



Barco Overture

User Manual

Manual for Overture UX 3.8.0 and Overture RMM 4.8.0

1 EULA

Overture

Product Specific End User License Agreement

THIS PRODUCT SPECIFIC USER LICENSE AGREEMENT (EULA) TOGETHER WITH THE BARCO GENERAL EULA ATTACHED HERETO SET OUT THE TERMS OF USE OF THE SOFTWARE.

PLEASE READ THIS DOCUMENT CAREFULLY BEFORE OPENING OR DOWNLOADING AND USING THE SOFTWARE.

DO NOT ACCEPT THE LICENSE, AND DO NOT INSTALL, DOWNLOAD, ACCESS, OR OTHERWISE COPY OR USE ALL OR ANY PORTION OF THE SOFTWARE UNLESS YOU CAN AGREE WITH ITS TERMS AS SET OUT IN THIS LICENSE AGREEMENT.

1.1 Metrics

Applicable prices for Barco Overture (the "Software") shall be invoiced and paid as per the applicable purchase order acknowledged by Barco.

1.1.0.1 a) Term

Barco Overture is available in 2 models offered "on-premises" (installed and run on computers on your premises):

- Perpetual license;
- Subscription license (yearly basis);

At the end of this time period, all rights associated with the use of the Software (including any associated updates or upgrades) cease.

1.1.0.1 b) Deployment

A single license is either

- restricted to the agreed number of Rooms, where a "Room" is defined as a collection of devices grouped in 1 virtual container, with a maximum of 40 devices per Room;
- restricted to the agreed number of control servers;
- or a combination of (i) and (ii).

You shall ensure that the Software connects through the internet with Barco and you acknowledge that the Software will send through such connection, at regular intervals a history of thumbprint of the rooms and devices actually connected, which may be used by Barco for license management and compliance.

1.1.0.1 c) Use

The Overture solution consists of (i) a UX server that may only be used on one (1) Machine and (ii) a Control server that may be used on multiple Machines.

A "Machine" means the smallest data processing unit on which it is possible to run the Software within a single computer device.

Any Device Drivers, GUI templates and CSS styling included in the Software may be modified by you as you deem required for your Intended Use of the Software.

1.2 Support

1.2.0.1 a) Perpetual Based

Maintenance, including the provision of upgrades and updates to the Software, and helpdesk support are available at your option on the terms of Barco's then current maintenance contract. Software maintenance may be included in the initial transaction if ordered and paid for additionally.

Updates and upgrades are not available otherwise. It is strongly suggested to maintain the software maintenance contract without interruption. Barco reserves the right not to restart maintenance following an interruption by the customer.

1.2.0.1 b) Subscription based

1.2.0.1 The provision of updates, upgrades and helpdesk support are included in the subscription license for the Term thereof.

1.3 Terms of Use

The Software can be used as set out in the Barco EULA attached hereto.

Any reference therein to the DRM is relevant only if and when the Software is licensed otherwise than on an "on premises" basis.

The provisions of this Product Specific EULA override the Barco generic EULA in case of conflicts or inconsistencies.

In case of (inadvertent or other) non-compliance (e.g. where the actual use overshoots the use authorized hereunder), Barco shall have the option to (i) cause you to procure such additional licenses required as per the actual usage and (ii) to suspend access to the Software until the non-compliance is remedied, failing of which Barco may terminate the License Agreement as set out herein.

1.4 Privacy

You are the controller (as defined under applicable privacy laws) for personal data which are being processed via the Software. Therefore, you remain solely responsible for complying with all applicable data protection laws and for implementing and maintaining privacy protection and security measures (especially for components that you provide or control). Barco disclaims any liability in this regard.

1.5 Other Terms

1.5.0.1 a) Other License Terms

The Overture solution contains software components released under an Open Source license.

A list of the third party software components used (open source and other) is available in the Software's README files, through the "My Barco" section of the Barco website or through other (online) means. The applicable license terms, copyright notices and, as relevant, source code access apply as set out in the Barco EULA attached hereto.

1.5.0.1 b) Safe and proper use of Devices

You remain solely responsible to operate any device which is controlled, monitored or analysed by the Software in accordance with the operating instructions and safety guidelines as intended and/or recommended by the relevant device manufacturers for a regular and routine usage of any such devices. Barco disclaims any liability resulting from an improper use of a device, even if such device is being controlled, monitored or analysed by the Software.

BARCO END USER LICENSE AGREEMENT



By accepting these terms (through a tick box or other mechanism designed to acknowledge agreement to the terms of an electronic copy of this License Agreement), or by installing, downloading, accessing, or otherwise copying or using all or any portion of the Software (as defined below), (i) you accept this License Agreement on behalf of the entity for which you are authorized to act (e.g., your employer) and you agree to act in a manner consistent with this License Agreement (or, if there is no such entity for which you are authorized to act, you accept this License Agreement on behalf of yourself as an individual and acknowledge that you are legally bound by this Agreement), and (ii) you represent and warrant that you have the right, power and authority to act on behalf of and bind such entity.

These terms apply to your use of the Software as of and for the original Term of your license. When you renew or purchase an additional license, or when Barco introduces elements that were not previously included in the Software, the then current license terms shall apply and will remain unchanged during the term of that license and/or in respect of such changed elements. The Product Specific EULA you accept applies in addition to these terms, and may evolve accordingly. You acknowledge that an electronic copy of this Agreement shall have the same proving value as a hard copy signed by the parties.

If you are unwilling to accept this License Agreement on these terms, or you do not have the right, power and authority to act on behalf of and bind such entity (or yourself as an individual if there is no such entity), DO NOT SELECT THE "I ACCEPT" BUTTON OR OTHERWISE CLICK ON ANY BUTTON OR OTHER MECHANISM DESIGNED TO ACKNOWLEDGE AGREEMENT, AND DO NOT INSTALL, DOWNLOAD, ACCESS, OR OTHERWISE COPY OR USE ALL OR ANY PORTION OF THE SOFTWARE.

- Definitions

"Affiliate" means any corporation or other entity directly or indirectly, controlling, controlled by or under common control with such corporation or entity.

For the purpose of the above, "control" shall mean (i) the ownership or control, directly or indirectly, of fifty percent (50%) or more of the equity capital or the shares or voting rights in the corporation or other entity in question or (ii) the control of the composition of the board of directors of the corporation or other entity in question.

"Barco" means Barco NV (company number 0473.191.041) with company address at Beneluxpark 21, 8500 Kortrijk, Belgium, or its designated Affiliate licensing to you the proprietary software which is the subject matter of this Agreement.

"Documentation" means all technical, reference and installation manuals, user guides, published performance specifications and other written documentation provided by Barco generally to its licensees with respect to the Software, along with any modifications and updates thereto;

"DRM" means Barco's digital rights management platform used to provide access to and access conditions of the Software.

"License Agreement" means this Barco End User License Agreement (EULA), incorporating the terms of the Product Specific EULA, and any modifications thereof as set out herein.

"Product Specific EULA" means the supplemental software terms applicable

"Software" means the Barco proprietary software which is being licensed hereunder, released in object code only.

"Term" means the period set out in article 9.1 hereof.

"you" means the entity on behalf of which these terms are accepted, and any of its representatives having access to the Software.

- License Grant

2.1 License Scope. Subject to compliance with all license terms and payment of applicable fees, Barco grants you a limited, non-exclusive, non-assignable, non-transferable user license (without the right to grant sublicenses). Save for the Product Specific EULA or any broader license terms confirmed through the DRM tool, (i) the license under this License Agreement applies to one (1) copy of the Software to be used on one single computing device and (ii) installation on a computing device that may be concurrently accessed by more than one user shall not constitute a permitted use and a separate license is required for each active user connected to a computing device on which the Software is being used

2.2 License Type. The applicable license type, and your rights in time, deployment and usage, are further detailed in the Product Specific EULA (in the absence of which the scope shall be as set in article 2.1 hereof).

2.3 License restrictions.

Intended Use. You agree to use the Software solely as permitted by this License Agreement (and any Product Specific EULA made part of it) and in a manner consistent with its design and Documentation.

No Transfer (License Agreement). You agree not to transfer, assign or sublicense your license rights to any other person or entity, unless Barco's prior written consent is obtained (which consent shall be reasonably given, but may come with a fee).



No Transfer (Software). If you deactivate or uninstall the Software from the computer device on which it was originally installed, this will terminate this License Agreement unless otherwise and specifically approved by Barco. You agree not to use the Software in association with other hardware or software that allows to pool connections, reroute information, reduce the number of devices or users that directly access or use the Software, or reduce the number of devices or users the Software directly manages (sometimes referred to as “multiplexing” or “pooling”) or otherwise attempt to reduce the number of licenses of any type that you need.

Authorized Users. The use of the Software is restricted to persons within your organization, or any third party representatives operating under your responsibility and control, provided any such persons have accepted the terms of this License Agreement. You agree not to use or permit the Software to be used to perform services for third parties, whether on a service bureau or time sharing basis or otherwise, without the prior written authorization of Barco. You shall not lease, rent, or otherwise transfer or grant a security or other interest in the Software.

No Modifications. You shall not make error corrections to or otherwise modify or adapt the Software or create derivative works based upon the Software, or permit third parties to do the same.

No Reverse Engineering. You agree not to reverse engineer or decompile, decrypt, disassemble or otherwise reduce the Software to human-readable form, except to the extent otherwise expressly permitted under applicable law notwithstanding this restriction, or except to the extent Barco is legally required to permit such specific activity pursuant to any applicable open source license.

Code required to ensure interoperability. To the extent required by law, and at your written request, Barco shall provide you with the interface information needed to achieve interoperability between the Software and another independently created program used by you, on payment of Barco’s applicable fee (if any). You shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with terms and conditions which Barco makes applicable.

No Unbundling. The Software may include various applications and components, may support multiple platforms and languages, and may be provided on multiple media or in multiple copies. Nonetheless, the Software is designed and provided to you as a single product to be used as a single product on devices as permitted herein. You agree not to unbundle the component parts of the Software for use on different computer devices.

Territory. You agree to use the Software solely in the territory or region where you obtained the Software from Barco or its authorized reseller or as otherwise stated in the Documentation. Any export if permitted shall comply with any applicable (export) laws and regulations.

2.4 Your Infrastructure. You remain responsible to procure and maintain hardware, operating system, network and other infrastructure (the “Infrastructure”) required to operate the Software and to keep such Infrastructure functioning and virus-free. You acknowledge that the Software is a complex computer software application, and that the performance thereof may vary depending hardware platform, software interactions and configuration. You acknowledge that the Software is not designed and produced specifically to meet your specific requirements and expectations and the selection of the Software by you is entirely your own choice and decision.

- Intellectual Property Rights.

3.1 Ownership. Any Software is licensed, not sold to you, on a non-exclusive basis for use only under the terms of this License Agreement, and Barco and its suppliers reserve all rights not expressly granted to you. You may own the carrier on which the Software is provided, but the Software is owned and copyrighted by Barco or by third party suppliers. Your license confers no title or ownership and is not a sale of any rights in the Software or its Documentation.

3.2 Third Party Materials. The Software may contain or require the use of certain third party technology (whether proprietary or open source software), identified by Barco in the Documentation, readme file, third-party click-accept, on www.barco.com or elsewhere (the “Identified Components”). Identified Components may be subject to additional and/ or different terms and you agree that the Identified Components are licensed under the terms, disclaimers and warranties of their respective licenses which in the forthcoming case shall override the provisions of this License Agreement.

3.3 Source Code Access. To the extent required under third party (open source) license terms, and for a period of 36 months following your acceptance of this License Agreement, Barco shall provide access to the source code controlled by a third party (open source) license, via email or download link. If the relevant license terms require so, you may require Barco (attn. its legal department, at the address stated above) to obtain such code on tangible medium against payment of the cost of media, shipping and handling.

3.4 Copyright. The Software is protected by national and international laws and treaty provisions. Copyright on the Software components belongs to the respective initial copyright holder, each additional contributor and/or their respective assignee(s), as may be identified in the Software Documentation, source code, README file, or otherwise. You shall not remove or obscure or otherwise alter the respective copyrights.

3.5 Trademarks. Brand and product names mentioned in relation to the Software may be trademarks, registered trademarks or copyrights of their respective (third party) holders. All such brand and product names mentioned in relation to the Software serve as comments or examples and are not to be understood as advertising for the products or their manufacturers.



3.6 Trade Secrets. You agree not to disclose, provide or otherwise make available trade secrets contained within the Software and Documentation in any form to any third party without the prior written consent of Barco. You shall implement reasonable security measures to protect such trade secrets.

- Support

4.1 Principle. Barco is under no obligation to provide support in respect of the Software, except as included in a Product Specific EULA and/or the extent you have entered into a separate maintenance agreement. Any unauthorized use of the Software may prohibit Barco from providing such support.

4.2 Support policy. Barco may provide to you maintenance releases to address bugs or security issues in the Software and you agree to install the same. Any other updates or upgrades can be obtained under the terms of a separate software maintenance which is being offered to you. You may have a right to downgrade your licensed Software application to (only) such earlier version of the same Software application as agreed by Barco in the forthcoming case.

Additional functionality may be licensed to you with and subject to additional or different terms.

- Warranty

EXCEPT FOR THE LIMITED WARRANTY THAT MAY APPLY AS PER THE PRODUCT SPECIFIC EULA, YOU UNDERSTAND THAT THE SOFTWARE IS BEING PROVIDED TO YOU "AS IS". BARCO DOES NOT MAKE NOR INTENDS TO MAKE ANY WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED AND SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY AND DOES NOT WARRANT THAT THE SOFTWARE WILL BE FREE FROM ERRORS OR THAT YOU WILL BE ABLE TO OPERATE THE SOFTWARE WITHOUT INTERRUPTIONS OR THAT SUCH ERRORS WILL BE CORRECTED BY BARCO. EXCEPT FOR ANY MAINTENANCE AND SUPPORT OBLIGATIONS SEPARATELY AGREED, YOU ARE SOLELY RESPONSIBLE FOR ALL COSTS AND EXPENSES ASSOCIATED WITH RECTIFICATION, REPAIR OR DAMAGE CAUSED BY SUCH ERRORS. in the forthcoming case, THE WARRANTY DISCLAIMER FOUND IN APPLICABLE OPEN SOURCE LICENSES shall override the provisions of this License Agreement.

- Compliance and Enforcement

6.1 Reporting and Audit. In addition to good practice record-keeping obligations, you agree to report the use of the Software and relating billing metrics in the DRM or otherwise as agreed. You grant to Barco and its designated auditors, at Barco's expenses, the right to verify your Software deployments and to examine your books, records and accounts during your normal business hours so as to verify your compliance with the License Agreement. In the event such audit discloses non-compliance with your payment obligations hereunder, you shall promptly pay to Barco the appropriate license fees plus the reasonable cost of conducting the audit.

6.2 Enforcement. Barco shall notify the then known user through the DRM (failing of which, otherwise in writing) of a substantial non-compliance, based on the triggers as per the Product Specific EULA. The non-compliance may result in an immediate or graduate denial of service (i.e. termination of the rights granted under the License Agreement), in part or in full, all based on the level of severity of the non-compliance [as per the Product Specific EULA].

6.3 Indemnification. YOU HEREBY AGREE TO INDEMNIFY, DEFEND AND HOLD HARMLESS BARCO AND BARCO'S AFFILIATES FROM AND AGAINST ANY AND ALL ACTIONS, PROCEEDINGS, LIABILITY, LOSS, DAMAGES, FEES AND COSTS (INCLUDING ATTORNEY FEES), AND OTHER EXPENSES INCURRED OR SUFFERED BY BARCO ARISING OUT OF OR IN CONNECTION WITH ANY BREACH BY YOU OF THE TERMS OF THIS SOFTWARE LICENSE.

- Limitation of Liability

TO THE MAXIMUM EXTENT PERMITTED BY LAW, BARCO ACCEPTS NO LIABILITY FOR ANY DAMAGES, LOSSES OR CLAIMS YOU OR ANY THIRD PARTY MAY SUFFER AS A RESULT OF YOUR USE OF THE SOFTWARE. IN JURISDICTIONS WHERE BARCO'S LIABILITY CANNOT BE EXCLUDED, BARCO'S LIABILITY FOR DIRECT DAMAGES SHALL BE LIMITED TO AN AMOUNT OF 250 EURO IN THE AGREGATE (OR TO THE MAXIMUM EXTENT PERMITTED BY LAW WHERE NO FURTHER EXCLUSION IS LEGALLY ALLOWED).

TO THE MAXIMUM EXTENT PERMITTED BY LAW, IN NO EVENT WILL BARCO BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS OR DAMAGES OF ANY KIND WHICH MAY ARISE OUT OF OR IN CONNECTION WITH THE SOFTWARE, THIS SOFTWARE LICENSE OR THE PERFORMANCE OR PURPORTED PERFORMANCE OF OR FAILURE IN THE PERFORMANCE OF BARCO'S OBLIGATIONS UNDER THIS SOFTWARE LICENSE OR FOR ANY ECONOMIC LOSS, LOSS OF BUSINESS, CONTRACTS, DATA, GOODWILL, PROFITS, TURNOVER, REVENUE, REPUTATION OR ANY LOSS ARISING FROM WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION OF THE SOFTWARE AND ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES WHICH MAY ARISE IN RESPECT OF USE OF THE SOFTWARE, EVEN IF BARCO HAS BEEN ADVISED OF THE POSSIBILITY OF THEIR OCCURRENCE.

- Confidentiality

P 6 / 270



8.1 Confidential Information. You will be receiving information which is proprietary and confidential to Barco during the negotiation and Term of this License Agreement. "Confidential Information" shall include (i) the underlying logic, source code and concepts of the Software or other trade secrets (the access to which is strictly limited as expressly set out herein), (ii) any information designated as confidential by Barco or which has the necessary quality of confidence about it and (iii) any license key provided by Barco to you hereunder.

8.2 Non-Disclosure. You agree not to divulge any Confidential Information to any persons without Barco's prior written consent provided that this article 8 shall not extend to information which was rightfully in your possession prior to the commencement of the negotiations leading to this License Agreement, which is already public knowledge or becomes so at a future date (otherwise than as a result of a breach of this article 8), to the extent it is required to be disclosed by law or which is trivial or obvious. You agree not to use any Confidential Information except for the authorized purpose hereunder. The foregoing obligations as to confidentiality shall survive the Term of this License Agreement.

- Term and Termination

9.1 Term. The duration of this License Agreement will be from the date of your acceptance (as set forth above) of the Software (whereby you acknowledge that use of the Software implies acceptance), until you de-activate the Software, discontinue the use of the device on which the Software was first installed for its intended use or the expiration of the limited time period set out in the Product Specific EULA, whichever comes first.

9.2 Termination. You may terminate this License Agreement at any time by destroying all copies of the Software then in your possession and returning all Documentation and associated materials, to Barco or the appointed Barco reseller that sold or provided these to you. Barco may terminate this License Agreement, immediately or gradually in accordance with article 6 hereof, by informing you at any time if any user is in breach of any of the License Agreement's terms.

9.3 Consequences of Termination. All rights associated with the use of the Software and the acquisition of updates and upgrades cease once the contract is terminated or expires. Cancelling your license will stop recurring fees going forward, but will not retroactively refund current or past payments.

- Other relevant terms

10.1 Personal Data. Whether or not Barco assumes the role of processor of personal data (as stated in the Product Specific EULA), you remain solely responsible for complying with all applicable data protection laws and for implementing and maintaining privacy protection and security measures (especially for components that you provide or control). Barco disclaims any liability for any data not provided by Barco, or any use of the Software outside the intended use as per this License Agreement or an applicable data processing annex.

10.2 Functional Information. Via the Software, Barco may gather technical information about (i) the functionality of the products which are connected through the Software, and/or (ii) as provided by you ("Functional Information"). Barco may make use of such Functional Information for purposes of analytics, providing services to your organization, allowing third party to access to such Functional Information and/or to provide services to your organization based on the legitimate interest of Barco.

- Final Clauses

11.1 Entire Agreement. This License Agreement is the only understanding and agreement between you and Barco for use of the Software. This License Agreement supersedes all other communications, understandings or agreements we had prior to this License Agreement (with the exception of any continuing confidentiality agreement).

11.2 Notices. Notices can be validly delivered to the parties' last known address.

11.3 Severability. This License Agreement shall not be altered, amended or varied. If any provision of this License Agreement is determined to be illegal, void or unenforceable, or if any court of competent jurisdiction in any final decision so determines, this License Agreement shall continue in full force save that such provision shall be deemed to be deleted with effect from the date of such decision, or such earlier date, and shall be replaced by a provision which is acceptable by law and which embodies the intention of this License Agreement as close as possible.

11.4 Export. You acknowledge that this Software may be subject to U.S. or other governments Export Jurisdiction. You agree to comply with all applicable international and national laws that apply to the Software, including the U.S. Export Administration Regulations, as well as end-user, end-use, and destination restrictions issued by the U.S. or other governments.

11.5 Survival. The provisions of articles 3, 5, 6, 7 and 8 will survive the termination of this License Agreement, howsoever caused, but this will not imply or create any continued right to use the Software after termination of this License Agreement.

11.6 Assignment. Barco shall be entitled to sub-contract all or any of Barco's obligations hereunder to a third party and/or any of Barco's Affiliates.

11.7 Law and Jurisdiction. The construction, validity and performance of this License Agreement shall be governed in all respects by the laws of Belgium, without recourse to its conflict of law principles. All disputes arising in any way out of or



affecting this License Agreement shall be subject to the exclusive jurisdiction of the courts of Kortrijk, without prejudice to enforcement of any judgment or order thereof in any other jurisdiction. The United Nations Convention on Contracts for the International Sale of Goods (the "Convention") shall not apply to this License Agreement, however, if the Convention is deemed by a court of competent jurisdiction to apply to this License Agreement, Barco shall not be liable for any claimed non-conformance of the Software under Article 35(2) of the Convention.

YOU HEREBY ACKNOWLEDGE TO HAVE READ, UNDERSTOOD AND ACCEPTED TO BE BOUND BY ALL THE TERMS AND CONDITIONS OF THIS LICENCE AGREEMENT AS INDICATED ABOVE



Table of Contents

1. [EULA](#)
2. [Overview](#)
 - 2.1. [Control Server](#)
 - 2.2. [User Experience\(UX\) Server](#)
 - 2.2.1. [Login to Overture](#)
 - 2.2.2. [Common Header](#)
 - 2.2.3. [Home](#)
 - 2.2.4. [Magic Menu](#)
 - 2.3. [RMM Monitor](#)
 - 2.3.1. [RMM Monitor Features](#)
 - 2.4. [RMM Insights](#)
 - 2.4.1. [RMM Insights Features](#)
 - 2.5. [Overall Overture functional architecture \(RMM, UX and CX\)](#)
3. [System Architecture](#)
 - 3.1. [Protocols](#)
 - 3.2. [Single Control Server](#)
 - 3.3. [Multiple Control Servers](#)
 - 3.4. [RMM Stackable Components](#)
 - 3.4.1. [RMM Implementation Cloud - On Premise](#)
 - 3.4.2. [RMM Implementation Cloud - Cloud](#)
4. [Setup](#)
 - 4.1. [Virtualized Install](#)
 - 4.1.1. [Installing Guide](#)
 - 4.1.2. [Requirements](#)
 - 4.1.3. [Attention:](#)
 - 4.1.4. [Oracle Virtual Box Installation](#)
 - 4.1.5. [VMware Player Installation](#)
 - 4.1.6. [VMware or Virtual Box Tools](#)
 - 4.1.7. [VMware vSphere Installation](#)
 - 4.1.8. [Microsoft Hyper-V Installation](#)
 - 4.1.9. [VM Advanced Settings](#)
 - 4.1.10. [VM Migration](#)
 - 4.1.11. [Starting The Machine](#)
 - 4.2. [Installing Separate Control Servers](#)
 - 4.2.1. [Proxy](#)
 - 4.2.2. [Using Windows Installer](#)
 - 4.2.3. [Using Docker](#)
 - 4.3. [Next Steps](#)
5. [Configurator](#)
 - 5.1. [System Configuration](#)
 - 5.1.1. [Licensing](#)
 - 5.1.2. [Password Policy](#)
 - 5.1.3. [Control Server Links](#)
 - 5.2. [Points](#)
 - 5.2.1. [Parent-Child-Sibling Relationships](#)
 - 5.2.2. [Point Attributes](#)
 - 5.2.3. [Map Fields](#)
 - 5.2.4. [Devices](#)
 - 5.2.5. [Containers](#)
 - 5.2.6. [Behaviors](#)
 - 5.2.7. [Built-in behaviors components](#)
 - 5.2.8. [Enum Labels](#)
 - 5.2.9. [Bulk duplicate](#)
 - 5.2.10. [Locations](#)



- 5.2.11. [Tasks / User Variables](#)
- 5.3. [Control Servers View](#)
 - 5.3.1. [Configuring CS2 Servers](#)
 - 5.3.2. [Configuring CS1 Servers](#)
- 5.4. [Plugins View](#)
- 5.5. [Assets View](#)
- 5.6. [Point Types View](#)
- 5.7. [Custom Fields](#)
 - 5.7.1. [Create your Custom Fields](#)
 - 5.7.2. [Editing Points' Custom Fields](#)
- 5.8. [Access Rights](#)
 - 5.8.1. [Basic Rights](#)
 - 5.8.2. [Multiple Users/Groups/Roles](#)
 - 5.8.3. [One Point With Multiple Roles](#)
 - 5.8.4. [One User In Multiple Groups](#)
 - 5.8.5. [Users/Groups](#)
 - 5.8.6. [Roles](#)
 - 5.8.7. [Identity Providers](#)
- 5.9. [Logs](#)
- 5.10. [Alarms](#)
 - 5.10.1. [Setting Up An Alarm](#)
 - 5.10.2. [Assigning Alarms](#)
 - 5.10.3. [Alarm Notifications](#)
 - 5.10.4. [Interacting with Alarms](#)
- 5.11. [CSV Import/Ingest](#)
 - 5.11.1. [Rules](#)
- 5.12. [Room Booking](#)
 - 5.12.1. [Room Booking configuration overview](#)
 - 5.12.2. [Fields for Exchange and Office 365](#)
 - 5.12.3. [Fields for Google Suite](#)
 - 5.12.4. [Common Room Booking Parameters](#)
 - 5.12.5. [Testing connections](#)
 - 5.12.6. [Microsoft Exchange Settings](#)
 - 5.12.7. [Google Suite Settings](#)
- 6. [GUI Editor](#)
 - 6.1. [Assets](#)
 - 6.1.1. [Default Assets](#)
 - 6.1.2. [Uploading Assets](#)
 - 6.1.3. [Assets Explorer](#)
 - 6.2. [Menu Builder](#)
 - 6.2.1. [Sections](#)
 - 6.2.2. [Options](#)
 - 6.3. [Expert Mode](#)
 - 6.3.1. [Scratch Pad](#)
 - 6.3.2. [Database Points](#)
 - 6.4. [Live Preview](#)
 - 6.4.1. [Size Options](#)
- 7. [Scheduler](#)
 - 7.1. [Event Definition](#)
 - 7.2. [Events View](#)
 - 7.2.1. [Creating Events](#)
 - 7.2.2. [Editing an Existing Event](#)
 - 7.2.3. [Searching for an Event](#)
 - 7.3. [Calendar View](#)
 - 7.3.1. [Navigation](#)
 - 7.3.2. [Calendar Month, Week and Day Views](#)
- 8. [Additional Information](#)



- 8.1. [Backups/Restoring](#)
 - 8.1.1. [Backups](#)
 - 8.1.2. [Restoring](#)
- 8.2. [Updating Overture](#)
 - 8.2.1. [Updating UX And CS Server](#)
 - 8.2.2. [Updating Extra Control Servers](#)
- 8.3. [Linking to Specific Maps or Pages](#)
 - 8.3.1. [Home](#)
 - 8.3.2. [Magic Menu](#)
- 8.4. [Home Extras](#)
 - 8.4.1. [Widgets](#)
 - 8.4.2. [Branding](#)
 - 8.4.3. [Dashboard Widget Chart Coloring](#)
- 8.5. [HTTPS](#)
 - 8.5.1. [Switching to HTTPS:](#)
 - 8.5.2. [Switching to HTTP:](#)
- 8.6. [Translations Quick Reference](#)
 - 8.6.1. [Introduction](#)
 - 8.6.2. [Getting Started](#)
 - 8.6.3. [Fallbacks](#)
- 9. [Monitor](#)
 - 9.1. [RMM Monitor Main Grid Widget](#)
 - 9.1.1. [Alarms is a filter providing 3 values:](#)
 - 9.1.2. [Tree View](#)
 - 9.1.3. [Main View](#)
 - 9.2. [Other RMM Monitor Widgets](#)
 - 9.2.1. [Big Text](#)
 - 9.2.2. [Date](#)
 - 9.2.3. [Grid](#)
 - 9.2.4. [Alarms](#)
 - 9.2.5. [Alarm Grid](#)
 - 9.2.6. [Table](#)
 - 9.2.7. [Pie Chart](#)
 - 9.2.8. [Bar Chart](#)
 - 9.2.9. [Common Widget Options](#)
 - 9.3. [RMM Alarms and Notifications](#)
 - 9.3.1. [Alarm Templates](#)
 - 9.3.2. [Monitor native alarms](#)
 - 9.3.3. [Overture UX alarms](#)
 - 9.3.4. [Notifications](#)
 - 9.3.5. [Alarms and Notifications considerations](#)
 - 9.4. [RMM User Management](#)
 - 9.4.1. [Adding new users](#)
 - 9.4.2. [Groups / Tenants](#)
 - 9.4.3. [Identity Providers](#)
 - 9.4.4. [Identity Provider Mappers](#)
 - 9.5. [RMM Custom Logo](#)
- 10. [Insights](#)
 - 10.1. [RMM Insights Access](#)
 - 10.2. [Selecting, customizing your report widget](#)
 - 10.2.1. [Insights Widget Overview](#)
 - 10.2.2. [Selecting and Customizing a Report](#)
 - 10.2.3. [Using Reports Options](#)
 - 10.2.4. [Customizing dashboard](#)
- 11. [APPENDIX A](#)
 - 11.1. [Using Overture with Version 1 Control Servers \(including Manager V6, Showmaster V2, and Overture CS1\)](#)

- [11.1.1. Control Server Setup](#)
 - [11.1.2. Overture Setup](#)
 - [11.1.3. Ingesting The Project](#)
- 12. [APPENDIX B](#)
 - [12.1. Expert Mode Programming](#)
 - [12.1.1. Tags](#)
 - [12.1.2. Attributes](#)
 - [12.1.3. AngularJS Items](#)
- 13. [APPENDIX C](#)
 - [13.1. Tasks / User Variable Programming](#)
 - [13.2. Tasks](#)
 - [13.2.1. Control Server Type 1](#)
 - [13.2.2. Control Server Type 2](#)
 - [13.2.3. Task General Format](#)
 - [13.3. Variables](#)
 - [13.3.1. Control Server Type 1](#)
 - [13.3.2. Control Server Type 2](#)
- 14. [APPENDIX D](#)
 - [14.1. Network Architecture and Security](#)
 - [14.2. Overture Architecture](#)
 - [14.2.1. Overture On-Premise architecture](#)
 - [14.2.2. Overture Cloud Architecture](#)
 - [14.2.3. Local network connectivity and traffic](#)
 - [14.3. Access to internet](#)
 - [14.3.1. Overture Driver Hub](#)
 - [14.3.2. Overture Upgrade Server](#)
 - [14.3.3. License servers](#)
 - [14.4. FAQ](#)
 - [14.4.1. What happens when the network infrastructure goes down?](#)
 - [14.4.2. Is any content data being shared?](#)
 - [14.4.3. Is the communication encrypted in any way?](#)
- 15. [APPENDIX E](#)
 - [15.1. Overture ServiceNow Integration Guide](#)
 - [15.2. ServiceNow Configuration Steps Overview](#)
 - [15.2.1. Steps for Incident](#)
 - [15.2.2. Steps for Case Management](#)
 - [15.3. Common Pre-requisite](#)
 - [15.3.1. Login as admin](#)
 - [15.3.2. Elevate admin role](#)
 - [15.4. Configuration for Incident](#)
 - [15.4.1. Role Configuration](#)
 - [15.4.2. User Configuration](#)
 - [15.4.3. oAUTH Configuration](#)
 - [15.4.4. Incident Access Controls](#)
 - [15.4.5. Dictionary Access Controls](#)
 - [15.5. Configuration for Case Management](#)
 - [15.5.1. Install CSM on your instance](#)
 - [15.5.2. Role Configuration](#)
 - [15.5.3. User Configuration](#)
 - [15.5.4. oAUTH Configuration](#)



2 Overview

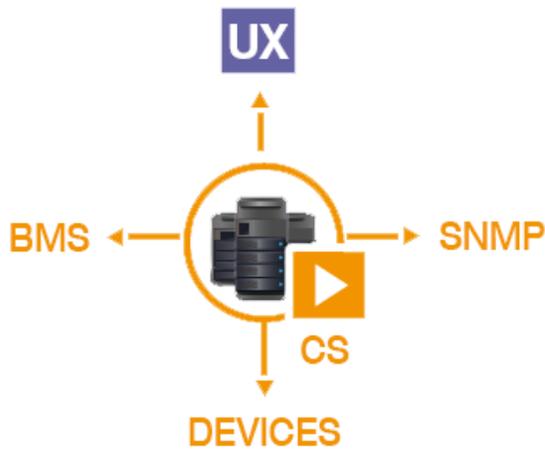
Barco Overture is an effective enterprise-wide A/V control software. It controls, monitors, and automates thousands of AV devices in multiple locations, integrates with IT services, and delivers highly interactive user interfaces.

Barco Overture, as Enterprise Class software, is designed to run in a virtualized fault tolerant environment and is made of three main software components:

- Control Server, which controls devices, runs behaviors (automation modules, see [Behaviors](#)), and
- User Experience Server, which generates and serves all the user interfaces (Help desk, BYOD, In-room GUI).
- RMM Monitor & Insights, for live monitoring and analytics across multiple UXs

A User Experience Server can be connected to one or several Control Servers, depending on the architecture required or the size of the project. A large venue may use one User Experience Server with one Control Server to manage all devices in the same building, while a corporation may use one User Experience Server at its headquarters, connected to 20 Control Servers—one in each branch or country—to centrally manage and control all its AV devices worldwide.

2.1 Control Server



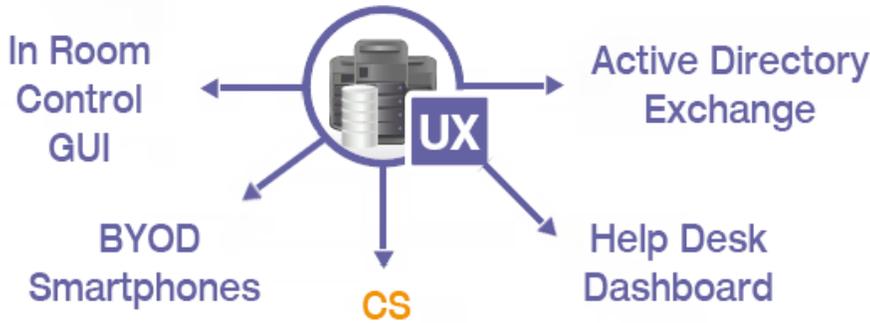
The Control Server controls audiovisual devices via TCP/IP and other protocols using IP to serial or I/O converters and includes drivers for industry standard audiovisual equipment.

Devices, such as projectors, displays, video conferencing, audio, lighting and HVAC systems, are controlled and monitored giving full building control.

It includes a graphical element for controlling devices directly and a complete SDK for developing custom drivers.

It also runs behaviors (see [Behaviors](#)), which are automation modules for rooms, floors, buildings or other entities.

2.2 User Experience(UX) Server



The User Experience Server categorizes all controlled A/V equipment in an infrastructure database by type, location, and an unlimited number of sub-categories and tags. It automatically generates all graphical user interfaces for the help desk, mobile devices, and in-room control interfaces using information from the infrastructure database.

Graphical user interfaces are served as HTML5 and use CSS to manage the look and feel, delivering a consistent user experience throughout the system, anywhere in the world, supporting multiple language versions. The help desk dashboard displays live data about device status, such as video projector lamp usage, rack room temperatures, or device failures, while the map interface uses a familiar graphical building plan model to geographically access remote room control interfaces as well as device control panels.

The UX Server consists of several GUIs:

- [Home](#) is the first interface into Overture. It houses facility maps, room or device based control panels, and a front-end dashboard for viewing helpful details at a moment's notice.
- [Magic Menu](#) is the mobile device GUI. It consists of multiple pages used for specific room or device based controls.
- [Scheduler](#) is the tool which allows scheduling automatic events at specific periods of time.
- [Configurator](#) is a gateway into the UX Server database. It's used for creating maps, rooms, devices and adding meta-data to those items. It's also used to create access rights for the system, and error notifications.
- [GUI Editor](#) is the assets management tool for the UX Server. It allows you to add your own files, such as images, and file structure to Overture. It also is used to create or edit HTML templates used by the interfaces

Note to the reader: [Home](#) is the name of the first user interface which hosts both the map and the dashboard views. [Home](#) used to be named [Dashboard](#) in previous Overture versions, therefore sometimes, [home](#) and [dashboard](#) terms are used indifferently.

2.2.1 Login to Overture

All applications in Overture use a single login mechanism. Login once and you'll continue to stay logged in throughout the application.

To login to Overture open up your web browser and type the URL of your UX HTTP server to be presented with the login screen.

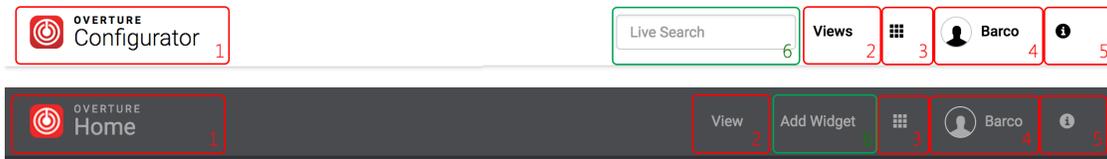
The screenshot shows a login form on a dark background. On the left, there are two input fields: 'Username' with the text 'Admin' and 'Password' with masked characters. Below the password field is a 'Remember me' checkbox. To the right of the password field is a red 'Log in' button. To the right of the username field is a grey 'ADFS Login' button.

You can choose to login with either a username and password or, if setup, using ADFS via the button to the right of the form.

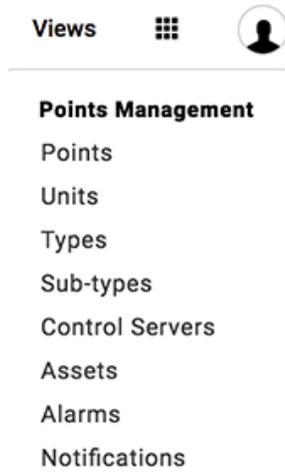
2.2.2 Common Header

All GUIs share the same header bar which gives access to global common functions and some application dedicated

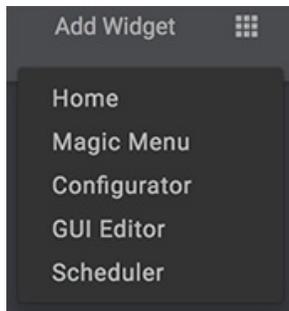
functions:



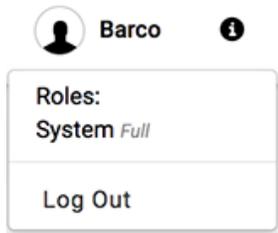
1. Application name.
2. View menu, gives access to the different application views (Configurator example):



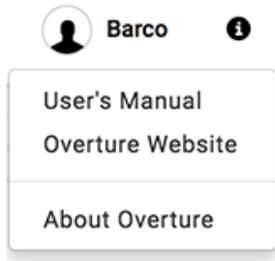
3. App Switcher, select the application to display (Home example):



4. User, displays the current user roles and log out menu (Configurator example):



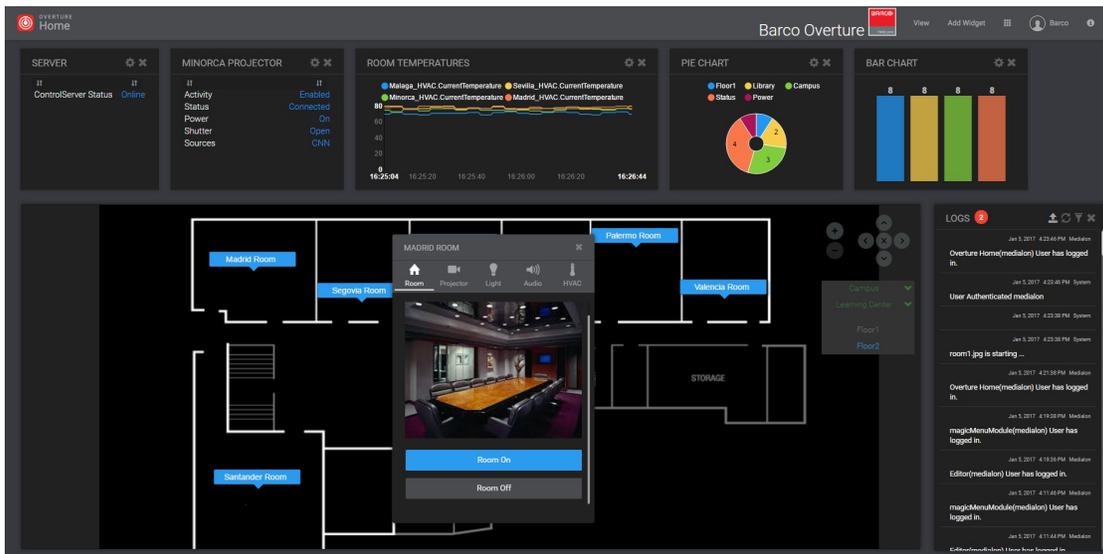
5. About menu and documentation, resources links (Configurator example):



6. Specific application menu.

- Configurator: 'Live Search' allows searching through points
- Home: 'Add Widget' allows to add a new widget in the Dashboard

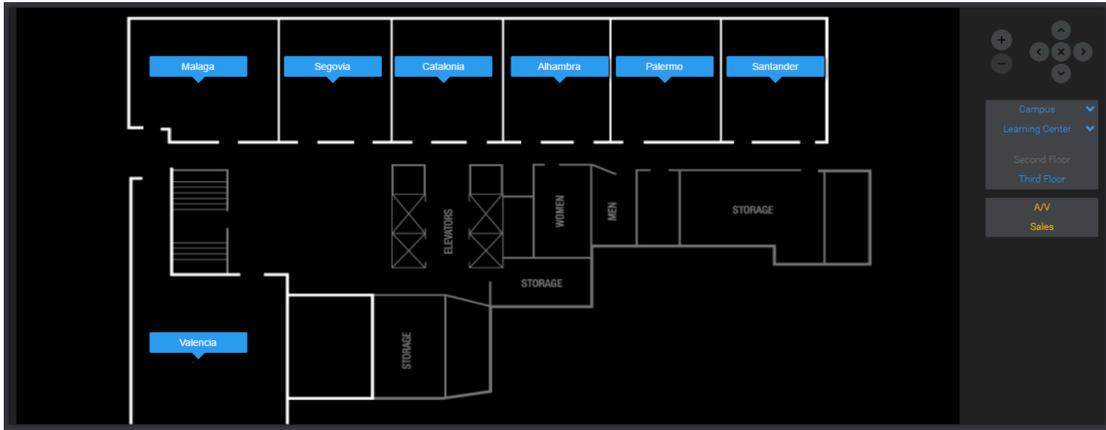
2.2.3 Home



Home is the main GUI in the Overture system. You first access Home by opening up a web browser and typing the IP address of your UX HTTP server. (Example: <http://10.0.50.231/>)

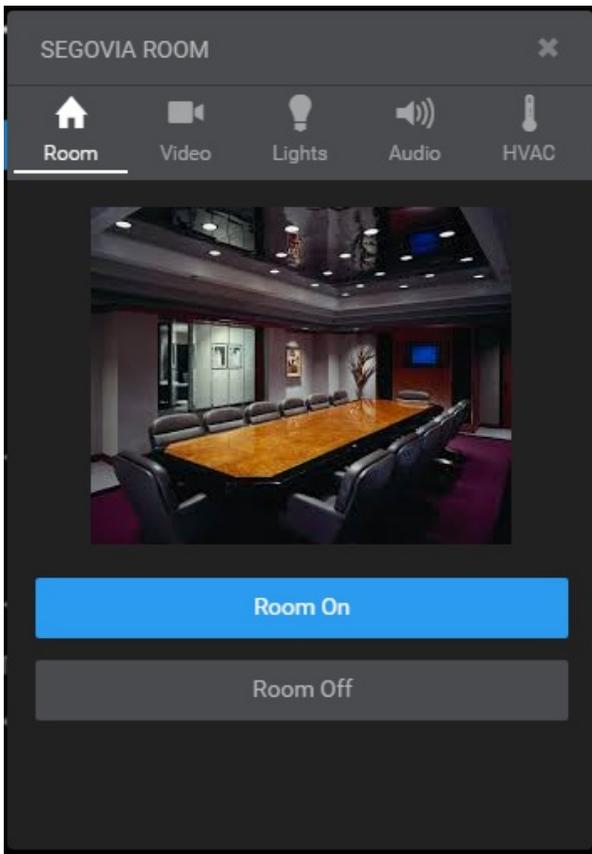
Your Home will consist of four main components to help you monitor and control devices: Maps, Dashboard, Control Panels and Logs

2.2.3.1 Maps



Maps are images that are displayed within the interface. Inside of a map, you can create links to control panels or other maps for a layered monitoring service.

2.2.3.2 Control Panels



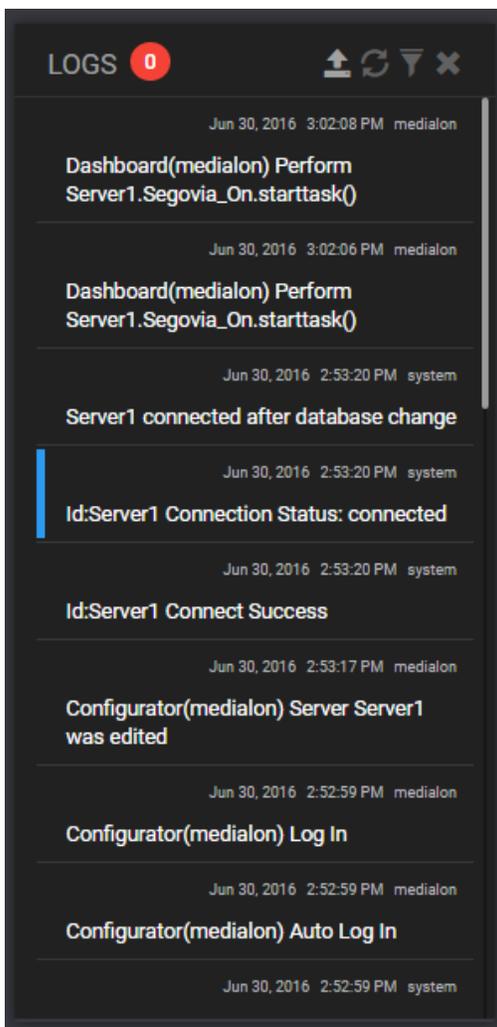
Control panels are HTML templates displayed within the interface. Inside of control panels is where you will interact and monitor your devices directly.

2.2.3.3 Dashboard



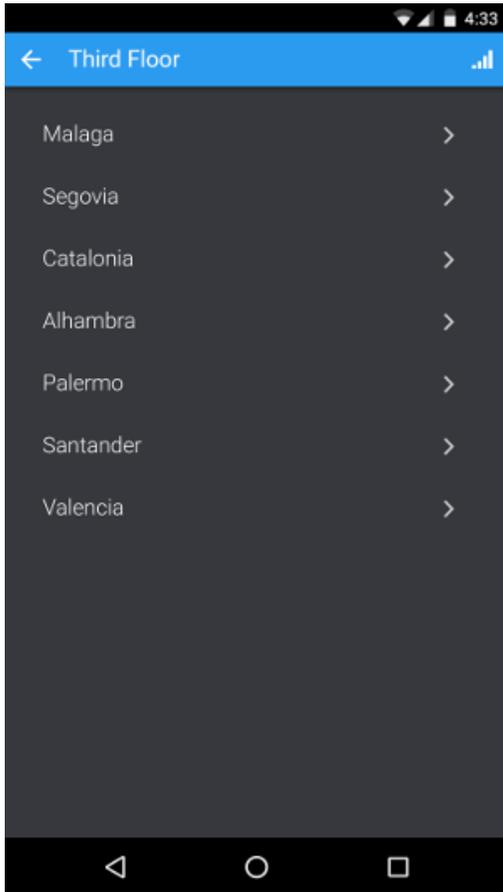
Dashboard is a view of multiple monitoring widgets. You can add graphs to track temperature, tables to display occupancy, or add bookmarks to open layered panels in one click.

2.2.3.4 Logs



Logs inside of the interface give detailed accounts of how devices inside your system are being used and what each user is doing within the system.

2.2.4 Magic Menu



(This is an example of a Magic Menu page)

Magic Menu is the second GUI in the Overture system. This is used for mobile and tablet devices. You can access it through Home, by going into the user menu, or by opening a browser and going to the URL of your UX Server and adding '/magicmenu'. (Example: <http://10.0.50.231/magicmenu>)

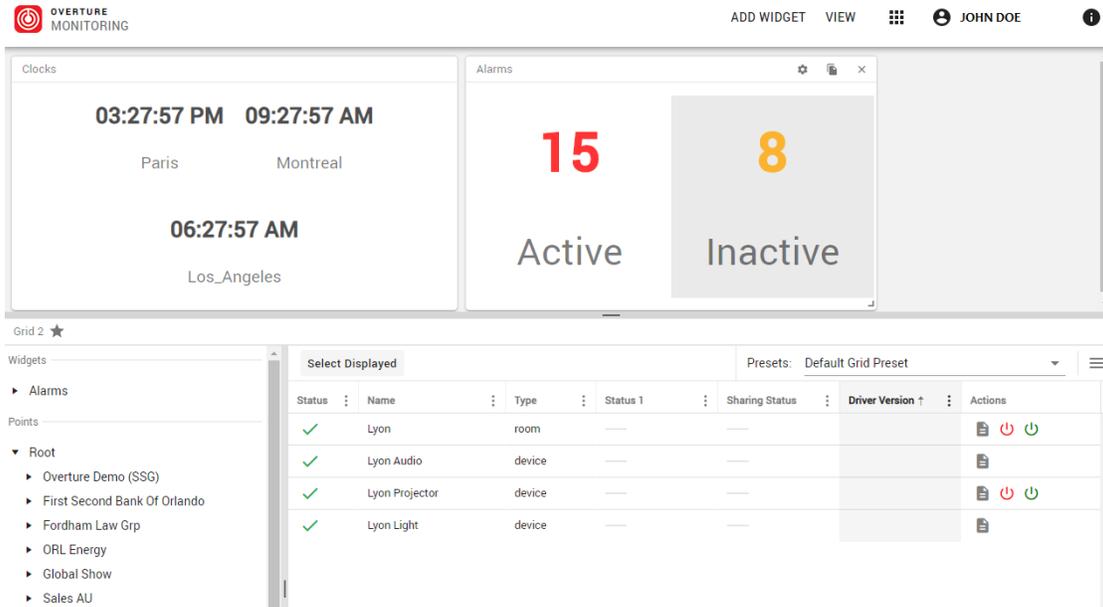
2.3 RMM Monitor

Overture RMM Monitor is a dashboard view web page. The dashboard view for the first-time user login is made of an Alarm widget, a Clock Widget and a main Grid Widget. However, the default Administrator can also customize his dashboard and his default dashboard will become the default dashboard of all newly create users.

The Dashboard provides two types of pages, the Monitor pages for real-time monitoring widgets and the Insights pages for historical data reports (new from RMM 4.8.0).

A dashboard can be customized by adding new widgets, or by adding new columns to the main grid widget within a Monitor page, or adding new reports in an Insights page. Multiple pages of each type can be created within the Dashboard.

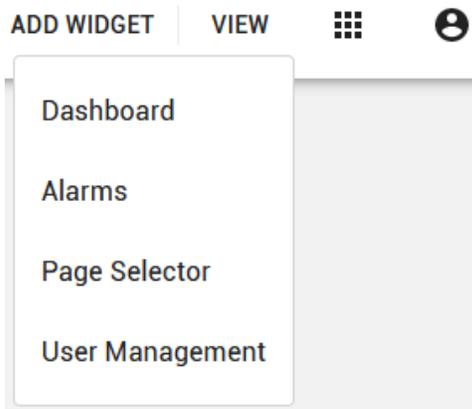
Currently RMM is available only as a Cloud.



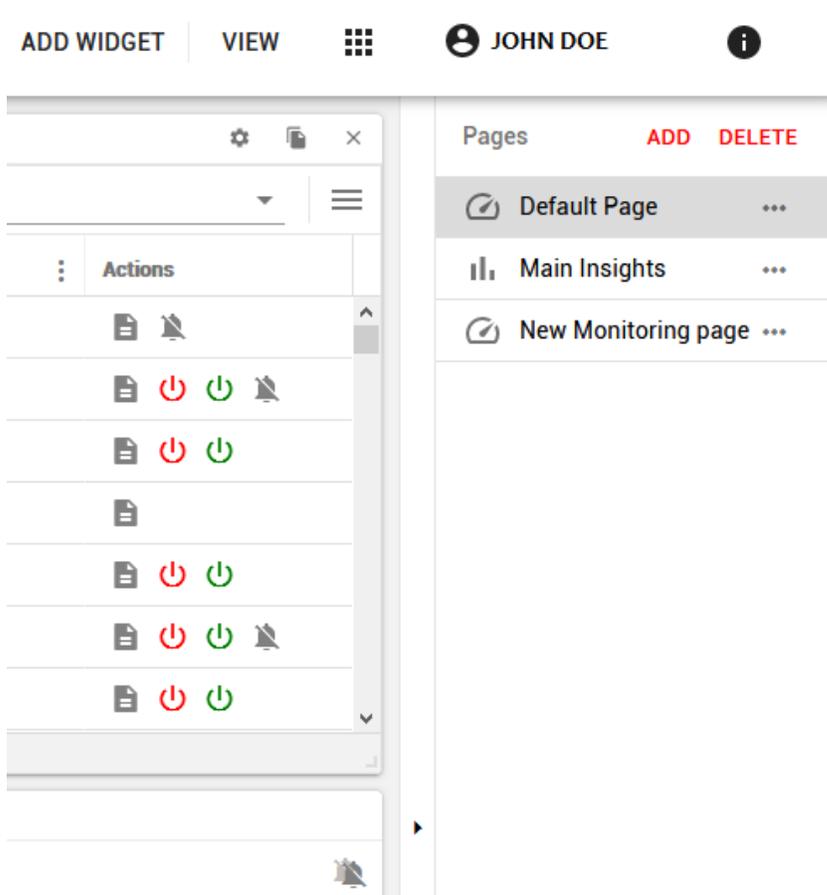
2.3.1 RMM Monitor Features

RMM dashboard is the placeholder for the monitoring widgets. Dashboard is represented by a page specific to its user. Each user can customize his dashboard by adding widgets specific to their needs. In addition, a user can add more dashboard pages to create different views.

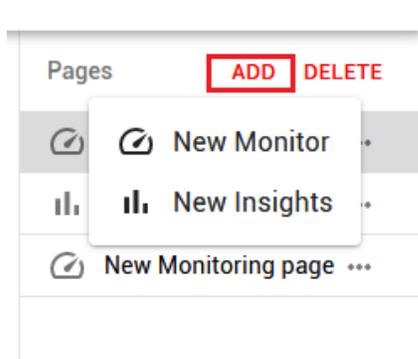
Adding more dashboard pages: the first dashboard page is your default page. In order to add more pages, either click on the arrow in the middle of the right corner or select under "VIEW" on the top menu and the "Page Selector".



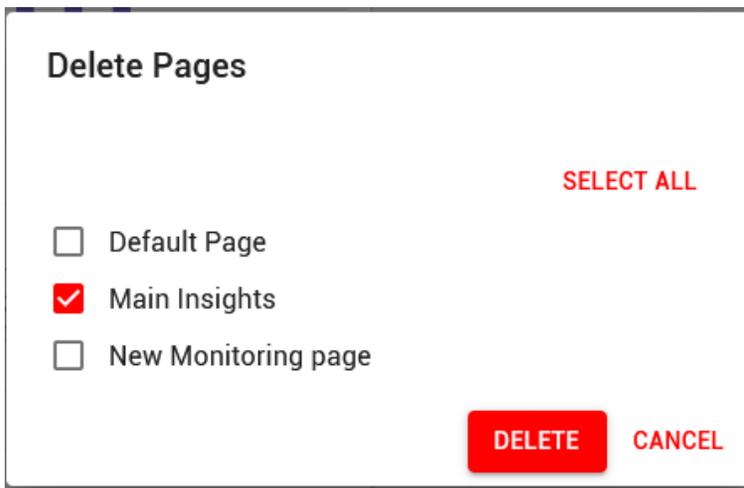
This action will open a right pane.



To create (add) a new page, click on "Add" button and select the type of page to create



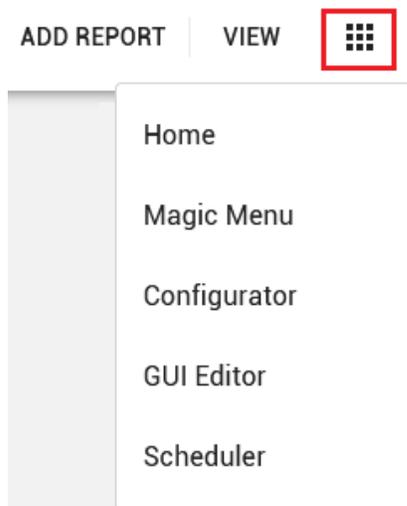
To delete a page, click on the "Delete" button, a dialog will open with the current active page selected for deletion. You can just click "Delete" to confirm, or refine your selection if you want to delete a different page or multiple pages at a time



Note: If you select all the pages to be deleted, a warning will let you know that this is not permitted.

You can also use the the 3 dots, right next to the page buttons, to Duplicate, Rename or Delete a particular page

Application switcher Icon on top right menu allows you to open other modules:



- *Home* Specific to tenant will need to select a tenant based on your tenant access right
- *Magic Menu* Specific to tenant will need to select a tenant based on your tenant access right
- *Configurator* Specific to tenant will need to select a tenant based on your tenant access right
- *Gui Editor* Specific to tenant will need to select a tenant based on your tenant access right
- *Scheduler* Specific to tenant will need to select a tenant based on your tenant access right

The "VIEW" in the top menu gives access to "Dashboard", "Page Selector" and "User Management" (based on access right).

Your "Name" in the top menu gives you access to Logout



Logo gives access to Overture Website and About Overture

2.4 RMM Insights

Insights is the reporting module of Overture RMM. Its access is available to all RMM users managed by RMM tenant access right. In other words, RMM users can see reports only for tenant points for which they have access to.

Since version 4.8.0, Insights module has been integrated as part of the RMM Monitor app through Dashboard dedicated Insights page type.

State changes on points for Insights are collected in real time, however there are transformation processes against these data that are happening on a regular basis before the data can be rendered in Insights reports. This data is first transformed and then it populates the Insights database, from which the reports are built.

RMM monitors and tracks devices and meeting rooms state. Those states are repeatedly changing and transitioning from one state to another based on various events, triggers, macros, scheduled events, etc. Sometimes issues are risen triggering alarms and requiring interventions. Rooms are booked at different time slots and not necessarily used, devices powered on more than the time they have been used and displayed unplugged by some users, etc.

Customers and service providers need to access those historical data in order to set corrective actions.



2.4.1 RMM Insights Features

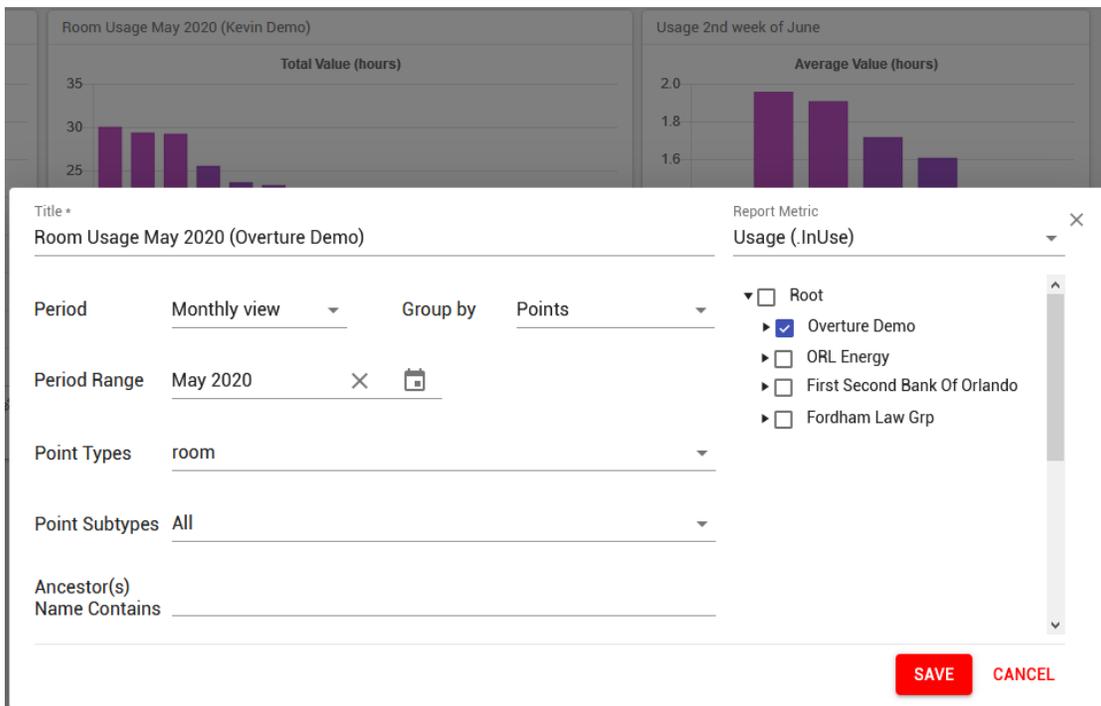
2.4.1.1 Report Types

The current version of Insights provides five types of reports:

- Usage
- Power
- Room Booking
- Room Booking vs Usage
- Connection Status

2.4.1.2 Usage

RMM Insights calculates the duration of usage in hours or in minutes. The usage reports are available for devices, rooms or any containers for which the InUse behavior has been configured. In all cases, the reporting is done based on the InUse variable.

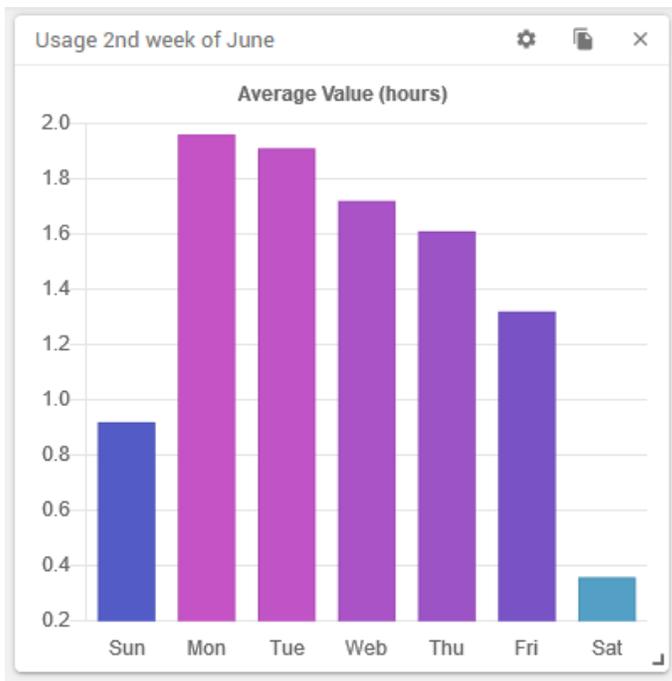


For devices, "InUse" is one of the harmonized variables. If present, the Usage report could display the total or the average duration of usage within the selected time slot (total values are provided for Yearly and Monthly periods whereas average values are provided for Daily and Hourly periods).

Having a device usage report has multiple benefits:

- See what types of AV devices is mostly been used
- See when to schedule a preventive maintenance
- Justify your future investment

For rooms, the usage is based on the availability of the "InUse" variable at the room level. A new "overture_room_inuse" behavior has been added in Overture Plugins Online Library which could be downloaded, configured and associated to rooms (or any containers such as floors or buildings if required). The presence of this variable at the room level will allow to track the room usage in a usage report.



2.4.1.3 Power

The Device power report is based on the availability of Power variable. If available, the Power report will calculate and display the total or the average power usage time for the select time slot (total values are provided for Yearly and Monthly periods whereas average values are provided for Daily and Hourly periods).

Device Power Monthly (Overture Demo)

Report Metric: Power

Period: Daily view | Group by: Period

Period Range: 06/14/2020 - 06/20/2020

Point Types: device

Point Subtypes: All

Ancestor(s) Name Contains: _____

- Root
- Overture Demo
- ORL Energy
- First Second Bank Of Orlando
- Fordham Law Grp

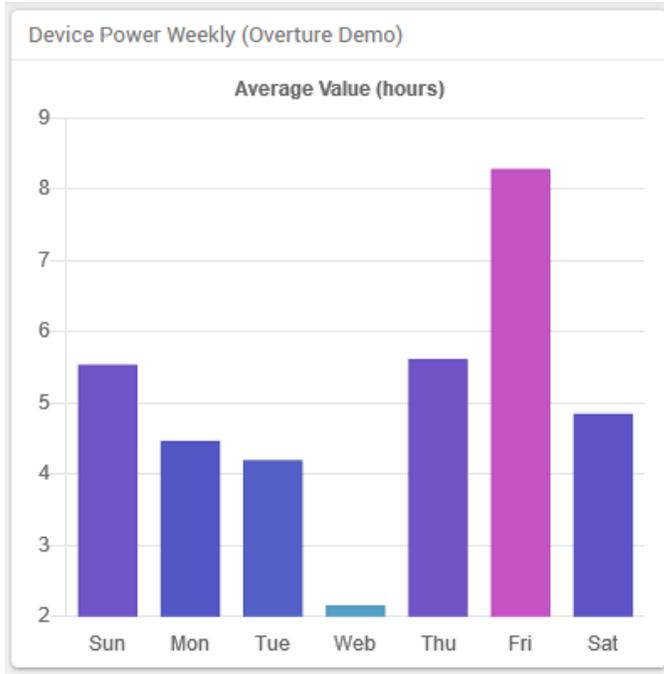
SAVE **CANCEL**

The Container/Room power report is based on the use of "Overture Power Cycle" behavior. If available, the Power report will be calculated based on the variable Power, the total or the average duration of the power usage with the selected time

slot (total values are provided for Yearly and Monthly periods whereas average values are provided for Daily and Hourly periods).

The main benefit of a Power report is the ability to:

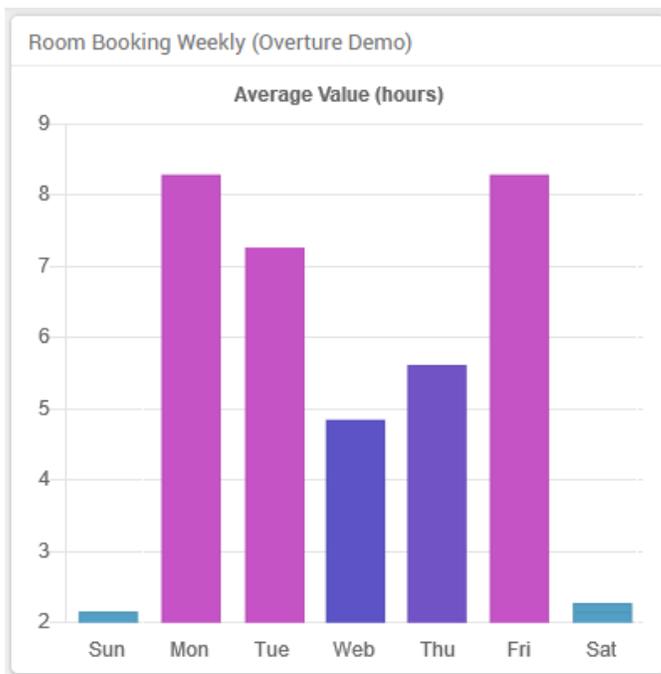
- Check the duration and peaks of devices or rooms in power
- Review the justification of the power usage as it may affect overall power consumption and raise the maintenance cycle



2.4.1.4 Room Booking

The Room booking report is based on the availability of "Meeting" variable associated to the room. The report will calculate the total or the average of room booking in hours for the selected time slot (total values are provided for Yearly and Monthly periods whereas average values are provided for Daily and Hourly periods).

A Room Booking report could indicate the length at which some rooms are booked versus other rooms. You could see if this is justified.

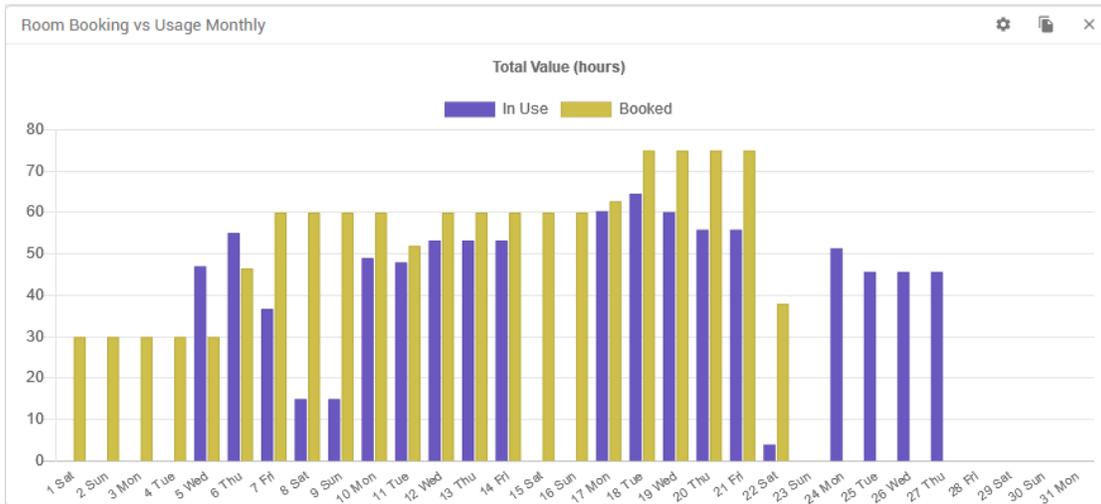


2.4.1.5 Booking versus Used

This report compares side by side the value for “Usage” and the value for “Booking” for rooms. This report aims at evaluating the effectiveness of booking in the room usage.

The dialog box is titled "Room Booking vs Usage Monthly". It contains the following fields and options:

- Title:** Room Booking vs Usage Monthly
- Report Metric:** Booking vs Usage (.Meeting/.InUse)
- Period:** Monthly view
- Group by:** Period
- Period Range:** June 2020
- Point Types:** device
- Point Subtypes:** All
- Ancestor(s) Name Contains:** (empty)
- Report Metric Filter:**
 - Root
 - Overture Demo
 - ORL Energy
 - First Second Bank Of Orlando
 - Fordham Law Grp
- Buttons:** SAVE, CANCEL



Since this report show two metrics in the same view, you can click one of them in the widget header to toggle this metric on or off in the view, resulting into showing only one of the two

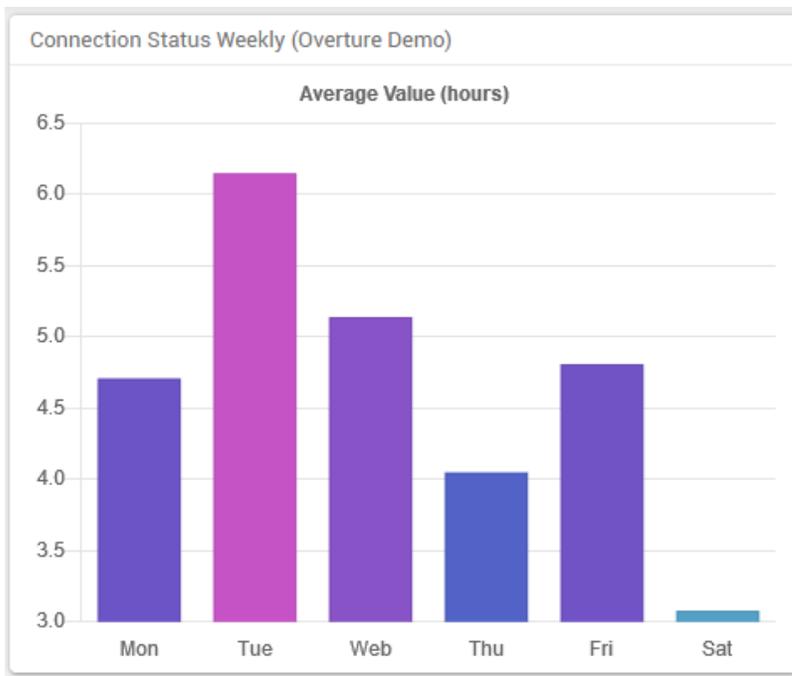




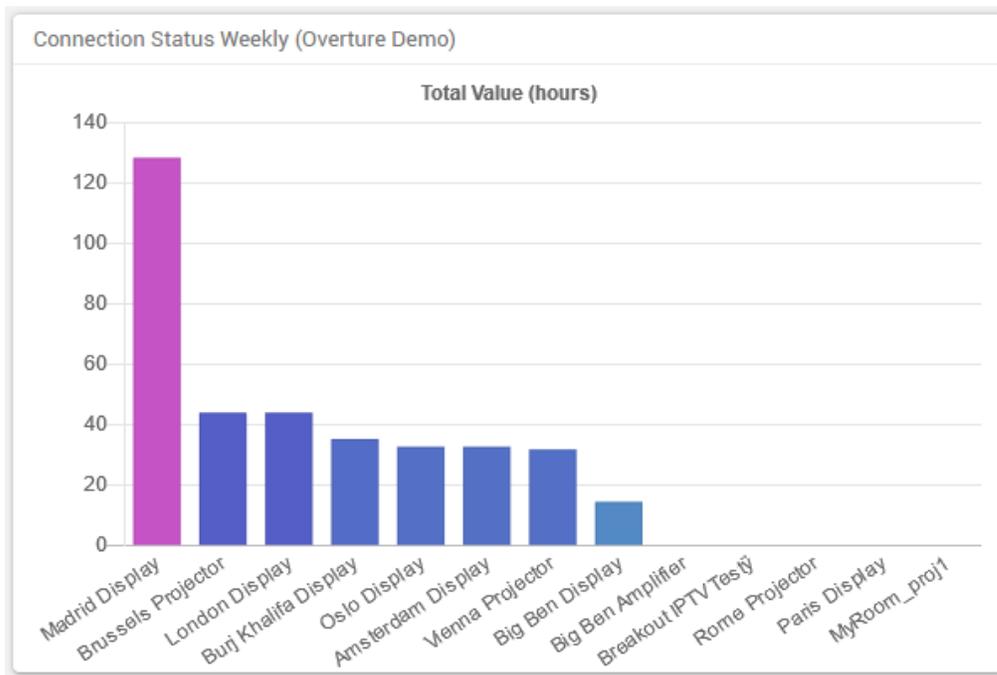
2.4.1.6 Connection Status

The status report is providing the uptime duration status for Devices, Tenants and Control Servers over a given period: an hour, a day, a week, a month or a year.

Daily report grouped by period range (days of the week)

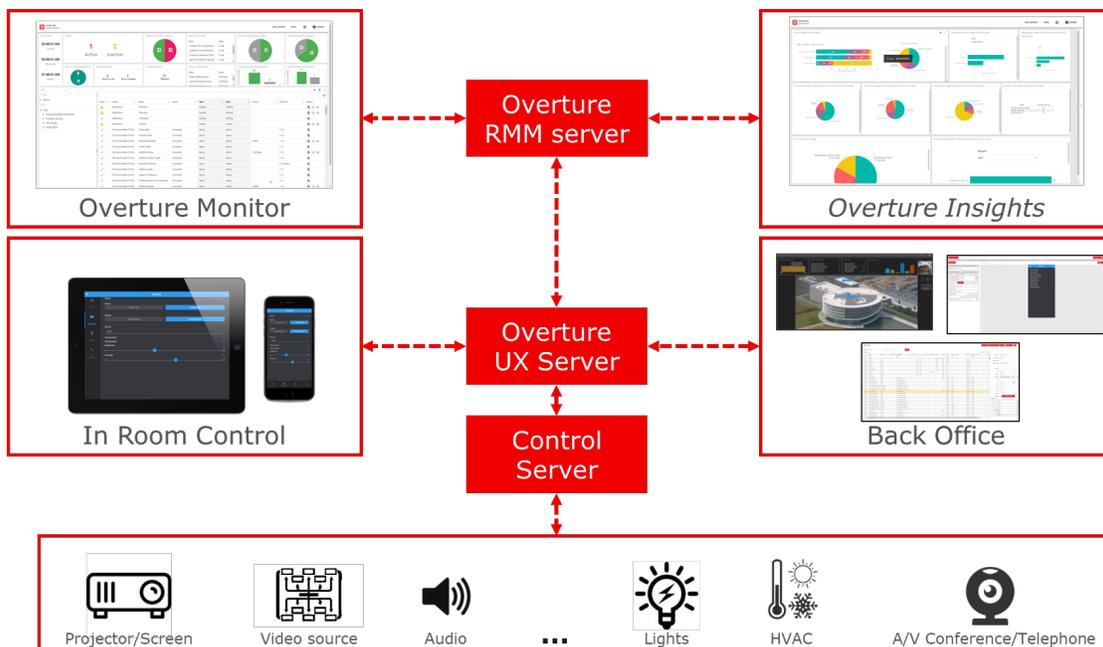


Daily report grouped by points



2.5 Overall Overture functional architecture (RMM, UX and CX)

RMM (Remote Monitoring & Management) and Overture Insights are the new Overture components. Although RMM is natively integrated with UX servers in this first deployment it requires its own user management and access right configuration. The merge of user management and access rights with UX is planned for 2020.



RMM is a multi tenant application capable of communicating and monitoring activities and device states from a multitude of UX servers each representing a different tenant. As such RMM is the perfect tool for Managed Service Providers in need of monitoring and controlling multiple customer installations from a single application.

3 System Architecture

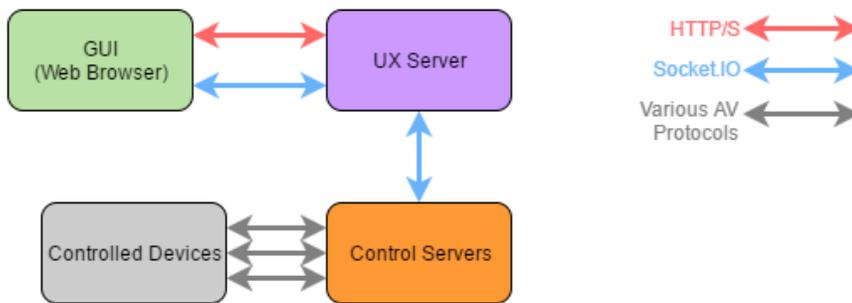
As enterprise class software, Overture is installed in a virtual environment. The needs of the system determine how the network topology is done.

Each topology will use the same components just in a different configuration.

- Browser: This is the web browser, whether it is on a PC or mobile device, that connects to UX Server and displays dynamically created content to the user.
- CS: A Control Server. It contains custom drivers for devices, opens TCP/IP communication with the devices, sends/receives information from the devices, and packages the information in a readable way back to the UX Server. It runs automation modules (see [Behaviors](#)) for rooms, floors or buildings for instance.
- UX: The User Experience Server. It contains a database for storing all of the information about the system, a file server for storing the HTML, CSS, and image files, and serves web pages directly to users.

3.1 Protocols

Protocols Used Between Overture Components



These protocols determine how each component(s) talk to each other in a common Overture installation.

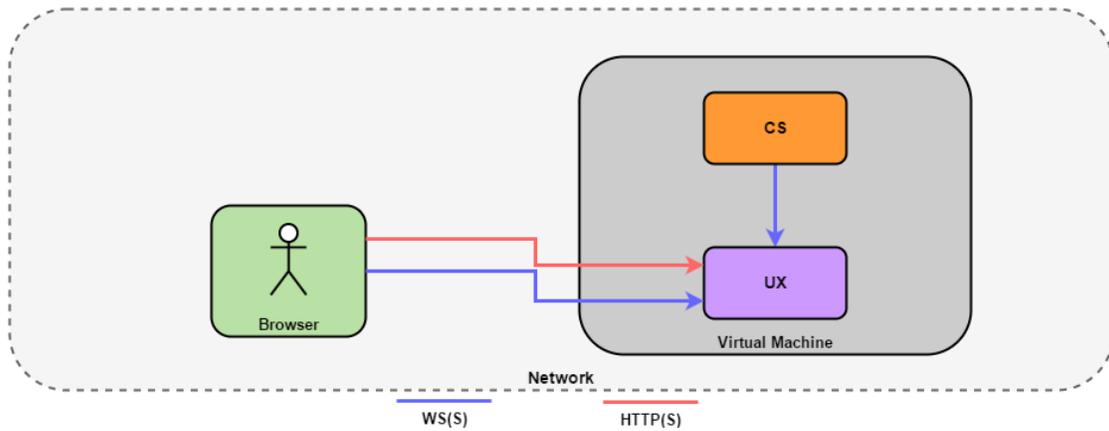
The browser talks with the UX Server via HTTP and WebSockets on port 80. If the connection is set to HTTPS, WebSocket Secure and port 443 are used instead.

The UX Server talks with the Control Servers also with WebSockets on port 80. If the connection is set to HTTPS, WebSocket Secure and port 443 are used instead.

The Control Servers will speak to the various devices on the network each with their own TCP, UDP, or HTTP based protocols. This includes things like JSON over TCP, REST, SNMP, ArtNet, etc.

