flashes (250 ms) with a 4,5 seconds periodicity	The Smart Busy Light application can not be executed. It should never happen. The device console web user interface is so not available. This sequence is displayed until a new firmware update is done with Qether tool. For further information, contact support@qeedji.tech .
Recovery	3 very short and consecutive blue flashes (250 ms) with a 5 seconds periodicity	The software resource of the SBL10e device set at factory are not valid. It should never happen. For further information, contact support@qeedji.tech .
Nominal or recovery	4 very short and consecutive blue flashes (250 ms) with a 5,5 seconds periodicity	A SBL10e device Firmware update is in progress. Please wait a couple of seconds.
Nominal	5 very short and consecutive blue flashes (150 ms)	A SBL10e device configuration is in progress. Please wait a couple of seconds.
Nominal	6 very short and consecutive blue flashes (150 ms)	The datasource is not consistent because some of the parameters are missing or the Date and time is not correct because the NTP server is not activated or not valid. For further information about the datasource form, refer to the chapter § Configuration > Servers. For further information about the reporting of the problems faced with some datasource parameters, refer to the chapter § Maintenance > Logs.

Part II

Applicative console Web user interface

2.1 Applicative console web user interface

The SBL10e device supports a console web user interface that can be accessed with a web browser. The supported web browsers are: Google Chrome , Mozilla Firefox , MS-Edge (Chromium) .

It is available from the URL: http://<device_IP_addr>/ .

The URL falls into the `m365_room` applicative user interface: http://<device_IP_addr>/webui/ . This pane allows to:

- watch the current light state/color:
 - *ON steady/green*: there is no more event scheduled today or the next event scheduled today happens in more than 15 minutes,
 - *ON steady/orange*: a next event happens in less than 15 minutes,
 - *ON steady/red*: an event has started,
- watch the event properties: title, start time, end time:
 - for the one which has started or,
 - for the next one happening today (whose the start date is today).

For any other LED behaviour, refer to the chapter § [Preprogrammed flashing sequence](#).

Examples:

This screenshot shows the 'Application' section of the SBL10e web user interface. It displays the following information:

LED light state:	ON steady
LED light color:	Green
Next booking subject:	No upcoming meeting today
Start date time:	
End date time:	

This screenshot shows the 'Application' section of the SBL10e web user interface. It displays the following information:

LED light state:	ON steady
LED light color:	Orange
Next booking subject:	Project Meeting 1
Start date time:	Tuesday, August 31, 2021, 11:45 AM
End date time:	Tuesday, August 31, 2021, 12:45 PM



Administration console

Application

LED light state: ON steady
LED light color: Red
Next booking subject: Project Meeting 1
Start date time: Tuesday, August 31, 2021, 11:15 AM
End date time: Tuesday, August 31, 2021, 12:15 PM



Administration console

Application

LED light state: ON steady
LED light color: Green
Next booking subject: Project Meeting 1
Start date time: Friday, September 3, 2021, 3:00 PM
End date time: Saturday, September 4, 2021, 6:30 PM

- After a device restart, the time to get the time from to the NTP server (few seconds), the LED color/state is orange/OFF .
- Only the first 128 bytes of the Next booking subject can be displayed.
- In case the meeting summary is more than 128 bytes and contains some unicode characters, the Next booking subject value, the start date and the end date could stay sometimes empty or the Next booking subject value may be displayed with a unexpected shortened value of the original meeting summary. However, the LED state/status continue to be displayed properly.
- Events taking place all day long and accross several days are supported. In this case, the start time and end time are displayed from the current day at 12:00 AM to the day after at 12:00 AM .

Part III

**Administration console Web user
interface**

3.1 Device configuration console web user interface

The SBL10e device supports a device configuration console web user interface that can be accessed with a web browser. The supported web browsers are: Google Chrome , Mozilla Firefox and MS-Edge (Chromium) .

It is available from the URL: http://<device_IP_addr>/ .

The default credentials values, put at factory, to access to the device console web user interface are:

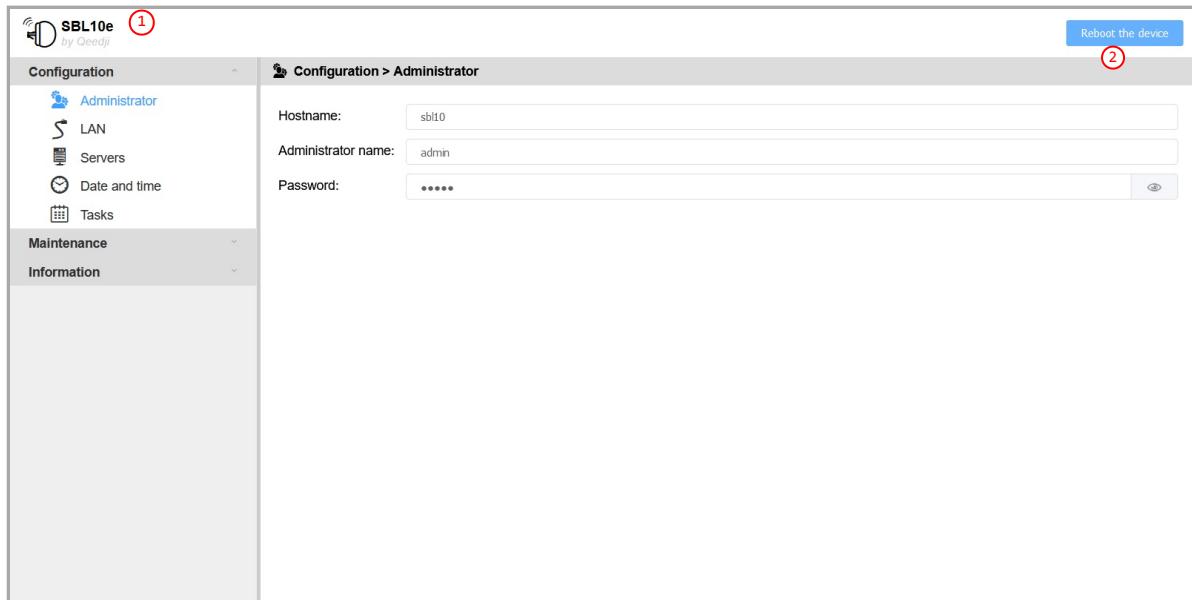
- login: admin ,
- password: admin .

The URL falls automatically into the applicative user interface¹. At the top right corner, click on the Administration Console button.

Administration console

¹ For further information, refer to the chapter § [Applicative console web user interface](#).

This is the device configuration console web user interface.



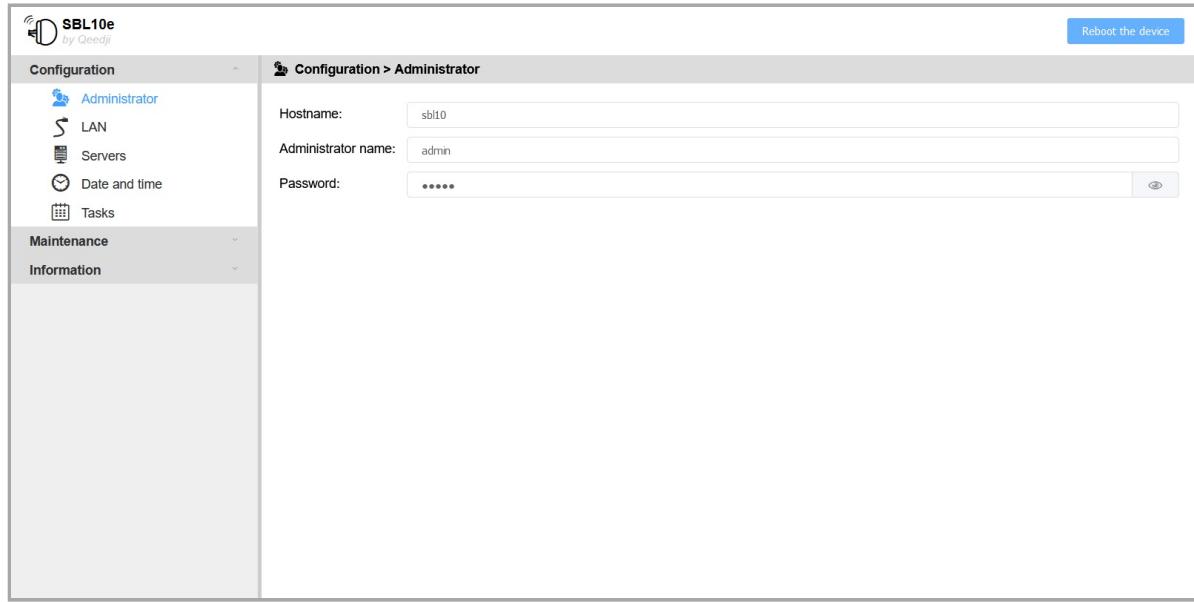
⚠ After you have changed and saved all your settings in the different panes, be sure to perform a device restart by clicking on the Reboot the device (2) button so that your changes are fully reflected.

Click on the device logo (1) at the left top corner to return to the applicative user interface.

3.1.1 Configuration > Administrator

In the Configuration tab, select the **Administrator** menu to change:

- the Hostname ,
- the login credentials:
 - Administrator name ,
 - Password .



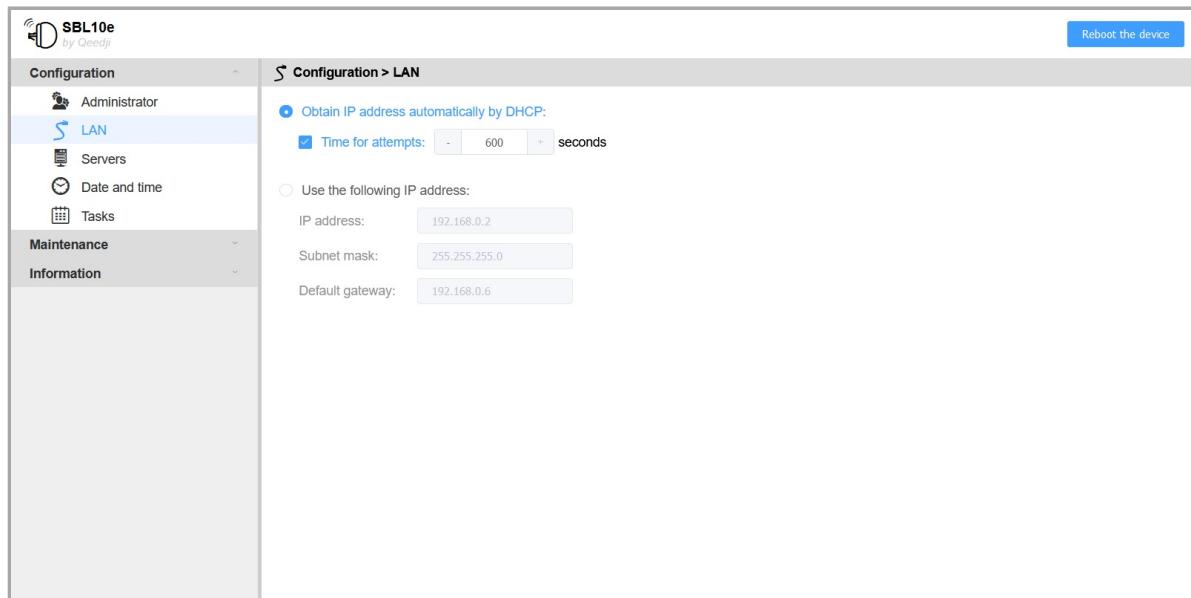
☞ It is recommended that you enter one unique *Hostname* value for each device. In case several SBL10e devices are located in different buildings or geographical locations, we recommend that you enter hostname values with information about the building and the location (e.g. *Hall-RD-Paris-1*).

For security reasons, it may be useful to change the login credentials values. Please keep them in a safe place afterwards.

☞ The same login credentials are used to access to the WebDAV server and to use web services.

3.1.2 Configuration > LAN

In the Configuration tab, select the **LAN** menu to set up the network configuration of the **LAN** interface of your device.



☞ The device supports the UPnP and can be for example detected automatically in the local network environment of your computer.

Enter a suitable LAN network configuration so that the device can access to the web to get the local time with a NTP server.

☞ By default, the device is configured with *Obtain an IP address automatically by DHCP* activated and *Time for attempts* deactivated. As soon as the DHCP server becomes available, the device ends by getting back a valid IP address given by the DHCP server within less than one minute.

☞ After a device reboot, when the device is configured with *Obtain an IP address automatically by DHCP* activated and *Time for attempts* is activated, in case the DHCP server is unavailable after the *Time for attempts* duration (ten minutes for the maximum and default value) has expired, the device ends up using the static IP address entered in the LAN configuration. The default static IP address is 192.168.0.2 when it has never been changed yet by the user. It is recommended to set an appropriate IP address, netmask and gateway if this case would happen. In case a daily reboot task is programmed, the device will restart this operation every days.

☞ When only the *Time for attempts* value is modified, press on TAB key of your keyboard to make appear the *Validate* button.