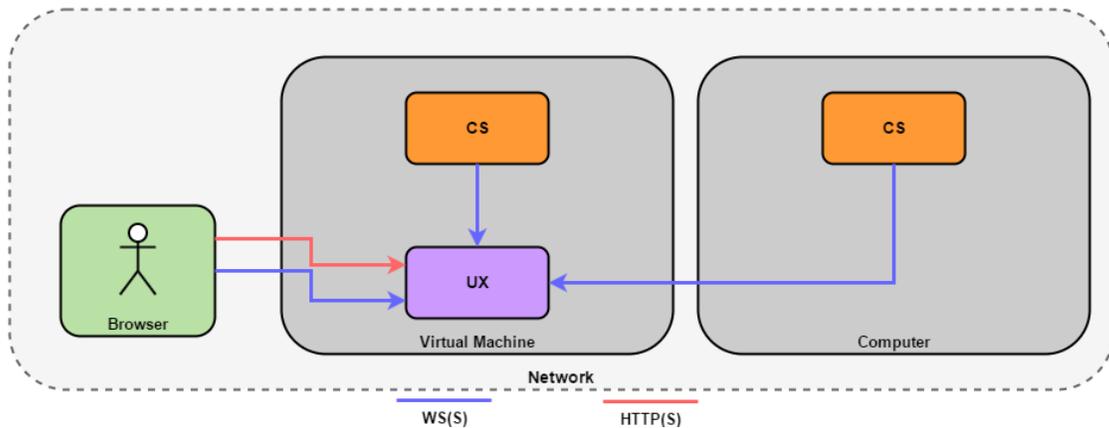
See also [APPENDIX D - Network Architecture and Security](#), for further network and security recommendations.

3.2 Single Control Server



In the basic setup, the virtual machine containing both the UX Server and the Control Server is installed on the network. The browser connects to the UX Server via HTTP(S), then starts a WS(S) connection to the UX Server, allowing information to be constantly updated in the browser. The Control Server also starts a WS(S) connection to the UX Server

3.3 Multiple Control Servers



When dealing with multiple control servers, the topology is very similar. The main difference is extra Control Servers will be installed on separate PCs across the network. The Control Servers still starts the connection to the UX Server via WS(S).

3.4 RMM Stackable Components

RMM can communication with several UX servers either in Cloud or On Premise.

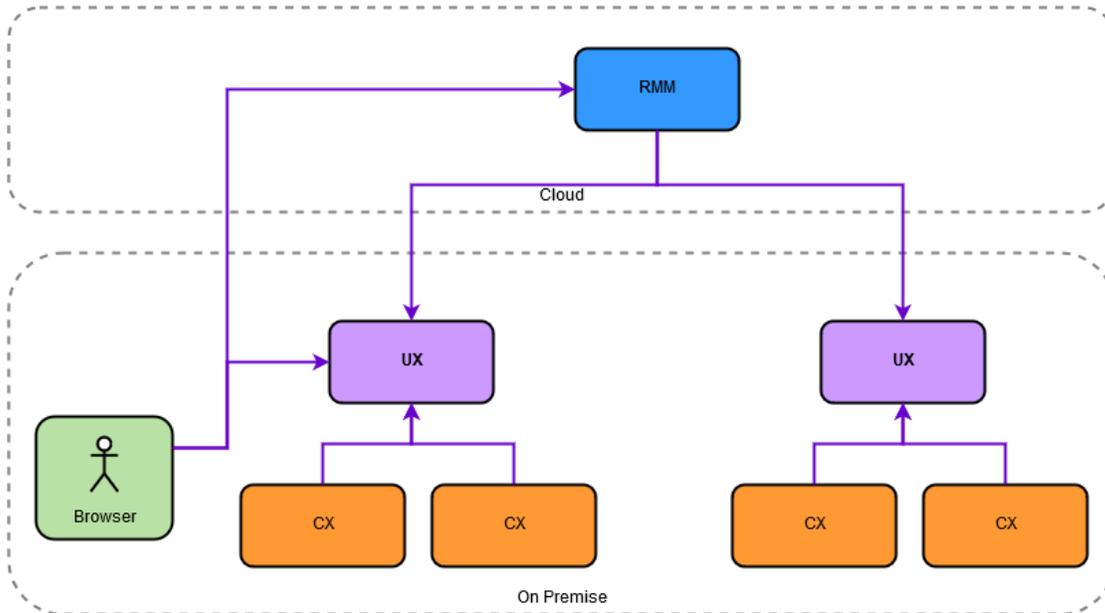


3.4.1 RMM Implementation Cloud - On Premise

For those customers with their Overture UX server (s) implemented On-Premise, it is necessary to connect an instance of Cloud RMM to the UX server On Premise.

The Overture RMM server needs to be able to contact the On Premise UX server. A port-forwarding must be setup into the On-Premise firewall. Protocol involved are http/https and websockets/secure websockets, depending on whether the UX On-Premise has been setup with http or https. In this setup, https is highly recommended.

The port-forwarding can use any entry port number, but the on-premise UX still expect communication on port 80/443. All communication between Overture RMM server and the UX server happens over https and secure websockets. It's a good best practice to put the UX server in the DMZ, such that there are no outside connections entering the actual network.

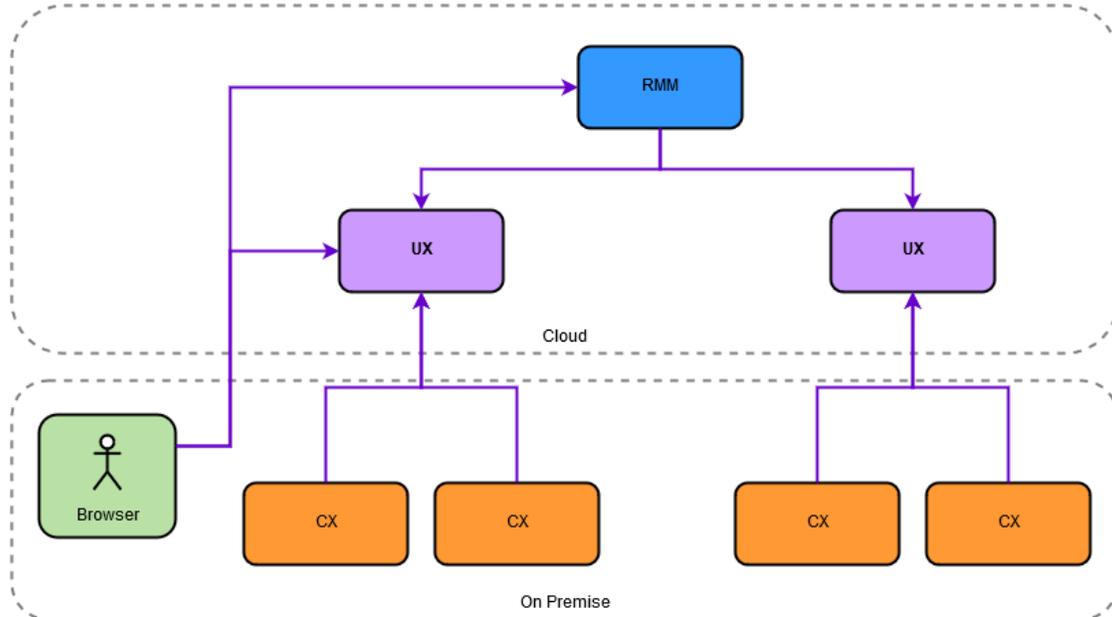


3.4.2 RMM Implementation Cloud - Cloud

In case of an Overture Cloud deployment, the RMM server and the UX server are set by Barco in the cloud (hosted on Amazon's AWS Ireland) and the CX server is installed On-Premise. Like with an Overture On-Premise installation, there can be multiple CX servers installed and configured per UX server.

Barco sets up a secure communication between the RMM server and the UX server, using https and secure websockets. All communication between the CX and the UX happens over https or secure websockets. The connection is initiated from

the CX, meaning that there is no need to open ports for incoming traffic in the firewall. In Figure 3 the arrows indicate where the communication originated (arrows point from client to server). The role of the UX and CX is the same between Overture On-Premise and Overture Cloud. As a result, similar communication to devices (for CX) and external cloud servers is expected.



4 Setup

Overture is run either on the Cloud or On-Premise inside of a virtual environment. In the virtual environment, the two components are installed together in one virtual machine. Additional Control Servers can also be setup and installed via a Windows installer or Docker container.

For further network and security information and recommendations, covering both Cloud and On-Premise deployments, see also [APPENDIX D - Network Architecture and Security](#).

4.1 Virtualized Install

Overture's virtualized install is done via a provided OVA file. This file contains the virtual machine with both UX Server, and a Control Server that optionally may be used. This file must be imported into your specific environment and set up.

4.1.1 Installing Guide

Follow the steps below to install

4.1.2 Requirements

1. 64bit OS
2. One of the following virtualized environment:
 - Oracle VirtualBox v.5.1.12 +
 - VMware Workstation 12 Player
 - VMware Workstation 12 Pro
 - VMware vSphere/vServer 6+
 - Microsoft Hyper-V 10+

4.1.3 Attention:

VM Internal Network

Overture VM uses 172.17.0.1/16 subnet internally. If this conflicts with your network, please follow the steps in the Docker section of this guide.

If you use a 32 bits machine, be aware that it must have Hardware Virtualization enabled at the BIOS level in order for a 64 bits virtual machine (like the Overture VM) to run.

VM Administrator Account Lockout

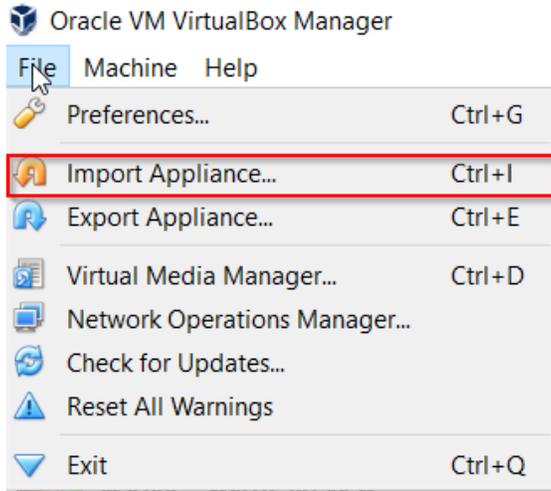
As a precaution against brute force attacks, the *Administrator* login will become locked for 10 minutes after 10 unsuccessful login attempts. If you notice this account locked, we recommend that you review your network policies and disable SSH access to VM from public networks. This could be an indication of unauthorized access.

4.1.4 Oracle Virtual Box Installation

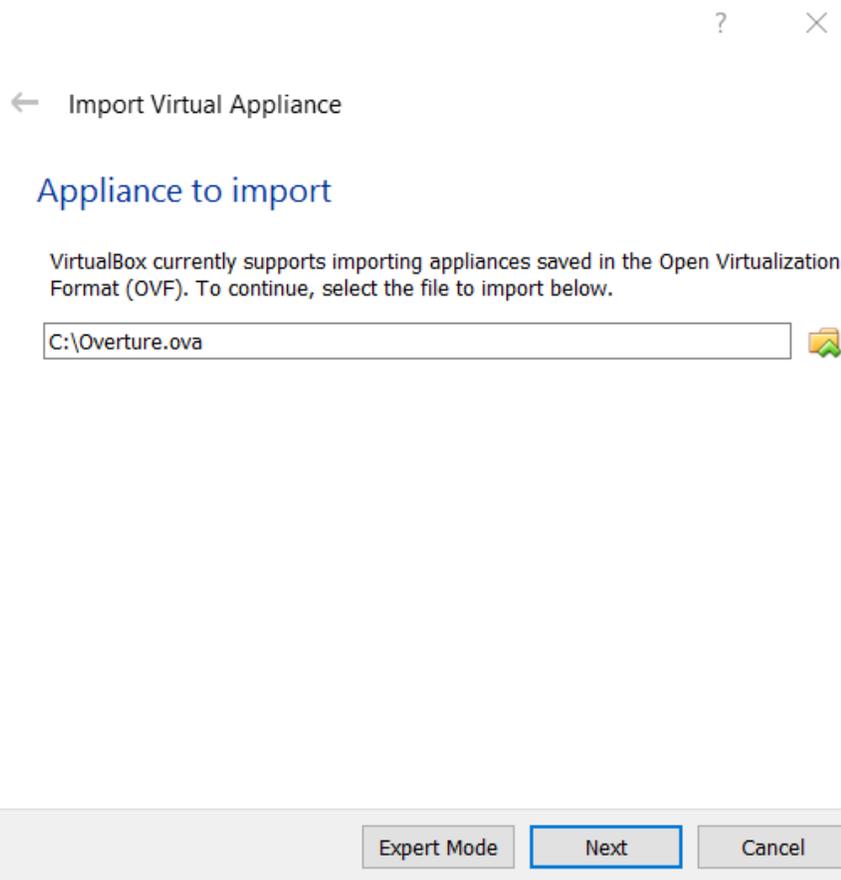
1. Open "Oracle VM VirtualBox Manager"
2. Click "File"



3. Click "Import Appliances"



4. Browse to the path of the Overture.ova file.



5. Click Next
6. Configure the VM settings or leave them as default
7. Click Import

← Import Virtual Appliance

Appliance settings

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

Description	Configuration
Virtual System 1	
Name	Overture
Guest OS Type	Debian (64-bit)
CPU	2
RAM	2048 MB
Sound Card	<input checked="" type="checkbox"/> ICH AC97
Network Adapter	<input checked="" type="checkbox"/> Intel PRO/1000 MT Desktop (82540EM)
Storage Controller (IDE)	PIIX4
Storage Controller (IDE)	PIIX4
Virtual Disk Image	C:\...disk1.vmdk

Importing Appliance ...: Importing appliance 'C:\Overture.ova'

Importing virtual disk image 'debian-8.6.0-amd64-v6-disk1.vmdk' ... (2/2)

6% X

3 minutes remaining

Reinitialize the MAC address of all network cards

Appliance is not signed

Restore Defaults Import Cancel

8. The VM is now installed .

Note: You will notice that the storage space is almost 300 Gb.

Storage

Controller: IDE Controller

IDE Primary Master: overture-disk1.vmdk (Normal, 976.56 GB)

This is an estimated size and is not provisioned in advance. The VM is allowed to potentially grow up to 300Gb. You can see the actual size as shown below.

- General
- System
- Display
- Storage
- Audio
- Network
- Serial Ports
- USB
- Shared Folders
- User Interface

Storage

Storage Tree

- Controller: IDE Controller
 - overture-disk1.vmdk

Attributes

Hard Disk: IDE Primary Master 🔗

Solid-state Drive

Information

Type (Format): Normal (VMDK)

Virtual Size: 976.56 GB

Actual Size: 3.17 GB

Details: Dynamically allocated differ...

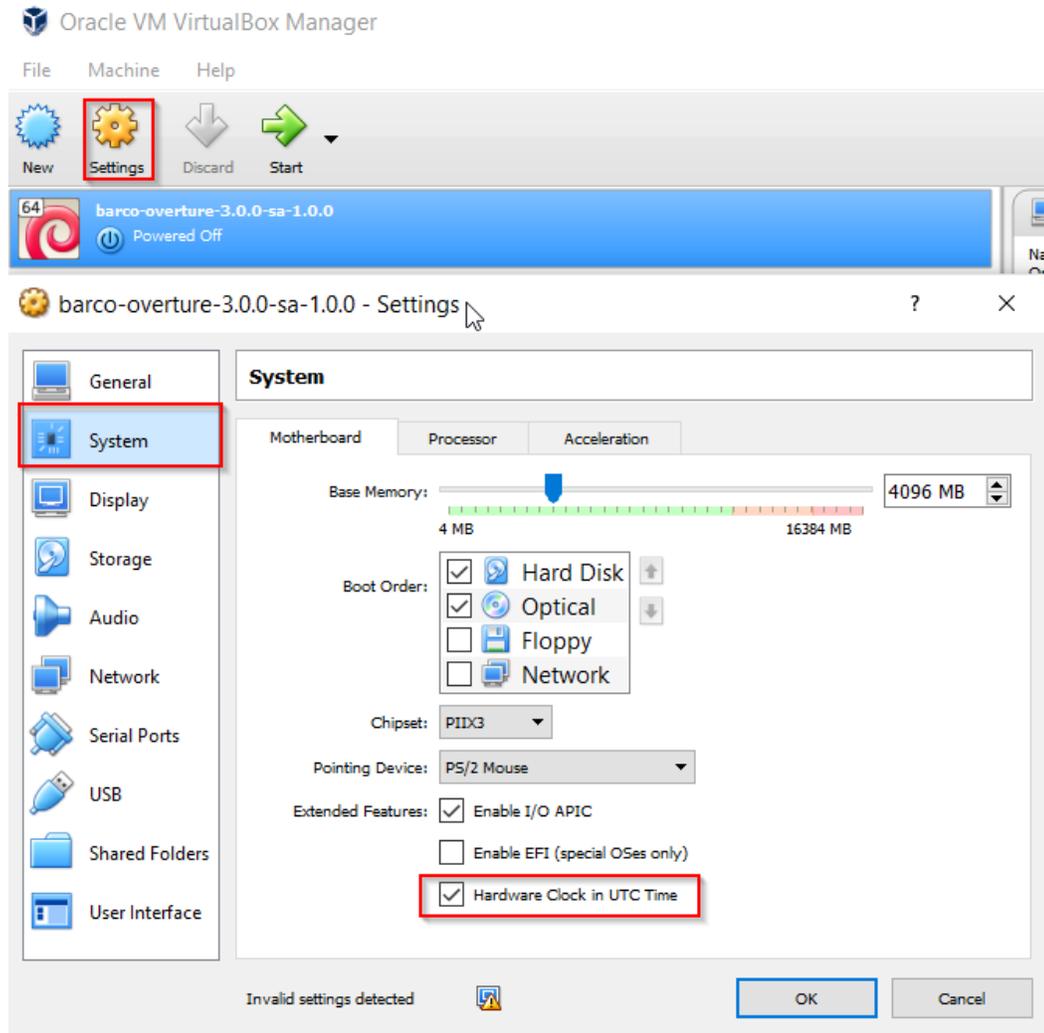
Location: C:\Users\Medialon\VirtualB...

Attached to: florian-build4 (Fresh install)

Encrypted with key: --

4.1.4.1 Oracle VirtualBox Hardware Clock Synchronization

1. Select your VM from the list.
2. Click Settings > System
3. Tick the check box "Hardware Clock in UTC time."
4. Click OK. * See how to set VM Timezone Section.

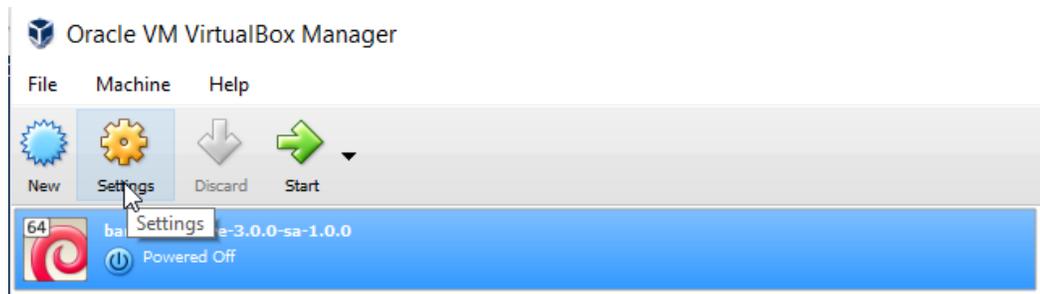


4.1.4.2 Oracle VirtualBox Network Configuration

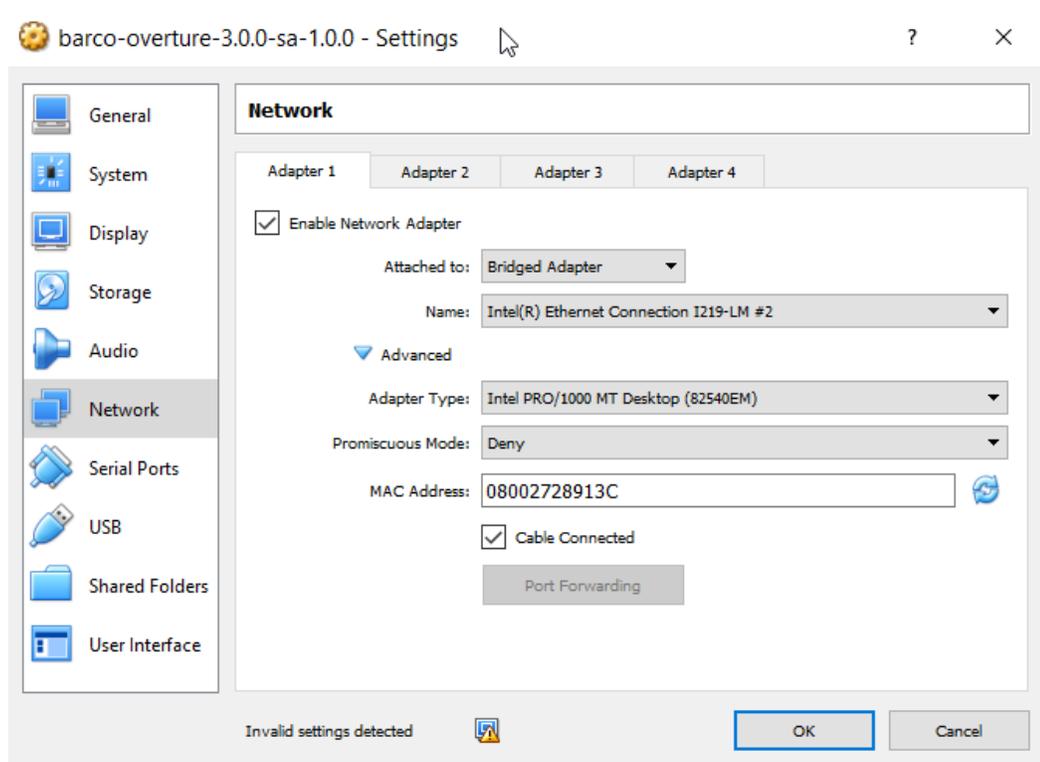
To access Overture, the VM must be connected to a network. You need to setup one of the following networks. You can setup more than one, if you need.

Bridged Network

1. Select the Overture VM and click Settings



2. Click Network and Select Bridged Adapter (select which interface card to use)

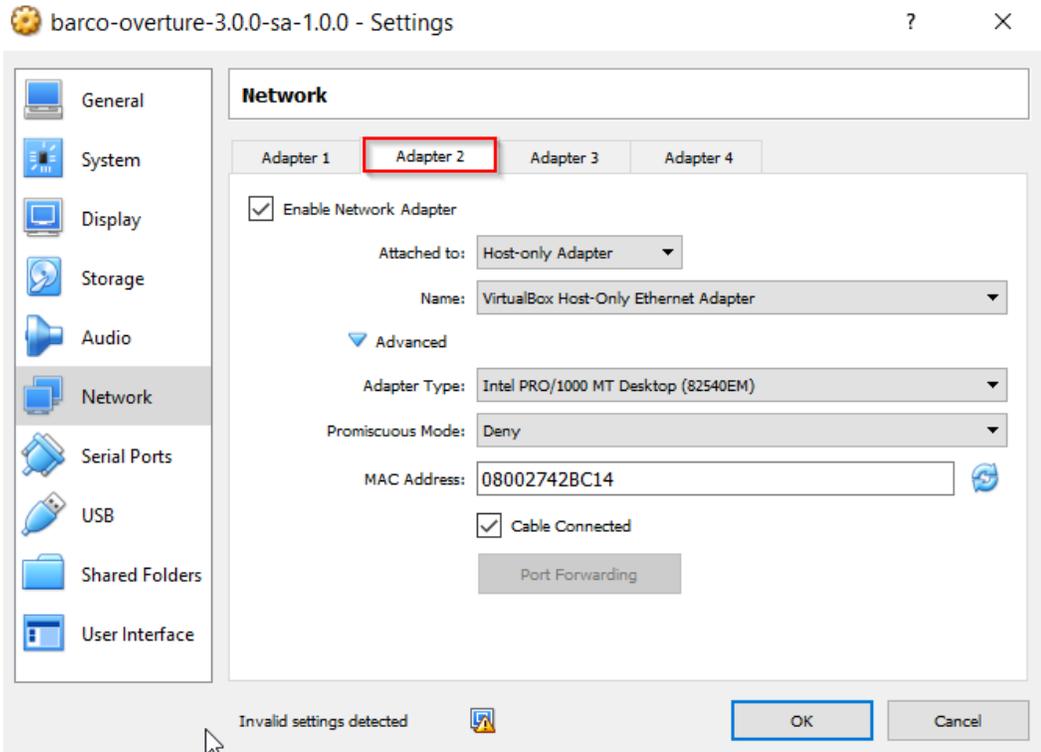


3. Optional step - Click on Refresh next to MAC Address. This will generate a new MAC for this machine. You should do it if you plan to run more than one VM on a network.



Host-Only Network

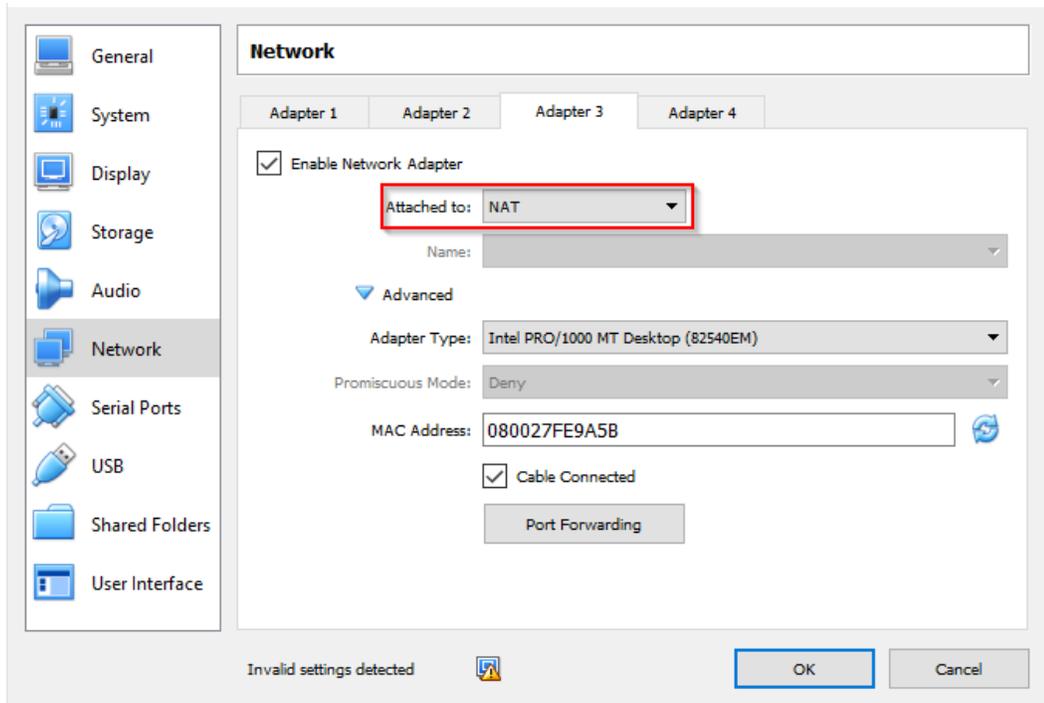
1. Click Adapter 2 if you want to add another Network Card. The VM will be available on multiple networks.

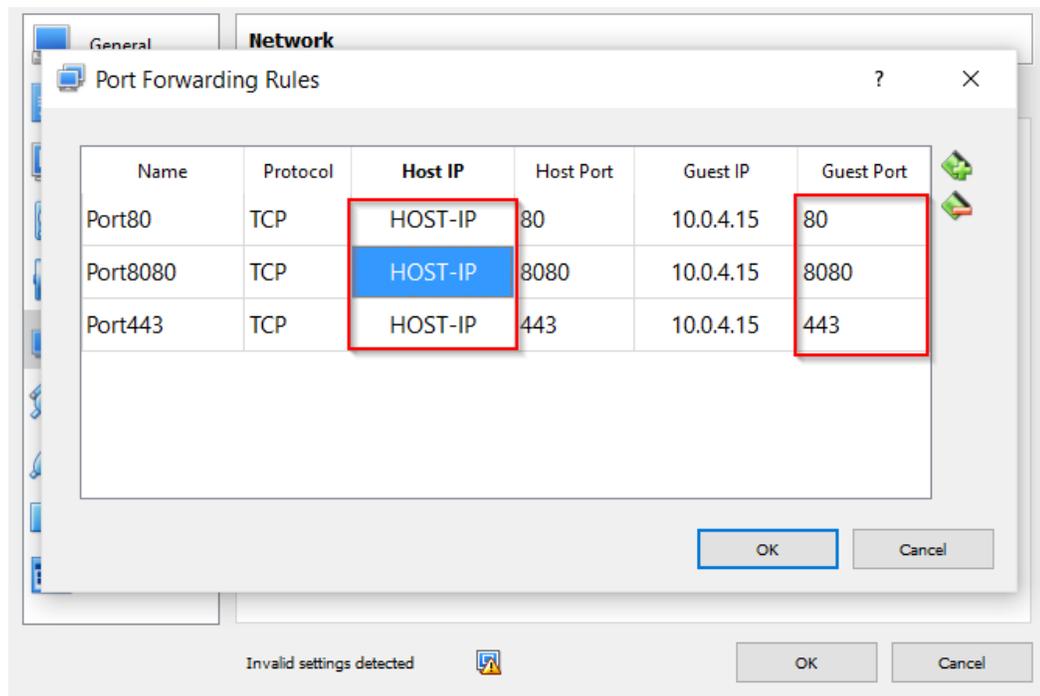


2. Click OK

NAT Network

1. Select "NAT" in Attached to drop-down:

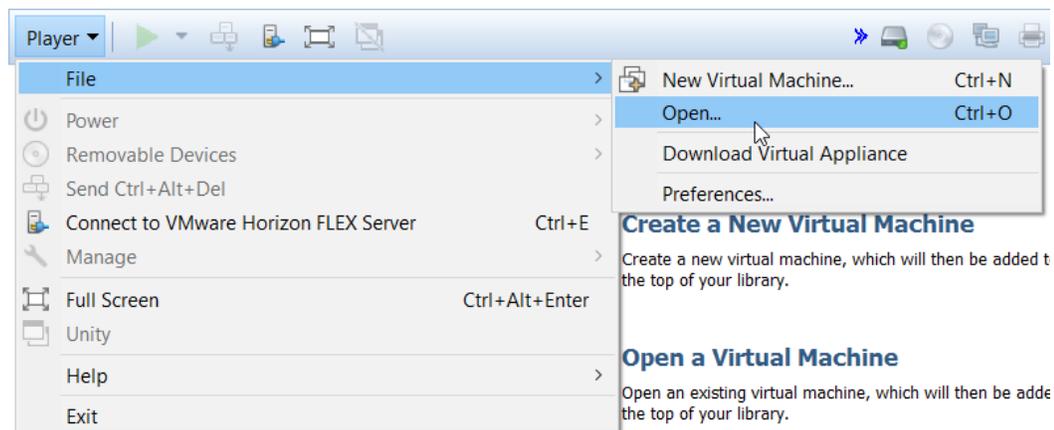




8. Click OK and OK.

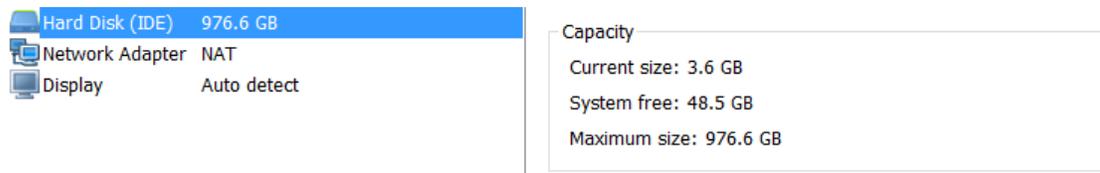
4.1.5 VMware Player Installation

1. Open "VMWare Workstation 12 Player"
2. Click Player
3. Click File and Open



4. Select Overture.ova and click Open
5. Click Import

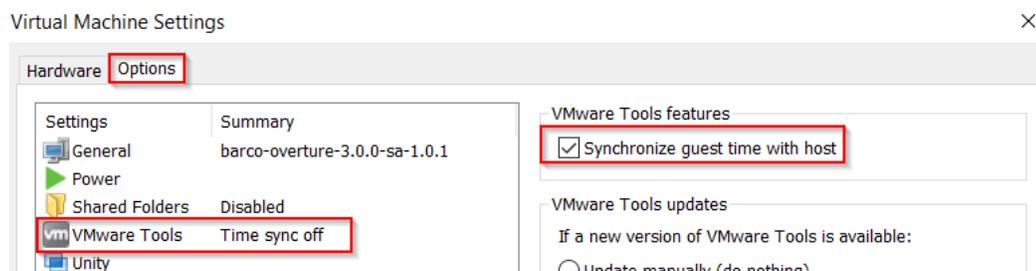
Note: You will notice that the storage space is almost 300Gb.



This is an estimated size and is not provisioned in advance. The VM is allowed to potentially grow up to 300Gb.

4.1.5.1 VMware Player/Workstation

1. Select your VM from the list.
2. Click "Edit virtual machine settings"
3. Click "Options" > VMware Tools
4. Tick the checkbox "Synchronize guest time with host"
5. Click OK.



4.1.5.2 VMware Player and VMware Workstation Pro Network Adapter Configuration

1. Click on "Edit virtual machine settings"



barco-overture-3.0.0-sa-1.0.1

State: Powered Off

OS: Other

Version: Workstation 6.5-7.x virtual machine

RAM: 4 GB

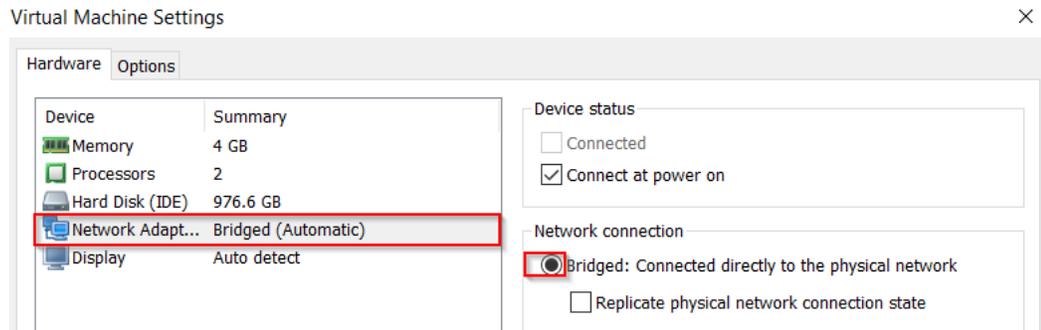
 Play virtual machine

 Edit virtual machine settings

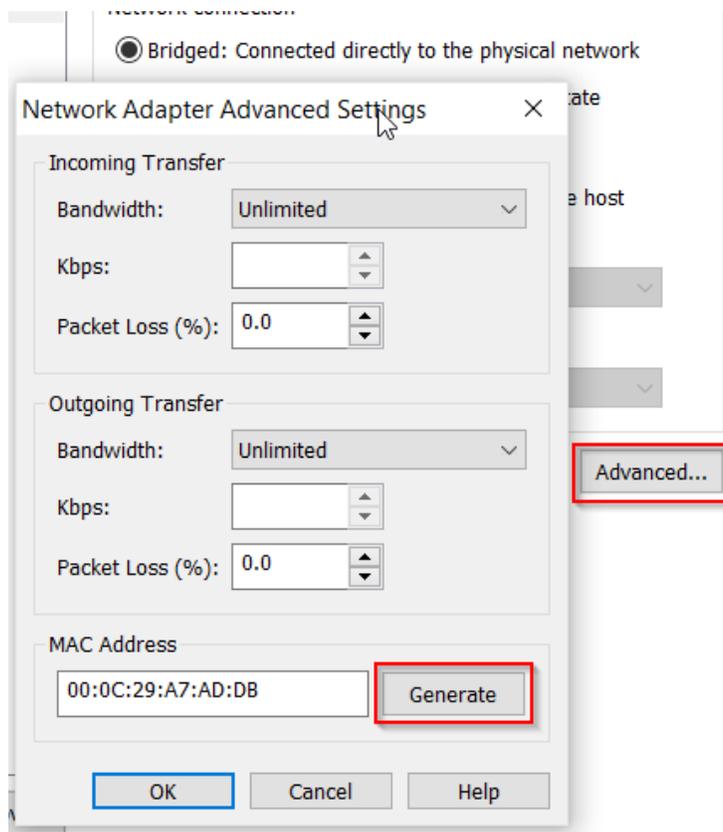
2. Click on "Network Adapter"

Bridged Network

1. Select "Bridged: Connected directly to the physical network"



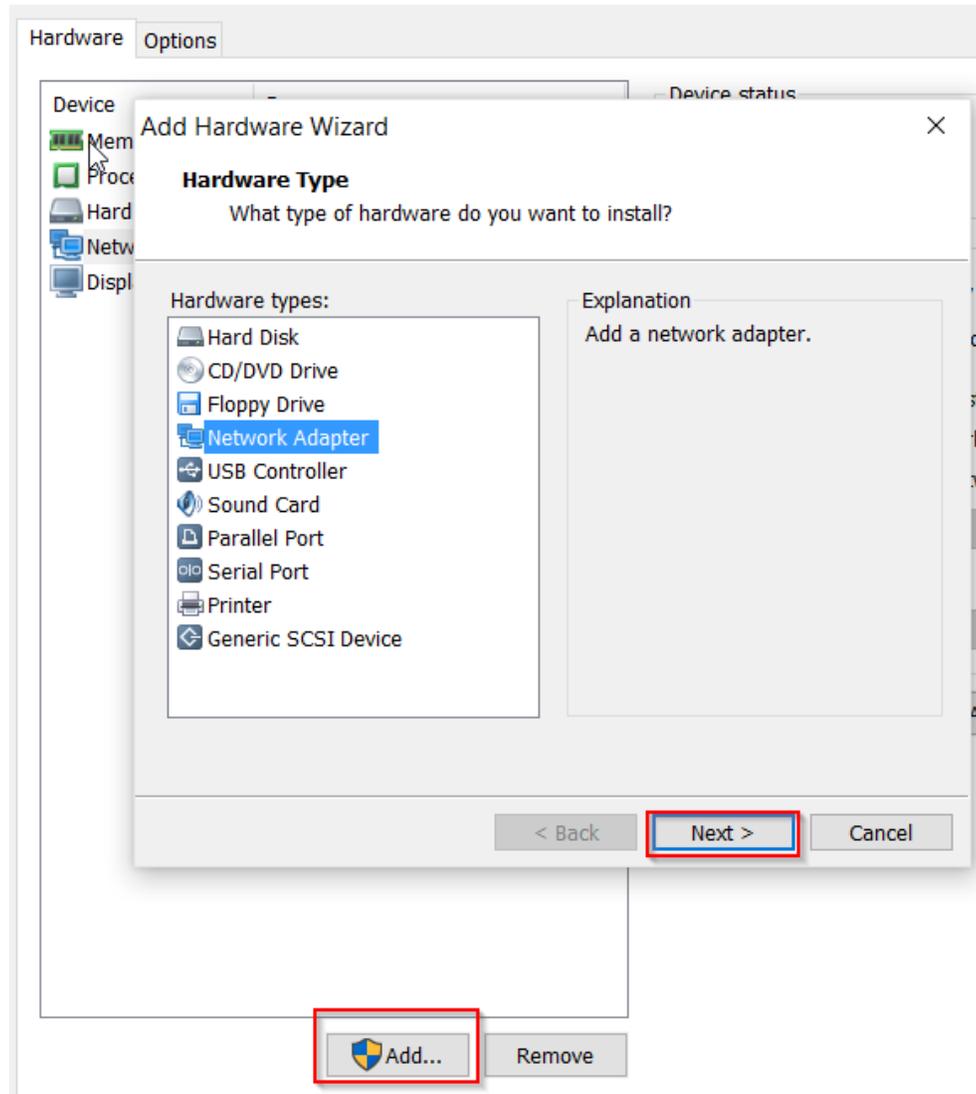
2. Optional Step - Click advance and regenerate Mac Address. You need this if you plan on hosting more than one VM on a network.



Host-Only Network

1. Add another network adapter if you want to make this VM available on more than one network.

Virtual Machine Settings



2. Select Host-Only

Network Adapter Type

What type of network adapter do you want to add?

Network connection

Bridged: Connected directly to the physical network

Replicate physical network connection state

NAT: Used to share the host's IP address

Host-only: A private network shared with the host

Device status

Connect at power on

< Back Finish Cancel

3. Click Finish.
4. Boot up the VM and note the new IP. This IP is only accessible from the host

NAT Network

1. Add a new Network Adapter.
2. Select NAT

Network Adapter Type

What type of network adapter do you want to add?

Network connection

Bridged: Connected directly to the physical network

Replicate physical network connection state

NAT: Used to share the host's IP address

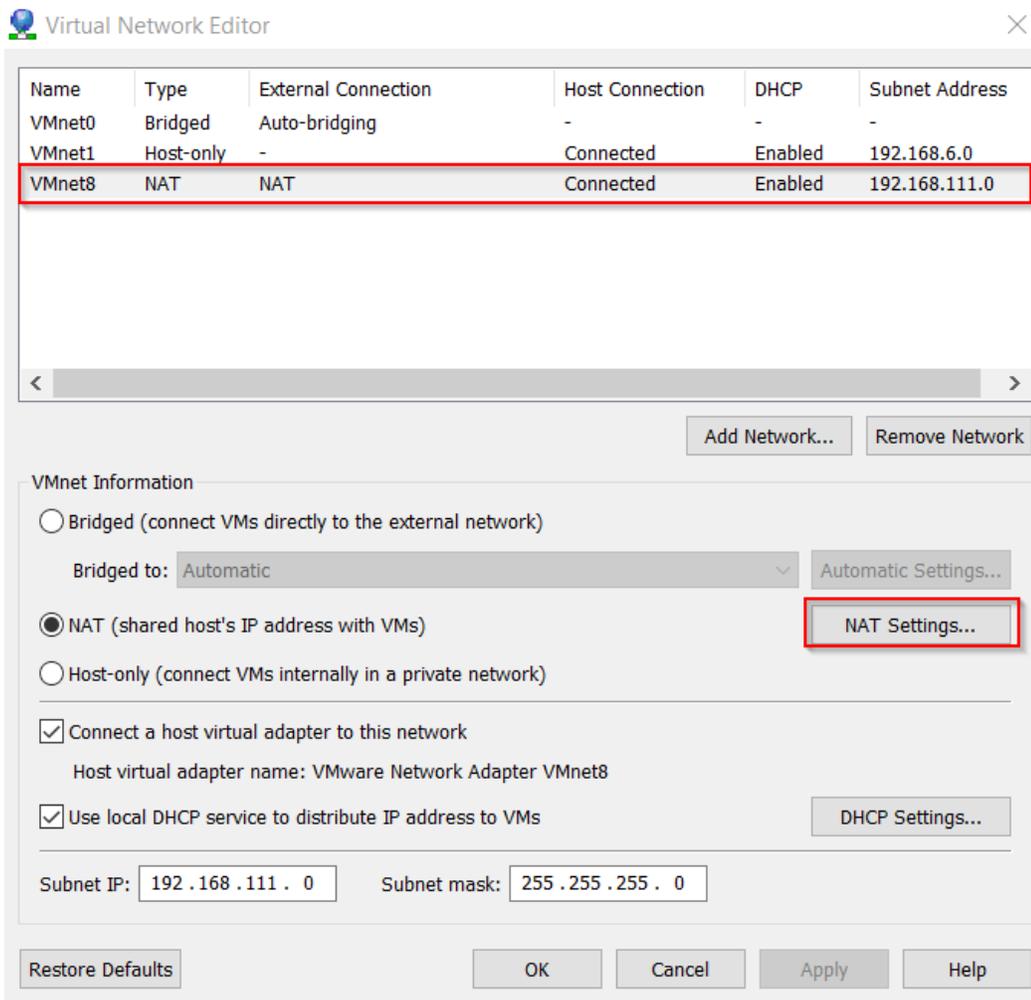
Host-only: A private network shared with the host

Device status

Connect at power on

< Back Finish Cancel

3. Click Finish
4. Boot Up the VM
5. Note the IP address.
6. Open VMware Virtual Network Editor. On Windows you must be Administrator. Start> Programs> VMware > Virtual Network Editor
7. Select NAT Settings



8. Create three ports that will map from host to VM. Ports 80, 8080 and 443. Make sure other applications don't use these ports on the HOST. Otherwise, you can replace them with other ports of your choice.

NAT Settings ×

Network: vmnet8
 Subnet IP: 192.168.111.0
 Subnet mask: 255.255.255.0
 Gateway IP:

Port Forwarding

Host P...	Type	Virtual Machine IP Addr...	Description
80	TCP	192.168.111.131:80	Port80
8080	TCP	192.168.111.131:8080	Port8080
443	TCP	192.168.111.131:443	Port443

Advanced

Allow active FTP
 Allow any Organizationally Unique Identifier

UDP timeout (in seconds):

Config port:

Enable IPv6

IPv6 prefix:

9. Click OK and Apply. NAT settings will be reloaded.
10. Use your Host IP:80, HOST IP:8080 in a browser to access Overture and its Control Server.

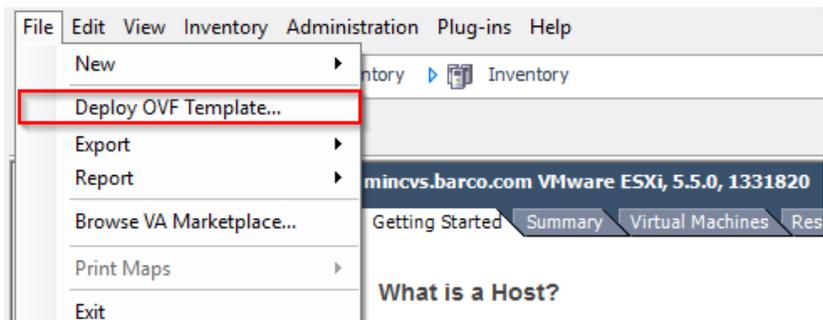
4.1.6 VMware or Virtual Box Tools

It is possible you will see a message asking you to install VMware/Vbox Tools. Please ignore it, the virtual machine does not need them.

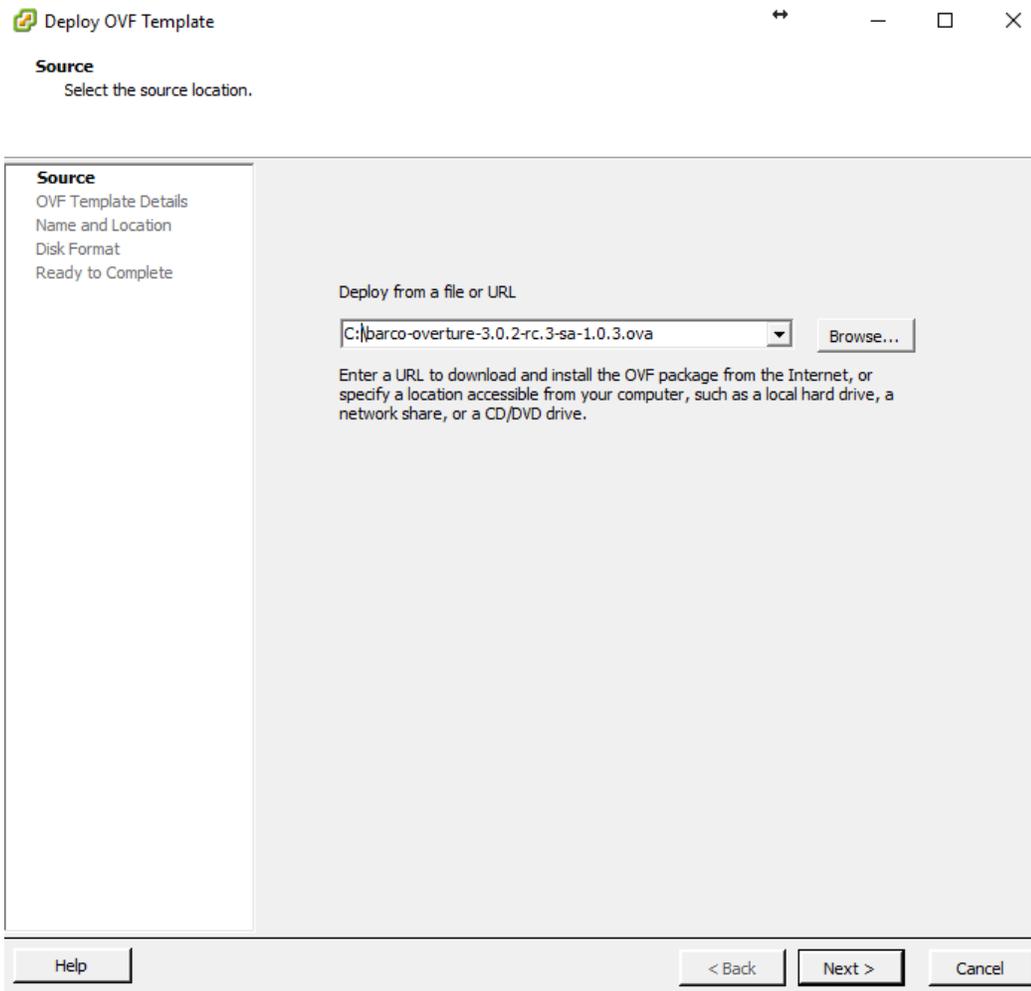
4.1.7 VMware vSphere Installation

The exact steps could be different from version to version. Please refer to VMware documentation. The following steps are for vSphere v 5.5.0

1. Login in vSphere Client
2. Click File
3. Click Deploy OVF Template ...



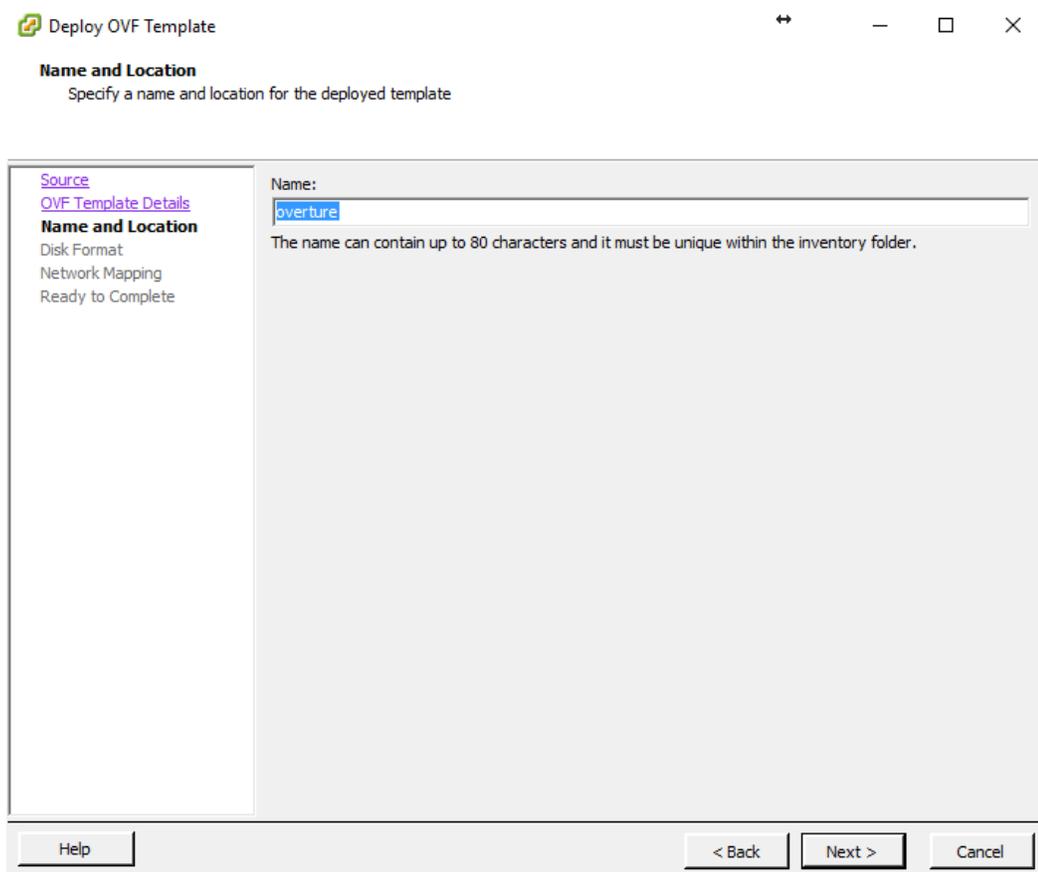
4. Select the ova file



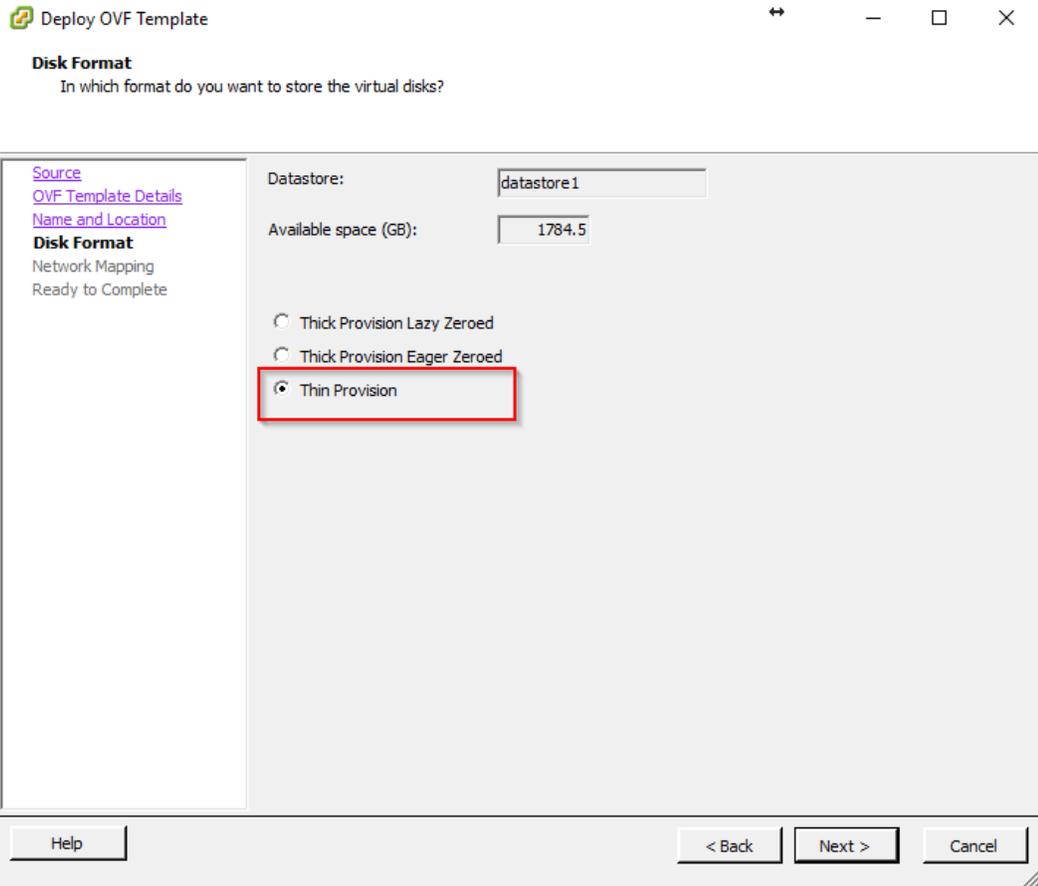
5. If you're prompted with a warning about the operating system is not supported. Click Yes.



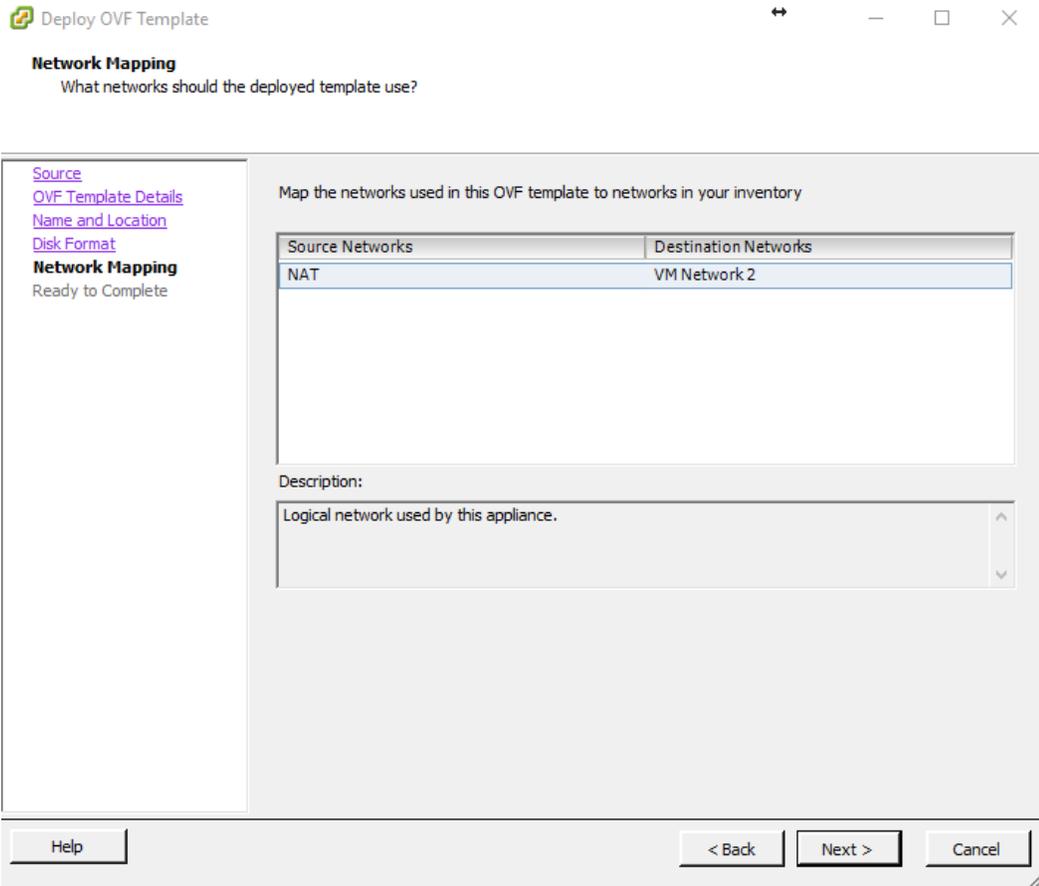
- Optional - Rename VM Name, Otherwise click Next



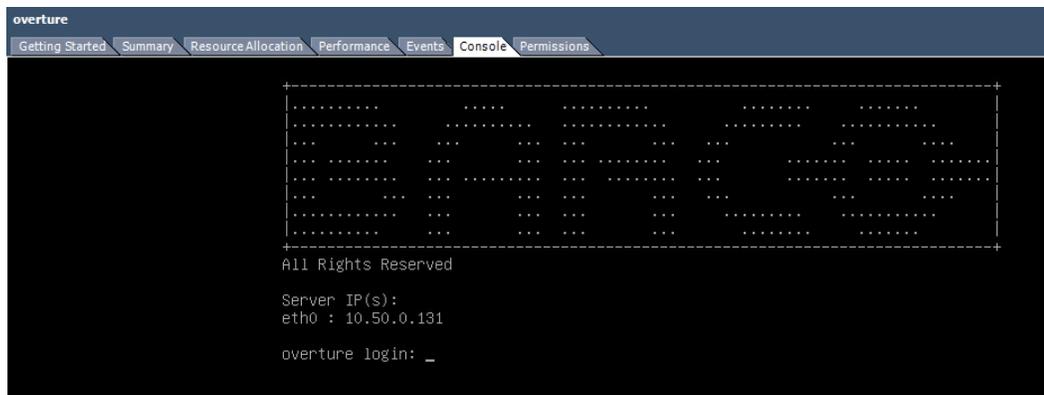
- Click Thin Provision. This will not pre-allocate disk space upfront. Instead the VM will be allowed to grow as needed.



8. Depending on your VMware vSphere configuration, choose the network mapping that allows VM to access Internet and be accessed by your users. The following screenshot is just an example.



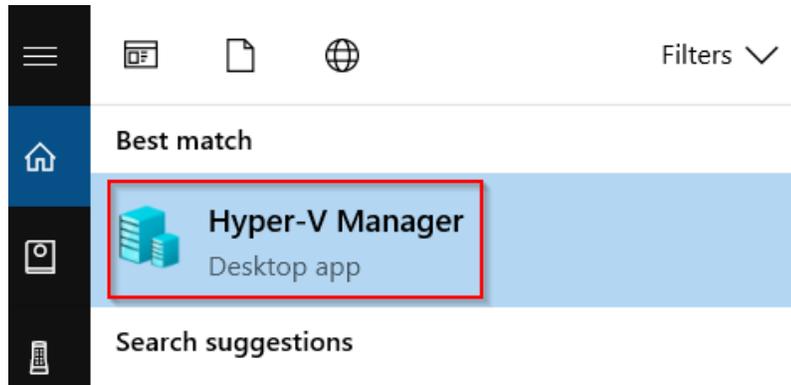
9. Verify the settings and click Finish. The deployment will take a few minutes depending on your network speed.
10. Power On the VM.
11. Click on the VM.
12. Click on Console.
13. Note the IP.



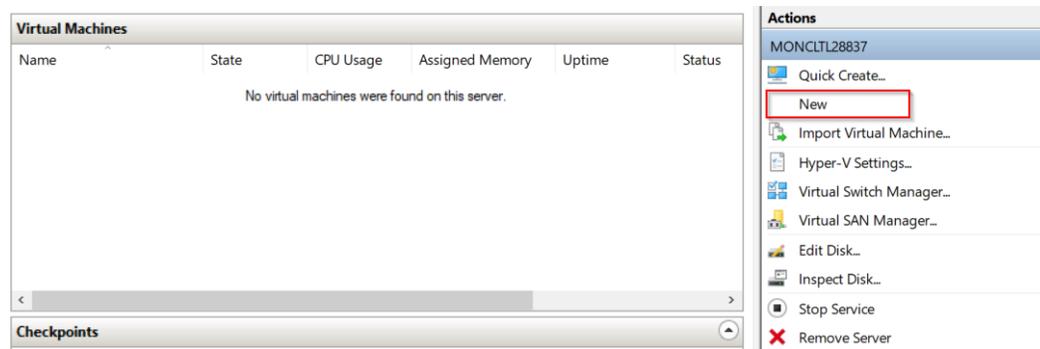
14. If you need to perform static IP setup, follow the instructions from VM Advanced Settings.

4.1.8 Microsoft Hyper-V Installation

1. Download Hyper-V compatible file from Barco Website.
2. Unzip the file and extract overture.vhd file.
3. Open Hyper-V Manager



4. Connect to Server... (if needed) and select it.
5. Click 'New' and choose 'Virtual Machine'



6. Click 'Next' until 'Specify Name and Location'. Enter user friendly name (i.e Overture). Click Next

Specify Name and Location

Before You Begin

- Specify Name and Location
- Specify Generation
- Assign Memory
- Configure Networking
- Connect Virtual Hard Disk
- Installation Options
- Summary

Choose a name and location for this virtual machine.

The name is displayed in Hyper-V Manager. We recommend that you use a name that helps you easily identify this virtual machine, such as the name of the guest operating system or workload.

Name:

You can create a folder or use an existing folder to store the virtual machine. If you don't select a folder, the virtual machine is stored in the default folder configured for this server.

Store the virtual machine in a different location

Location:

! If you plan to take checkpoints of this virtual machine, select a location that has enough free space. Checkpoints include virtual machine data and may require a large amount of space.

< Previous **Next >** Finish Cancel

7. In 'Specify Generation' choose 'Generation 1' and click 'Next'

Specify Generation

Before You Begin

- Specify Name and Location
- Specify Generation
- Assign Memory
- Configure Networking
- Connect Virtual Hard Disk
- Installation Options
- Summary

Choose the generation of this virtual machine.

Generation 1

This virtual machine generation supports 32-bit and 64-bit guest operating systems and provides virtual hardware which has been available in all previous versions of Hyper-V.

Generation 2

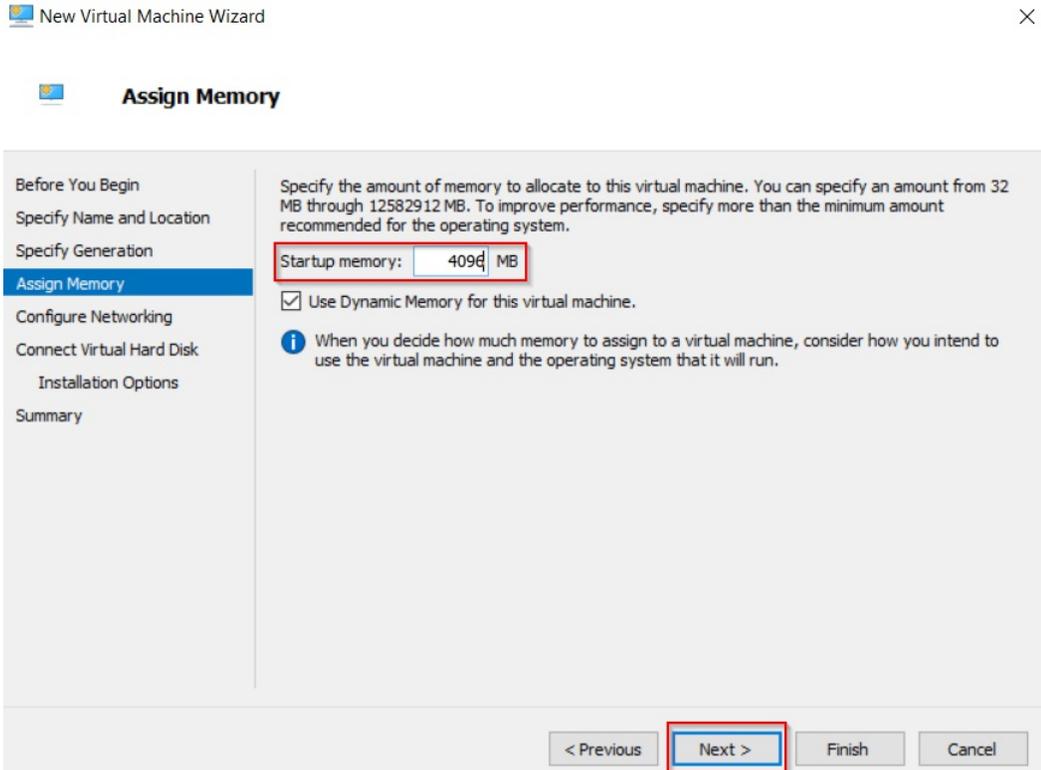
This virtual machine generation provides support for newer virtualization features, has UEFI-based firmware, and requires a supported 64-bit guest operating system.

! Once a virtual machine has been created, you cannot change its generation.

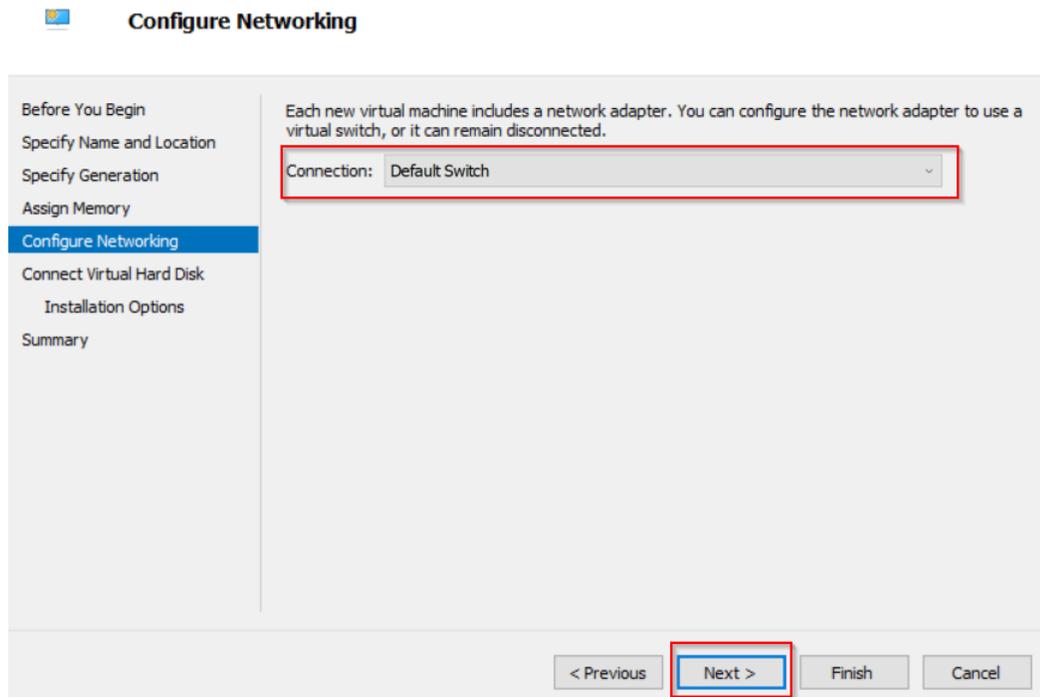
[More about virtual machine generation support](#)

< Previous **Next >** Finish Cancel

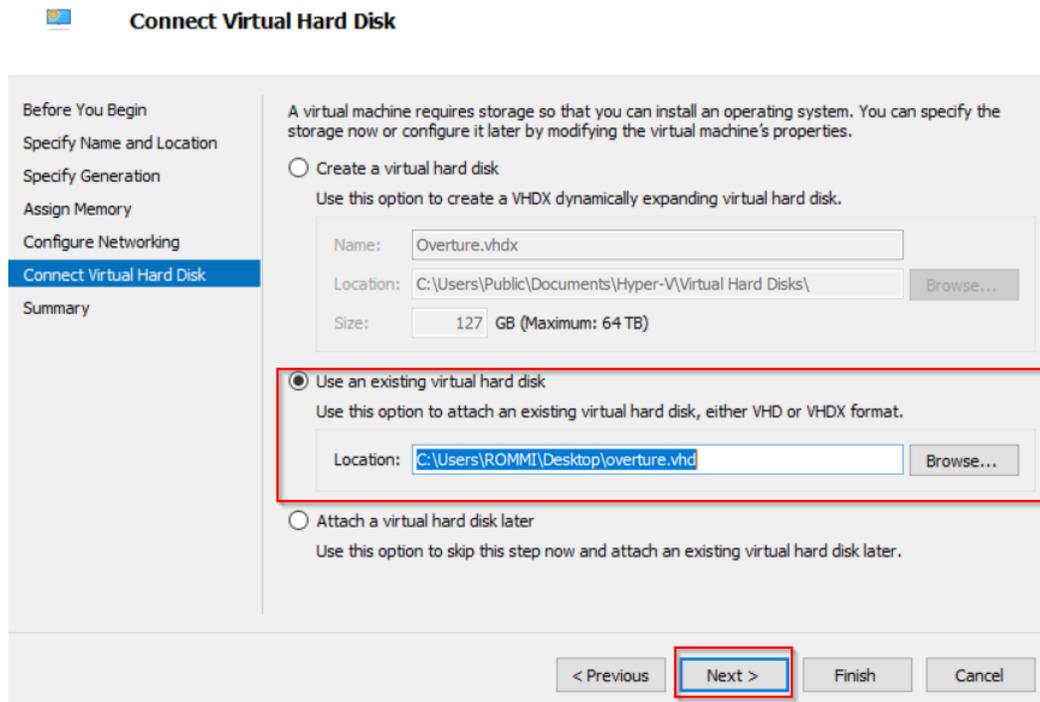
- In 'Assign Memory' enter '4096'. You can add more RAM if your system allows it.



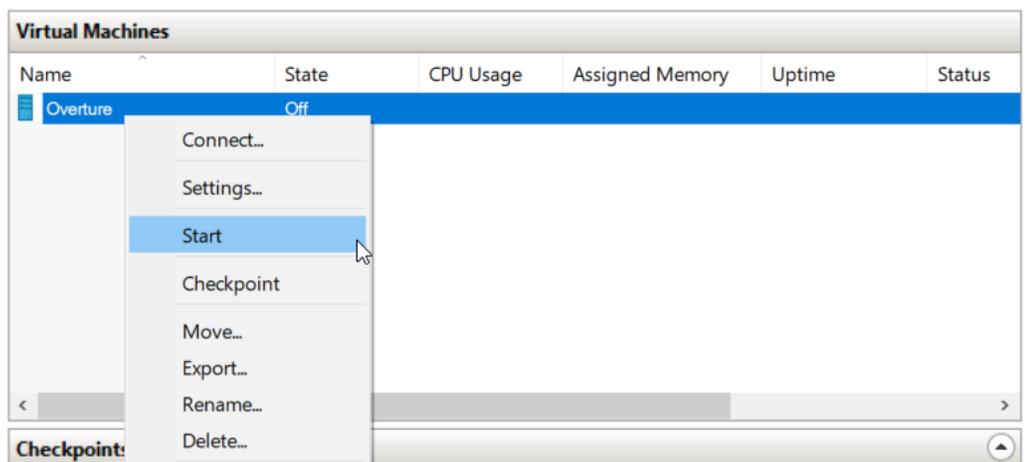
- In 'Configure Networking' choose your Networking Configuration. You may need to refer to your IT specifications to finish this step. The screenshot below is an example only.



10. In 'Connect Virtual Hard Disk' choose 'Use an existing virtual hard disk' and select the VHD file from step 2. Click next.

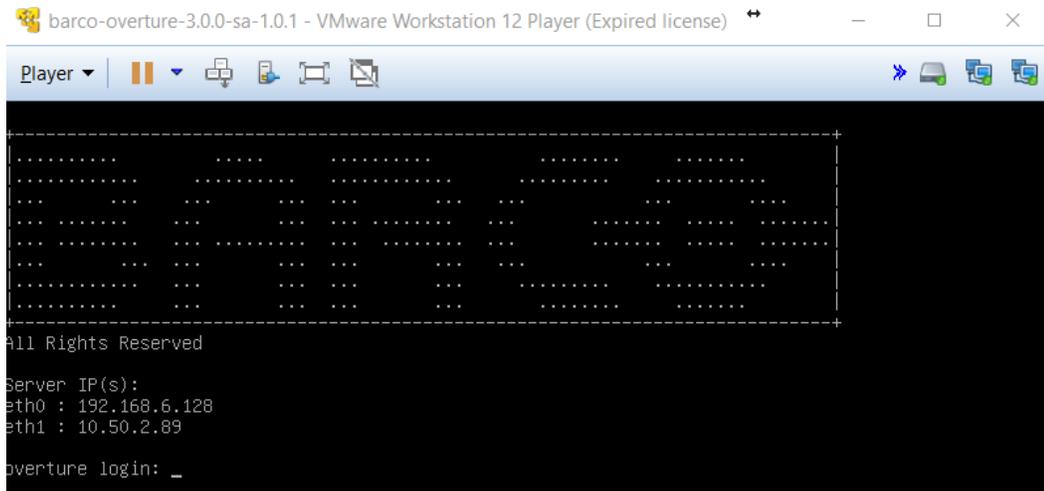


- When the VM has been imported, start it.



4.1.9 VM Advanced Settings

- Boot up the VM



2. Login: administrator
3. Password: Adm1n1str@t0r!



4.1.9.1 Alternative access to Administrator's menu

Another option to access the administrator's menu is via an SSH client (i.e. powershell, PuTty, or default clients on Linux, MacOS). Below are the steps for PowerShell, but they will be almost the same in other clients. This option is preferable for users with non-English keyboard layouts.

1. Boot up the VM
2. Open your ssh client
3. type 'ssh administrator@your-vm-ip'
4. Accept authenticity of host, if you are prompted.
5. Press Enter when asked "Enter passphrase for key"
6. Enter administrator's password. Adm1n1str@t0r!
7. You will be presented with the same menu as in the previous section.

4.1.9.2 Name servers configurations

This option allows you to set or change DNS servers.

If you have DHCP setup, the nameservers will be automatically populated.

4.1.9.3 Network Configuration

Here you can check the VM Network configuration or change them. Follow the instructions.

```
Network configuration
=====
Number of interfaces: 6

Index   Name      Address          Netmask          Gateway          Remarks
0       br-e6ee2f9e2ddb 172.18.0.1       255.255.0.0     10.50.0.1
1       docker0 172.17.0.1       255.255.0.0     10.50.0.1       *
2       eth0     10.50.0.191     255.255.254.0   10.50.0.1
3       veth237c5b8 -.-.-.-         -.-.-.-         -.-.-.-         *
4       veth6ff9104 -.-.-.-         -.-.-.-         -.-.-.-         *
5       vetha40077c -.-.-.-         -.-.-.-         -.-.-.-         *

Remarks:
docker0: Network cable unplugged.
veth237c5b8: System is unable to retrieve configuration using DHCP.
veth6ff9104: System is unable to retrieve configuration using DHCP.
vetha40077c: System is unable to retrieve configuration using DHCP.

Please check your network infrastructure.

1: Change configuration
q: Return
```

4.1.9.4 Hostname

Change the VM Network name here.

4.1.9.5 Time Zone

Default (ETC/UTC). You should change the TimeZone to where your VM is located.

4.1.9.6 Password

It is recommended to change administrator's password.

4.1.9.7 Reboot

Reboots the VM

4.1.9.8 Shutdown

Shut downs the VM.



4.1.9.9 Docker

If your networks reserve 172.17.0.1 and 172.18.0.1, you will need to change VM's Docker network to avoid conflicts. This section allows you to do the configuration.

```
Docker
=====
The current Docker0 Bridge is configured as
      inet addr:172.17.0.1 Bcast:0.0.0.0 Mask:255.255.255.0
      inet6 addr: fe80::42:bff:fe3b:3c9a/64 Scope:Link

1: Change Docker subnet
q: Return

>_
```

1. Click 1: Change Docker subnet
2. Enter new Docker0 IP. It is important to select a subnet that will not conflict with any of your existing subnets.
3. Enter Administrator's Password.

```
Docker
=====
The current Docker0 Bridge is configured as
      inet addr:172.17.0.1 Bcast:0.0.0.0 Mask:255.255.255.0
      inet6 addr: fe80::42:bff:fe3b:3c9a/64 Scope:Link

1: Change Docker subnet
q: Return

>1
Enter new Docker0 IP (ex: 192.168.1.5)
Bridge IP: 155.155.0.1
New IP: 155.155.0.1
[sudo] password for administrator: _
```

4. You will see the new configuration after a short delay.

```
Docker
=====
The current Docker0 Bridge is configured as
      inet addr:155.155.0.1 Bcast:0.0.0.0 Mask:255.255.255.0
      inet6 addr: fe80::42:bff:fe3b:3c9a/64 Scope:Link

1: Change Docker subnet
q: Return

>_
```

5. Reboot the VM.

4.1.9.10 Proxy

This section explains how to configure the Virtual Appliance to connect to a HTTP Proxy.

1. Log on as Administrator inside the VM console.



```
Admin console
=====
1: Name servers configuration
2: Network configuration
3: Host name
4: Domain name
5: Time zone
6: Password
7: Reboot
8: Shutdown
9: Docker
10: Proxy
q: Quit

>_
```

2. Type **10** for Proxy.

Set HTTP Proxy

1. Type **2** to Set HTTP Proxy.

```
HTTP Proxy
=====
HTTP Proxy: None

HTTPS Proxy: None

1: Clear Proxy
2: Set HTTP Proxy
q: Quit

>2_
```

2. You will be asked to login as `administrator`.
3. Enter `http_proxy` and `https_proxy` addresses.

```
HTTP Proxy
=====
HTTP Proxy: None

HTTPS Proxy: None

1: Clear Proxy
2: Set HTTP Proxy
q: Quit

>2
[sudo] password for administrator:
> http_proxy: http://10.50.0.137:3128
Changed http_proxy to http://10.50.0.137:3128
> https_proxy: https://10.50.0.137:3128_
```

4. Press **Enter**. Notice the changes.



```
HTTP Proxy
=====
HTTP Proxy: http://10.50.0.137:3128
HTTPS Proxy: https://10.50.0.137:3128

1: Clear Proxy
2: Set HTTP Proxy
q: Quit

> _
```

5. Click **q** to Quit.
6. Click **7** to Reboot. After the reboot, both the UXServer and Control Server will have their http traffic redirected to your proxy.

Note: If your proxy is user and password protected. The format is : [http://\[USERNAME\]:\[PASSWORD\]@\[PROXYIP\]:\[PROXYPORT\]](http://[USERNAME]:[PASSWORD]@[PROXYIP]:[PROXYPORT])

4.1.9.11 NTP Configuration

This option allows you to configure the NTP client by adding, editing or deleting NTP servers.