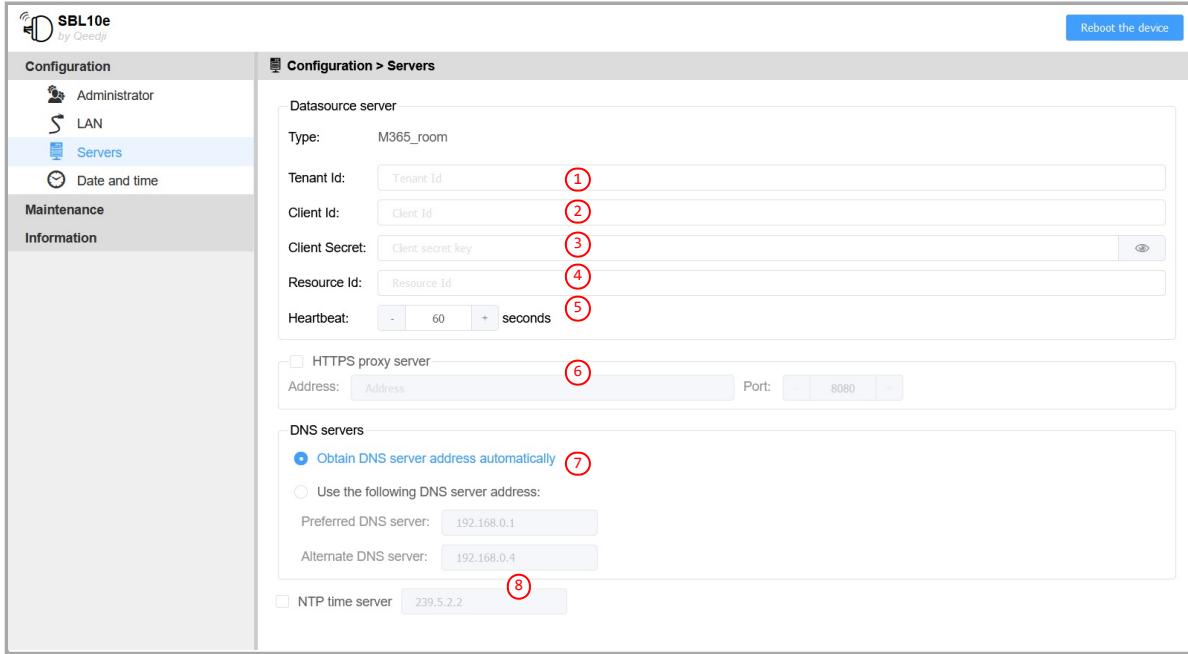
3.1.3 Configuration > Servers

In the Configuration tab, select the **Servers** menu to enter the configuration to connect to the servers peripheral to your device.

The Datasource Server allows, through Azure AD, to access to a dedicated resource id's calendar available in your M365/Office. Depending whether there is meeting programmed for the resource id or not, the `m365_room` application device is applying a state/color value to the busylight:

- Steady/green : previous event finished and no next event happening is less than 15 minutes,
- Steady/Orange : next event happening in less that 15 minutes,
- Steady/Red : an event has started.

☞ When only the heartbeat value is modified, press on TAB key of your keyboard to make appear the `Validate` button.



- Datasource Server :
 - Type : M365_room,
 - Tenant Id ①: Azure AD application Directory Tenant Id ,
 - Client Id ②: Azure AD application client Id ,
 - Client Secret ③: Azure AD application secret value,
 - Resource Id ④: M365 resource email,
 - Heartbeat ⑤: periodicity of the connection to the M365 calendar:
 - from 10 (default value) to 900 seconds,
- HTTPS proxy server ⑥¹
 - Address : enter the IPv4 address, or the domain name of your proxy server,
 - Port : enter the operating port of your proxy server,
- DNS servers ⑦
- NTP time server ⑧: ensure that `NTP time server` is checked and has a valid IP address.

⚠ The `m365_room` application can work properly only when the NTP server is activated, when a valid IP address value of domain name value for the NTP server is set and when a valid IP address id set for the DNS, allowing to the `SBL10e` device to stay always on time.

Using M365 (or Microsoft 365) implies to create an *Azure Active Directory (Azure AD) application* for the busylight `m365_room` application. After consent success, fill with the appropriate values:

- Azure AD application Tenant Id ①,
- Azure AD application Client Id ②,
- Azure AD application Client Secret ③.

☞ The same Tenant Id, Client Id, Client Secret value can be then used for all your SBL10e having a `m365_room` application and connected to resources id coming from the same M365 account.

¹ *HTTPS Proxy server is not supported in the `m365_room` application. Activating proxy server may prevent the device to connect to M365.*

For further information about the procedure to create an *Azure Active Directory* application, refer to the chapter § [Appendix: Azure AD](#).

⚠ Some organization can use an alias for the resource Id email instead of using the official one. For the room Resource Id ④, do ensure to only enter the User principal Name of the resource. For further information, refer to the chapter § [Appendix: Azure AD User Principal Name](#).

This is an example with a M365 room resource id called `room1@contoso.onmicrosoft.com` (fake values):

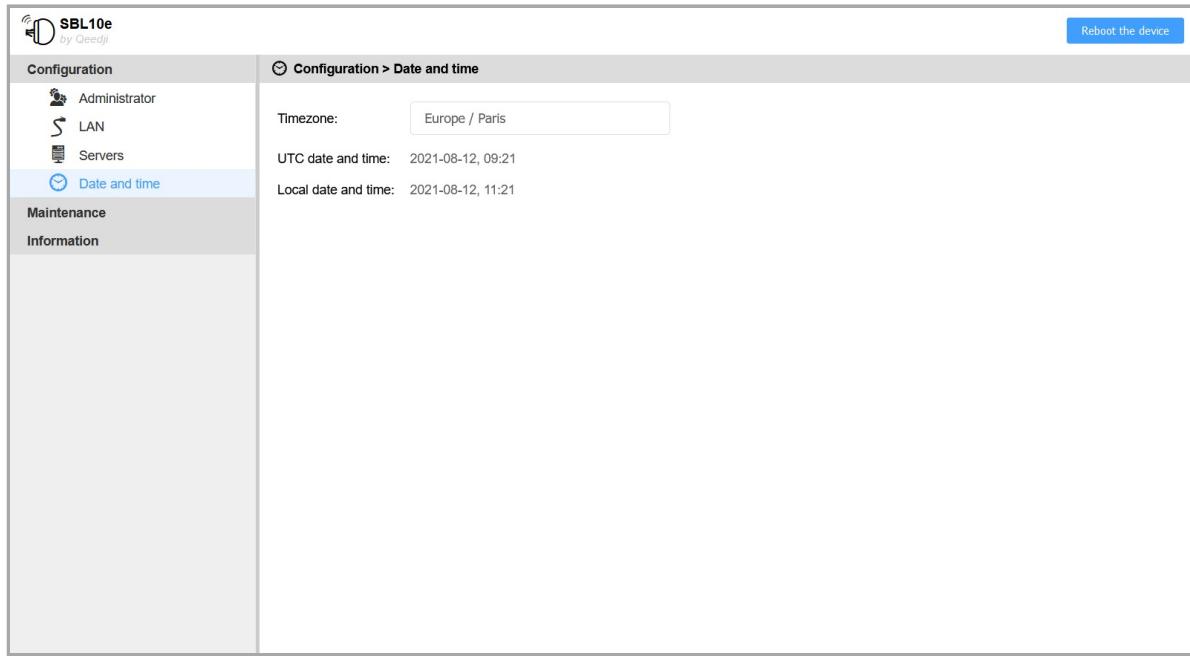
- Azure AD application Client Id : `269d8878-f581-43fe-b53d-fea6c181b7f4`
- Azure AD application Tenant Id : `967b7cc3-847f-41ec-bd74-997a4df1855b`
- Azure AD application Client secret : `6-CdOVxv6p1wwH4y0Q6Yr11SY7.dU-Tt`
- M365 Resource Id : `room1@contoso.onmicrosoft.com`

- When only the `Heartbeat` value is modified, press on TAB key of your keyboard to make appear the `Validate` button.
 - If the server is not available after 20 (default value stored in the `appli.network.datasource.nb_retries_before_cache_reset` user preference) consecutive unsuccessful connection attempts, the light is switched Off until the next successful connection attempt.
- ⚠️** Upgrading the device with another application type will clear the current datasource configuration data. When the device is properly configured, it is advised to build and save an appropriate configuration script (`.js`) by using the configuration script template or save at least the `prefs.json` configuration file of your device. For further information, refer to the chapter § [Maintenance > Files](#).

3.1.4 Configuration > Date and time

In the Configuration tab, select the **Date and Time** menu to check the time configuration:

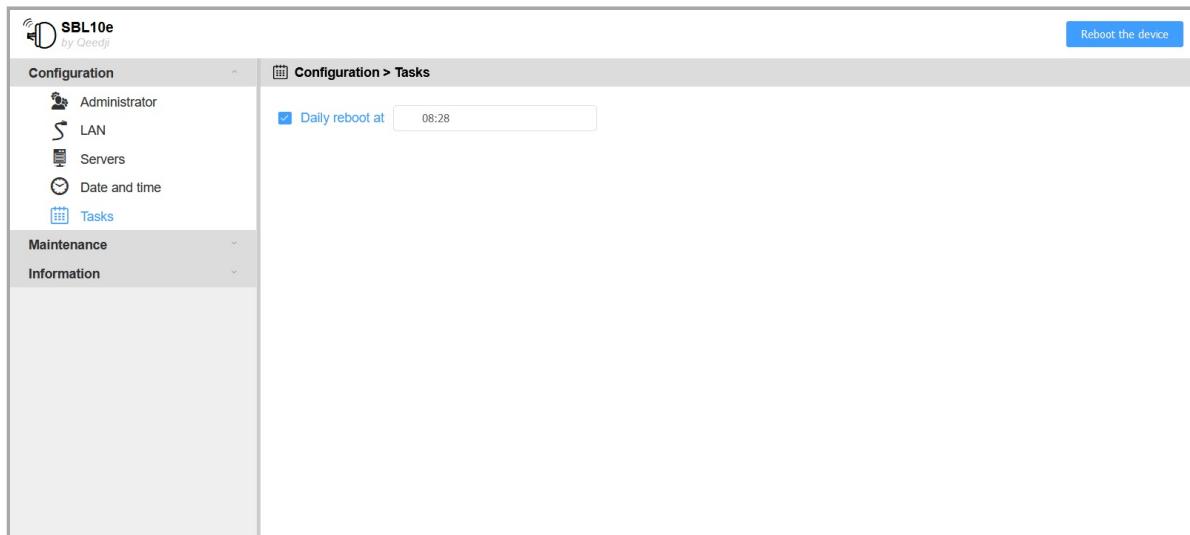
- timezone,
- system date of your device (day and time).



! Your device must be on time and a valid NTP server must be defined. For further information, refer to the chapter § [Configuration > Servers](#).

3.1.5 Configuration > Tasks

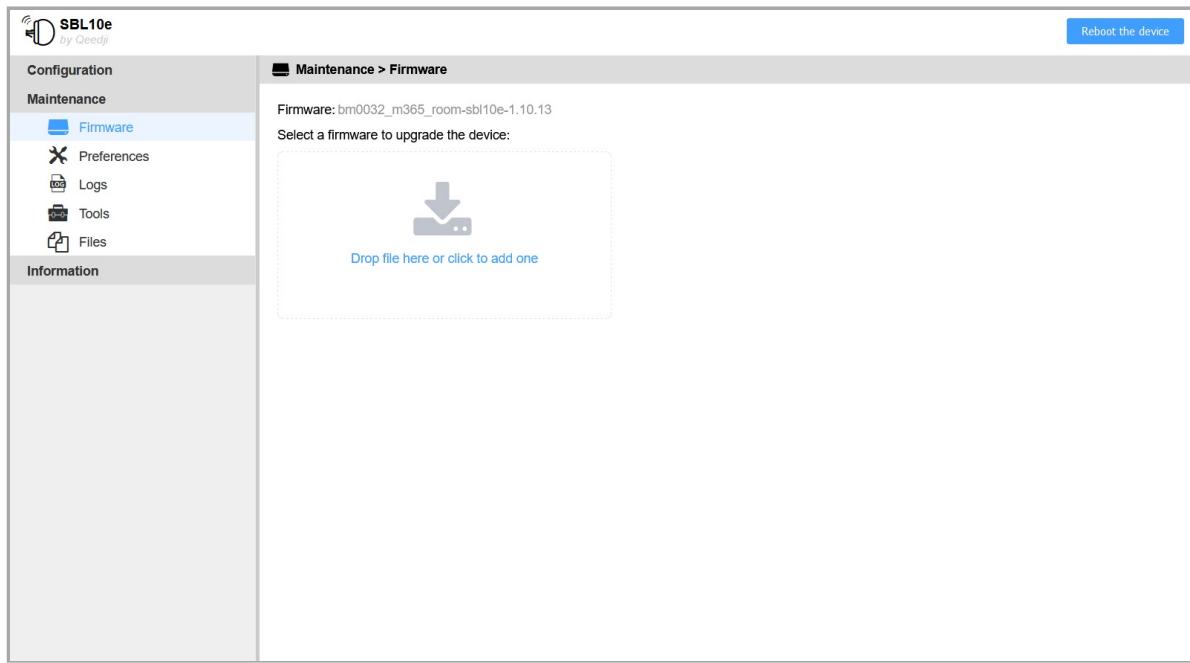
In the Configuration tab, select the **Tasks** menu to activate a device reboot manager task and adjust the reboot manager task time.



- During the reboot task, the light state is *off* for a couple of seconds until the next data source server connection.
- If the NTP server set by the user is not available anymore and the `system.task.reboot.enable` user preference is true, the device is rebooting automatically every days, 24 hours after the last device reboot.

3.1.6 Maintenance > Firmware

In the Maintenance tab, select the **Firmware** menu to view the version of the application installed on your device.



Corrective and evolutive maintenance software versions are regularly made available in the support tab of the official Qeedji website http://www.qeedji.tech/en/support/index.php?SBL10e/M365_room. It is therefore advised to regularly update the device firmware version. From this website, download the appropriate latest firmware version available for your device model (.bin file). For further information, contact support@qeedji.tech.