1. Log in as Administrator inside the VM console.

```
Admin console
=====
1: Name servers configuration
2: Network configuration
3: Host name
4: Domain name
5: Time zone
6: Password
7: Reboot
8: Shutdown
9: Docker
10: Proxy
11: NTP Configuration
q: Quit

>
```

2. Type '11' for NTP Configuration. You will see current NTP server names in the list. Please note by default Debian NTP server-names are set for the VM Machine.

```
NTP configuration
=====
Number of servers: 4

Index  Server name
0      0.debian.pool.ntp.org
1      1.debian.pool.ntp.org
2      2.debian.pool.ntp.org
3      3.debian.pool.ntp.org

1: Add entry
2: Replace entry
3: Remove entry
q: Return

>_
```

Add a new NTP server

1. Type '1'
2. Put your NTP server-name and hit 'Enter'.

```
NTP configuration
=====
Number of servers: 4

Index  Server name
0      0.debian.pool.ntp.org
1      1.debian.pool.ntp.org
2      2.debian.pool.ntp.org
3      3.debian.pool.ntp.org

1: Add entry
2: Replace entry
3: Remove entry
q: Return

>1
Enter server name: ntp.yourcompany.com
```

3. The new entry will be added to the list.



```
NTP configuration
=====
Number of servers: 5

Index  Server name
0      0.debian.pool.ntp.org
1      1.debian.pool.ntp.org
2      2.debian.pool.ntp.org
3      3.debian.pool.ntp.org
4      ntp.yourcompany.com

1: Add entry
2: Replace entry
3: Remove entry
q: Return

>
```

Edit (replace) a NTP server

1. Type '2'
2. Enter the index number of server-name you want to edit.

```
NTP configuration
=====
Number of servers: 5

Index  Server name
0      0.debian.pool.ntp.org
1      1.debian.pool.ntp.org
2      2.debian.pool.ntp.org
3      3.debian.pool.ntp.org
4      ntp.yourcompany.com

1: Add entry
2: Replace entry
3: Remove entry
q: Return

>2
Enter index: 4
Enter server name: newntp.yourcompany.com_
```

3. Put the new NTP server-name and hit 'Enter'.



```
NTP configuration
=====
Number of servers: 5

Index  Server name
0      0.debian.pool.ntp.org
1      1.debian.pool.ntp.org
2      2.debian.pool.ntp.org
3      3.debian.pool.ntp.org
4      newntp.yourcompany.com

1: Add entry
2: Replace entry
3: Remove entry
q: Return

>
```

Remove (delete) a NTP Server

1. Type '3'
2. Enter the index number of server-name you want to remove.

```
NTP configuration
=====
Number of servers: 5

Index  Server name
0      0.debian.pool.ntp.org
1      1.debian.pool.ntp.org
2      2.debian.pool.ntp.org
3      3.debian.pool.ntp.org
4      newntp.yourcompany.com

1: Add entry
2: Replace entry
3: Remove entry
q: Return

>3
Enter index: 4_
```

3. Hit Enter. The item will be removed from the list.



```

NTP configuration
=====
Number of servers: 4

Index  Server name
0      0.debian.pool.ntp.org
1      1.debian.pool.ntp.org
2      2.debian.pool.ntp.org
3      3.debian.pool.ntp.org
-----
1: Add entry
2: Replace entry
3: Remove entry
q: Return
>_

```

4.1.9.12 Update Control Server link to Overture

Both an UXServer and a Control Server run inside the VM. If you elect to use VM's Control Server with Proxy, you will need to update the UXServer address in the Control Server or add <http://uxserver> to your DNS server. This will allow your Proxy to correctly route traffic between applications.

To update the address of the UXServer in the ControlServer shipped with the VM:

1. Open [http://\[VM_IP\]:8080](http://[VM_IP]:8080) in a browser.
2. Click [Edit](#), in [Server](#) tab.
3. Replace <http://uxserver> with [http://\[VM_IP\]](http://[VM_IP]) or [http://\[UXSERVER-DNS\]](http://[UXSERVER-DNS])
4. Click [Save](#).

After a few seconds, the Control Server will reconnect to the UXServer.

4.1.9.13 Quit

Logs out the administrator.

4.1.10 VM Migration

This section explains how to migrate virtual machine and Overture to a new version.

You would need to *migrate* your VM if you want to benefit from new Virtual Appliance version.

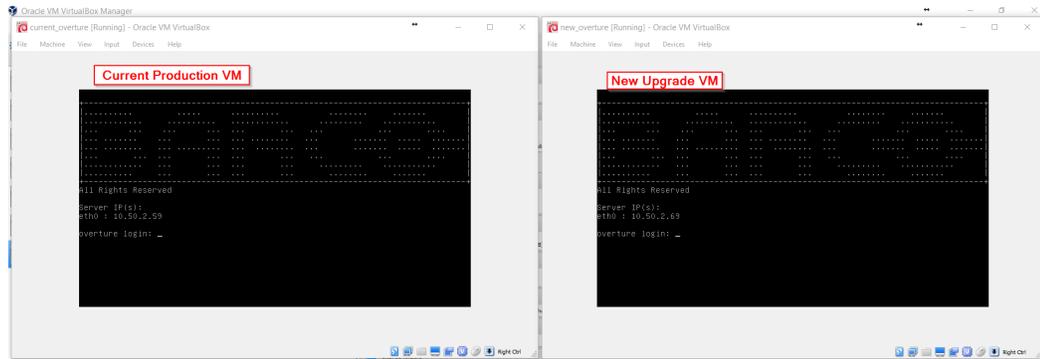
You cannot simply install a new VM and restore the backup of the previous one, because your license might be detected as an attempt of fraud (double usage).

So if you need to move from one VM to another your Overture installation, please follow the migration procedure.

Warning: Once the migration is initiated, the older version will become unavailable and cannot be used again. This is needed in order to prevent double license usage detection.

4.1.10.1 Generate Migration Package

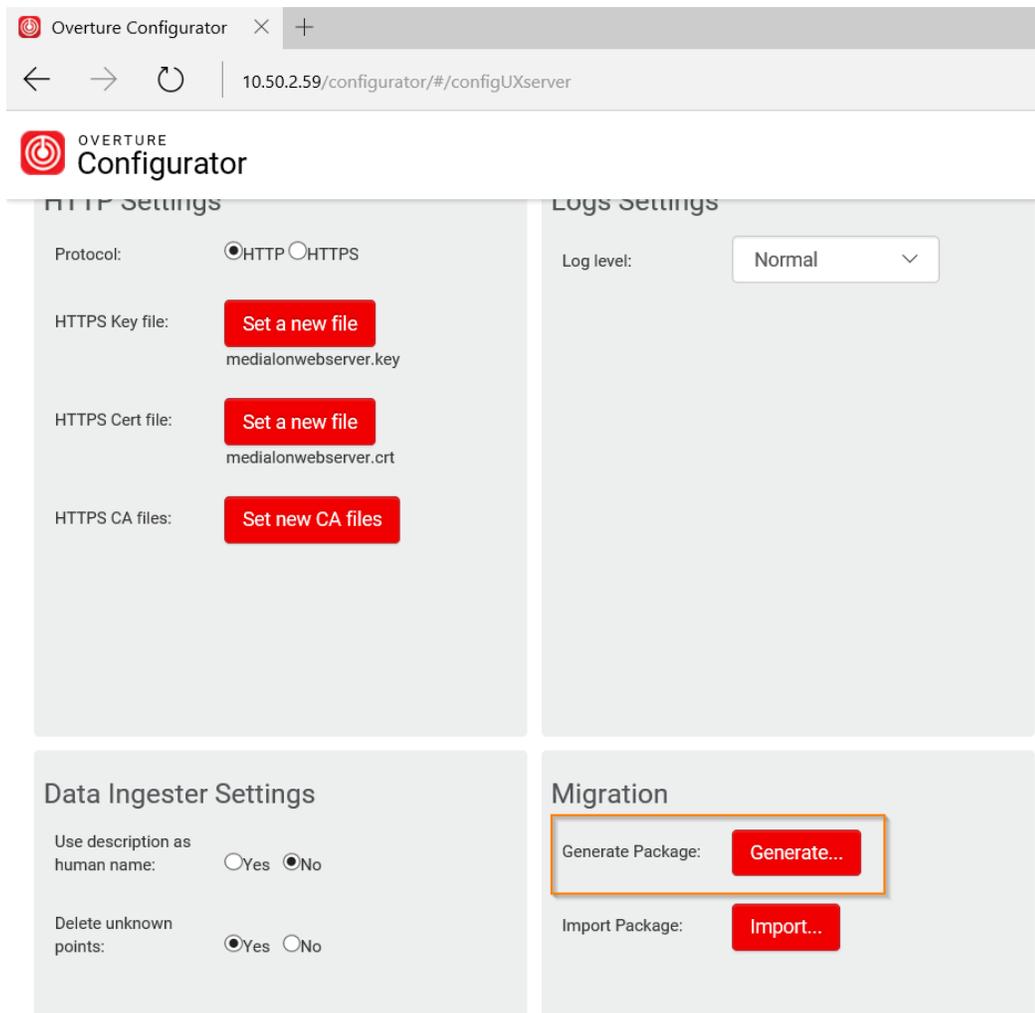
1. Download and deploy a new VM. Follow one of the installation guides from above. The below screenshot is just for the sake of an example, and could differ for you.



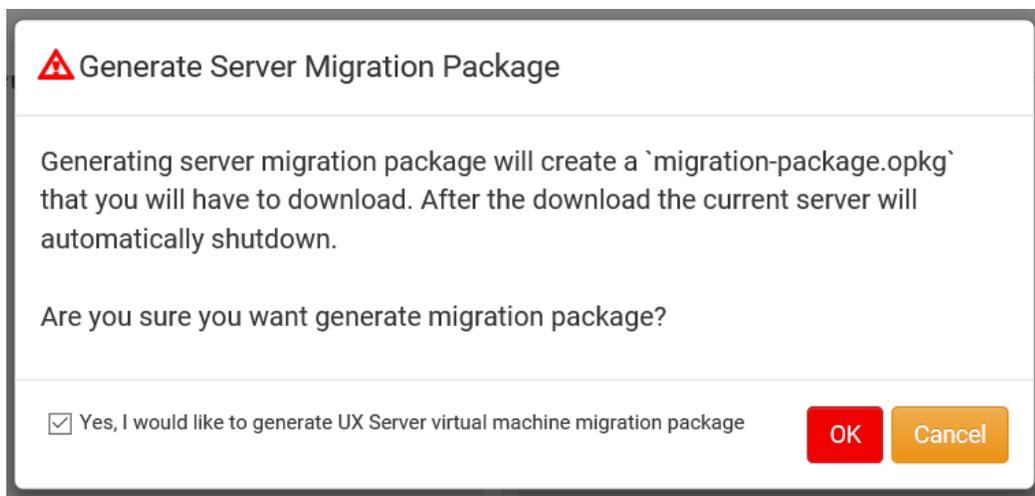
2. Login in Configurator with System:Admin privileges in your production VM. Back up your Overture. This step is optional and needed in case migration fails. (Configurator > Backups View > Backup up now > Download backup)



3. Go to Configurator > Views > Ux Server Config.
4. Click "Generate Package: Generate ..." button in the *Migration* section.



5. Read the migration message , accept the conditions and click 'OK'.



6. After a moment, the package will be ready to download.

Note: Once the download is completed, UX Server will self shutdown. Click "Migrate UX Server" when ready!



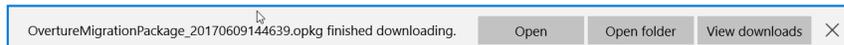
The UX Server migration package is ready to download. Click on migrate UX Server to download the migration package and shutdown the current UX Server Virtual Machine.



7. When Download Completed you're ready to import it in the new VM.



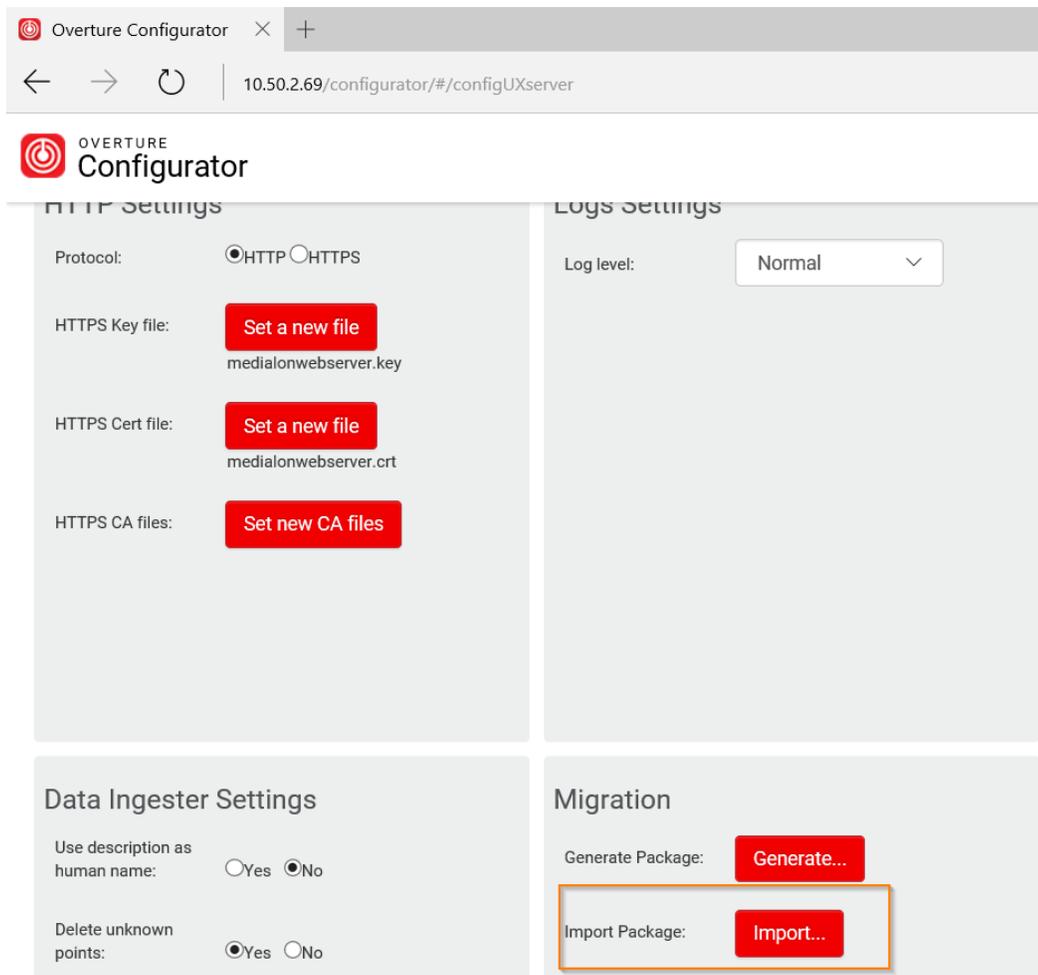
The UX Server migration package is now downloading. When the download completes the UX Server VM will shutdown. The download package will be in the Downloads directory of your browser and might be accessible at the bottom of this page.



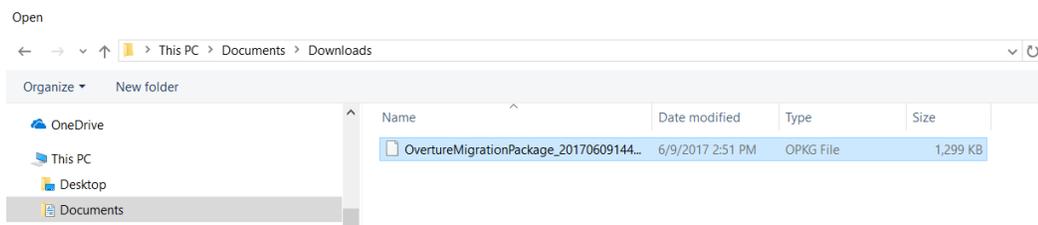
You can now shutdown and delete the current VM. It will not be usable anymore.

4.1.10.2 Import Migration Package

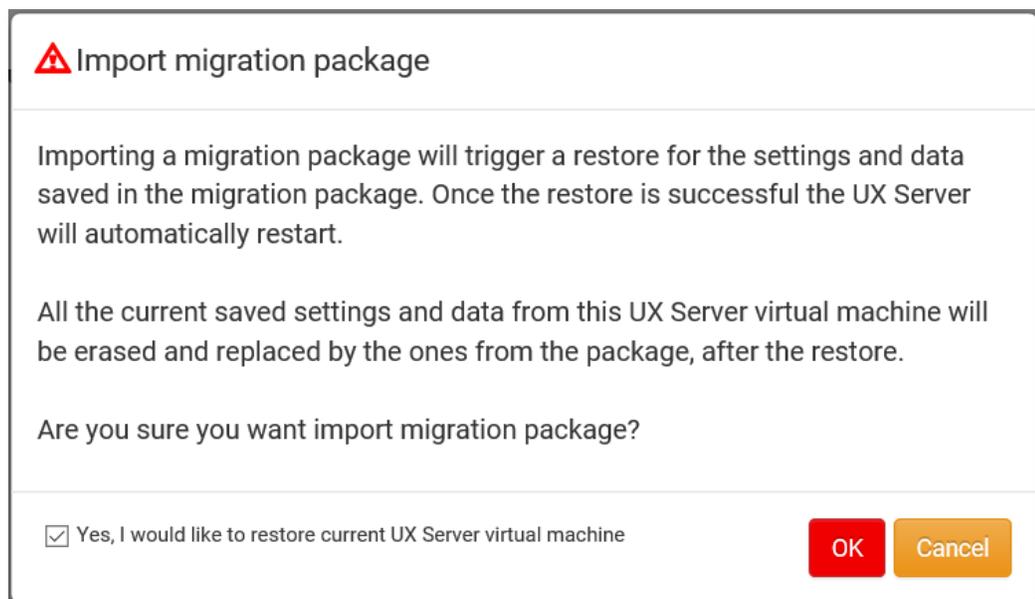
1. Login in Configurator with System:Admin privileges in your new VM.
2. Go to Configurator > Views > UXserver Config
3. Click on "Import Package: Import ..." button in *Migration* section.



4. Select the package you created in the previous guide.



5. Click "Open"
6. Read the instructions, and accept the conditions, and click "OK".



7. The Server will restart after a few minutes.
8. The migration is now completed.

The license has been restored with this migration backup, no need to re-import it.

Domain Name Resolution:

If you access your Overture by a domain name, you will need to update the DNS entry with a new IP. You can also, change the VM IP to what the same IP that the old VM had.

4.1.11 Starting The Machine

Once the virtual machine has been powered on, a boot screen will give you details on your machine IP. You will need this to access and further configure your Overture system.

4.2 Installing Separate Control Servers

Additional Control Servers can be installed on Windows PCs or run in a separate Docker installation.

4.2.1 Proxy

From version 1.8.0, the Control Server supports proxy. If the proxy is hosted in the distributed Virtual Machine, the proxy address is inherited from the ones set in the administration console.

- [Proxy Usage](#) dropdown is informing the CX whether or not to use the given proxy to communicate with the UX Server.
- [HTTP Proxy Settings](#) fields are providing the address, port and optional credentials of the proxy. The data is used if [Proxy Usage](#) is set to [Use Http Settings](#), and always available to drivers.
- [HTTPS Proxy Settings](#) fields are providing the address, port and credentials of an optional and fully certified HTTPS Proxy. It is used if [Proxy Usage](#) is set to [Use Https Settings](#), and always available to drivers.

Drivers can make use of any of the provided proxy addresses even if the [Proxy usage](#) is set to [Not used](#).

4.2.2 Using Windows Installer

A Windows Installer can be run to install Control Server as a service on the PC. When running the installer you will need to configure a few settings.

There are two settings to configure:

- HTTP Port: The port you wish to access the Control Server debugger on. *8080* by default.
- UX Server URL: The URL of the UX Server. This is generally the IP address of your virtual machine, unless you assigned a domain name to it. Cannot end with a '\'. (Example: <http://192.168.1.10>)

After installing, the installer will provide you with following information:

- CS ID: The unique ID needed for the Control Server and UX Server to talk.
- CS URL: The URL to access the Control Server debugger.

Control Server is installed as a service on the PC. It is named 'OvertureCS'.

4.2.3 Using Docker

To install a new Docker containing the Control Server, you will need to load the proper container and then run the container.

4.2.3.1 Loading

An archive will be provided with the container inside. To load this container, in your docker environment use the following command:

```
docker load -i myarchive.tar
```

4.2.3.2 Running The Container

The Docker container can be run using docker-compose and a YML containing the configuration information.

```
version: '2'

services:
  felix:
    image: barcooverture/controlserver:latest
    ports:
      - 8080:8080
    environment:
      - driverPath=/data/controlserver/drivers
      - data=/data
    volumes:
      - data:/data
    restart: always

volumes:
  data:
```

Except for port settings (see [Ports adjustment](#)), all other settings should remain. Once, configured use `docker-compose up` to run.

4.2.3.3 Ports adjustment

The port may be mapped differently if needed. If you want to use a different port value, you can replace 8080 by another value, for example 5050. The YML file will then become:

```

version: '2'

services:
  felix:
    image: barcoverture/controlserver:latest
    ports:
      - 5050:8080
    environment:
      - driverPath=/data/controlserver/drivers
      - data=/data
    volumes:
      - data:/data
    restart: always

volumes:
  data:

```

Note: The port settings has two values, **HOST:CONTAINER**, **HOST** is the value seen by the "external world", whereas **CONTAINER** is the value seen by Docker container. In the example above, we expect the reach the Control Server on port 5050, but the Control Server within the container is still listening on 8080.

In certain cases, some driver requires to use a port to receive incoming TCP connections or UDP traffic. In such a case, these ports must be declared in the YML file, to be exposed. For example, if a driver expect an incoming TCP connection on port 3000, then the YML file should be modified as follow:

```

version: '2'

services:
  felix:
    image: barcoverture/controlserver:latest
    ports:
      - 8080:8080
      - 3000:3000
    environment:
      - driverPath=/data/controlserver/drivers
      - data=/data
    volumes:
      - data:/data
    restart: always

volumes:
  data:

```

Note: UDP ports can be specified with a /udp suffix. Example 3000:3000/udp

4.3 Next Steps

To fully get the system online, you will need to configure the system. Please see [System Configuration](#) in the Configurator section to finalize the configuration.

Default credentials to login in apps are:

- username: [medialon](#)
- password: [medialon](#)



5 Configurator

The screenshot shows the Configurator interface. At the top, there are buttons for '+ NEW', 'DUPLICATE', and 'SAVE'. Below these is a search bar with 'Search criteria: Search' and 'Tags: Add a tag' with a 'Search' button. A 'Multi Selection' checkbox is visible. The main part of the interface is a table with columns: Name, Short Name, Alternative Name, Variable Name, Icon, Order, License, Type, Sub-type, Unit, Parent, and Control Server. The table lists various system components like System, ControlServer Status, Campus, Learning Center, Library, Floor1, Floor2, Malaga Room, Malaga Projector, Malaga_Projector Activity, Malaga_Projector Status, Malaga_Projector Power, Malaga_Projector Shutter, Malaga_Projector Sources, Malaga_Projector Brightness, Malaga_Projector Contrast, Malaga_Projector Temperature, Malaga_Projector HoursLa..., Malaga_Lighting, Malaga_Lighting Activity, Malaga_Lighting Status, Malaga_Lighting Presets, Malaga_Lighting LevelChan..., and Malaga_Lighting LevelChan... The right side of the interface shows a configuration panel for a selected item, 'Malaga Projector'. It includes fields for Name, Short Name, Alternative Name, Variable Name, Icon, Point Order, Type, Sub-type, Driver/Setup, Unit, Parent, Server, Roles, and Tags. There is also a 'Map Fields' section at the bottom right.

Configurator is the tool for interfacing with five parts of the Overture system:

- System Configuration: The Overture system configuration. This includes licensing, HTTP(S) settings, security settings and Control Server links.
- Points: The various buildings, floors, rooms, devices, and variables across Overture that you can monitor or control.
- Access Rights: The Overture system's users, groups, and roles that control what users can see and control on the system.
- Assets: The panels and maps that the Home displays and provides links to.
- Alarms: Notifications set up in the system that monitor and alert users of abnormal operating events.

The Configurator can be accessed through the App Switcher (in the top right of the header ) or by going directly to its URL inside of a browser. (Example: <http://10.0.50.231/configurator>)

You may need to log in to the system using your user credentials. If this is your first time logging in, use the default user name and password. (Default username/password: medialon/medialon)

It is highly recommended you change the password for this user once you have set up the system, as well as setting up a strong password policy.

Inside Configurator, the top of the screen is the main control bar. This is where most interaction with items occurs. Each view inside Configurator uses the control bar to create new items, save changes, or change which view of items you are looking at. Each view also has a search bar that helps to find items when there are a lot of items in the database.

The User Menu displays an overview of the currently logged in user's roles, links to user interfaces and administration tools within Overture, and tells you what version of Configurator you are using.

The Views Menu is how you navigate to the management tools for each of the various types of items in the Configurator.

5.1 System Configuration

Overture's System Configuration is in two parts of the Configurator.

The first part is the UX Server Config view. Here you can between HTTP and HTTPS (see [HTTPS](https://)), restart/update the server, license your system, and find other settings for the server.

5.1.1 Licensing

5.1.1.1 Introduction

Overture uses a subscription based license model. Licenses are now entitlements that contain all information pertaining to your installation; such as rooms and servers.

5.1.1.2 Entitlements

When dealing with Entitlements there are two ways to act upon your Entitlement: offline and online.

Offline Actions

All offline actions will give you controls to download and upload a file. Once you download the file you will need to upload it to the correct page. Offline verify and offline activate both need to be uploaded to www.activate.barco.com/upload, while offline release needs to be uploaded to www.activate.barco.com/release.

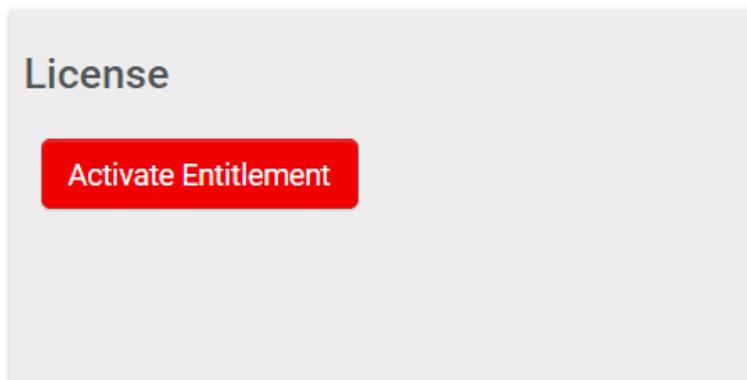
Online actions

All online actions require no extra actions from the user than clicking the button. Online actions can only be performed once you have a valid entitlement associated with your Overture installation.

Adding your first entitlement

When you first load up Overture the software will not have any entitlements. After purchasing an entitlement from www.barco.com you will get an entitlement id.

1. Login to [Configurator](#), and navigate to [Server Configuration](#) view ([Settings](#) section).
2. You need to click on activate:



3. Add your entitlement id and click offline activate:

License

Entitlement ID: Z1-S00008627

Cancel

Offline Activate

4. Download a request file using download button, that you will use on the activation portal:

License

The Request File is ready

Entitlement ID: Z1-S00008627

Cancel

Offline Activate

Step 1 - Download
Request File and upload to www.activate.barco.com/upload

Download

Step 2 - Upload
Response File from www.activate.barco.com/upload

Upload

5. Activate your request file over at www.activate.barco.com/upload:
 6. Upload the activation file you received from the portal using Upload button:
- After uploading your activation file you will now have an active entitlement.

License

Entitlement ID(s):

Z1-S00008627

Feature(s):

Room Container - 10

Activate Entitlement

Offline Verify

Online Verify

Offline Release

Online Release

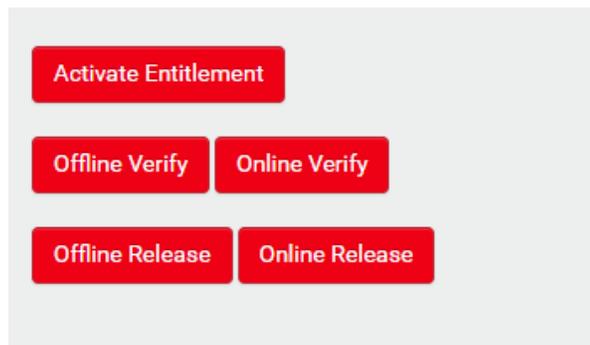
All future entitlement activations will have the online activate option enabled. In step two just click online activate and the system will add your entitlement for you without any need to download it.

Releasing an entitlement

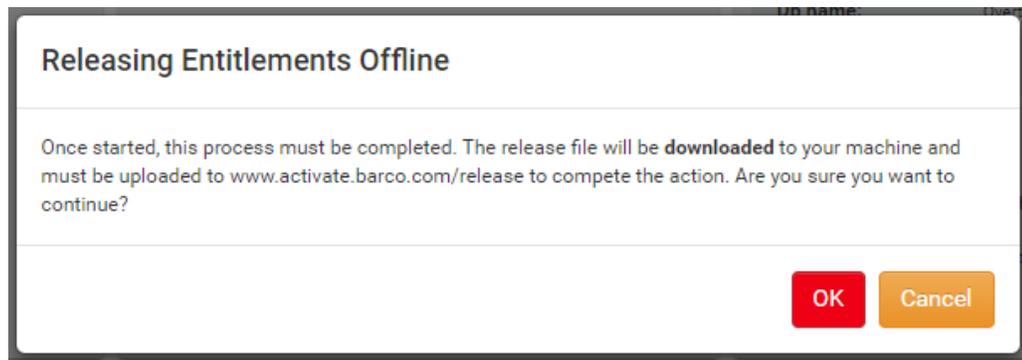
Releasing an entitlement removes it from your Overture VM but does not invalidate your entitlement. This allows you to remove an entitlement from one Overture VM and apply it to another. One use case for this is taking an entitlement from your testing environment and move it to your production.

Releasing entitlements offline

1. Login to [Configurator](#), and navigate to [Server Configuration](#) view ([Settings](#) section).
2. Click the offline release button



3. Confirm that you want to release your entitlement by clicking ok in the dialog.

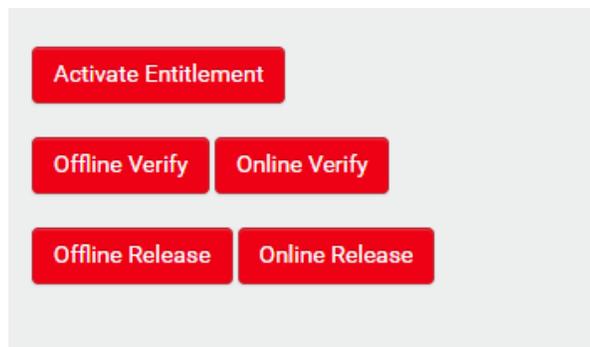


4. Upload the downloaded file to www.activate.barco.com/release

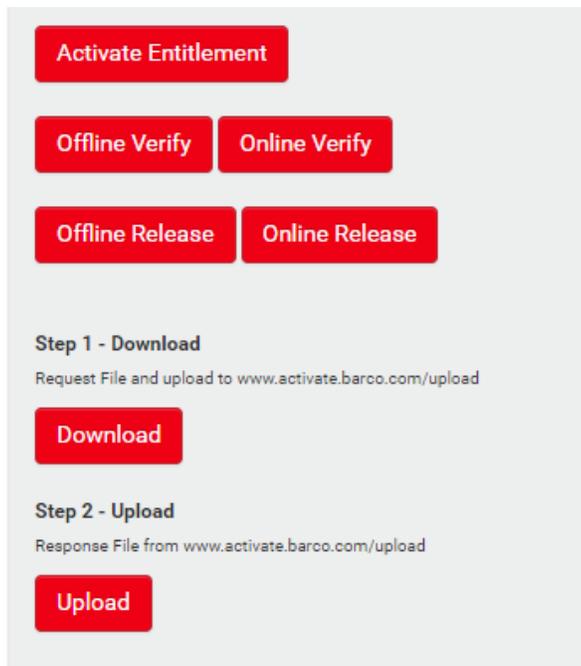
Verifying your entitlements

When updating your entitlements through www.activate.barco.com/upload your changes will not showing on Overture right away. To see the changes for your entitlement you will need to verify it. You can either verify online with the Online Verify button or verify it offline. Offline verifications are the same as activating entitlements offline.

1. Login to [Configurator](#), and navigate to [Server Configuration](#) view ([Settings](#) section).
2. Click on the Offline Verify Button:



3. Download the entitlement using download button:



4. Activate your entitlement over at www.activate.barco.com/upload:

5. Upload the entitlement using Upload button:

Your entitlement information should now be updated.

5.1.2 Password Policy

From Overture UX 3.7.0 onwards a password policy section has been introduced in the Server Configuration page to reinforce security. By default, this added feature introduces a minimum policy as described below as 'default parameter'. A password policy dictates what form a password must respect as well as other criteria such as change frequency and re-usability of an already used password.

Password Policy

Minimum Length:

Digits:

Lowercase Characters:

Uppercase Characters:

Special Characters:

Not Username:

Not Recently Used:

Parameters:

- *Minimum Length* Indicates the minimum number of characters a password is allowed to have (default value is 8)
- *Digits* Indicates the minimum number of numerical characters a password must have (default value is 1)
- *Lowercase Characters* Indicates the minimum number of lowercase characters a password must have (default value is 1).
- *Uppercase Characters* Indicates the minimum number of uppercase characters a password must have (default value is 1).
- *Special Characters* Indicates the minimum number of special characters a password must have (default value is 1). Special characters are characters like '!#%\$@'.
- *Not Username* Prevent to use the username as the value for the password (default value is checked).
- *Not Recently Used* Prevent reusing an already used password for the last x times. This policy saves a history of previous passwords. The number of old passwords stored is configurable. When a user changes their password they cannot use any stored passwords (default value is 3).

5.1.3 Control Server Links

In order to create and control devices within the system, you must link the UX Server together with the Control Server(s) on the network.

5.1.3.1 Control Server



OvertureCS

SERVER DEVICES TASKS VARIABLES

Status

Disconnected

Setup

OvertureCS ID *
MONCLT28832-9dab6557

UX Server URL *
http://192.168.1.19

HTTP Server Port (OvertureCS will restart if changed)
8080

EDIT

The first step is making sure the Control Server is pointed at the UX Server. To do this, you must first open the Control Server debugger interface. To access the debugger, open a web browser and go to the CS URL. The CS URL is 'http://' plus the IP address of the Control Server machine, plus the HTTP port (default: 8080). The URL will look similar to this: <http://192.168.1.10:8080>.

Click the 'Edit' button, and change the UX Server URL to the appropriate URL if needed. The URL should not end with a '\'. Click Save.

Make note of the OvertureCS ID as will need this information in the UX Server.

5.1.3.2 UX Server

Servers

+ NEW ✓ SAVE

Refresh

Name	Type	Description	Reconnection	
ControlServer	CS2	UID: LocalControlServer		

Name: *

Type: *

UID: *

Use Password: * (Supported by ControlServer 1.6.0 or greater)

Login Password: *

Confirm Password:

The second step is to define the link on the UX Server side. To do this, go to the 'Control Server' views in Configurator. From there you can add, or edit existing, Control Servers.

Select the Control Server you wish to edit, or click 'New'. Define the following settings. (See the [Control Servers](#) section for more details):

- Name: A human name for the Control Server. For example 'Building 1'.
- Type: The type of Control Server (CS1/CS2). CS2 is default. If using CS1, please see [Appendix A](#) for interacting with them.
- UID: The ID gathered from the Control Server. This is where you will enter the ID you made note of earlier.
- Use Password: Check this option if you want to protect the user interface of the Control Server with a password (Note: password protection is available on Control Server 1.6.0 or greater).
- Password and Confirm Password: Type the password and confirm.

Note: Control Server passwords follow the password policy defined in the UX Server Setup, see [Password Policy](#) Chapter.

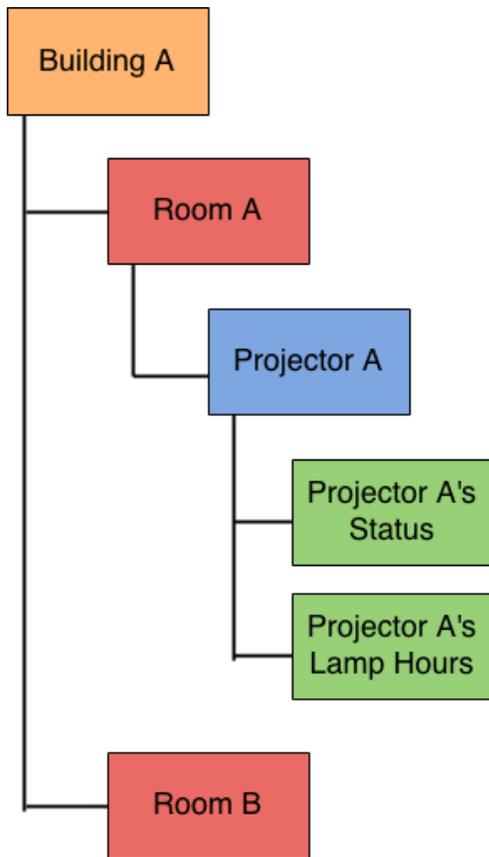
Click 'Save' at the top of the screen.

Once the information is entered and saved, the link between the devices will be created. For each Control Server, a Status variable will be created allowing you to monitor the connection between UX and the CS. You can also see this connection in the Control Server debugger by looking at the 'Status' section. If the link is there, the status should say 'Connected'. If it is not connected, and the information entered was correct, you may need to troubleshoot your network.

5.2 Points

A point is an item in the system. It represents physical items like devices, rooms, and buildings as well as their variables. Points have a number of attributes related to them and also act in a parent/child/sibling relationship between each other. Several point's types are available as built-in types, such as Room, Device, Buildind, etc..., but custom types can also be created to suit virtually any organisation structure (See [Point Types View](#))

5.2.1 Parent-Child-Sibling Relationships



One of the main features of the Overture system is how points relate to each other. A point can be defined as parent of another point. In the above example, a projector's status and lamp hour variables are children of that device, the projector device is a child of Room A. Both Room A and Room B are children of Building A. When the system creates points, like device variables, it will automatically assign the parenting to them.

Parenting is how devices are displayed in rooms, and rooms are displays in locations.

5.2.2 Point Attributes

- ID (read-only): Every point registered with [Overture Configurator](#) is automatically assigned a unique ID by the system. It can be used to refer to a single, specific point in [Editor](#) templates.
- Human Name: The full, human-readable name for a point that must be chosen manually by the user. This is the label of the point that will be used to represent the point in most cases throughout the [Overture](#) system. *Example: Projector A Status*
- Short Name: A much shorter, *context-dependent* human-readable name that replaces [Human Name](#) for smaller labels (e.g. in the [Widget Dashboard](#) and [Map of Home](#)) *Example: Status* (e.g. in the context of a widget focused specifically on [Projector A](#), this would imply the point is [Projector A Status](#))
- Alternative Name: This is the human-readable name that will be used instead when an [Overture](#) application is set to alternative language mode.
- Variable Name: The full "machine name" that must uniquely identify a point under the control of a specific [Control Server](#). *Cannot contain spaces.*
- Point Order: Used for changing the view order of a point when multiple points are selected within the same template. By default, points will keep the order in which they are specified in the database (i.e., the order of points

as seen in [Configurator Points View](#)).

Note: When displaying points using [med-repeat](#), sorting is done for items at the same level of child/parent relationship first, then for other levels. So the room variables are always shown first, before the devices variables, respecting the order separately.

- **Type:** The broad type of the point (e.g. [device](#), See [Point Types View](#))
- **Sub-type:** The specific type of the point (e.g. [projector](#), which would could be registered as a sub-type of [type: device](#)).

N.B. [Type](#) and [Sub-type](#) are used in conjunction to determine default settings, such as icons and templates, with [Sub-type](#) taking priority over [Type](#).

- **Unit:** The unit of measurement that the value of the point represents. Usually appears as a suffix, added to the end of the displayed value of a point. *Example:* °C/°F, %, dB etc.
- **Parent:** The parent of the point. Used to define inheritance, or show assets on a map. *Example:* [Projector A](#) as the [parent](#) of [Projector A Status](#)
- **Room Calendar Name:** Used to associate the point to the Room Booking System. Applies an automation logic when the meetings starts and ends through an 'overture_room_calendar' behavior. *This field is displayed only if the point is a Container (See [Containers](#)) and a connection with the Room Booking System is configured correctly (See [Room Booking](#)).*
- **Timezone:** For certain point's type, when the timezone option is activated (See [Point Types View](#)), a timezone can be specified for a point. The timezone setting works hierarchically, whereas a child timezone takes precedence to a parent timezone. For instance, if a timezone 'A' is set at a country level in a hierarchy, all children below are considered in timezone 'A', even though they don't show a specific timezone in their properties. But if a building, child of this country, has a different timezone set (i.e timezone 'B'), then all children below this building are considered in timezone 'B'. Timezone option for a point's type can be setup into the Point's Type view. Timezone information is not directly used in Overture UX, but will be used within Overture Monitoring & Insights.
- **Server:** References the [Control Server](#) the point is controlled by, if any. *Used only for devices, tasks, and user variables.*
- **Roles:** The [roles](#) associated with the point, used to specify access rights. *Example:* If a point has a [Security](#) role, only users whose [group](#) has at least [read](#) access to the [Security](#) role are able to see the point displayed. (See [Access Rights](#)).
- **Tags:** A searchable list of common terms used to identify groups of points. Useful for cases where multiple points share something in common, but not a [type](#) or [sub-type](#) for example. *Example:* An "Audio" tag might be created and assigned to both an audio volume point, and an audio source point. [Tags](#) can also be used as a part of a search filter in templates (e.g. in the [med-repeat](#) attribute).
- **Alarms:** One or more references to [Alarms](#) that may be triggered upon this point's value satisfying a certain condition. (See [Alarms](#))
- **Metadata:** Additional JSON-formatted information used to supplement the properties listed here. Accessible via `{{ point.metadata.<property> }}`, this is used for more advanced/specialized templating. *Example:*

```
{  
  "image": "/assets/images/room2.jpeg"  
}
```

...and in a template with access to [point](#):

```

```

- License: (read-only) Whether or not the point needs a license, and if that license is valid.
- Show on Device UI / Show on UI: Whether or not to display a widget in the control panels for this variable's point. Enabled by default. It is an alternative to roles access rights, or a custom control panel, to hide a variable from all control panels.
- Show on Container UI: (Only for device's variables) Whether or not to display this variable on the "container" of the device owning this variable. For example: Show the Projector Source on the [Room](#) tab.
- Show on Container UI Condition: (Only if [Show on Container UI](#) is enabled): Apply a condition on the visibility of a variable on the container control panel. For example: Show the Projector Source, only if the projector is turned on.

For the *Container UI* logic to work, your "container" control panel (usually the room) need to do a [med-repeat](#) finding all the variables of its devices. Usually the [room_tab.html](#) uses [med-auto-widget](#) on all variables of the room, depth 2.

5.2.3 Map Fields

In order to be accessible on a map in [Overture Home](#), a point needs the following:

- A parent (or any direct ancestor point) that has a map asset assigned to it
- [X](#) and [Y](#) coordinates, set with respect to the parent/ancestor map image to be used

Map fields: +

The screenshot shows a configuration panel for map fields. It includes the following elements:

- X:** Input field containing '675' with 'px' units and a magnifying glass icon to its right.
- Y:** Input field containing '431' with 'px' units and a magnifying glass icon to its right.
- Z:** Empty input field.
- Control Panel:** Dropdown menu showing 'Room Panel' with a refresh icon to its right.
- Map:** Empty dropdown menu with a refresh icon to its right.
- Default:** A checkbox that is currently unchecked.

5.2.3.1 X, Y and Z Coordinates

To define the point's [X](#) and [Y](#) coordinates, click on the magnifying glass icon to the right of the [X](#) field. This will open a map image in the main area of [Configurator](#). Clicking on an area of this map image will set the point's coordinates to the area corresponding to the area clicked. You can also set the [X](#) and [Y](#) coordinates manually via their corresponding fields.

You can also define a "Zoom Coordinate" ([Z](#)) for the point. This allows the point to only show up on the map at a certain level of zoom, which is useful when you have a lot of points to display in a relatively small area. The higher the number the more zoomed in you must be to see the point. The value should be between 0 and 300. The exact amount of zooming needed to see the point depends on the map image size and view port of the browser, but in general, a value of 0 (or when it is not set) means that the point should be visible on all levels of zoom, and a value of 300 means that the point should only be visible when the map is completely zoomed in. *Setting any value above the maximum of 300 will make the point invisible no matter what the level of zoom is.*

5.2.3.2 Control Panel Selection

Points can have also control panels assigned to them. These control panels will be how your points will be displayed in map of [Overture Home](#). A default set of basic templates is built into all distributions of [Overture](#), but more can be created via the [Assets View](#). These newly-created control panels assets (and even the default ones, though it is not recommended) can be edited to suit your needs using [Overture Editor](#).

Without an explicit reference to a [Control Panel](#) asset, a default control panel HTML template file is chosen with the following priority order:

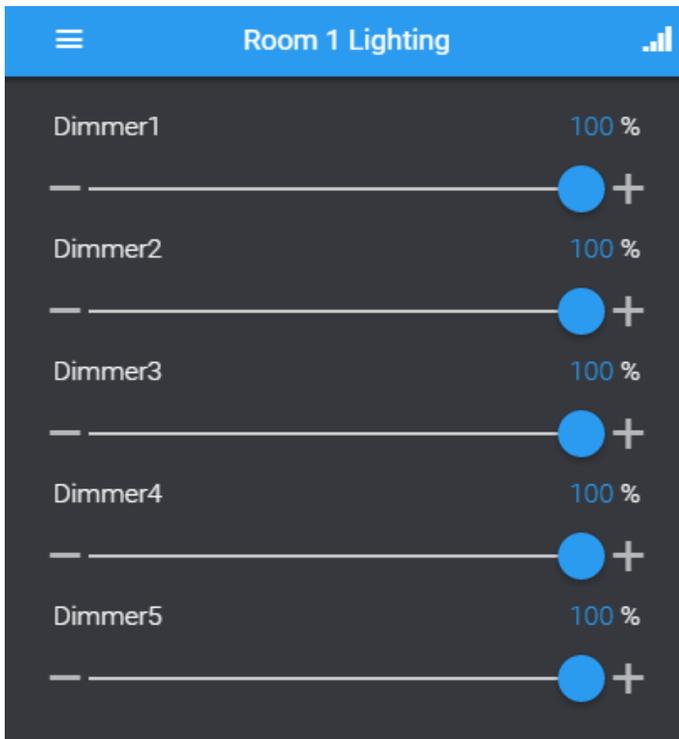
1. [views/common/<sub-type>.html](#) where [<sub-type>](#) is the ID of the point's sub-type (in lowercase)
2. [views/common/<type>.html](#) where [<type>](#) is the ID of the point's type (in lowercase)
3. [views/common/generic.html](#)
4. [views/generic.html](#)

[Overture](#) will attempt to use each of these files in order, and will choose the first file it can successfully find to be the HTML template used for the point's control panel.

Example: A point [Malaga Device 1](#), with no [sub-type](#), but with a type of [Device](#) will use [views/common/device.html](#) if it exists, or the generic template if it does not.

NOTE: A point can have both a map and a control panel at the same time. In this case, in the map area of [Overture Home](#), you can display the point's control panel by clicking on the point's name, and navigate to its map by clicking on its label's magnifying glass icon.

5.2.4 Devices



Devices are a specific type of point in the system. They represent devices on the network that are being monitored or controlled. They must be linked to a Control Server and have a variable name.

To create a device in the system, follow these steps:

- Create a new point by clicking 'New' in the points view.
- Give the device a human name and variable name of your choosing.
- Select 'Device' from the type field.

- Select an appropriate sub-type for the device. If the default ones are not appropriate, you can add more in the 'Sub-Type' view of Configurator.
- Click the magnifying glass next to 'Driver/Setup'.
- Select the appropriate driver from the Plugin Selection. If your driver is not listed, see the '[Plugins View](#)' for more info.
- Fill out the required setup information for your device, this will be different depending on the driver. You can change the setup information, by selecting the gear next to 'Driver/Setup'
- Select which Control Server this device will be created on.
- Optionally, you may fill in any extra information you have such as parent, roles, tags, etc.
- Click 'Save'. The device, and variables associated with that device will be created in the database.

It's important to know that the device and its variables will not be licensed until you select a parent that is an appropriate type like Room, Building, or Floor.

5.2.5 Containers

Containers are points like rooms (see [Rooms](#)), floors, or other large areas in your system, that contain devices, variables and other containers.

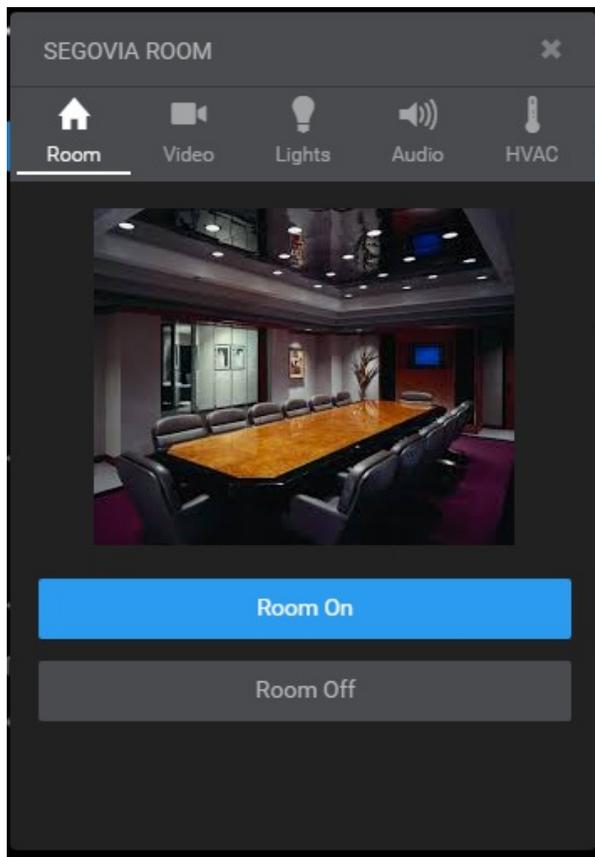
To create a container in the system, follow these steps:

- Create a new point, by clicking 'New' in the 'Points' view.
- Give the container a human name.
- Select the type of container, for example 'Room', from the type field.
- Optionally, you may fill in any extra information you have such as parent, sub-type, etc.
- Click 'Save'. The room is created in the database.

If the container is a 'Room', you should now change the 'Parent' attribute of any devices that are in this room, to this point.

Note: Devices don't have to be only children of a 'Room', any type of container can contain devices.

5.2.5.1 Rooms



Rooms are another special type of point in the system. They represent rooms or large areas in your system that contain devices. Rooms, by default, have an associated template to them that displays all of the devices within it.

5.2.5.2 Other type of Containers

Like rooms, other type of points are considered as containers (they can contain other points), the build-in types are:

- Building
- City
- Country
- Floor
- Site
- Zone

From the Types view, new types can be created to build your organisation tree.

5.2.6 Behaviors

5.2.6.1 What is a behavior

Like for a device's point, where a driver can be attached, behaviors can be attached to a container's point. Behaviors are like drivers, they are running into a Control Server and provide an automation logic for this container.

Overture supports two generations of Behaviors, Behavior 1.0 (introduced in version 3.1.0) and Behavior 2.0 (introduced in version 3.2.0). Although it is recommended to exclusively use Behaviors 2.0, the support of Behavior 1.0 gives the ability to upgrade an existing configuration progressively.

Behavior 1.0 gives this ability to attach one behavior to a point, like a driver, and configure this behavior to provide the automation logic proposed by the chosen behavior

Behavior 2.0 brings the behavior concept to the next level, giving this ability to:

- Attach one or more behaviors to a single point
- Configure each attached behavior independently
- Interact between behaviors through variable's states
- Save the combination behaviors in one template, which can then be applied to other points, providing the same automation logic than this point
- When using the same template on multiple points, changing the configuration of this template changes the automation logic for all points simultaneously, avoiding the need to re-edit each point individually.

5.2.6.2 Behavior Setup

To attach a behavior to a container:

- Select the container point, e.g. a point with type 'Room', 'Floor' or 'Building', for instance
- Click on the 'Behavior Setup' field to open the behavior selector:

The screenshot shows a configuration form with three fields: 'Type' with a dropdown menu set to 'Room', 'Sub-type' with a dropdown menu, and 'Behavior Setup' with a button icon. The 'Behavior Setup' field is highlighted with a red rectangular box.

- Select the behavior from the list:

The screenshot shows a 'Plugin Selection' dialog box overlaid on a configuration screen. The dialog has a search bar and a table of results. The first row, 'overture_power_cycle', is highlighted with a red box. The background configuration screen shows buttons for '+ NEW', 'DUPLICATE', and 'SAVE', and various input fields.

Name	Description
overture_power_cycle	Behavior which manages a power on/off cycle
overture_custom_setup_dialog	Behavior which manages a power on/off cycle, tri...
overture_onoff_container	Container On/Off Behavior
overture_triggers	Behavior which manages triggers and actions
overture_macros	Behavior which manages macros

- Configure the selected behavior, as in following screen 'overture_power_cycle':

Room Setup

Overture Power Cycle ✕ Add...

Behavior: `overture_power_cycle` Version: 2.0.0

Powering On

Variable/Statement	Value	Condition
London.Power	On	

+

Power On

Variable/Statement	Value	Condition
--------------------	-------	-----------

+

Powering Off

Variable/Statement	Value	Condition
--------------------	-------	-----------

+

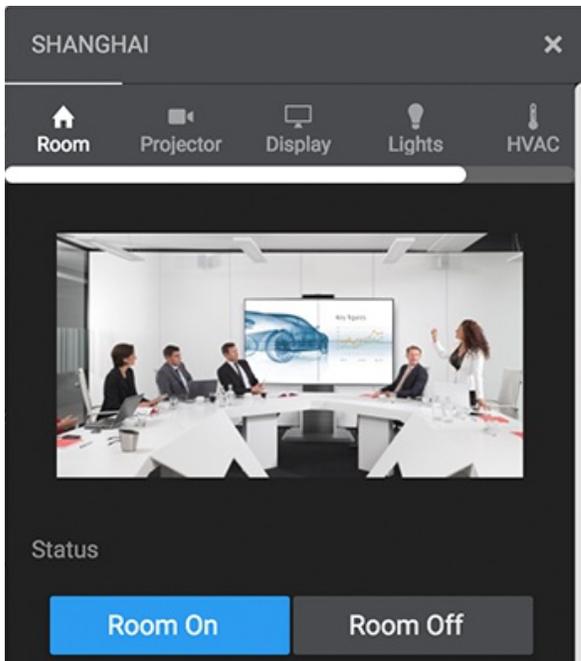
Power Off

Variable/Statement	Value	Condition
London.Power	Off	

+

- Click Save

When a behavior is attached to a container, the template for the container displays the behavior controls, like in this room preset example:



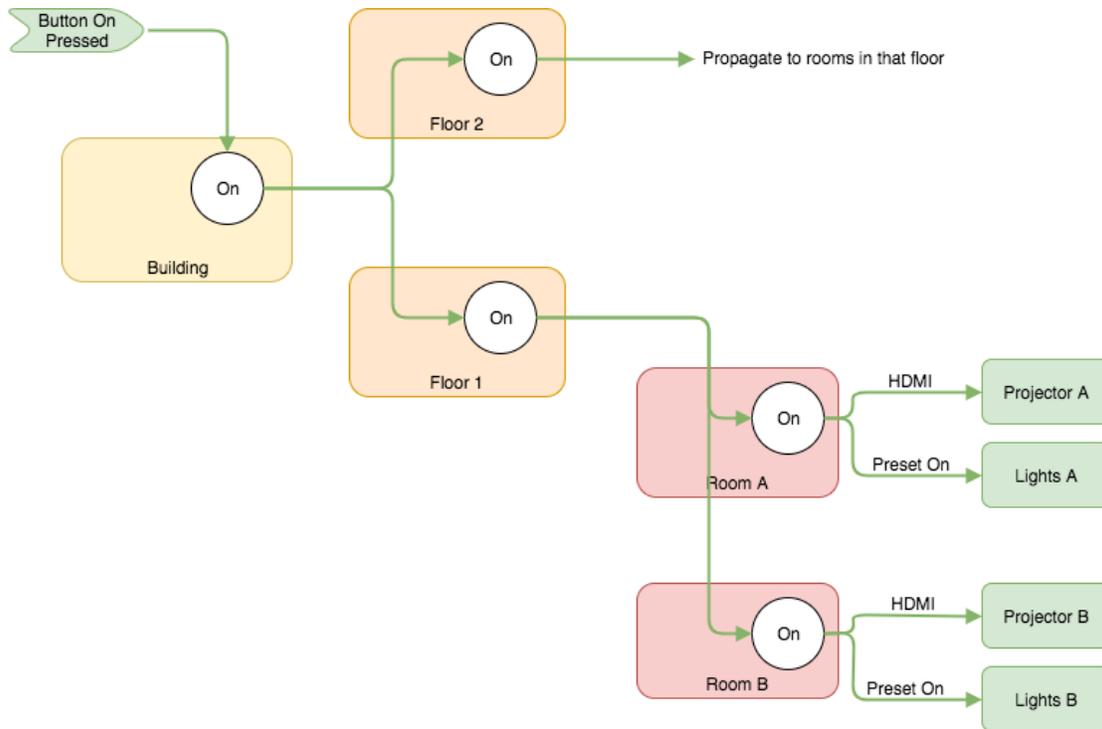
In this example, with the 'overture_room_onoff_button' attached to room 'Shanghai', the UI generates two buttons, 'Room On' and 'Room Off'. This will respectively change the room's status to On and Off, using the parameters specified in the

configuration of this behavior, as illustrated above (HDMI will be selected as source input and light preset 'On' will be recalled when the room's status is changed to on, whereas the light preset 'Off' will be recalled when the room's status is changed to off).

5.2.6.3 Behavior Cascading

A behavior performs actions in the container it is attached to. Those actions, for instance [Power On/Power Off](#) in the case of [overture_onoff_container](#), are applied to the children of this container, therefore, if those children also have the same behavior attached to them (or a behavior which accepts the same actions), they will also propagate these actions to their children.

This cascade mechanism may be used to change the status of an entire building to On or Off, like illustrated below:



5.2.6.4 Behavior Combining

Behavior Combining allows for the grouping of behaviors together which can either then be attached directly to a point, or saved out as a Behavior Template and attached to multiple points at once. Behavior combining is a powerful tool to define various automation logics, based on simple behaviors that are configurable.

To combine behaviors

- Click the [Add...](#) button tab.

Plugin Selection

Search:

Search results:

Name	Description
overture_onoff_container	Container On/Off Behavior Driver
overture_room_onoff_button	Behavior which sets up a room when a button is...
overture_macros	Behavior which manages macros

This will open the plugin selection box (as seen above).

- Select a behavior
- Setup the behavior
- If you want to add more behaviors click [Add...](#) again
- Once you are done click [OK](#) , or [Save as Template](#) if you wish to save the combination as a [Behavior Template](#)

5.2.6.5 Behavior Templates

Behavior templates offer a mechanism for the packaging behaviors into reusable templates. These can then be easily attached to points without having to recreate the multiple versions of the same behavior(s), as illustrated below:

There are a few ways to create a Behavior Template:

Creating within the Behavior Template tool

Within the Overture Configurator tool select [Views](#) > [Behavior Templates](#)

Behavior Templates

Name	Behaviors	Linked Points
------	-----------	---------------

Name: *

To create a new template.

- Click [NEW](#)
- Enter a name for the new Behavior Template in the [Name](#) box.
- Clicking [Edit <new name>](#) box.
- [Create and combine the desired behaviors](#)
- Click [OK](#)
- Click [SAVE](#)

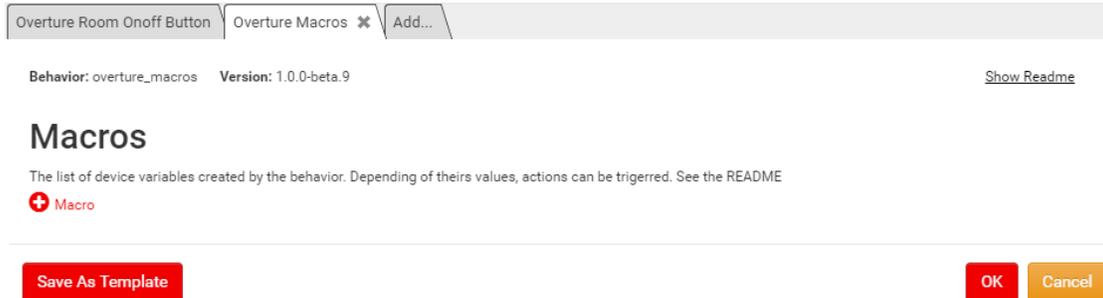
Create within the Points tool.

From within the Points Configurator ([Configurator](#) > [Views](#) > [Points](#)) select a point that allows for attaching behaviors.

Behavior Setup: 

- Click the  icon.
- Select [Setup Behavior](#)
- [Create and combine the desired behaviors](#)
- Click [OK](#)

Create within the Behavior tools.



Behavior: overture_macros Version: 1.0.0-beta.9 [Show Readme](#)

Macros

The list of device variables created by the behavior. Depending of their values, actions can be triggered. See the README

[+ Macro](#)

[Save As Template](#) [OK](#) [Cancel](#)

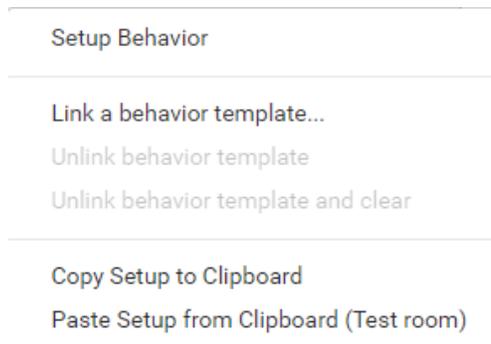
- By saving a group of behaviors to a template from within the [Behavior Setup](#) in the [Configurator > Views > Points](#) tool. Once a behaviors are configured as desired,
 - click the [Save As Template](#) button
 - entering a name for your template in in the [Behavior Template Name](#) box
 - clicking [Save](#).

Attaching a Behavior Template to a Point

From within the Points Configurator ([Configurator > Views > Points](#)) select a point that allows for attaching behaviors.

Behavior Setup: 

- Click the  icon.



- Setup Behavior
- Link a behavior template...
- Unlink behavior template
- Unlink behavior template and clear
- Copy Setup to Clipboard
- Paste Setup from Clipboard (Test room)

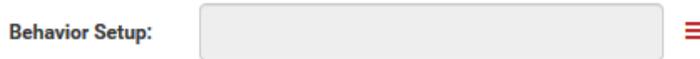
- Select [Link a behavior template...](#) from the context menu
- Select required Behavior Template
- Click [OK](#)

Un-linking a Behavior Template

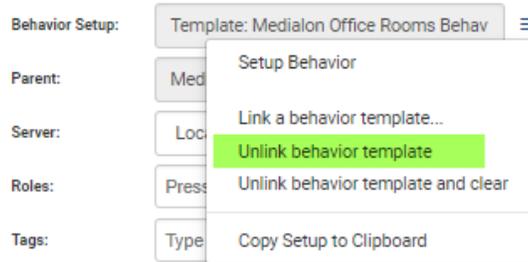
P 95 / 270

Un-linking a behavior will convert it into normal behavior and not a linked template. After un-linking, any changes will not propagate to any other points linked to the original behavior template.

From within the Points Configurator ([Configurator](#) > [Views](#) > [Points](#)) select a point that has a behavior attached.



- Click the > icon.

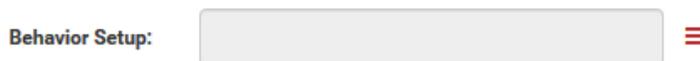


- Select [Unlink behavior template](#) from the context menu

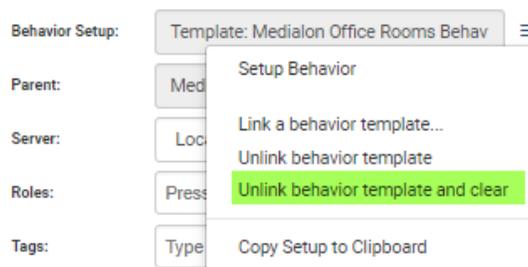
Clearing a Behavior Template

Clearing a behavior template will remove the reference from the point. The behavior will no longer be attached to the point.

From within the Points Configurator ([Configurator](#) > [Views](#) > [Points](#)) select a point that has a behavior attached.



- Click the  icon.



- Select [Unlink behavior template and clear](#) from the context menu

Copy and Pasting a Behavior Template

From within the Points Configurator ([Configurator](#) > [Views](#) > [Points](#)) :

- Select a point that has an attached behavior that you want to copy.
- Next to [Behavior Setup](#) click the  icon.
- Click [Copy Setup to Clipboard](#) from the context menu.
- Select the point you wish to copy to behavior information too
- Next to [Behavior Setup](#) click the  icon.
- Click [Paste Setup from Clipboard](#) from the context menu

5.2.7 Built-in behaviors components

Most of Overture behaviors are build upon common behavior's components which provide common services. This chapter

lists these common behavior's components and explains how to use them. When reading the manual of a particular behavior, the section which involves one of these components will direct you to this chapter for further explanations.

A typical example of such shared common behavior's components is the [Action List](#), which is used by [overture_macros](#), [overture_trigger](#) and [overture_power_cycle](#) behaviors.

5.2.7.1 Action Lists

[Action lists](#) allow a user to build a sequential list of actions that can be executed either unconditionally or conditionally. [See: Conditions Editor](#). Another terminology for [Action List](#) which is often used in certain applications is [Task](#), and for [Action](#) this is often the term [Cue](#).

<code>+_Projector.Sources</code>	<code>DVI</code>	
<code>wait</code>	<code>1000</code>	
<code>waitfor</code>	<code>10000</code>	<code>+_Projector.Sources equals DVI</code>

The above example has three columns. The first is the action you want to execute. The second is the value related to that action. And the third is the condition for which is action will be executed under.

When a condition is left empty it is considered to be equal to [true](#)

Actions

There are five different types of actions:

- **Set Variable**

Change the value of a variable.

The action is a point selector. There are three types selectors available

- [DeviceName.VariableName](#) is used to create an absolute path to a point.
- [.DeviceName](#) creates a relative path within the current context.
- [+Name.VariableName](#) create a concatenated path of the room and device names for matching points.
For example, you have a room with the variable name of [QA_Room](#). Within that room, you have a lighting device with the variable name [QA_Room_Lights](#). You can reference the [status](#) value of this light using [+_Lights.Status](#) because the room name [QA_Room](#) is concatenated with the [_Lights](#) and the device is found.

This feature can be really helpful when Rooms and Devices share a standardized variable naming convention. [Meeting_Room_1](#), [Meeting_Room_1_Lights](#) and [Meeting_Room_1_Projector](#) for example.

- **Wait**

Cause a delay in milliseconds.

For example `wait = 4000` will not process any further commands for 4000 milliseconds. *Not adding a value, setting the value to 0 or adding a negative number will result in the [wait](#) action to have no effect.*

- **Wait For**

Wait x milliseconds for the condition to be true.

For example `waitfor = 4000` will wait for the condition to be true for 4000 milliseconds. *If you fail to set a value, set the value to 0 or set a negative number the action will wait forever if the condition never becomes true!*

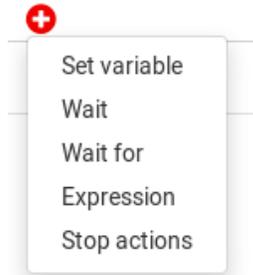
- **Stop Actions**

For example `stop =`

Will stop any further actions in the current task (action list) from running. This can be useful when testing if you want to stop any further actions from happening.

- Expression
For example: `exp = perform('Projector', 'Set Power', {'Status': 'On'})`

Creating an Action list



To create an action list

- click the plus button and select the action type you wish to use
- If you are choosing a variable you can manually enter this in the [Variable/Statement](#) box or click the  icon to open the point selector [See: Point Selector](#) and enter the new value in the [value](#) box.
- If you choose [Wait](#) enter the millisecond value you wish to wait for in the [value](#) box.
- If you choose [WaitFor](#) enter the millisecond value you wish to wait for in the [value](#) box before timing out.
- If you choose [Stop Action](#) no value is required.
- If you choose [Expression](#) enter the expression in the value box.

You can then click on the empty condition box to open up the [Condition Editor](#)

Ordering an Action List

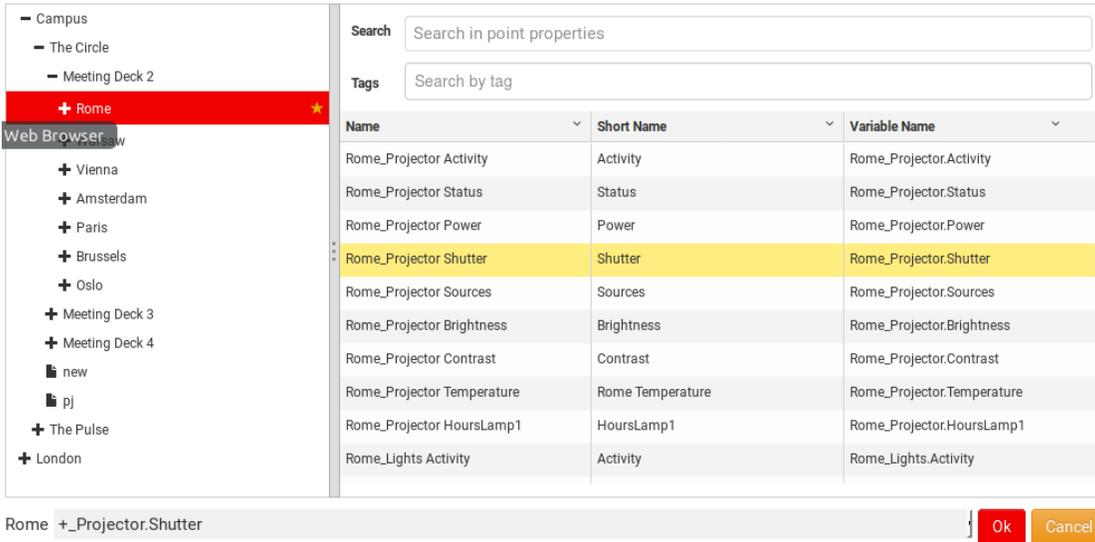
Hovering over an action with your mouse will reveal the order icon . This can be dragged by using the mouse cursor to reorder the Action List.

5.2.7.2 Points Selector

The point selector tool provides an easy way of finding one or many points using a tree view or filter approach. The point selector is used whenever a point, or a collection of points needs to be selected.

For example, it can be launched from within an [Action List](#) by adding a new action, selecting [Set Variable](#) and then clicking on the  icon.

Tree View



The tree view tools allow for a specific point to be selected by filtering points based on their location in the point hierarchy within the system.

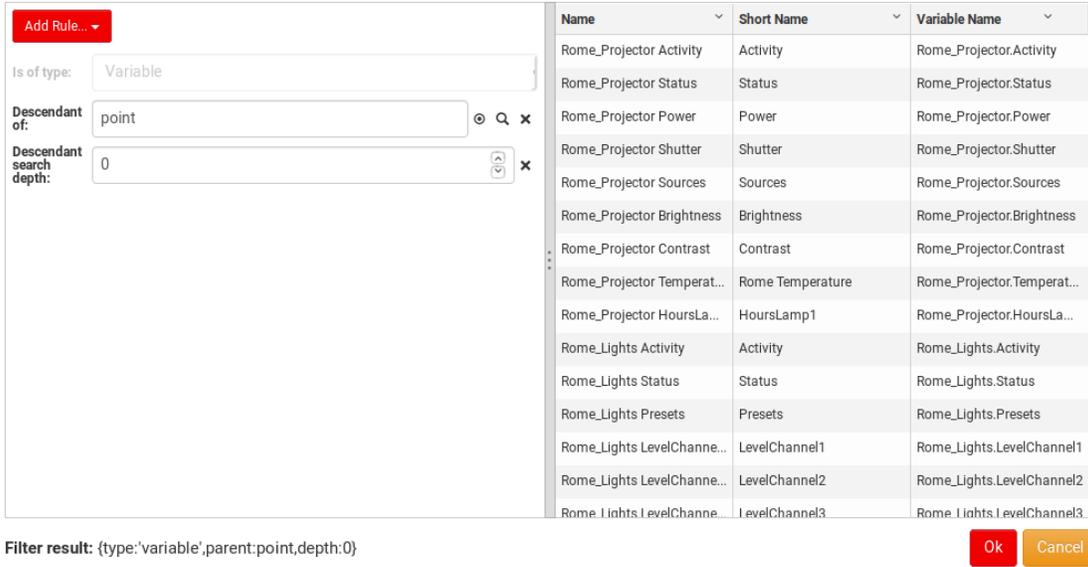
A gold star ★ can be seen to the right of current point.

If a selected point is a child of the current point the selector will be a relative path. For example `+_Lights.Shutter`.

If a user wishes they can use the dropdown box to set the path to as absolute. For example `Rome_Projector.Shutter`.

Filter

The filter tool is split into the panes. The left pane is where the filter is defined. The right pane shows the current result of the filter.



By default, the filter will show all descendants of the current point with a search of depth of 0. Additional rules can be added by using the **Add Rule...** button.

The rules include:



- Descendant of
The base point for which the filter will apply. Entering [point](#) will make the current point the base.
- Descendant search depth
How deep the filter should go. Setting [0](#) will show all descendants. [1](#) will limit the results the immediate child points. [2](#) will show the children of those child elements.
- Human name contains
Filter within the [human name](#) of the point.
- Short name contains
Filter within the [short name](#) of the point.
- Variable name contains
Filter within the [variable name](#) of the point.
- Is of type
The [type](#) of the point.
- Is of subtype
The [subtype](#) of the point. For example [Integer](#), [String](#) or [Enum](#)
- Has Role(s):
Filter by [role](#) or [roles](#) assigned to the point.
- Has tag(s):
Filter by [tag](#) or [tags](#) assigned to the point.
- Under control server
Filter by the control server the point is assigned to.

5.2.7.3 Conditions Editor

Conditions specify the rules that must exist for the action to take place or continue. They are built using the conditions editor behavior.

Condition



The Conditions Editor can be opened from within [Action List](#) by hovering over the Conditions box and clicking on the  icon or by simply clicking on the conditions box.



Preview

+_Lights.Presets equals Special Event AND (+_Projector.Shutter equals Opened AND+_Projector.Brightness is greater than 80)

Rules are build using the following equality conditions:

- equals
- does not equal
- less that
- less than or equal
- greater than
- greater than or equal

And to detect when a value has changed:

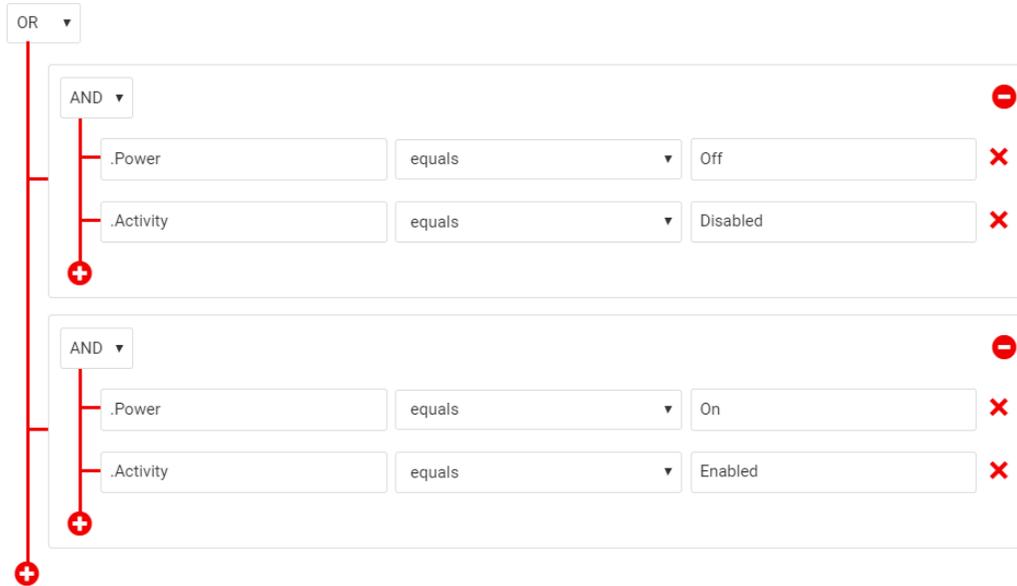
- changes

Rule groups



Rule groups allow for rules be group together. This is helpful if you want to have an **OR** condition where either of the rules can be **true** for the condition to be met.

It is also possible to group multiple Rule groups together.



Preview

```
(.Power equals Off AND .Activity equals Disabled) OR (.Power equals On AND .Activity equals Enabled)
```

The above example shows two [Rules group](#) containing two statements surrounded by an [AND](#) condition. These two rules groups are in turn part of another [Rules group](#) surrounded by an [OR](#) condition.

So, either of the internal [Rules groups](#) can be true for the condition to be fulfilled.

5.2.8 Enum Labels

To edit the Enum Labels select a variable with the sub-type [Enum](#) and click on the label icon:

Type:

Sub-type:

There are two modes where the user can edit the labels:

- Basic Mode When just labels are present and a label is defined for an enum the [label](#) is used throughout overture

Enum Labels

Advanced

Name	Label
PC	
DVI	
HDMI1	
HDMI2	
DisplayPort	

- Advanced mode Where the labels are present with different types



Enum Labels

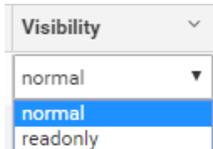
 Copy  Paste Advanced

Name	Label	Alt Label	Visibility	Cmd Label	Alt Cmd Label	Icon
PC			normal			
DVI			normal			
HDMI1			normal			
HDMI2			normal			
DisplayPort			normal			

The priority of displaying the labels will be in this order if they are defined (From low to high)

1. Label
2. Alt Label (Alternative Label)
3. Cmd Label (Command Label)
4. Alt Cmd Label (Alternative Command Label)
5. Icon

By default the visibility is **normal** meaning the item is always visible regardless of the type of widgets. It can also be **read-only**, meaning the item is not visible in command widgets.



5.2.8.1 Copy and Pasting Enum Labels

From within the enum labels dialog

 Copy  Paste

- Click **Copy**
- Select the point with the sub-type enum with the labels you wish to copy
- If the enums names don't match (Different names and/or size) the paste option will not be available

Cannot paste because the enum names don't match

 Copy  Paste Advanced

- Click **Paste** if the enums match

5.2.9 Bulk duplicate

Bulk duplicate can easily create dozens of rooms, floors, building of the same type, plus devices and its variables(with tags, units, icons, alarms..) in one go.

In order to open the bulk duplicate dialog, select a point (with any type except variables) and choose from the action menus BULK DUPLICATE...

For the point selected, one or more replacement rule(s) can be specified for a duplication. The point can be duplicated as much as the user wants, using the plus icon with the duplicate label.

By default Duplicate children too is checked, this will duplicate the children, grand-children, etc.. of the point using the same replacement rules as the parent point. The parent/child relationships are also maintained.

Example 1: We have a room called Madrid with two devices(One Display and one ClickShare) and we want to make a bulk duplicate to another two rooms(Barcelona and Valencia), we fill the bulk duplicate fields without specifying extra replacement rules as follows:

Bulk Duplicate

Replace	With
Madrid	Barcelona
+ Replacement Rule	
Madrid	Valencia
+ Replacement Rule	
+ Duplicate	

Duplicate Children Too

Duplicate **Cancel**

As a result, all children(devices, variables, etc.) get duplicated with the new names(Barcelona Display, Valencia ClickShare, etc.)

Example 2: We want to duplicate the same Room Madrid to another room Sevilla, and we are going to change the IP addresses of the Display and the ClickShare respectively to 10.201.115.12 and 192.168.2.15, we fill the bulk duplicate fields by adding new replacement rules as follows:

Bulk Duplicate

Replace	With
Madrid	Sevilla
10.201.114.83	10.201.115.12
192.168.1.10	192.168.2.15
+ Replacement Rule	
+ Duplicate	

Duplicate Children Too

Duplicate **Cancel**

As a result, the new duplicated devices got the specified IP addresses and for instance, we can see the new IP address of the Sevilla Display device:

Sevilla Display Setup

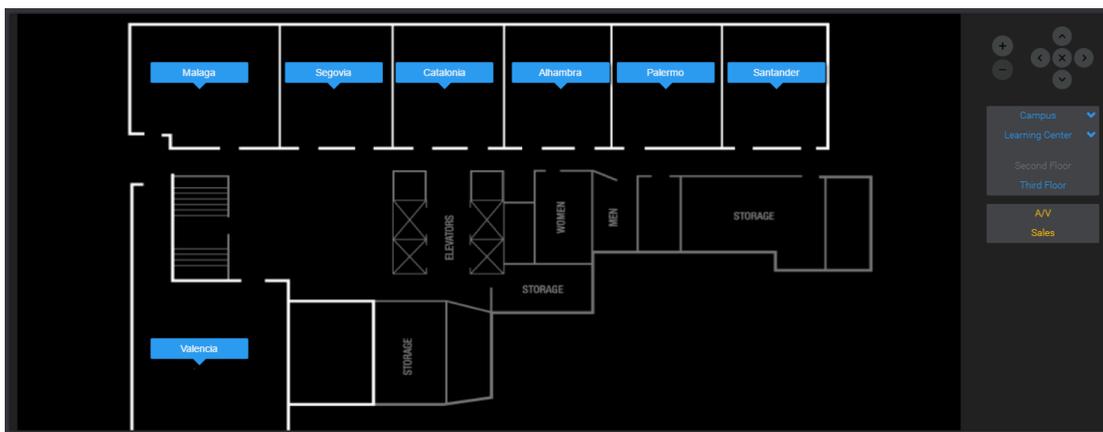
Device Driver: barco_display Version: 1.1.8

Barco Display

host

Host address

5.2.10 Locations



Locations refer to points that display an image (map) on the Home section. They normally have a type of Floor, Building, Site, etc. but can be any type. These points normally are parents to rooms, devices, or other maps, but this may be different depending on need.

To create a location, you will need to upload an asset and create the point.

To upload the asset:

- Go to 'Assets' view. ([More information on the 'Assets' view can be found here.](#)) Click 'New'.
- Give the location a name.
- Click 'Show uploading section'. You may skip this step and the next, if the image you want is already uploaded to your assets.
- Select which folder of your assets the image will be uploaded to ('images' is good choice.) by clicking the magnifying glass. Then click 'Upload Asset' to select which image on your computer you want to use.
- Select the image in your assets folder by clicking the magnifying glass next to 'URL'. This information is auto populated if you uploaded your image.
- Select type 'image', if it's not done automatically. Click 'Save'.

When uploading an image, the following applies:

The Home map is calibrated to images of size 1876 x 830 px, a ratio of ~2.25. Map images with ratios as low as 2.1 are not perfect, but they are workable. Size differences can be significant as well, up to a couple hundred pixels (as long as the ratio is nearly the same).

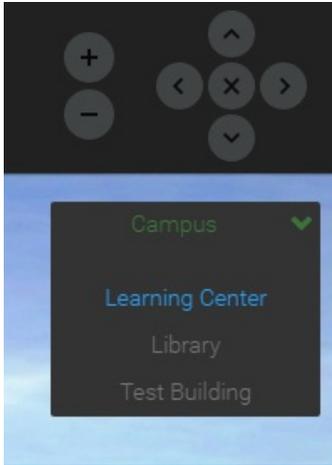
To create the point:

- Go to the 'Points' view and follow the steps for creating a Room listed above, but selecting the appropriate type for this location (Floor, Building, Site, etc.).