Drop your .bin file in the Drop file here or click to add one location or click on it to add one, then click on the Send button to update the firmware version of your device. Wait a couple of seconds, the time to load and install the new firmware version. Connect again to the device configuration console web user interface and check the new firmware version.

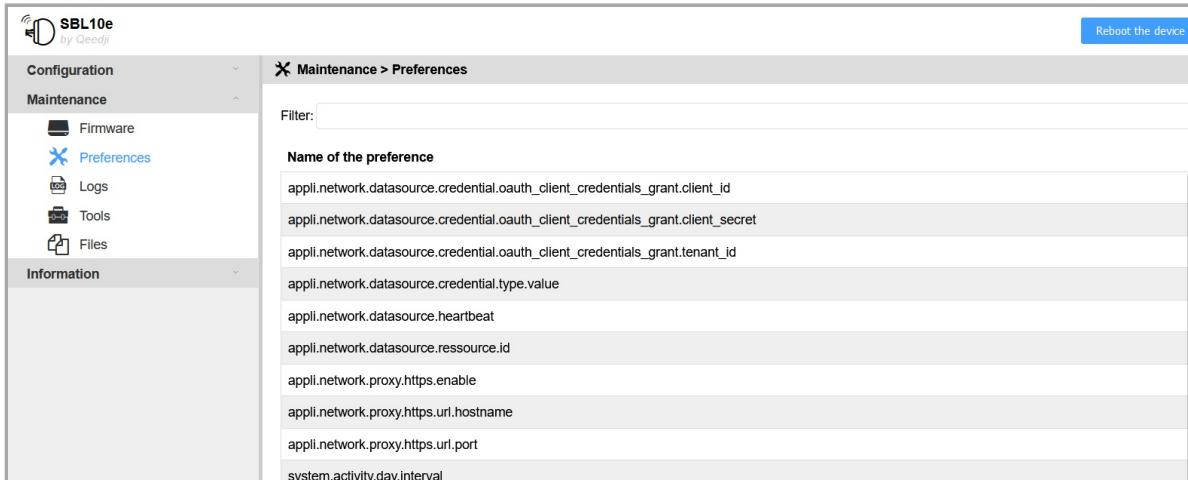
The user preferences common to the applications are kept when upgrading from regular application to another application (and reciprocally) meaning:

- network configuration,
- file system,
- hostname,
- web user interface credentials.

Do not electrically disconnect the device during the firmware upgrade.

3.1.7 Maintenance > Preferences

In the Maintenance tab, select the **Preferences** menu to view all the preferences.



The filter allows to display only the preferences whose name contains the string entered in the filter. All the preferences have optimal default values. Double click on a preference to change its value.

At the bottom right of the page, the `Restore factory preferences` button allows to reset a subset of preferences allowing the device to reprogram its factory preferences. In this case, the LAN network configuration returns to DHCP.

Click on the `Reboot the device` button so that the modifications are taken into account.

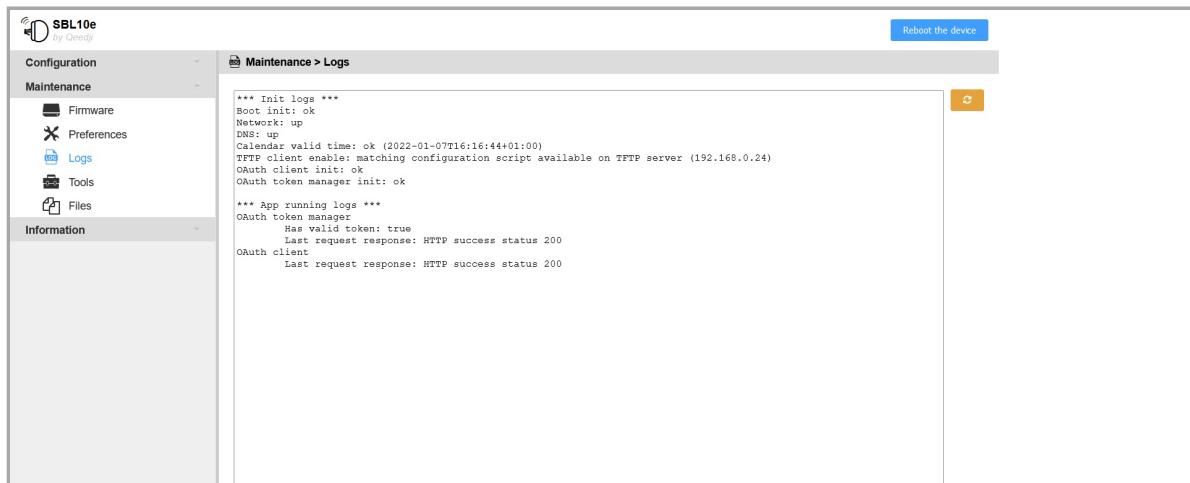
⚠ After a user preference restoration, in case a `.js` configuration script, suitable for the application of this SBL10e device, is available on the `TFTP` server, the user preference `system.tftp.enable` is set to `true`. Consequently, the SBL10e device is rebooting once again to take into account the `.js` configuration script available on the `TFTP` server.

Timezone

Continent	Country/Town pair values supported for the <code>system.datetime.timezone</code> preference
Africa	Africa/Brazzaville, Africa/Cairo, Africa/Casablanca, Africa/Harare, Africa/Lagos, Africa/Nairobi, Africa/Onitsha, Africa/Windhoek
America	America/Anchorage, America/Argentina/Buenos_Aires, America/Barbados, America/Bogota, America/Boston, America/Caracas, America/Chicago, America/Chihuahua, America/Costa_Rica, America/Dallas, America/Denver, America/Godthab, America/Halifax, America/Houston, America/Los_Angeles, America/Manaus, America/Mexico_City, America/Montevideo, America/New_York, America/Phoenix, America/Recife, America/Regina, America/Rio_de_Janeiro, America/San_Francisco, America/Santiago, America/Sao_Paulo, America/St_Johns, America/Tijuana, America/Washington,_D.C.
Asia	Asia/Ahmedabad, Asia/Almaty, Asia/Amman, Asia/Baghdad, Asia/Baku, Asia/Bangalore, Asia/Bangkok, Asia/Beijing, Asia/Beirut, Asia/Chengdu, Asia/Chennai, Asia/Chongqing, Asia/Colombo, Asia/Delhi, Asia/Dongguan, Asia/Dubai, Asia/Guangzhou, Asia/Hangzhou, Asia/Hanoi, Asia/Ho_Chi_Minh, Asia/Hong_Kong, Asia/Hyderabad, Asia/Irkutsk, Asia/Jakarta, Asia/Jerusalem, Asia/Kabul, Asia/Karachi, Asia/Kathmandu, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Kuala_Lumpur, Asia/Kuwait, Asia/Lahore, Asia/Magadan, Asia/Mumbai, Asia/Nagoya, Asia/Nanjing, Asia/Oral, Asia/Osaka, Asia/Pune, Asia/Quanzhou, Asia/Seoul, Asia/Shanghai, Asia/Shenyang, Asia/Shenzhen, Asia/Surat, Asia/Taipei, Asia/Tbilisi, Asia/Tehran, Asia/Tianjin, Asia/Tokyo, Asia/Vladivostok, Asia/Wuhan, Asia/Xi'an, Asia/Yakutsk, Asia/Yangon, Asia/Yekaterinburg, Asia/Yerevan, Asia/Zhengzhou
Atlantic	Atlantic/Azores, Atlantic/Cape_Verde, Atlantic/South_Georgia
Australia	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Hobart, Australia/Perth, Australia/Sydney
Europe	Europe/Amsterdam, Europe/Athens, Europe/Belgrade, Europe/Berlin, Europe/Brussels, Europe/Dusseldorf, Europe/Helsinki, Europe/Istanbul, Europe/London, Europe/Madrid, Europe/Minsk, Europe/Moscow, Europe/Paris, Europe/Sarajevo, Europe/Warsaw
Pacific	Pacific/Auckland, Pacific/Fiji, Pacific/Guam, Pacific/Honolulu, Pacific/Majuro, Pacific/Midway, Pacific/Noumea, Pacific/Tongatapu
UTC	Etc/UTC

3.1.8 Maintenance > Logs

In the Maintenance tab, select the **Logs** menu to view the logs.



In the example, there is no error raised in the logs.

When the `system.tftp.server` user preference is `true`:

- in case there is some available `.js` configuration script on the `TFTP` server with the appropriate file name pattern, this message is printed: `TFTP client enable: matching configuration script available on TFTP server (<IP address>)`.
- in case there is no `.js` configuration script on the `TFTP` server with the appropriate file name pattern, this message is printed: `TFTP client enable: no configuration script available on TFTP server for this device (<IP address>)`.
- in case the `TFTP` server is not available, this message is printed: `TFTP client enable: error server did not respond (<IP address>)`.

When the `system.tftp.server` user preference is `false`: this message is printed: `TFTP client disable`.

⚠️ To be successfully taken into account, the content of the `.js` configuration script available on the `TFTP` server must also be suitable for the SBL10e device and for the middleware version.

In case your device is flashing 6 times every 4 seconds meaning that the device configuration is probably not correct, you are invited to check the logs in this window to try to fix the trouble.

The logs are allowing to know whether:

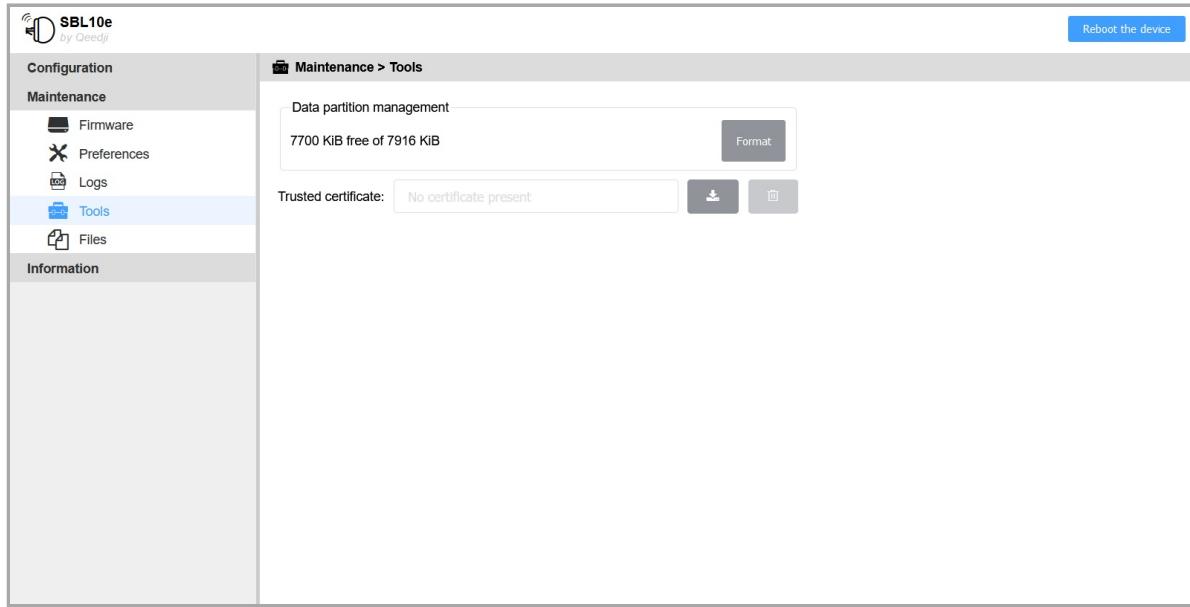
- **the M365 server is available:** if not, check your internet connection and check with your administrator account that your M365 system is available,
- **the Tenant Id is valid:** if not check again that your directory `Tenant Id` value is really existing and some `m365_room` application has been already registered with your Azure AD account,
- **the Client Id is valid:** if not check again that your application `Client Id` value is really existing and some `m365_room` application has been already registered with your Azure AD account,
- **the Client Secret is valid:** if not check again that your application `Client secret` value has been properly copied/pasted at the right location and is really existing and some `m365_room` application has been already registered with your Azure AD account. In case it has been generated with AAD Powershell script, ensure it has been generated with the version of AAD Powershell script V1.10.13 (or above),
- **the Resource Id is valid:** if not, check again that your application `Resource Id` value (resource email) is really existing in your M365/Office account, and is well the `User Principal Name` of the resource (and not an alias).
- **the system date is valid:** if not, check that `NTP time server` is activated and has a valid IP address. After a reboot, in case a connection to the console web user interface is available, the device should be on time.

For any other error, contact support@qeedji.tech.

3.1.9 Maintenance > Tools

In the Maintenance tab, select the **Tools** menu to:

- view the available space on the flash memory storage¹ (max 7916 KiB),
- format the flash memory storage¹,
- add the Trusted certificate (.crt) for the Datasource server .



■ The trusted certificate is not checked in this version.

¹ The flash memory storage is used to store all the directories and files hosted at the root of the WebDAV directory, and the user preferences as well. In case a flash formatting is done, the device returns to the default factory settings. In this case, the trusted certificate is erased and the datasource server data are cleared.

3.1.10 Maintenance > Files

In the **Maintenance** tab, select the **Files** menu to see the directories and files hosted at the root directory of the WebDAV server.

As soon as a modification is done through the device configuration console web user interface, a `prefs.json` file, corresponding to the new device configuration, is created in the `.conf` folder.

When the user preference `system.tftp.enable` is `true`, a `tftp_crc` file, containing the CRC of the `.js` configuration script downloaded from the `TFTP` server is written in the `.conf` folder. To be downloaded again from the `TFTP` server, either the suitable configuration script must be modified on the `TFTP` server, or the `tftp_crc` file must be removed.

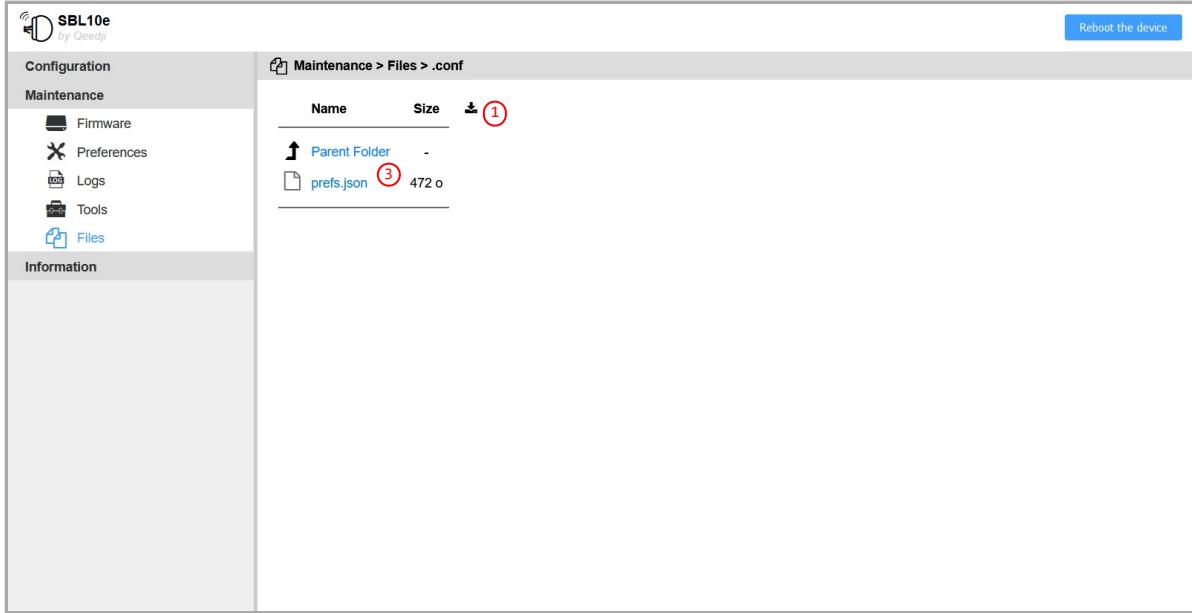
⚠ After having pressed on the `Restore factory preferences` button, the `prefs.json` file and the `tftp_crc` file are erased.

⚠ The content of the `prefs.json` (3) configuration file can be different for `m365_room` application and other applications.

Save or restore the device configuration

⚠ It is recommended to save the `configuration.js` previously to configure your SBL10e device in order to be able to restore its configuration afterwards.

⚠ The content of the `configuration.js` is depending on the used application. Do use the suitable `configuration.js` for the `m365-room` application.



Date and time

The system date and time can only be set and updated thanks to a NTP server. In this case, you have to:

- First:
 - define the timezone value,
 - define the NTP server IP address,
 - check that the NTP time server is activated (activated by default).
- Secondly:
 - define the gateway URL to access to internet,
 - define valid primary and secondary DNS servers.

For further information, refer to the chapter:

- § Configuration > Servers,
- § Configuration > LAN,
- § Configuration > Date and time,

☞ In case the SBL10e device can not fetch a valid date and time through NTP at device boot-up, the clock does not progress and stays with the value `01/01/2020 00:00`. The date and time metadata of the files added after this date on the file system is also `01/01/2020 00:00`. The last modification date for a file can be only be seen with a WebDAV client.

☞ When the server NTP is activated and the device is properly configured, the date and time for the SBL10e device is updated automatically by NTP at the device boot-up. Then it is progressing every seconds.

☞ The support for date and time file metadata display in this pane will be available in a next version.

Application upgrade

The current application can be replaced by pushing a new firmware file `bm0032_m365_room-sbl10e-xx.yy.zz.bin` at the root of the device WebDAV directory `http://<device-ip-addr>/` with a WebDAV client.

After the firmware file pushing, a device reboot is required so that the new firmware file is taken into account.

Configuration update

The configuration of the application can be updated also by pushing an appropriate `.js` configuration script (or a suitable `prefs.json` file) suitable for your application in the `.conf` WebDAV directory (`http://<device-ip-addr>/conf`) with the console web user interface or with a WebDAV client.

⚠ Loading a wrong `prefs.json` would lead to some loss of data like the datasource server configuration. So check the consistency of the `prefs.json` file before uploading it in the device. To avoid any error on the configuration of the application and the configuration of the SBL10e device, it is advised to use a `.js` configuration script (and not with a `prefs.json` file) which is testing before executing anything that it is suitable for the SBL10e device and suitable for the application running on the device. Qeedji provides configuration script template. It is then highly recommended for the user to save an appropriate `.js` configuration script for each SBL10e device installed in his building.

A `000000000000.js` template is available for download [here](#).

In this case, the file pattern must be either:

- `configuration.js` : suitable for any device whatever its MAC address,
- `000000000000.js` : suitable for any device whatever its MAC address,
- `<device_LAN1_MAC_address>.js` (with the format `ABCDEFABCDEF.js`) : suitable for device whose MAC address is matching.

After having downloaded the configuration script template:

- edit the `000000000000.js` configuration script and uncomment/modify the appropriate lines according to your needs,
- rename the configuration script if required,
- once saved, drop it in the `.conf` WebDAV directory like explained above,
- when the `.js` configuration script is satisfying, save it preciously to be able to restore its configuration after wards.

After a `.js` configuration script uploading in the device, the device is rebooting automatically once to take the new configuration into account.

☞ The `prefs.json` file is available in the `.conf` WebDAV directory of the devive as soon as the SBL10e device configuration is modified at least once by the user. After a device configuration updating with a `prefs.json` file, a device reboot is required so that the new configuration is taken into account.

☞ Pushing a `.js` configuration script in the `.conf` WebDAV directory (`http://<device-ip-addr>/conf`) with a WebDAV client could raise a warning at the WebDAV client end, after the `.js` file transferring is completed because the device is automatically rebooting once when it is received. For example, after the `.js` file sending with BitKinex WebDAV is done, another network request is done by the WebDAV client while the device is currently rebooting. So a WebDAV error at the WebDAV client end leads to an automatic file resending which is causing another device reboot and so on, and this, until the WebDAV client application is closed. For example, after the `.js` file sending with CarotDAV WebDAV client, the error leads only to the displaying of a warning message. The user has just to ignore the error at the WebDAV client end.

3.1.11 Information > Device

In the **Information** tab, select the **Device** menu to view system information about the device.

The screenshot shows the Qeedji web interface with the title "SBL10e by Qeedji". On the left, there is a sidebar with navigation links: Configuration, Maintenance, Information, Device (selected), and Network. The main content area is titled "Information > Device" and displays the following device information:

Firmware:	bm0032_m365_room-sbl10e-1.10.13
Model:	SBL10e
Manufacturer:	Qeedji
Manufacturer URL:	www.qeedji.tech
Hostname:	sbl10
UUID:	08400004-0000-0000-0000-001ce6024cad
PSN:	01320-00004

A blue button in the top right corner says "Reboot the device".

- **Firmware** : label and version of the firmware embedded in the device,
- **Model** : model of the Qeedji device,
- **Manufacturer** : product manufacturer name,
- **Manufacturer URL** : manufacturer Website,
- **Hostname** : name of the device on the network,
- **UUID** : Universal Unique IDentifier,
- **PSN** : Product Serial Number.

3.1.12 Information > Network

In the **Information** tab, select the **Network** menu to view a summary of the device's network configuration.



The screenshot shows the SBL10e device interface. On the left, there is a navigation sidebar with the following options: Configuration, Maintenance, Information, Device, and Network. The Network option is currently selected, indicated by a blue background. The main content area is titled "Information > Network". Under this title, there is a section for "LAN_1" with the following details:

LAN_1	
MAC Address:	00:1c:e6:02:4c:ad
IPv4 Address:	192.168.1.47/17 [DHCP]
Default Gateway:	192.168.0.1
DNS Servers:	192.168.0.1

At the top right of the main content area, there is a blue button labeled "Reboot the device".

Part IV

Technical information

4.1 Technical specifications

Model	Manufacturer
SBL10e	Qeedji
Power supply	Information
PoE IEEE802.3af	POE power supply input: ES1 / PS2 (48 V – 100 VA)
Processors	
CPU	Nordic Semiconductor nRF52
Security processor	ARM CryptoCell 310
Storage	
Flash Memory for file system	8 MBytes
Network	Other information
1x Ethernet	10/100 Base T, male connector
WPAN	
Bluetooth Low Energy 5	
Frequency band: 2.402 to 2.480 GHz	
Tx Power: +8 dBm	
Operating temperature	Storage temperature
+0 °C to +40 °C	-20 °C to +60 °C
+32 °F to +104 °F	-4 °F to +140 °F
Operating humidity	Storage humidity
< 80 %	< 85 %
Weight	Dimensions (W x H x D) (RJ45 male connector included)
35 g	60,5 mm x 60,5 mm x 67 mm
0,077 lb	2,36" x 2,36" x 2,63"
Plastic enclosure flame rating	
Base: PVC UL 94-5VA, bulb: Polycarbonate UL 94 V-2	
Warranty	
1 year	

4.2 Conformities

EUROPE

In conformity with the following European directives:

- LVD 2014/35/EU ,
- EMC 2014/30/EU ,
- RED 2014/53/EU .

USA

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference's by one or more of the following measures:

- reorient or relocate the receiving antenna,
- increase the separation between the equipment and the receiver,
- connect the equipment into an outlet on a circuit different from that to which the receiver is connected,
- consult the dealer or an experienced radio/TV technician for help.

This equipment complies with FCC's radiation exposure limits set forth for an uncontrolled environment under the following conditions:

- this equipment should be installed and operated such that a minimum separation distance of 20 cm is maintained between the radiator (antenna) and user's/nearby person's body at all times,
- this transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

- this device may not cause harmful interference,
- this device must accept any interference received, including interference that may cause undesired operation.

Qeedji is not responsible for any changes or modifications not expressly approved by the party responsible for compliance. such modifications could void the user's authority to operate the equipment.

CANADA

This device contains licence-exempt transmitter(s)/receiver(s) that comply with Innovation, Science and Economic Development Canada's licence-exempt RSS(s). Operation is subject to the following two conditions:

- this device may not cause interference,
- this device must accept any interference, including interference that may cause undesired operation of the device.

Part V

Contacts

5.1 Contacts

For further information, please contact us:

- **Technical support:** support@qeedji.tech,
- **Sales department:** sales@qeedji.tech.

Refer to the Qeedji Website for FAQ, application notes, and software downloads: <https://www.qeedji.tech/>

Qeedji FRANCE
INOVELEC INNES SA
5A rue Pierre Joseph Colin
35700 RENNES

Tel: +33 (0)2 23 20 01 62

Part VI

Appendix

6.1 Appendix: web services

These are the supported web services for the `m365_room` application to command and control the SBL10e:

Webservice path	HTTP method				Query string parameters	Body	Function	From the <code>m365_room</code> application version
	GET	POST	PUT	DELETE				
<code>api/v1/sys/power</code>		yes			<code><state>=rebooting</code>	""	Reboot the device	1.10.13
<code>api/v1/software/version</code>	yes				None	i.e.: {"value": "1.10.13"}	Get the device delivery software version	1.10.13
<code>api/v1/software/label</code>	yes				None	{"value": "bm0032_m365_room"}	Get the device delivery software label	1.10.13
<code>api/v1/sys/sn</code>	yes					PSN Short representation: i.e.: {"value": "01320-00004"}	Get the device SN	1.10.13
<code>api/v1/sys/model-name</code>	yes				None	{"value": "SBL10e"}	Get the device model name	1.10.13
<code>api/v1/sys/manufacturer</code>	yes				None	{"value": "Quedji"}	Get the manufacturer	1.10.13
<code>api/v1/sys/manufacturer-url</code>	yes				None	{"value": "www.quedji.tech"}	Get Web Site of manufacturer	1.10.13
<code>api/v1/sys/uuid</code>	yes				None	Uuid string value: <uuid> = <psn>-<48x0>-<mce-48> i.e.: {"value": "08400004-0000-0000-0000-001ce6024cad"}	Get the device UUID	1.10.13
<code>api/v1/wpan1/mac</code>	yes				None	Bluetooth MAC address value user formatted: i.e.: {"value": "db:f0:8c:72:64:a3"}	Get the device Bluetooth MAC address	1.10.13
<code>api/v1/leds/light</code>	yes				None	Get (plain text, separator CR): i.e.: {"state": "steady", "color": "blue"} <code><state>= off steady flashing</code> <code><color>= red orange blue yellow green</code>	Get busylight led color and state	1.10.13
<code>api/v1/sys/datetime</code>	yes				None	{"value": "2021-08-11T06:09:58-02:30"}	Get busylight date and time	1.10.13

Examples syntax with CURL tool:

- reboot the device:

```
curl --user "<USERNAME>:<PASSWORD>" -i -X PUT "http://<DEVICE_IP_ADDR>/api/v1/sys/power?state=rebooting"
```

- get device state & color:

```
curl --user "<USERNAME>:<PASSWORD>" -X GET "http://<DEVICE_IP_ADDR>/api/v1/leds/light"
```

- get device firmware version:

```
curl --user "<USERNAME>:<PASSWORD>" -X GET "http://<DEVICE_IP_ADDR>/api/v1/software/version"
```

6.2 Appendix: Qether

In case an application can not be executed, the SBL10e returns to a `Recovery` mode, waiting for firmware update.