- Under 'Maps Fields', select the asset you created from the 'Map' drop down. This point will now be associated with this image in Home.

If a map has no parent, it will be the first map displayed in Home. If multiple maps have no parents, Home displays the map with the lowest ID first. If you would like multiple locations with no parents, but would like to define which map gets shown first you can select the 'Default' check box and that location will be the first displayed.

To switch to other locations, use the breadcrumb menu on the right.



5.2.11 Tasks / User Variables

A task is a sequence of commands which are executed by a Control Server. It is a way to automate actions in Overture.

User variables are variables that are not tied to devices and can be effected by tasks (like a room status).

The creation of tasks and user variables are for advanced programmers only. To create them, [See: Task Programming](#)

5.3 Control Servers View

The Control Servers View is used to define and manage the Control Servers used in the installation. Overture can handle two types of control servers:

- CS2: Overture Control Servers. This is the default and most common choice.
- CS1: Medialon Show Control servers such as Medialon Showmaster and Medialon Manager intended to be used in specific installations.

5.3.1 Configuring CS2 Servers

Servers + NEW SAVE

Name	Type	Description	Reconnection	
ControlServer	CS2	UID: LocalControlServer		

Refresh

Name: *

Type: *

UID: *

Use Password: * (Supported by ControlServer 1.6.0 or greater)

Login Password: *

Confirm Password:

- Name: the name of the control server.
- Type: **CS2** for CS2 control servers.
- UID: the unique ID of the control server. This ID can be obtained when installing the control server.
- Use Password: whether the access to the user interface of the control server should be password protected. Password protection is supported by control servers version 1.6.0 or greater. It is highly recommended to activate this option.
- Password and Confirm Password: the protection password.

Note: CS2 type Control Server passwords follow the password policy defined in the UX Server Setup, see [Password Policy](#) Chapter. Upon update from previous versions to UX 3.7.0/CX 1.9.0, the control server password needs to be changed before being able to login into the control server.

Note: Overture VM includes a pre-configured control server. The default password of this control server is **BarcoBarco** and it is highly recommended to change this password as soon as possible. Several messages will regularly appear in Configurator in order to remind you to change the password if the default password is used by one or several control servers.

5.3.2 Configuring CS1 Servers

Servers + NEW SAVE

Name	Type	Description	Reconnection	
ControlServer	CS2	UID: LocalControlServer		
Showmaster1	CS1	Address: 192.168.1.22 : 92...		

Refresh

Name: *

Type: *

Address: *

Port: *

Shared Group: *

Login User:

Login Password:

Confirm Password:

Proxy Address:

Proxy Port:

- Name: the name of the control server.
- Type: **CS1** for CS1 control servers.
- Address: the IP address or DNS name of the show control server.
- Port: the listen port of the show control server.
- Shared Group: the shared group exposed by the show control server (Overture can only control and monitor objects part of this shared group).
- Login User: (optional) the login username.
- Login Password and Confirm Password: (optional) the login password.
- Proxy Address: the IP address of the Medialon Show Control Proxy.
- Proxy Port: the port of the Medialon Show Control Proxy.

Note: CS1 type Control Server passwords DO NOT follow the password policy defined in the UX Server Setup.

5.4 Plugins View

Plug-Ins View



Search criteria:

Refresh

Name	Version	Latest Version	Type	
lutron_nova_li...	1.1.4	1.1.5	device	✘
barco_projector	1.1.0	1.1.0	device	✘
bss_soundwe...	1.1.3	1.1.3	device	✘
honeywell_spy...	1.1.0	1.1.0	device	✘
overture_email...	1.0.2		notification	✘

Select Plugin Package

Select from Local Device

OR

Browse Online Library

The 'Plugins' view is used to manage your plugins, which can be either drivers for devices or behavior plugins (automation modules). From here, you can add new plugins as well as update existing plugins.

There are two ways to create or update plugins:

You can select a plugin from the online plugin repository by selecting 'Browse Online Library'. All available plugins will populate the list, giving a brief description of themselves. You can also create or update a plugin with a zip file from your computer by selecting 'Select From Local Device'. This will open a file prompt, asking which zip to select.

When creating a new plugin with either method, select 'New' at the top. Once a plugin is selected from the list or a zip is provided, the plugin will show up in the list and be available in the 'Points' view.

To filter the plugins view, click either on "Show Drivers", "Show Behaviors" or both to display the type of plugins you want.



Plug-Ins Repository

Search:

Plug-in type:

Plug-ins:

Name	Type
barco clickshare	device
barco projector hdx-series	device
barco projector f-series	device
barco projector rim-series	device
global cache io	device
axis camera	device
barco projector pg-series	device
barco projector c-series	device
panasonic display lfr-series	device
overture pjlink projector	device
bss soundweb audio	device
honeywell spyder hvac	device
lutron nova lighting	device
barco projector	device

Description:

ClickShare by Barco is a wireless presentation system intended to replace wired setups and enhance meetings, conferences and presentations in a variety of ways. USB ClickShare buttons plug into your device, or for mobile devices an app is downloaded. Then, by clicking the button, you can wirelessly share your devices display through the central video screen. Supported models: CSM-1, CSE-200, CSC-1.

Select Cancel

If a driver is part of the online plugin repository and has a new version, the version field will be red and display the newest version available. To update an existing driver with either method, select the driver in the list. Then provide a new zip, or select the appropriate driver from the online repository.

5.5 Assets View

Assets

+ NEW ✓ SAVE

Search criteria:

Refresh

ID	Name	Type	Uri	
1	Room Panel	control panel	views/co...	✕
2	Projector Panel	control panel	views/co...	✕
3	Lighting Panel	control panel	views/co...	✕
4	Player Panel	control panel	views/co...	✕
5	Camera Panel	control panel	views/co...	✕
6	Audio Panel	control panel	views/co...	✕
7	Display Panel	control panel	views/co...	✕
8	Power Panel	control panel	views/co...	✕
9	HVAC Panel	control panel	views/co...	✕
10	Conference Panel	control panel	views/co...	✕
11	Lift Panel	control panel	views/co...	✕
12	Shades Panel	control panel	views/co...	✕
13	Campus	image	images/c...	✕

Name: *

Uri:

Type:

Edit in GUI Editor

Show uploading section:

The 'Assets' view is used to create maps or control panels, that can then be associated to points in the database.

In this view, you can create a new map or a control panel by clicking 'New'. You will need to select an appropriate image or html file on the server. You can also upload an image or html file directly to the server from this view. However, creating, renaming, deleting, and other management of the assets must be done in [GUI Editor](#).

5.6 Point Types View

+ NEW ✓ SAVE

Types

Refresh

Id	Name	Icon	
variable	Variable		
task	Task		
command	Command		
zone	Zone		✕
floor	Floor		✕
building	Building		✕
country	Country		✕
city	City		✕
site	Site		✕
room	Room	🏠	
device	Device	⚙️	

Id: *

Name: *

Icon:

Timezone:

The Point Types View is used to configure built-in point types as well as defining custom point types. These types are used to define your organisation structure, combined with the parent-child relationship between points (see [Parent-Child-Sibling Relationships](#)). Adjustable parameters for a point type are:

- Name: the human name of the type, ex: Building
- Icon: the icon used to represent this type
- Timezone: whether this type has the option to enable selecting a timezone for a point.

Note: Certain types do not allow changing the name, or does not offer the timezone option.

5.7 Custom Fields

In Overture Configurator, you can create infinite number of meta data fields for your rooms or your devices to fit any types of business scenario...

For example: You need to set a date stamp on devices when they have received a service maintenance or to trace AV assets using their serial numbers.

5.7.1 Create your Custom Fields

Select "Views" in top menu in Overture Configurator and select "Custom Fields" from the list:

The screenshot shows the Overture Configurator interface. At the top, there is a navigation bar with the Overture logo, a search bar, and a user profile for John Doe. A dropdown menu is open, showing various categories: Points Management, Access Rights, Utilities, and Settings. Under Points Management, 'Custom Fields' is highlighted in red. Below the menu, a table lists existing custom fields: Serial Number, Warranty Expiration, ID, and Mac Address. To the right, there are '+ NEW' and 'SAVE' buttons.

Overture allows you to create a new "Custom Field", name it, select the field type (String or Date), and then associate it to any point's type like Building, Device, Room, etc. and save it.

The screenshot shows the Overture Configurator interface for creating a new custom field. The top navigation bar is the same as in the previous screenshot. Below it, the 'Custom Fields' section is active, showing a table of existing fields and a form for creating a new one. The table has columns for Name, Type, and Point Types. The form has fields for Name, Type, and Point Types. The 'Point Types' dropdown is open, showing a list of options including Building, City, Command, Country, Device, Floor, Room, Site, Task, Variable, and Zone.

Name	Type	Point Types	
Serial Number	String	Device	✘
Warranty Expiration	Date	Device	✘
ID	String	Room	✘
Mac Address	String	Device	✘

5.7.2 Editing Points' Custom Fields

Select Points from the "Views" menu. Select any points that have your new Custom Field associated with from the list of points. On the right panel you have a new section called "Custom Fields: +"

Expand the section by clicking on the "+" sign and fill in the values for the Custom Fields. "Save".

Points

+ NEW DUPLICATE BULK DUPLICATE... SAVE

Multi Selection

Save grid layout Refresh

Name	Variable Name	Type	Sub-type	Unit	Parent	Cont
Paris Light Presets	Paris_Light.Presets	Variable	Enum		Paris Light	Local
Paris Light Status	Paris_Light.Status	Variable	Enum		Paris Light	Local
Paris Light Level Channel 1	Paris_Light.LevelChannel1	Variable	Integer		Paris Light	Local
Paris Light Level Channel 3	Paris_Light.LevelChannel3	Variable	Integer		Paris Light	Local
Paris Light Level Channel 2	Paris_Light.LevelChannel2	Variable	Integer		Paris Light	Local
Paris Projector	Paris_Projector	Device	Projector		Floor 1 Paris	Local
Paris Projector Activity	Paris_Projector.Activity	Variable	Enum		Paris Proje...	Local
Paris Projector Power	Paris_Projector.Power	Variable	Enum		Paris Proje...	Local
Paris Projector Status	Paris_Projector.Status	Variable	Enum		Paris Proje...	Local
Paris Projector Shutter	Paris_Projector.Shutter	Variable	Enum		Paris Proje...	Local
Paris Projector Sources	Paris_Projector.Sources	Variable	Enum		Paris Proje...	Local
Paris Projector Contrast	Paris_Projector.Contrast	Variable	Integer		Paris Proje...	Local
Paris Projector Brightness	Paris_Projector.Brightness	Variable	Integer		Paris Proje...	Local
Paris Projector Hours Lamp 1	Paris_Projector.HoursLamp1	Variable	Integer	Hours	Paris Proje...	Local
Paris Projector Temperature	Paris_Projector.Temperature	Variable	Integer		Paris Proje...	Local
Paris AVPresets	Paris_AVPresets	Variable	Enum		Floor 1 Paris	Local
Floor 1 Paris Activity	Floor1_Paris.Activity	Variable	Enum		Floor 1 Paris	Local
Floor 1 Paris Power	Floor1_Paris.Power	Variable	Enum		Floor 1 Paris	Local

Point Order:

Type: Device

Sub-type: Projector

Driver Setup: Projector Simulator

Parent: Floor 1 Paris

Server: LocalControlServer

Roles: Press SpaceBar

Tags: Type or SpaceBar

Metadata: [Edit metadata](#)

Custom Fields: +

Serial Number: S-45679813

Warranty Expiration: 2020-08-31

Mac Address:

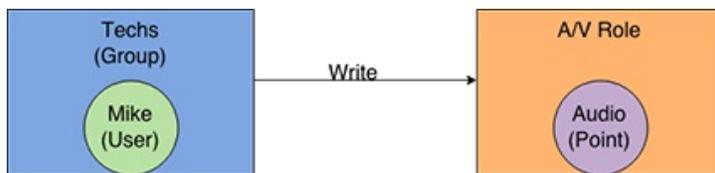
5.8 Access Rights

Overture's built in access rights allow various level of controls to help both security and organization of users on the system.

Access rights have three major components:

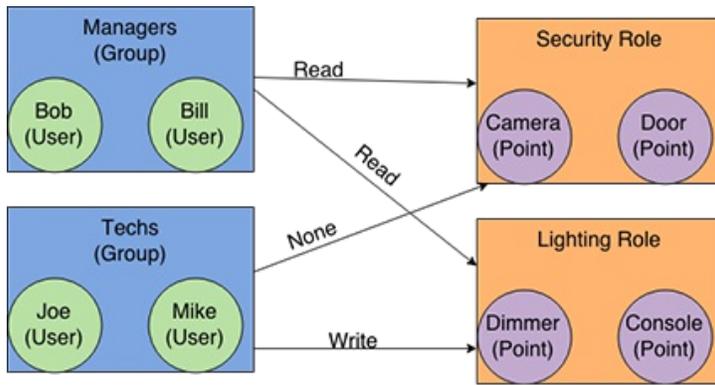
- Users are the people on your system. You can give each person who will access your system a separate username and password to login.
- Groups are groups of users. Rather than needing to assign read/write privileges for each user, Overture assigns those privileges to each group of users.
- Roles are definitions, assigned to points in the system, that control whether that point is read/writeable by a group.

5.8.1 Basic Rights



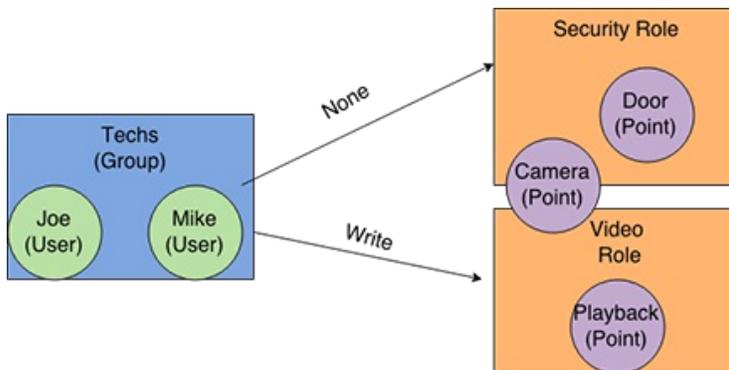
This is a basic example of access rights in the Overture system. Mike is a user who is assigned the Techs group. He will be able to control the Audio point because the Techs group has Write access to the A/V role.

5.8.2 Multiple Users/Groups/Roles



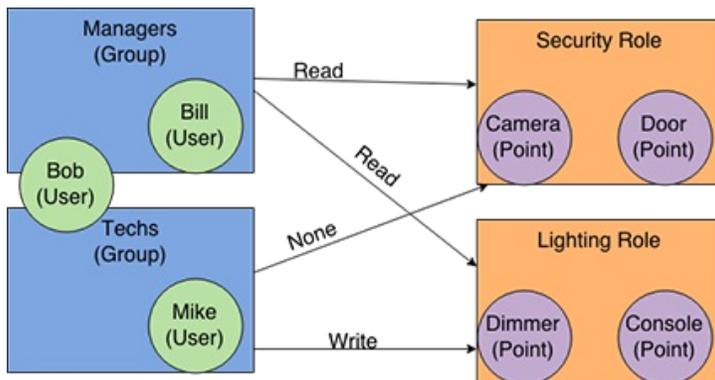
This more advanced example shows multiple groups and multiple roles on the system. Bob and Bill, who are both in the Managers group, can see points that have the Security and Lighting roles but cannot control them. Joe and Mike, both in the Techs group, can control points with the Lighting role but cannot see points with the Security role.

5.8.3 One Point With Multiple Roles



This is an example of a point having two roles defined to it. The Techs group needs to be able to control the Camera point to be able to fix it but doesn't need to be able to see the Door point. Groups can access points if any of the point's Roles match the Group's access rights.

5.8.4 One User In Multiple Groups



This example shows how a user can be assigned to two different groups that each have access to different roles. Bill is a user who can only see the Dimmer point, but not change it. Mike can change the Dimmer point, but cannot see the Cameras point. Bob is a member of both the Managers group and the Techs group, allowing him to change the Dimmer point (via the Write access of the Techs group) and also see the Camera point (via the Read access of the Managers

group).

5.8.5 Users/Groups

Under 'Users' in 'Views', a list of users on the system is shown. You can create new users for the system here or change passwords for current users in the system.

Creating groups or adding users to existing groups is done in the 'Groups' section of 'Views'. You should use groups to organize users who will need similar access rights on the system. (IT, Maintenance, Executives, etc.)

5.8.6 Roles

Inside the 'Roles' section of 'Views', is where you will create new roles and define how each group can interact with it.

- None: Points with this role will not show up in any GUI for that group.
- Read: Users are able to see the point and its value but are not able to change it in any way. Example: Seeing what the audio volume is set at, but not being able to change it.
- Write: Users have full control of this point in any GUI that it's in.

You will also see options related to alarms:

- See: The group can see alarms triggered by points with this role, but cannot affect them in any way.
- Acknowledge: The group can acknowledge an alarm, clearing it from the current list of active alarms.

If group has neither of the alarm items selected, alarms will not show up for points associated with that role.

As well as user defined roles, there is an already created 'System' role that changes how users interact with the Overture software itself. Inside of this role you can determine if a user can see the Configurator, GUI Editor, or change points inside of the Configurator. The following defines what each item means:

SYSTEM ROLE

	EDIT	CONFIG	FULL
View Configurator (read only) (without server config & backups)	✓	✓	✓
Edit panels/pages in Editor	✓	✓	✓
Edit the configuration (without users, groups & roles management)	✗	✓	✓
Unlimited access to points & alarms without worry about specific roles	✗	✓	✓
See backups list and manually trigger	✗	✗	✓
Edit users, groups & roles	✗	✗	✓
Edit server configuration	✗	✗	✓

5.8.7 Identity Providers

Overture can federate existing external user databases, Overture support ADFS (Active Directory Federation Services) and LDAP (Lightweight Directory Access Protocol) In order to create an identity provider, click 'Identity Providers' under 'Access Rights'.

5.8.7.1 ADFS

ADFS rely on the SAML v2.0 protocol in order to authenticate and authorize users. To begin configuring an ADFS as Identity provider(IDP), click 'New' and select ADFS as a Type.

You'll see the following configuration options available:

Configuration	Description
---------------	-------------



Configuration	Description
Alias	The alias is a unique identifier for an identity provider. It is used to reference an identity provider internally. Some protocols such as SAML require a redirect URI or callback url in order to communicate with an identity provider. In this case, the alias is used to build the redirect URI. Every single identity provider must have an alias
Display Name	Friendly Name for the ADFS provider
Enabled	Turn the provider on/off
Store Tokens	Whether or not to store the token received from the identity provider.
Stored Tokens Readable	Whether or not users are allowed to retrieve the stored identity provider token. This also applies to the broker client-level role read token
Trust Email	If the identity provider supplies an email address this email address will be trusted.
Hide on Login Page	When this switch is on, this provider will not be shown as a login option on the login page.
First Login Flow	This is the authentication flow that will be triggered for users that log into Overture through this IDP for the first time ever.
Post Login Flow	Authentication flow that is triggered after the user finishes logging in with the external identity provider.
Single Sign-On Service URL	This is a required field and specifies the SAML endpoint to start the authentication process. If your SAML IDP publishes an IDP entity descriptor, the value of this field will be specified there.
Single Logout Service URL	This is an optional field that specifies the SAML logout endpoint. If your SAML IDP publishes an IDP entity descriptor, the value of this field will be specified there.
Backchannel Logout	Enable if your SAML IDP supports backchannel logout
NameID Policy Format	Specifies the URI reference corresponding to a name identifier format. Defaults to urn:oasis:names:tc:SAML:2.0:nameid-format:persistent.
HTTP-POST Binding Response	When Overture responds to any SAML requests sent by the external IDP, which SAML binding should be used? If set to off, then the Redirect Binding will be used.
HTTP-POST Binding for AuthnRequest	When Overture requests authentication from the external SAML IDP, which SAML binding should be used? If set to off, then the Redirect Binding will be used.
Want AuthnRequests Signed	If true, the external SAML IDP expect a signed request
Want Assertions Signed	If true, the service provider expect a signed assertion
Want Assertions Encrypted	If true, the service provider expect an encrypted assertion
Force Authentication	Indicates that the user will be forced to enter in their credentials at the external IDP even if they are already logged in.

P 115 / 270

Configuration	Description
Validate Signature	Whether or not Overture should expect that SAML requests and responses from the external IDP be digitally signed.
Validating X509 Certificate	The public certificate that will be used to validate the signatures of SAML requests and responses from the external IDP.

Example of an ADFS set up using Windows Server 2012 R2:

Type:

Redirect URI:

Alias*:

Display Name:

Enabled:

Store Tokens:

Stored Tokens Readable:

Trust Email:

Account Linking Only:

Hide on Login Page:

First Login Flow:

Post Login Flow:

SAML Config

Single Sign-On Service URL*:

Single Logout Service URL:

Backchannel Logout:

NameID Policy Format:

HTTP-POST Binding Response:

HTTP-POST Binding for AuthnRequest:

HTTP-POST Binding Logout:

Want AuthnRequests Signed:

Once you configure ADFS, it will appear on the Overture login page.



Mappers

You can import the SAML metadata provided by ADFS you are authenticating. This allows you to extract user profile metadata and other information so that you can make it available to Overture. Below there is a list with all mapper types supported and its form fields:

Username Template Importer

Used to format the username to import.

Field Name	Description
Template	Use to format the username to import. Substitutions are enclosed in \${}. For example: '\${ALIAS}.\${NAMEID}'. ALIAS is the provider alias. NAMEID is that SAML name id assertion. ATTRIBUTE.\ references a SAML attribute where name is the attribute name or friendly name

Hardcoded User Session Attribute

When user is imported from provider, hardcode a value to a specific user session attribute.

Field Name	Description
User Session Attribute	Name of user session attribute you want to hardcode
User Session Attribute Value	Value you want to hardcode

Attribute Importer

Import declared SAML attribute if it exists in assertion into the specified user property or attribute.

Field Name	Description
Attribute Name	Name of attribute to search for in assertion. A full list of the attribute names can be found here: https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/technical-reference/the-role-of-claims#what-are-claim-types . You can leave this blank and specify a friendly name instead.
Friendly Name	Friendly name of attribute to search for in assertion. You can leave this blank and specify a name instead

Field Name	Description
User Attribute Name	User attribute name to store SAML attribute. Use email, lastName, and firstName to map to those predefined user properties

Hardcoded Attribute

When user is imported from provider, hardcode a value to a specific user attribute.

Field Name	Description
User Attribute	Name of user attribute you want to hardcode
User Attribute Value	Value you want to hardcode

SAML Attribute to Group

If a claim exists, grant the user the specified group.

Field Name	Description
Attribute Name	Name of attribute to search for in assertion. A full list of the attribute names can be found here: https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/technical-reference/the-role-of-claims#what-are-claim-types . You can leave this blank and specify a friendly name instead
Friendly Name	Friendly name of attribute to search for in assertion. You can leave this blank and specify a name instead
Attribute Value	Value the attribute must have. If the attribute is a list, then the value must be contained in the list
Group	Overture Group Name to grant to user

Username Generation

In order to login to ADFS, the user must use the user logon name that can be the email address or the pre-Windows 2000 logon name provided in this format: domain_name\username

Example: 14april\johsm

5.8.7.2 LDAP

Overture comes with a built-in LDAP/AD provider, it is possible to federate just one LDAP server. You can map LDAP user attributes into the Overture common model. By default it maps username, email, first name and last name, but you are free to configure additional mappers(For example groups, see below). The LDAP provider supports password validation via LDAP/AD protocols and different storage and synchronization modes.

By default, Overture will import users from LDAP into the local Overture user database. This copy of the user is either synchronized on demand, or through a periodic background task.

To begin configuring LDAP as Identity provider(IDP), click 'New' and select LDAP as a Type.

You'll see the following configuration options available:

Configuration	Description
Console Display Name	Name used when this provider is referenced in the admin console
Priority	The priority of this provider when looking up users or for adding registrations.
Import Users	If true, LDAP users will be imported into Overture DB and synced via the configured sync policies
Sync Registrations	Does your LDAP support adding new users? Enable if you want new users created by Overture to be added to LDAP.
Vendor	LDAP vendor (provider)
Username LDAP attribute	Name of LDAP attribute, which is mapped as Overture username. For many LDAP server vendors it can be 'uid'. For Active directory it can be 'sAMAccountName' or 'cn'. The attribute should be filled for all LDAP user records you want to import from LDAP to Overture
RDN LDAP attribute	RName of LDAP attribute, which is used as RDN (top attribute) of typical user DN. Usually it's the same as Username LDAP attribute, however it's not required. For example for Active directory it's common to use 'cn' as RDN attribute when username attribute might be 'sAMAccountName'
UUID LDAP attribute	Name of LDAP attribute, which is used as unique object identifier (UUID) for objects in LDAP. For many LDAP server vendors it's 'entryUUID' however some are different. For example for Active directory it should be 'objectGUID'. If your LDAP server really doesn't support the notion of UUID, you can use any other attribute, which is supposed to be unique among LDAP users in tree. For example 'uid' or 'entryDN'
User Object Classes	All values of LDAP objectClass attribute for users in LDAP divided by comma. For example: 'inetOrgPerson, organizationalPerson'. Newly created Overture users will be written to LDAP with all those object classes and existing LDAP user records are found just if they contain all those object classes
Connection URL	Connection URL to your LDAP server
Users DN	Full DN of LDAP tree where your users are. This DN is parent of LDAP users. It could be for example 'ou=users,dc=example,dc=com' assuming that your typical user will have DN like 'uid=john,ou=users,dc=example,dc=com'
Authentication Type	LDAP Authentication type. Right now just 'none' (anonymous LDAP authentication) or 'simple' (Bind credential + Bind password authentication) mechanisms are available
Bind DN	DN of LDAP admin, which will be used by Overture to access LDAP server
Bind Credential	Password of LDAP admin
Custom User LDAP Filter	Additional LDAP Filter for filtering searched users. Leave this empty if you don't need additional filter. Make sure that it starts with '(' and ends with ')'
Search Scope	For one level, we search for users just in DNs specified by User DNs. For subtree, we search in whole of their subtree. See LDAP documentation for more details
Use Truststore SPI	Specifies whether LDAP connection will use the truststore SPI with the truststore configured in standalone.xml/domain.xml. 'Always' means that it will always use it. 'Never' means that it won't use it. 'Only for Idaps' means that it will use if your connection URL use Idaps. Note even if standalone.xml/domain.xml is not configured, the default Java cacerts or certificate specified by 'javax.net.ssl.trustStore' property will be used

Configuration	Description
Connection Pooling	Does Overture should use connection pooling for accessing LDAP server
Connection Timeout	LDAP Connection Timeout in milliseconds
Read Timeout	LDAP Read Timeout in milliseconds. This timeout applies for LDAP read operations
Pagination	Does the LDAP server support pagination
Allow Kerberos authentication	Enable Kerberos/SPNEGO authentication in Overture with users data provisioned from LDAP
Use Kerberos For Password Authentication	Use Kerberos login module for authenticate username/password against Kerberos server instead of authenticating against LDAP server with Directory Service API
Batch Size	Count of LDAP users to be imported from LDAP to Overture within single transaction
Periodic Full Sync	Does periodic full synchronization of LDAP users to Overture should be enabled or not
Full Sync Period:	Period for full synchronization in seconds
Periodic Changed Users Sync	Does periodic synchronization of changed or newly created provider users to Overture should be enabled or not
Changed Users Sync Period	Period for synchronization of changed or newly created LDAP users in seconds
Cache Policy	Cache Policy for this storage provider. 'DEFAULT' is whatever the default settings are for the global user cache. 'EVICT_DAILY' is a time of day every day that the user cache will be invalidated. 'EVICT_WEEKLY' is a day of the week and time the cache will be invalidated. 'MAX-LIFESPAN' is the time in milliseconds that will be the lifespan of a cache entry

Sync of LDAP users

If the import is enabled, the LDAP Provider will automatically take care of synchronization (import) of needed LDAP users into the Overture local database. As users log in, the LDAP provider will import the LDAP user into the Overture database and then authenticate against the LDAP password. This is the only time users will be imported.

If you want to sync all LDAP users into the Overture database, you may configure and enable the Sync Settings of the LDAP provider you configured. There are 2 types of synchronization:

- Periodic Full sync: This will synchronize all LDAP users into Overture DB. Those LDAP users, which already exist in Overture and were changed in LDAP directly will be updated in Overture DB
- Periodic Changed users sync: When syncing occurs, only those users that were created or updated after the last sync will be updated and/or imported.

The best way to handle syncing is to click the Synchronize all users button when you first create the LDAP provider, then set up a periodic sync of changed users. The configuration page for your LDAP Provider has several options to support you.

Mappers

You can import user data from LDAP you are authenticating. This allows you to extract user profile data and other



information so that you can make it available to Overture. Below there is a list with all mapper types supported:

All LDAP mappers have the following form fields in common:

Field Name	Description
Name	Name of the mapper
Mapper Type	Used to map single attribute from LDAP user to attribute of Overture
User Model Attribute	Name of the User property or attribute you want to map the LDAP attribute into. For example 'firstName', 'lastName', 'email', etc.

[msad-user-account-control-mapper](#)

Mapper specific to MSAD. It's able to integrate the MSAD user account state into Overture account state (account enabled, password is expired etc). It's using userAccountControl and pwdLastSet MSAD attributes for that. For example if pwdLastSet is 0, the Overture user is required to update password, if userAccountControl is 514 (disabled account) the Overture user is disabled as well etc. Mapper is also able to handle exception code from LDAP user authentication.

Field Name	Description
LDAP Attribute	Name of mapped attribute on LDAP object. For example 'cn', 'sn', 'mail', 'street' etc.
Password Policy Hints Enabled	Applicable just for writable MSAD. If on, then updating password of MSAD user will use LDAP_SERVER_POLICY_HINTS_OID extension, which means that advanced MSAD password policies like 'password history' or 'minimal password age' will be applied. This extension works just for MSAD 2008 R2 or newer.

[msad-lds-user-account-control-mapper](#)

Mapper specific to MSAD LDS. It's able to integrate the MSAD LDS user account state into Overture account state (account enabled, password is expired etc). It's using msDS-UserAccountDisabled and pwdLastSet MSAD attributes for that. For example if pwdLastSet is 0, the Overture user is required to update password, if msDS-UserAccountDisabled is 'TRUE' the Overture user is disabled as well etc. Mapper is also able to handle exception code from LDAP user authentication.

[group-ldap-mapper](#)

Used to map group mappings of groups from some LDAP DN to Overture group mappings.

Field Name	Description
LDAP Groups DN	LDAP DN where are groups of this tree saved. For example 'ou=groups,dc=example,dc=org'
Group Name LDAP Attribute	Name of LDAP attribute, which is used in group objects for name and RDN of group. Usually it will be 'cn'. In this case typical group/role object may have DN like 'cn=Group1,ou=groups,dc=example,dc=org'
Group Object Classes	Object class (or classes) of the group object. It's divided by comma if more classes needed. In typical LDAP deployment it could be 'groupOfNames'. In Active Directory it's usually 'group'
Preserve Group Inheritance	Flag whether group inheritance from LDAP should be propagated to Overture. If false, then all LDAP groups will be mapped as flat top-level groups in Overture. Otherwise group inheritance is preserved into Overture, but the group sync might fail if LDAP structure contains recursions or multiple parent groups per child groups



Field Name	Description
Ignore Missing Groups	Ignore missing groups in the group hierarchy
Membership LDAP Attribute	Name of LDAP attribute on group, which is used for membership mappings. Usually it will be 'member'. However when 'Membership Attribute Type' is 'UID' then 'Membership LDAP Attribute' could be typically 'memberUid'
Membership Attribute Type	DN means that LDAP group has its members declared in form of their full DN. For example 'member: uid=john,ou=users,dc=example,dc=com'. UID means that LDAP group has its members declared in form of pure user uids. For example 'memberUid: john'
Membership User LDAP Attribute	Used just if Membership Attribute Type is UID. It is name of LDAP attribute on user, which is used for membership mappings. Usually it will be 'uid'. For example if value of 'Membership User LDAP Attribute' is 'uid' and LDAP group has 'memberUid: john', then it is expected that particular LDAP user will have attribute 'uid: john'
LDAP Filter	LDAP Filter adds additional custom filter to the whole query for retrieve LDAP groups. Leave this empty if no additional filtering is needed and you want to retrieve all groups from LDAP. Otherwise make sure that filter starts with '(' and ends with ')'
Mode	LDAP_ONLY means that all group mappings of users are retrieved from LDAP and saved into LDAP. READ_ONLY is Read-only LDAP mode where group mappings are retrieved from both LDAP and DB and merged together. New group joins are not saved to LDAP but to DB. IMPORT is Read-only LDAP mode where group mappings are retrieved from LDAP just at the time when user is imported from LDAP and then they are saved to Overture
User Groups Retrieve Strategy	Specify how to retrieve groups of user. LOAD_GROUPS_BY_MEMBER_ATTRIBUTE means that roles of user will be retrieved by sending LDAP query to retrieve all groups where 'member' is our user. GET_GROUPS_FROM_USER_MEMBEROF_ATTRIBUTE means that groups of user will be retrieved from 'memberOf' attribute of our user. Or from the other attribute specified by 'Member-Of LDAP Attribute'. LOAD_GROUPS_BY_MEMBER_ATTRIBUTE_RECURSIVELY is applicable just in Active Directory and it means that groups of user will be retrieved recursively with usage of LDAP_MATCHING_RULE_IN_CHAIN Ldap extension
Member-Of LDAP Attribute	Used just when 'User Roles Retrieve Strategy' is GET_GROUPS_FROM_USER_MEMBEROF_ATTRIBUTE. It specifies the name of the LDAP attribute on the LDAP user, which contains the groups, which the user is member of. Usually it will be 'memberOf' and that's also the default value
Mapped Group Attributes	List of names of attributes divided by comma. This points to the list of attributes on LDAP group, which will be mapped as attributes of Group in Overture. Leave this empty if no additional group attributes are required to be mapped in Overture
Drop non-existing groups during sync	If this flag is true, then during sync of groups from LDAP to Overture, we will keep just those Overture groups, which still exists in LDAP. Rest will be deleted.

In order to synchronize LDAP groups to Overture groups or to synchronize Overture groups to LDAP groups, after the proper mapper configuration, the following buttons can be used:

Sync LDAP Groups To Keycloak
Sync Keycloak Groups To LDAP

ID:

Name*:

Mapper Type:

LDAP Groups DN:

user-attribute-ldap-mapper

Used to map single attribute from LDAP user to attribute of UserModel in Overture DB.

Field Name	Description
LDAP Attribute	Name of mapped attribute on LDAP object. For example 'cn', 'sn', 'mail', 'street' etc.
Read Only	Read-only attribute is imported from LDAP to UserModel, but it's not saved back to LDAP when user is updated in Overture
Always Read Value From LDAP	If on, then during reading of the LDAP attribute value will always used instead of the value from Overture
Is Mandatory In LDAP	If true, attribute is mandatory in LDAP. Hence if there is no value in Overture, the empty value will be set to be propagated to LDAP
Is Binary Attribute	Should be true for binary LDAP attributes

full-name-ldap-mapper

Used to map full-name of user from single attribute in LDAP (usually 'cn' attribute) to firstName and lastName attributes of UserModel in Overture.

Field Name	Description
LDAP Full Name Attribute	Name of LDAP attribute, which contains fullName of user. Usually it will be 'cn'
Read Only	For Read-only is data imported from LDAP to Overture, but it's not saved back to LDAP when user is updated in Overture
Write Only	For Write-only is data propagated to LDAP when user is created or updated in Overture. But this mapper is not used to propagate data from LDAP back into Overture. This setting is useful if you configured separate firstName and lastName attribute mappers and you want to use those to read attribute from LDAP into Overture.

hardcoded-ldap-attribute-mapper

This mapper is supported just if syncRegistrations is enabled. When new user is registered in Overture, he will be written to the LDAP with the hardcoded value of some specified attribute.



Field Name	Description
LDAP Attribute Name	Name of the LDAP attribute, which will be added to the new user during registration
LDAP Attribute Value	Value of the LDAP attribute, which will be added to the new user during registration. You can either hardcode any value like 'foo' but you can also use some special tokens. Only supported token right now is '\${RANDOM}', which will be replaced with some randomly generated String

Username Generation

In order to login to LDAP, the user must use the generated username, by default it's the imported Full Name if it exists, otherwise it will be: Firsname Lastname

5.9 Logs

Inside 'System Logs' users can see what is happening within the Overture system itself. You can search to see just what certain users are doing, or just what points are being changed. Logs will also help you troubleshoot the system when you aren't getting the expected result you wanted from your control panels or pages. A user's access to the logs are based on their access to the 'System' role, and their access to point's roles.

Applications are logging at different [Log levels](#):

- **Alarm**: Errors, potentially causing issues with the system.
- **Warning**: Unusual, might need attention.
- **Info**: Information, useful to be aware of important operations in the system.
- **Normal**: Regular actions.
- **Debug**: Low-level log of system operations. For example, each variable change has a debug log.
- **Core**: A very low-level log of the system operations. For instance, detailed frames can be log at this level.

Not all logs sent by Overture Clients or UX Server are stored. You can choose from which level you want to start to record logs on [UX Server Config](#) view. Generally, **Normal** level is enough. But if you need some more detailed information, you might want to start recording from Debug Level.

The screenshot shows the Overture Configurator interface. At the top, there is a logo for Overture Configurator, a search bar labeled 'Live Search', and navigation icons for 'Views', a grid, and a user profile. Below this is the 'Server Configuration' section. It is divided into three main panels: 'HTTP Settings', 'Logs Settings', and 'Databases'. The 'Logs Settings' panel is highlighted with a green border. It contains a 'Log level' dropdown menu set to 'Normal' and a 'Log User Info' checkbox which is checked. The 'HTTP Settings' panel shows options for Protocol (HTTP selected), HTTPS Key file, HTTPS Cert file, and a 'Delete Key and Cert Files' button. The 'Databases' panel shows 'Connection retry delay (s)' set to 10, and configuration for 'Main database' and 'Log database' with fields for Address, Db name, and Username.

Log User Info: Enable or disable logging the name of the user at the origin of an action. (Default: Enabled) For privacy reasons you might want to hide *who* is doing *what* in Overture. If enabled, the `user` property of logs will be replaced by `Unknown user`, and the name which might be logged in the log message will be removed.

5.10 Alarms

Alarm Id	Alarm Name	Expression	Delay (seconds)
1	OverTemp	value > 90	
2	LampHours	value > 900	
3	Connection	value == 0	30

Name: * Connection
Expression: * value == 0
Delay (sec): 30
Notifications: AdminOnly Add a notification

Alarms are triggered events that notify users of abnormal system operations. They can be set up for a variety of things including when a projector is at a high temperature, a PLC has gone offline, or a door has been left open for longer than 10 minutes. A user's access to alarms are based on their access to the role associated to that alarm's point.

Alarms can be separated into two parts:

- Setting Alarms Up In The System
- Interacting with Alarms

5.10.1 Setting Up An Alarm

Name: * Connection

Expression: * value == 0

Delay (sec): 30

Notifications: AdminOnly Add a notification

To setup an alarm, go to the 'Alarms' view inside of Configurator.

Next, you will need to define what triggers it. This is known as the alarm expression. The alarm expression is the logic the system uses to see whether an alarm should be triggered and relates directly to the point.

For example, if you are writing an alarm that triggers when a projector is above 90 degrees, your expression would be: `value > 90`. This tells the system whenever the point associated with this alarm has a value over 90, trigger an alarm.

When writing your alarms you will have the following options related to points:

- value: or point.value. This is the value of the point stored in the Control Server. It may be a string, or a number depending on what type of variable the point is. If the point is an enum, the value is the current index of the enum.
- string: or point.string. This is the string associated with the current index of an enum. For example, Projector.Power might be an enum with 'Off' and 'On'. If the Projector is off, the value would be 0, but the string would be 'Off'.

Next, you will need to write the comparison. `<`, `>`, `<=`, `>=`, `==` all work. Then you will need to write a value (For example: 42 or "On").

Expressions, can be combined with `&&` or `||` for AND/OR comparisons. For example, `value > 100 || value < 20`. The alarm would be triggered if the point's value was above 100 or below 20.

An expression can also directly reference a value of another point within the system. For example, you can attach an alarm to `Room1_Projector.Power` and for the alarm expression use `Server1.Room1_Projector.Power.string == "On" && Server1.Room2_Projector.Power.string=="On"`. In that case, the alarm will only trigger if both Projectors are on at the same time.

Note The logs only indicate an alarm to points that are tied to the alarm even if other points are referenced in those alarm

expressions.

Expressions can also use some of the inheritance within the system. If the point tied with this alarm is a device variable, you can check the value or string of any sibling device variables by calling `parent.variableName.value(or.string)`. For example, if you need to check the shutter and power of the projector you can set an alarm to the `Projector.Power` point and use an alarm expression : `string == "On" && parent.Shutter.string == "Closed"`. Now, you have an alarm that will trigger only when both the Projector Power is on, and the shutter is closed.

Alarm expressions also have an advanced function called `replace()` that can be used when you want to check other device variables in the system but also use the alarm expression as a template, so you can apply it to multiple points. For example, if you have multiple rooms that have projectors and occupancy sensors and you want have one complex alarm that monitors these attributes together you can use `replace()`. Your alarm expression would be something like this : `value > 40 && replace('Projector.Temperature', 'HVAC.Occupancy').string == 'Occupied'`

This complex alarm only works when variable names of the points have the same prefix (ex: `Room1_Projector.Temperature` & `Room1_HVAC.Occupancy`). In this example, the alarm must be tied to the `Projector.Temperature` point. When that point's value is above 40 and then the point with a similar name but `HVAC.Occupancy` string is equal to 'Occupied' the alarm will trigger. This allows you to create template alarms for devices that exist in the same room or other scenarios where more information is needed for the alarm.

Inside of the Alarms view, an alarm can have a specific amount of time associated with it before it alerts in the system. For example, the room temperature becoming 90 degrees once for 2 seconds might not matter but staying that way for 2 minutes may indicate a problem.

You can enter a Delay value in seconds to indicate how long the alarm expression must remain true before alerting the system. A floating point number can be given to increase precision.

5.10.2 Assigning Alarms

To assign an alarm to a point, go to the 'Points' view. Click on a point and specify the name of the alarm in the alarms section. Typing a space, will bring up all available alarms. For example, you could specify an OverTemp alarm you've created to `.Temperature` variable points.

The screenshot shows a configuration form for an alarm. The fields are as follows:

- Parent:** A text input field containing "Malaga Projector" with a red 'X' icon and a magnifying glass icon to its right.
- Server:** A dropdown menu showing "ControlServer" with a refresh icon to its right.
- Roles:** A text input field containing "Add a role".
- Tags:** A text input field containing "Add a tag".
- Alarms:** A text input field containing "Over".
- Metadata:** A text input field containing "OverTemp".

Below the form, there is a "Map fields: +" button.

5.10.3 Alarm Notifications

Name: * Admin Email

Type: * Email 

Send To: * Administrators  Add a group of users

Subject: * Overture

Plain Text HTML

Body: *
 %pointname% had
 %alarmname% error on
 %date%.

Send

Alarms that are triggered can automatically send notifications out to users on the system. This is set-up in the Notifications view.

A single alarm can have multiple notifications.

There are a few options when setting up a new notification:

- Name: The name of notification. Alarms can have multiple notifications tagged to it.
- Type: For now, the only type is Email.
- Send To: The Overture user group that should receive the notification.
- Subject: Subject of the notification.
- Plain Text/ HTML: If writing an HTML notification, you will need to write all the HTML. (Since v3.0.3, the format is determined automatically)
- Body: The actual message being sent to the users. This message as well as the subject can use variables, as described in the next section.

5.10.3.1 Variables In The Subject/Body

The subject and body can have the following variables used in them for a more dynamic email:

- %pointname%: The name of the point which has triggered the alarm
- %pointvalue%: The value of the point on which the alarm triggered.
- %alarmname%: The name of the alarm
- %date%: The date the trigger condition has been detected. (Format: [Thu Jun 22 2017 17:47:47 GMT+0000 \(UTC\)](#).)
- %trigger%: The alarm expression

5.10.3.2 Setting Up Email

Type: * email  

Settings

Service: *

Account name:

User Login Name: *

Password: *

Account Email Address:

SMTP Host: *

Port: *

Secure:

Ignore TLS:

Alias: Enter all the values separated by comma (,)

Domain: Enter all the values separated by comma (,)

Auth Method:

TLS:

Client Host Name:

You will need to set-up the SMTP server that Overture should use for sending emails out. You can specify from an existing list of common email services, or you specify a custom email service.

This setting is common for all email notifications. If you apply a change to any of the notifications, it will be used for all existing notifications.

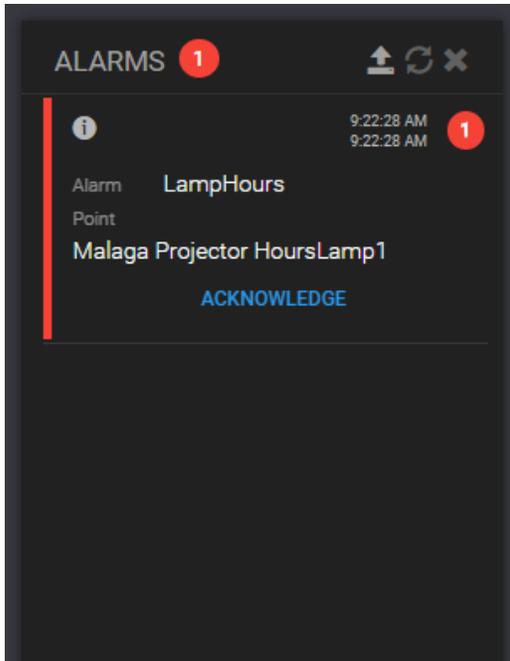
The follow options are available for a custom email service:

- Account: Sender name.
- User: Username to login to the email server.
- Password: Password to login to the email server.
- Account Email Address: Sender email address. (Mandatory)
- Host: Host name or IP of the SMTP server. (Mandatory)

- Port: Port for the server. (Mandatory)
- Secure: Whether to use SSL or not.
- Ignore TLS: Turns off STARTTLS if true.
- Alias: Aliases that maybe needed by the email server
- Domains: Domains that maybe needed by the email server.
- Auth Method: Defines the preferred authentication method.
- TLS: Which encryption to use if using TLS.
- Name: Optional hostname of the client, used for identifying to the server, defaults to hostname of the machine.

Once set up, you can send a test email by clicking the send button. This allows you to make sure the email server is working correctly.

5.10.4 Interacting with Alarms



Alarms appear as specialized logs in the system. In Home, they appear in logs, at the top bar, and also on the maps. Clicking that number will take you to the 'Alarms' view of the logs.

Here you can see the alarm name, the time it first happened, the time it last happened, the number of times it has triggered, the name and expression of the alarm, the point the alarm happened on, and the current value of that point.

The same alarm number shows up next to views in Home, in case you have logs hidden. In MagicMenu, the alarms show up as a red number in the title bar.

If you know an item should be shown as an alarm, but isn't, make sure your access rights for that point allow you to see alarms.

Alarms have four different states:

- Triggered and Active: This means the alarm expression has become true, is still true, and no one has acknowledged it.
- Triggered and Non-Active: This means the alarm expression has become true, no one has acknowledged it, but the expression is no longer true.
- Acknowledged and Active: This means the alarm expression has become true, a user has acknowledged it, but the expression is still true. This will clear it from the current list of alarms but the system will know it's still active.
- Acknowledged and Non-Active: This means the alarm expression has become true, is no longer true, and the user has acknowledged it. This clears the alarm from the system. If the alarm is re-triggered it's counted as a new instance of the alarm.

5.11 CSV Import/Ingest

When deploying a new installation, creating all the points within configurator can be a long process. You can gain significant time with embedded [Duplicate](#) and [Bulk Duplicate](#) features from Point's View.

If you've used a Microsoft Excel Project Description File, you can import all your points at once using CSV Ingest in Configurator.

The *Project Description File* is a file based on a template provided by Barco project team, intended to prepare and organize the deployment of new installations.

This file uses macros in order to auto-fill some hidden columns required for the CSV import (ID, Name). When opening the file, you will get a Security Warning which disables the macros, you need to *Enable Content*.

Overture is able to Ingest the information contained in the following tabs of the Excel file:

- *Overture Servers*: Control Servers to create or update.
- *Containers*: All "passive" points, which can have behaviors or not, like rooms and floors.
- *Devices*: All devices including their drivers.

Once ready to import in Overture, export the file from the *Export* tab. This will generate a zip containing 3 files:

- [containers.csv](#)
- [CX.csv](#)
- [devices.csv](#)

Before exporting the excel file, the macros will make sure that all points have an ID and other mandatory fields if they are not already existing. This allows to create new points and control servers but also update existing ones, if you already exported the file before.

To ingest the file, login to [Configurator](#) > [Views](#) > [Utilities](#) > [Ingest](#) > [Import CSV zip file](#).



The files will be verified before being ingested. If any errors are detected, they will appear in the log and the ingest will stop. If you face errors while ingesting, make sure that the excel file does not contain some empty lines that could be used by the macros and generating an invalid entry for Overture. Some columns are hidden by default in the excel file (Generated names, ID), you might need to *Unhide* them to debug your excel file.

A backup is automatically triggered before ingesting the CSV files. If an error occurs during ingestion, a restore of the backup is automatically triggered.

The *ingester* will download the latest published versions of missing drivers if the UX has access to the internet and if the drivers are available in Overture Hub.

5.11.1 Rules

The content of the worksheet have to comply to the following rules:

- control servers, devices, rooms, and other entities must not have duplicate names (either within the worksheet, or between the worksheet and database points which have not been imported previously).
- device types and room types must match types defined in the database. Otherwise, they are ignored.
- device driver names must match the driver name as displayed in Overture Hub "name" column. Otherwise, they are ignored.
- device driver names must refer to existing drivers in Overture Hub. The current user must have read access to these drivers (eg the drivers must be "public"). Otherwise, they are ignored.

5.12 Room Booking

Room booking is the system to connect Overture to your calendar system. The current supported systems are Exchange, Office 365 and Google Suite.

5.12.1 Room Booking configuration overview

The room booking configuration view has several sections, some for which the content is generic for all room booking systems, some for which the content is specific.

- (1) *Calendar Source* Specifies which room booking system will be used, currently are supported
 - Exchange 2010, 2013, 2016
 - Office 365
 - Google Suite (Google Calendar)
- (2) This section content depends on the selected calendar source. It essentially provides connection parameters for the given calendar source
- (3) This section provides a connection log view as well as global parameters for Overture Room Booking feature
- (4) Test connection button as well as configuration management

Room Booking Config

Calendar Source: Office 365 Exchange 2010 Exchange 2013 Exchange 2016 Office 365 Google Apps

Url: https://outlook.office365.com/Ews/Exchange.asmx Authentication: Basic Username: felix@barcooverture.onmicrosoft.com Password: password Please re-enter your password

Pre Meeting Time Offset (in minutes): 5 Post Meeting Time Offset (in minutes): 5 Enable All-Day Meetings:

CLEAR CONFIG SAVE TEST CONNECTION

Select first the Calendar Source you want to use, example:

Calendar Source:

Exchange 2010

5.12.2 Fields for Exchange and Office 365

- [URL](#)
- [Authentication](#)
- [Username](#)
- [Password](#)

Please refer to [Microsoft Exchange Settings](#)

5.12.2.1 URL

Uri:

https://mail.server.com/ews/exchange.asmx

REQUIRED

P 131 / 270

URL to the Exchange server's EWS library. Make sure to include the 'asmx' extension.

5.12.2.2 Authentication

Authentication:

The type of authentication used to connect to the server:

- NTLM
- Basic

5.12.2.3 Username

Username:

REQUIRED

Username used to connect to Exchange

5.12.2.4 Password

Password:

REQUIRED

Password used to connect to Exchange. The password will not be render on page reload. If a password is saved the message "Please re-enter your password" is displayed below the password field:

Password:

Please re-enter your password

In order to save new configuration you need to re-enter your password again before saving

5.12.3 Fields for Google Suite

Feature available from Overture UX 3.8.0

- [Service Account email](#)
- [User email](#)
- [Private Key](#)

Please refer to [Google Suite Settings](#)

5.12.3.1 Service Account email

Service Account Email is the client_email property from the GSuite Service Account key file.

P 132 / 270

Service account email:

ovt-733@overture-room-booking.

5.12.3.2 User email

User Email of the user associated to the Service Account, which has at least read access to the calendars that Overture will need to have access to.

User email:

admin@overture-development.con

5.12.3.3 Private Key

Private Key is the private_key property from the GSuite Service Account key file.

Private key:

.....

5.12.4 Common Room Booking Parameters

- [Pre Meeting Time Offset](#)
- [Post Meeting Time Offset](#)
- [Enable All-Day Meetings](#)

5.12.4.1 Pre Meeting Time Offset

Pre Meeting Time Offset (in minutes) ⓘ

0

Controls the when the meeting start automation will happen. For example if you set 15 (minutes) and have a meeting starting at 7:30am, then the meeting start automation will take effect at 7:15am.

Default value: 0. Meeting will start in the same minute of your actual meeting start time.

5.12.4.2 Post Meeting Time Offset

Post Meeting Time Offset (in minutes) ⓘ

0

Controls the when the meeting end automation will happen. For example if you set 15 (minutes) and have a meeting ending at 9:30am, then the meeting end automation will take effect at 9:45am.

Default value: 0. Meeting will end in the same minute of your actual meeting end time.

5.12.4.3 Enable All-Day Meetings



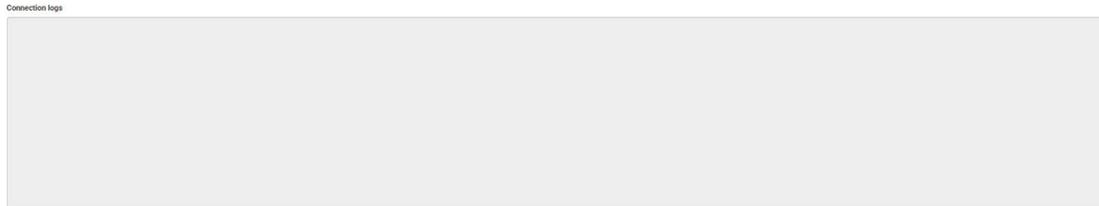
Enable All-Day Meetings 

Controls whether or not All-Day events will trigger the automation

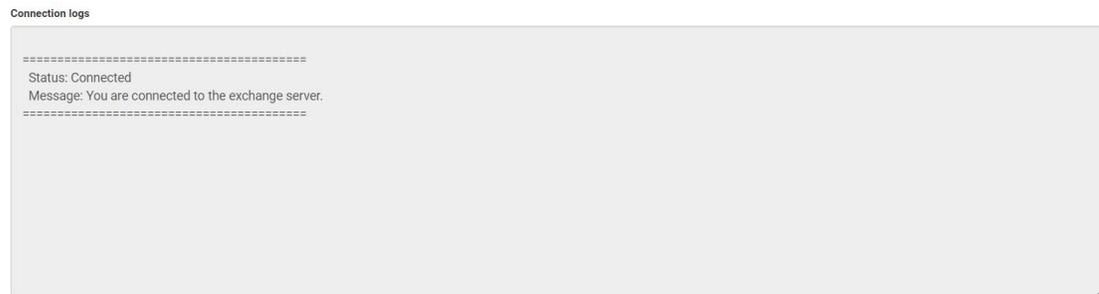
5.12.5 Testing connections



Will display status of the connection in connection logs to the Source Calendar based on the values entered in the form. Utilize this tool before saving your settings. Saving the form does not verify the connection to the Source Calendar. The result of test connection will be displayed in the connection logs section:



Upon filling out the form with the correct information, the status message will be shown as follows:



Once this message is displayed; click the save button and the room booking system will be set up correctly

Testing with Google Suite:

Room Booking Config

Calendar source:	Service account email:	User email:	Private key:	<input type="button" value="✓ TEST CONNECTION"/>
Google Suite ▼	ovt-733@overture-room-booking..	admin@overture-development.con	

Connection logs

5.12.6 Microsoft Exchange Settings

In order for Overture to be able to connect and work properly with Microsoft Exchange, the following settings are needed:

5.12.6.1 Create a Room List Distribution Group



For Microsoft Exchange Rooms to be tracked by Overture (through Microsoft Exchange's API), they need to be part of a Room List Distribution Group. This can be achieved using the following example (using Microsoft Exchange for Office 365):

1. Connect to Microsoft Exchange through PowerShell;
2. Save all rooms in a variable: `$rooms = Get-Mailbox -Filter{(RecipientTypeDetails -eq "RoomMailbox")} | select -ExpandProperty Alias;`
3. Create the Room List Distribution Group using `$rooms` as the value for the `Members` parameter: `New-DistributionGroup -Name "Meeting Rooms" -RoomList -Members $rooms.`

Note to the reader: Please make sure that the `-RoomList` option is set on every Distribution Group that will be used.

- See Microsoft reference for more details:
 - [https://technet.microsoft.com/en-us/library/ee633471\(v=exchg.141\).aspx](https://technet.microsoft.com/en-us/library/ee633471(v=exchg.141).aspx)
 - <https://docs.microsoft.com/en-us/powershell/module/exchange/users-and-groups/New-DistributionGroup?view=exchange-ps>

5.12.6.2 Grant `ReadItems` Access Right to Microsoft Exchange API User

To be able to read meetings information, the Microsoft Exchange API User configured in the Room Booking Configuration, requires to have at least `ReadItems` permission access to the Calendar folder of each Microsoft Exchange Room that will be accessed by Overture. This can be achieved using the following example (using Microsoft Exchange for Office 365):

1. Connect to Microsoft Exchange through PowerShell;
2. Add the permission to the Microsoft Exchange Room `Calendar Folder` that the API User will have access to: `Add-MailboxFolderPermission -Identity newroom@barcooverture.onmicrosoft.com:\Calendar -User felix@barcooverture.onmicrosoft.com -AccessRights ReadItems`

- See Microsoft reference for more details:
 - <https://docs.microsoft.com/en-us/powershell/module/exchange/mailboxes/add-mailboxfolderpermission?view=exchange-ps>

5.12.7 Google Suite Settings

Feature available from Overture UX 3.8.0

Google Suite access is done through a Service Account, for which calendar resources are given read-only access, for the calendars intended to be used by Overture.

5.12.7.1 Pre-requisites

The following steps assume that you have a G-Suite account with a Domain already setup in your organization, and that you have enough access rights required for each step. Overture will use calendar associated to resources, such as room, which are shared with a user which will be used as part of the connection parameters (see [Fields for Google Suite](#))

5.12.7.2 Create a Service Account in Google Cloud

1. Go to Google Cloud Platform, under your Domain and Project, and select Service Account (i.e <https://console.cloud.google.com/iam-admin/serviceaccounts>).
2. Create a Service Account which will be dedicated to Overture access.

Note: if you don't have a project, you'll need to create one into your Domain.

Google Cloud Platform Overture Room Booking Search products and resources

API Library

Google Calendar API

Google

Integrate with Google Calendar using the Calendar API.

ENABLE TRY THIS API

Type
APIs & services

Last updated
9/1/17, 1:29 PM

Category
G Suite

Service name
calendar-
json.googleapis.com

Overview

The Calendar API lets you display, create and modify calendar events as well as work with many other calendar-related objects, such as calendars or access controls.

[Learn more](#)

About Google

Google's mission is to organize the world's information and make it universally accessible and useful. Through products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and YouTube, Google plays a meaningful role in the daily lives of billions of people.

[Tutorials and documentation](#)

5.12.7.5 Provide scoped access in Google Suite to the Service Account

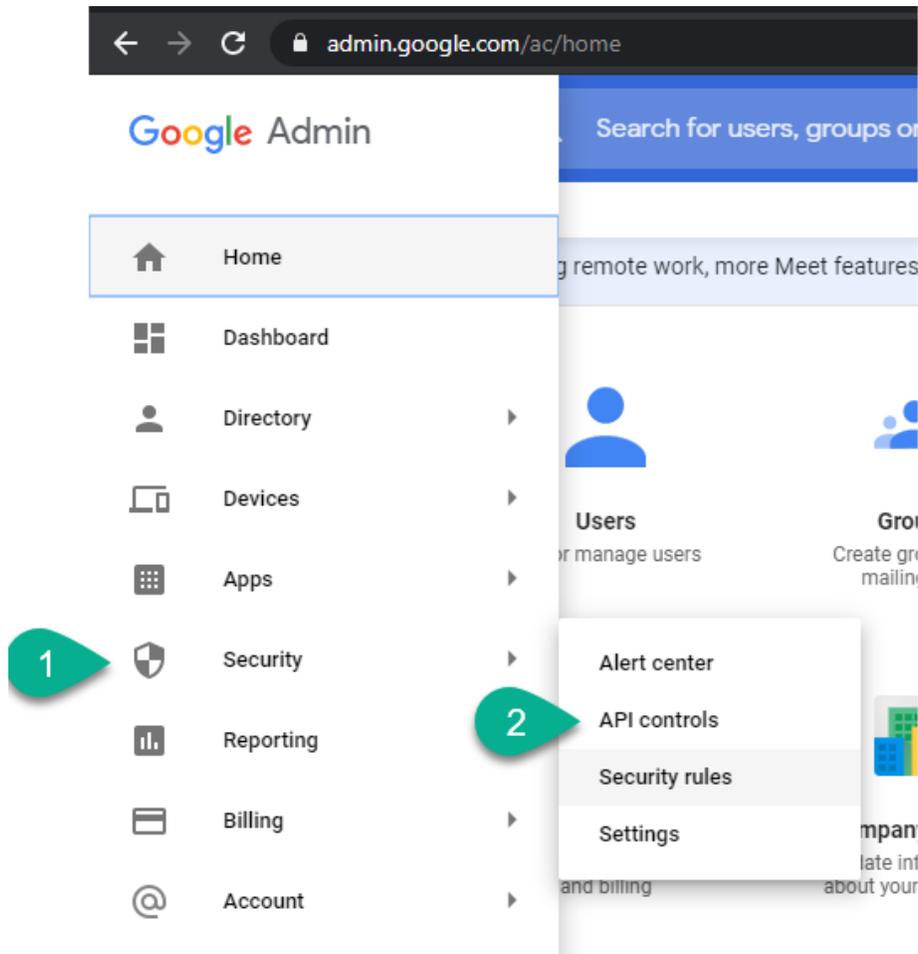
Specific scopes have to be configured for the Service Account. In the following page (<https://admin.google.com/ac/owl/domainwidedelegation>), a new Client Access has to be defined by using the Unique Id of the Service Account (`client_id` property of the Key file download above).

The following scopes must be associated to the Service Account:

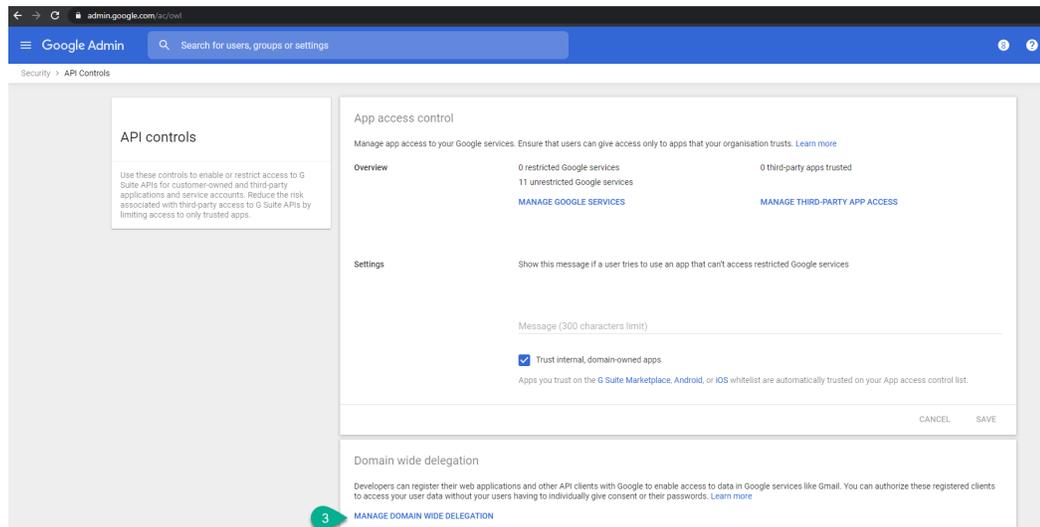
- <https://www.googleapis.com/auth/admin.directory.resource.calendar.readonly>
- <https://www.googleapis.com/auth/calendar.events.readonly>

Steps to assign scopes:

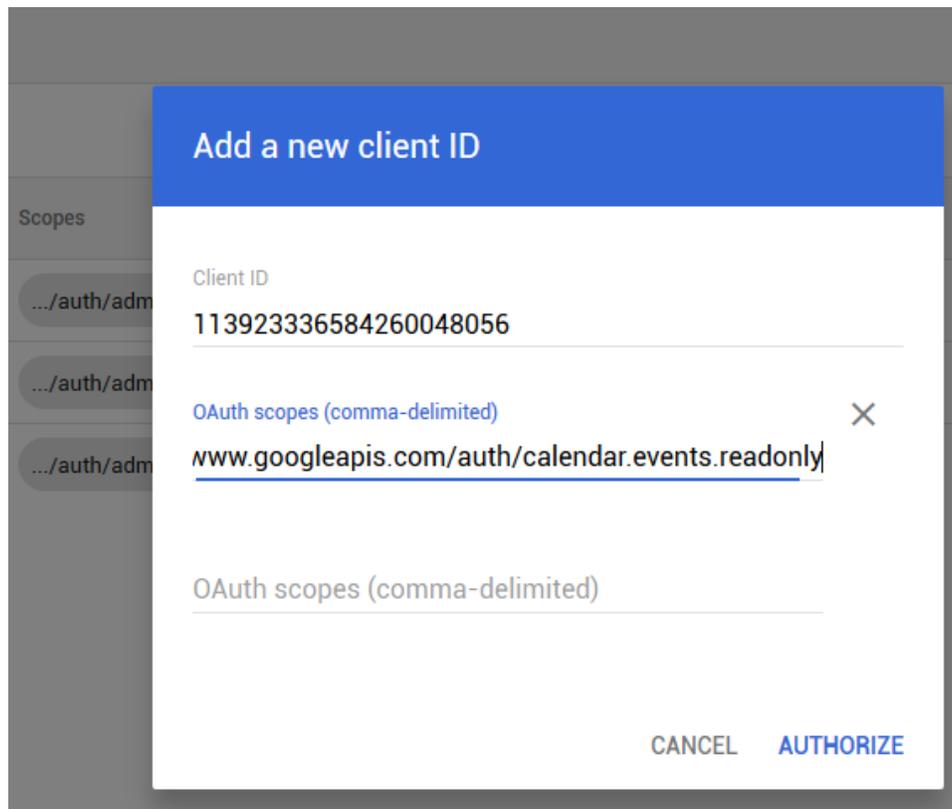
1. Go to <https://admin.google.com/>
2. Select Security -> API Controls



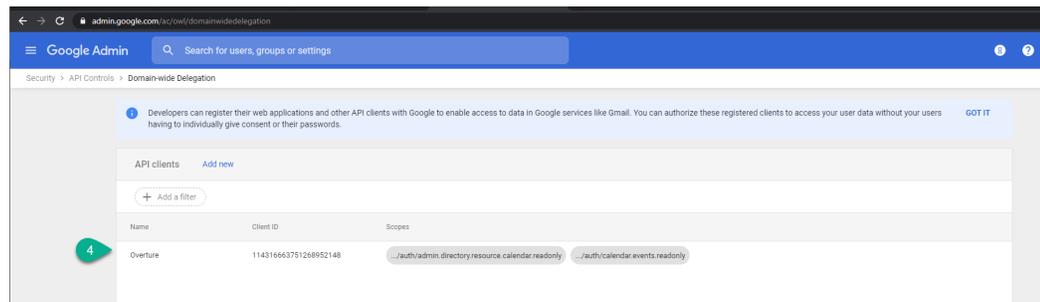
3. Click Manage Domain Wide Delegation



4. Click Add New



5. Enter the Client ID (`client_id` property of the Key file)
6. Enter the authorized scopes using comma separated string:
<https://www.googleapis.com/auth/admin.directory.resource.calendar.readonly>,<https://www.googleapis.com/auth/calendar.events.readonly>
7. Click "Authorize"
8. A new API Client entry should be listed



5.12.7.6 Note for Overture running in a VM

When testing the GSuite connection, the following error may appear:

"Message: Google Suite getRooms error: invalid_grant: Invalid JWT: Token must be a short-lived token (60 minutes) and in a reasonable timeframe. Check your `iat` and `exp` values and use a clock with skew to account for clock differences between systems."

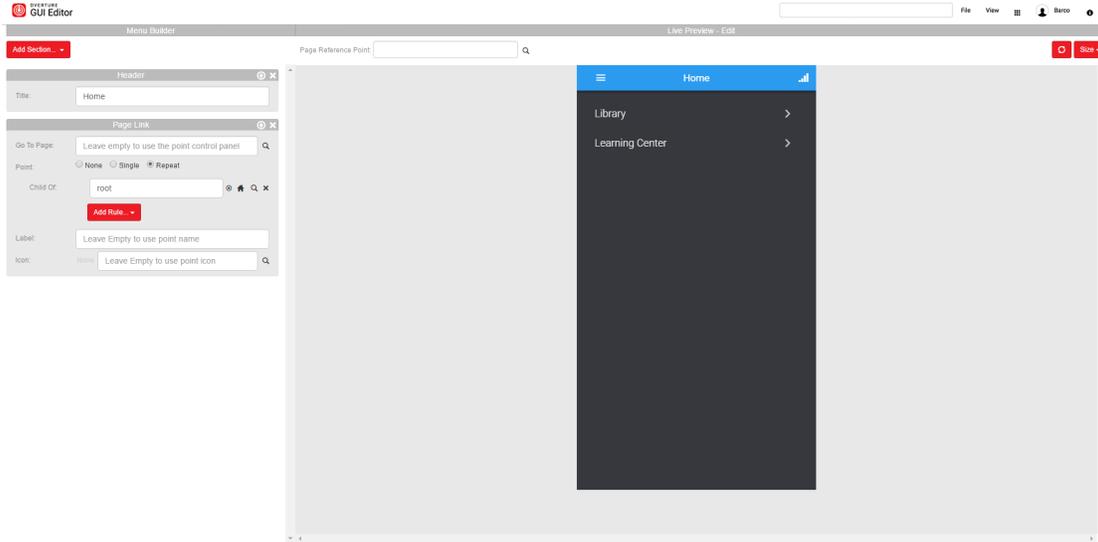
To fix that, an additional NTP server is required in the VM config, ex: 'time.google.com'

Please note that an 'overture_room_calendar' behavior should be added to a point in order for the Room Booking System to keep track of the point's respective meetings. You may refer to the Room Calendar Behavior readme in order to know

how to use this behavior.



6 GUI Editor



The GUI Editor is the asset management tool of Overture. It interacts with the Overture file system and allows for viewing, editing, and creating of templates. There are two different versions of the tool:

- Menu Builder: An easy to use template building wizard.
- Expert Mode: A text editor that allows direct editing of the templates.

It can be accessed by adding 'guieditor/' to your server URL or through the App Switcher (in the top right of the header



). You must have sufficient access rights in order to use GUI Editor (see [Access Rights](#)).

6.1 Assets

Assets are the important pieces to the Overture project and relate directly to how things are shown in the GUI. Every asset, whether it be an HTML file, a CSS document, or an image must be stored in the assets folder of your project.

Assets are stored in your assets folder of the UX Server. The main assets for your project include:

- Images: Images used as maps or displayed inside of html pages. (Use the following formats: .jpeg, .png, .bmp)
- Styles: CSS files used to style the HTML inside the browser.
- Views: HTML files that are displayed either as control panels or Magic Menu pages. The same HTML file can be used as both a control panel and a Magic Menu page.

6.1.1 Default Assets

Overture provides a set of starting assets available for immediate use after installation:

6.1.1.1 Images

The only image provided in the default Overture distribution is the one used for the default background of Magic Menu. Replacing this image with another with the same name will change the background of the Magic Menu.

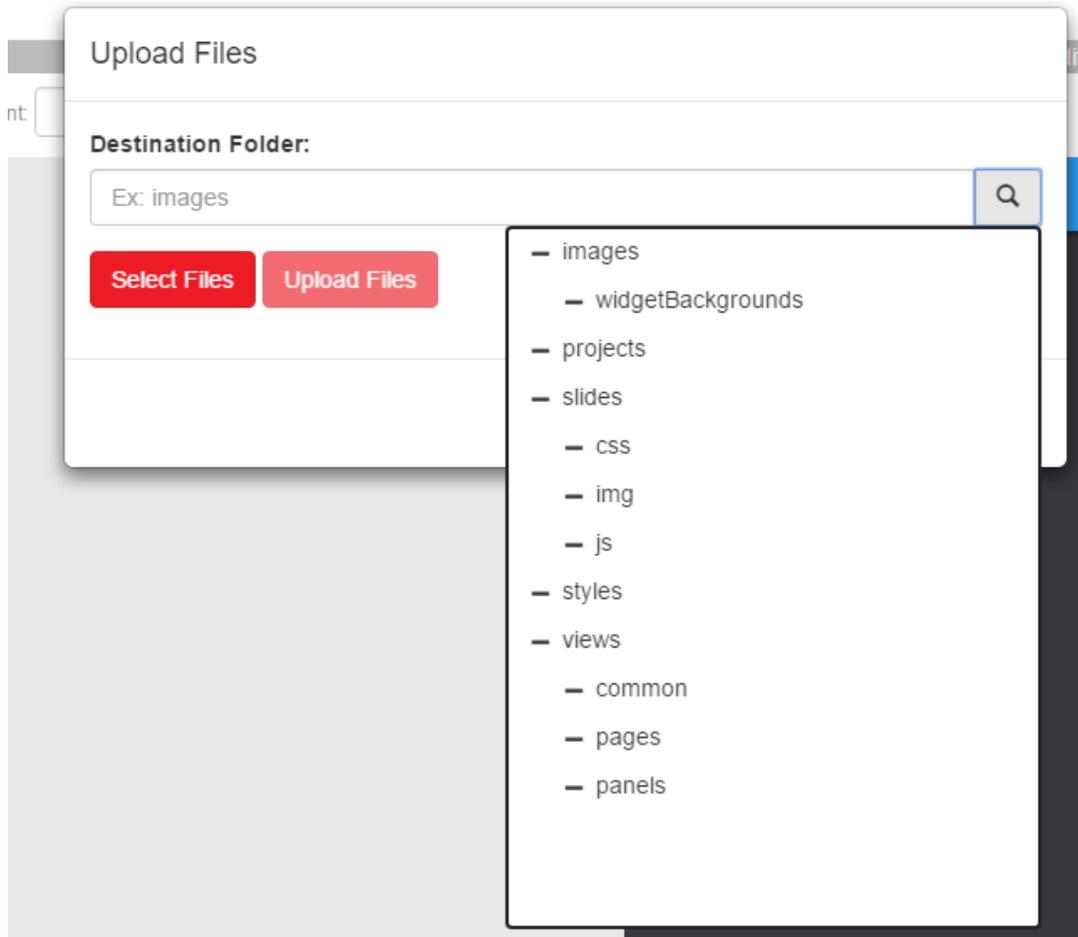
6.1.1.2 Views

P 141 / 270

[Overture](#) includes some built-in HTML templates:

- **Device Templates:** These are files used to control or view data from specific devices based on the subtype (projector.html, lighting.html, etc). To use these, assign them in the Configurator via 'Control Panel' selection.
- **Room Templates:** This is the [room.html](#) in the [views/common](#) folder. This file will create a room control panel based on the devices that are children of the room point. This is assigned in the Configurator via 'Control Panel' selection.
- **Home Page:** The starting page of [Magic Menu](#)

6.1.2 Uploading Assets



Files can be uploaded directly in the assets folder from the GUI Editor. To do so, select 'Upload' from the File menu. From there, select which sub-folder to upload your file into. You will then select the files to upload and click 'Upload Files'.

6.1.3 Assets Explorer

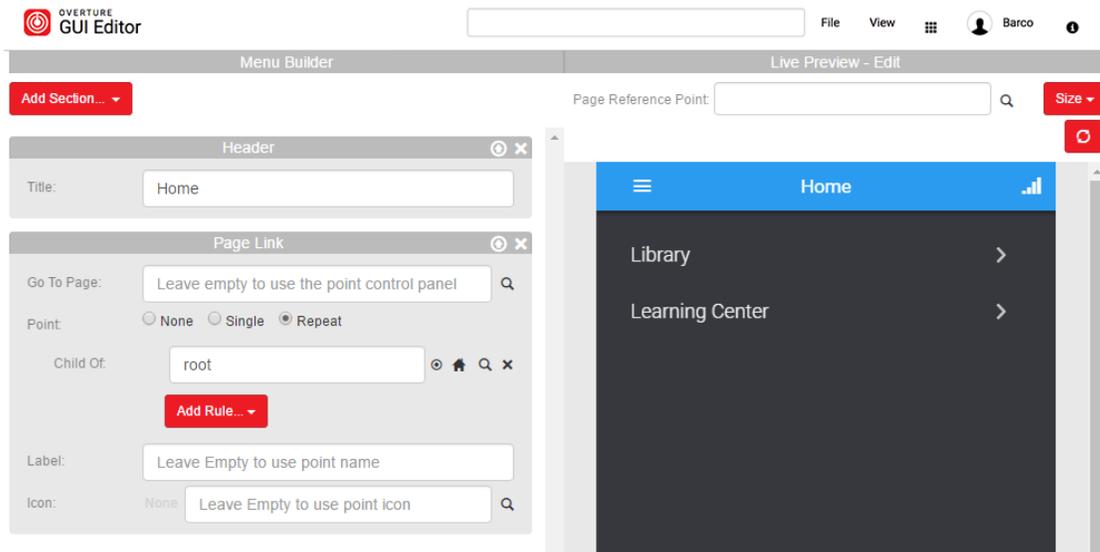


Files in the assets folder can be altered via GUI Editor using the Assets Explorer. The Assets Explorer can be accessed from the File menu.

Once there, you can create new sub-folders, move files, rename files, or delete files. When a non-html file is selected (like, for instance, a stylesheet, markdown file, or a picture), it will be handled in expert mode. If an [html](#) file is selected, it will be opened in menubuilder mode if it's compatible with it, and it will be opened in expert mode if it's incompatible with menubuilder (contains incompatible elements).

Note: [views/home.html](#) cannot be moved, renamed, or deleted as it is necessary for Magic Menu to function.

6.2 Menu Builder

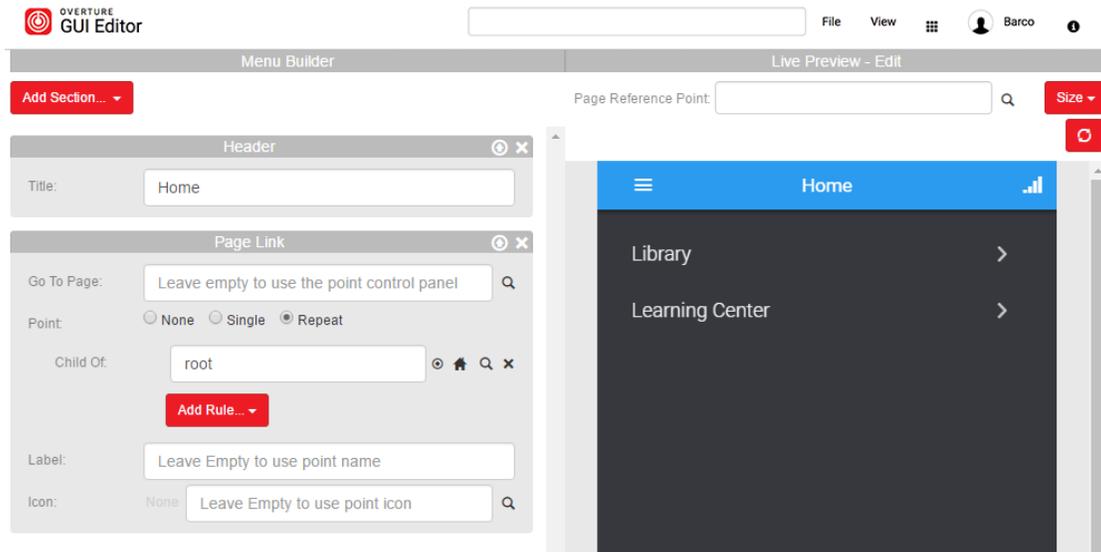


Menu Builder is a tool to build your templates. While in this mode, you can build templates using drop down selections. The [Live Preview](#) section will update automatically as you make changes.

Menu Builder can only create new html files, or open files already created by the Menu Builder. Trying to open a file that wasn't created by the Menu Builder will display an error and not allow you to edit the file. To edit or create non Menu Builder pages see the [Advanced Programming](#) section.

6.2.1 Sections

6.2.1.1 Intro



In order to create your Magic Menu page, you will need to add sections each with their own unique options. When adding a section, you can click the expand icon to see all available options. You can also hide the section panel, still adding it to the page, by clicking the eye. Sections are deleted by clicking the X.

6.2.1.2 Display

Displays the value of a variable.

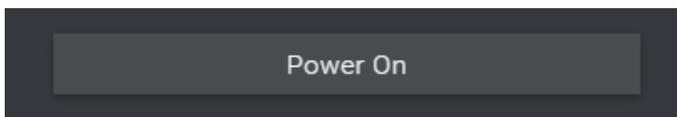


Options:

- **Point:** The point to be displayed. See [Points](#).
- **Label:** The text displayed to the left of the value. See [Labels](#).
- **Suffix:** The suffix to be appended to the point's value. Overrides a point's unit from the database if specified.
- **Size:** The size of the text.
- **Date Format:** The format a time variable should be displayed. See [Date Format](#).
- **Icon:** The icon displayed next to the label. See [Icons](#).

6.2.1.3 Button

Creates a button that triggers an action when clicked (or tapped). It can also change its appearance depending on the value of a specified point.