The provided `Qether` tool allows to make some remote operations on the SBL10e, based on its device MAC address like:

- SBL10e device firmware upgrade,
- SBL10e device configuration update,
- SBL10e device reboot.

The `<product_type>` is an extract of the device PSN value. For example, the `0132x-xxxx` PSN value leads to the `0132 <product_type>`.

The `<SBL10e_device_MAC_address>` is the MAC address of the device with the format `00:1C:E6:AB:CD:EF`.

 The MAC address of the device is written on the label stuck at the back of the SBL10e device with the format `00-1C-E6-AB-CD-EF`.

Discover command example

This command allows to find out the SBL10e devices available on the local network:

```
qether.exe FF 0132 discover
```

Configuration command syntax

Send a `.js` configuration script and apply it (default parameters):

```
qether.exe <SBL10e_device_MAC_address> <product_type> configure -f configuration.js
```

- When using `Qether`, no specific filename pattern is required for the `.js` configuration script, except the `.js` file extension.
- The `system.httpd.username` preference value is limited to 15 characters max. The `system.httpd.password` preference value is limited to 100 characters max. The alphanumeric characters and the following characters `{}/~[]!#$_$&()/:<=@|^%?+~(((),'` are supported for the `system.httpd.username` and `system.httpd.password` preference values.
- The `system.hostname` preference value is limited to 15 characters max. The alphanumeric characters, the character `-` and the character `.` are supported for the `system.hostname` preference value.
- To get an IP address with the DHCP server, set `system.lan1.ipv4.static-addr` with the value `false`. Else to work with a static IP address, set `system.lan1.ipv4.static-addr` with the value `true`.

Reboot command syntax

Reboot the target device:

```
qether.exe <SBL10e_device_MAC_address> <product_type> reboot
```

Firmware upgrade command syntax

Send a firmware file, with default transfer parameters, and install it. For example:

```
qether.exe <SBL10e_device_MAC_address> <product_type> install -f bm0032_m365_room-sbl10e-setup-1.11.12.bin
```

 *Qether needs first to be installed first on your MS-Windows computer. For further information, refer to the [Qether user manual](#).*

6.3 Appendix: Device configuration with TFTP server (+ DHCP server code 66)

The SBL10e device can be configured thanks to a `.js` configuration script (Javascript) hosted on a `TFTP`¹ server associated to a `DHCP` server (code 66 option) properly configured and available on the local network.

¹ Trivial File Transfer Protocol

The `.js` configuration script downloading can be done as soon as a `DHCP` server is available, even whether the device is configured with a static IP address. Once connected to the `DHCP` server, the device can get:

- the IP address value of its network interface, when the option `Obtain IP address automatically by DHCP` is activated then,
- the primary DNS value when the `system.lan1.dns.static` user preference is `false` then,
- the `.js` configuration script from the `TFTP` server when the `system.tftp.enable` user preference is `true`.

Prerequisites:

- the appropriate `.js` configuration script must be available in the exported directory of the `TFTP` server. It must:
 - be suitable for the device, its firmware type and its firmware version,
 - match an appropriate filename pattern:
 - `000000000000.js` or,
 - `<device_LAN1_MAC_address>.js` (with the format `ABCDEFABCDEF.js`).

When a `.js` configuration script is modified on the `TFTP` server, the device must be restarted once so that the new configuration script is taken into account by the device.

☞ When using a `TFTP` server, the `configuration.js` filename pattern is not supported.

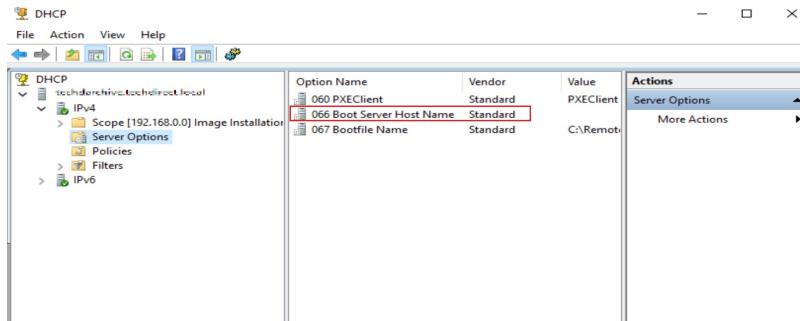
⚠ The downloading of a `.js` configuration script from a `TFTP` server can be done only at the device booting-up and when the device has never downloaded it before or when the script content has been modified since the last download (CRC check).

DHCP server configuration

The `DHCP` server must be configured to be associated to a `TFTP` server. For that, you need to use code 66 option (TFTP Server), using the IPv4 address value of the `TFTP` server.

For example, for a Microsoft `DHCP` server, you need to define the option `Boot Server Host Name` and give the IPv4 address of the `TFTP` server. It can be in `Extended option` and/or `Server Options`.

☞ The service must be restarted so that the modifications are fully reflected.

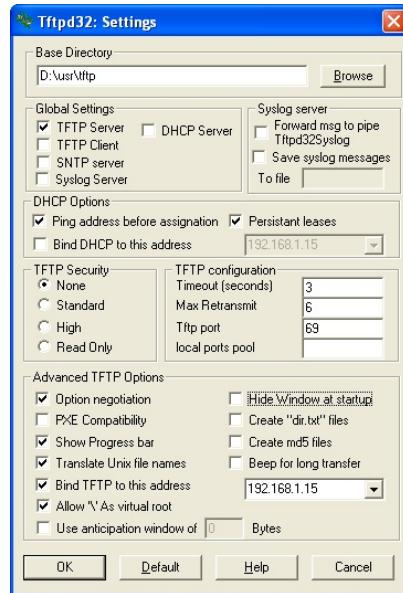


TFTP server configuration

The configuration is depending on the used software client. In all cases, you need to:

- get the directory URL that can be seen by `TFTP` clients,
- choose a `TFTP` security `None`,
- keep the default port (69).

Here is an example of the `tftpd32` software with MS-Windows.



In this example, the `TFTP` server address is `192.168.1.15` and the exported directory is `D:/usr/tftp`.

☞ In this pane, enter the IP address of the `TFTP` server. Indeed entering the `TFTP` server domain name may prevent the feature to work properly.

Copy the `.js` configuration script in the exported directory of the `TFTP` server.

☞ It is recommended to have one `.js` configuration script per device by following the pattern `<MAC>.js`.

6.4 Appendix: Check Azure AD User Principal Name

To get the only suitable name for your resource, you have to use the *User principal name* of your resource.

Connect to the Microsoft Azure portal with your Administrator login credentials then open the `Users` menu on the left.

Display Name	Email Address	Member	Last Sign-in	Object ID
CS 02	cs02@innesrd.onmicrosoft.com	Member	No	innesrd.onmicrosoft.co
CS 03	cs03@innesrd.onmicrosoft.com	Member	No	innesrd.onmicrosoft.co
CS 04	cs04@innesrd.onmicrosoft.com	Member	No	innesrd.onmicrosoft.co
CS 05	cs05@innesrd.onmicrosoft.com	Member	No	innesrd.onmicrosoft.co
CS 06	cs06@innesrd.onmicrosoft.com	Member	No	innesrd.onmicrosoft.co
CS 07	cs07@innesrd.onmicrosoft.com	Member	No	innesrd.onmicrosoft.co
Customer Su...	cs@innesrd.onmicrosoft.com	No	No	innesrd.onmicrosoft.co
Delegate2	delegate2@innesrd.onmicrosoft.com	No	No	innesrd.onmicrosoft.co
DEMO 01	demo01@innesrd.onmicrosoft.com	Member	No	innesrd.onmicrosoft.co
DEMO 02	demo02@innesrd.onmicrosoft.com	Member	No	innesrd.onmicrosoft.co
DEMO 03	demo03@innesrd.onmicrosoft.com	Member	No	innesrd.onmicrosoft.co

Select the appropriate resource to see its *User principal name*.

Name	Value
User Principal Name	demo01@innesrd.onmicrosoft.com

Get list of Azure AD User Principal Names with Powershell

For the Graph module of Azure Active Directory PowerShell, you must use PowerShell version 5.1.

```
PS C:\WINDOWS\system32> $PSVersionTable

Name           Value
----           -----
PSVersion      5.1.19041.1023
PSEdition      Desktop
PSCompatibleVersions {1.0, 2.0, 3.0, 4.0...}
BuildVersion   10.0.19041.1023
CLRVersion     4.0.30319.42000
WSManStackVersion 3.0
PSRemotingProtocolVersion 2.3
SerializationVersion 1.1.0.1
```

These procedures are intended for users who are members of a Microsoft 365 administrator role group.

Open an elevated MS-Windows PowerShell command prompt window running MS-Windows PowerShell as an administrator.

```
PS C:\WINDOWS\system32> Install-Module -Name AzureAD
```

By default, the *PowerShell Gallery (PSGallery)* is not configured as a trusted repository for *PowerShellGet*. The first time you use the *PSGallery*, you will see the following message:

Untrusted repository

You are installing the modules from an untrusted repository. If you trust this repository, change its InstallationPolicy value by running the `Set-PSRepository` cmdlet.

Are you sure you want to install the modules from 'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"):

Type [A] for Yes to All.

```
PS C:\WINDOWS\system32> Connect-AzureAD
```

Once connected, you can use the cmdlets of *Azure Active Directory PowerShell module for Graph*.

```
PS C:\WINDOWS\system32> Get-AzureADUser
```

ObjectId	DisplayName	UserPrincipalName	UserType
bb1ff602-943c-42dc-a890-45caf0504afa	DEMO 01	demo01@innesrd.onmicrosoft.com	Member
bb1ff602-943c-42dc-a890-45caf0504afb	DEMO 02	demo02@innesrd.onmicrosoft.com	Member
bb1ff602-943c-42dc-a890-45caf0504afc	DEMO 03	demo03@innesrd.onmicrosoft.com	Member
bb1ff602-943c-42dc-a890-45caf0504afd	DEMO 04	demo04@innesrd.onmicrosoft.com	Member

Get the resource user principal name from a resource email alias

Some organization use a resource email alias instead of using the resource user principal name, to avoid to use very long resource email values. The resource email alias is not supported in SBL10e device. To know the user principal name of a resource email value, type the cmdlet with the syntax below:

```
Get-Mailbox -Identity <resource_email_address> | Format-List UserPrincipalName
```

Example

```
PS C:\WINDOWS\system32> Get-Mailbox -Identity demo01_alias@innes.com | Format-List UserPrincipalName
```

```
UserPrincipalName: demo01@innesrd.onmicrosoft.com
```

6.5 Appendix: Create M365 application with Powershell

Download the `Powershell_Innes_AAD-1.10.18.zip` from the [Innes Site Web](#) then follow the instructions below.

Introduction

This set of `Powershell` functions allows to:

- create an *Azure Active Directory* application, with the `New-AADApplication` function,
- remove an *Azure Active Directory* application, with the `Remove-AADApplication` function.

These functions are defined in the `PSAAD` PowerShell module stored in the `Modules\PSAAD\` directory.

The result of the `Powershell` functions is also stored in a JSON file.

Edit the file and store previously the values which could be required for your application:

- the `clientId` value,
- the `tenantId` value,
- the `clientSecret` value.

Security

By default, the execution of local `Powershell` scripts are not allowed. You can change their execution rights by changing the `PowerShell` security policy. This modification has to be done once with the `Set-ExecutionPolicy` `Powershell` function. Your organization may have to change it according to your security rules.

For example, to authorize the execution of all scripts, launch a `Powershell` console with administrator rights, and type:

```
PS > Set-ExecutionPolicy -ExecutionPolicy Unrestricted -scope CurrentUser
```

For further information, look at the cmdlet `Set-ExecutionPolicy` help page.

If you cannot allow the execution of unsigned local scripts, you can install the provided certificate in the list of authorized root certificates with the command:

```
PS > cd <your_path_to_the_scripts>\Powershell_Innes_AAD\Certificate\  
PS > Import-PfxCertificate -FilePath InnesCodeSigningRootCA_1.pfx -CertStoreLocation  
cert:\CurrentUser\Root -Password $(ConvertTo-SecureString "1234" -AsPlainText -Force)
```

To import the `.pfx` certificate, you can also use the MS-Windows application `certmgr.msc`, select the `Trusted Root Certification Authorities`, right click on `All Tasks`, select the `Import` item, select the file and enter the password `1234`. When ended, close the current `Powershell` console.

Prerequisite

Install the Azure AD module

Install the `AzureAD` module with the command below:

```
PS > Install-Module -name AzureAD -scope CurrentUser
```

Dependency

If this message is prompted, enter `Y`.

```
The NuGet supplier is required to continue  
PowerShellGet requires the NuGet vendor, version 2.8.5.201 or later, to interact with the repositories.  
The NuGet provider must be available in "C:\Program Files\PackageManagement\ProviderAssemblies" or ..../..  
"C:\Users\<username>\AppData\Local\PackageManagement\ProviderAssemblies".  
You can also install the provider NuGet by executing the command "Install-PackageProvider -Name NuGet ..../..  
-MinimumVersion 2.8.5.201 -Force". Do you want that PowerShellGet installs and imports the NuGet provider now?  
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"):
```

If this message is prompted, enter `Y`.

```
Unapproved repository  
You install the modules from an unapproved repository. If you approve this repository, change its ..../..  
InstallationPolicy value by running the Set-PSRepository command applet. Do you really want to install From PSGallery ?  
[Y] Yes [T] Yes for all [N] No [U] No for all [S] Suspend [?] Help (default is "N"):
```

Usage

To use one of the `Powershell` modules, you have to define the environment variable for `PSAAD`. You have 3 possibilities:

1. Either copy the directories under `Modules\` into a standard Powershell module installation directory, for example `c:\Program Files\WindowsPowerShell\Modules`. Then launch a Powershell console.
2. Or redefine the search variable for Powershell modules (the `$Env:PSModulePath` Powershell variable) each time you will use these functions. In this case, launch a Powershell console, and type the line below, adapting it to your path. Each time you launch a new Powershell console, you need to enter it again.