Options:

- **Point:** The point to be monitored. See [Points](#).
- **Label:** The text displayed in the button. See [Labels](#).
- **Click:** The action triggered when the button is clicked. See [Actions](#).

- **OnState**: The value of the point that causes the button to appear highlighted. (Example: '1' would mean when the point's value is equal to 1, the button will highlight.)
- **Icon** : The icon displayed next to the label. See [Icons](#).

6.2.1.4 Slider

Creates a slider that sends commands when the user moves it. The position of the slider marker is determined by the value of a point.

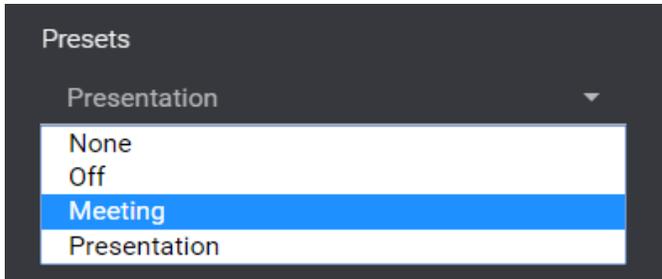


Options:

- **Label**: The text displayed with the slider. See [Labels](#).
- **Point**: The point to be monitored. See [Points](#).
- **Change**: The action triggered when the slider is moved. See [Actions](#).
- **Step**: The amount moved 'up' and 'down' icons. Defaults to 1.
- **Min**: The minimum value of the slider. Uses the `min` attribute of the point if not specified.
- **Max**: The maximum value of the slider. Uses the `max` attribute of the point if not specified.
- **Suffix**: The suffix to be appended to the point's value. Overrides a point's unit from the database if specified.
- **IconUp**: The icon displayed to the right of the slider. See [Icons](#)
- **IconDown**: The icon displayed to the left of the slider. See [Icons](#)

6.2.1.5 Select

Creates a multiple-choice selection box, which allows the user to select an item from a list. It is intended to be used with `enum` type variables.



Options:

- **Point**: The point associated to the list. Intended to be sub-type `enum`. See [Points](#).
- **Label**: The text displayed above the selection. See [Labels](#).
- **Change**: The action triggered when an item is selected. See [Actions](#).
- **Icon** : The icon displayed above the label. See [Icons](#).

6.2.1.6 Checkbox

Creates a checkbox, allowing the toggling of values.



Options:

- **Label**: The text displayed with the checkbox. See [Labels](#).
- **Point**: The point to be monitored. See [Points](#).
- **Change**: The action triggered when the box is checked or unchecked. See [Actions](#).
- **ChangeOn**: The action triggered when the box is checked. See [Actions](#).
- **ChangeOff**: The action triggered when the box is unchecked. See [Actions](#).

6.2.1.7 Toggle

Creates a switch style button, allowing the toggling of values.



Options:

- **Label**: The text displayed with the button. See [Labels](#).
- **Point**: The point to be monitored. See [Points](#).
- **Change**: The action triggered when the button is switched on or off. See [Actions](#).
- **ChangeOn**: The action triggered when the button is switched on. See [Actions](#).
- **ChangeOff**: The action triggered when the button is switched off. See [Actions](#).

6.2.1.8 Auto Widget

Creates a widget whose type depends upon the type of the specified point(s) and additional information provided in the point's metadata. For example: Auto Widget automatically creates a Slider widget if the point is a variable of type [integer](#).

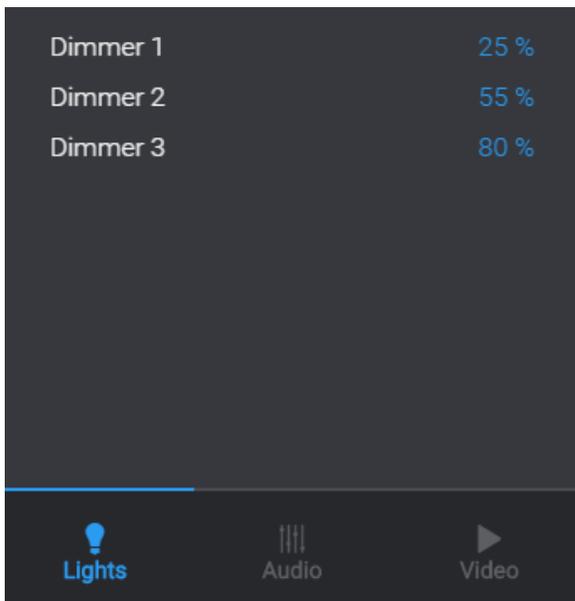
Auto Widget Options:

- **Point**: The point to associate the widget to. See [Points](#).
- **Label**: The label of the widget. See [Labels](#).

6.2.1.9 Tabs

Tabs are special containers that show or hide content based on which container is selected. It consists of two components: Tab Control and Tab Content.

Tab Control creates the tab bar that has links to the containers. Tab Contents are the containers themselves.



Tab Control Options:

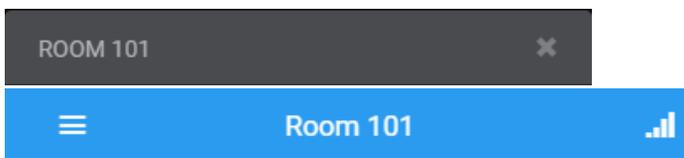
- [App](#): The application the tab control bar shows up in.
- [MagicMenu Position Small](#): The position to place in tab control bar when the MagicMenu interface is small.
- [MagicMenu Position Medium](#): The position to place in tab control bar when the MagicMenu interface is small.
- [MagicMenu Position Large](#): The position to place in tab control bar when the MagicMenu interface is small.

Tab Content Options:

- [Point](#): The point to associate the tab container to. See [Points](#).
- [Label](#): The text displayed in tab control bar link. See [Labels](#).
- [Icon](#): The icon displayed next to the label. See [Icons](#).

6.2.1.10 Header

Creates a name at the top of the Magic Menu page or Control Panel. If no header is specified, the panel defaults to the name of the point.



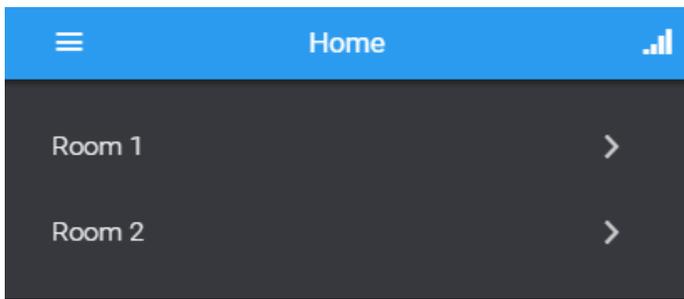
Options:

- [Title](#): The text displayed at the top of the page.

6.2.1.11 Page Link

Magic Menu Only

Creates a button which displays a new page when clicked. This page must be an html document that exists in the views section of the assets folder.

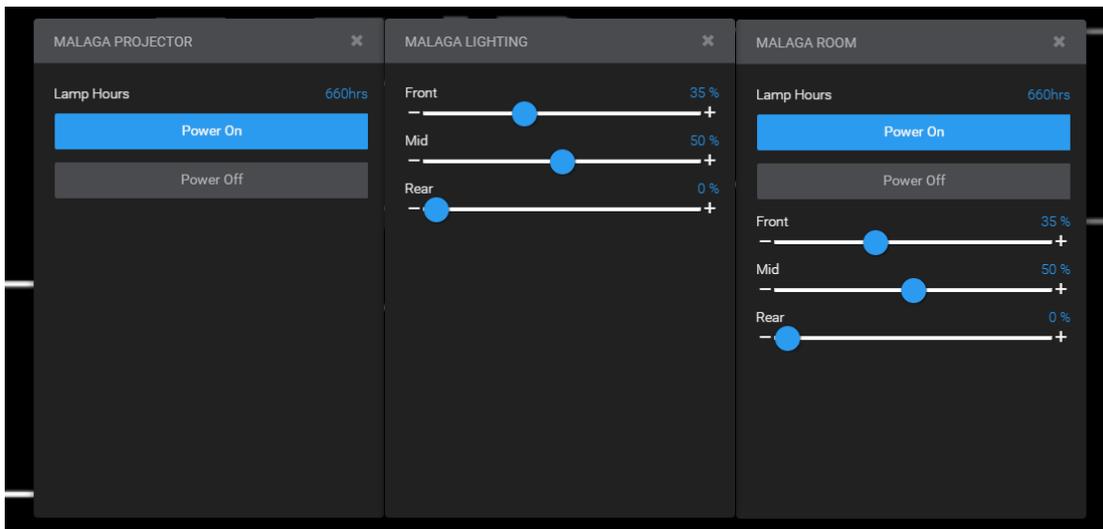


Options:

- **Go To Page**: The HTML document you wish to load when the link is clicked. Must be format of **FOLDER/NAME** and no file type after. If not specified, the page loads the **med-point/med-repeat**'s HTML file associated to the 'Control Panel' section of the Configurator.
- **Point**: The point associated with the page loaded. See [Points](#).
- **Label**: The text displayed in the button. See [Labels](#).
- **Context**: Extra information you wish to pass into the new html page. Must be in form of an object `{item: value}`.
- **Roles**: The specific roles that can see the page link button. Example: `\med-roles="A/V,Building,IT"`. The name or the IDs of the roles may be specified.
- **Icon**: The icon displayed in the button. See [Icons](#).

6.2.1.12 Include

The **med-include** tag loads another HTML file in the control panel. It is useful for creating templates by allowing small portions of code to be reused in many panels.



Options:

- **Include Page**: The HTML document you wish to include. Must be format of **FOLDER/NAME** and no file type after. If not specified, the page includes the **med-point/med-repeat**'s HTML file specified in the 'Control Panel' section of the Configurator.
- **Point**: The point associated with the page included. See [Points](#).
- **Context**: Extra information you wish to pass into the included page. Must be in form of an object `{item: value}`.

6.2.2 Options

6.2.2.1 Intro

Some options exist for multiple sections, and have specialized ways they can be used.

6.2.2.2 Points

Points can be selected and add context to items.

Single Point Selection

Go To Page:

Point: None Single Repeat

Single:

Label:

Icon:

- Server1
 - Campus
 - Library
 - + Sevilla Room
 - + Minorca Room
 - + Learning Center

When assigning a single point to an item, you can type the variable name into the field or human name if there is no variable name. You can also use the magnifying glass to select an existing point from your database.

Repeat Point Selection

Menu Builder | Live Preview - Edit

Add Section... | Page Reference Point:

Header: Title:

Page Link: Go To Page:

Point: None Single Repeat

Child Of:

Has Type:

Label:

Icon:

Home

- 🏠 Malaga >
- 🏠 Segovia >
- 🏠 Catalonia >
- 🏠 Alhambra >
- 🏠 Palermo >
- 🏠 Santander >
- 🏠 Valencia >

When assigning points using repeat, the Menu Builder will dynamically create a number of items based on the rules added. These rules will find points based on database criteria. The following rules can be used:

- Child Of : the parent of the point(s) you want to find.
- Child Depth : how far down the hierarchy the point could be.
- Name Contains : the string contained in the human name of the point(s) you want to find.
- Short Name Contains : the string contained in the short name of the point(s) you want to find.

- Variable Name Contains : the string contained in the variable name of the point(s) you want to find.
- Has Type: the type of the point(s) you want to find.
- Has Subtype: the sub type of the point(s) you want to find.
- Has Role(s): the roles associated to the point(s) you wish to find.
- Has Tag(s): the tags associated to the point(s) you wish to find.

6.2.2.3 Labels

Labels provide text to items on the page.

You can provide text, or use a label context variable to provide dynamic information. The following variables work:

- \$humanname: The human name of the point associated to the item.
- \$shortname: The short name of the point associated to the item.
- \$parent.name: The human name of the parent of the point associated to the item.
- \$parent.shortname: The short name of the parent of the point associated to the item.

Labels will default when no information is provided. The system will go down the following hierarchy, using the associated point, until it finds something:

Label -> Point's Short Name -> Point's Human Name -> Point's Variable Name -> Blank

6.2.2.4 Actions

Actions are how interactive items interact with the system. The action is called when the item is interacted with (For example, clicking a button or moving a slider). They are JavaScript functions that normally interact with the Control Server. The following actions can be used:

- `setVariable()`: This is used to change the value of a variable type point. It's the default action when none is provided.
- `perform()`: This is used to interact with tasks or devices directly. It performs a device command directly, or starts/stops a task on the Control Server.
- `openUrl()`: This is used for buttons that need to open other web pages.

Set Variable

Used to set a variable to a desired value. When using a Control Server Version 2 or above, this action will generally be enough to tell the driver which device command to perform.

`setVariable(point, value)` has two parameters:

- *point* : The name of the point you want to change the value of.
- *value* : The value to change the point to.

Example:

```
setVariable('Projector.Brightness', 50)
```

Perform

It can be set up two ways. The first way interacting with a task.

`perform(task, command)` has two parameters:

- *task*: The name of the task to interact with.
- *command*: The command to perform on the task. Must be either 'starttask' or 'stoptask'.

Example:

```
perform('Room_On', 'starttask')
```

It can also be used to directly perform a device command.

`perform(device, command,[parameters])` has two required parameters and one optional parameter:



- *device*: The name of the device to interact with.
- *command*: The command to perform on the device. The command name comes from the driver documentation.
- *parameters*: The command parameters if needed. Must be in the form of an object ({name1:value1, name2:value2, etc.}). This information is in the driver documentation.

Example:

```
perform('Projector', 'Set Power', {'Status': 'On'})
```

Open Url

Used to open a web page in a new tab on the browser.

`openUrl(url)` has one parameter:

- *url*: The URL to open in a new tab.

Example:

```
openUrl("http://www.barco.com")
```

Action Context Variables

Actions have context variables that can be used inside of their functions to help build more dynamic pages.

point

point is used to describe the point of the current page. It's useful for substituting the device parameter in a `perform()`.

Example: `perform(point, 'Set Power', {'Status': 'On'})`.

\$point

\$point is used to describe the point associated to the item. See [Points](#).

Examples: `perform($point, 'starttask')`. `setVariable($point, 50)`.

\$value

\$value is used to describe the value of the point associated to the item. For example, when you move the slider to 50. *\$value* becomes 50 before it is passed into the function.

Examples: `perform(point, 'Set Level', {'Level': $value})`. `setVariable($point, $value)` (This is called when no action is provided).

\$string

\$string is used to describe the string value of an enum variable point associated to the item. For example if an enum variable point has the enums of "Off" and "On" in that order, the *\$value* is the index (0 or 1) and the *\$string* will be "On" or "Off". It's needed for most device commands that interact with enums.

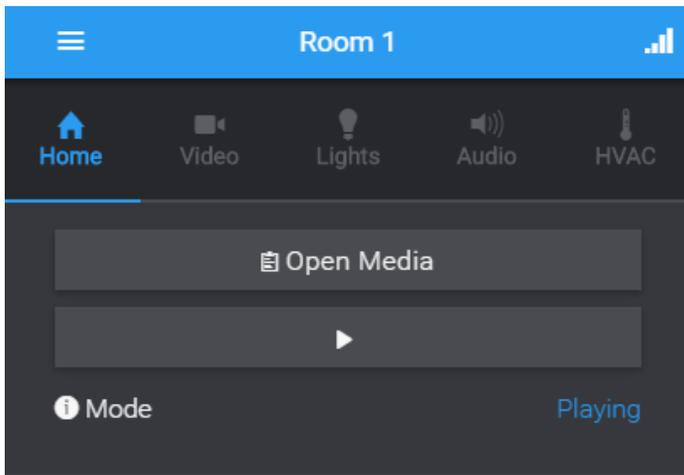
Example: `perform(point, 'Select Source', {'Name': $string})`.

\$parent

\$point is used to describe the parent of the point associated to the item.

Example: `perform($parent, 'Play')`.

6.2.2.5 Icons



Some items have an Icon option. This allows you to add or change icons on tags within the system.

You can select an icon from the Icon Browser by using the magnifying glass to the right. The icon name will auto-populated.

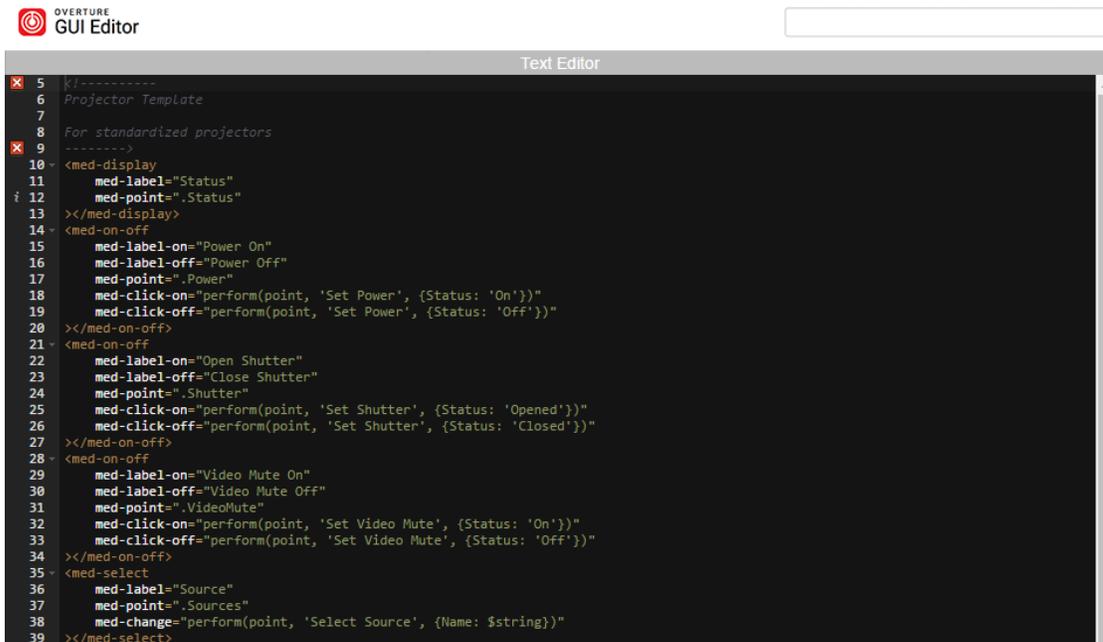
Icons will default when no information is provided. The system will go down the following hierarchy, using the associated point, until it finds something:

Icon -> Point's Icon -> Point's Sub-Type Icon -> Point's Type Icon* -> Blank**

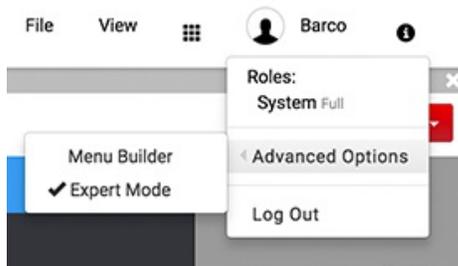
6.2.2.6 Date

Some tags will have a Date Format option. This allows the incoming value to be formatted in more user friendly way. For example, time variables may come in with values like "09:00:00/00", where as you might just want to tell the user "09:00 AM". The drop down will provide an example of each style and will automatically fill in the appropriate information.

6.3 Expert Mode



Expert Mode is used for editing HTML files directly. It is recommended only if you are comfortable with HTML and JavaScript programming. To access Expert Mode, go to the User menu, and under Advanced Options and select Expert Mode.

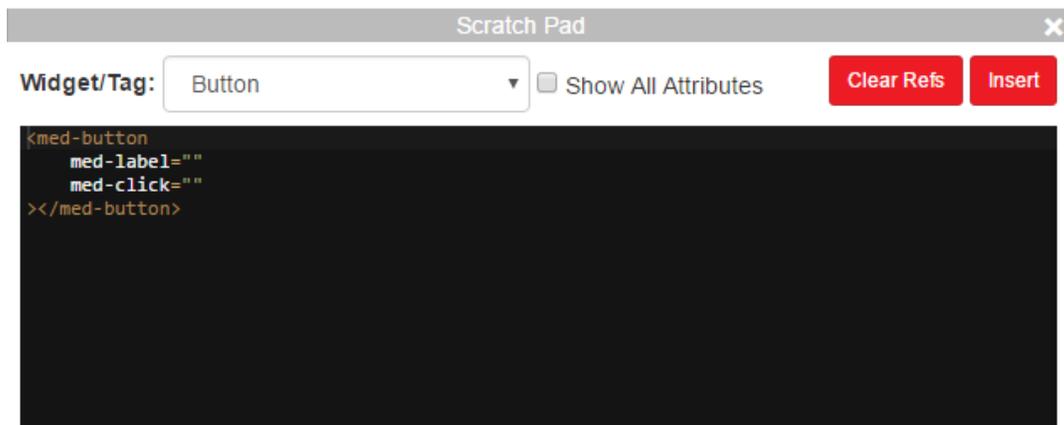


For more information on how to use Expert Mode, see the [Advanced Programming](#) section.

Note: If you edit a Menu Builder file in Expert Mode, and then go back to Menu Builder Mode you may lose some of your work due to the conversion.

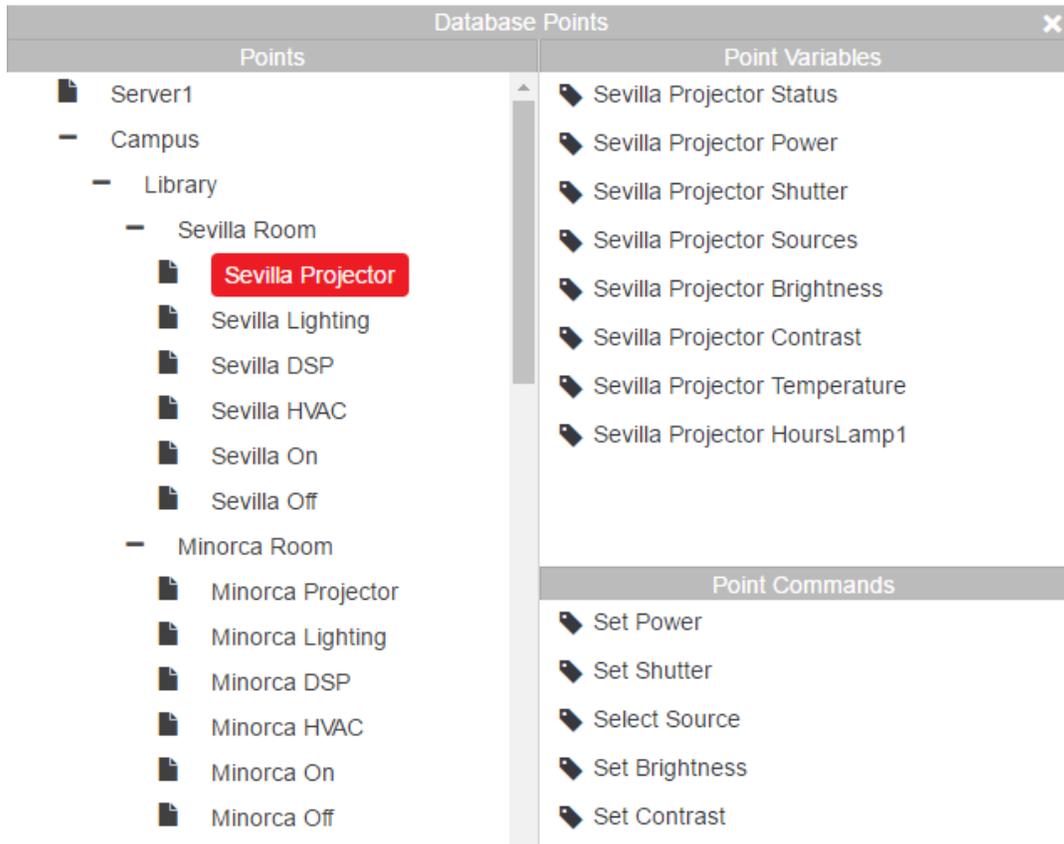
You can customize the way Expert Mode looks and feels for you by going to 'Views':

6.3.1 Scratch Pad



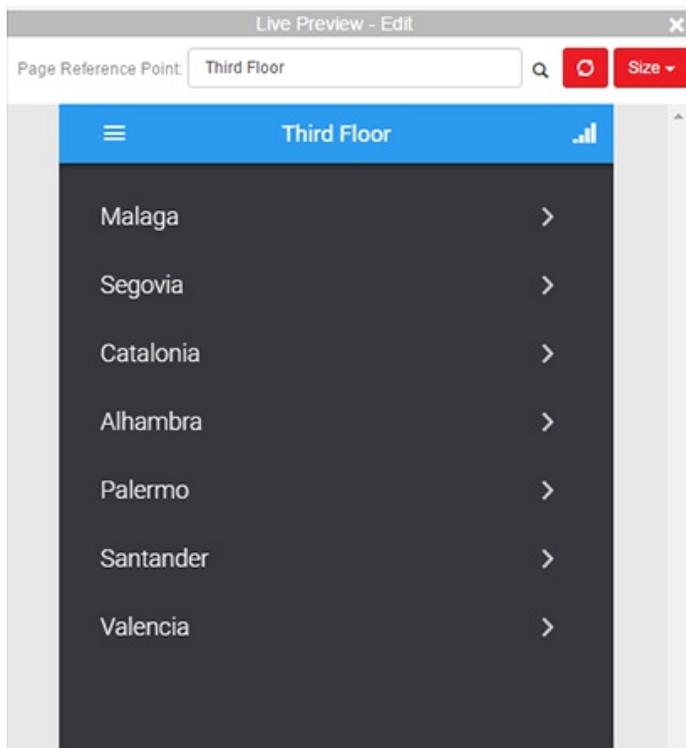
Scratch Pad allows you to auto fill some of the programming when using Expert Mode. A drop down will show all of the possible HTML tags available and insert it directly into the Expert Mode editor.

6.3.2 Database Points



Database Points will show a hierarchy view of your database. From there, you can directly select points, commands, and variables to fill attributes in the Scratch Pad.

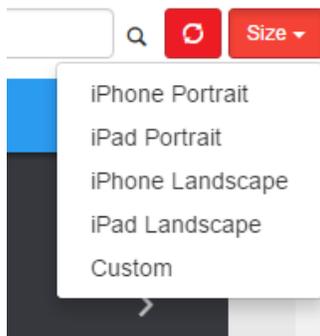
6.4 Live Preview



GUI Editor has a built live preview. This allows you to check what was done in Menu Builder or Expert Mode by giving you a live view of the Magic Menu. If you would like to put a point context in to the live preview, you can add it into the Page Reference Point field. Using the magnifying glass allows you to select it from the database.

Note The preview will only update the current page being edited and not any pages clicked in the preview.

6.4.1 Size Options



The live preview can be sized according to which device and position you will be viewing the Magic Menu. There is also a custom option to specify your exact dimensions.

7 Scheduler

The screenshot shows the 'Events' page in the Overture Scheduler interface. At the top left is the 'OVERTURE Scheduler' logo. On the right, there are navigation links for 'Events', 'Calendar', and 'Medialon', along with a user profile icon. A red 'ADD NEW EVENT' button is located in the top right corner. Below the header is a search bar. The main content area displays a list of five event entries, each with a title, a description of the event's schedule and actions, and a status indicator on the right. The events are: 'General Off' (status: 'in a day'), 'Circle Off Night' (status: 'in 6 hours'), 'Noon show' (status: 'in 4 days'), and two 'Event' entries (status: 'Passed' and 'Disabled').

[Overture Scheduler](#) is the event management tool. It allows the user to create events which trigger state changes at designated moments. There are two different views available:

- Events: A list of past, present and future events
- Calendar: A calendar view of these same events

You can navigate to [Scheduler](#) by selecting it in the [App Switcher](#) (in the top right of the header ) , or by adding </scheduler/> to your server URL. You must have sufficient access rights in order to use [Scheduler](#) (see [Access Rights](#)).

7.1 Event Definition

The primary purpose of [Overture Scheduler](#) is the ability to specify [events](#), which in turn can set the value of a particular variable at the moment specified in the event. [Scheduler](#) events can be recurring, or can be limited to just a single occurrence.

When an event triggers the setting of a new value to be set on a "status" variable, [Overture](#) will automatically trigger that variable's state to change in order to reflect this new value.

For example, consider an audio level variable, [TVScreen.Volume](#). Setting its value to [50](#) would trigger [Overture](#) to send a command to the [TVScreen](#) device to adjust the volume to 50.

As another example, consider the case of a meeting room point with a "Presets" variable, [MeetingRoom.Presets](#). When the value of this variable is changed to [Meeting Mode](#), [Overture](#) signals the corresponding device(s) to perform whatever action is needed such that the room is set to the [Meeting Mode](#) configuration.

7.2 Events View

The [Events View](#) displays past, present and future events as a list. Each event's listing contains the following information:

- The name of the event
- The time of the event
- What the event action is
- An indication of one of the following, when applicable:

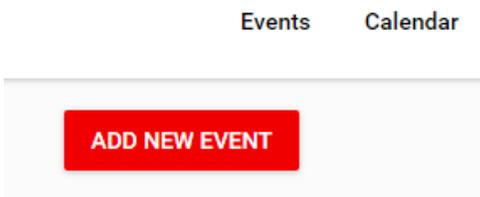
- The time of the next occurrence of the event OR
- The fact that the event has already occurred OR
- The fact that the event is disabled

Circle Off Night <i>Every Monday, Tuesday, Wednesday, Thursday, Friday starting on the 11th day of August 20:00 EDT</i> <i>Set The Circle Status to Off</i>	in 11 hours
Noon show <i>Every Saturday starting on the 11th day of August 12:00</i> <i>Set Paris_Lights Presets to 5</i>	in 2 days
Event <i>Starting on the 18th day of August 14:00</i> <i>Set London_Display Status to Connected</i>	Passed
Event <i>Starting on the 18th day of August 14:24</i> <i>Set London_Display Activity to Enabled</i>	Disabled

7.2.1 Creating Events

7.2.1.1 One-Time (Non-Recurring) Events

In the [Events View](#), click the [Add New Event](#) button on the right top side of the page to create a new event.



This will open the event creation/edit dialog, where you can specify the event's properties.

The image shows a form for creating or editing an event. The form has the following fields and controls:

- Event name:** A text input field.
- Activated:** A toggle switch currently turned on (red).
- Date:** A text input field containing '2017-08-22'.
- Time:** A text input field containing '14:52'.
- Timezone:** A dropdown menu showing '(GMT-04:00) Eastern Time'.
- Points:** A text input field.
- Value:** A text input field.
- Repeat:** A dropdown menu showing 'No'.
- Buttons:** 'CLOSE' and 'SAVE' buttons at the bottom right.

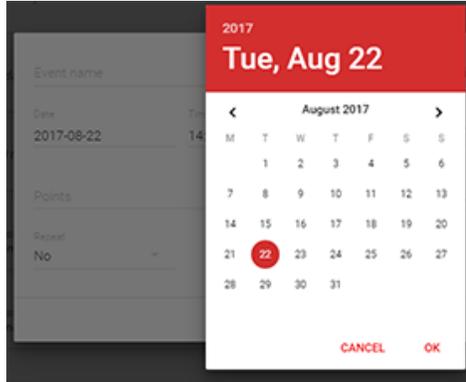
This dialog provides you with the following options:

- **Event name:** Set the name of the event that will appear on the different views of [Scheduler](#)
- **Activated:** Activate or disable the event
- **Date & Time:** Set the date and time at which the event is to be triggered

- Timezone: Set the time zone in which the time of the event is defined (the browser's local time zone is the default)
- Points & Value: Specify the value of the point that will be set when the event's action is triggered
- Repeat: Specify whether the event should occur once, or whether it should be executed recurrently (see [Recurring Events](#))

When a property is clicked, a simplified user interface is opened for the value to be changed, e.g.:

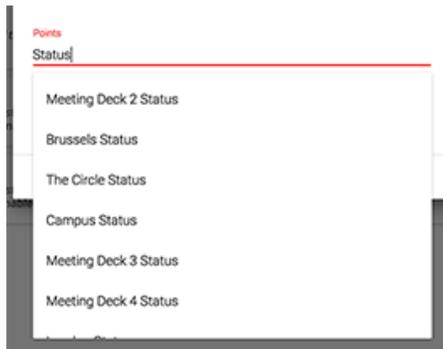
- Date selection:



- Time selection:



- Point selection:



Note that the point selector has a search field, which allows filtering the listed points.

The example below shows the creation of an event that will set the variable [The Circle Status](#) to [On](#) on [August 15th, 2017](#), at [3pm](#):

7.2.1.2 Recurring Events

[Scheduler](#) also supports the creation of events that occur at some specified interval. This enables a user to, for example, automatically power off rooms at the end of every day. To set a recurring event, set the [Repeat](#) property to [On](#) in the event creation/edit dialog:

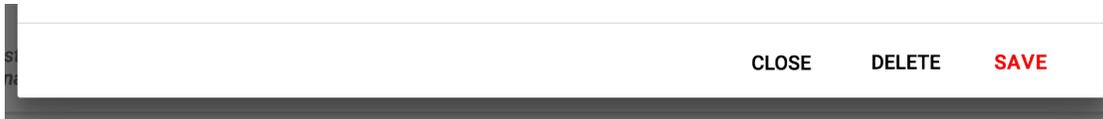
The recurrence interval can be configured with the following options:

- Day Of Week: Select specific days of the week during which the event should occur
- End Repeat: (Optional) The last execution of the event. If not defined, the event will recur indefinitely.
- Add Exception: Allow the user to define dates at which the event won't be triggered. Multiple dates can be entered here.

7.2.2 Editing an Existing Event

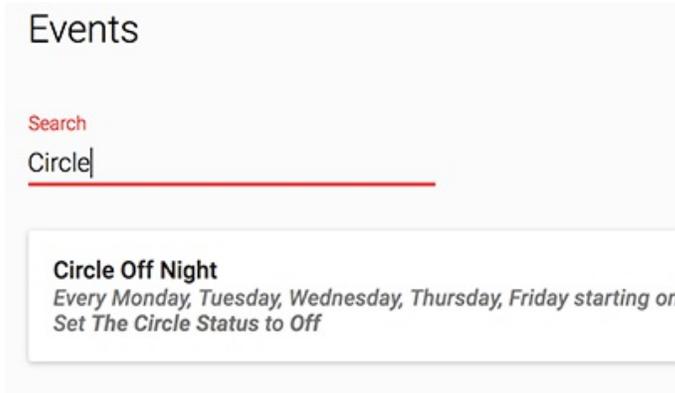
Clicking on any existing event in the [Events View](#) reopens the event creation/edit dialog, allowing this event to be edited:

The event can be also be deleted from within the event creation/edit dialog, by clicking the [Delete](#) button:



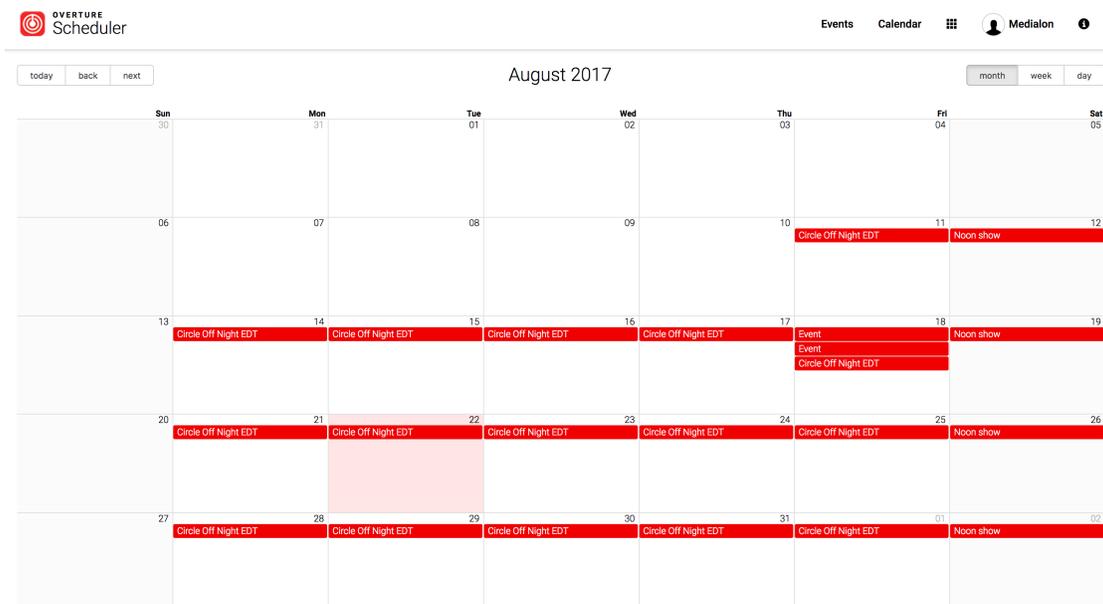
7.2.3 Searching for an Event

You can search for an event by name via the [Search](#) field at the top-left side of the [Events View](#):



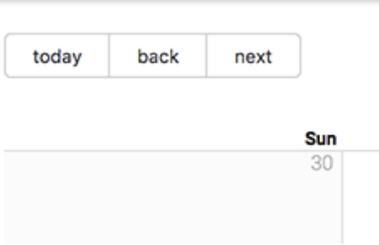
7.3 Calendar View

The [Calendar View](#) displays [Scheduler](#) events in a calendar, whose time window can be set to [Month](#), [Week](#) or [Day](#):



7.3.1 Navigation

The [Today](#), [Back](#) and [Next](#) buttons on the left top side of the calendar view allow the user to navigate between dates. Depending on the calendar mode, the [Back](#) and [Next](#) buttons will either select the previous/next [Month](#), [Week](#) or [Day](#).

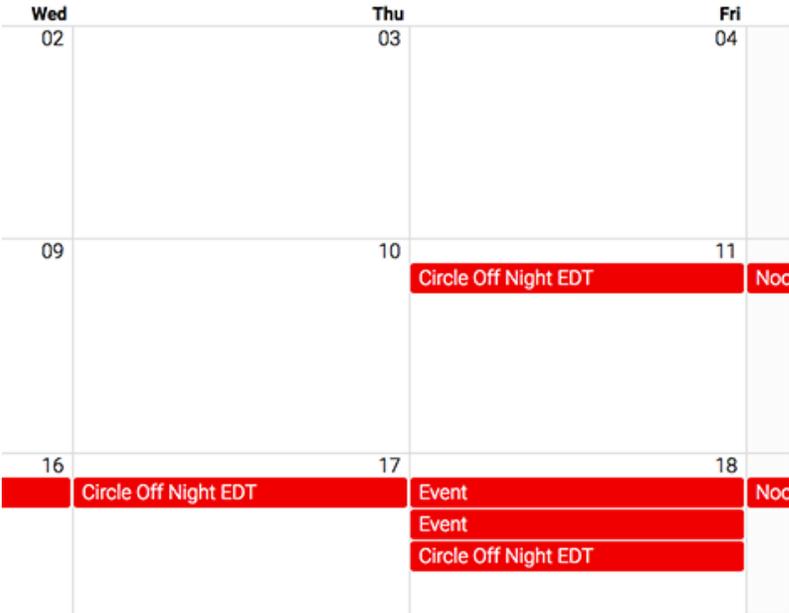


Clicking on any calendar view will either edit an existing event or create a new event, depending on whether the user clicked on an existing event or on an empty area.

7.3.2 Calendar Month, Week and Day Views

In the [Calendar Month View](#), [Calendar Week View](#) and [Calendar Day View](#), events are displayed along with an event summary, which includes the event's name and its activation time. If the event's time zone was set to one different from that of the current browser, the event's name is followed by a short time zone indication. See the corresponding sections below for examples, each involving the [Circle Off Night](#) event:

7.3.2.1 Month View



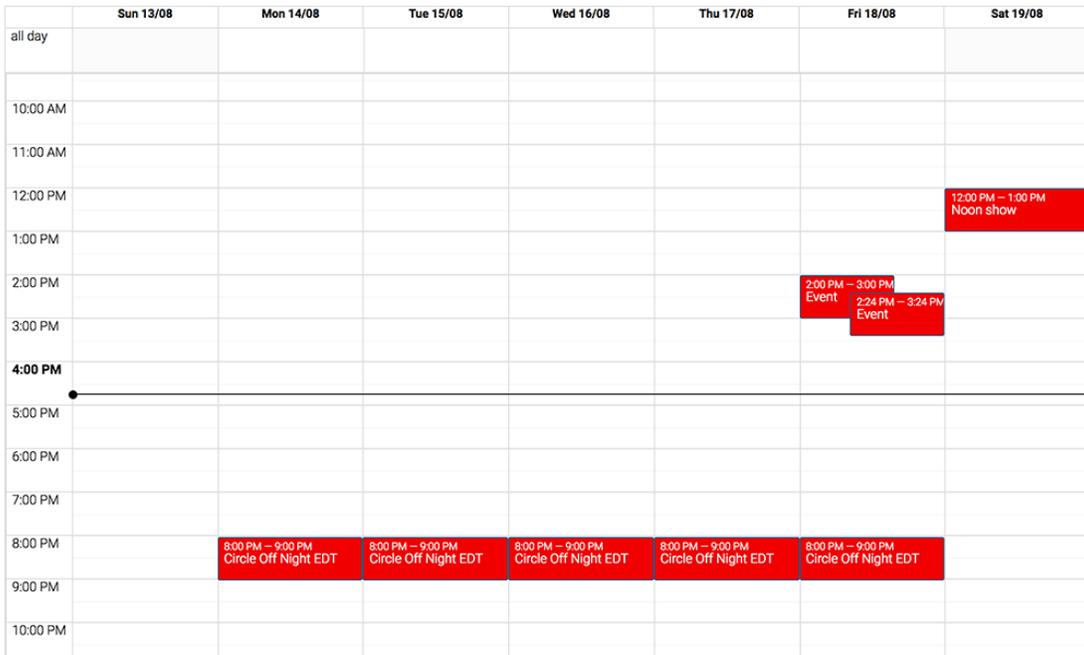
7.3.2.2 Week View



today back next

Aug 13 - 19

month week day

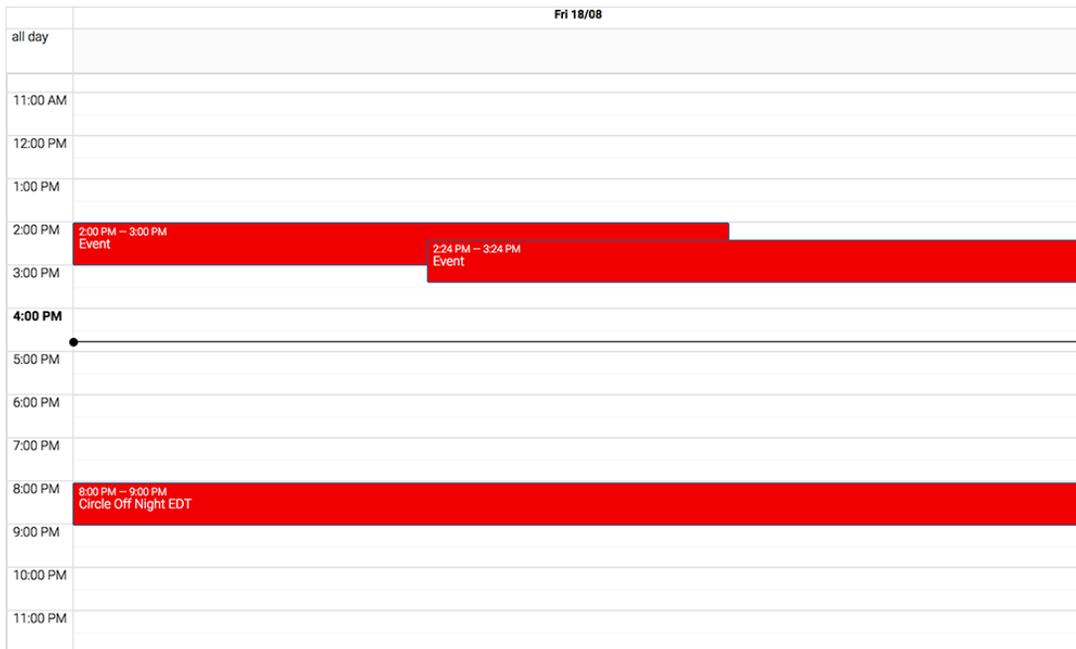


7.3.2.3 Day View

today back next

Friday Aug 18

month week day



8 Additional Information

After the system has been programmed, Overture has some nice extra features to help build a better user experience within the system.

8.1 Backups/Restoring

UX Server has the ability to backup all of your data and allow restoration in case of data corruption or error.

8.1.1 Backups

There are two ways to back up your data in the UX Server. The first way to backup your data is manually through the Configurator's 'Backups' view.

Backup date	Filename	Database version	Restore	Download	Delete
2016/07/05-10:18:06	OvertureDataBackup_20160705101806.zip	1.2.15	C	↓	✕
2016/07/05-08:37:52	OvertureDataBackup_20160705083752.zip	1.2.15	C	↓	✕

In this view, you can see the current list of backups in the file server as well as adding one manually by clicking 'Back Up Now'.

You can also upload a backup from your PC, restore a backup, download a backup to your PC, or delete any backup in the list.

Any non-backup files found in this view will have a strike through them, but can still be downloaded or deleted.

Backups are also created automatically through the 'Automatic Backup Settings'. This is found in the 'UX Server Config' view in the Configurator.

Automatic Backup Settings

Active:

Frequency: Every Week

Monday Tuesday Wednesday Thursday
 Friday Saturday Sunday

Time: 12:00 AM

Keep backups for: 30 Days

The time settings are about the server local time (America/New_York)

Here you can specify, whether or not to backup, how frequent, and on what date and time to do the backup. You can also specify how long the backups should stay before the system deletes them.

8.1.2 Restoring

You can restore any of the backups in the [Backups](#) view of Configurator.

Each backup in the list of backups has the option to be restored, as long as the database version of that backup is not greater than the current database.

Click the restore icon. Once done, you will need to confirm the process. Restoring will overwrite your current assets and database information with the backups assets and information.

After the update process, your server will be restarted.

8.1.2.1 Restoring backups from a lower version than 3.3.x to a version greater or equals than 3.3.x

For integration with LDAP and ADFS, Overture had to change the way it manages its users internally. Because of that, the process of restoring backups from any version lower than 3.3.x is a bit different. After restoring a backup from a version lower than 3.3.x, all Overture users will be recreated and a temporary password will be set for these users.

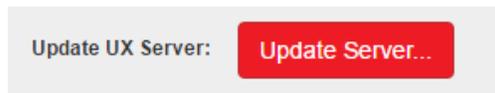
After the server restart, a download of a CSV file which contains all users and their new temporary passwords will start. After login in for the first time with a temporary password, the user will be demanded to change its password.

8.2 Updating Overture

Depending on your installation there are a few ways to update your system. Always make a back up, and download it before updating the system

8.2.1 Updating UX And CS Server

When updating the UX and CS in the virtual enviroment, you will need to upload a new yml in the Configurator. To do so, go to the 'UX Server Config' view and click the 'Update Server' button.



You will need to provide the new yml file, once it has been validated the server will update and restart. Note the server will need to be connected to the internet to update.

8.2.2 Updating Extra Control Servers

If you are updating an extra Windows PC Control Server, you will need to install the newest version of the software with an executable.

If you are using Docker, you will need to load the newest archive file (or have an internet connection) and re run the compose file. Please see Setup for more info on those processes.

8.3 Linking to Specific Maps or Pages

Overture links can be sent to users that only allow them access to certain layers. With Magic Menu, this is a specific HTML page created. With Home, this will be a specific map layer.

8.3.1 Home

8.3.1.1 Direct Linking to Maps

Links to Home can directly reference certain map layers via the `?maps` parameter. The value of this parameter works the same way as `findPoint()`, i.e., you can reference a point via human name, variable name, etc. This parameter is *case sensitive*. Example: <http://localhost/?map=Building B1>

N.B. Depending on your browser, certain characters in the name are usually automatically converted to/from their HTML

escape sequences when typed in. For example, [Building B1](#) might become [Building%20B1](#) (or vice-versa), but Overture accepts both as equivalent.

This link will direct you to the map layer for the point named Building B1 if it exists, otherwise it will just direct you to the default top-level map as usual.

If the point referenced by this parameter value has a map, that map will be shown as the first map in Home. If the point does not have a map, Home will instead display the nearest ancestor point with a map, and, if applicable, open the control panel associated with that point.

8.3.1.2 Exposing the Map Hierarchy

Whenever you directly reference a map through the `?map` parameter, by default, that point's map (or closest ancestor's map) becomes the root of the map hierarchy in Home. In this case, *you will not be able to navigate to maps/points beyond this root point.*

Although this helps to focus on portions of larger map trees/hierarchies, the user may opt out of this map tree restriction with the `?showrootmap` parameter. Example: <http://localhost/?map=Floor1&showrootmap=1>

Just like before, this will navigate to a point named "Floor1", and assuming it has a map, this map will be shown first in Home. By default, the user would have been restricted to "Floor1" and its descendant points, but with `showrootmap=1`, the map for Floor1 will be shown directly as usual, but the user is free to access all other points accessible from the Home Map section.

8.3.2 Magic Menu

By default, the root point displayed as the starting interface in Magic Menu, should have a map associated with it. If you need to assign a point without a map as the starting interface, use the HTML template to connect to as described below.

Besides restricting access to certain points, Magic Menu links with parameters can be used to create different interfaces, for example, if you want to create specific in-room interfaces (similar to control panels), such as for tablets.

These Magic Menu pages, like control panels, provide a way to monitor or control devices on the Overture system. They are generated with HTML templates created or edited by the user. These may also contain links to other pages, giving your user interface an organized layout.

In such a scenario, you can write another HTML page, (e.g. `inRoom.html`) and use the link parameter syntax provided in the following sections to only launch that page with the browser.

N.B. In Overture, the HTML template files you write can be used either as a Magic Menu page or a Home control panel.

8.3.2.1 Auto-login to Magic Menu via URL

If you need to enter a panel (or even just the regular Magic Menu) immediately but need to bypass the login page, you can also be automatically logged into Magic Menu if you pass in a valid token in the URL with the `?token` parameter, example: http://10.0.50.231/magicmenu?token=<ADD_TOKEN_HERE>

Although any valid token can be used this way, please note that it is designed specifically for use with permanent tokens generated in special situations only, for example, if you have large number of instances to have been launched and authenticated immediately as part of your installation's setup.

The `token=<TOKEN_VALUE>` is then removed from the URI for security reason. If you need to keep the token in the URI, for example you want to bookmark the URI or create a iOS/Android Web App then you should use `?pt=<ADD_TOKEN_HERE>`, example: http://10.0.50.231/magicmenu?pt=<ADD_TOKEN_HERE>. The parameter is kept in the URI.

8.3.2.2 Page

You can access directly to a specific page using `home` parameter referencing the file after `views` folder.

For example, to go to the `room` page in `common` folder, you can write: <http://localhost/magicmenu/?home=common+room>

8.3.2.3 Point context

P 165 / 270

You can give the context to a page using the [point](#) parameter. You could do this to gain direct access to a room even if that room is just a template.

For example to go to the [room](#) page in [common](#) folder, using [Malaga_Room](#) as context, you can write: http://localhost/magicmenu/?point=Malaga_Room&home=common+room

8.3.2.4 Style

You can specify a custom CSS file to be used by Overture Magic Menu. By default, Magic Menu will load the stylesheet in the path [assets\styles\magicmenu.css](#). Adding a [theme=<filename>](#) parameter in the URI that directs to the Magic Menu, you can specify an alternative filename within [assets\styles\](#).

For example, to use an alternative CSS file with the path [assets\styles\red.css](#), add [?theme=red](#) to the Magic Menu's URI in the address bar: <http://localhost/magicmenu/?theme=red>

8.3.2.5 Auto Logoff Delay

[autoLogoffDelay](#) parameter is used to force the logout delay of Magic Menu in minutes. (The default time depends if you selected [Remember me](#) or not while login.)

Example: <http://localhost/magicmenu?autoLogoffDelay=5>.

In this case, the user will be redirected to login page every 5 minutes.

This setting is stored in the *session storage* of the browser. Which means it will persist during the lifetime of the page session (the tab).

If the parameter [autoLogoffDelay](#) is removed of the URL, MagicMenu will keep disconnecting the user with the specified delay until the page session is closed (close the tab), the session storage is cleared manually or the parameter is given the value 0: [autoLogoffDelay=0](#)

8.4 Home Extras

8.4.1 Widgets

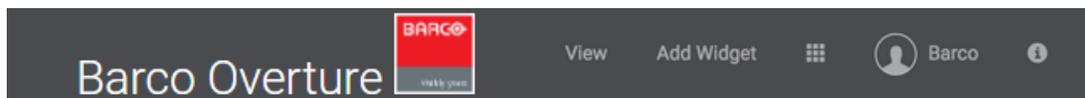
Home's dashboard is used to display information at a quick glance, or display data in a graph for easy comparisons. Each user is able to set up their own dashboard. Changing the definition of widgets in one user's login does not affect the widgets for other users.

By clicking "Add Widget", you can specify what type of widget you would like to see on the screen. Once chosen, you must decide what points to display with the widget. Each user is only able to specify points that he has 'read' access to. No other points are shown.

Table widgets are used to display information. If the point specified is also connected to a control panel, clicking the name of the point in the Table widget acts like a bookmark. The map jumps to where that control panel is and opens it up.

Graph widgets show information of points relative to each point. This is used when you need to track information against other information, like Projector Temperatures. Clicking on a point's name in a graph type widget hides that point's information.

8.4.2 Branding



Company branding can be added to Home to make the project more unique. To do so, you must add an HTML file in the [assets](#) named [home.branding.html](#) (formerly [dashboard.branding.html](#); see below).

This HTML is only meant to have images and basic inline styling (i.e., no `med-*` tags):

```
<span style="font-size: 24px; color: teal;">
  MyCompanyName
</span>


```

N.B. For users updating to 3.0.3/3.1.x and later who have already configured [dashboard.branding.html](#):

You can continue to use and update [dashboard.branding.html](#) as usual so long as either [home.branding.html](#) does not exist or is empty (ignoring HTML comments and whitespace). It is however strongly recommended that you use [home.branding.html](#) in anticipation of future versions, where [dashboard.branding.html](#) may not be supported anymore.

8.4.3 Dashboard Widget Chart Coloring

In addition to branding, [Overture](#) allows you to configure the colors used by the charts that appear as widgets in the [Home](#) dashboard (top) area. These colors can be changed in the file [home.config.json](#). The file contains a JSON-formatted list of hexadecimal color codes that will be applied to each chart in their order of appearance:

```
{
  "graphicColors": [
    "#00bcd4",
    "#009688",
    "#4caf50",
    ... etc
  ]
}
```

The portion of a chart that is affected by this color depends on its type:

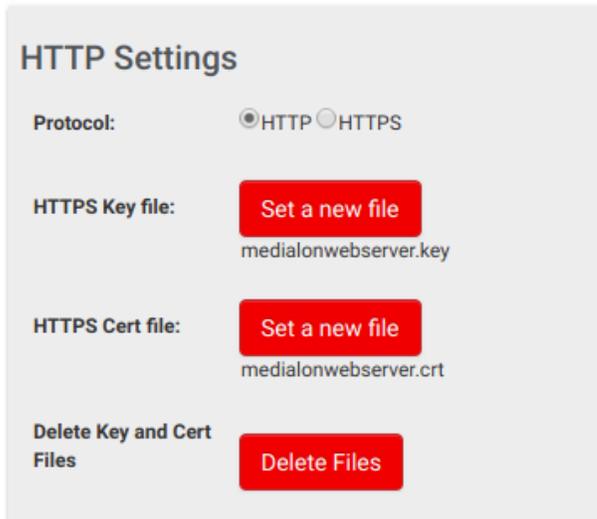
- Pie Chart: each segment, in clockwise order, each representing live variable values
- Bar Chart: each bar, in sequence from left to right, also representing live variable values
- Line/Trend Chart: each series (line), representing a variable's value over time

If there are more chart elements to be colored than there are colors in this list, the list simply loops back to the first color. This selection process repeats until all elements are colored. If no colors are in this list, or the [home.config.json](#) file is missing or otherwise cannot be read, colors for each chart element will be automatically generated.

N.B. For users updating to 3.0.3/3.1.x and later who have already configured [dashboard.config.json](#):

You can continue to use and update [dashboard.config.json](#) as usual so long as either [home.config.json](#) does not exist (i.e. is deleted from the default [Overture](#) assets distribution) or the color list contained in this file is removed. It is however strongly recommended that you use [home.config.json](#), which follows the exact same format, in anticipation of future versions, where [dashboard.config.json](#) may not be supported anymore.

8.5 HTTPS



If you wish to switch communications over to HTTPS, you can do so from the Configurator.

You will need to have the key and cert files provided by a trusted certifying company or else the browser will display an error and claim Overture to be "Not secure".

If you are running Overture On Premise, you can use the default Key and Cert files. Overture server won't be certified, and thus your browser will show an error, but still your communications will be encrypted.

8.5.1 Switching to HTTPS:

Switch the setting to HTTPS, then reboot your VM (Since UX version 3.3.1). Then you can go back in configurator and put your own key and cert files.

If the files are proper, the browser will no longer give errors when connecting. If not, please see your trusted certifying company.

If the key file is invalid, the web server will start in http instead of https, even if [https](#) is selected in configurator.

8.5.2 Switching to HTTP:

Important Note: From Overture 3.7.0 onwards, if the communication settings are set back to HTTP from a HTTPS configuration, Browser Cookies will need to be cleared in order to be able to login again through HTTP. This is linked to added security feature from version 3.7.0

8.6 Translations Quick Reference

8.6.1 Introduction

The [translations.json](#) file in the assets directory contains translation information for the alternative language used in Overture. This file is provided by default and can be edited at any time.

8.6.2 Getting Started

Open the [translations.json](#) file in Overture Editor. The top-level keys provided in the [translations.json](#) file provided with the installation are as follows:

8.6.2.1 [language](#) section

- This portion of the JSON represents some basic language information.
- Under "name", you can edit the language's internal name.
- Under "label", you can edit the name of the language that is displayed in the UI (in Overture Home under Views for example).
- Under "locale", you can edit the locale code used in the [MomentJs](#) library. This library is used by Overture for the purposes of date and time reporting (e.g. for logs). Please visit <http://momentjs.com/> under the section *Multiple Locale Support* as a reference for supported country codes.

8.6.2.2 application-name (e.g.: home, magicmenu) section

- This is where the Overture client app-specific translations can be found.
- The keys of this portion of JSON correspond either to literal text found in the UI, or labels corresponding to things like menu item text, etc.
- The values of this portion correspond to the desired translation of each of these terms/labels found in the UI.
- The [translations.json](#) file included in the installation contains translations to English to be used as a template that can be edited with desired values.

8.6.3 Fallbacks

- English UI text is currently built into all Overture client applications. If the [translations.json](#) file is unreadable or missing, English text will be used as a fallback.
- If the translation for a particular 'key' cannot be found (i.e. neither the key or value exist), English will automatically be used as the fallback language for that particular piece of UI text.
- This means that you are able to provide partial translations to the UI; any omitted translation key-value pairs will silently default to English.

9 Monitor

The screenshot shows the Overture Monitoring interface. At the top left is the logo and 'OVERTURE MONITORING'. On the right, there are navigation options: 'ADD WIDGET', 'VIEW', a grid icon, a user profile 'JOHN DOE', and an info icon. The main content area is divided into two widgets. The 'Clocks' widget displays three times: '03:27:57 PM' for Paris, '09:27:57 AM' for Montreal, and '06:27:57 AM' for Los_Angeles. The 'Alarms' widget shows '15 Active' alarms and '8 Inactive' alarms. Below these is a 'Grid 2' section with a 'Select Displayed' filter and a table of alarms.

Status	Name	Type	Status 1	Sharing Status	Driver Version ↑	Actions
✓	Lyon	room	---	---		[Icon] [Icon] [Icon]
✓	Lyon Audio	device	---	---		[Icon]
✓	Lyon Projector	device	---	---		[Icon] [Icon] [Icon]
✓	Lyon Light	device	---	---		[Icon]

9.1 RMM Monitor Main Grid Widget

A Grid Widget is made of three sections

9.1.1 Alarms is a filter providing 3 values:

All Alarms, Active Alarms, Inactive Alarms. When you select any of these values, the main grid section data gets filtered by your selection.

This screenshot shows the 'Alarms' filter selected in the 'Grid 2' section. The 'All Alarms' option is highlighted in the left sidebar. The table below shows a filtered list of alarms.

Status	Name	Type	Status 1
⚠	London ClickShare (SSG)	tenant	Connected
✓	London ClickShare	device	Connected
✓	Brussels Projector	device	Connected
✓	Madrid	room	---
✓	Brussels Lights	device	Connected
⚠	Paris	room	---
✓	Big Ben Lights	device	Connected
✓	Burj Khalifa Display	device	Connected

Active Alarms: Alarms that are still pending. The issue has not been resolved Inactive Alarms: Alarms for which the initial condition is not present anymore, but not yet acknowledged. The issues could have been resolved.

9.1.2 Tree View

Points are a tree view filter displaying all the tenants for which you have an access. Each tenant represents a node in the tree view. You can expand the node to access to its children levels until you get to the room level

When selecting a node (tenant) or its children, the grid data gets filtered by the selected level.

9.1.3 Main View

The main section displays the information on all points/variables from the previous filters.

By default, the columns are:

- **Status:** Shows active alerts, variable status error, Inactive Alerts, connection and ok status.
- **Name:** of the point
- **Type :** point type like Devices, Rooms, etc.
- **Actions:** Logs, Alarm acknowledge, On/Off
 - **Logs:** Logs are showing an aggregation of changes and issues on points and the children of points both from UX server and RMM. RMM will keep and display up to 1 million lines of logs. You can search logs by keyword, by type or by date. The Auto-Refresh can be turned on or off for the refresh of the log window.
 - **On/Off buttons:** On devices, rooms, floors, or any point which has a Power variable, it will turn the power on or off.
 - **Alarm Acknowledgement:** Depending if it is selected at a point with children, alarms are inherited by the parent point from children points. You may see several alarms under one acknowledgment. You can either acknowledge all at once or individually. The recommended best practice is to fix the issue before acknowledging it.

Next to each column, you can select 3 dots for additional actions: Filter/Sort the values, Hide/Remove or edit the column configuration

9.1.3.1 Add Column

Additional column could be added by selecting the hamburger icon on the top right corner, "Add Column". You can then access the point property/variable selector to select the relative point.

- **Type Variable** allows you to select from the tree view the value of the variable you want to capture in the added column
- **Type Point Property** allows you to select the value for various point's properties, to be in the added column, including:
 - Sub Type
 - Variable Name
 - Parent Name
 - Tenant Name
 - Path
 - Device Brand
 - Driver Name
 - Driver Version
 - Control Server Name

Each column can be resized, dragged and dropped to different location or to be regrouped.

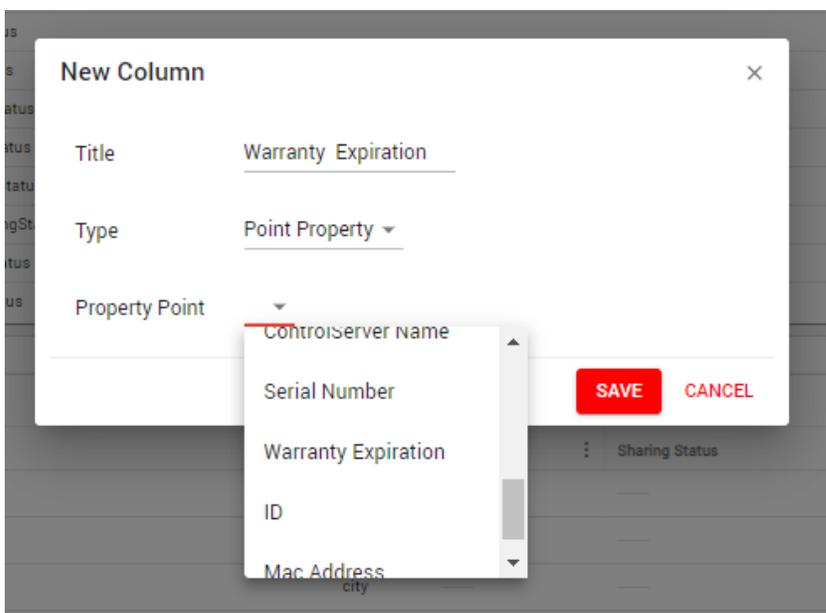
9.1.3.2 Custom Fields as Point Property Column

In RMM you can monitor the state of your AV assets tagged with "Custom Fields" in Grid widget. The Custom Fields are available with "Point Property" columns. You can filter or sort them to keep them under your radar!

Status	Name	Type	Status 1	Sharing Status	Driver Version ↑	Warranty Expiration	Serial Number	Mac Ad	Actions
✓	KEVRAM	tenant	Connected	---					📄
✓	Building site	building	---	---					📄
✓	Canada Map	city	---	---					📄
✓	Floor 1		---	---					📄
✓	Floor 1 Paris	room	---	---					📄 🔌
✓	Paris Light	device	Connected	---		2020-07-01	S-1234579	10.121.15.10	📄 🔌
✓	Paris Projector	device	Connected	---		2020-08-31	S-45679813		📄 🔌
✓	Paris Audio	device	Connected	---					📄 🔌
✓	Lyon	room	---	---					📄 🔌
✓	Lyon Audio	device	Connected	---					📄 🔌

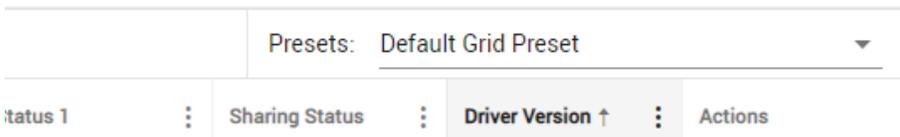
Showing Points: 12

You need to select "+ Add Column" in the "Grid Menu". Name the column with the name of the Custom Field, select "Property Point", and select the Custom Field from the list, save.



9.1.3.3 Grid Presets

Represents a single Grid configured with the specific columns, ordering, filtering and layout. Additional Grid Presets could be created to represent other settings (columns, ordering, filtering and layout) within the same grid. Each preset can be duplicated, renamed or deleted. This can be accessed via the 3 dots menu next to the preset name.



9.1.3.4 Grid Copy

The whole grid could be copied via the duplicate icon on top right corner. Once the grid has been duplicated you can select which grid will be your main grid.

9.1.3.5 The main grid

Only the main grid is filtered by the Alarm widget. The secondary grids will not be filtered by the Alarm widget.

9.1.3.6 Multiselecting

Within the grid, lines can be multi-selected by using shift and control buttons. Once several lines have been selected, a new menu will show on top left corner to Select/Deselect or to perform actions all at once:

- *Select Displayed*: Allows you to select all the points that you currently see in the Grid
- *Deselect Displayed*: Allows you to deselect all the points that are selected and you currently see in the Grid
- *Deselect all*: Allows you to deselect any point that may or may not currently display in the Grid. You may have selected points with a different filter and you do not currently see them in the Grid.

When multi-select is active, Bulk actions buttons are displayed on the top right side of the grid:

- *Bulk Log view*: gives you an aggregated view of logs of all selected points
- *Bulk Power on/off*: These commands will turn the power on or off for all the selected points if the Power variable exist
- *Bulk Acknowledge Alarms*: This action will acknowledge all alarms (if there is any) for all the selected points

9.2 Other RMM Monitor Widgets

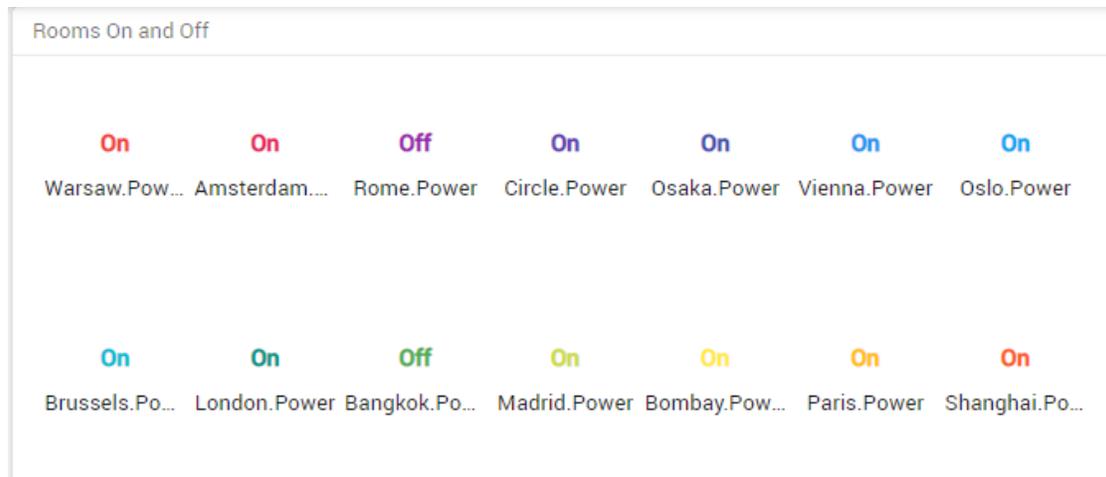
Overture RMM provides the versatility to users to create as many widgets as they need to further customize their dashboard.

Select "ADD WIDGET" on top menu:

Widget types:

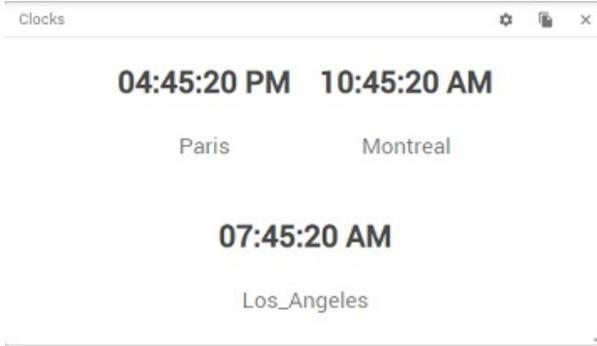
9.2.1 Big Text

A widget that displays the data as big text. You can name the widget, and then select the points/variables you want to report on.



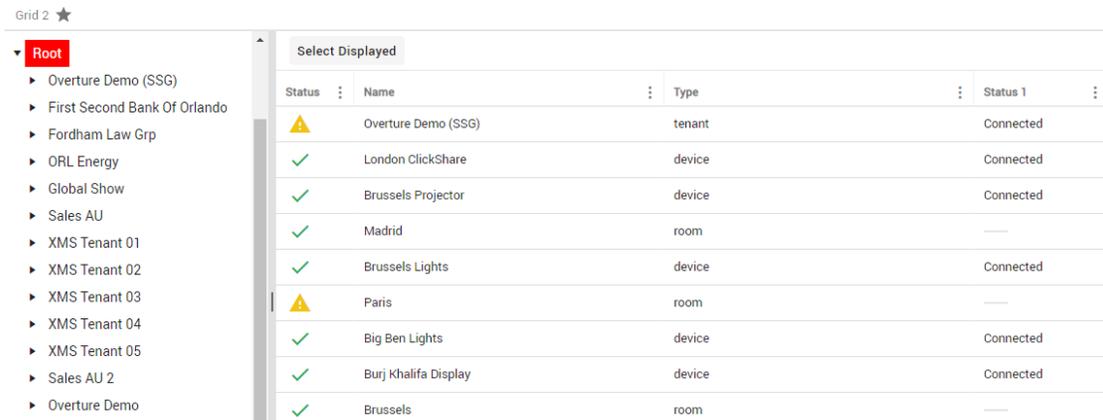
9.2.2 Date

You can select different time zones to be shown.



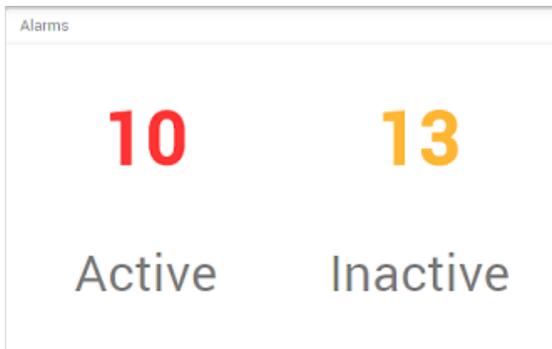
9.2.3 Grid

Same as the main grid.



9.2.4 Alarms

Displays Active and Inactive Alarms Counts. By selecting any of the two options, the main grid will be filtered to display points with active or inactive alarms.



9.2.5 Alarm Grid

Introduced with version 4.6.0, the Alarm Grid Widget displays all the Active and Inactive alarms by default. As with the Point Grid, Alarm Grid provides several columns to display alarms properties

Alarms Ongoing

Point	Alarm Name	Status	Seve.	Tenant	Variable	Value	Ticket Status	Actions
Osaka HVAC	testAlarm (CurrentTemperature) 3	Active	-	XMS Tenant 05	CurrentTemperature	73		
Osaka HVAC	testAlarm (CurrentTemperature) 2	Active	-	XMS Tenant 04	CurrentTemperature	73		
Osaka Projector	testAlarm (Temperature)	Active	-	Kevin Demo	Temperature	83		
Osaka HVAC	testAlarm (CurrentTemperature) 4	Active	-	XMS Tenant 03	CurrentTemperature	74		
Lincoln Display	Brightness too high	Active	Medium	First Second Bank Of Orlando	Brightness	100	INC0010001 Resolved	
Grant Display	Brightness too high	Active	Medium	First Second Bank Of Orlando	Brightness	100	INC0010002 New	
Rome Lights	testAlarm (LevelChannel3) 5	Active	-	XMS Tenant 05	LevelChannel3	90		

Alarm properties:

- **Point Name** of the point source of this alarm
- **Alarm Name** Name of the template source of this alarm (see [RMM Alarms](#) below)
- **Status** Current status of the alarm, possible statuses are:
 - *Idle* there is an alarm defined (an Alarm template associated with a point) but the condition is not currently met
 - *Active* the condition defined for this alarm is currently met
 - *Inactive* the condition defined for this alarm is not met anymore, but it was previously active and was not acknowledged
 - *Delaying* the condition defined for this alarm is currently met, but the delay defined for this alarm is not elapsed yet
 - *Snoozed* the condition defined for this alarm was met but the alarm has been put in snooze (Note: this functionality is not available yet)
 - *Disabled* there is an alarm defined (an Alarm template associated with a point) but it is disabled (Note: this functionality is not available yet)
- **Severity** this information is only available on native Monitor Alarm (see [RMM Alarms](#) below for details on native Monitor Alarm vs UX Alarms)
- **Tenant** Name of the tenant the point and alarm belong to
- **Variable** Name of the variable being monitored in this alarm
- **Value** Value of the variable being monitored in this alarm, which caused to alarm to go off. This value is not updated, as it indicates the value which has triggered the alarm.
- **Ticket Status** When an alarm is configured to trigger a notification, it indicates the ticket reference and status which has been created (see [RMM Alarms](#) for details on Notification).

Variable	Value	Ticket Status	Actions
CurrentTemperature	73		
CurrentTemperature	73		
Temperature	83		
CurrentTemperature	74		
Brightness	100	INC0010001 Resolved	
Brightness	100	INC0010002 New	
LevelChannel3	90		

- The ticket status indicates the ticket number/reference and the ticket current state
- The ticket number/reference is dependent on the ticketing system (this example is an incident from ServiceNow ITSM)
- The ticket state is also dependent on the ticketing system (this example is from ServiceNow ITSM, the state's values depend on the workflow defined in ServiceNow)
- **Actions** List of actions that can be done on an alarm
 - *Acknowledge* this button allows acknowledging the alarm. When acknowledged, an alarm will go off again



if the condition goes back to false and then becomes true again.

The Alarm Grid Widget has a tree view on the left which allows narrowing down (filtering) the displayed alarms, up to the lower leaf of the tree (generally the room). Like in the Point Grid widget, next to each column, you can select 3 dots for additional actions: Filter/Sort the values.

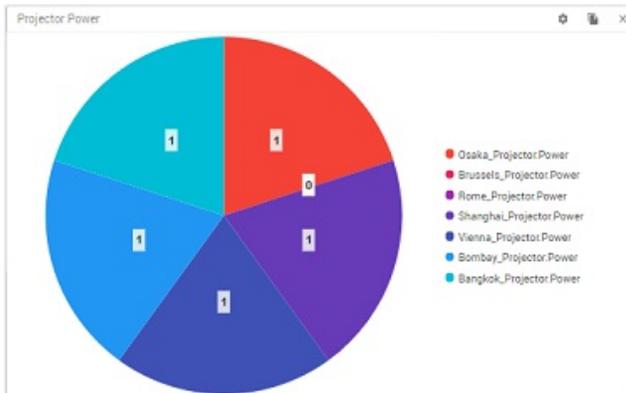
9.2.6 Table

Displays data in table format with the name of the point and its value.

ClickShare Sharing status	
Name	Value
Warsaw_ClickShare.SharingStatus	Not Sharing
Paris_ClickShare.SharingStatus	Not Sharing
Oslo_ClickShare.SharingStatus	Sharing
London_ClickShare.SharingStatus	Sharing
Madrid_ClickShare.SharingStatus	Sharing
Brussels_ClickShare.SharingStatus	Sharing
Amsterdam_ClickShare.SharingStatus	Not Sharing
Vienna_ClickShare.SharingStatus	Sharing
Rome_ClickShare.SharingStatus	Not Sharing

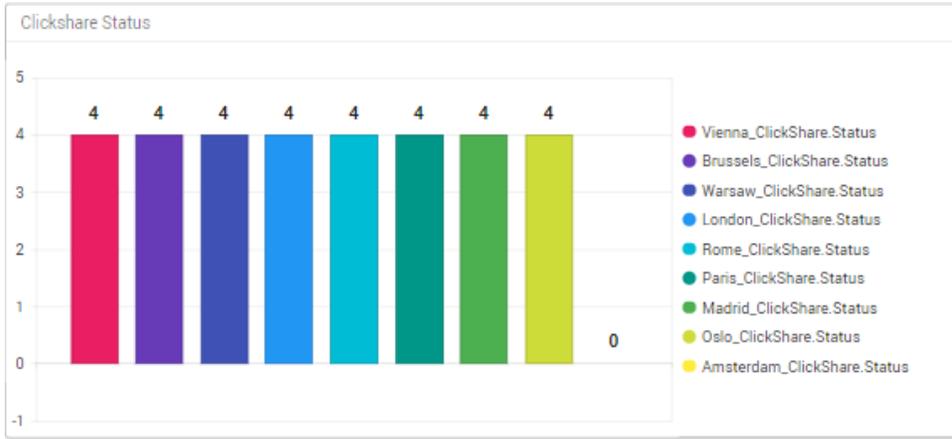
9.2.7 Pie Chart

Displays the data in a pie chart. The selected points could be shown with different color codes for better visibility.



9.2.8 Bar Chart

Displays the data in a multitude of bars with color coding for better visibility. You could define minimum and Maximum limits for Y axis to show



9.2.9 Common Widget Options

All Widget must have a title and a type.

For Grids: you need to select a Position (Floating, Bottom, Top, Left, Right) and a check box to force it to be considered as the Main Grid.

For Big Text, Table, Pie Chart and Bar Chart, you need to select from Label either Point Variable Name, Human Name or Short Name. The selected value for the Label will be displayed on the widget.

Once the label is selected, you need to click on "SELECT POINTS"

TREE VIEW FILTER

- ▼ Root
- ▶ Overture Demo (SSG)
- ▶ First Second Bank Of Orlando
- ▶ Fordham Law Grp
- ▶ ORL Energy
- ▶ Global Show
- ▶ Sales AU
- ▶ XMS Tenant 01
- ▶ XMS Tenant 02
- ▶ XMS Tenant 03
- ▶ XMS Tenant 04
- ▶ XMS Tenant 05
- ▶ Sales AU 2
- ▶ Overture Demo
- ▶ KEVRAM
- ▶ Gateway NYC

Search

<input type="checkbox"/>	Name	Type	Tags	Tenant
<input type="checkbox"/>	Overture Demo (SSG)	tenant		Overture Demo (SSG)
<input type="checkbox"/>	Overture Demo (SSG) Status	variable		Overture Demo (SSG)
<input type="checkbox"/>	Shanghai AV Conference AddressBook	variable		Overture Demo (SSG)
<input type="checkbox"/>	BigBen Display Status	variable		Overture Demo (SSG)
<input type="checkbox"/>	Paris Lights Status	variable		Overture Demo (SSG)
<input type="checkbox"/>	Rome Lights Status	variable		Overture Demo (SSG)
<input type="checkbox"/>	Meeting Deck 4 Power	variable		Overture Demo (SSG)
<input type="checkbox"/>	London Display Contrast	variable		Overture Demo (SSG)
<input type="checkbox"/>	Paris Lights Level (Share) 1	variable		Overture Demo (SSG)

Number of selected points: 0 of 11265

SAVE **CANCEL**

Title
Type
✕

Clickshare Status
Bar Chart

Label

Points

Point Variable Name ▾

SELECT POINTS

DELETE ALL POINTS

Vienna ClickShare Status	● #E91E63	
Brussels ClickShare Status	● #673AB7	
Warsaw ClickShare Status	● #3F51B5	
London ClickShare Status	● #2196F3	
Rome ClickShare Status	● #00BCD4	
Paris ClickShare Status	● #009688	
Madrid ClickShare Status	● #4CAF50	
Oslo ClickShare Status	● #CDDC39	
Amsterdam ClickShare Status	● #FFEB3B	

Limits

Min	Max
-1	5

SAVE

CANCEL

In the Popup point selector, you find 2 tabs: The tree view that allows you to drill down to the exact point or node called the absolute path to a point or the Filter which allows you to add rules and specify the relative path within the current context.

You can do this by selecting points that belong to a given tenant or are descendant of a given point and even specify the depth of the parent child level, or by searching based on human name, short name, variable name or by the type, sub-type or tags. You can also have multiple rules that add up.

Once all points are selected, you select color coding for each point (only available for Big text, Pie and Bar Charts) based on the type of widget.

Notes:

- Once a widget is created, you can edit it by changing the selected points or the title with the exception the chart type.
- Thanks to auto refresh all widgets are showing live data. Any changes on the variable or state of points will be displayed in real time.
- Widget could be resized, duplicated and moved with the dashboard page.

9.3 RMM Alarms and Notifications

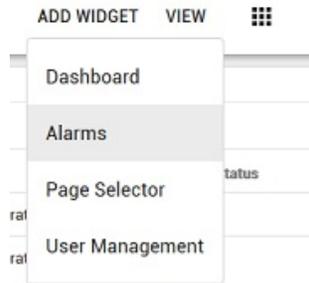
Introduced with version 4.6.0, Monitor provides native alarms, which can be configured and monitored directly from Monitor. Overture UX alarms are still taken along, with restricted functionalities as opposed to native Monitor Alarms.

Alarms apply to points (for instance a device or a room), and are defined by a condition on one or more variables of those points.

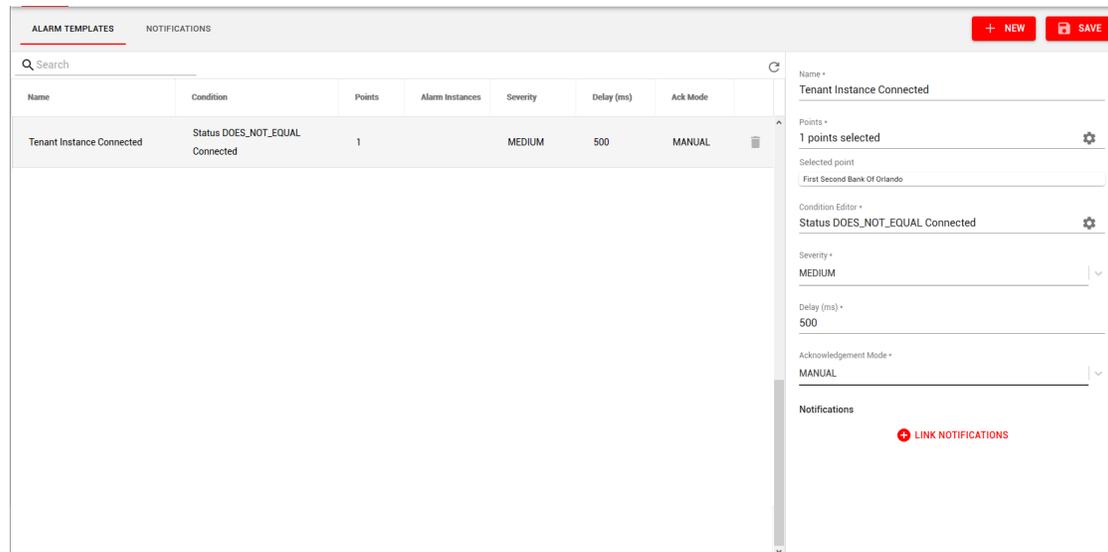
9.3.1 Alarm Templates

Like many elements within Overture, Alarms are defined through templates. An alarm template defines rules for an alarm, as well as the points to which this template applies. This is a flexible way to define one rule and apply it for multiple points.

Alarm templates are accessible using the View -> Alarms menu, and then 'Alarm Templates' tab



Alarm Templates view lists all templates, whether they are native Monitor Alarm templates or alarm templates imported from Overture UX alarms. Use the Search bar to filter out the templates.



9.3.2 Monitor native alarms

A Monitor native alarm is an alarm created from an alarm template defined in Monitor (see [Alarm Templates](#) chapter to access Alarm Templates page)

Click on the New button to create a new alarm template, and enter the required parameters:

+ NEW
SAVE

Name *

Tenant Instance Connected

Points *

1 points selected ⚙️

Selected point

First Second Bank Of Orlando

Condition Editor *

Status DOES_NOT_EQUAL Connected ⚙️

Severity *

MEDIUM | v

Delay (ms) *

500

Acknowledgement Mode *

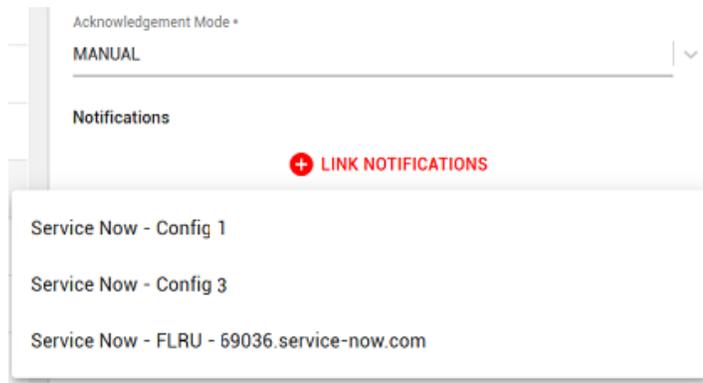
MANUAL | v

Notifications

+ **LINK NOTIFICATIONS**

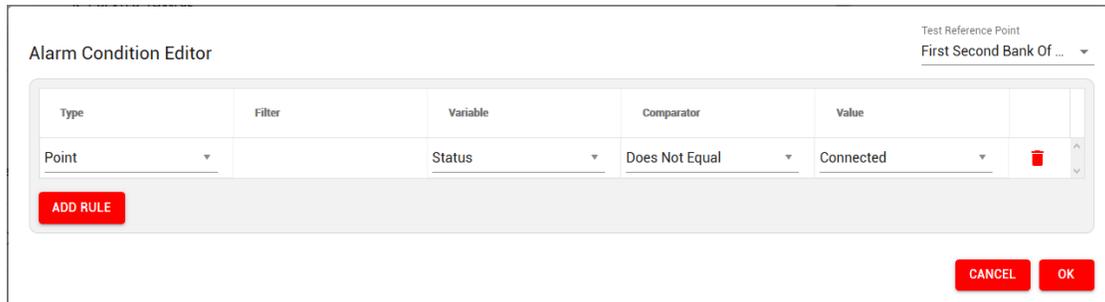
Alarm Template parameters:

- *Name* is the name of the template, which will also be the name reported when an alarm from that template goes off. It is therefore important to give a meaningful name to your template.
- *Points* is the list of points selected through the point selector (Point Selector opens when the gear icon on the right is clicked), to which this template will apply.
- *Condition Editor* is the condition that defines the alarm (Condition Editor opens when the gear icon on the right is clicked). An alarm condition is defined through rules on the point values and/or parent, grand-parent, child, ancestor or sibling's point values. See [Alarm Condition Editor](#) chapter for details on Condition Editor
- *Severity* Defines the level of severity of alarms, levels are
 - *Highest*
 - *High*
 - *Medium*
 - *Low*
 - *Lowest*
- *Delay* is the time in millisecond during which an alarm condition must be true before the alarm actually goes off
- *Acknowledgement Mode*:
 - *Manual* Whether the alarm is Active or Inactive, a manual acknowledgement is required to acknowledge the alarm
 - *Auto* If an Active alarm goes back to Inactive (the alarm condition has been true but is not valid anymore), the alarm will automatically be acknowledged
- *Link Notifications* A notification can be added to the alarm template to be sent when an alarm goes off (see [Notifications](#) chapter for details on Notifications)



9.3.2.1 Alarm Condition Editor

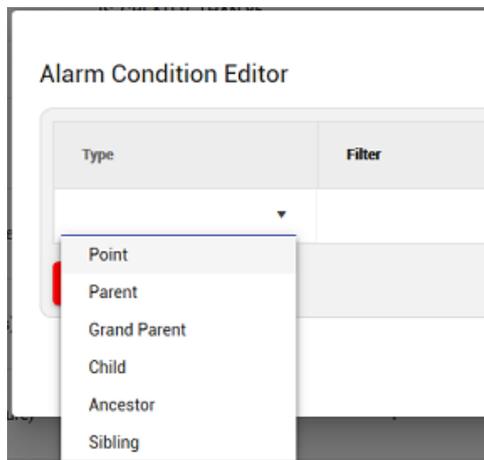
Alarm Condition Editor allows to define the condition that characterises an alarm template. An alarm template being associated with multiple points, this condition will be valid for all associated points, provided the properties defined in the condition are shared between the points.



Once the Alarm Condition Editor is opened, the first up-right corner setting is the point reference which will be used as reference to create the condition. You can select any points that have been associated with this alarm template.

Below, there is a series of rules, one being the minimum required, which compose the condition. For each rule, you can define:

- *Type* is the point itself, a sibling, a descendant or ascendant of the current point. Available types are:
 - *Point* is the point itself
 - *Parent* is the point parent to the current point. Example current point is a Projector, parent point is the Room the projector belongs to
 - *Grand Parent* is the point grand parent to the current point.
 - *Child* is the point child to the current point. Example current point is a Room, child point is a Projector belonging to this room.
 - *Ancestor* is any point ancestor to the current point. This includes Parent and Grand Parent points described above.
 - *Sibling* is a point with the same parent as the current point. Example current point is a Projector, sibling point is a Video Conferencing device, both belonging the same parent

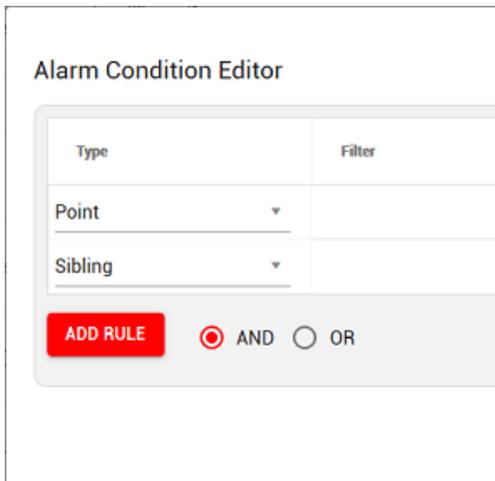


- *Variable* is the property to be checked for the condition
- *Comparator* is the operator used to check the property
- *Value* is the value of the property to check against, using the operator

In the example above (Condition Editor view), the condition checks the Status variable of the current point to be anything but Connected (Does Not Equal to Connected). Any point associated with this alarm template, for which his Status variable is not Connected will raise an alarm.

Adding rules to a condition

More sophisticated conditions can be defined adding rules to a condition, as illustrated on the picture below:



When multiple rules are defined within a condition, the rules can be either evaluated with an OR or an AND operator between them. For example, if the point is a Projector and the second rule refers to a sibling Conferencing device, the condition could be:

- 'Projector.PowerStatus' 'Does Not Equal' 'On' AND 'Conferencing.SharingStatus' 'Equal' 'Sharing'
- 'Projector.Status' 'Does Not Equal' 'Connected' OR 'Conferencing.SharingStatus' 'Does Not Equal' 'Connected'

9.3.3 Overture UX alarms

Alarms which have been defined at Tenant level (i.e. in Overture Configurator for a tenant) are also available in Monitor, with few limitations compared to Monitor native alarms. Overture UX Alarms being based on templates as well, they are imported as templates in Monitor Alarm Templates view:

Name	Condition	Points	Alarm Instances	Severity	Delay (ms)	Ack Mode
ShutterClosed (Shutter)	Shutter EQUALS Closed	3	0	N/A	0	MANUAL
TempReallyTOHot (CpuTemperature)	CpuTemperature IS_GREATER_THAN 165	1	0	N/A	0	MANUAL
LampHours1500 (HoursLamp1)	HoursLamp1 IS_GREATER_THAN 1500	1	0	N/A	0	MANUAL
testAlarm (CurrentTemperature)	CurrentTemperature IS_GREATER_THAN 50	1	1	MEDIUM	0	MANUAL
DeviceOff (Power)	Power EQUALS Off	3	3	MEDIUM	0	MANUAL
TempTooHigh (LevelChannel3)	LevelChannel3 IS_GREATER_THAN 85	1	1	MEDIUM	10000	MANUAL
TempTooHigh (Status)	Status IS_GREATER_THAN 85	1	1	MEDIUM	10000	MANUAL
VolumeTooHigh (AudioLevel)	AudioLevel IS_GREATER_THAN 80	2	2	N/A	0	MANUAL
VolumeTooHigh (Brightness)	Brightness IS_GREATER_THAN 80	1	1	N/A	0	MANUAL

The picture above shows a list of Alarm templates, some of them are Monitor native Alarms, others are Overture UX Alarms.

Differences between native and UX alarms:

- Overture UX Alarms don't have severity, displayed as N/A
- Overture UX Alarms cannot be edited/deleted within Monitor, right properties panel is grayed out and Delete icon is not available. They can still be modified from the Tenant Overture Configurator page.
- Notifications can be linked to both Overture UX and Monitor native Alarms

9.3.4 Notifications

Together with Alarms, version 4.6.0 introduces Notifications, which can be linked to an Alarm Template (see [Alarm Templates](#) for details). Essentially, when an Alarm goes off, besides visualizing this alarm in Monitor, a notification can also be triggered.

Current supported Notifications:

- ServiceNow Ticketing System (please see [APPENDIX E - Overture ServiceNow Integration Guide](#) for details on how to configuration ServiceNow for Overture integration)

Notifications are accessible using the View -> Alarms menu, and then 'Notifications' tab

The Notifications view lists the Notifications systems which have been configured.

Name	TYPE
TEST - ssg666.service-now.com	Ticketing
PROD - ssg888.service-now.com	Ticketing
FLRU - 69036.service-now.com	Ticketing

On the right settings pane, the default fields are the Notification Type and the name:

- Type is the kind of Notification. Only Ticketing is supported currently.

- *Name* is the name given to a notification configuration. For example, if you have a ServiceNow account for your organisation, and another one for a customer, you may want to create a configuration for each, providing a meaningful name

Additional settings are dependent on the Notification system being configured. However, for most of them, the settings are splitted in steps, the number of steps also depends on the Notification system being configured.

The example below shows the steps for a ServiceNow configuration:

- Click on New to create a new notification, select the type of notification (here ServiceNow) and provide a name for this configuration

Name	TYPE
Config Test 1	Ticketing
Config test 1-1	Ticketing
FLRU	Ticketing
Daro	Ticketing

- *Step 1* is the connection part, fills-in with the parameter from your ServiceNow instance (please see [APPENDIX E - Overture ServiceNow Integration Guide](#) for the ServiceNow part)

Notification Type
Ticketing - ServiceNow

Name *

Daro

ServiceNow Config - Step 1

URL: *

https://T822.service-now.com

User Name: *

barco_rest_user

Password: *

●●●●●●●●

client Id: *

cb291bceb4d41010fde5e61010d2ff6e

client Secret: *

●●●●●●●●

- *URL* is the address of your instance
- *User Name* is the name of the user that you created for this access in ServiceNow
- *Password* is the password associated to this user
- *Client ID* is provided while the oAuth API access has been configured in ServiceNow

- *Client Secret* is provided while the oAuth API access has been configured in ServiceNow
 - Click on *Apply* to go to Step 2 (this action verifies the connection with ServiceNow, and if successful, displays Step 2)
- *Step 2* is the actual configuration part, mostly allowing to define the mapping between Overture information and ServiceNow ticket information

ServiceNow Config - Step 2

category *
software ▼

newStateName *
New ▼

resolvedStateName *
Resolved ▼

canceledStateName *
Canceled ▼

closedStateName *
Closed ▼

Resolved Reason: *
Acknowledged by Overture User

Field Mapping:

Mapping	
overtureField Serial number ▼	-
serviceNowField Configuration item ▼	

+ ADD ITEM

APPLY

- Select the category of incident you want Overture to create
- Select the state's names accordingly to your workflow in ServiceNow, Overture will use them to know what value are used to define if an incident is New, InProgress or Closed/Resolved
 - *Canceled State Name*: The status in which Overture should set the ServiceNow ticket when the alarm is acknowledged while the option "Auto Cancel Ticket" is enabled.
 - *Closed State Name*: The status of the ServiceNow ticket in which Overture will acknowledge the alarm if the option "Auto Acknowledge Alarm" is enabled. It also indicates to Overture which state a ticket is considered closed.
- *Resolved Reason* is a text which will be used when a Incident is automatically Canceled by Overture
- *Field Mapping* section is optional, and gives the ability to map a particular Overture alarm property to a ServiceNow incident field. In the example above, the "Serial Number" custom field in Overture is mapped to the "Configuration Item" field in the ServiceNow Incident. Note that certain fields in ServiceNow depends on other entities existing in ServiceNow like "Configuration Item". In order for the mapping to work, the values provided by Overture must already exist in ServiceNow

- Click on *Apply* to validate, which, upon success will enable the save button, on the top right corner
- When all the steps have been performed, you need to click on the Save button (upper right corner) to actually store this configuration into Overture.

The screenshot shows a configuration form titled "ServiceNow Config - Step 2". At the top right, there are two red buttons: "+ NEW" and "SAVE". The form contains several dropdown menus and text fields:

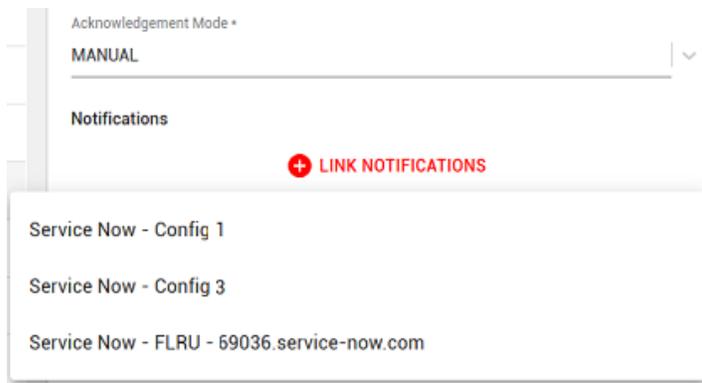
- client Secret: * (masked with dots)
- category * (dropdown menu with "hardware" selected)
- newStateName * (dropdown menu with "New" selected)
- resolvedStateName * (dropdown menu with "Resolved" selected)
- canceledStateName * (dropdown menu with "Canceled" selected)
- closedStateName * (dropdown menu with "Closed" selected)
- Resolved Reason: * (text field with "Acknowledged by Overture User")
- Field Mapping: (section with an empty list and a "+ ADD ITEM" button)
- APPLY (button highlighted with a red box)

Note: If the category or states couldn't be selected or if the Field Mapping couldn't display the available fields and property, this is an indication that your ServiceNow instance might not be configured properly, especially regarding access rights, see [APPENDIX E - Overture ServiceNow Integration Guide](#) for details.

9.3.4.1 Link notification to an alarm

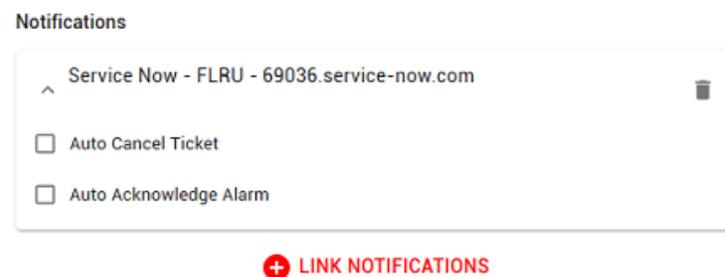
Whether with a Monitor native or UX alarm template, a notification can be linked to an alarm template using *Link Notifications* option (see [Monitor native alarms](#))

Click on the *Link Notifications* button to add a notification to an alarm template



Select the notification service you'd like to use. Most likely only one notification service will be configured at a time, but for cases where multiple ticketing systems are used, this menu will show them all.

The settings found in the link configuration depend on each notification service. The example below shows the settings for a ServiceNow notification service:



- *Auto Cancel Ticket* when checked, this option will cause a ticket to be cancelled if this alarm is acknowledged.
- *Auto Acknowledge Alarm* when checked, this option will cause this alarm to be acknowledged automatically when the ticket initially created got resolved.

Note : Although Monitor can have multiple notification services configured, only one link per alarm template at a time is possible.

9.3.5 Alarms and Notifications considerations

Defining an alarm which triggers a notification is handy, but could also result in far too many notifications which becomes useless, if not configured appropriately. This is particularly true when the notification is a service ticketing system such as ServiceNow. Here are some recommendations, best practice to avoid such situations:

- Add a delay in the alarm definition (see [Monitor native alarms](#)), so temporary alarm conditions are not accounted and does not trigger a notification. A notification is only triggered when the alarm condition has been stable for that period of time
- Make sure to select [Acknowledgement Mode:Manual](#) when the [Auto Cancel Ticket](#) option is used in Service Now notification, to avoid multiple tickets to be created in case the alarm condition changes back and forth between [Active](#) and [Inactive](#)
- Use the [Auto Cancel Ticket](#) option in Service Now notification, when really necessary
- When [Auto Acknowledge Alarm](#) option in Service Now notification is used, make sure to close/resolve the ticket after the issue which has generated the alarm has been fixed

9.4 RMM User Management

Access to User Management is through VIEW menu

The screenshot displays the 'User Management' interface in the Overture Monitoring application. At the top, there is a navigation bar with 'VIEW', a user profile for 'JOHN DOE', and a notification icon. Below this is a sub-navigation bar with 'USERS', 'GROUPS/TENANTS', and 'IDENTITY PROVIDERS'. A 'Dashboard' dropdown menu is visible, and there are '+ NEW' and 'SAVE' buttons. The main area is divided into two parts: a table of users on the left and a detailed form for the selected user on the right.

Name	Username	Identity Provider
	admin	
admin2 admin2	admin2	
John Doe	john_doe	
Jessica Abasta	jessicaab	
Evan Westbury	evan_westbury	
Jodie Lashua	jodielashua	
Tamra Aslest	tamas	
Milford Lakin	milford.lakin	
Rose Nesbit	rosenes	
Ahmad Deem	ahmad_deem	
Gretta Shibata	greshibata	
Fermin Puthoff	fermin.puthoff	

The detailed form for 'John Doe' includes the following fields and options:

- Username: john_doe
- First Name*: John
- Last Name*: Doe
- Email*: john.doe@ssg-corp.com
- CHANGE PASSWORD button
- Groups:
 - Employees
 - Insights Team
 - Core Team
 - admin

You can create users manually or specify the identity provider of the company.

9.4.1 Adding new users

When creating users, you select the +New button. You need to enter

- the Username
- the First Name
- The Last Name
- The user email address
- Specify if you want him to change his password upon first login
- Make him a member of an existing group.

9.4.2 Groups / Tenants

The Groups and Tenants allow you to specify the access right for the user. On the tenant level, you allow the user to have visibility on selected tenants.

GROUPS/TENANTS		IDENTITY PROVIDERS	
		Name*	Barco Employees
		Access Rights	<input type="checkbox"/> Administration Rights
		Tenants Access	<input checked="" type="checkbox"/> Overture Demo (SSG) <input checked="" type="checkbox"/> First Second Bank Of Orlando <input checked="" type="checkbox"/> Fordham Law Grp <input checked="" type="checkbox"/> ORL Energy <input checked="" type="checkbox"/> Global Show <input checked="" type="checkbox"/> Sales AU

Groups membership gives to its members certain access rights.

You can create a New Group by selecting +New.

You need specify:

- The Name
- If you want to give the Administration Rights (this will give the user access to User Management section)
- Which tenant(s) user can see
- Users part of this group

9.4.3 Identity Providers

Currently only 1 Identify Provider of type ADFS is available. To set it up you need to select +New and provide the following information:

- Redirection URL
- Alias
- Display Name
- Communication mode
- Alias of authentication flow after first login
- Alias of authentication flow after each login
- Single Sign-On Service URL
- Backchannel Logout option
- Name ID Policy Format
- Handshake options

9.4.4 Identity Provider Mappers

You access the mapping by selecting the Gearbox Icon on the ADFS Name line

USERS		GROUPS/TENANTS		IDENTITY PROVIDERS	
ID	Name	Redirection			
barcoadfs	Barco Login				

Type
ADFS

Redirection URI
https://sbg-rmm-tst02.overture.barco.cloud/auth/realms/ovt-tst02-s

Alias *
barcoadfs

Display Name
Barco Login

Enabled Store Token Store Tokens Readable

Hide On Login Page

First Login Flow
First Broker Login

Post Login Flow

SAML Config

Single Sign-On Service URL *
https://signin.barco.com/adfs/ls/IdpInitiatedSignon.aspx

Single Logout Service URL

Backchannel Logout

NameID Policy Format
Windows Domain Qualified Name

HTTP-POST Binding Response HTTP-POST Binding for

Want AuthnRequests Signed Want Assertions Signed

Force Authentication Validate Signature

You can associate the user groups that will use the company ADFS

By selecting the Mapper Type :

- Attribute Importer or SAML Attribute to Group
- Setting a name
- Attribute Name
- Friendly Name
- Groupe selection

barcoadfs Mappers [NEW](#) [SAVE](#)

Name	Type	Category	
http://schemas.xmlsoap.org/claims/Group	SAML Attribute to Group	Role Mapper	
First Name	Attribute Importer	Attribute Importer	
Last Name	Attribute Importer	Attribute Importer	
Email	Attribute Importer	Attribute Importer	

Mapper Type ▼

Required Format the username to import.

Name *

Attribute Name

Friendly Name

Attribute Value

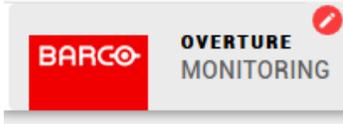
Group ▼

Values the attribute must have. If the attribute is a list, then the value must be contained in the list.

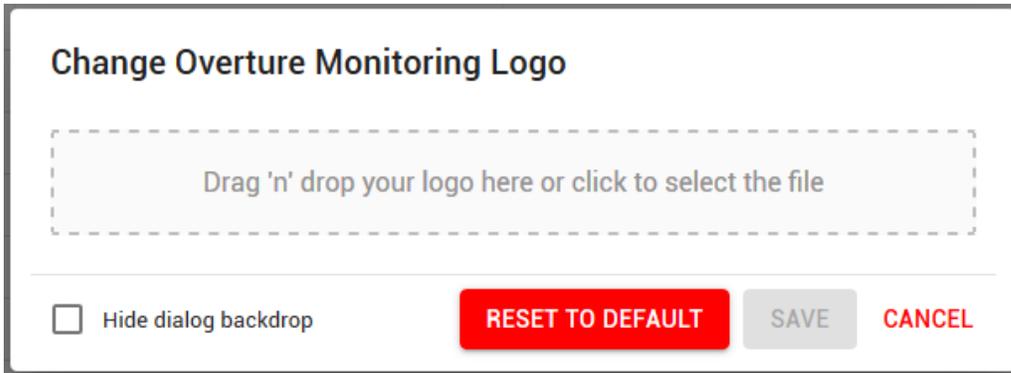
9.5 RMM Custom Logo

Both RMM Monitor and RMM Insights header could be customized. The default displays Barco Logo. This could be replaced with your own logo. This functionality is optional and to make it available, you would need to make a request through your dedicated Barco sales associate. Although a custom logo is applied both on RMM Monitor and RMM Insights, the button to change the logo is only available on RMM Monitor.

On RMM Monitor, hovering over the logo on the top left corner of the menu bar, shows an edit icon to change the logo

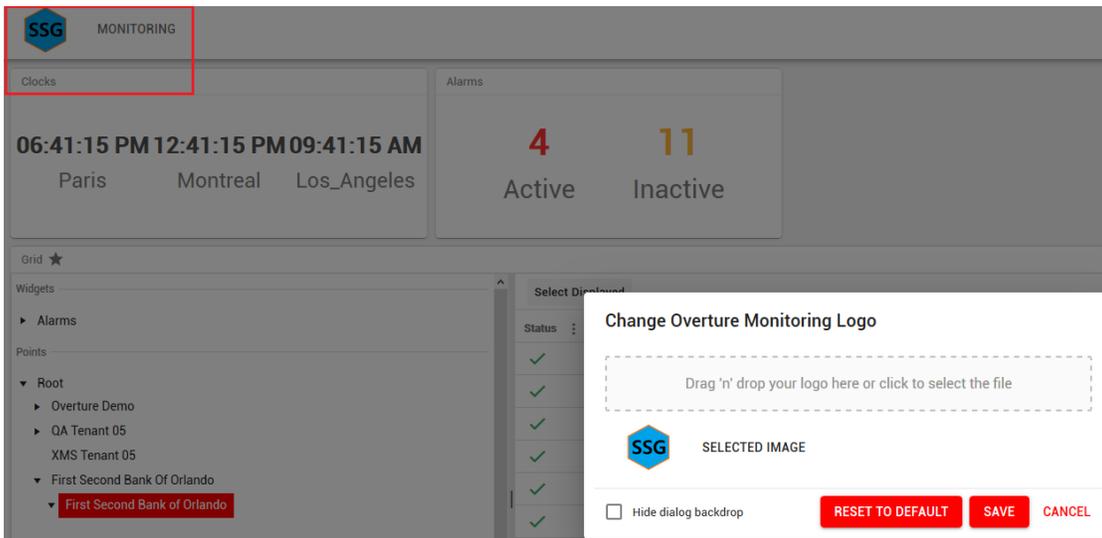


Clicking the edit button opens a dialog to change the logo



- *Hide dialog backdrop* Show or hide the backdrop behind the dialog to help visualizing the logo when changed
- *Reset to Default* Restore the original Barco logo
- *Save* Confirm the new uploaded logo and save it
- *Cancel* Close the dialog without further action

Upload the new logo into the drag'n drop area or click to open the open dialog box. The maximum size of the logo is 300x300px, it is then recommended to use an image file with either a maximum width or maximum height of 300px. Accepted image file formats are png, jpg, jpeg, gif and svg.



Once uploaded, the logo is displayed both in the upload dialog and in the menu bar as a preview. Use *Hide dialog backdrop* to better preview the new logo.

Clicking on *Save* confirm the change to the uploaded logo, the upload dialog closes and the new logo is now displayed on RMM Monitor as well as RMM Insights.

Note: Overture name is not displayed when a custom logo is used.

MONITORING

Clocks	Alarms
<p>06:47:33 PM 12:47:33 PM 09:47:33 AM</p> <p>Paris Montreal Los_Angelos</p>	<p>4 11</p> <p>Active Inactive</p>



10 Insights

10.1 RMM Insights Access

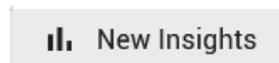
All users having access to RMM Monitor can create Insights pages. The access rights on tenant (which tenants are visible to the user) is also valid for Insights reports.

To create an Insights page

- Expand the pages pane, on the right end side if the Monitor view.
- Click on *Add* button
- Select *New Insights*

The screenshot shows the RMM Insights interface. At the top, there's a header with 'ACTIVE HOURS 5H00 - 20H00', 'ADD REPORT', 'VIEW', a grid icon, a user profile 'JOHN DOE', and an info icon. Below the header is a large empty space for reports. On the right, a 'Pages' sidebar is visible with 'ADD' and 'DELETE' buttons. The sidebar contains a search icon, a 'New Monitor' button, a 'New Insights' button (highlighted), and two existing insights: 'Insights Power' and 'Insights RB vs Usage'. A tooltip points to the 'New Insights' button, stating: 'The Insights page type allows adding widgets to see charts with historical data only'. Below the sidebar, a bar chart is displayed with the x-axis labeled with dates from '20 Fri' to '31 Sun' and the y-axis representing values. The bars show varying heights, with the highest bar on '20 Fri' and the lowest on '29 Mon'.

- Select the new create Insights page



- Name the new created Insights page to reflect the type of reports you intend to display in this page

10.2 Selecting, customizing your report widget

10.2.1 Insights Widget Overview

All reports are displayed in Insights dashboard pages as a Widget. In order to add a Report Widget, the user must select

“ADD REPORT” in the top menu.



Widgets can be dragged and dropped in different section of the dashboard and resized based on the user preferences.

Note: The color coding is based on values. All similar values have the same color tones.

10.2.2 Selecting and Customizing a Report

Through the Report setup dialog, each report widget can be targeted to a specific report metric, and customized using the various options.

Title *
Device Power Montly

Report Metric
Power

Period Monthly view Group by Period

Period Range June 2020

Point Types device

Point Subtypes All

Ancestor(s)
Name Contains

- Root
 - Overture Demo
 - ORL Energy
 - First Second Bank Of Orlando
 - Fordham Law Grp

SAVE CANCEL

Once the proper report is selected and configured, you must select the save button.

10.2.3 Using Reports Options

Report widget setup dialog sections

Title * 2
Device Power Montly

Report Metric 1
Power

Period Monthly view Group by Period 3

Period Range June 2020

Point Types device 4

Point Subtypes All

Ancestor(s)
Name Contains

- Root 5
 - Overture Demo
 - ORL Energy
 - First Second Bank Of Orlando
 - Fordham Law Grp

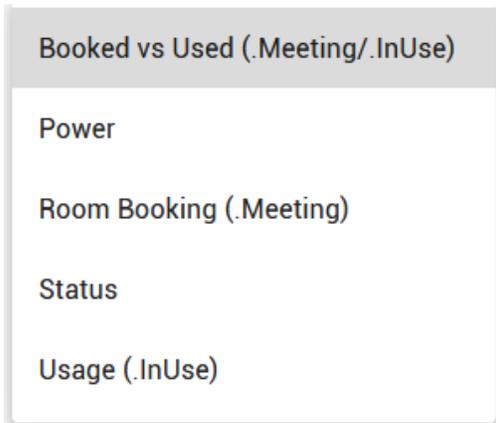
SAVE CANCEL

10.2.3.1 Report Metric

Select (section 1 of the Report widget setup dialog) the Report Metric for the widget, current available metrics are:

- Usage
- Power
- Room Booking
- Room Booking vs Usage
- Connection Status

Selection of a Report Metric



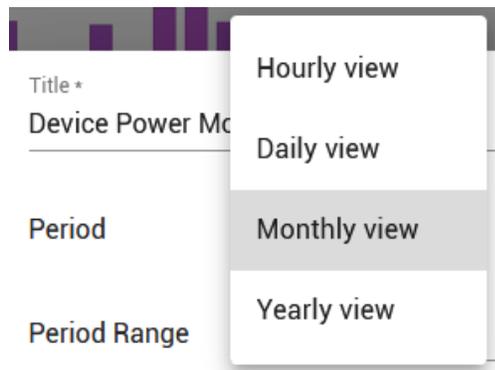
10.2.3.2 Title

Give a meaningful title to your widget will help reading your dashboard page and differentiate your reports (section 2 of the Report widget setup dialog).

10.2.3.3 Period and Group by

The section 3 defines the period for your report as well as the way the data will be grouped.

- Available *Periods* are:
 - Hourly View (gives a value, either an average or a sum depending on the Metric, for each hour of the day)
 - Daily View (gives a value, either an average or a sum depending on the Metric, for each day of the week)
 - Monthly View (gives a value, either an average or a sum depending on the Metric, for each day of the month)
 - Yearly View (gives a value, either an average or a sum depending on the Metric, for each month of the year)



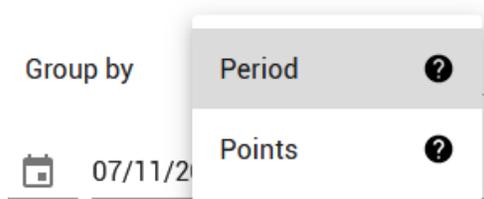
Depending of the selected *Period*, the *Period Range* has different parameters:

- *Hourly View* Period Range defines the first day and the last day of the period. Since the view displays each hour of the day, if the period covers several days, an average is done for each hour of the days across the period.
- *Daily View* Period Range defines the first day and the last day of the period. Since the view displays each day of the week, it is recommended to select a period ranging from a Sunday to a Saturday. If the period covers several weeks, an average is done for each day of the week across the period, average of Mondays, Tuesdays, etc...
- *Monthly View* Period Range defines the month and the year (ex: May 2020)
- *Yearly View* Period Range defines the year (ex: 2020)

The resulting dataset is displayed grouped by the *Group by* option:

- *Period* Gives the values aggregated for the selected period, per period division, ex: Monthly gives aggregated values for every day of the month (aggregation can be a sum or an average depending on the report metric)
- *Points* Gives the values aggregated for the selected points, per point, ex: Monthly aggregated gives values for the month for every point (aggregation can be a sum or an average depending on the report metric)

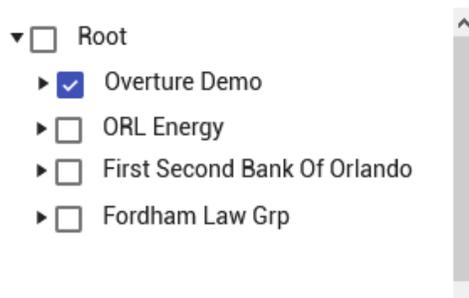
Selection of the Group by option



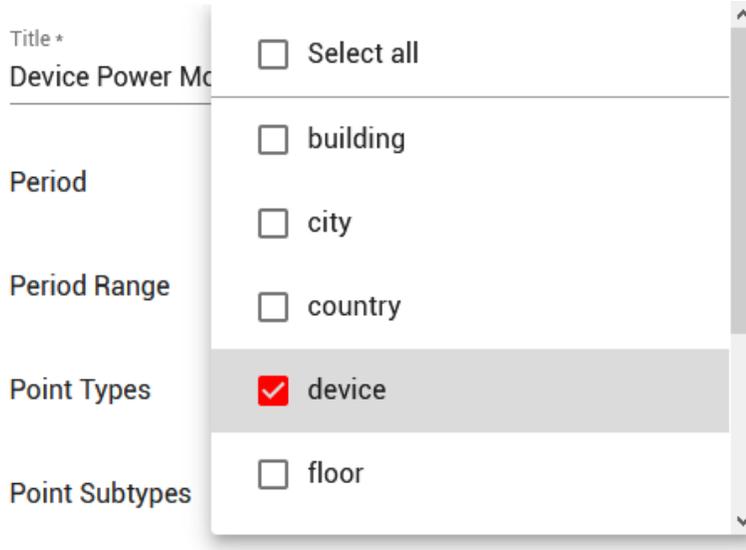
10.2.3.4 Point Filtering

By default a new report will select all points, across all tenants for this particular metric. Several filters are available to narrow down the points selection, so your report can be more specific.

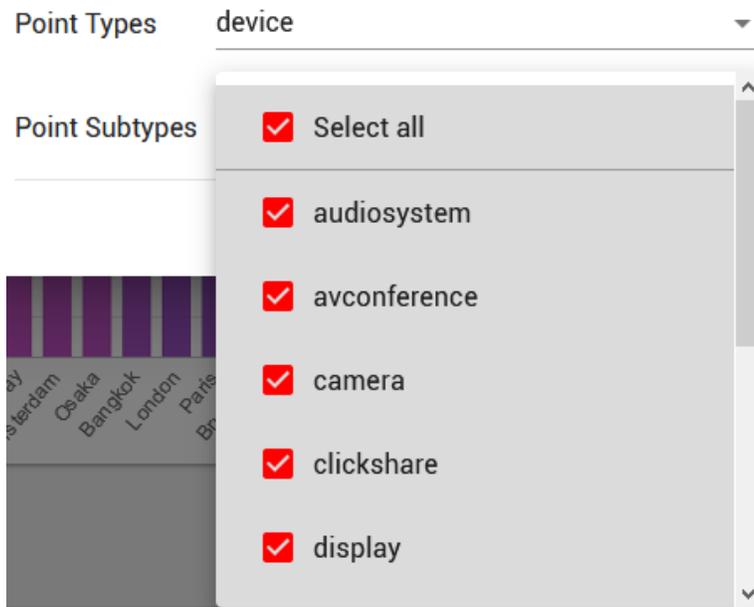
- *Tenant selection* The tree view on the right part of the widget setup allows for tenant selection



- *Point Types* Dropdown menu allows for selecting one or multiple point's types



- *Points Subtypes* Dropdown menu allows for selecting one or multiple point's subtypes



- *Ancestor(s) Name Contains* Field allows for filtering for all points for which on of the parent's human name (ancestor) or the point itself contains a text

Point Subtypes **All** ▼

Ancestor(s)
Name Contains

Note: The available options on the types are not interlinked nor linked to the type of metric or the tenant selection. This

allows for a selection that can dynamically adapt overtime. For example, if the type is *device* and subtypes are *display* and *projector* for a yearly period, but at the time of the report setup there is no projectors data, the report will only show *display* related values. However, if projectors are added in the tenant configuration and start producing data, then the report will automatically reflect these data across the period.

In this example below, the point selection is restricted to the Display devices belonging to the "Overture Demo" tenant.

10.2.4 Customizing dashboard

Like for Monitor pages, you can create additional Insights pages in your dashboard. To create additional pages, follow the steps described in [RMM Insights Access](#).

You can also rename, duplicate or delete pages by clicking on the ellipsis (...)

Each Insights page has global page parameters which apply to all widgets in that page.

Global parameters:

- **Active Hours** Defines the first and last hour of your active day so the report will discard data outside of these hours

Note: Important to note that the current hour value are interpreted as UTC Time, therefore if you are located in the North America east time zone and your active hours are 9am-5pm, you must select 13h00 - 21h00, (1pm-9:00pm)

You can either use the hour dropdown menu or the sliders to adjust start and end hours

11 APPENDIX A

11.1 Using Overture with Version 1 Control Servers (including Manager V6, Showmaster V2, and Overture CS1)

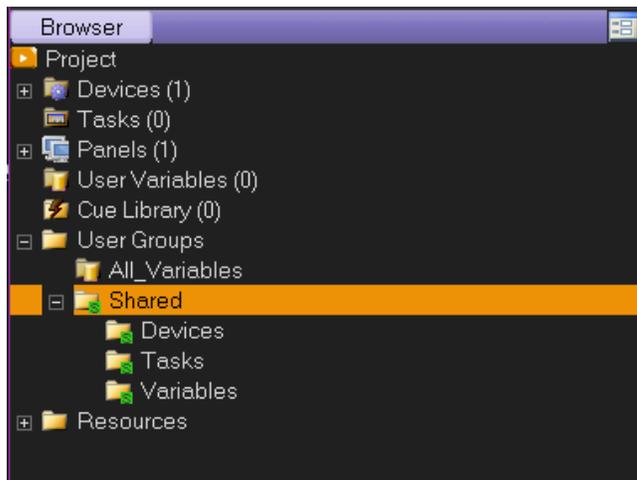
Overture can work directly with Version 1 Control Server products. However, setting up these products as your controller is different than working directly with Version 2 Control Server.

11.1.1 Control Server Setup

Version 1 Control Server (Medialon servers and Overture v1 server) products each have their own standalone interface for the creation and programming of devices, tasks, and user variables.

These products have an extensive knowledge base for programming your project that can be found in the "Medialon Control System Reference Manual" that comes with the products.

To work with Overture, these products all need a link to the UX Server. To create this link, you will need to create and share a user group. You can name the user group however you wish, the easiest is 'Shared'. Make sure the 'Shared' property is checked.



Inside that user group, you will need to create three separate groups: one for devices, one for tasks, and one for variables you wish to share with the UX Server.

Once those groups are created, drag devices, tasks, or variables you want the UX Server to interact with into its respective folder. It's important to know you must share the device variables into the variables subfolder in order for Overture to interact with them. Sharing just the device itself is not enough.

UX Server will need this shared group in order to populate your database and have its own link established.

11.1.2 Overture Setup

Once the project file is finished, you will need to populate the database with all the information you placed in the user group. The first thing you must do is tell Overture you are using a Version 1 Control Server product.

To do so, go into the 'Control Server' view of the Configurator. Change the existing default Control Server by selecting the "Type", and switching it to "CS1". After doing so, you will need to enter information related to the Control Server 1 product you are using:

- Address: The IP address or hostname of the PC the software is installed on, or the Showmaster.
- Port: The port used to talk to the CS1 product. The default port is 9255, but this is changeable in the preferences in the product.
- Shared Group: The name of the user group you created to share information. As explained earlier, "Shared" is

the easiest.

Optionally, you may enter some advanced settings if needed:

- Login User: The username needed to connect to the CS1 Product. This is not needed by default and set up in the CS1 product.
- Login Password: The password needed to connect to the CS1 Product. This is not needed by default and set up in the CS1 product.
- Proxy Address: The IP address or hostname of the PC where the special WS Proxy is if installed separately.
- Proxy Port: The port used to talk to the WS Proxy if installed separately.

This process is repeated for each CS1 type product you are using.

11.1.3 Ingesting The Project

Overture allows you to ingest your Control Server project, which automatically fills the database with points based on what you have programmed.

Overture UX's ingest engine reads the XML data of the project file, creating a point for each shared device, variable, and task. The information is not retrieved from the Control Server in real time; the ingest must be manually initiated using the CS project file.

Ingesting automatically adds as much metadata about the points as it can. For example, Devices are given Device type, Variables are given Variable type and specific sub-type of Integer, Real, Enum, etc. Device Variables are assigned the corresponding Device point as their parent. By default, points that are associated with the selected Server but do not exist in the ingested CS project file are erased during the ingest process. This helps to clean up the database when re-ingesting projects where devices, tasks, or variables have been deleted.

To ingest a project, select the Control Server that the project is running on from the drop-down.

Next, click the "Upload Overture CS Project..." button. Find your project file and click open. Then click the "Update Database" button to the right. The ingest engine uploads the project file, extracts the information from the Shared user group, and updates the database, adding or deleting points as needed.

If you need to make changes to the Control Server project, such as adding a variable, you can ingest the project again and the database update, add or delete points accordingly.

Note: If you do not want the ingest engine to automatically delete points associated with variables/devices/tasks that are no longer shared in the CS project, you can modify the default behavior in the 'UX Server Config' settings. Points without the type variable, device, or task will not be modified regardless of the settings.

Once the project is ingested, you can continue setting up UX Server as the User Manual explains. Make sure the CS1 product is in a run mode and Overture will interact with it completely.

Variables, tasks and commands need to be part of shared groups in order to be controllable by Overture

"Use description as human name" option will use the Description field as human name when importing points

Share objects to be able to ingest them

List "CS1" type Control Servers

12 APPENDIX B

12.1 Expert Mode Programming

This section is for advanced programmers who are looking for things to help them specialize their programming. We recommend to read this section when you are comfortable with Overture core concepts.

The following section will explain how make your own templates, or edit existing templates in Expert Mode. Overture panels and pages are written in HTML. More precisely, they are written in a slightly customized version of HTML that uses pieces of JavaScript to help control things. It is highly recommended you have a good foundation in those languages to proceed.

12.1.1 Tags

12.1.1.1 Intro

Tags are the HTML elements needed to create graphical elements on the page. While most standard HTML tags work as well, Overture has specialized tags meant to interact with the system. Please note that even though most standard HTML tags work, Overture is not a pure web-site designer. JavaScript libraries and extra frameworks cannot be added to the system.

12.1.1.2 Display

Displays the value of a variable.



Example

```
<med-display  
  med-label="Audio Volume"  
  med-point="Player1.AudioVolume"  
  med-suffix="dB"  
></med-display>
```

Attributes

- **med-label**: The text displayed to the left of the value. See [Labels](#).
- **med-point/med-repeat**: The point to be displayed. See [Points](#).
- **med-suffix**: The suffix to be appended to the point's value. Overrides a point's unit from the database if specified.
- **med-date-format**: The format a manager time variable should be displayed. See [Date Format](#) for options.
- **med-icon**: The icon displayed next to the label. See [Icons](#).

12.1.1.3 Point Table

Displays a list of point values. Useful when gathering all the data from a single device.

Power	On
Shutter	Closed
Lamp Hours	701 hrs

Example

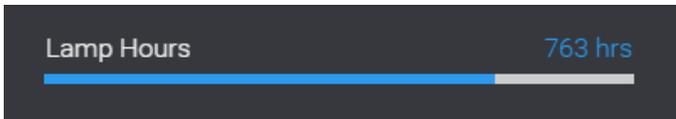
```
<med-point-table
  med-point="Projector1"
  med-subpoints=".Power,.Shutter,.HoursLamp1"
></med-point-table>
```

Attributes

- **med-point/med-repeat**: The parent of the point to be displayed. See [Points](#).
- **med-subpoints**: The name of the subpoints, separated by a comma, which, combined with the root name, lists the points you want to display. If nothing is specified it displays all children of the point.
- **med-date-format**: The format a time variable should be displayed. See [Date Format](#).

12.1.1.4 Progress Bar

Creates a gauge that monitors a variable. This is used for comparing the current value against a minimum or maximum value.



Example

```
<med-progress
  med-point="Room_2_Projector.LampHours"
  med-min="0"
  med-max="1000"
></med-progress>
```

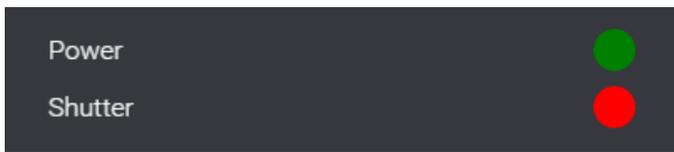
Attributes

- **med-label**: The text displayed to the left of the value. See [Labels](#).
- **med-point/med-repeat**: The point to be displayed. See [Points](#).
- **med-suffix**: The suffix to be appended to the point's value. Overrides a point's unit from the database if specified.
- **med-icon**: The icon displayed next to the label. See [Icons](#).
- **med-max**: The maximum value of the progress bar. Uses the **max** attribute of the point if not specified.
- **med-min**: The minimum value of the progress bar. Uses the **min** attribute of the point if not specified.

12.1.1.5 LED

Creates an LED that changes color based on the value provided to it. It's useful for monitoring enums or integers and providing a color index to their values.





Example

```
<med-led
  med-label="Power"
  med-point="Projector.Power"
  med-colors="red, #0000FF, #FFFF00, green"
></med-led>
<med-led
  med-label="Shutter"
  med-point="Projector.Shutter"
  med-colors="red, green"
></med-led>
```

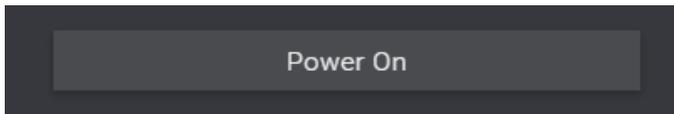
Note: In the above example, the Projector's power is an enum with a value of 3, and the shutter is an enum with a value of 0.

Attributes

- **med-label**: The text displayed to the left of the LED. Defaults to the name of the point. See [Labels](#).
- **med-point/med-repeat**: The point to be displayed. See [Points](#).
- **med-colors**: The comma separated list of colors relating to the value of the point being monitored. If the value is not a number, the color will default to the zero index (First item in the list). If the value of the point is greater than number of colors listed, it will take the last color specified in the list. (By default colors are red(0) and green(1))

12.1.1.6 Button

Creates a button that triggers an action when clicked (or tapped). It can also change its appearance depending on the value of a specified point.



Example

```
<med-button
  med-label="Power On"
  med-click="perform('PowerOn', 'starttask')"
  med-point="Projector.PowerStatus"
  med-on-state="1"
></med-button>
```

Attributes

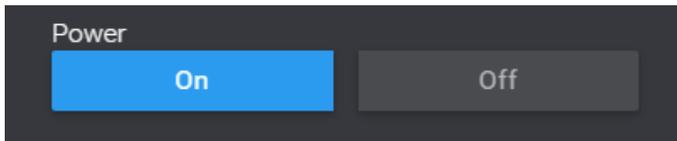
- **med-label**: The text displayed in the button. See [Labels](#).
- **med-click**: The action triggered when the button is clicked. See [Actions](#).
- **med-mouse-down**: The action triggered when the mouse button is pressed down. Example: `med-mouse-down="perform(point, 'Set Zoom', {Direction: 'In'})"`. See [Actions](#).

P 203 / 270

- **med-mouse-up**: The action triggered when the mouse button is released. Example: `med-mouse-up="perform(point, 'Set StopMove')"`. See [Actions](#).
- **med-click**: The action triggered when the button is clicked. See [Actions](#).
- **med-point/med-repeat**: The point to be monitored. See [Points](#).
- **med-on-state**: The value of the point that causes the button to appear highlighted.
- **med-icon**: The icon displayed in the button. See [Icons](#).

12.1.1.7 On/Off Button

Creates two buttons, inline with each other, monitoring the same point.



Example

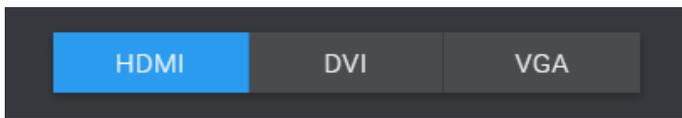
```
<med-on-off
med-point="Projector.PowerStatus"
med-label="Power"
med-label-on="On"
med-state-on="1"
med-click-on="perform('Projector','Power On')"
med-label-off="Off"
med-state-off="0"
med-click-off="perform('Projector','Power Off')"
></med-on-off>
```

Attributes

- **med-point/med-repeat**: The point to be monitored. See [Points](#).
- **med-label-on**: The text displayed in the left button. See [Labels](#).
- **med-state-on**: The value of the point that causes the left button to appear highlighted.
- **med-click-on**: The action triggered when the left button is clicked. See [Actions](#).
- **med-on-mouse-down**: The action triggered when the mouse button is pressed down on the left button. Example: `med-on-mouse-down="perform(point, 'Set Zoom', {Direction: 'Out'})"`. See [Actions](#).
- **med-on-mouse-up**: The action triggered when the mouse button is released on the left button. Example: `med-on-mouse-up="perform(point, 'Set Zoom', {Direction: 'Stop'})"`. See [Actions](#).
- **med-label-off**: The text displayed in the right button. See [Labels](#).
- **med-state-off**: The value of the point that causes the right button to appear highlighted.
- **med-click-off**: The action triggered when the right button is clicked. See [Actions](#).
- **med-off-mouse-down**: The action triggered when the mouse button is pressed down on the right button. Example: `med-off-mouse-down="perform(point, 'Set Zoom', {Direction: 'In'})"`. See [Actions](#).
- **med-off-mouse-up**: The action triggered when the mouse button is released on the right button. Example: `med-off-mouse-up="perform(point, 'Set Zoom', {Direction: 'Stop'})"`. See [Actions](#).
- **med-label**: The text displayed above both buttons. See [Labels](#).

12.1.1.8 Button Bar

A container that holds other buttons. It displays the contained buttons inline instead of vertically, one after the other.



Example

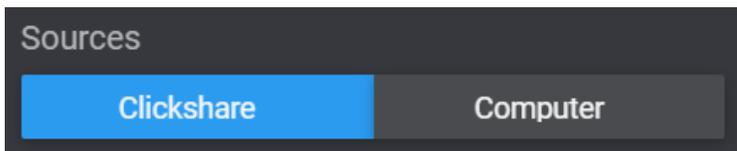
```
<med-button-bar>
  <med-button
    med-label="HDMI"
    med-point="Room_2_Projector.Input"
    med-click="perform('Room_2_Projector','Set Input',{ Input:'HDMI ' })"
    med-on-state="0"
  ></med-button>
  <med-button
    med-label="DVI"
    med-point="Room_2_Projector.Input"
    med-click="perform('Room_2_Projector','Set Input',{ Input:'DVI' })"
    med-on-state="1"
  ></med-button>
  <med-button
    med-label="VGA"
    med-point="Room_2_Projector.Input"
    med-click="perform('Room_2_Projector','Set Input',{ Input:'VGA' })"
    med-on-state="2"
  ></med-button>
</med-button-bar>
```

Attributes

There are no specific attributes for the button bar.

12.1.1.9 Enum Button Bar

This is meant to be used on a variable point of type `enum`. It will generate automatically one button per enum value.



Example

```
<med-enum-button-bar
  med-point="Sources"
></med-enum-button-bar>
```

Attributes

There are no specific attributes for the button bar.

12.1.1.10 Slider

Creates a slider that sends commands when the user moves it. The position of the slider marker is determined by the value of a point.



Example

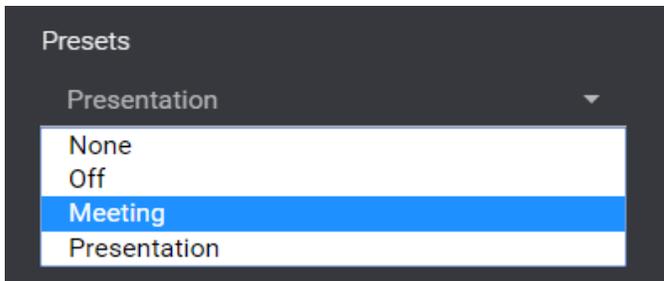
```
<med-slider
  med-label="Volume"
  med-point="Player1.AudioVolume"
  med-change="perform('Player1', 'Set Volume', {Volume: value('Player1.AudioVolume')})"
  med-max="0"
  med-min="-80"
  med-step="1"
  med-icon-up="+"
  med-icon-down="-"
></med-slider>
```

Attributes

- **med-label**: The text displayed with the slider. See [Labels](#).
- **med-point/med-repeat**: The point to be monitored. See [Points](#).
- **med-change**: The action triggered when the slider is moved. See [Actions](#).
- **med-max**: The maximum value of the slider. Uses the **max** attribute of the point if not specified.
- **med-min**: The minimum value of the slider. Uses the **min** attribute of the point if not specified.
- **med-step**: The amount moved 'up' and 'down' icons. Defaults to 1.
- **med-suffix**: The suffix to be appended to the point's value. Overrides a point's unit from the database if specified.
- **med-icon-up**: The icon displayed to the right of the slider. See [Icons](#)
- **med-icon-down**: The icon displayed to the left of the slider. See [Icons](#)

12.1.1.11 Select

Creates a multiple-choice selection box, which allows the user to select an item from a list. It is intended to be used with **enum** type variables.



Example

```
<med-select
  med-label="Presets"
  med-point="Room_2_DSP.Presets"
  med-change="perform('Room_2_DSP','Recall Preset',{Name: $string})"
></med-select>
```

P 206 / 270

ENABLING BRIGHT OUTCOMES

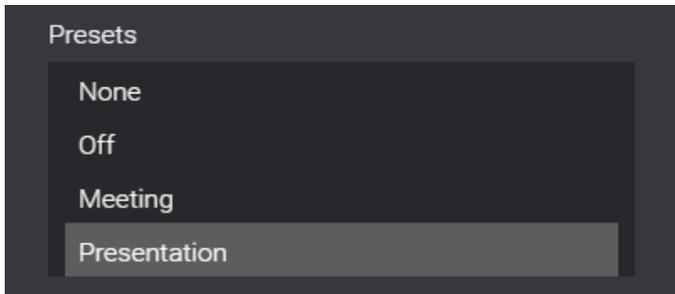


Attributes

- **med-label**: The text displayed above the selection. See [Labels](#).
- **med-point/med-repeat**: The point associated to the list. Intended to be sub-type `enum`. See [Points](#).
- **med-change**: The action triggered when an item is selected. See [Actions](#).
- **med-icon**: The icon displayed above the label. See [Icons](#).

12.1.1.12 List

Displays a list of items, and highlights the selected item in the list. It is intended to be used with 'enum' type variables.



Example 1

```
<med-list  
  med-label="Presets"  
  med-point="Room_1_VC.Presets"  
  med-change="perform('Room_1_DSP','Recall Preset',{Name: $string})"  
></med-list>
```

Attributes

- **med-label**: The text displayed above the list. See [Labels](#).
- **med-point/med-repeat**: The point associated to the list. Intended to be sub-type `enum`. See [Points](#).
- **med-change**: The action triggered when an item is selected. See [Actions](#).
- **med-count**: The number of list items displayed without needing to scroll. Defaults to 10.

12.1.1.13 Text Input

Creates an input box that changes a variable when you type into it. It is used when precision is needed. There are two modes for this tag.

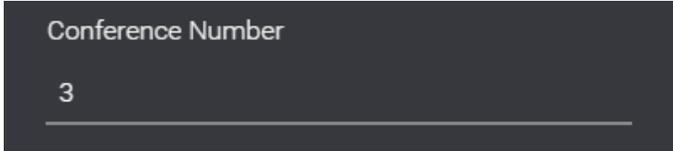
The default mode has a confirm button and clear button and will only perform the `med-change` when the confirm button is pressed. The second mode, eliminates the confirm and clear boxes and performs the `med-change` whenever the input box is changed(with each character entered).



Example 1

P 207 / 270

```
<med-text-input
  med-label="Number To Dial"
  med-point="Room_1_VC.NumberToDial"
  med-change="perform('Room_1_VC','Dial',{Number: $value})"
></med-text-input>
```



Example 2

```
<med-text-input
  med-label="Conference Number"
  med-point="Room_1_VC.Key"
  med-change="perform('Room_1_VC','Key Press',{Number: $value})"
  med-confirm="0"
></med-text-input>
```

Attributes

- **med-label**: The text displayed with the input. See [Labels](#).
- **med-change**: The action triggered when the input is modified. See [Actions](#).
- **med-point/med-repeat**: The point(s) to be monitored. See [Points](#).
- **med-confirm**: The mode in which the input is operating. Can be 1 or 0. Defaults to 1 (with Confirm/Clear buttons).

12.1.1.14 Checkbox

Creates a checkbox, allowing the toggling of values.



Example

```
<med-checkbox
  med-label="Auto Mode"
  med-point="AutoMode"
  med-change-on="perform('SetAutoModeOn','starttask')"
  med-change-off="perform('SetAutoModeOff','starttask')"
></med-checkbox>
```

Attributes

- **med-label**: The text displayed with the checkbox. See [Labels](#).
- **med-point/med-repeat**: The point to be monitored. See [Points](#).
- **med-change**: The action triggered when the box is checked or unchecked. See [Actions](#).
- **med-change-on**: The action triggered when the box is checked. See [Actions](#).

P 208 / 270

- `med-change-off`: The action triggered when the box is unchecked. See [Actions](#).

12.1.1.15 Toggle

Creates a switch style button, allowing the toggling of values.



Example

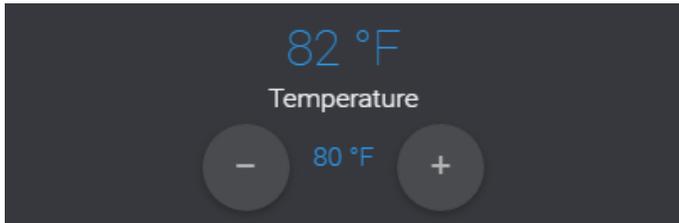
```
<med-toggle
  med-label="Auto Mode"
  med-point="Projector.Power"
  med-change-on="perform('Projector', 'Set Power', {Status: 'On'})"
  med-change-off="perform('Projector', 'Set Power', {Status: 'Off'})"
></med-toggle>
```

Attributes

- `med-label`: The text displayed with the button. See [Labels](#).
- `med-point/med-repeat`: The point to be monitored. See [Points](#).
- `med-change`: The action triggered when the button is switched on or off. See [Actions](#).
- `med-change-on`: The action triggered when the button is switched on. See [Actions](#).
- `med-change-off`: The action triggered when the button is switched off. See [Actions](#).

12.1.1.16 Set Up/Down

Creates a component composed of two variables to display and two arrows (up and down). It is useful for adjusting a target value while also monitoring the actual value of a variable.



Example

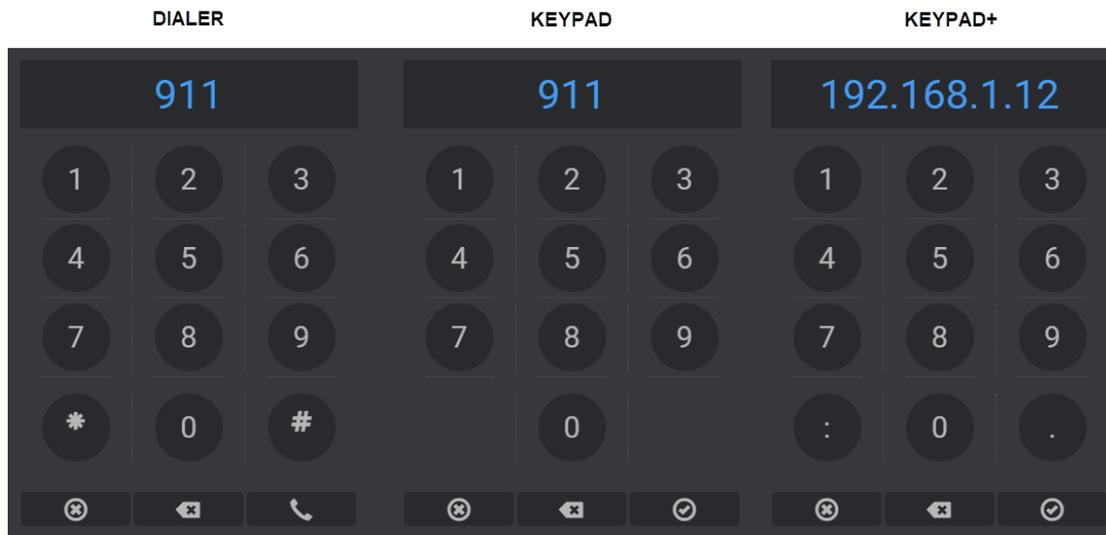
```
<med-up-down
  med-point=".CurrentTemperature"
  med-label="Temperature"
  med-target-point=".TargetTemperature"
  med-change-up="perform('HVAC', 'Set Temperature', { Mode: 'Up'})"
  med-change-down="perform('HVAC', 'Set Temperature', { Mode: 'Down'})"
></med-up-down>
```

Attributes

- **med-point**: The point to be monitored on the top display. See [Points](#).
- **med-label**: The label of the point on the top display. See [Labels](#).
- **med-target-point**: The point to be monitored in the bottom display. See [Points](#).
- **med-change-up**: The action triggered when the right (+) button is clicked. See [Actions](#).
- **med-up-mouse-down**: The action triggered when the mouse button is pressed down on the the right (+) button. Example: `med-up-mouse-down="perform(point, 'Set Focus', {Direction: 'Near'})"`. See [Actions](#).
- **med-up-mouse-up**: The action triggered when the mouse button is released on the the right (+) button. Example: `med-up-mouse-up="perform(point, 'Set Focus', {Direction: 'Stop'})"`. See [Actions](#).
- **med-change-down**: The action triggered when the left (-) button is clicked. See [Actions](#).
- **med-down-mouse-down**: The action triggered when the mouse button is pressed down on the the left (-) button. Example: `med-down-mouse-down="perform(point, 'Set Focus', {Direction: 'Far'})"`. See [Actions](#).
- **med-down-mouse-up**: The action triggered when the mouse button is released on the the left (-) button. Example: `med-down-mouse-up="perform(point, 'Set Focus', {Direction: 'Stop'})"`. See [Actions](#).
- **med-suffix**: The suffix to be appended to the point's value. Overrides a point's unit from the database if specified.
- **med-target-suffix**: The suffix to be appended to the target point's value. Overrides a point's unit from the database if specified.

12.1.1.17 Keypad

Creates a display of buttons for making a call or for entering number based information. It's useful for interfacing with a video conference system.



Example

```
<med-keypad
  med-point="Room_1_VC.NumberToDial"
  med-change="perform('Room_1_VC','Dial',{Number: $value})"
  med-type="dialer"
></med-keypad>
```

Attributes

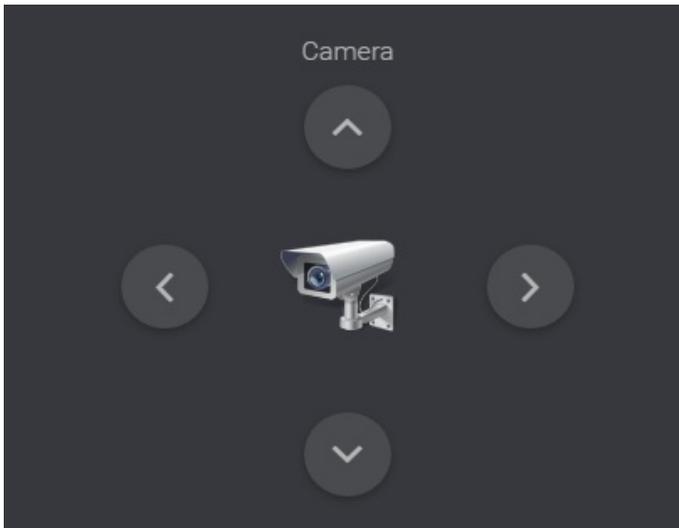
- **med-point/med-repeat**: The point to be monitored. See [Points](#).
- **med-change**: The action triggered when the 'Call' or 'Ok' buttons are pressed. See [Actions](#).
- **med-type**: The configuration of the buttons.

P 210 / 270

- The **dialer** type has * and # buttons, with a phone icon for validation. (Default option)
- The **keypad** type displays only buttons 0-9, with a **Clear** and **Ok** button.
- The **keypad+** type is like **keypad** but adds . and : buttons, useful for IP addresses.

12.1.1.18 Remote

Creates a set of buttons for controlling a camera or a TV. The buttons shown are customized based on if the commands are needed.



Example

```
<med-remote
  med-label="Camera"
  med-x-point=".Pan"
  med-y-point=".Tilt"
  med-z-point=".Zoom"
  med-up-click="perform(point, 'Set Tilt', {Position: ($value + 5)})"
  med-down-click="perform(point, 'Set Tilt', {Position: ($value - 5)})"
  med-left-click="perform(point, 'Set Pan', {Position: ($value + 5)})"
  med-right-click="perform(point, 'Set Pan', {Position: ($value - 5)})"
  med-button-a-click="perform(point, 'Set Zoom', {Position: ($value - 1000)})"
  med-button-b-click="perform(point, 'Set Zoom', {Position: ($value + 1000)})"
></med-remote>
```

Attributes

- **med-point**: Associates the remote to a point, giving it the same access rights as that point. See [Points](#).
- **med-x-point**: Associates the **med-left-click** and **med-right-click** to a point so that \$value can be used in the perform. Also associates the left and right arrows with the point's access rights.
- **med-y-point**: Associates the **med-up-click** and **med-down-click** to a point so that \$value can be used in the perform. Also associates the up and down arrows with the point's access rights.
- **med-z-point**: Associates the **med-button-a-click** and **med-button-b-click** to a point so that \$value can be used in the perform. Also associates the a and b buttons with the point's access rights.
- **med-img**: The image used in between the arrows. Example: "/assets/images/camera.png" If nothing is provided defaults to a security camera. If no image is desired, provide attribute with no value.
- **med-up-click**: The action triggered when the up arrow is clicked. See [Actions](#).
- **med-up-mouse-down**: The action triggered when the mouse button is pressed down on up arrow. Example: **med-**

- up-mouse-down="perform(point, 'Set Tilt', {Direction: 'Top'})". See [Actions](#).
- med-up-mouse-up: The action triggered when the mouse button is released on up arrow. Example: med-up-mouse-up="perform(point, 'Set Tilt', {Direction: 'Top'})". See [Actions](#).
- med-down-click: The action triggered when the down arrow is clicked. See [Actions](#).
- med-down-mouse-down: The action triggered when the mouse button is pressed down on down arrow. Example: med-down-mouse-down="perform(point, 'Set Tilt', {Direction: 'Down'})". See [Actions](#).
- med-down-mouse-up: The action triggered when the mouse button is released on down arrow. Example: med-down-mouse-up="perform(point, 'Set Tilt', {Direction: 'Stop'})". See [Actions](#).
- med-left-click: The action triggered when the left arrow is clicked. See [Actions](#).
- med-left-mouse-down: The action triggered when the mouse button is pressed down on left arrow. Example: med-left-mouse-down="perform(point, 'Set Pan', {Direction: 'Left'})". See [Actions](#).
- med-left-mouse-up: The action triggered when the mouse button is released on left arrow. Example: med-left-mouse-up="perform(point, 'Set StopMove')". See [Actions](#).
- med-right-click: The action triggered when the right arrow is clicked. See [Actions](#).
- med-right-mouse-down: The action triggered when the mouse button is pressed down on right arrow. Example: med-right-mouse-down="perform(point, 'Set Pan', {Direction: 'Right'})". See [Actions](#).
- med-right-mouse-up: The action triggered when the mouse button is released on right arrow. Example: med-right-mouse-up="perform(point, 'Set StopMove')". See [Actions](#).
- med-button-a-click: The action triggered when the bottom left button is clicked. See [Actions](#).
- med-button-a-mouse-down: The action triggered when the mouse button is pressed down on the bottom left button. Example: med-button-a-mouse-down="perform(point, 'Set Zoom', {Direction: 'Out'})". See [Actions](#).
- med-button-a-mouse-up: The action triggered when the mouse button is released on the bottom left button. Example: med-button-a-mouse-up="perform(point, 'Set Zoom', {Direction: 'Stop'})". See [Actions](#).
- med-button-b-click: The action triggered when the bottom right button is clicked. See [Actions](#).
- med-button-b-mouse-down: The action triggered when the mouse button is pressed down on the bottom right button. Example: med-button-b-mouse-down="perform(point, 'Set Zoom', {Direction: 'In'})". See [Actions](#).
- med-button-b-mouse-up: The action triggered when the mouse button is released on the bottom right button. Example: med-button-b-mouse-up="perform(point, 'Set StopMove')". See [Actions](#).
- med-button-a-icon: The icon for the button left button. Default is 'glyphicons-minus-sign'. See [Icons](#).
- med-button-b-icon: The icon for the button right button. Default is 'glyphicons-plus-sign'. See [Icons](#).

For the corresponding button to be shown, either {buttonname}-click, {buttonname}-mouse-down or {buttonname}-mouse-up must be provided. For example: the right button will be shown only if any of med-right-click, med-right-mouse-down or med-right-mouse-up are present (one or more).

12.1.1.19 Auto Widget

Creates a widget whose type depends upon the type of the specified point(s) and additional information provided in the point's metadata. For example: Auto Widget automatically creates a Slider widget if the point is a variable of type [integer](#).

How does med-auto-widget infer the type of widget?

The [med-auto-widget](#) widget uses the following rules in order to determine the automatic type of widget for a point:

- If the point is not a variable and the current application is Magic Menu: [med-auto-widget](#) creates a [med-page-link](#) widget which points to the point.
- Otherwise if the [widget](#) property is specified in the point's [metadata](#) key, [med-auto-widget](#) creates a widget of the specified type. Note that the [metadata.widget](#) key is usually automatically set by a device driver and is not meant to be modified by the user.
- Otherwise if the point is "read only" (that is the [readOnly](#) property of the point's metadata key is [true](#) OR there's no [perform](#) property in the point's metadata): [med-auto-widget](#) creates a [med-display](#) widget. Note that the [metadata.readOnly](#) or [metadata.perform`](#) keys are usually automatically set by a device driver and is not meant to be modified by the user.
- Otherwise [med-auto-widget](#) uses the following rules:
 - if the points subtype is [integer](#) or [real](#): [med-auto-widget](#) creates a [med-slider](#) widget.

- if the points subtype is `string`, `date` or `time`: `med-auto-widget` creates a `med-text-input` widget.
- if the points subtype is `enum`: `med-auto-widget` uses the following rules:
 - if the points have 2 or more enum values and its variable name is `".Power"`: `med-auto-widget` creates a `med-on-off` widget.
 - if the points have 4 or less enum values: `med-auto-widget` creates a `med-enum-button-bar` widget.
 - if the points have 5 or more enum values: `med-auto-widget` creates a `med-select` widget.
- there are also some special cases where no widget are created at all:
 - if the `hidden` property of the point's `metadata` is set to `true`.
 - if the point's variable name contains `.Activity`.

Example

The following example creates widgets for all of the children of the current point.

```
<med-auto-widget
  med-label="$humanname"
  med-repeat="{ parent: point }"
></med-auto-widget>
```

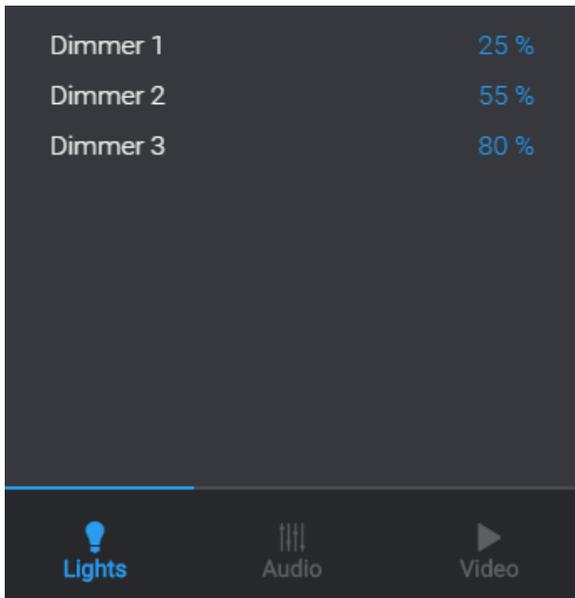
Attributes

- `med-label`: (optional) The label text for this widget. The value of this attribute is passed 'as is' as the `med-label` attributes of the automatically created widgets. See [Labels](#).
- `med-point/med-repeat`: The point associated to the list. See [Points](#).

12.1.1.20 Tabs

Tabs are special containers that show or hide content based on which container is selected. It consists of two components: Tab Control and Tab Content.

Tab Control creates the tab bar that has links to the containers. Tab Contents are the containers themselves.



Example

P 213 / 270

```

<med-tab-control></med-tab-control>

<med-tab-content med-tab-name="Lights" med-icon="ion-lightbulb">
  <med-point-table med-point="Room_2_Lighting"></med-point-table>
</med-tab-content>

<med-tab-content med-tab-name="Audio" med-icon="ion-levels">
  <med-point-table med-point="Room_2_Audio"></med-point-table>
</med-tab-content>

<med-tab-content med-tab-name="Video" med-icon="ion-play">
  <med-point-table med-point="Room_2_Player"></med-point-table>
</med-tab-content>

```

Tab Control Attributes

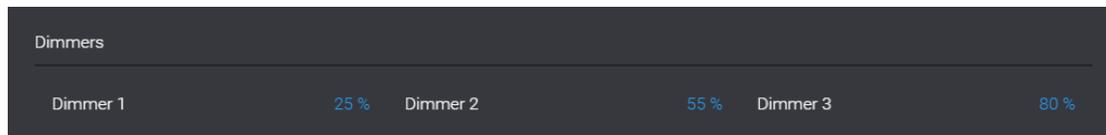
- **med-app**: This allows the tabs to only show up in a specific app. Options are 'magicmenu' or 'home'. Shows in both, if not specified.
- **med-position-small**: The position to place in tab control bar when the MagicMenu interface is small. Can be "top", "bottom", "left", "right". Default is "bottom".
- **med-position-medium**: The position to place in tab control bar when the MagicMenu interface is medium sized. Can be "top", "bottom", "left", "right". Default is "left".
- **med-position-large**: The position to place in tab control bar when the MagicMenu interface is large. Can be "top", "bottom", "left", "right". Default is "left".

Tab Content Attributes

- **med-point/med-repeat**: The point to associate the tab container to. See [Points](#).
- **med-tab-name**: The text displayed in tab control bar link. See [Labels](#).
- **med-icon**: The icon displayed next to the label. See [Icons](#).
- **med-auto-include**: If set to '1', the tab content will automatically be filled with the Control Panel link with the [med-point/med-repeat](#).

12.1.1.21 Container

Creates a container that groups items. The container can be hidden/shown or organized based on the attributes supplied to it.



Example

This example shows a group of displays that will organize the displays horizontally based on the size of the screen.

```

<med-container med-label="Dimmers">
  <med-display med-point="Room_2_Lighting.Dimmer1Value" med-label="Dimmer 1"></med-display>
  <med-display med-point="Room_2_Lighting.Dimmer2Value" med-label="Dimmer 2"></med-display>
  <med-display med-point="Room_2_Lighting.Dimmer3Value" med-label="Dimmer 3"></med-display>
</med-container>

```

Attributes

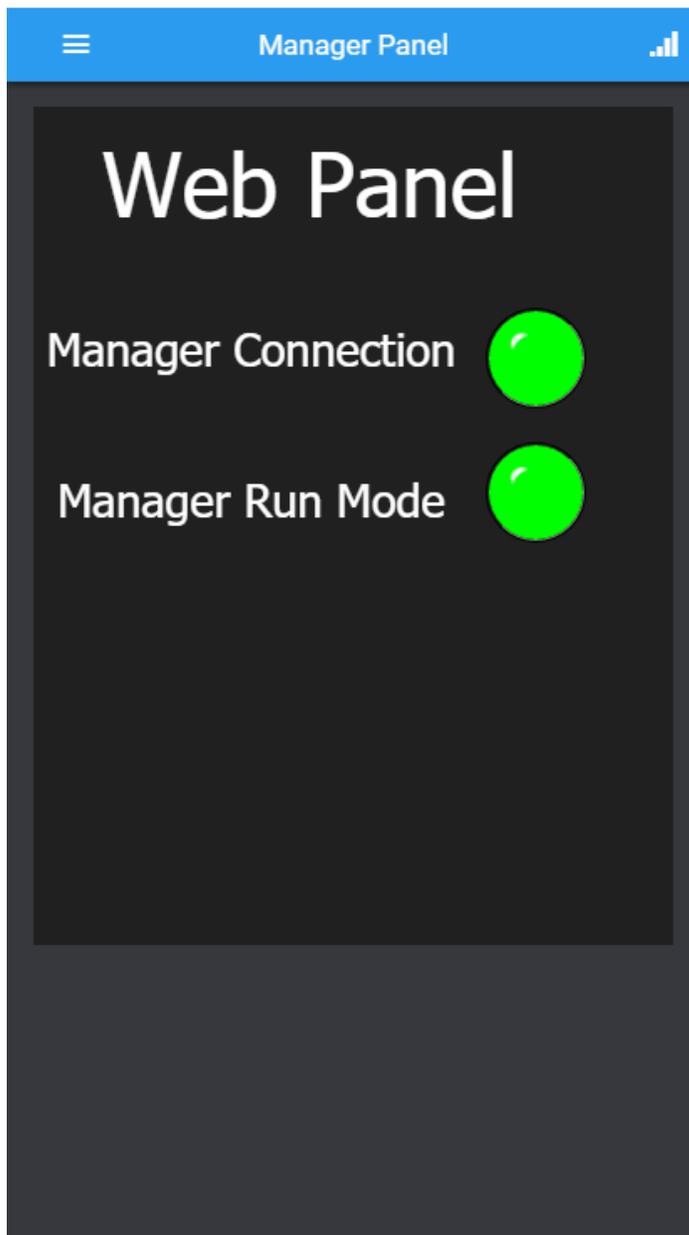
P 214 / 270

- `med-point/med-repeat` : The point to associate the container to. See [Points](#).
- `med-roles` : The specific roles that can see the contents of this container. Example: `med-roles="A/V,Building,IT"`. The name or the IDs of the roles may be specified.
- `med-app` : The app to show the contents of the container in. Options are 'magicmenu' or 'home'. Shows in both, if not specified.
- `med-label` : The text displayed above the container. See [Labels](#).
- `med-visible` : Whether the container is expanded or retracted when the page loads. Can be "true" or "false". If not specified, the container is not retractable.

Note: If `med-point` and `med-roles` are both used, the user will need to have access rights to both the role and the point for the container to be visible. If neither are used, the container will also be visible.

12.1.1.22 Frame

Creates a container that shows another website within the container. This is similar to HTML `iFrame` tag but allows dynamic changing of the source. It is useful when displaying camera feeds into your templates.