Example:

```
PS > $Env:PSModulePath="$Env:PSModulePath;C:\Program Files (x86)\WindowsPowerShell\Modules"
```

3. Or redefine the search variable for Powershell modules in the Windows environment variables. For that, add the path `<your_path_to_the_scripts>\Powershell_Innes_AAD\Modules` to the environment variable `PSModulePath`. Then, launch afterwards a Powershell console.

To use the functions or get help, you must then import the module(s) with the `Import-Module` function. Example:

```
PS > Import-Module PSAAD
```

Depending on how you get the scripts, you may have this following warning:

```
Security Warning Run only scripts that you trust. While scripts from the Internet can be useful,
this script can potentially harm your computer. Do you want to run \server\scripts\my.ps1?
[D] Do not run [R] Run once [S] Suspend [?] Help (default is "D"):
```

To avoid this message, you can unblock the script files (to do only once):

```
PS > cd <your_path_to_the_scripts>\Powershell_Innes_AAD\
PS > dir -Recurse | Unblock-File
```

The `Get-Command` function allows you to list the functions defined in a module. Example:

```
PS > Get-Command -Module PSAAD
```

Answer example:

CommandType	Name	Version	Source
-----	---	-----	-----
Function	New-AADApplication	1.10.18	PSAAD
Function	Remove-AADApplication	1.10.18	PSAAD

You can get help on each function of the module by using the standard cmdlet `Get-Help` with options:

- `-detailed`,
- `-full`,
- `-examples`.

Example:

```
PS > Get-Help -detailed New-AADApplication
```

NAME
New-AADApplication

SYNOPSIS
This function creates a Azure Active Directory application.

SYNTAX
New-AADApplication [[-Credential] <PSCredential>] [[-tenantId] <String>] [-appName] <String> [-authorizations] <String[]> [[-LogFile] <String>] [[-PublicClient] <Boolean>] [[-multiTenants] <Boolean>] [<CommonParameters>]

DESCRIPTION
This function creates a Azure Active Directory application.

PARAMETERS
-Credential <PSCredential>
 Credential (admin profile) used to create the Azure Active Directory application. If absent, a dialog is displayed in the browser to enter the credentials.

-tenantId <String>
 Azure Active Directory Tenant Id of the tenant in which the application has been created. This parameter is not mandatory. If absent, the tenantId is retrieved automatically after the credentials have been entered in the dialog.

-appName <String>
 Name of the Azure Active Directory application.

-authorizations <String[]>
 Authorization type:
 - "signcom_m365" : to access to M365 files and folders resources and Websites for SignCom application
 - "url_launcher_m365" : to access to M365 Websites for URL launcher application
 - "signmeeting_ews": to access to MS-Exchange room mailbox resources for SignMeeting MS-Exchange application
 - "signmeeting_m365": to access to M365 room mailbox resources for SignMeeting-M365 application
 - "briva_calendar_ews": to access to MS-Exchange room mailbox resources for Briva Calendar EWS application
 - "m365_room": to access to M365 room mailbox resource for SBL10e m365_room application
 - "m365_user": to access to M365 user presence resource for SBL10e m365_user application
 - "powerbi": to access to Power BI report and Power BI dashboards

-LogFile <String>
 Log file path

-PublicClient <Boolean>
 The application is a public client (false by default).
 In this mode, it is possible to obtain a user type connection with the pair (username, password) only (without secret client).
 But then the application type premissions and the secret client are no longer generated; it is then no longer possible to connect in application mode.
 Only authorization types "signcom_m365", "url_launcher_m365", "m365_user" and "powerbi" are allowed for a public client.

-multiTenants <Boolean>

<CommonParameters>
 This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<https://go.microsoft.com/fwlink/?LinkID=113216>).

----- EXAMPLE 1 -----

PS C:\>\$result = New-AADApplication -appname "SignMeeting" -authorizations "signmeeting_ews"

A consent request will be sent in 30 seconds in your browser.
You must log into an administrator account of your organization and grant the necessary permissions.

PS C:\>\$result

Name	Value
clientId	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
objectId	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
spId	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
name	SignMeeting
tenantId	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
clientSecret	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

REMARKS
To see the examples, type: "get-help New-AADApplication -examples".
For more information, type: "get-help New-AADApplication -detailed".
For technical information, type: "get-help New-AADApplication -full".

Create an Azure Active Directory application for SBL10e

For example, to create a *SBL10e-M365room* (free text) Azure AD application for *m365_room*, generate the client Id, the tenant Id and the client secret and store temporarily these values in the *sbl10em365_room* variable:

```
PS > $sbl10e_m365_room = New-AADApplication -appname "SBL10e-M365_room" -authorizations "m365_room"
```

- Don't use an already existing appname else an error is returned.
- Don't use space characters in appname else an error is returned.
- ⚠ Clicking on a Powershell window can suspend the command. In this case click again in the window to resume the command.

A login popup is displayed. Enter once your M365 login credentials.

This message is then displayed in a *Powershell* context.

```
You must log into an administrator account of your organization and grant the necessary permissions.  
A consent request will be sent within 30 seconds in your browser.
```

After 30 seconds, a login popup should be prompted (<https://login.microsoftonline.com/>) automatically in your default web browser.

Enter again your M365 login credentials.

A new popup message with the *Permission requested, review for your organization* title is prompted in your web browser. Press on the *Accept* button. Then a message is displayed in your web browser showing that the consent is successful: *Success of the consent request*.

You can view the data of the created application by typing the following syntax

- ⚠ The following variable name is the same as the one you have used in the previous command above.

For example, to display the result of the previous command allowing to watch the *client Id*, the *tenant Id* and the *client secret* values:

```
PS > $sbl10e_m365_room  
Name          Value  
----          ----  
clientId      xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx  
objectId      xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx  
spId          xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx  
name          SBL10e-M365_room  
tenantId     xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx  
clientSecret  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

The result of the *Powershell* function is also stored in a JSON file (in the example: *SBL10e-M365_room.json*).

Edit the file and store preciously the values required for your application:

- the *clientId* value,
- the *tenantId* value,
- the *clientSecret* value.

Example to delete an Azure Active Directory application