Example

```
<med-frame  
  med-src="http://localhost?panel=WebPanel_1"  
  med-height="400"  
  med-width="400"  
></med-frame>
```

Attributes

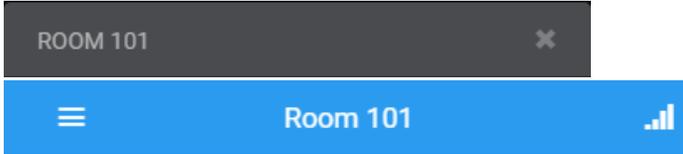
- **med-src**: The web page to display in the container. Example "http://192.168.1.10/?panel=WebPanel_1".
- **med-height**: The height of the container in pixels. Defaults to 300px.

P 216 / 270

- `med-width`: The width of the container in pixels. Defaults to the width of the control panel or magic menu page.
- `med-point/med-repeat`: The point to associate the container to. See [Points](#).

12.1.1.23 Header

Creates a name at the top of the Magic Menu page or Control Panel. If no header is specified, the panel defaults to the name of the point.



Example

```
<med-header
  med-title="Room 101"
></med-header>
```

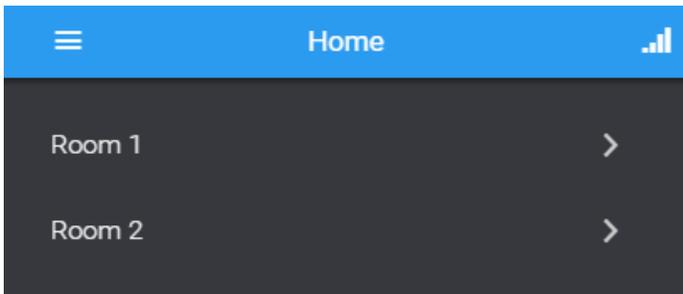
Attributes

- `med-title`: The text displayed at the top of the page.

12.1.1.24 Page Link

Magic Menu Only

Creates a button which displays a new page when clicked. This page must be an html document that exists in the views section of the assets folder.



Example

```
<med-page-link
  med-page="common/room"
  med-point="Room 1"
></med-page-link>
<med-page-link
  med-page="common/room"
  med-point="Room 2"
></med-page-link>
```

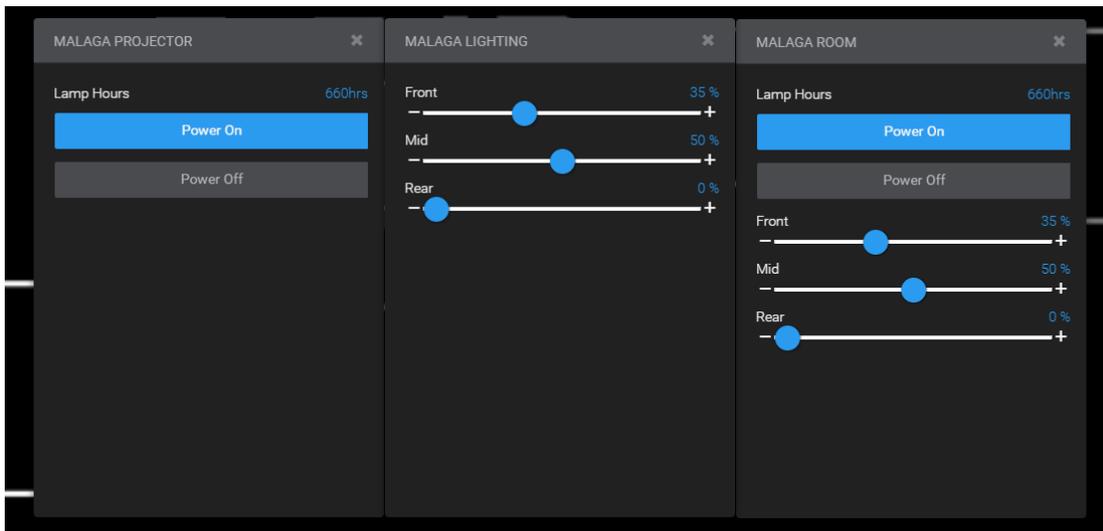
P 217 / 270

Attributes

- **med-label**: The text displayed in the button. See [Labels](#).
- **med-page**: The HTML document you wish to load when the link is clicked. Must be format of **FOLDER/NAME** and no file type after. If not specified, the page loads the **med-point/med-repeat's** HTML file associated to the 'Control Panel' section of the Configurator.
- **med-point/med-repeat**: The point associated with the page loaded. See [Points](#).
- **med-context**: Extra information you wish to pass into the new html page. Must be in form of an object `{item: value}`.
- **med-roles**: The specific roles that can see the page link button. Example: `med-roles="A/V,Building,IT"`. The name or the IDs of the roles may be specified.
- **med-icon** : The icon displayed in the button. See [Icons](#).

12.1.1.25 Include

The **med-include** tag loads another HTML file in the control panel. It is useful for creating templates by allowing small portions of code to be reused in many panels.



Example

```
<med-include
  med-point="Room_1_Projector"
  med-file="common/proj"
></med-include>
<med-include
  med-repeat="{parent: point, subtype: 'lighting'}"
  med-file="common/lx"
></med-include>
```

Attributes

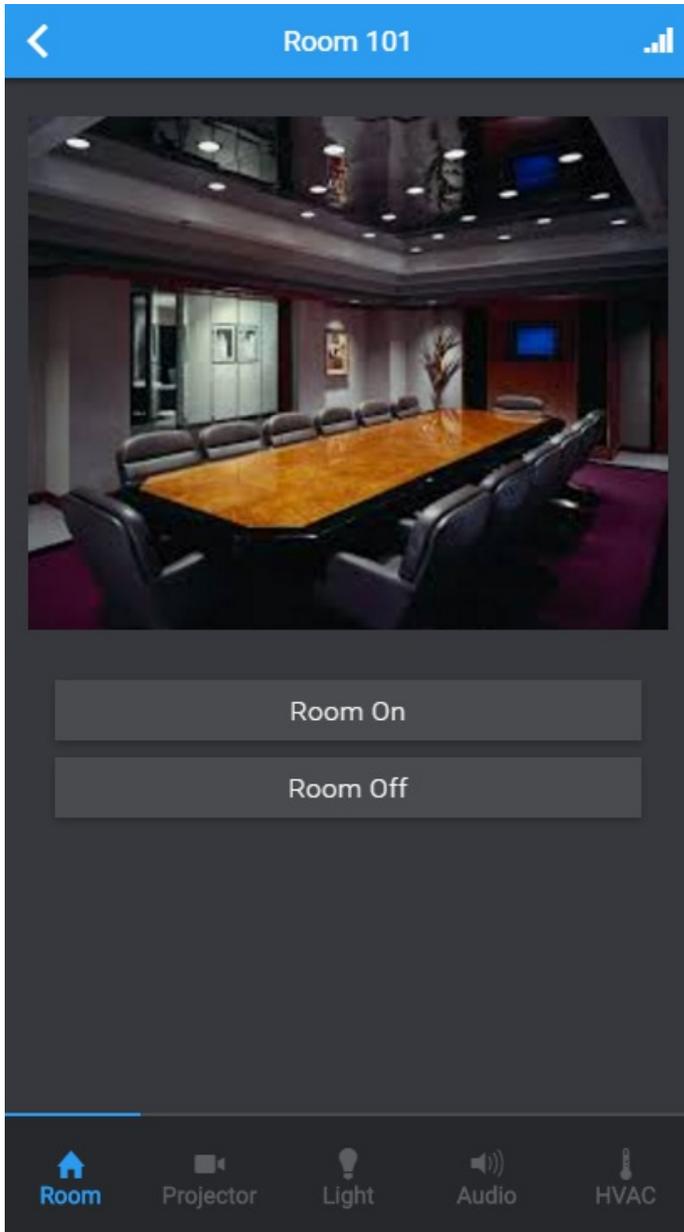
- **med-file**: The HTML document you wish to include. Must be format of **FOLDER/NAME** and no file type after. If not specified, the page includes the **med-point/med-repeat's** HTML file specified in the 'Control Panel' section of the Configurator.
- **med-point/med-repeat**:The point associated with the page included. See [Points](#).
- **med-context**: Extra information you wish to pass into the included page. Must be in form of an object `{item: value}`.

P 218 / 270

12.1.1.26 Footer

Magic Menu Only

Creates a container that is displayed at the bottom of the magic menu. This is used to create a bottom navigation bar, or always available content like a slider.



Example

```

<!--room.html-->
<med-tab-content med-tab-name="Room" med-icon="ion-home">
  
  <med-button
    med-point="roomStatus"
    med-label="Room On"
    med-click="perform(roomOn, 'starttask')"
    med-on-state="1"
  ></med-button>
  <med-button
    med-point="roomStatus"
    med-label="Room Off"
    med-click="perform(roomOff, 'starttask')"
    med-on-state="0"
  ></med-button>
</med-tab-content>

<med-tab-content
  med-repeat="{parent: point, type: 'device'}"
  med-auto-include="1"
  med-icon="ion-levels"
></med-tab-content>

<med-footer med-file="pages/footer"></med-footer>

<!--footer.html-->
<med-tab-control ></med-tab-control>
<!-->

```

Attributes

- **med-file**: The HTML document you wish to include. Must be format of **FOLDER/NAME** and no file type after.
- **med-height**: The height of the footer in pixels. If not specified, the footer has the default height of 44px.

12.1.2 Attributes

12.1.2.1 Intro

Some options exist for multiple tags, and have specialized ways they can be used.

12.1.2.2 Points

Points can be selected and add context to items.

med-point

Used when associating a tag with a single point. The point can specified in the following ways:

- **Full Name**: This is the full variable name, as shown in [Configurator](#), e.g.: `Malaga_Projector.Status`, e.g.:

```
<med-display med-point="Malaga_Projector.Status"></med-display>
```

- **Device Variable**: The partial/contextual variable name of the device. This only works if the the tag is in the context of a point (usually the page/control panel), e.g.:

```
<med-display med-point=".Status"></med-display>
```

If this `med-display` tag is in the context of, for example, a point called `Malaga_Projector`, either set by the page/control panel or some container `Overture` will try to resolve the point in the following ways:

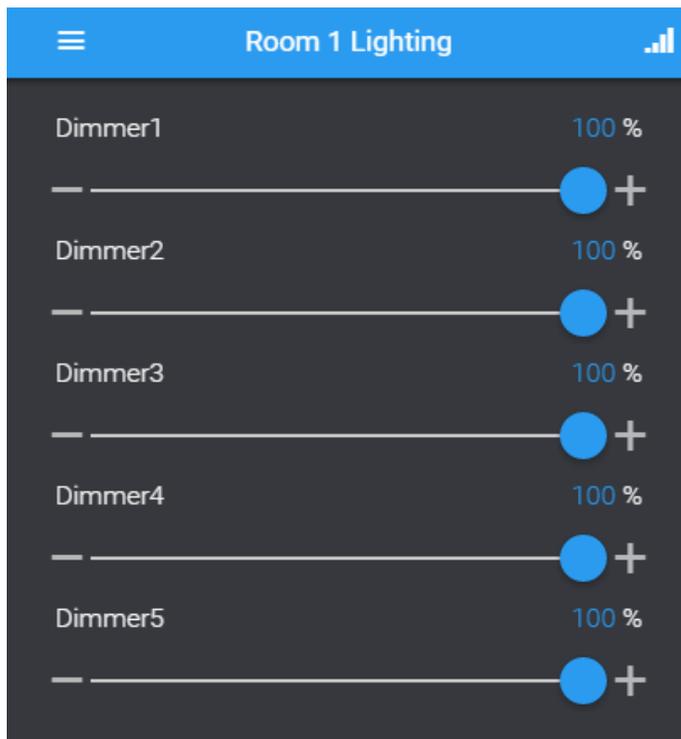
1) Look for a point that with a `variable name` that contains the literal concatenation of the context point and the `device variable` name, which in this case would be `Malaga_Projector` and `.Status`, i.e., `Malaga_Projector.Status`. This expression thus resolves to the same point as in the previous example.

2) If the previous method fails, `med-point` will look for a point that is a *direct child* of the context, `Malaga_Projector`, whose variable name simply *ends with* the expression after the dot (`.`). In this case, passing in `.Status` will look for a child point with a `variable name` that ends in `Status`, which would still match `Malaga_Projector.Status`, but would not match, for example, `Malaga_Projector.Statuses`. "Deep paths" like `Malaga_Projector.Lighting.Status` are simply ignored and won't be resolved by this method.

- **Keywords:**
 - `point`: refers to the current context point, which by default refers to the point of the page (`Magic Menu`) or control panel (`Home`), unless the associated tag is inside a container (e.g. `med-container`, `med-tab-content`) which itself has a point context, in which case the context of the container is used
 - `point.parent` refers to the parent of the current context point (the same one referred to by the `point` keyword)

med-repeat

Tags can be also be associated with points using the `med-repeat` attribute. Unlike `med-point` however, `med-repeat` requires a "filter" to be specified. This filter generates a list of points, containing *any and every point* that matches the criteria specified in the filter properties. For each of these points, the tag is "repeated", with each "copy" being associated with each one of the points generated by the filter. The filter passed into `med-repeat` must be in "object" format.



Room 1 Lighting Dimmer1V...	Dimmer 1		Room_1_Lighting Dimmer1...
Room 1 Lighting Dimmer2V...	Dimmer 2		Room_1_Lighting Dimmer2...
Room 1 Lighting Dimmer3V...	Dimmer 3		Room_1_Lighting Dimmer3...
Room 1 Lighting Dimmer4V...	Dimmer 4		Room_1_Lighting Dimmer4...
Room 1 Lighting Dimmer5V...	Dimmer 5		Room_1_Lighting Dimmer5...

Consider the example of a context point representing a projector, [Example_Projector](#), which has 3 child points of type [variable](#), each representing its [Power](#), [Brightness](#) and [Contrast](#) settings. If you wanted to create 3 [med-slider](#) widgets, each associated to one of these variables, you could write:

```
<med-slider med-point="Example_Projector.Power"></med-slider>
<med-slider med-point="Example_Projector.Brightness"></med-slider>
<med-slider med-point="Example_Projector.Contrast"></med-slider>
```

Assuming the projector only has these 3 variables, with [med-repeat](#), you can shorten the above to the following, to get the same result:

```
<med-slider
  med-repeat="{parent: point, type: 'variable'}"
></med-slider>
```

[Overture](#) currently supports the following filter properties:

- **name**: Select every point whose [human name](#) contains the passed-in string
- **shortname**: Select every point whose [short name](#) contains the passed-in string
- **variablename**: Select every point whose [variable name](#) contains the passed-in string
- **roles**: Select every points which has *all* of the specified roles. Must be specified as an array, eg: `roles: ['Building', 'Energy']`
- **tags**: Select every point which has *all* of the specified roles. Must be specified as an array, eg: `tags: ['Lighting', 'Audio', 'Display']`
- **type**: Select every point whose [type](#) matches the passed-in string
- **subtype**: Select every point whose [subtype](#) matches the passed-in string
- **parent**: Select every point that has the passed-in point as a parent, i.e., every point that is a child of the passed-in point. This point can be specified the same way as a [med-point](#), e.g., as a keyword (`{ parent: point }`) or a substring of the human name (`{ parent: 'Example Projector' }`)
- **depth**: Used with the [parent](#) filter property, this number specifies the *maximum* number of levels in the hierarchy of points (children, grand-children, etc.) to search. By default (i.e. when not specified), [depth](#) is set to **1**, meaning the results must be *direct children* of the point only; setting [depth: 2](#) specifies that result points must either be direct children *or* grand-children of the [parent](#); setting [depth: 0](#) specifies no limit
- **maxitems**: This number specifies the *maximum* number of points to be returned. By default [maxitems](#) is set to **50**. Maximum is **500**
- **server**: Select every point that is associated with the control server who name matches the passed-in string, e.g. `{ server: control_server_1 }`

N.B. When these filter properties are used together (i.e. in the same [med-repeat](#)), this specifies that the resulting points must satisfy *all* of the filter conditions. For example:

```
med-repeat="{parent: 'Meeting Room 1', type: 'device', tags: ['Lighting, HVAC']}"
```

This filter would specify a list of points that:

- are direct children of a point with the name 'Meeting Room 1' AND

- are of type 'device' AND
- have have *both* of the tags listed, i.e., 'Lighting' and 'HVAC'

12.1.2.3 Labels

Labels provide text to items on the page.

You can provide text, or use a label context variable to provide dynamic information. The following variables work:

- `$humanname`: The human name of the point associated to the item.
- `$shortname`: The short name of the point associated to the item.
- `$parent.name`: The human name of the parent of the point associated to the item.
- `$parent.shortname`: The short name of the parent of the point associated to the item.

Labels will default when no information is provided. The system will go down the following hierarchy, using the associated point, until it finds something:

Label -> Point's Short Name -> Point's Human Name -> Point's Variable Name -> Blank

12.1.2.4 Actions

Actions are how interactive items interact with the system. The action is called when the item is interacted with (For example, clicking a button or moving a slider). They are JavaScript functions that normally interact with the Control Server. The following actions can be used:

- `setVariable()`: This is used to change the value of a variable type point. It's the default action when none is provided.
- `perform()`: This is used to interact with tasks or devices directly. It performs a device command directly, or starts/stops a task on the Control Server.
- `openUrl()`: This is used for buttons that need to open other web pages.

Set Variable

Used to set a variable to a desired value. When using a Control Server Version 2 or above, this action will generally be enough to tell the driver which device command to perform.

`setVariable(point, value)` has two parameters:

- *point*: The name of the point you want to change the value of.
- *value*: The value to change the point to.

Example:

```
setVariable('Projector.Brightness', 50)
```

Perform

It can be set up two ways. The first way interacting with a task.

`perform(task, command)` has two parameters:

- *task*: The name of the task to interact with.
- *command*: The command to perform on the task. Must be either 'starttask' or 'stoptask'.

Example:

```
perform('Room_On', 'starttask')
```

It can also be used to directly perform a device command.

`perform(device, command,[parameters])` has two required parameters and one optional parameter:

- *device*: The name of the device to interact with.
- *command*: The command to perform on the device. The command name comes from the driver documentation.
- *parameters*: The command parameters if needed. Must be in the form of an object (`{name1:value1,`



name2:value2, etc.}). This information is in the driver documentation.

Example:

```
perform('Projector', 'Set Power', {'Status': 'On'})
```

Open Url

Used to open a web page in a new tab on the browser.

`openUrl(url)` has one parameter:

- `url`: The URL to open in a new tab.

Example:

```
openUrl("http://www.barco.com")
```

Action Context Variables

Actions have context variables that can be used inside of their functions to help build more dynamic pages.

point

`point` is used to describe the point of the current page. It's useful for substituting the device parameter in a `perform()`.

Example: `perform(point, 'Set Power', {'Status': 'On'})`.

\$point

`$point` is used to describe the point associated to the item. See [Points](#).

Examples: `perform($point, 'starttask')`. `setVariable($point, 50)`.

\$value

`$value` is used to describe the value of the point associated to the item. For example, when you move the slider to 50. `$value` becomes 50 before it is passed into the function.

Examples: `perform(point, 'Set Level', {'Level': $value})`. `setVariable($point, $value)` (This is called when no action is provided).

\$string

`$string` is used to describe the string value of an enum variable point associated to the item. For example if an enum variable point has the enums of "Off" and "On" in that order, the `$value` is the index (0 or 1) and the `$string` will be "On" or "Off". It's needed for most device commands that interact with enums.

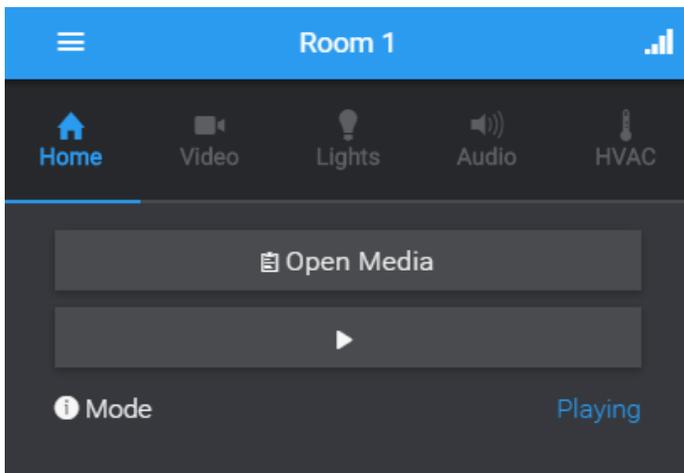
Example: `perform(point, 'Select Source', {'Name': $string})`.

\$parent

`$parent` is used to describe the parent of the point associated to the item.

Example: `perform($parent, 'Play')`.

12.1.2.5 Icons



Some tags have a `med-icon` attribute. This allows you to add or change icons on tags within the system.

To add the icon, in the attribute specify the full name of the icon: `med-icon="ion-home"`. There are two sets of icons:
 Ionicons: <http://ionicons.com/>
 Glyphicons: <http://glyphicons.com/>

Please use the Icon Browser or individual websites to see all available icons.

Example

```
<med-tab-content med-tab-name="Home" med-icon="ion-home">
  <med-button
    med-label="Open Media"
    med-icon="ion-clipboard"
    med-click="perform(point, 'Open Media')"
  ></med-button>
  <med-button
    med-icon="ion-play"
    med-click="perform(point, 'Play')"
  ></med-button>
  <med-display
    med-point=".Mode"
    med-label="Mode"
    med-icon="ion-information-circled"
  ></med-display>
</med-tab-content>
```

Icons will default when no information is provided. The system will go down the following hierarchy, using the associated point, until it finds something:

Icon -> Point's Icon -> Point's Sub-Type Icon -> Point's Type Icon* -> Blank**

12.1.2.6 Date

Some tags will have a `med-date-format` attribute. This allows the incoming value to be formatted in more user friendly way. For example, time variables may come in with values like: "09:00:00/00", whereas you might just want to tell the user "09:00 AM". The following options can apply:

- `medium`: equivalent to 'MMM d, y h:mm:ss a' for en_US locale (e.g. Sep 3, 2010 12:05:08 PM)
- `short`: equivalent to 'M/d/yy h:mm a' for en_US locale (e.g. 9/3/10 12:05 PM)
- `fullDate`: equivalent to 'EEEE, MMMM d, y' for en_US locale (e.g. Friday, September 3, 2010)
- `longDate`: equivalent to 'MMMM d, y' for en_US locale (e.g. September 3, 2010)

- **mediumDate**: equivalent to 'MMM d, y' for en_US locale (e.g. Sep 3, 2010)
- **shortDate**: equivalent to 'M/d/yy' for en_US locale (e.g. 9/3/10)
- **mediumTime**: equivalent to 'h:mm:ss a' for en_US locale (e.g. 12:05:08 PM)
- **shortTime**: equivalent to 'h:mm a' for en_US locale (e.g. 12:05 PM)

Example

```
<med-display
  med-label="Current Time"
  med-point="Overture.CurrentTime"
  med-date-format="shortTime"
></med-display>
```

If **med-date-format** is not defined, the system will try to get its value from the point unit name. The format string will be extracted from expression within square brackets in the unit name, if it's available.

You can also use any custom combinations of format strings available for angular 'date' filter. The list of format strings is available at: <https://docs.angularjs.org/api/ng/filter/date>

12.1.3 AngularJS Items

12.1.3.1 Intro

The framework for providing HTML panels is AngularJS. As a result, some built in AngularJS attributes for tags are available for your use in creating panels.

12.1.3.2 ng-init

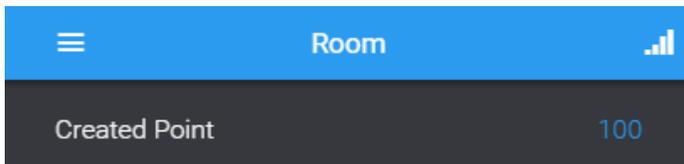
The **ng-init** is used to add items to the scope. This can be finding points from the database, or adding virtual variables that you may reference.

Example

```
<div ng-init="
  vol = findPoint({parent: point, variableName: '_Audio_Volume'});
"></div>
<med-display
  med-point="vol"
></med-display>
```

Direct Assignment

ng-init can be used to assign values directly or create virtual objects.



```

<div style ng-init="
  myPoint = {name : 'Created Point', value: 100};
"></div>
<med-display
  med-point="myPoint"
></med-display>

```

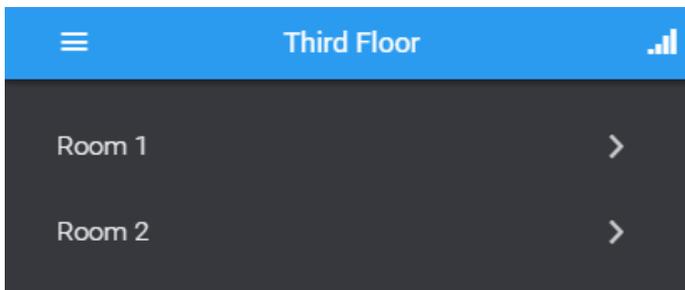
This allows you to store some information that isn't tied to a control server variable and use it while the panel is open. The value resets once the panel is closed, as this type of point is only used while the control panel is open. This behaves like a normal point, which means its value can be dynamically affected by `med-change` or `med-click` attributes.

Functions

`ng-init` also has a set of defined functions that can be used.

setPoint()

`setPoint()` is used in Magic Menu and sets the current page to refer to whatever point you specify. This is useful in giving `home.html` some type of beginning context.



```

<div ng-init="setPoint('Room 1'); "></div>
<med-page-link
  med-repeat="{parent: point, type: 'device'}"
></med-page-link>

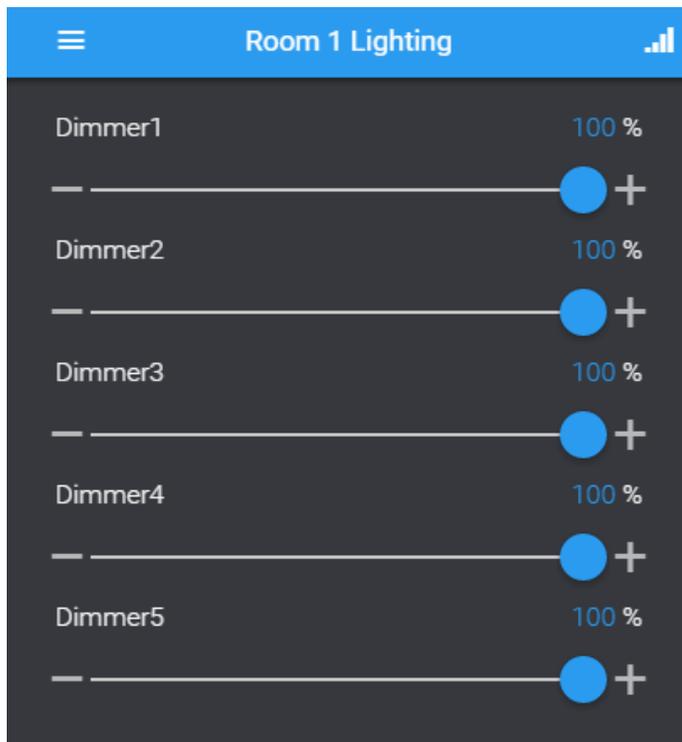
```

findPoint / findPoints

The `findPoint()/findPoints()` functions are used to find a group or groups of points and store them into a variable used by the page. The `findPoint()` function only returns the first point object that matches the filter applied. The `findPoints()` function returns an array of points, that is used with `ng-repeat` to access each point in the array. These functions should be set up with the following format:

`ng-init="result= findPoint(filter, subscribe);"` or `ng-init="result = findPoints(filter, subscribe);"`

- `result` is the name of the item added to the scope. For example, `myPoints = findPoints(filter)`.
- `filter` should be passed as an object with any filter parameters in it. For example `{parent: point, variablename: '_Audio'}`
- `subscribe` is an optional boolean. If set to false, the point/s value will not be updated automatically unless added to a tag. If true or left out, the point/s value will be updated automatically.



```

<div
  ng-init="dimmers = findPoints({parent: point, variablename: 'Dimmer'});
"></div>
<med-slider
  ng-repeat="dim in dimmers"
  med-point="dim"
></med-slider>

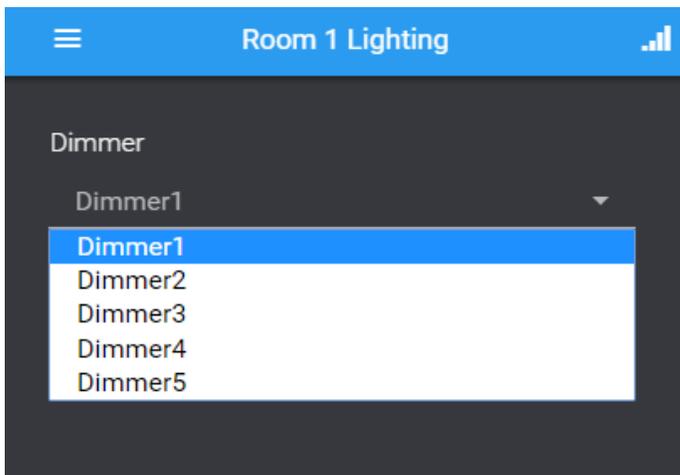
```

The following items can be used as filtering objects for both `findPoint` and `findPoints`.

- `parent` : the parent of the point you want to find.
- `depth` : how far down the hierarchy the point you want to find is. `depth: 1` , would be a direct children.
- `name` : the string contained in the human name of the point you want to find.
- `shortname` : the string contained in the short name of the point you want to find.
- `variablename` : the string contained in the variable name of the point you want to find.
- `type` : the type of the point you want to find. Refer to the types view in Configurator for available names.
- `subtype` : the sub type of the point you want to find. Refer to the types view in Configurator for available names.
- `tags` : the tags associated to the point you wish to find. Must be in an array Example: `[tag1, tag2]`.
- `roles`: the roles associated to the point you wish to find. Must be in an array Example: `[role1, role2]`.
- `'server'`: the control server id or name the point is associated with.

`createEnum()`

`createEnum` can be used to take an array of items, and then create a virtual enum with it. This allows that array to be put into a `med-select` without needing to manipulate the `enumStrings` property.



```

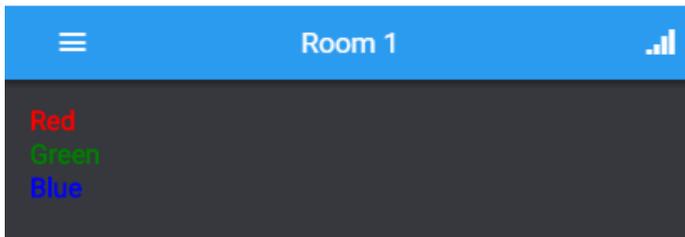
<div
  ng-init="dimmers = findPoints({parent: point, variablename: 'Dimmer'});
  dimmerEnum = createEnum(dimmers, 'short_name');
"></div>
<med-select
  med-label="Dimmer"
  med-point="dimmerEnum"
></med-select>

```

`createEnum()` has two items that it can take in. First is the array you wish to create the enum from. The second item is optional. It is the property from the points you wish to create the enumStrings from. If nothing is specified, the name property is used.

12.1.3.3 ng-repeat

The `ng-repeat` attribute can be used to iterate through an array. It will create elements equal to the length of the array. It allows for each element to have an item from the array passed into it.



Example

```

<div style ng-init="
  myColors = ['Red', 'Green', 'Blue']
"></div>

<div ng-repeat = "color in myColors"
  style="color: {{color}};">
  {{color}}
</div>

```

Attributes

- **ng-repeat**: must be specified in 'item in arrayName' format. The 'item' object represents the object in the array. The 'arrayName' object is the array you wish to iterate through.



13 APPENDIX C

13.1 Tasks / User Variable Programming

This section is for advanced programmers who are looking for things to help them specialize their programming. We recommend to read this section when you are comfortable with Overture core concepts.>

13.2 Tasks

A task is a sequence of commands which are executed by a Control Server. It is a way to automate actions in Overture.

Both Control Server 1 and Control Server 2 can execute tasks but they do it a different way.

13.2.1 Control Server Type 1

Tasks managed by Control Servers of type 1 (a.k.a CS1) cannot be edited via Overture Configurator. They are programmed using the Control Server 1 programming environment. These tasks are added to the database when a CS1 project is ingested in Overture Configurator.

13.2.2 Control Server Type 2

On the other hand, Overture Configurator allows creating, modifying and deleting tasks managed by a Control Server of type 2 (a.k.a CS2).

13.2.2.1 Creating a CS2 task

- click the "Add" button
- select type "Task"
- select the Control Server which will execute the task
- click on the "Edit Metadata" button
- enter the task content (see: [Task General Format](#))
- click OK in order to close the Metadata dialog box
- click the "Save" button to save the task

13.2.2.2 Modifying a CS2 task

- click on the "Edit Metadata" button
- enter the task content (see: [Task General Format](#))
- click OK in order to close the Metadata dialog box
- click the "Save" button to save the task

13.2.2.3 Deleting a CS2 task

- select the task point in the Points View
- click the "Delete" button

13.2.3 Task General Format

A task is defined by a JSON formatted text. The JSON text describes:



- the name of the task
- an array of 'cues'.

Example of a simple task:

```
{
  "name": "Malaga_On",
  "cues": [
    {
      "command": "exp",
      "exp": "perform('Malaga_DSP','Recall Preset',{Name:'Meeting'})"
    },
    {
      "command": "exp",
      "exp": "perform('Malaga_Lighting','Recall Preset',{Name:'Meeting'})"
    }
  ]
}
```

13.2.4 Referencing variable points in a task expression

Expressions can refer variable points managed by the same control server by using the `dom` namespace and the `variable_name` of the point. For example: `dom.Malaga_HVAC.CurrentTemperature`. Expressions can then use the `.value` or `.required` properties of the variable.

- `.value`: represents the actual (monitored) value of the variable
- `.required`: represents the required value of the variable

Examples:

This test whether the current temperature of the Malaga Room is over 80.

```
if (dom.Malaga_HVAC.CurrentTemperature.value > 80)
```

This sets the required value of the target temperature of the Malaga Room

```
dom.Malaga_HVAC.TargetTemperature.required = 75
```

13.2.5 Cue syntax

Cue can be of 4 different types:

- `exp`: simple JavaScript expressions
- `if/else/end`: if/then/else control flow
- `while/endwhile`: while/endwhile control flow
- `wait`: pauses the task for the specified time

The type of the cue is specified using the `command` property of the cue.

13.2.5.1 Expression cues

Expression cues are simple Javascript expressions. They can be used to set variable values or to send commands to devices.

Setting a setting a variable required value:



```
dom.Malaga_HVAC.TargetTemperature.required = 75
```

Sending a command to a device using the `perform()` function:

```
perform('Malaga_DSP','Recall Preset',{Name:'Meeting'})
```

The perform command format is exactly the same as the one used in the Control Panels templates.

Example of a task which recalls a 'Meeting' preset in both Malaga Room and Sevilla Room:

```
{
  "name": "example",
  "cues": [
    {
      "command": "exp", "exp": "perform('Malaga_DSP','Recall Preset',{Name:'Meeting'})"
    },
    {
      "command": "exp", "exp": "perform('Sevilla_DSP','Recall Preset',{Name:'Meeting'})"
    }
  ]
}
```

13.2.5.2 If/Then/Else cues

If/Then/Else cues allow to test a condition and to take decisions depending on the condition.

The overall format is:

```
if test_expression
do something if the expression is true
else
do something else if the condition is false
endif
```

If/Then/Else cues are written by the specifying 3 different types of `command`:

- `if`: this cue tests the condition expressed in the `exp` property. This cue starts the block of actions which must be executed if the condition is true.
- `else`: (optional) this cue starts the block of cues which must be executed when the condition is false
- `endif`: this cues ends the block opened by the `if` cue.

Note that If/Then/Else cues can be nested.

Example:

```

{
  "name": "If_Example",
  "cues": [
    {
      "command": "if", "exp": "dom.Malaga_HVAC.CurrentTemperature.value > 80"
    },
    {
      "command": "exp", "exp": "perform('Malaga_Lighting','Recall Preset',{Name:'Meeting'})"
    },
    {
      "command": "else"
    },
    {
      "command": "exp", "exp": "perform('Malaga_Lighting','Recall Preset',{Name:'Night'})"
    },
    {
      "command": "endif"
    }
  ]
}

```

The condition of the **if** statement is a simple JavaScript expression which uses the same format as noted in the previous sections. The expression is described in the **exp** property of the cue.

Note that test expressions can use all of the Javascript operators like:

- **==** or **===**: is equal
- **!=** or **!==**: is different
- **&&**: and
- **||**: or
- **>**: greater than
- etc.

13.2.5.3 While/Endwhile cues

While/Endwhile cues allows performing loops of cues.

The overall format is:

```

while test_expression
do something if the expression is true
endwhile

```

While/Endwhile cues are written by the specifying 2 different types of **command**:

- **while**: this cue tests the condition expressed in the **exp** property. This cue starts the block of actions which must be executed if the condition is true.
- **endwhile**: this cue ends the block of cues which must be executed. The task execution jumps to the matching **while** cue.

Note that While/Endwhile cues can be nested.

Example: changing the level of the first 10 channels of a Lighting System to 100:

```

{
  "name": "While_Example",
  "cues": [
    {
      "command": "exp", "exp": "local.index = 1"
    },
    {
      "command": "while", "exp": "local.index <= 10"
    },
    {
      "command": "exp",
      "exp": "perform('Malaga_Lighting','Set Level',{ Channel : local.index, Value: 100})"
    },
    {
      "command": "exp", "exp": "local.index = local.index + 1"
    },
    {
      "command": "endwhile"
    }
  ]
}

```

13.2.5.4 Wait cues

Wait cues allows the task to pause for the specified time and then resume. The `command` property is `wait`. The time the task will pause for is specified in milliseconds in the `delay` property.

Example: powering a projector on, waiting 2 secs and recalling a light preset

```

{
  "name": "Wait_Example",
  "cues": [
    {
      "command": "exp",
      "exp": "perform('Malaga_Projector','Set Power',{ Value : 'On' })"
    },
    {
      "command": "wait", "delay": 2000
    },
    {
      "command": "exp",
      "exp": "perform('Malaga_Lighting','Set Level',{ Channel : 2, Value: 100})"
    }
  ]
}

```

13.2.5.5 Using local task variables

A task can use the `local` object in order to manage task local variables.

Local task variable are created with expression which sets the value of a property of the `local` object. For example the following line create a local task variable of with the name `counter` with an initial value of `0`.

```

{
  "command": "exp", "exp": "local.counter = 0"
}

```

A task can use the value of a local task variable by merely referencing the variable. The following example shows how to test a local variable value.

```
{
  "command": "if", "exp": "local.counter === 0"
},
{
  "command": "exp", "exp": "local.counter = 1"
},
{
  "command": "endif"
}
```

13.3 Variables

The user can define points of type 'variable' which are managed by a Control Server.

The way these points are handled depends on the type of Control Server.

13.3.1 Control Server Type 1

Variables managed by Control Server of type 1 are defined in the Control Server project itself. These variables cannot be edited via Overture Configurator and are added to the Overture database when the CS1 project is ingested.

13.3.2 Control Server Type 2

On the other hand, Overture Configurator allows creating, modifying and deleting variables managed by a Control Server of type 2.

13.3.2.1 Creating a CS2 Variable

- on the Configurator Points View
- click on the 'Add' button
- select type 'Variable'
- select the subtype of the variable point:
 - integer
 - string
 - enum. See [Enum Variable Metadata](#)
 - real
 - time
 - data
- select the Control Server which will manage this variable
- click the 'Save' button to save the variable

13.3.2.2 Modifying a CS2 Variable

- on the Configurator Points View
- select the variable point to modify
- change the subtype of the variable point (if required)
- change the Control Server which manages this variable (if required)
- for variable point with subtype 'enum', edit the Metadata if required. See [Enum Variable Metadata](#).

- click the 'Save' button to save the variable.

13.3.2.3 Deleting a CS2 Variable

- on the Configurator Points View
- select the variable point to delete
- click the 'Delete' button

13.3.2.4 Enum Variable Metadata

A User must further configure variable points with have a subtype 'Enum' by specifying the point metadata as a JSON text.

- on the configurator Points View
- select the variable point to delete
- click the 'Edit Metadata' button
- enter variable points metadata in JSON format
- click OK in order to close the Metata dialog
- click the 'Save' button to save the point

The format of the metadata describing an enum variable is:

```
{
  "enums": [
    "Off",
    "On"
  ]
}
```

The JSON must have a root property named `enums` which is an array of strings. Each string of the array is the string which will be associated to an index value of the enum variable. The first string is associated to the value `0`, the second string is associated to the value `1` and so on.

14 APPENDIX D

14.1 Network Architecture and Security

This section is for system integrators, and will give an overview of the network architecture of Overture, for different deployment option. In the section thereafter, we will describe the need for and details about internet access required for Overture installations, as well as network setup recommendation.

14.2 Overture Architecture

Overture is a software solution that is running on multiple servers, which can be installed on-premise and/or in the cloud. Three servers are required:

- Overture RMM Server: Hosting the Monitor application, communicating with one or multiple UX servers. Only available in a Cloud deployment.
- Overture UX server: Hosting, amongst others, the Overture Home, Magic Menu and configurator applications. Communicates optionally with one RMM server, and one or more multiple UX servers. Available both in cloud as on-premise deployment
- Overture CX server: responsible for the communication to the devices. Communicates with one UX server. Available as on-premise Deployment

Deployment types per servers

Server	Cloud Deployment	On-premise Deployment
RMM	X	-
UX	X	X
CX	-	X

The diagrams in this chapter use the following legend

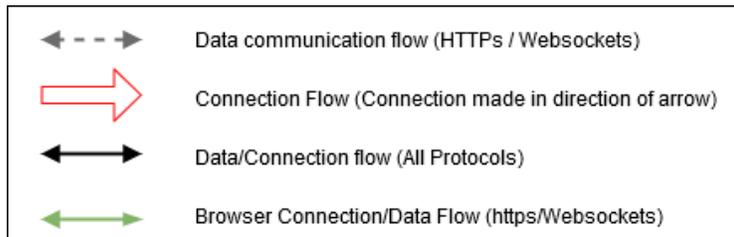
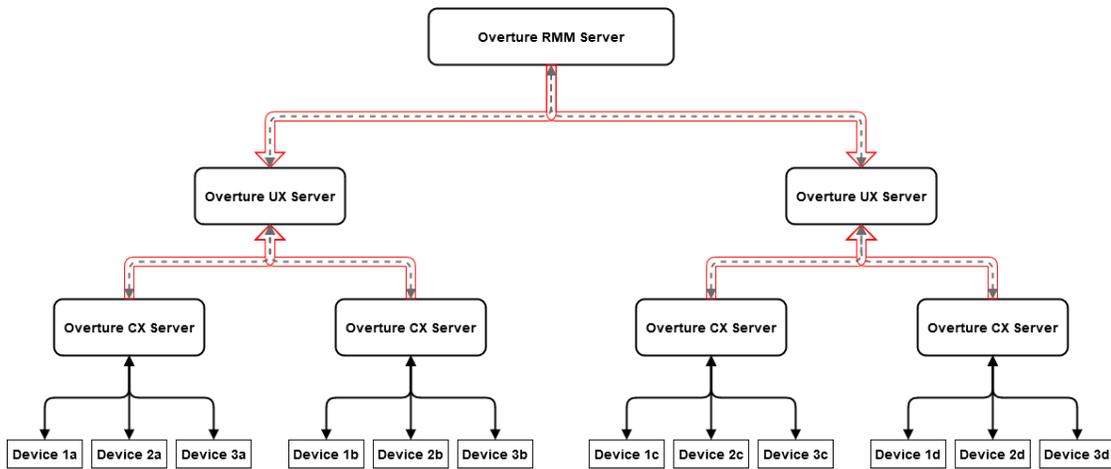


Figure 1: Overall Overture architecture



14.2.1 Overture On-Premise architecture

When we talk about Overture On-Premise, it means that both the UX as the CX server are installed on-premise, while the RMM server is installed in the cloud. The RMM server is set up by Barco in the cloud (hosted on Amazon's AWS Ireland). In the simplest form, the UX and CX software are installed as one virtual machine (VM) image, running in a Virtualization environment or Hypervisor (see Figure 2).

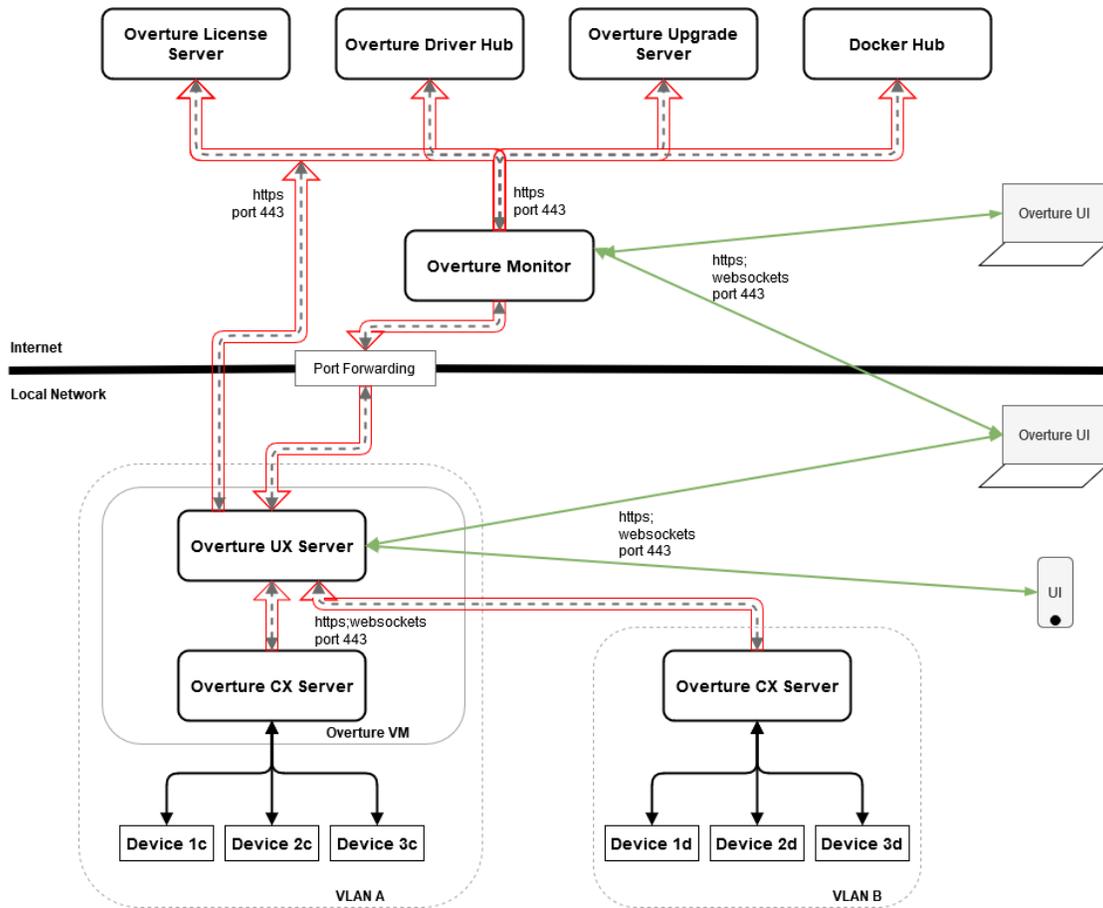
In this diagram, the communication between the different components is indicated by the arrows. The direction of the arrow indicates where the communication originated (arrows point from TCP client to TCP server).

Let's zoom in into the communication between the different servers. First, we'll look at the communication of the on-premise servers (UX and CX): Communication between those two servers happens over http or https and normal or secure web sockets (this can be configured in the Overture Software). The communication happening between CX and UX are updates of the status of the variables of the devices, as well as triggers to execute commands on the devices. The CX is responsible for handling all communication with the actual devices. The protocol and ports used for this communication are defined by the device manufacturers. The UX server itself connects through https to three Overture servers and one third-party server that are hosted on the internet. This connection to the internet is required for 3 purposes: software updates and upgrades, access to device drivers and license validation. More information on the content that is shared with the internet servers can be found in section 3. Access to internet.

In this architecture, it is important that the CX server can access the devices without any problem. A best practice to achieve this is by making sure they are part of the same VLAN. When faced with a more complex setup, this is often not possible with one CX server (e.g. the devices are spread over different locations). In that case, one should use an architecture, where one puts a separate CX per VLAN, which all connect to the same UX server. Of course, this means that all CX servers on the separate VLANs need to be able to contact the centrally located UX server. This is shown with VLAN A and VLAN B in Figure 2. Next to the on-premise servers, there is also the server in the cloud, the Overture RMM server. This Overture RMM server needs to be able to contact the on-premise UX server. Therefore, a port-forwarding must be setup into the on-premise firewall. Protocol involved are http/https and websockets/secure websockets, depending on whether the UX on-premise has been setup with http or https. In this setup, https is highly recommended.

The port-forwarding can use any entry port number, but the on-premise UX still expect communication on port 80/443. All communication between Overture RMM server and the UX server happens over https and secure websockets. It's a good best practice to put the UX server in the DMZ, such that there are no outside connections entering the actual network.

Figure 2: Overture On-Premise with multiple CX servers

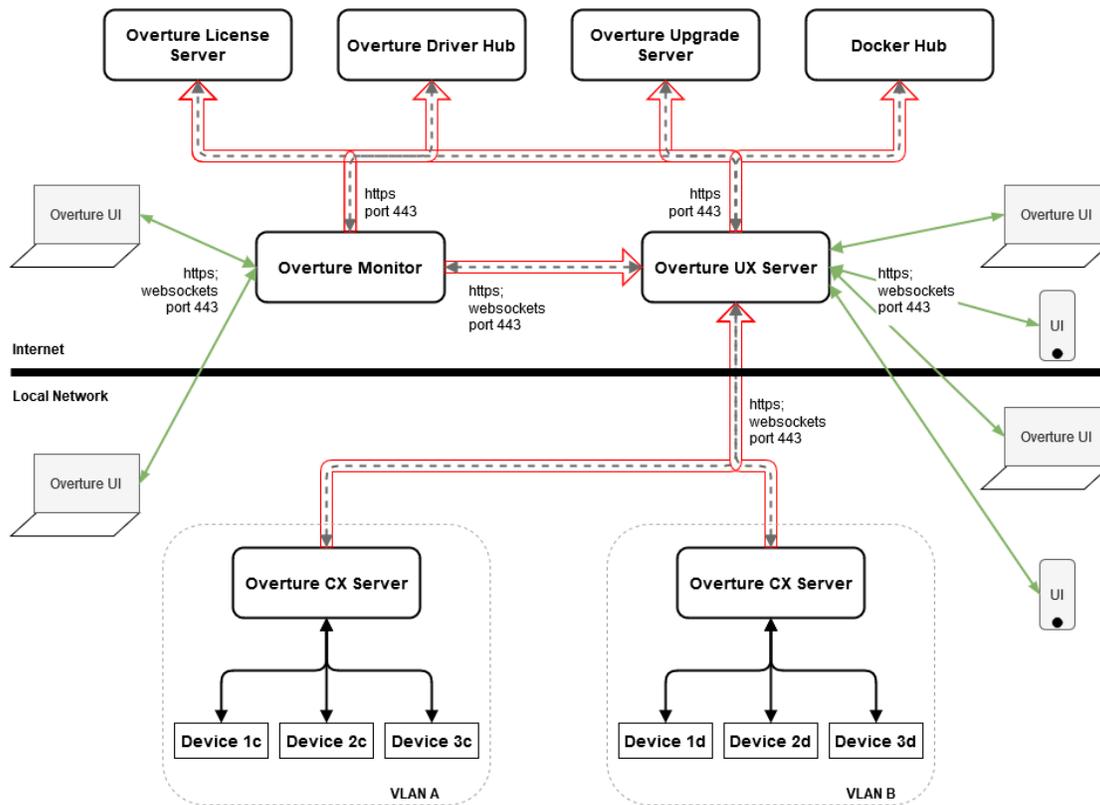


14.2.2 Overture Cloud Architecture

In case of an Overture Cloud deployment, the RMM server and the UX server are set up by Barco in the cloud (hosted on Amazon's AWS Ireland) and the CX server is installed on-premise. Like with an Overture On-Premise installation, there can be multiple CX servers installed and configured per UX server.

Barco sets up secure communication between the RMM server and the UX server, using https and secure websockets. All communication between the CX and the UX happens over https or secure websockets. The connection is initiated from the CX, meaning that there is no need to open ports for incoming traffic in the firewall. In Figure 3 the arrows indicate where the communication originated (arrows point from client to server). The role of the UX and CX is the same between Overture On-Premise and Overture Cloud. As a result, similar communication to devices (for CX) and external cloud servers is expected.

Figure 3: Overture Cloud with multiple CX servers



14.2.3 Local network connectivity and traffic

On both Cloud architecture and On-Premise architecture, CX Server is designed to communicate with all A/V devices that are controlled and monitored, through the local network (LAN). To perform these operations, the CX Server uses various protocols to communicate with these devices. The protocols involved are device dependent, therefore there is no specific security measure nor consistency on the nature of the traffic.

However, A/V devices control generally relies on small network packets frequently exchanged with the devices. While this type of traffic uses a small network bandwidth, all processes that relies on packet inspection, such as firewalls, may have their load increased.

As stated in [Overture On-Premise architecture](#) Chapter, as a layer of defense to improve security, as well as a measure to isolate this traffic, a best practice is by making sure that CX Server and A/V devices are part of the same VLAN. When faced with a more complex setup, this is often not possible with one CX server (e.g. the devices are spread over different locations). In that case, one should use an architecture, where one puts a separate CX per VLAN, which all connect to the same UX server. Of course, this means that all CX servers on the separate VLANs need to be able to contact the centrally located UX server.

A/V Devices are identified either by IP address or a DNS. When DNS are involved, resolving these DNS into IP addresses involves a DNS Server. Therefore, it is recommended to make sure that reaching the DNS Servers does not require to pass through a firewall, as this may increase the load of the firewall.

14.3 Access to internet

As described in [Overture Architecture](#) chapter of this document, the Overture system requires the ability to communicate with servers on the internet through https over port 443. This is required to provide access to software updates & upgrades (On-Premise) as well as new & updated device drivers (On-Premise and Cloud). Additionally, a license validation-check, as well as entitlement updates, are done via this communication path.

In this chapter, we will focus on the 3 types of communication to the internet in case of an on-premise installation

14.3.1 Overture Driver Hub

Overture uses device drivers to control and monitor AV and IT devices via the network. As there are continuously new devices made available, or changes made to existing devices, the repository of Overture device drivers is continuously updated by Barco and its partners. To ensure all Overture users have access to the latest and newest drivers, those drivers are made publicly available via a cloud-based server. Every Overture installation requires access to the Driver Hub. All communication with the Driver Hub happens via https calls on port 443. When accessing the driver management functionality in Overture, REST API calls retrieve the latest information from the Driver Hub. When installing a new driver, a zip archive is downloaded via https from this web server. Apart from the Overture installation's unique identifier, no other information about the Overture installation is sent to the Driver Hub. All communication is initiated from within the local network.

Connection details:

Address: <https://drivers.overture.barco.com>

Port: 443

Encryption: using a strong protocol (TLS 1.2), a strong key exchange (ECDHE_RSA with P- 256), and a strong cipher (AES_256_GCM).

Certificate issuer: GlobalSign CA.

14.3.2 Overture Upgrade Server

When there is an update available for the Overture software, the easiest way to install it is from within the Overture Configurator. To enable this functionality, the Overture software contacts the Overture upgrade server regularly to check whether any updates are available. When updates are available on the server, they are visible in the Overture Configurator. The update availability check is done using a REST API over an https connection. When triggering the software update, a small file is downloaded from the Overture Upgrade server. The file itself is signed to prevent any abuse. After validation of the signature, the actual software image is downloaded from a central, third-party software repository (<https://hub.docker.com>). No information about the Overture installation is sent to this server. All communication is initiated from within the local network.

Connection details:

Address: <https://update-services.overture.barco.com/>

Port: 443

Encryption: using a strong protocol (TLS 1.2), a strong key exchange (ECDHE_RSA with P-256), and a strong cipher (AES_128_GCM)

Certificate issuer: COMODO CA Limited

Address: <https://hub.docker.com>

Port: 443

Encryption: using a strong protocol (TLS 1.2), a strong key exchange (ECDHE_RSA with P- 256), and a strong cipher (AES_128_GCM).

Certificate issuer: Amazon/Let's Encrypt

14.3.3 License servers

For licensing purposes, Overture connects to a license server regularly. Depending on the version of Overture as well as the licensing model you're in, Overture can either use the Overture License Server or the Barco License Server.

Overture License Server is used with all versions prior to 3.2.0 or with version 3.2.0 using legacy license model. Barco License Server is used with version 3.2.0 with subscription license model using entitlements.

14.3.3.1 Overture License server

The licensing mechanism for Overture requires that at least once every 30 days, a connection is established with the Overture license server. Once this connection is established, a history of fingerprints of the Overture configuration of that installation is shared with the Overture License server. The information is required by the Overture license server for license enforcement reasons. The information that is shared is a non-reversible fingerprint history. It does not contain any usable information about the installation or its usage. The fingerprints cannot be interpreted or processed in any way to deduce any usable information from the system or its use. All communication is initiated from within the local network.

Connection details:

Address: <https://license-services.overture.barco.com/>

P 242 / 270



Port: 443

Encryption: using a strong protocol (TLS 1.2), a strong key exchange (ECDHE_RSA with P- 256), and a strong cipher (AES_128_GCM).

Certificate Issuer: COMODO CA Limited

14.3.3.2 Barco License Server

The licensing mechanism for Overture requires that at least once every 30 days, a connection is established with the Barco license server. Once this connection is established, a history of fingerprints of the Overture configuration of that installation is shared with the Barco License server, and an update of the entitlement is performed, if necessary. The information is required by the Overture license server for license enforcement reasons as well as for maintaining the entitlement information, such as validity and the associated feature list. When the licensing enforcement is performed, the information that is shared is a non- reversible fingerprint history. It does not contain any usable information about the installation or its usage. The fingerprints cannot be interpreted or processed in any way to deduce any usable information from the system or its use. All communication is initiated from within the local network.

Connection details:

Addresses:

- <https://iam.cmp.barco.com/>
- <https://licensemanager.cmp.barco.com/>

Port: 443

Encryption: using a strong protocol (TLS 1.2), a strong key exchange (ECDHE_RSA with P- 256), and a strong cipher (AES_128_GCM).

Certificate issuer: Amazon, US/Let's Encrypt

14.4 FAQ

14.4.1 What happens when the network infrastructure goes down?

As with all solutions that rely on a central server and depend on network technology, the moment that the network goes down, the solution will stop working (just like your connection to mail server and business critical applications would be down). The moment the network is up and running again, the solution is immediately fully functional again. Luckily, network infrastructure and architectures have evolved a lot in the past 10 years, making those network outages very rare.

14.4.2 Is any content data being shared?

Overture has no access to content, so cannot share any content from the meeting rooms. The only information that Overture can access are the control commands and the statuses of the A/V devices (e.g. the volume level of the display, the active input being used on a projector).

14.4.3 Is the communication encrypted in any way?

In Overture On-Premise installations, the system can be optionally configured to encrypt the communication between CX and UX, and to web applications hosted by the UX through https and secure web sockets. This requires the installation of a certificate, purchased from a trusted certification authority. In case of Overture Cloud, all communication between CX and UX is encrypted through https and secure websockets, using Barco's certificate.



15 APPENDIX E

15.1 Overture ServiceNow Integration Guide

This guide gives the instructions on how to configure ServiceNow ticketing functionality to be integrated with Overture. Once configured, Overture can make a link with ServiceNow and create tickets when alarms are going off.

This guide covers the usage of Incident (ITSM) and Customer Service Management CSM (Case)

15.2 ServiceNow Configuration Steps Overview

15.2.1 Steps for Incident

- Create a Role for Barco REST API
- Assign the personalize_choices role Barco role
- Create a User and Password in ServiceNow for Barco REST API
- Create an Application Registry entry for Barco REST API
- Create ACL on the Incident table for the Barco user

15.2.2 Steps for Case Management

- Install the Customer Service Management CSM (for Case)
- Create a Role for Barco REST API
- Assign the personalize_choices role Barco role
- Create a User and Password in ServiceNow for Barco REST API
- Create an Application Registry entry for Barco REST API
- Create ACL on the Incident table for the Barco user

15.3 Common Pre-requisite

15.3.1 Login as admin

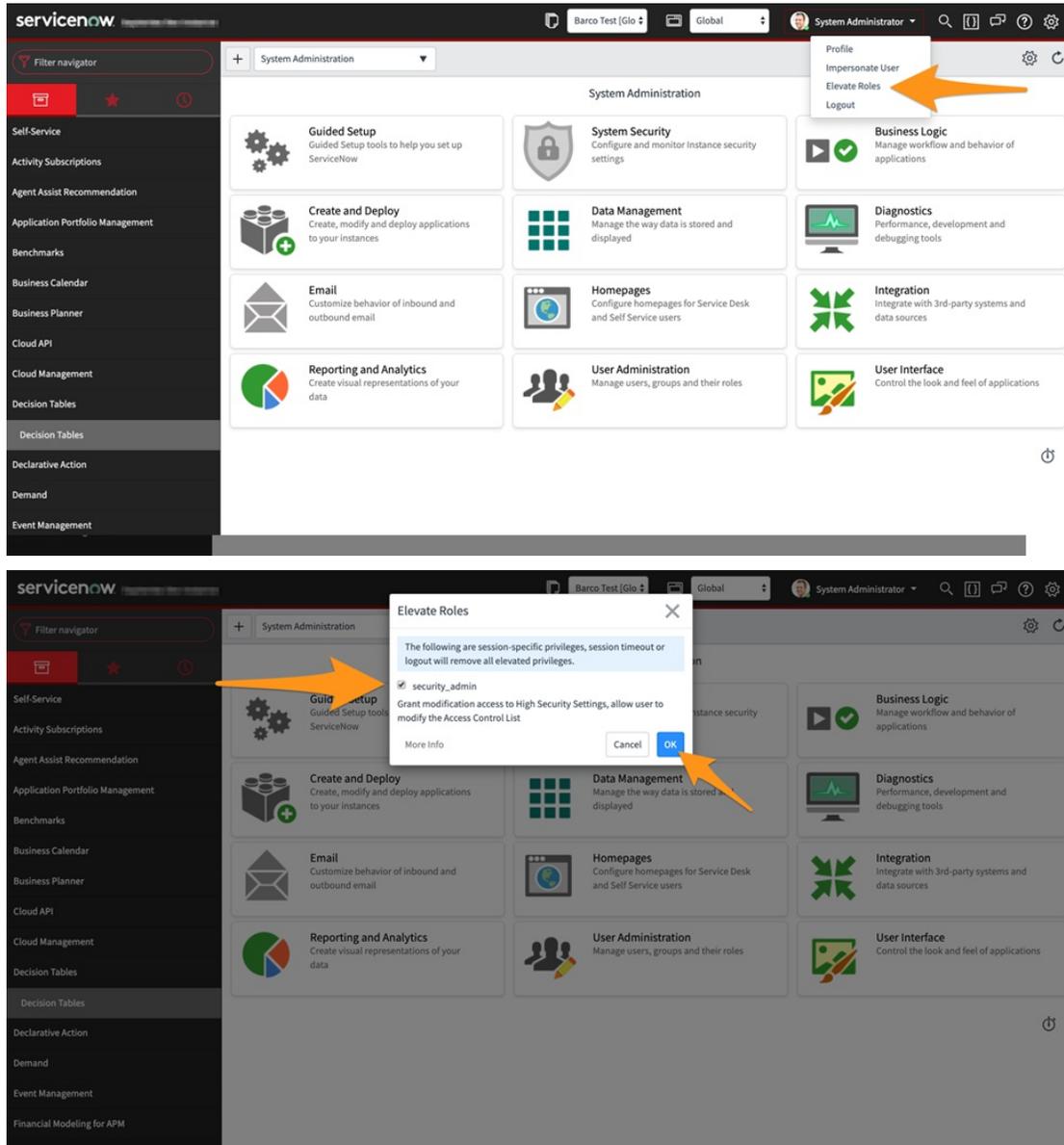
Login as admin in Barco ServiceNow instance to be able to create the role, user and ACL required to configure the REST API for the integration.



The screenshot shows the ServiceNow login interface. At the top, there is a dark header with the 'servicenow' logo. Below the header is a white login box containing two text input fields: 'User name' and 'Password'. To the right of the 'Password' field is a green 'Login' button. Below the 'User name' field is a link that says 'Forgot Password?'.

15.3.2 Elevate admin role

Elevate the admin role to `security_admin` temporarily (using the login user menu) to be able to create role, user and access controls.



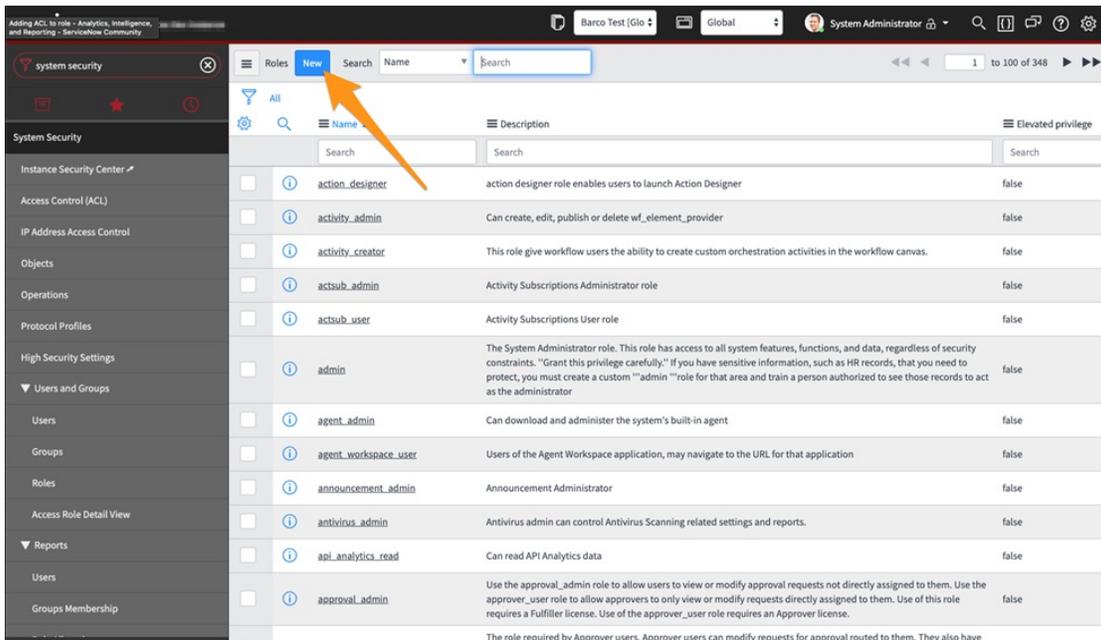
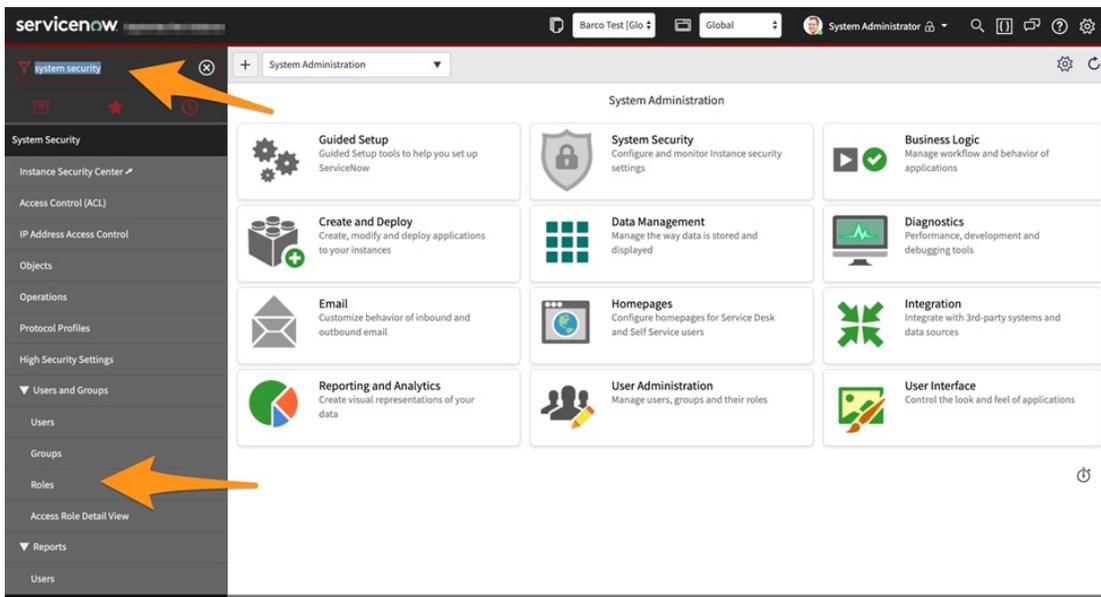
15.4 Configuration for Incident

15.4.1 Role Configuration

15.4.1.1 Role list

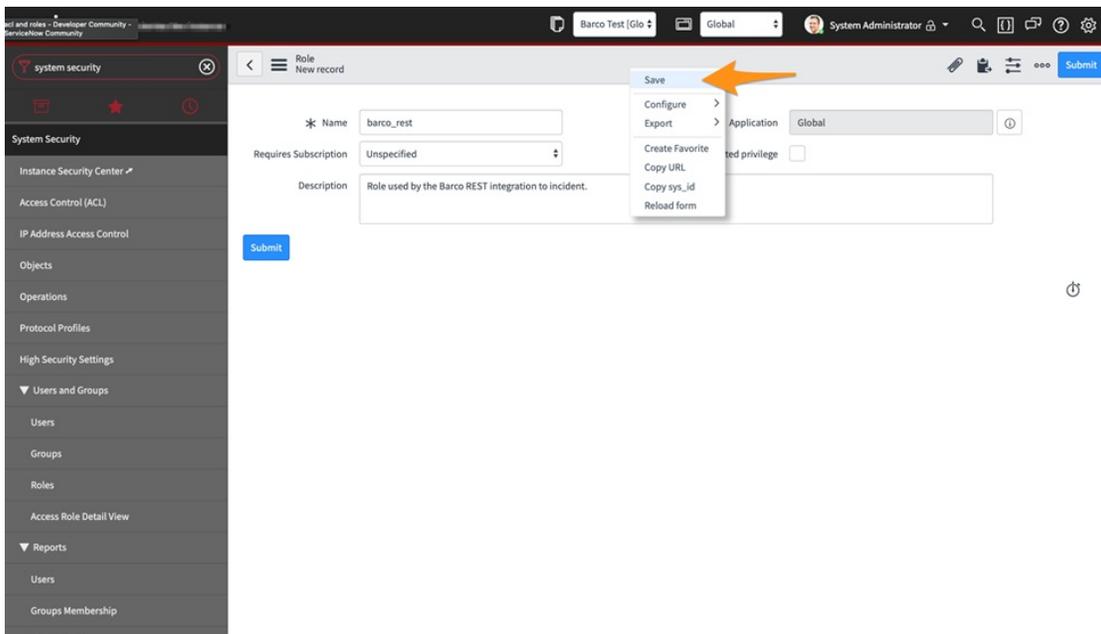
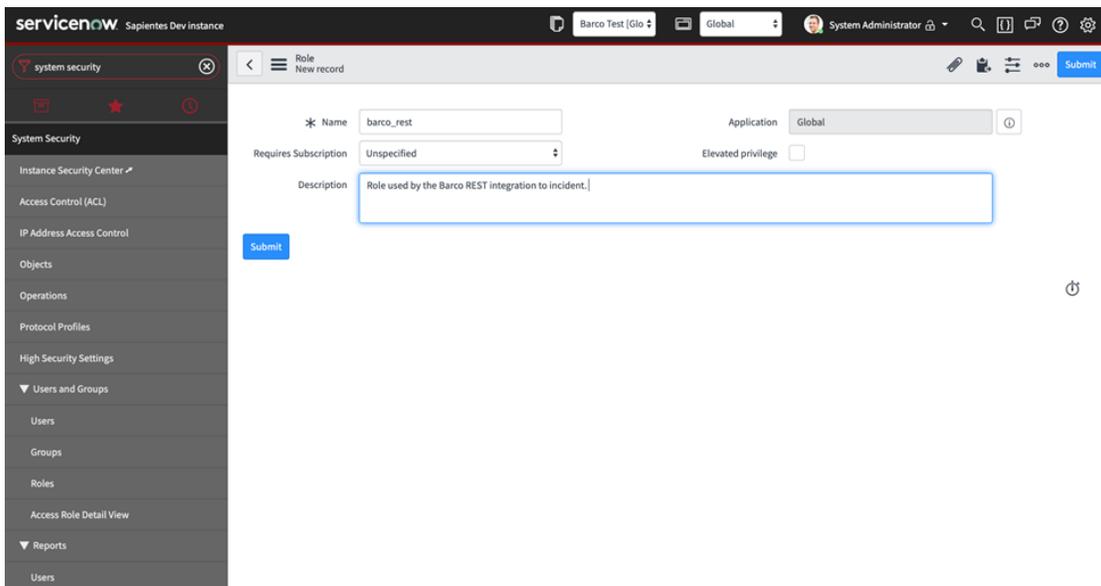
Go to the Role list in ServiceNow by using the Navigation filter to find 'System Security' and click on the Roles menu item. Once on the role list, click on the New button to create a new role.

P 245 / 270



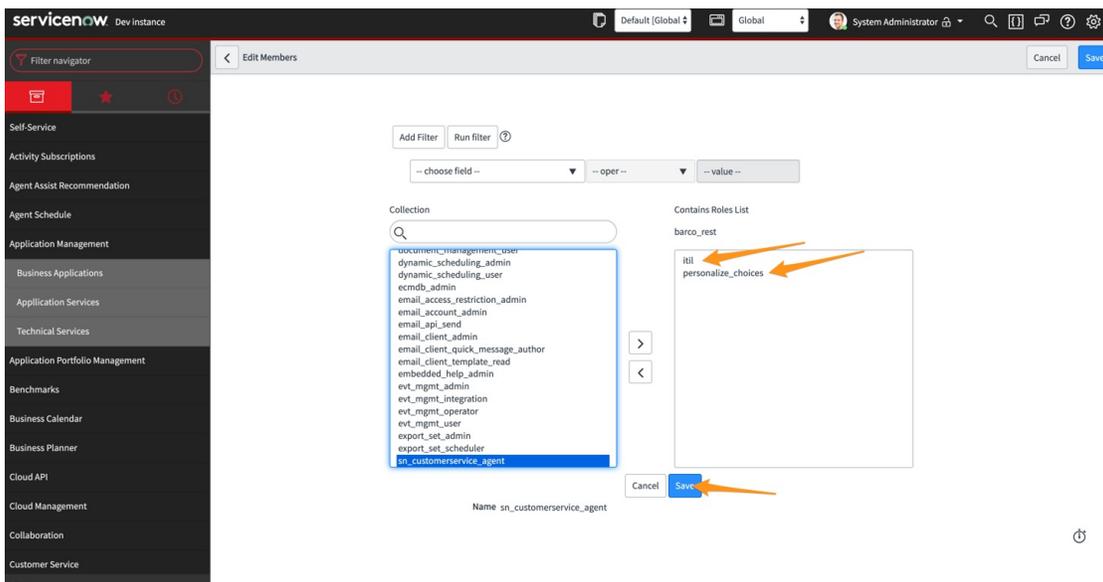
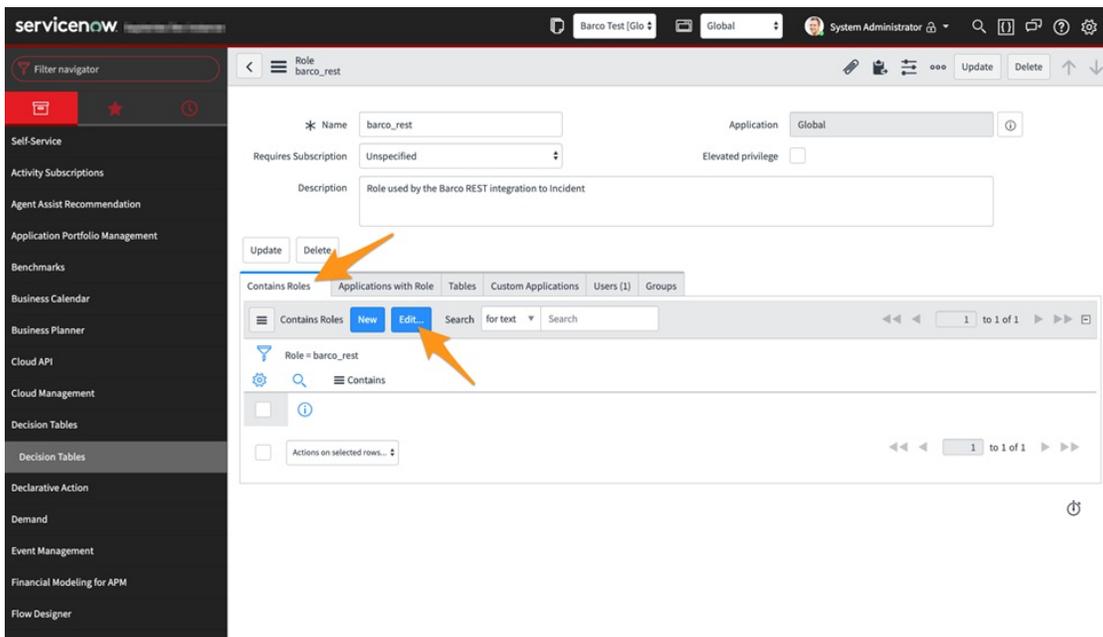
15.4.1.2 Create a role

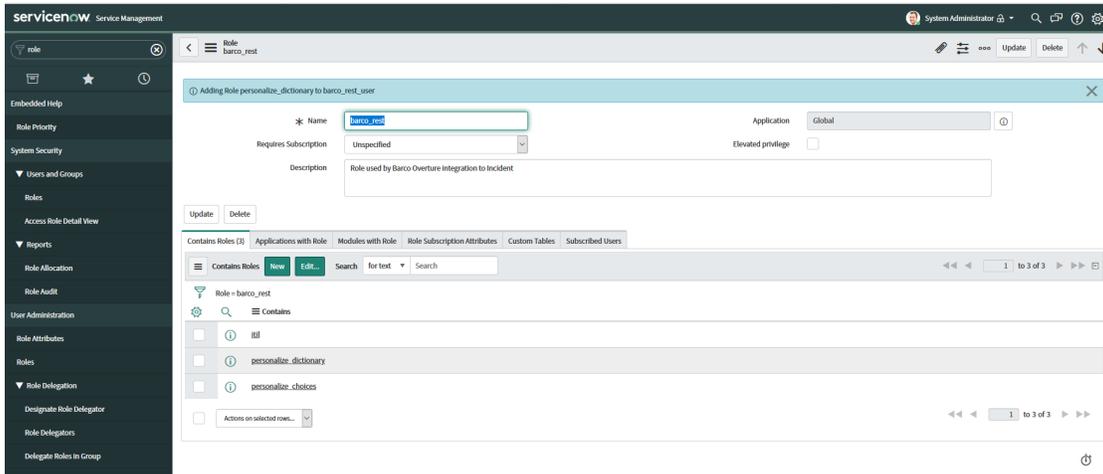
Enter the role information as shown. The role name can be changed to match Barco naming convention if it has no space. Once done, right click on the top of the form (grey bar) and select Save to save the record and keep it open.



15.4.1.3 Add roles to Barco role

Select the Contains Roles and click on edit to add the role require to be able to use the [sys_choices](#) table. Add the [personalize_choices](#) role and click save. The role should now appear in the Contains Roles tab list. Repeat the operation to add the [personalize_dictionary](#) and [itil](#) roles. The "Contains Roles" should now look like the 3rd picture below. Click update at the top right of the form to update the Barco role and exit the form. The role will be used to create the ACL and it will be the role assigned to the Barco user.