```
PS > Remove-AADApplication -appname "SBL10e-M365_room"
```

A login popup is opened. Enter your M365 credentials.

In case the values do not allow SBL10e m365_room to work properly, check in Microsoft Azure portal that the application has been created successfully and the rights are properly granted. If not, wait for a while, the rights granting may take few hours.

6.6 Appendix: Create M365 application with Azure AD portal

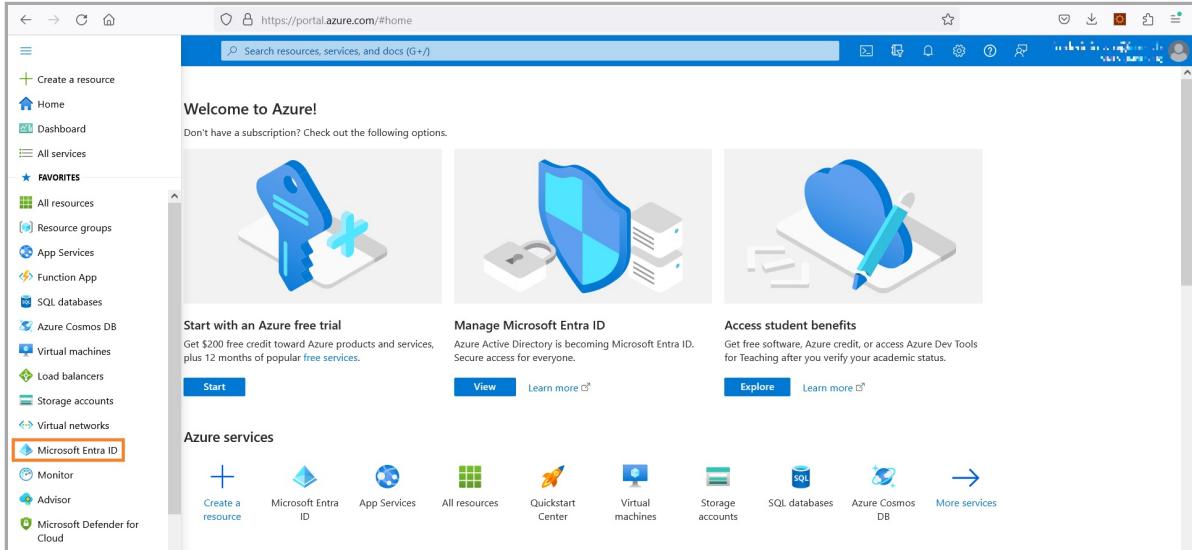
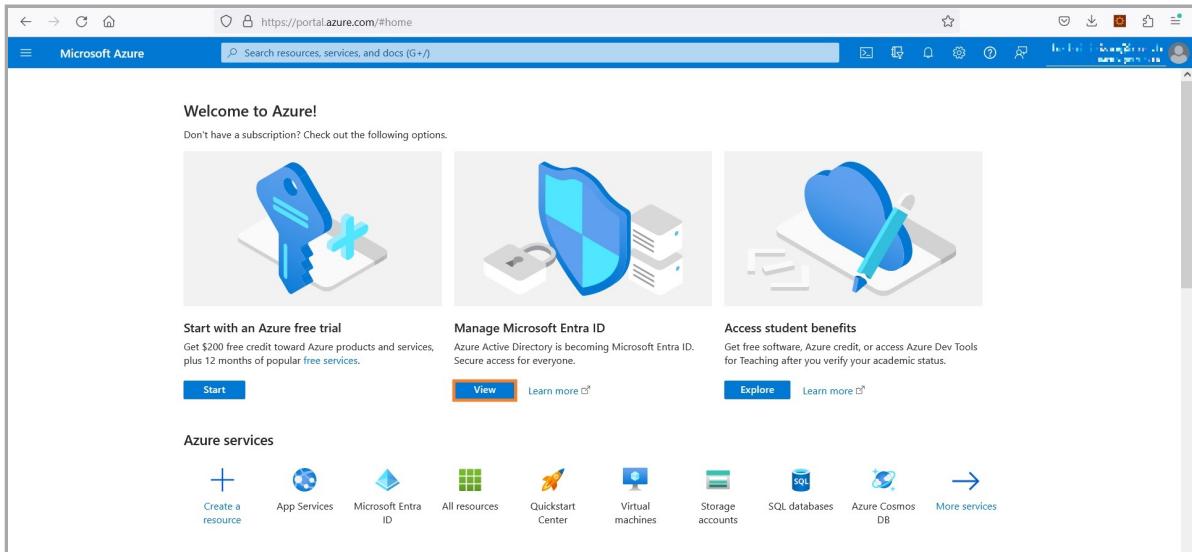
You can create your Azure Active Directory (or Azure AD) application by following this Microsoft tutorial <https://docs.microsoft.com/en-us/graph/auth-register-app-v2>.

A procedure example is shown here after by connecting to the Microsoft Azure portal.

This procedure allows to generate you own ID and SECRET required in SBL10e configuration pane. To support Microsoft 365 :

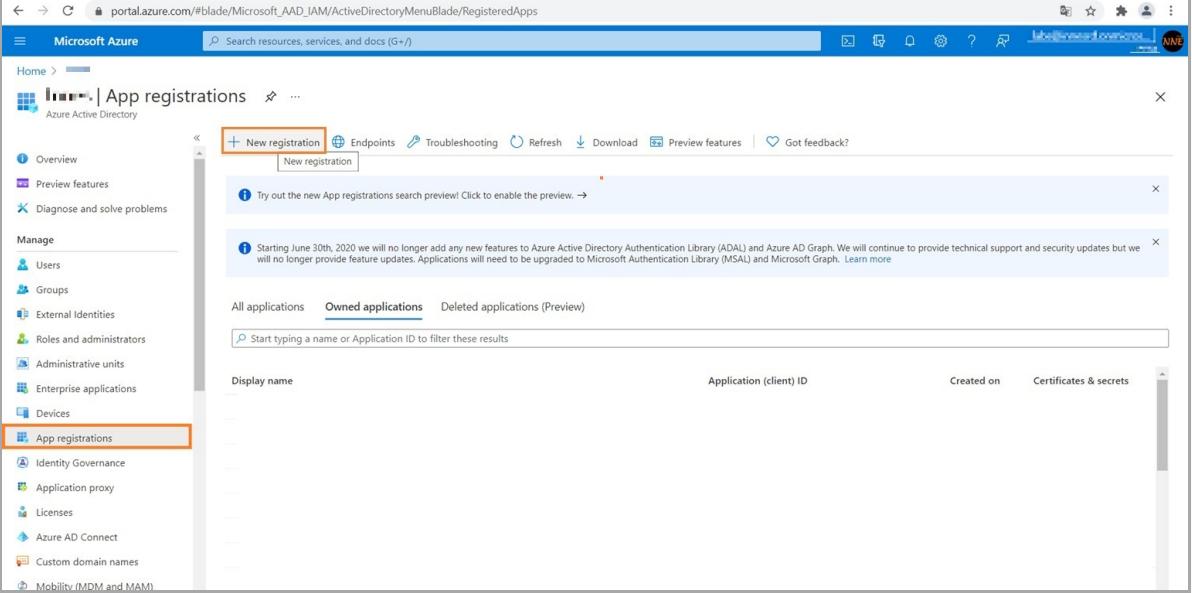
- Application (client) ID ,
- Directory (Tenant) ID ,
- Client secret .

Connect on Microsoft Azure portal: <https://portal.azure.com/> and sign in with your Microsoft 365 (M365) administrator account login credentials. Click on the left top menu and choose the Azure Active Directory item.



Application (client) ID and directory (Tenant) ID

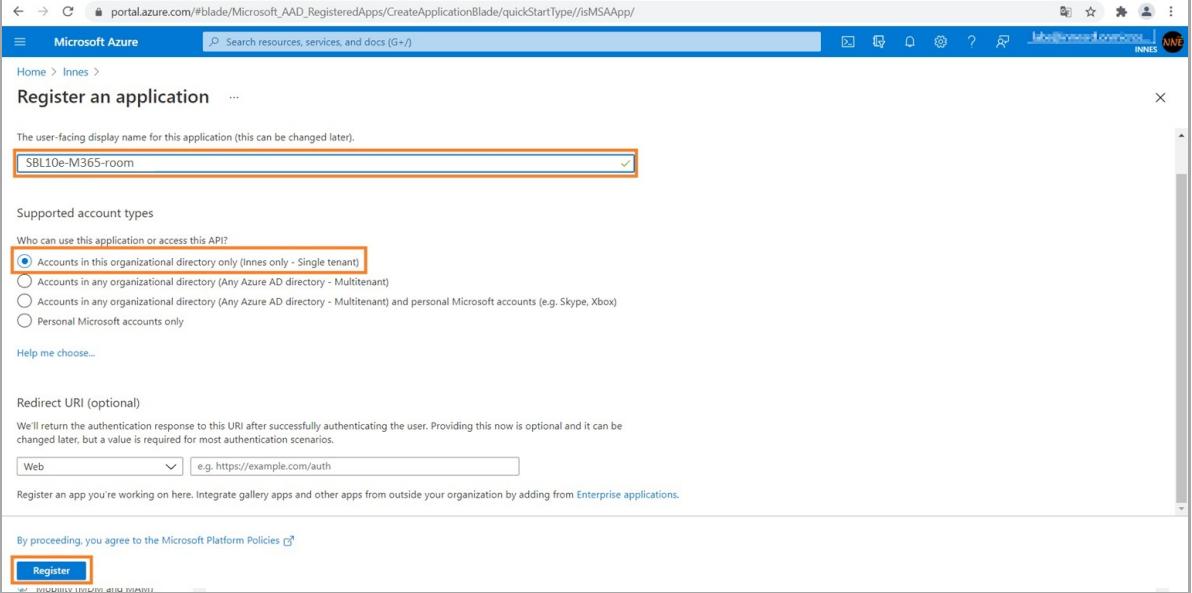
On the App registrations menu, click on *New registration*.



The screenshot shows the Microsoft Azure portal's 'App registrations' blade. On the left, a sidebar lists various Azure services: Overview, Preview features, Diagnose and solve problems, Manage (with options for Users, Groups, External Identities, Roles and administrators, Administrative units, Enterprise applications, and Devices), and App registrations (which is selected and highlighted with an orange box). The main area has a header with 'New registration' and 'Endpoints', 'Troubleshooting', 'Refresh', 'Download', 'Preview features', and 'Got feedback?' buttons. A message bar at the top says 'Try out the new App registrations search preview! Click to enable the preview.' Below it, another message bar states 'Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph.' The main content area shows tabs for 'All applications', 'Owned applications' (which is selected and highlighted with a blue underline), and 'Deleted applications (Preview)'. A search bar says 'Start typing a name or Application ID to filter these results'. The table below lists applications under 'Owned applications':

Display name	Application (client) ID	Created on	Certificates & secrets
SBL10e-M365-room			

Enter an application name (e.g.: *SBL10e-M365-room*), Select the appropriate Account in the organization directory only (organization only – Single tenant) radio button, and press on the **Register** button.



The screenshot shows the 'Register an application' blade. At the top, it says 'The user-facing display name for this application (this can be changed later.)' with a text input field containing 'SBL10e-M365-room' (which is highlighted with an orange box). Below this is a section for 'Supported account types' with a heading 'Who can use this application or access this API?'. There are four radio buttons: 'Accounts in this organizational directory only (Innes only - Single tenant)' (which is selected and highlighted with an orange box), 'Accounts in any organizational directory (Any Azure AD directory - Multitenant)', 'Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)', and 'Personal Microsoft accounts only'. Below these is a 'Help me choose...' link. The next section is 'Redirect URI (optional)' with a note: 'We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.' It includes a dropdown menu set to 'Web' and a text input field with 'e.g. https://example.com/auth'. Below this is a note: 'Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from Enterprise applications.' The final section at the bottom says 'By proceeding, you agree to the Microsoft Platform Policies' with a link, and a large blue 'Register' button (which is highlighted with an orange box).

In the **Overview** menu, copy to clipboard the **Application (client) ID** value, the 1st value required in SBL10e configuration tab and store it preciously.

The screenshot shows the Microsoft Azure portal interface. The left sidebar has a tree view with 'Overview' selected. The main content area is titled 'SBL10e-M365-room'. Under the 'Essentials' section, the 'Application (client) ID' field is highlighted with a red box. To its right is a 'Copy to clipboard' button. Other fields shown include 'Display name' (SignMeeting), 'Object ID', 'Directory (tenant) ID', 'Client credentials', 'Redirect URIs', 'Application ID URI', and 'Managed application in ...'. Below the essentials section, there are two informational cards: one about the new App registrations experience and another about the end of support for ADAL and Azure AD Graph.

In the Overview menu, copy to clipboard the Directory (tenant) ID value, the 2nd value required in SBL10e configuration tab and store it preciously.

This screenshot is identical to the one above, showing the 'Overview' tab for the same app. The 'Directory (tenant) ID' field is now highlighted with a red box, indicating it is the value to be copied. The rest of the interface is the same, including the informational cards at the bottom.

Client secret

In the Certificates & secrets menu, click on the New client secret button.

Enter a name (e.g.: my_client_secret) and press on the Add button.

Copy into clipboard the client secret value, the 3rd input for the SBL10e configuration tab and store it preciously.

⚠ Do it right now because the client secret value is not visible anymore as soon as you click on a new web page.

The screenshot shows the Microsoft Azure portal interface. The URL in the address bar is https://portal.azure.com/#blade/Microsoft_AAD_RegisteredApps/ApplicationMenuBlade/Credentials/quickStartType/sourceType/Microsoft_AAD_IAM/appId/269d8888-f581-43fe-b53dfea6c181b7f4/objectId/00000000-0000-0000-0000-000000000000/resourceId/00000000-0000-0000-0000-000000000000. The page title is "SBL10e-M365-room Certificates & secrets". The left sidebar shows navigation options like Overview, Quicksart, Integration assistant, Manage (Branding, Authentication, Certificates & secrets, Token configuration, API permissions, Expose an API, App roles, Owners, Roles and administrators, Manifest), Support + Troubleshooting (Troubleshooting, New support request), and a search bar. The main content area is titled "Certificates" and contains a note about using certificates for authentication. It shows a table for "Certificates" with columns: Thumbprint, Start date, Expires, and Certificate ID. A message states "No certificates have been added for this application." Below this is the "Client secrets" section, which defines a secret string for proving identity. It shows a table with columns: Description, Expires, and Value. A new client secret named "my_client_secret" is listed, with its value being "6-CdOVxv6p1wwH4y8Q6Yr11_SY7dU-T_t Ba3b5df4-fd5e-40b1-8127-3fce2607d75". Buttons for "Copy to clipboard" and "Get ID" are available for this entry.

Thumbprint	Start date	Expires	Certificate ID
No certificates have been added for this application.			

Description	Expires	Value	Copy to clipboard	Get ID
my_client_secret	6/24/2023	6-CdOVxv6p1wwH4y8Q6Yr11_SY7dU-T_t Ba3b5df4-fd5e-40b1-8127-3fce2607d75	<input type="button" value="Copy to clipboard"/>	<input type="button" value="Get ID"/>

Grant permissions

In the API permissions menu, press on the `Add a permission` button.

For `m365_room` application, these permissions must be granted:

- `Calendars.Read`,
- `User.Read.All`.

The screenshot shows the Microsoft Azure portal interface. The left sidebar has a 'Manage' section with 'API permissions' highlighted. The main content area shows the 'Configured permissions' table with one row for 'Microsoft Graph (1)' with permission 'User.Read'. A tooltip above the table explains the 'Admin consent required' column. Below the table, there's a note about trying 'Enterprise applications'.

Click on the `Microsoft graph` button.

The screenshot shows the 'Request API permissions' dialog. The 'Microsoft APIs' tab is selected. The 'Commonly used Microsoft APIs' section is expanded, showing various services like Microsoft Graph, Azure DevOps, and others. The 'Microsoft Graph' card is highlighted with an orange box.

Select then the `Application permissions` button.

The screenshot shows the Microsoft Azure portal interface. On the left, the navigation menu includes 'Overview', 'Quickstart', 'Integration assistant', 'Manage' (with 'Branding', 'Authentication', 'Certificates & secrets', 'Token configuration', 'API permissions' selected), 'Expose an API', 'App roles', 'Owners', 'Roles and administrators | Pre...', 'Manifest', 'Support + Troubleshooting', 'Troubleshooting', and 'New support request'. The main content area displays the 'SBL10e-M365-room API permissions' blade. A modal window titled 'Request API permissions' is open, showing the 'Microsoft Graph' section. It asks 'What type of permissions does your application require?' with options for 'Delegated permissions' (selected) and 'Application permissions'. The 'Application permissions' section notes: 'Your application runs as a background service or daemon without a signed-in user.' At the bottom of the modal are 'Add permissions' and 'Discard' buttons.

In the display filter input, enter the text **calendar** and check the option `Calendars.Read`.

Do not press now on the `Add permissions` button right now.

This screenshot shows the same Microsoft Azure portal and API permissions blade as the previous one. The 'Request API permissions' dialog is still open over the Microsoft Graph section. In the 'Select permissions' area, the search bar contains the text 'calendar'. Below it, under 'Permissions', there is a list item 'Calendars (1)' which is expanded. Inside this list, the permission 'Calendars.Read' is selected, indicated by a blue checkbox and the status 'Yes'. The 'Add permissions' and 'Discard' buttons are visible at the bottom of the dialog.

In the display filter input, enter the text **User** and scroll to the bottom to find the **User** entry.

The screenshot shows the Microsoft Azure portal's API permissions configuration page for the 'SBL10e-M365-room' app. On the left, the navigation menu includes 'Overview', 'Quickstart', 'Integration assistant', 'Manage' (with 'Branding', 'Authentication', 'Certificates & secrets', 'Token configuration', and 'API permissions' selected), 'Expose an API', 'App roles', 'Owners', 'Roles and administrators', 'Manifest', 'Support + Troubleshooting', and 'Troubleshooting'. The main area displays 'Configured permissions' for the 'Microsoft Graph' API, showing one permission: 'User.Read' (Delegated, Sign in and read). A modal window titled 'Request API permissions' is open, showing the 'Application permissions' section with 'User' selected. An orange arrow points down to the 'Add permissions' button at the bottom of the modal.

Select the option `User.ReadAll`.

The screenshot shows the same API permissions configuration page. The 'User' permission under 'Microsoft Graph' is now expanded, revealing several options: 'User.Export.All', 'User.Invite.All', 'User.ManageIdentities.All', 'User.Read.All' (which has a checked checkbox and a tooltip 'Read all users' full profiles'), and 'User.ReadWrite.All'. An orange arrow points to the checked 'User.Read.All' checkbox.

Click on the `Add permissions` button.

At this step, the permissions are not yet granted.

The screenshot shows the Microsoft Azure portal's API permissions page for the application 'SBL10e-M365-room'. On the left, there's a navigation sidebar with options like Overview, Quickstart, Integration assistant, Manage, Branding, Authentication, Certificates & secrets, Token configuration, API permissions (which is selected), Expose an API, App roles, Owners, Roles and administrators | Preview, and Manifest. The main content area has a search bar at the top. A message says 'You are editing permission(s) to your application; users will have to consent even if they've already done so previously.' Below this, a note about 'Configured permissions' is shown, followed by a table listing two permissions under 'Microsoft Graph (2)': 'Calendars.Read' (Application, Read calendars in all mailboxes, Yes, Not granted for Innes) and 'User.Read.All' (Application, Read all users' full profiles, Yes, Not granted for Innes). At the bottom, it says 'To view and manage permissions and user consent, try Enterprise applications.'

Click on the Grant admin consent for <your_organization> button.

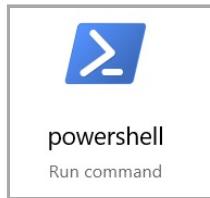
This screenshot shows a 'Grant admin consent confirmation' dialog box overlaid on the API permissions page. The dialog asks, 'Do you want to grant consent for the requested permissions for all accounts in Innes? This will update any existing admin consent records this application already has to match what is listed below.' There are 'Yes' and 'No' buttons. Behind the dialog, the API permissions table is visible, showing the same two permissions as the previous screenshot.

Now the permissions are granted.

This screenshot shows the API permissions page again, but now with a success message at the top: 'Successfully granted admin consent for the requested permissions.' The table shows the two permissions with their 'Status' column updated to 'Granted for Innes' (indicated by a green circle icon).

6.7 Appendix: Configuration using PowerShell for Microsoft 365 (M365)

On a MS-Windows computer, launch `powershell` with administrator rights.



■ *The command lines have been tested on PowerShell 7.X.*

Install ExchangeOnlineManagement module

To connect to your M365, you must install once, the `ExchangeOnlineManagement` module.

```
Install-Module -Name ExchangeOnlineManagement
```

Execute Powershell commands for Microsoft 365 (M365)

On a MS-Windows computer, open `powershell` command with administrator rights and type this command line by replacing the parameters values by yours (e.g. `my365AdminAccount@contoso.onmicrosoft.com`):

```
Connect-ExchangeOnline -UserPrincipalName my365AdminAccount@contoso.onmicrosoft.com
```

Room creation (if not already created)

To create a new resource mailbox, type this command line by replacing the parameters values by yours (e.g. `myresource1`):

```
New-Mailbox -Name "myresource1" -Room
```

Room configuration

To define the typical default configuration for this mailbox suitable for the SBL10e App, type on the same line, these three command lines parts by replacing the parameters values by yours (e.g. `myresource1`):

```
Set-CalendarProcessing -Identity "myresource1" -AutomateProcessing AutoAccept  
-AddOrganizerToSubject $false -DeleteSubject $false -ScheduleOnlyDuringWorkHours $true  
-DeleteComments $False -DeleteAttachments $False -RemovePrivateProperty $False
```

■ *When a room is deleted (or modified) by the App, it is deleted (or modified) only in room calendar (and kept unmodified in the delegate calendar).*

■ *DeleteAttachments = false to keep attachment in events.*

■ *RemovePrivateProperty = false to display private events as well.*

To check the room configuration, type this command line by replacing the parameters values by yours (e.g. `myresource1`):

```
Get-CalendarProcessing "myresource1" | fl
```

Working hours and time zone policy

To configure the working range policy and timezone, type on the same line, these four command line parts, by replacing the parameters values by yours (e.g. myresource1):

```
Set-MailboxCalendarConfiguration "myresource1"  
-WorkingHoursStartTime 08:00:00 -WorkingHoursEndTime 19:00:00  
-Workdays Weekdays -WeekStartDay Monday  
-WorkingHoursTimeZone "Central Europe Standard Time"
```

To check your room configuration, type this command line by replacing the parameter value by yours:

```
Get-MailboxCalendarConfiguration "myresource1"
```

Delegate account creation

To create a new user account, known also as delegate account, type on a same line, these four lines by replacing the parameters values by yours:

```
New-Mailbox -Alias MyDelegate -Name "mydelegate" -FirstName "My" -LastName "Delegate"  
-DisplayName "myDelegate" -MicrosoftOnlineServicesID "mydelegate@contoso.onmicrosoft.com"  
-Password (ConvertTo-SecureString -String "1234abcd" -AsPlainText -Force)  
-ResetPasswordOnNextLogon $false
```

Grant access to delegate account

To grant access to the resource mailbox for the delegate account, type on the same line these two command line parts, by replacing the parameters values by yours (e.g. mydelegate):

```
Add-MailboxPermission -Identity myresource@contoso.onmicrosoft.com -User "mydelegate"  
-AccessRights FullAccess -InheritanceType All -automapping $true
```

In this example, a delegate account `mydelegate@contoso.onmicrosoft.com` has been created using password `1234abcd`. The delegate account can handle the room `myresource1@contoso.onmicrosoft.com`. You can now use these parameters to configure the datasource and associated resources in Screen Composer for your App.

Note for PowerShell execution

SSL is requested by the `powershell` client. If the SSL error is raised, unencrypted traffic is disabled in the client configuration. A temporary solution is to disable SSL for this powershell session. In this case, type these three commands lines.

```
cd WSMAN:\localhost\Client
```

```
set-item .\allowunencrypted $true
```

```
set-item .\trustedhosts IPAddressofyourpowershellclientcomputer
```

6.8 Appendix: Devices configuration using Powershell

Your park of Qeedji devices can be configured and maintained using the `PSDevice` Powershell module.

Introduction

This set of *Powershell* functions allows to:

for AQS devices :

- retrieve general information of device, with the `Get-AqsInfos` , functions,
- to retrieve all installed APPs, with the `Get-AqsApps` function,
- install new App with the `Install-AqsApp` function,
- remove an App with the `Remove-AqsApp` function,
- restart new App with the `Restart-AqsApp` function,
- stop new App with the `Stop-AqsApp` function,
- install a new firmware with the `Install-AqsFirmware` function.
- install a configuration script with the `Install-AqsConfiguration` function.

for Bm0032 devices :

- retrieve general information of device, with the `Get-Bm0032Infos` , functions,
- install a new firmware with the `Install-Bm0032Firmware` function.
- install a configuration script with the `Install-Bm0032Configuration` function.

for Gekkota devices :

- retrieve general information of device, with the `Get-GtkInfos` , functions,
- install a new firmware with the `Install-GtkFirmware` function,
- install a configuration script with the `Install-GtkConfiguration` function.

These functions are defined in the `PSDevice` PowerShell module stored in the `Modules\PSDevice\` directory.

Security

By default, the execution of local *Powershell* scripts are not allowed. You can change their execution rights by changing the *PowerShell* security policy. This modification has to be done once with the `Set-ExecutionPolicy` *Powershell* function. Your organization may have to change it according to your security rules.

For example, to authorize the execution of all scripts, launch a *Powershell* console with administrator rights, and type:

```
PS > Set-ExecutionPolicy -ExecutionPolicy Unrestricted -scope CurrentUser
```

For further information, look at the cmdlet `Set-ExecutionPolicy` help page.

If you cannot allow the execution of unsigned local scripts, you can install the provided certificate in the list of authorized root certificates with the command:

```
PS > cd <your_path_to_the_scripts>\Powershell_Innes_Device\Certificate\  
PS > Import-PfxCertificate -FilePath InnesCodeSigningRootCA_1.pfx -CertStoreLocation  
cert:\CurrentUser\Root -Password $(ConvertTo-SecureString "1234" -AsPlainText -Force)
```

To import the `.pfx` certificate, you can also use the MS-Windows application `certmgr.msc`, select the `Trusted Root Certification Authorities`, right clic on `All Tasks`, select the `Import` item, select the file and enter the password `1234`. When ended, close the current Powershell console.

Usage

To use Device *Powershell* modules, you have 3 possibilities:

1. Either copy the directories under `Modules\` into a standard *Powershell* module installation directory, for example "C:\Program Files\WindowsPowerShell\Modules". Then launch a *Powershell* console
2. Redefine the search variable for *Powershell* modules (the `$Env:PSModulePath` *Powershell* variable) each time you will use these functions. In this case, launch a *Powershell* console, and type the line below, adapting it to your path. Each time you will launch a new *Powershell* console, you will have to enter it again

For example:

```
PS > $Env:PSModulePath="$Env:PSModulePath;<your_path_to_the_scripts>\Powershell_Innes_Device\Modules"
```

3. Or redefine the search variable for *Powershell* modules in the Windows environment variables. For that, add the path `<your_path_to_the_scripts>\Powershell_Innes_Device\Modules` to the environment variable `PSModulePath`. Then, launch afterwards a *Powershell* console.

To use the functions or get help, you must then import the module(s) with the `Import-Module` function. Example:

```
PS > Import-Module PSDevice
```

Depending on how you get the scripts, you may have this following warning:

```
Security Warning Run only scripts that you trust. While scripts from the Internet can be useful,
this script can potentially harm your computer. Do you want to run \device\scripts\my.ps1?
[D] Do not run [R] Run once [S] Suspend [?] Help (default is "D"):
```

To avoid this message, you can unblock the script files (to do only once):

```
PS > cd <your_path_to_the_scripts>\Powershell_Innes_Device\
PS > dir -Recurse | Unblock-File
```

The `Get-Command` function allows you to list the functions defined in a module. Example:

```
PS > Get-Command -Module PSDevice
```

Answer example:

CommandType	Name	Version	Source
-----	----	-----	-----
Function	Disable-AqsApp	1.10.10	PSDevice
Function	Enable-AqsApp	1.10.10	PSDevice
Function	Get-AqsApps	1.10.10	PSDevice
Function	Get-AqsInfos	1.10.10	PSDevice
Function	Get-Bm0032Infos	1.10.10	PSDevice
Function	Get-GktInfos	1.10.10	PSDevice
Function	Install-AqsApp	1.10.10	PSDevice
Function	Install-AqsConfiguration	1.10.10	PSDevice
Function	Install-AqsFirmware	1.10.10	PSDevice
Function	Install-Bm0032Configuration	1.10.10	PSDevice
Function	Install-Bm0032Firmware	1.10.10	PSDevice
Function	Install-GktConfiguration	1.10.10	PSDevice
Function	Install-GktFirmware	1.10.10	PSDevice
Function	LogWrite	1.10.10	PSDevice
Function	Remove-AqsApp	1.10.10	PSDevice
Function	Restart-AqsApp	1.10.10	PSDevice
Function	Stop-AqsApp	1.10.10	PSDevice
Function	Test-AqsDevice	1.10.10	PSDevice
Function	Test-Bm0032Device	1.10.10	PSDevice
Function	Test-GktDevice	1.10.10	PSDevice

You can get help on each function of the module by using the standard cmdlet `Get-Help` with options:

- `-detailed`,
- `-full`,
- `-examples`.

Example:

```
PS > Get-Help -detailed Install-AqsApp
```

Examples

To use the examples, copy the directories `Examples\` into a standard *Powershell* module installation directory, for example "C:\Program Files\WindowsPowerShell\Modules".

In the directory `Examples`, you can find different powershell scripts which uses the functions of the modules.

You can get help on each example scripts, for example:

```
PS > Get-Help -detailed .\Examples\Example1\Get-DevicesInfos.ps1
```

Example 1: Get-DevicesInfos

The script `Examples\Example1\Get-DevicesInfos` is an example to retrieve informations about a pool of devices described in a json file. It uses the module `PSDevice`.

Example:

```
PS > cd <your_path_to_the_scripts>\Examples\Example1\  
PS > .\Get-PSDevice.ps1 -LogFile result.json
```

If any error occurs, look at the logfile (`result.json` in the example) to see what the problem may be.

This is an example of report:

```
[  
  {  
    "host": "192.168.0.74",  
    "info": {  
      "Psn": "01540-00657",  
      "Platform": "AMP300",  
      "Version": "9.10.19",  
      "Hostname": "AMP300-floor1",  
      "runningApps": [  
        {  
          "Label": "PowerPoint Publisher",  
          "Version": "1.15.10"  
        },  
        {  
          "Label": "PowerPoint Publisher UI",  
          "Version": "1.15.10"  
        }  
      ]  
    }  
  },  
  {  
    "host": "192.168.0.92",  
    "info": {  
      "Psn": "01320-00039",  
      "Hostname": "sbl10-floor2",  
      "Firmware": "bm0032_m365_user-sbl10e-1.12.10",  
      "Platform": "SBL10e"  
    }  
  },  
  {  
    "host": "192.168.0.91",  
    "info": {  
      "Psn": "00910-00216",  
      "Hostname": "DMB400-hall1",  
      "FirmwareVersion": "5.11.13",  
      "Platform": "dmb400"  
    }  
  },  
  {  
    "host": "192.168.10.91",  
    "error": "Not responding"  
  }  
]
```

Example 2.1: upgrade firmware

The script `Examples\Example1\Install-Devices` is an example to install firmware, an app, or a configuration script on a pool of devices described in a json file. The type of component to install is specified with the `-installType` option which can be "install", "app" or "configuration". The components to be installed are stored in the directory specified by the `"installDirPath"` option. Each type of device has an associated subdirectory:

- `aqs` for Aqs device,
- `gekkota` for Gekkota device,
- `bm0032` for Bm0032 device.

It uses the module `PSDevice`.

Open the `Examples\Example2\devices.json` and update with the values corresponding to the devices concerned:

- IPV4 adress,
- WebDAV server login,
- WebDAV server password.

In case you wish to upgrade the firmware of your park of AQS devices, do copy the appropriate firmware `.fqs` into the following directory:

- `Examples\Example2\install\aqsl`

In case you wish to upgrade the firmware of your park of Gekkota devices, do copy the appropriate firmware `.frm` into the following directory:

- `Examples\Example2\install\gekkota\`

In case you wish to upgrade the firmware of your park of Bm0032 devices, do copy the appropriate firmware `.bin` into the following directory:

- `Examples\Example2\install\bm0032\`

Command line example:

```
PS > cd <your_path_to_the_scripts>\Examples\Example2\  
PS > .\Install-Devices.ps1
```

If any error occurs, look at the logfile (`result.json` in the example) to see what the problem may be.

Example 2.2: change device configuration

In case you wish to change the configuration of your park of AQS devices, do copy the appropriate `<MAC>.js` configuration script into the following directory:

- `Examples\Example2\install\aqsl`

In case you wish to change the configuration of your park of Gekkota devices, do copy the appropriate `<MAC>.js` configuration script into the following directory:

- `Examples\Example2\install\gekkota\`

In case you wish to change the configuration of your park of Bm0032 devices, do copy the appropriate `<MAC>.js` configuration script into the following directory:

- `Examples\Example2\install\bm0032\`

Copy the `Install-Devices.ps1` file, paste it in the same directory and rename it `Configure-Devices.ps1`. Replace in it the part `[string] $installType = "install"`, by the part `[string] $installType = "configuration"`,

Open the `Examples\Example2\devices.json` and update with the values corresponding to the devices concerned:

- IPV4 adress,
- WebDAV server login,
- WebDAV server password.

Example 2.3: install APK

In case you wish to install some APK in your park of AQS devices, do copy the appropriate `.apk` into the following directory:

- `Examples\Example2\install\aqsl`

Copy the `Install-Devices.ps1` file, paste it in the same directory and rename it `InstallApk-Devices.ps1`. Replace in it the part `[string] $installType = "install"`, by the part `[string] $installType = "app"`,

Open the `Examples\Example2\devices.json` and update with the values corresponding to the devices concerned:

- IPV4 adress,
- WebDAV server login,
- WebDAV server password.