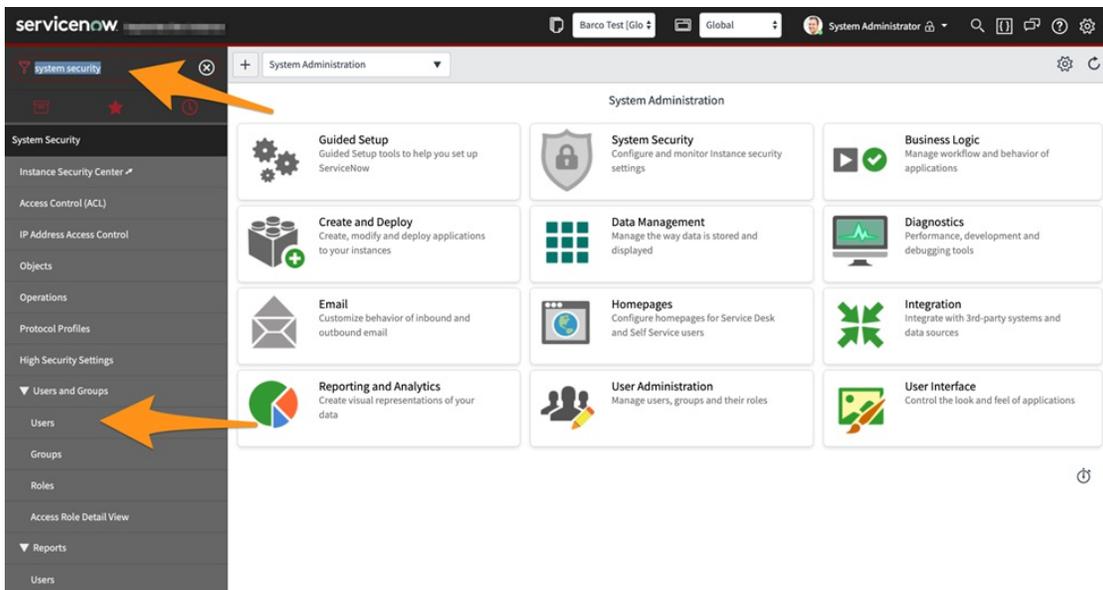


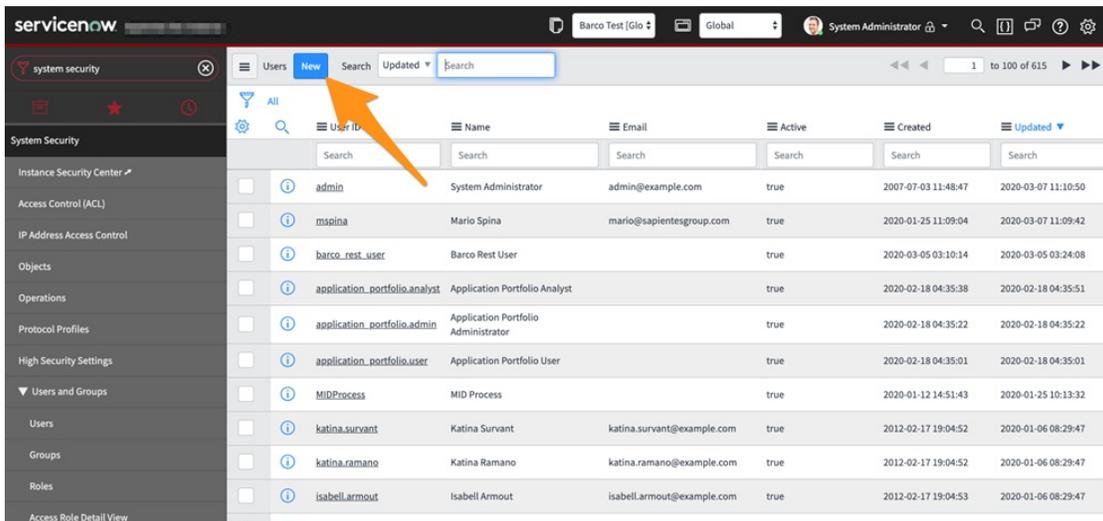


15.4.2 User Configuration

15.4.2.1 User list

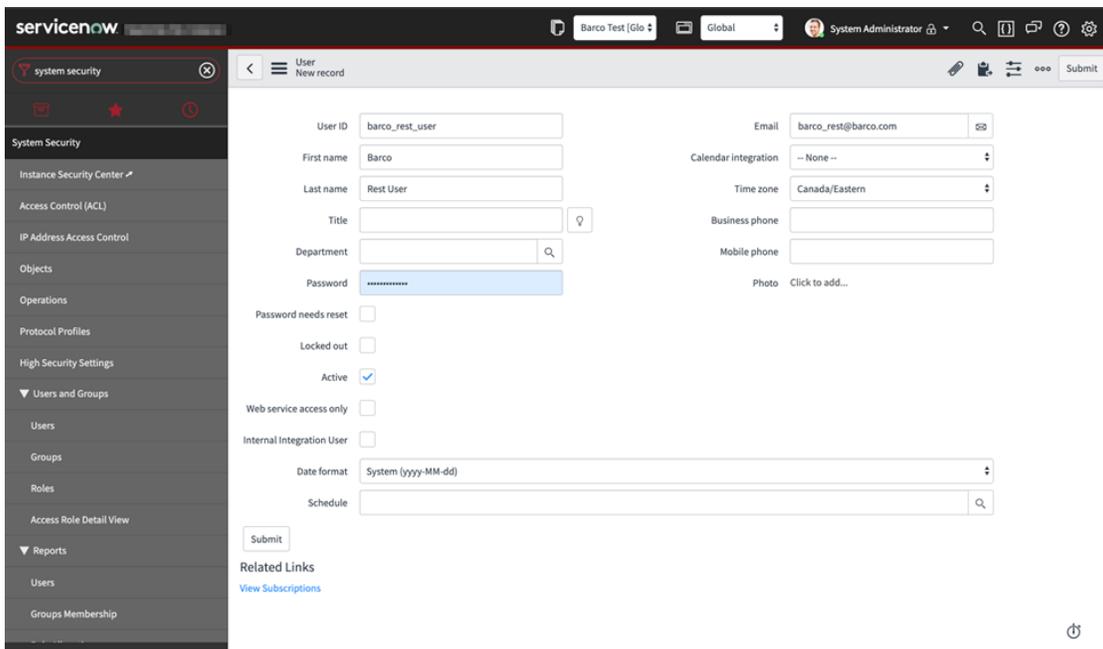
Go to the User list in ServiceNow by using the Navigation filter to find 'System Security' and click on the Users menu item. Once on the User list, click on the New button to create a new user.

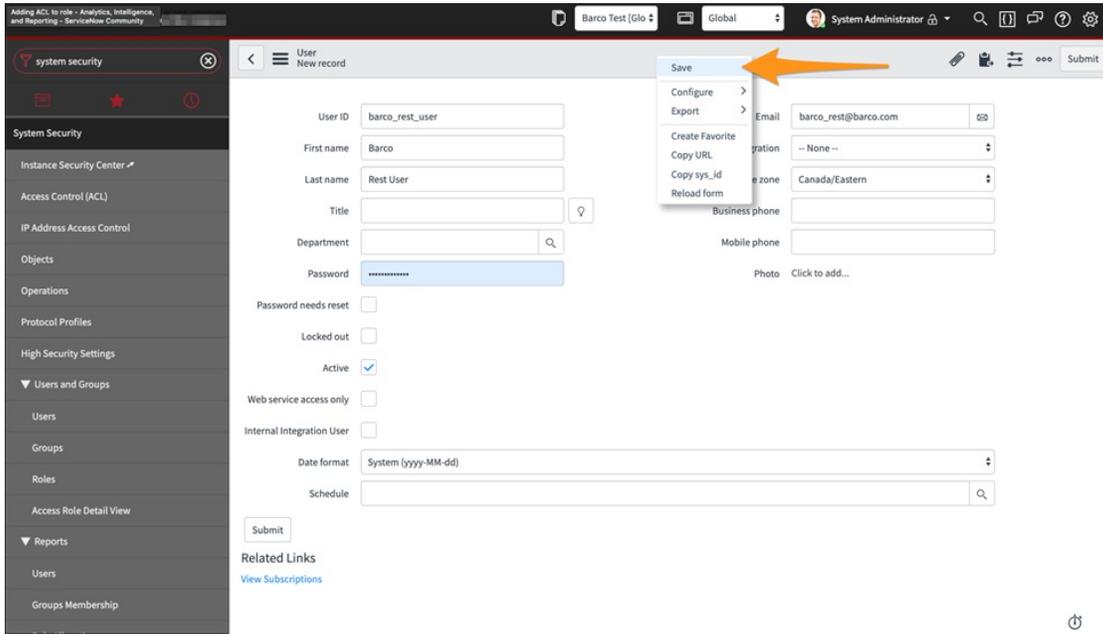




15.4.2.2 Create a user

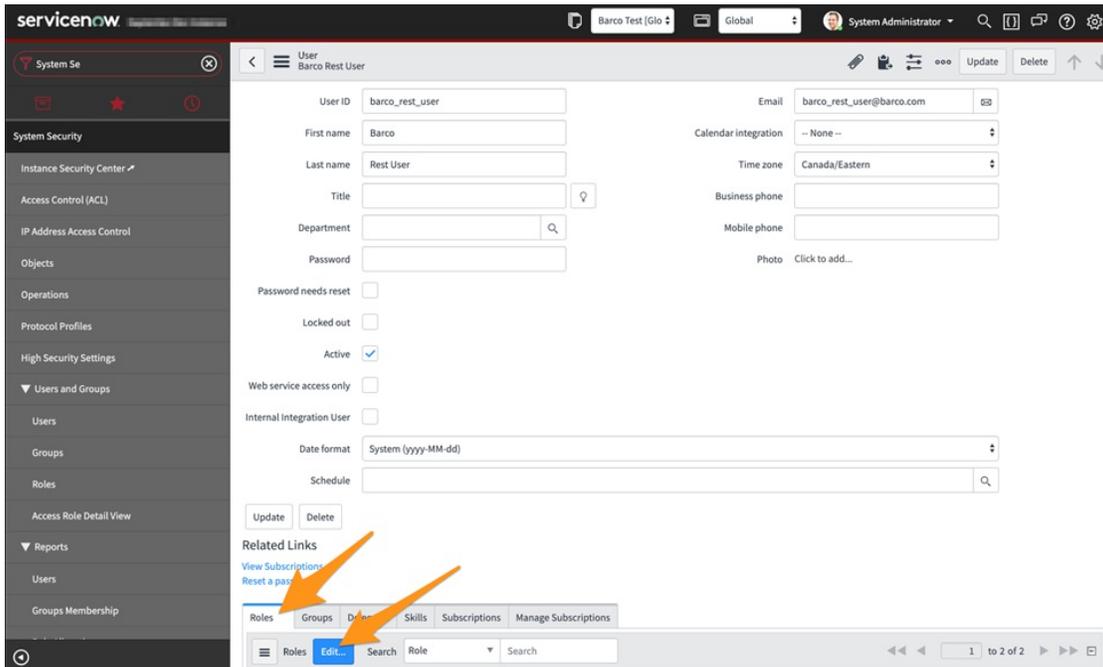
Enter the user information as shown. The user name can be changed to match Barco naming convention if it has no space. Once done, right click on the top of the form (grey bar) and select Save to save the record and keep it open.

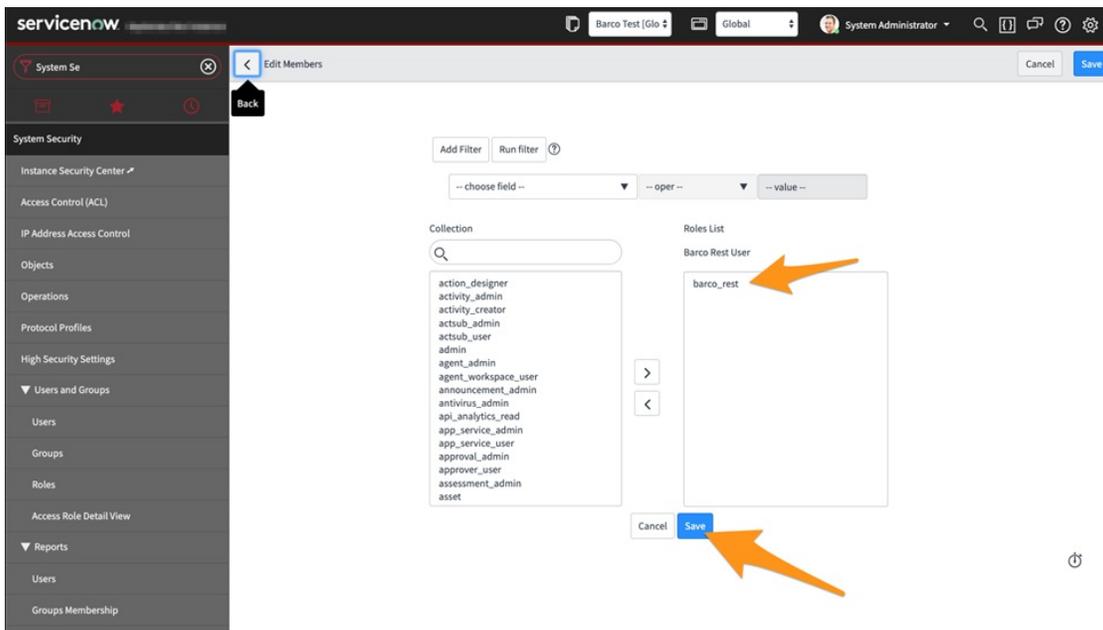




15.4.2.3 Add role to user

In the record for the Barco user, find the Roles tab in the bottom section of the form. Click on Edit to add the Barco role to the Barco user. Search for the Barco role and add it to the Role list column. Click on save once done. Validate that 2 roles are now in the list under the Roles tab. One for the Barco role and one for the [personalize_choices](#).

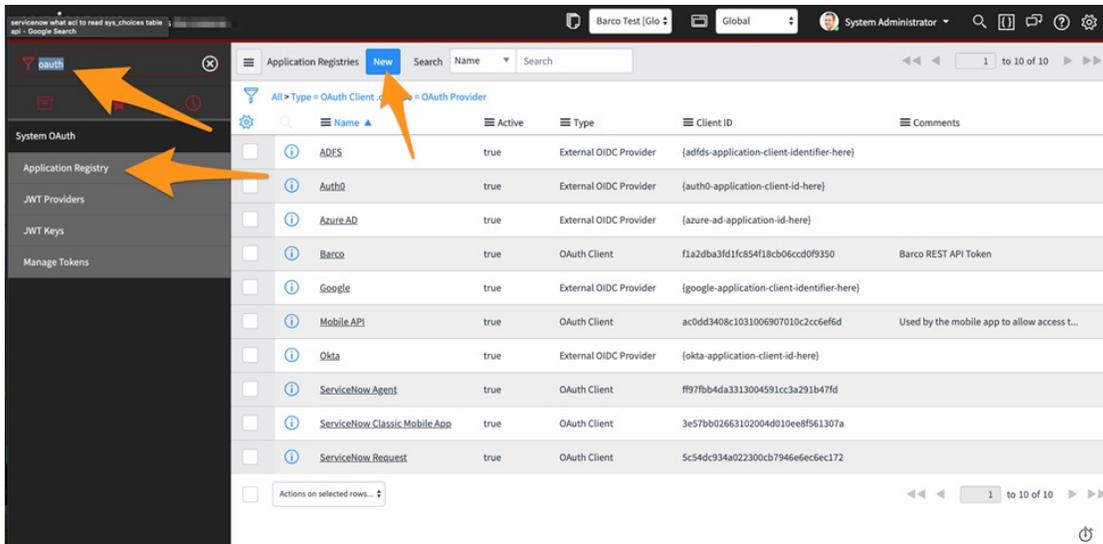


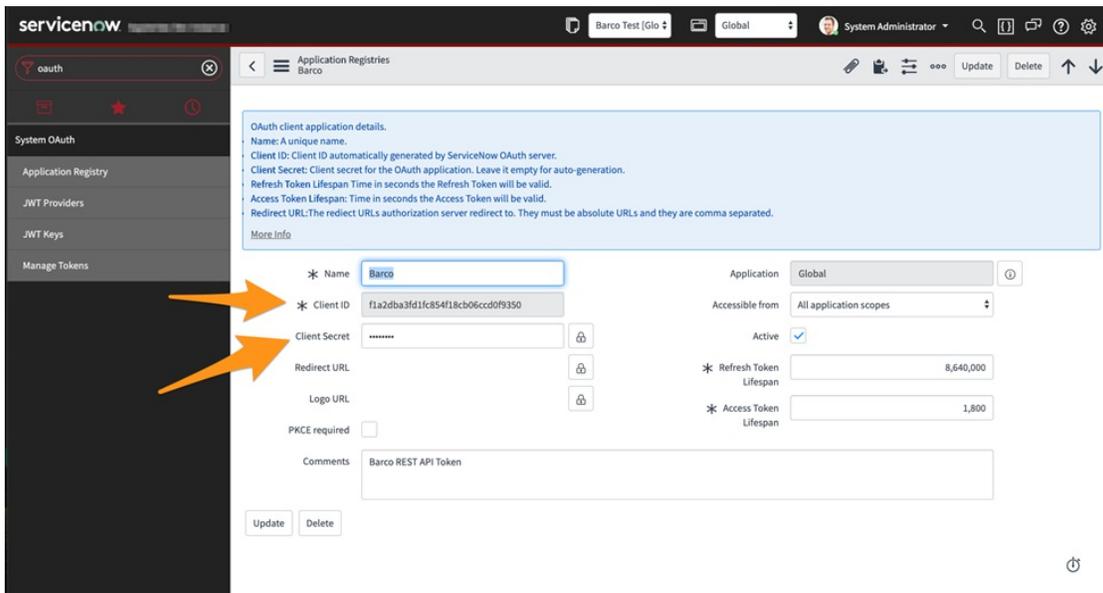


15.4.3 oAUTH Configuration

15.4.3.1 Create an application registry

To be able to use oAuth v 2.0 with the Table API, you need to create an Application Registry. Look for oAuth in the navigation filter, select application registry in the menu and click New to create a new one. Entered the data as shown and write down the Client ID and Client Secret for the REST API calls, click submit to create the entry. The refresh and token lifespans are in seconds.

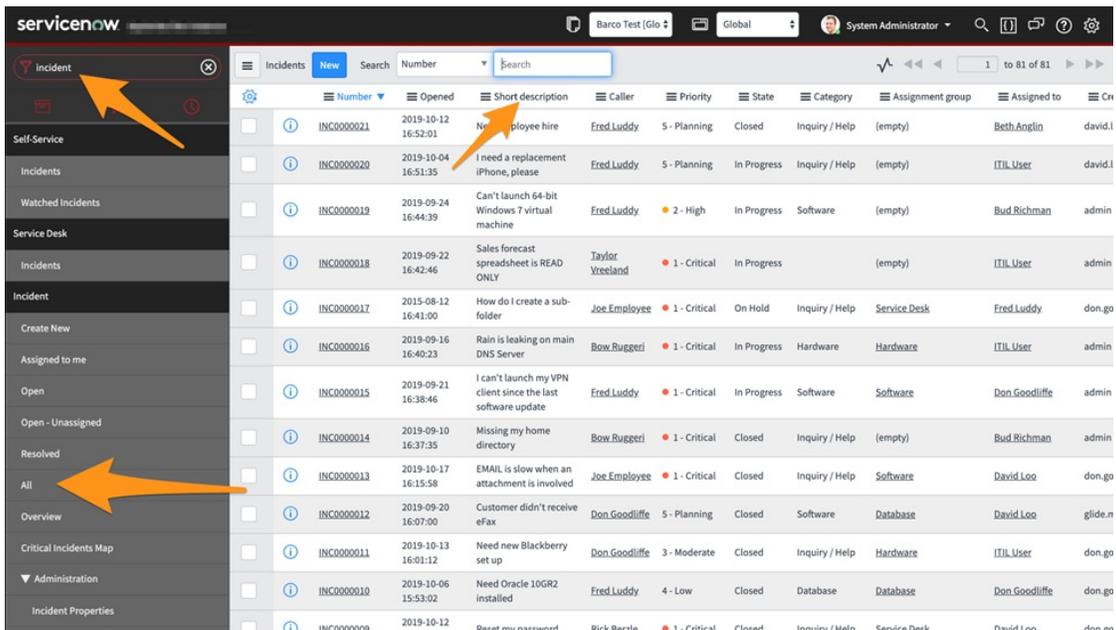


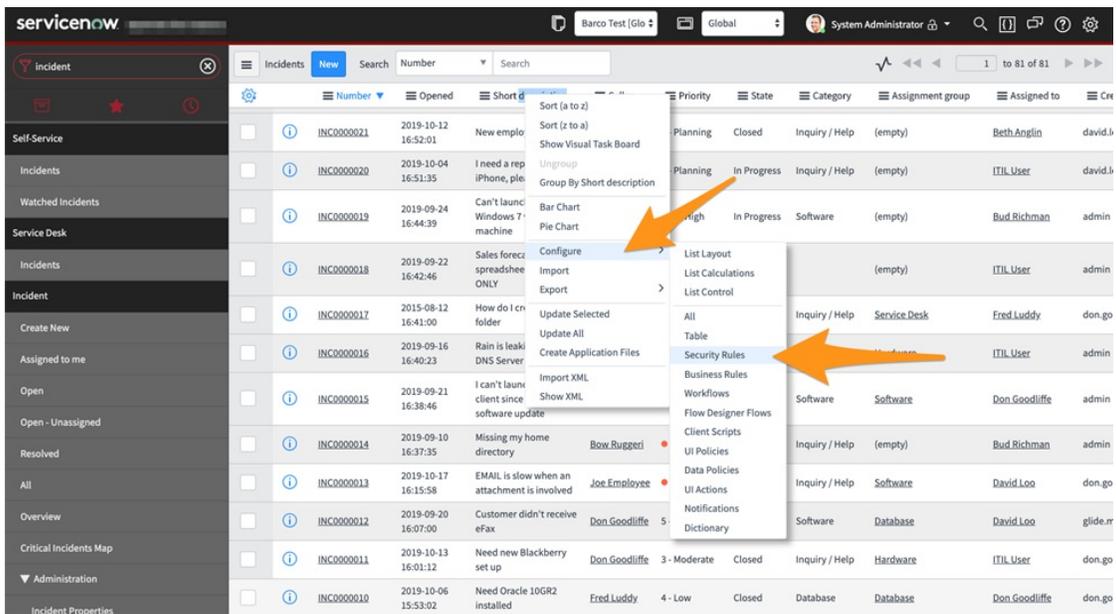


15.4.4 Incident Access Controls

15.4.4.1 Navigate to the Incident list

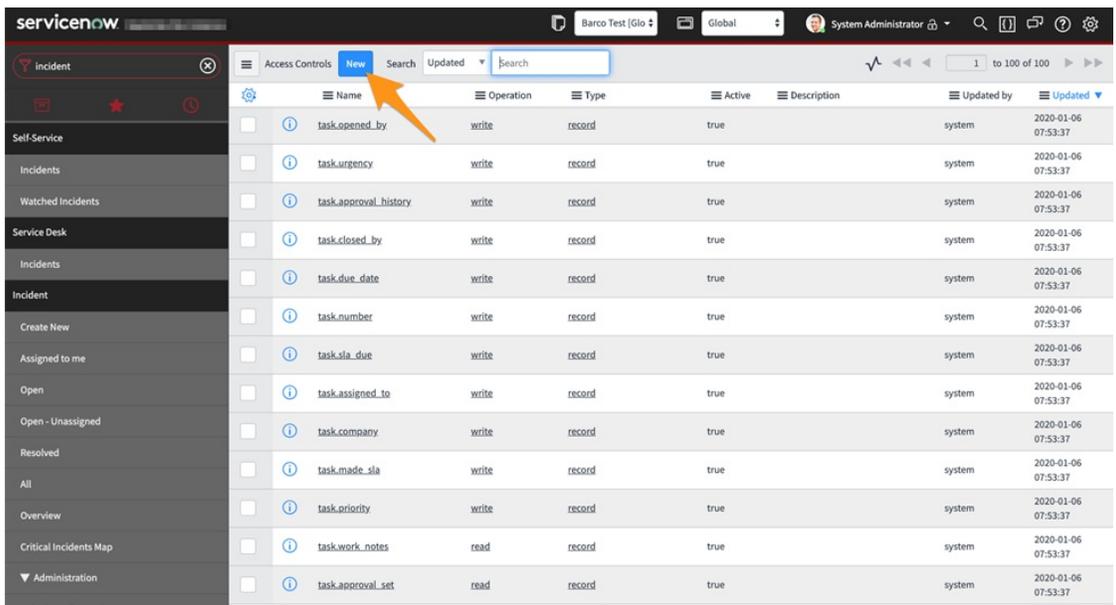
Navigate to the Incident list and right click on the title of a column. Select configure then Security Rules to get the list of ACL related to the Incident table. Make sure that you are still `security_admin` before doing so.

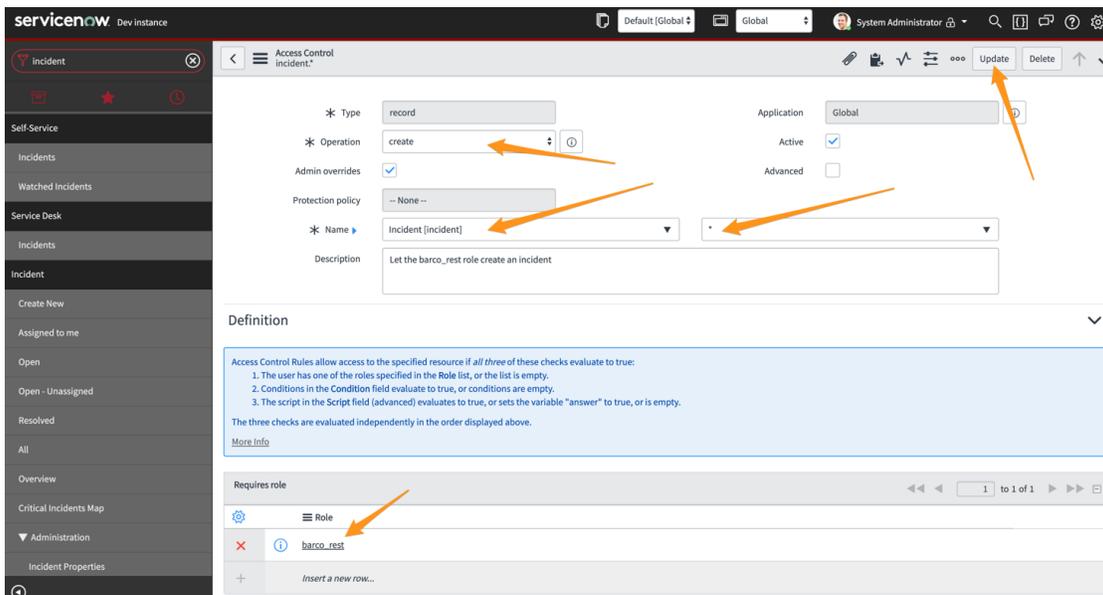




15.4.4.2 Create ACL for Incident Creation

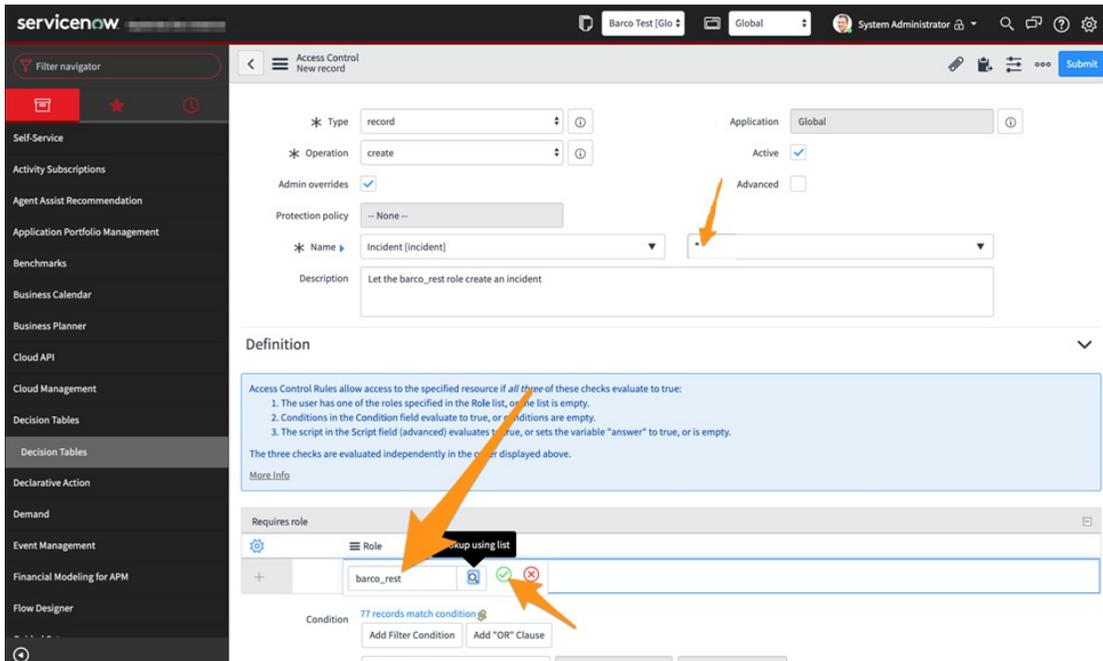
Once on the ACL list, click on New to create a new ACL for Incident. Enter the data as shown below. Make sure that the selected name is "Incident" and the operation is "create". Right click on the top of the form (grey section) and select save to save in place.

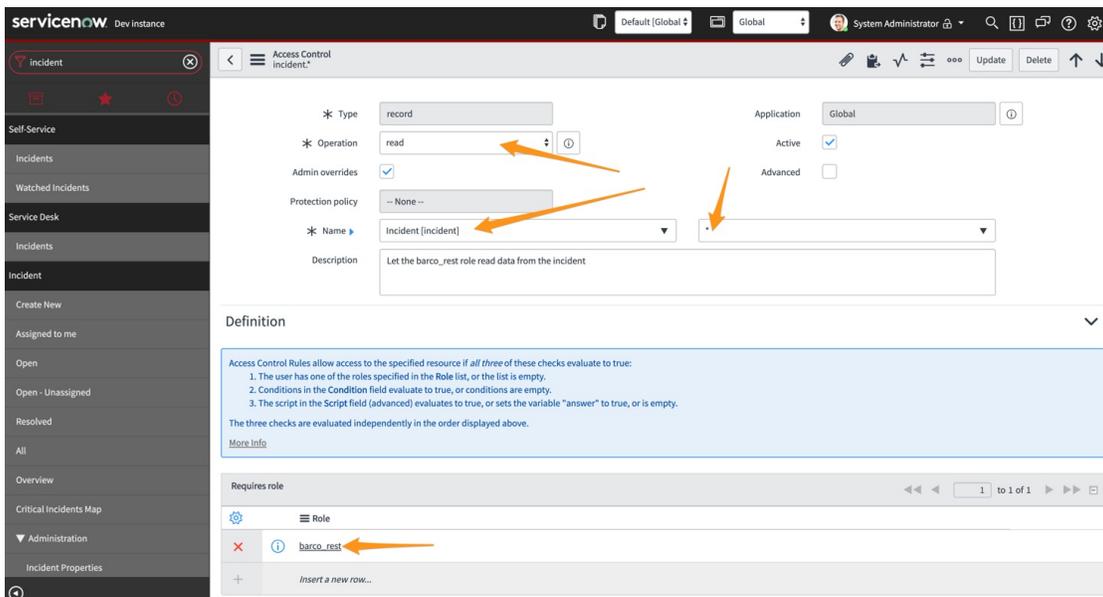




15.4.4.3 Add the role to the incident ACL

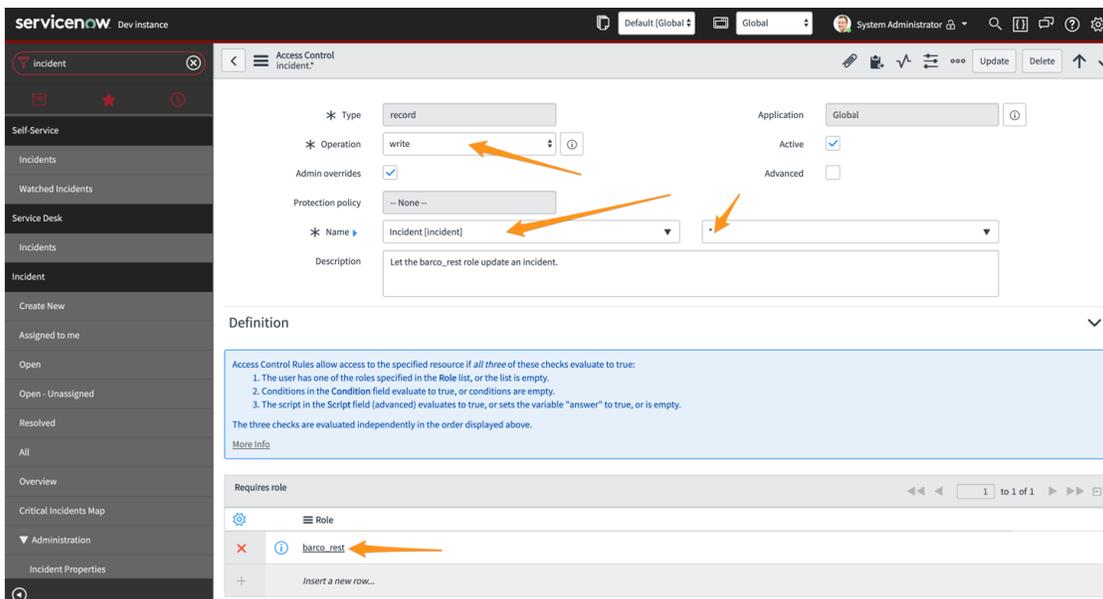
Once the create ACL is saved, add the Barco role to it by searching for the role and adding it to the ACL. Click update or submit at the top of the form to save the ACL and exit back to the list. You will need to create 2 more ACLs following the same process for the Read operation and for the Write operation. Do not forget to add the Barco role to each of them.





15.4.4.4 ACL for write operation

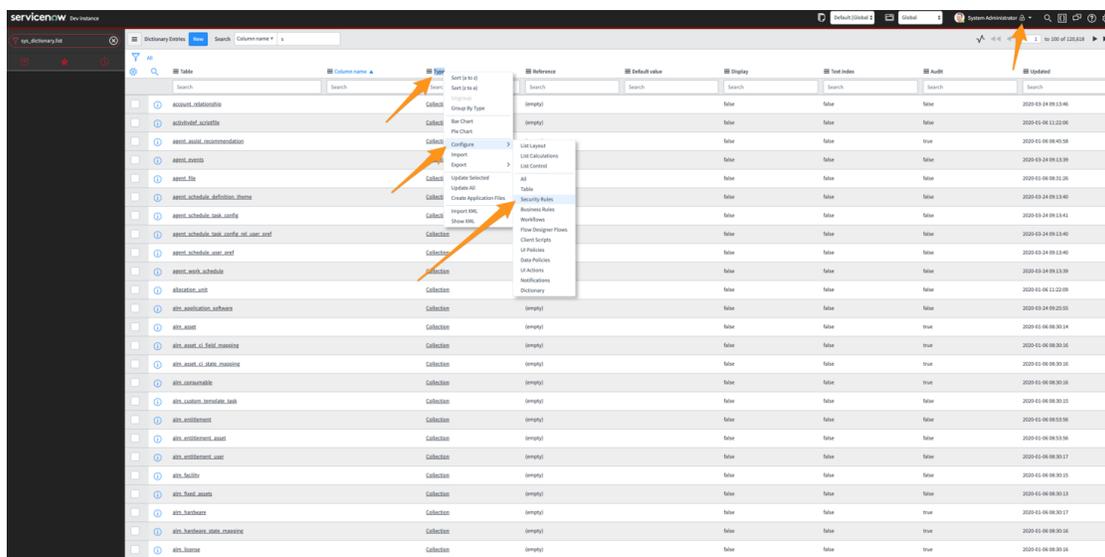
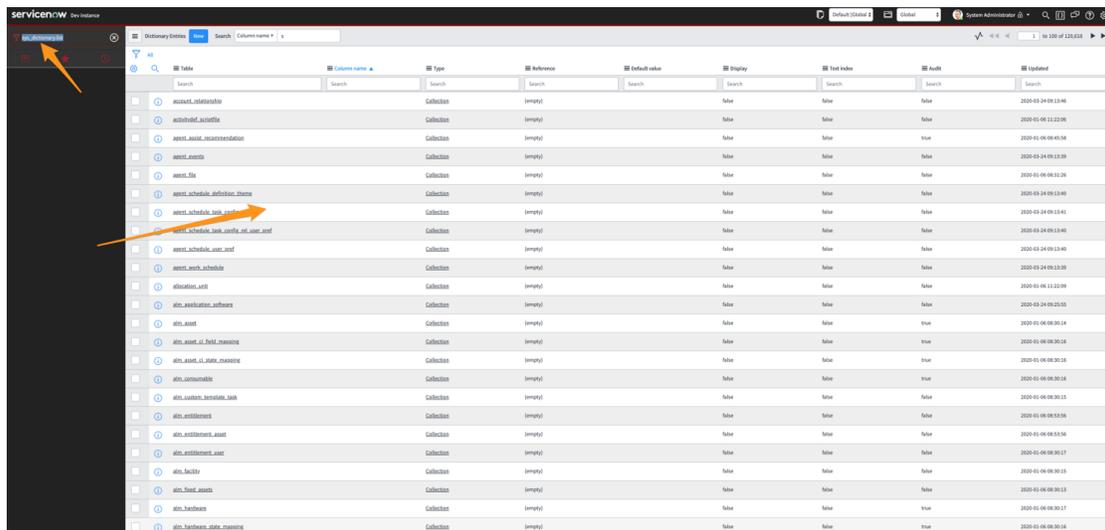
After the creation of the last ACL for the Incident table you should be able to run the table API for Incident and [Sys_Choices](#).



15.4.5 Dictionary Access Controls

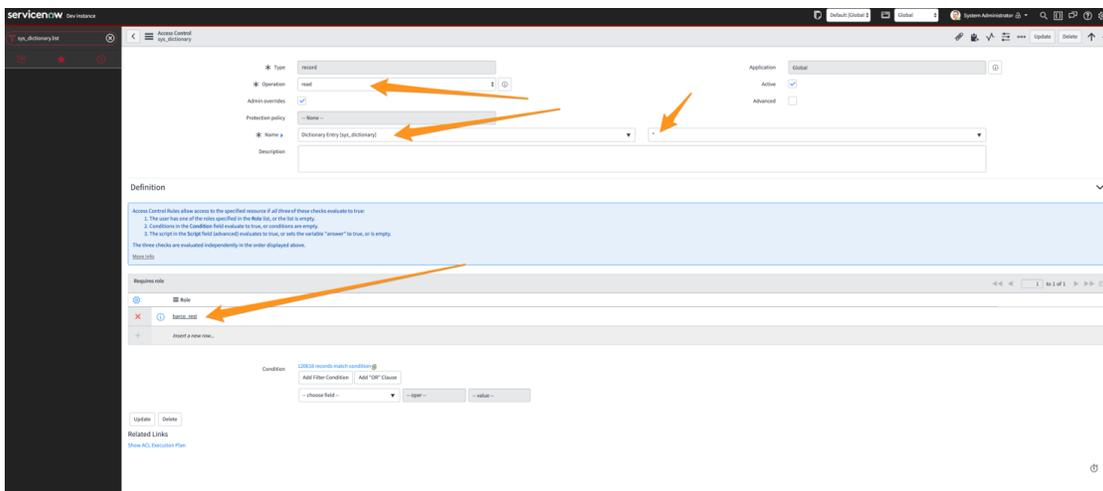
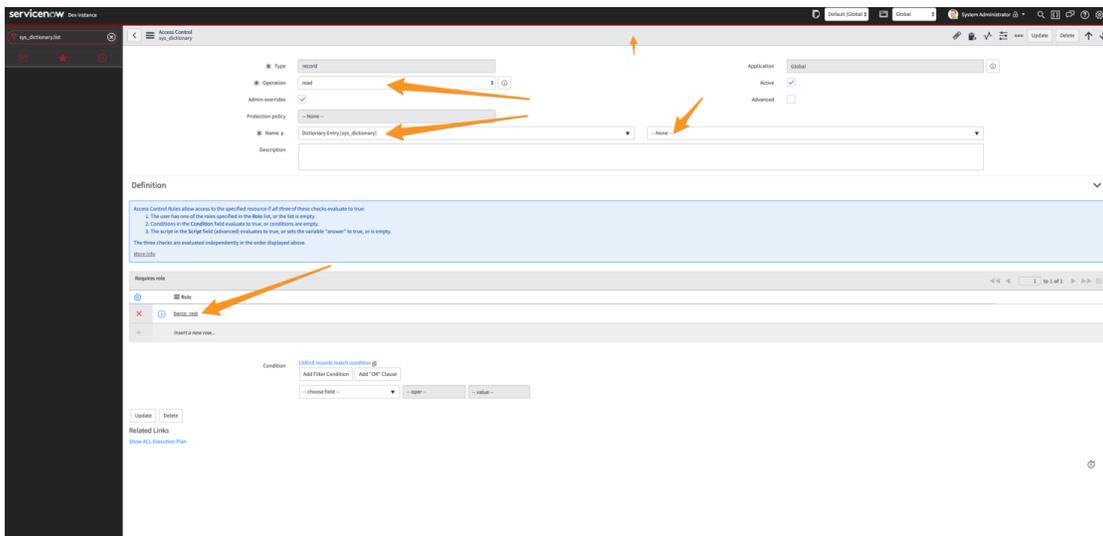
15.4.5.1 Navigate to the dictionary list

Navigate to the Dictionary list and right click on the title of a column. Select configure then Security Rules to get the list of ACL related to the Dictionary table. Make sure that you are still [security_admin](#) (see [Elevate admin role](#)) before doing so.



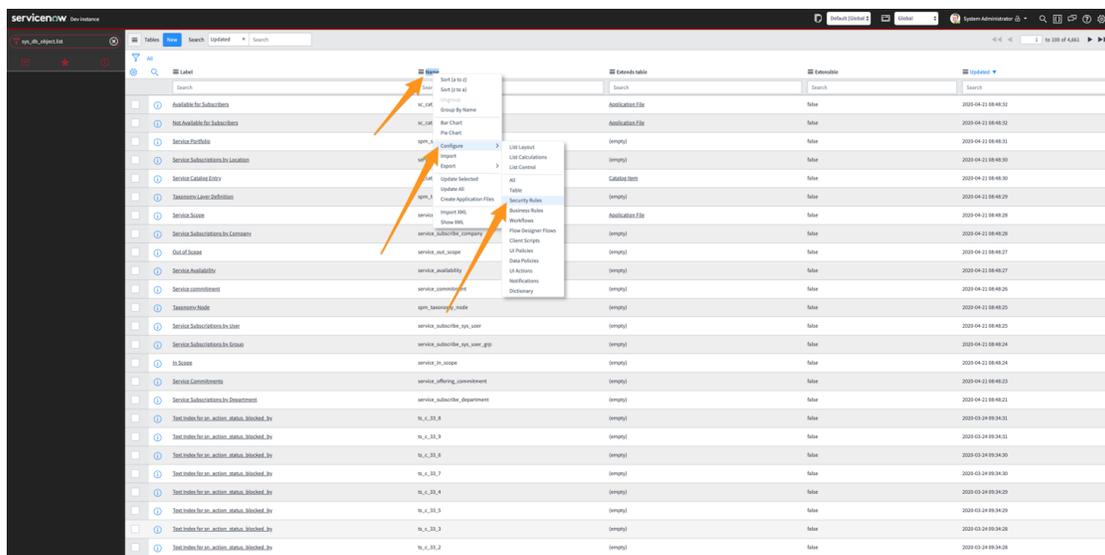
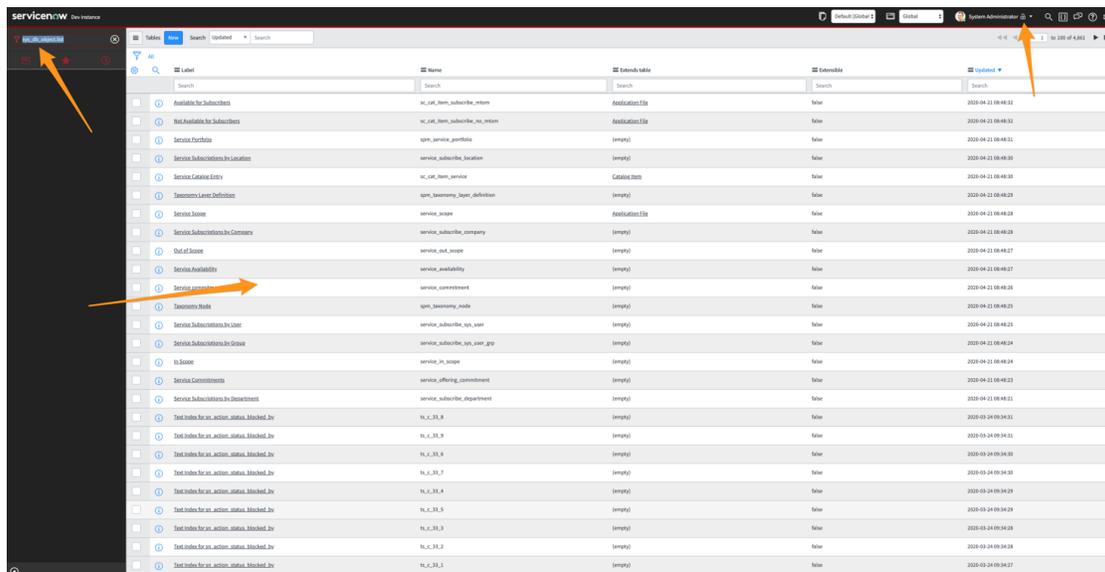
15.4.5.2 Create ACL for dictionary read

Once on the ACL list, click on **New** to create a new ACL for Incident. Enter the data as shown below. Make sure that the selected name is **sys_dictionary** and the operation is **read**. Add the **barco_rest** role (created on **Role Configuration**) to the ACL. Right click on the top of the form (grey section) and select **save** to save in place. You need to create two **READ** ACL entries one with the field selection set to **none** and the other one set to *****.



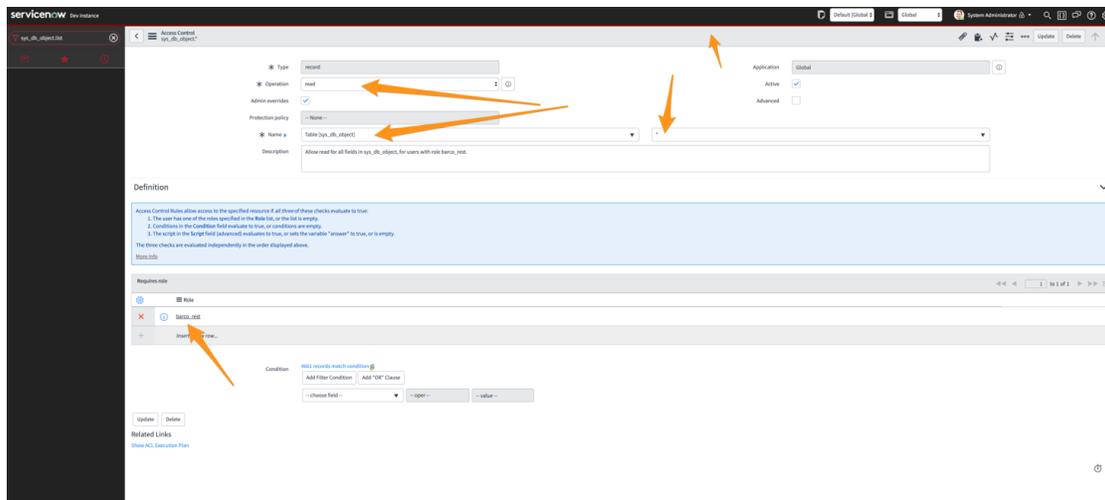
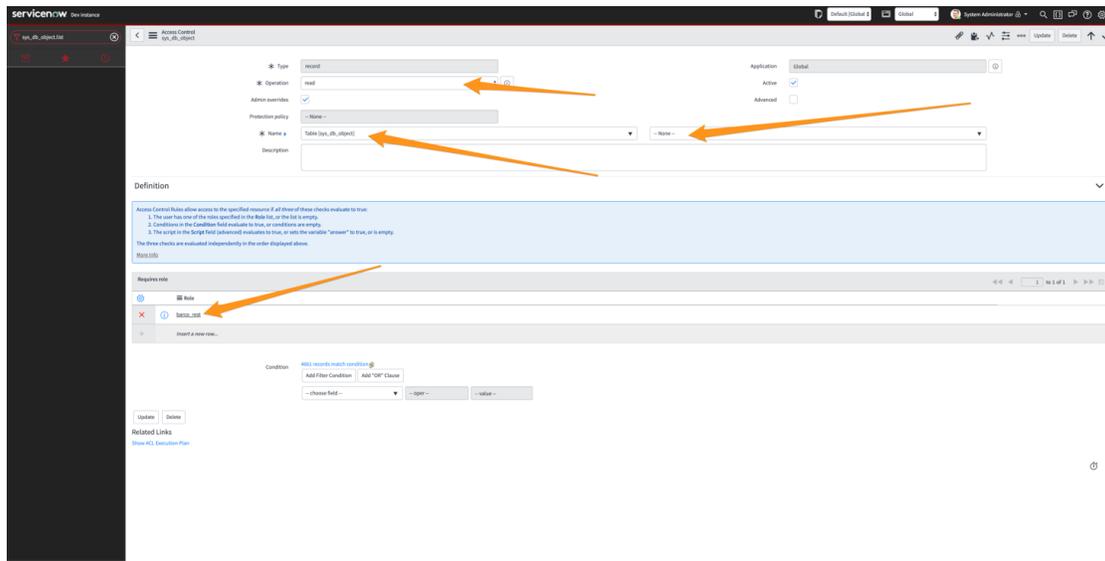
15.4.5.3 Navigate to the DB Object list

Navigate to the DB Object list and right click on the title of a column. Select configure then Security Rules to get the list of ACL related to the DB Object table. Make sure that you are still [security_admin](#) (see [Elevate admin role](#)) before doing so.



15.4.5.4 Create ACL for db object read

Once on the ACL list, click on New to create a new ACL for DB OBJECT. Enter the data as shown below. Make sure that the selected name is **sys_db_object** and the operation is **read**. Add the **barco_rest** role (created on [Role Configuration](#)) to the ACL. Right click on the top of the form (grey section) and select save to save in place. You need to create two READ ACL entries one with the field selection set to **none** and the other one set to *****.

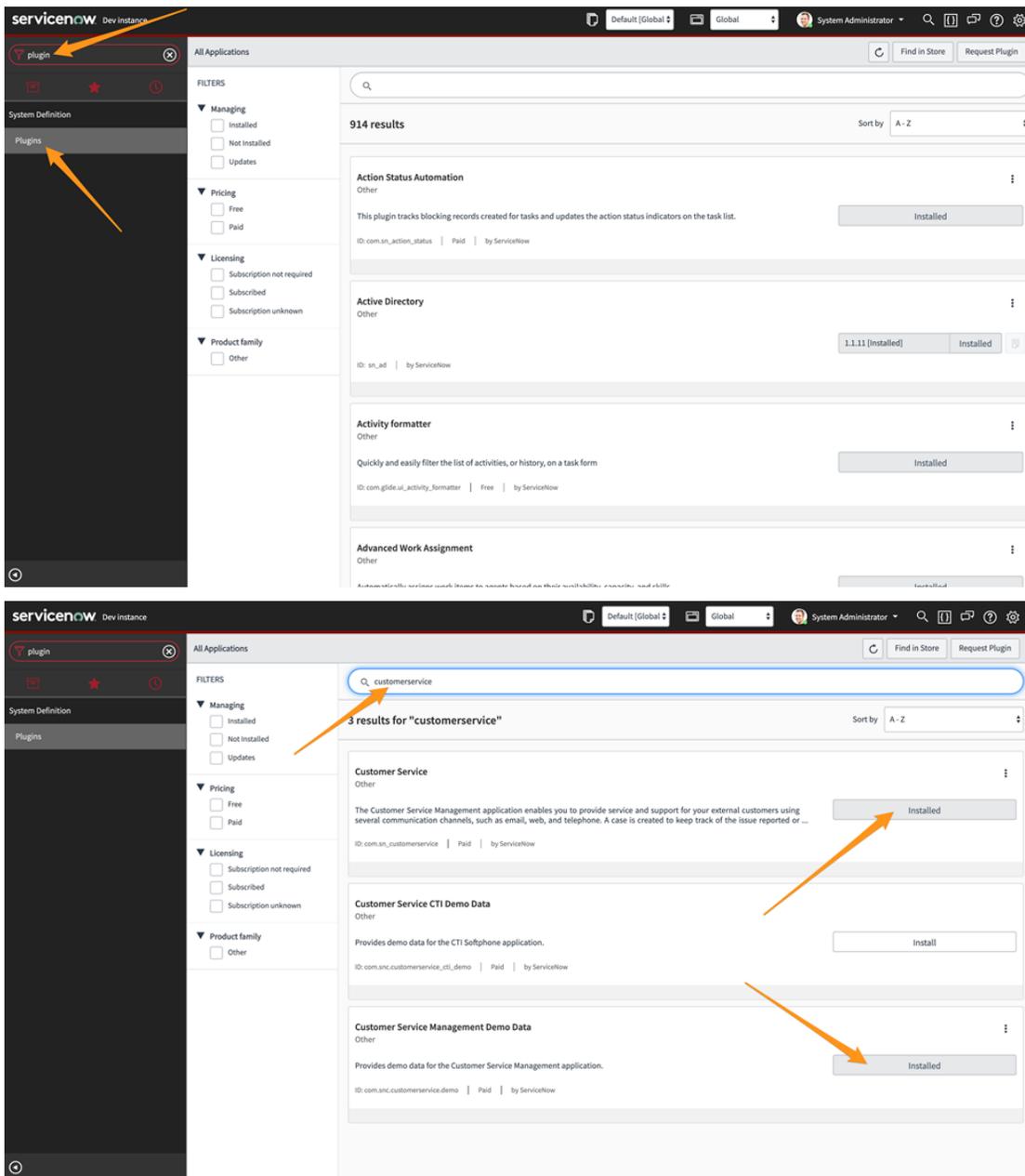


15.5 Configuration for Case Management

15.5.1 Install CSM on your instance

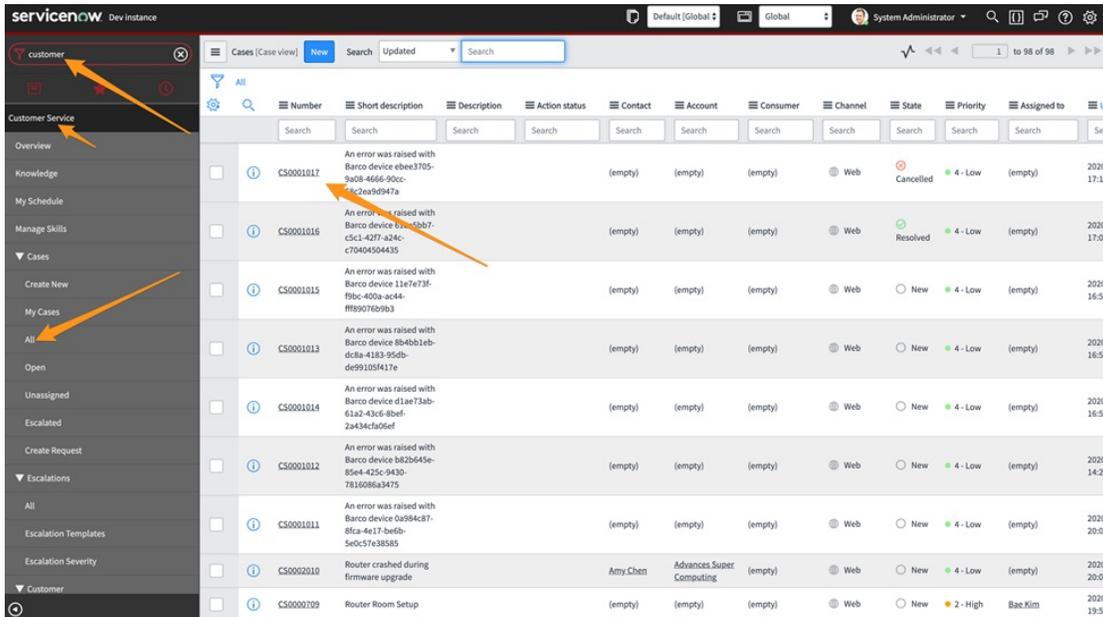
15.5.1.1 The Plugins screen

The Customer Service Management is a separate module that need to be installed manually on a ServiceNow instance. To install it, navigate to the plugin list and search for the [customerservice](#) plugins. Install the Customer Service (required) and the Customer Service Demo data (optionally). Just click the install button and confirm to install. It will take a bit of time to install.



15.5.1.2 Validate and Navigate to Cases

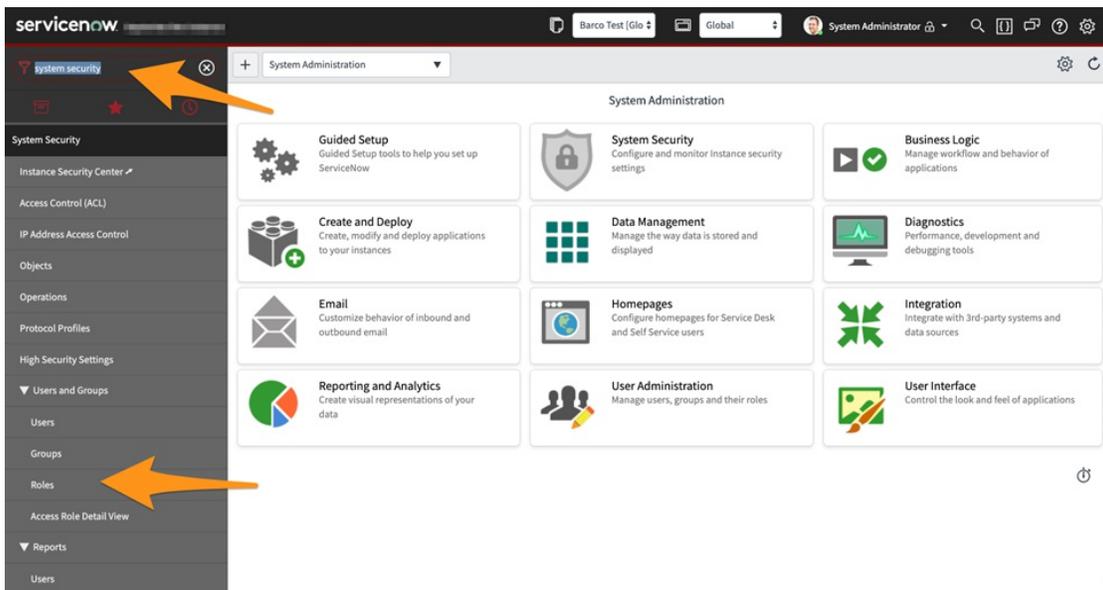
To validate if the installation as completed successfully you can navigate to the Customer Service application menu and click on the **All** case list to view the cases.

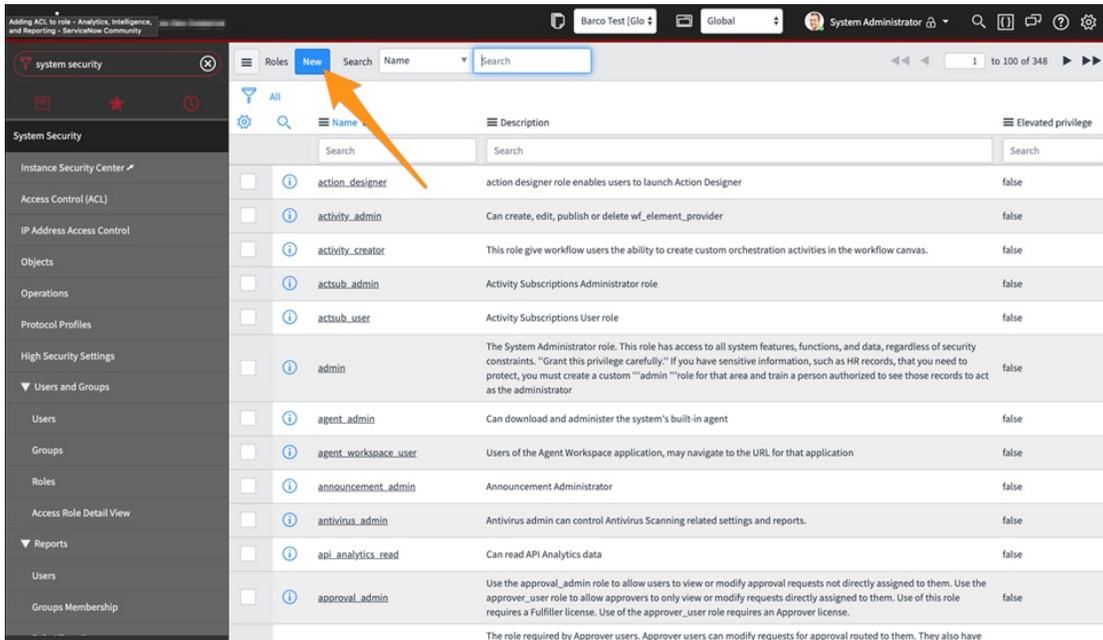


15.5.2 Role Configuration

15.5.2.1 Role list

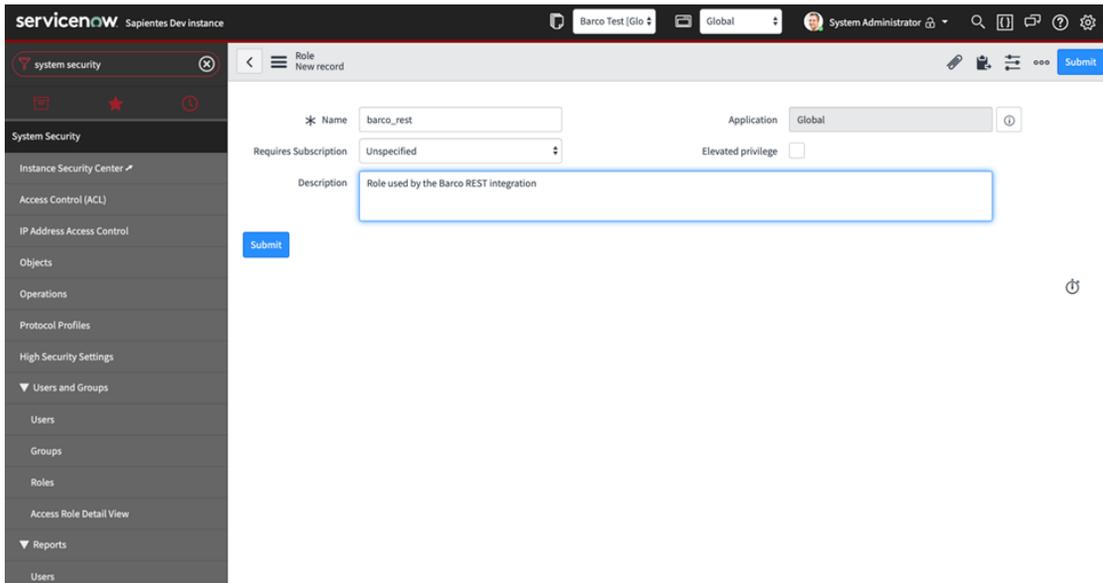
Go to the Role list in ServiceNow by using the Navigation filter to find 'System Security' and click on the Roles menu item. Once on the role list, click on the New button to create a new role.

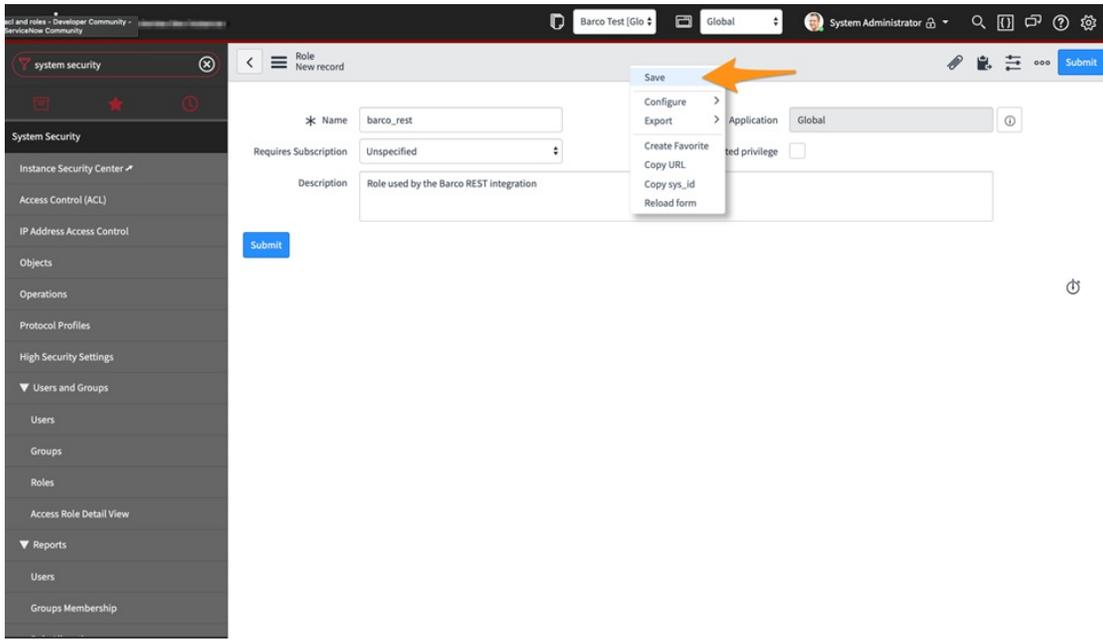




15.5.2.2 Create a role

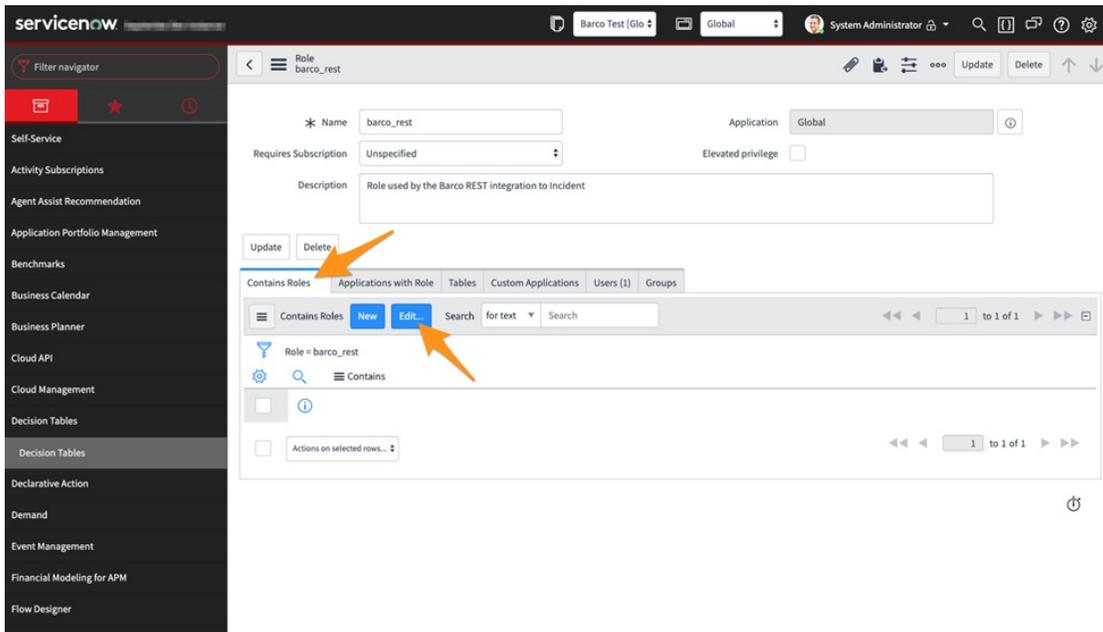
Enter the role information as shown. The role name can be changed to match Barco naming convention if it has no space. Once done, right click on the top of the form (grey bar) and select Save to save the record and keep it open.

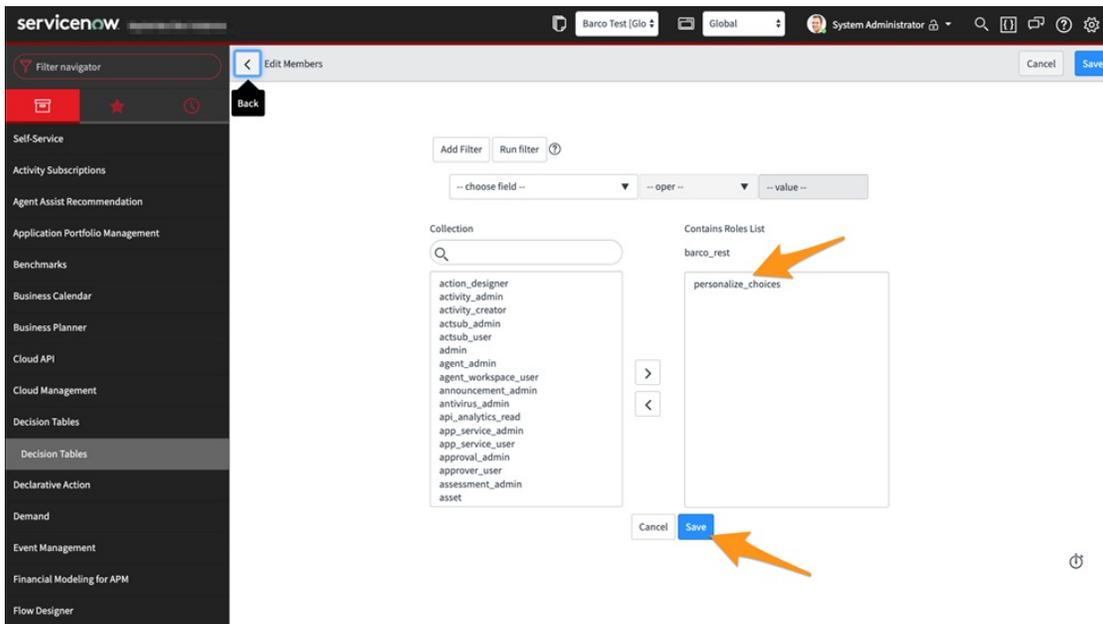




15.5.2.3 Add roles to Barco role

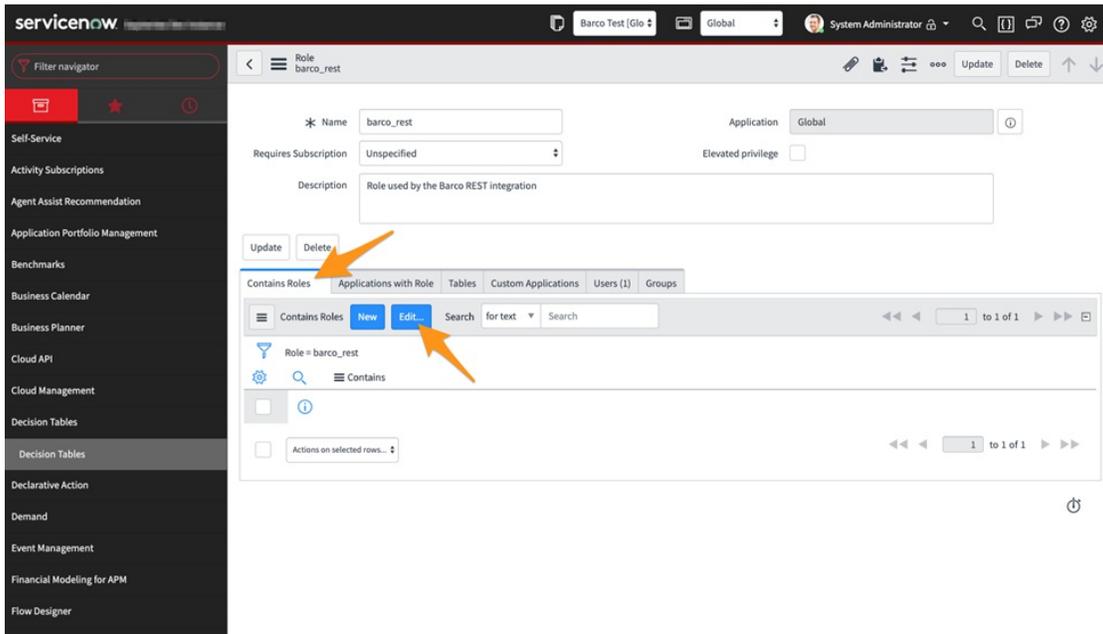
Select the Contains Roles and click on edit to add the role require to be able to use the `sys_choices` table. Add the `personalize_choices` role and the `itil` role then click save. The roles should now appear in the Contains Roles tab list. Click update at the top right of the form to update the Barco role and exit the form. The role will be used to create the ACL and it will be the role assigned to the Barco user.

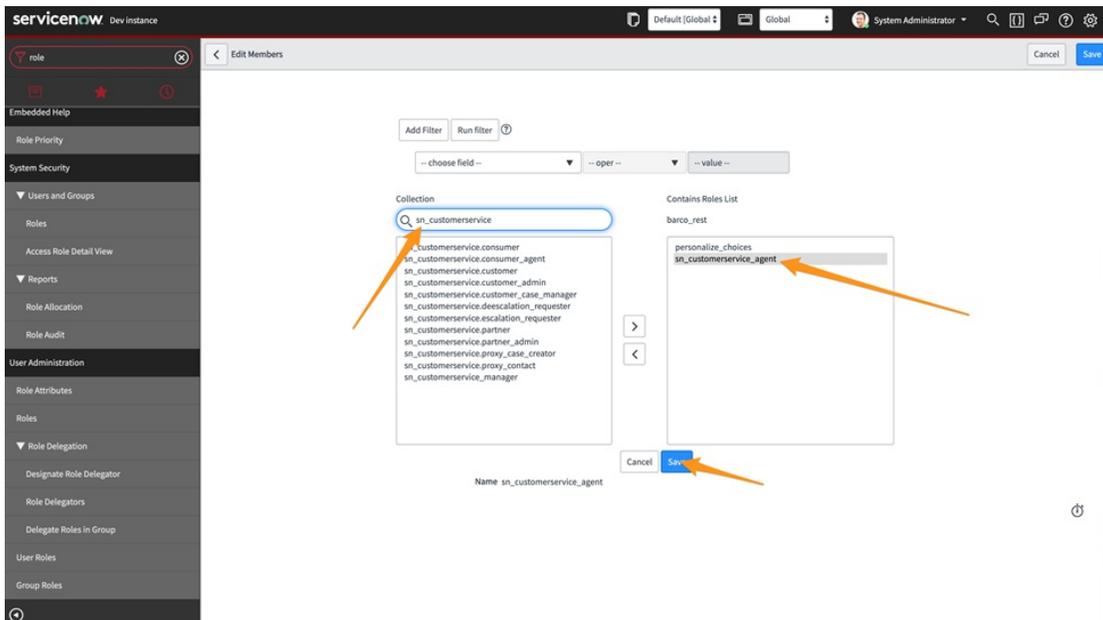




15.5.2.4 Add a role for Case if required

To be able to create some Customer Service Cases (instead of Incident) in the ServiceNow instance, just add the [sn_customerservice_agent](#) role to the Barco role so the user with the Barco role will be able to Create, Update and Read the Customer Case in ServiceNow through the APIs.

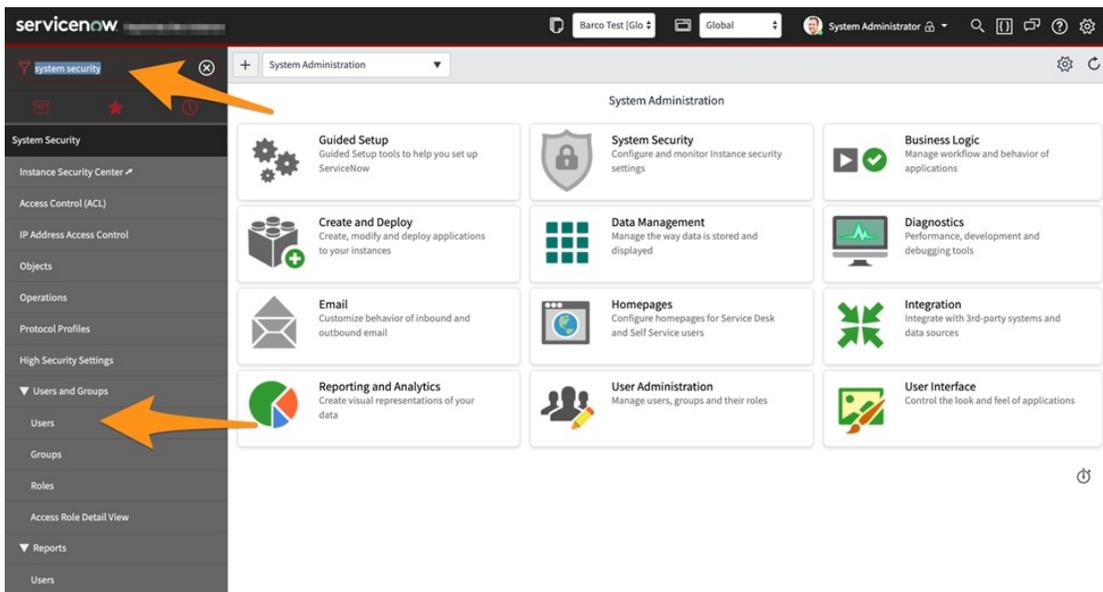




15.5.3 User Configuration

15.5.3.1 User list

Go to the User list in ServiceNow by using the Navigation filter to find 'System Security' and click on the Users menu item. Once on the User list, click on the New button to create a new user.



User ID	Name	Email	Active	Created	Updated
admin	System Administrator	admin@example.com	true	2007-07-03 11:48:47	2020-03-07 11:10:50
mospina	Mario Spina	mario@sapientesgroup.com	true	2020-01-25 11:09:04	2020-03-07 11:09:42
barco_rest_user	Barco Rest User		true	2020-03-05 03:10:14	2020-03-05 03:24:08
application_portfolio.analyst	Application Portfolio Analyst		true	2020-02-18 04:35:38	2020-02-18 04:35:51
anoplication_portfolio.admin	Application Portfolio Administrator		true	2020-02-18 04:35:22	2020-02-18 04:35:22
application_portfolio.user	Application Portfolio User		true	2020-02-18 04:35:01	2020-02-18 04:35:01
MIDProcess	MID Process		true	2020-01-12 14:51:43	2020-01-25 10:13:32
katina.servant	Katina Servant	katina.servant@example.com	true	2012-02-17 19:04:52	2020-01-06 08:29:47
katina.ramano	Katina Ramano	katina.ramano@example.com	true	2012-02-17 19:04:52	2020-01-06 08:29:47
isabell.armout	Isabell Armout	isabellarmout@example.com	true	2012-02-17 19:04:53	2020-01-06 08:29:47

15.5.3.2 Create a User

Enter the user information as shown. The user name can be changed to match Barco naming convention if it has no space. Once done, right click on the top of the form (grey bar) and select Save to save the record and keep it open.

User ID barco_rest_user **Email** barco_rest@barco.com

First name Barco **Calendar integration** -- None --

Last name Rest User **Time zone** Canada/Eastern

Title **Business phone**

Department **Mobile phone**

Password ***** **Photo** Click to add...

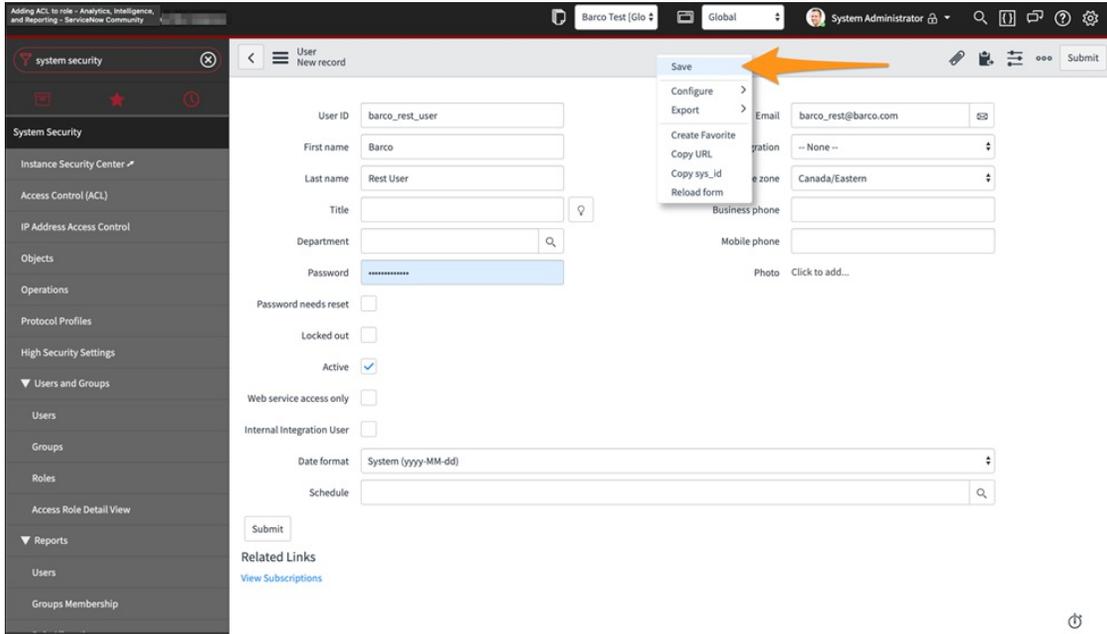
Password needs reset
 Locked out
 Active
 Web service access only
 Internal Integration User

Date format System (yyyy-MM-dd)

Schedule

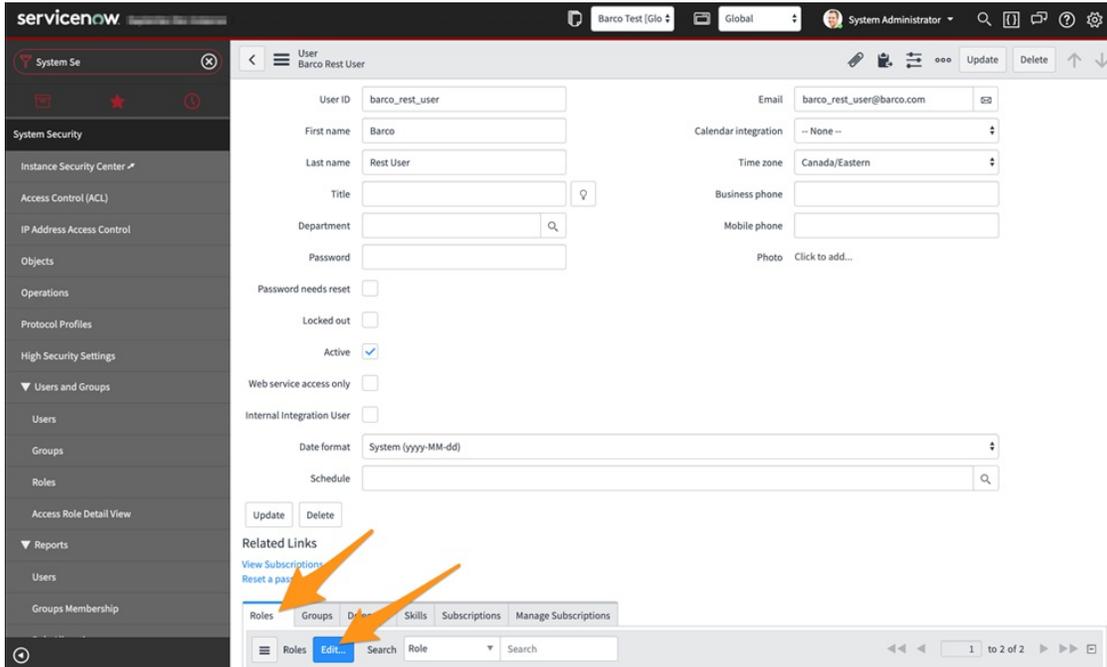
Submit

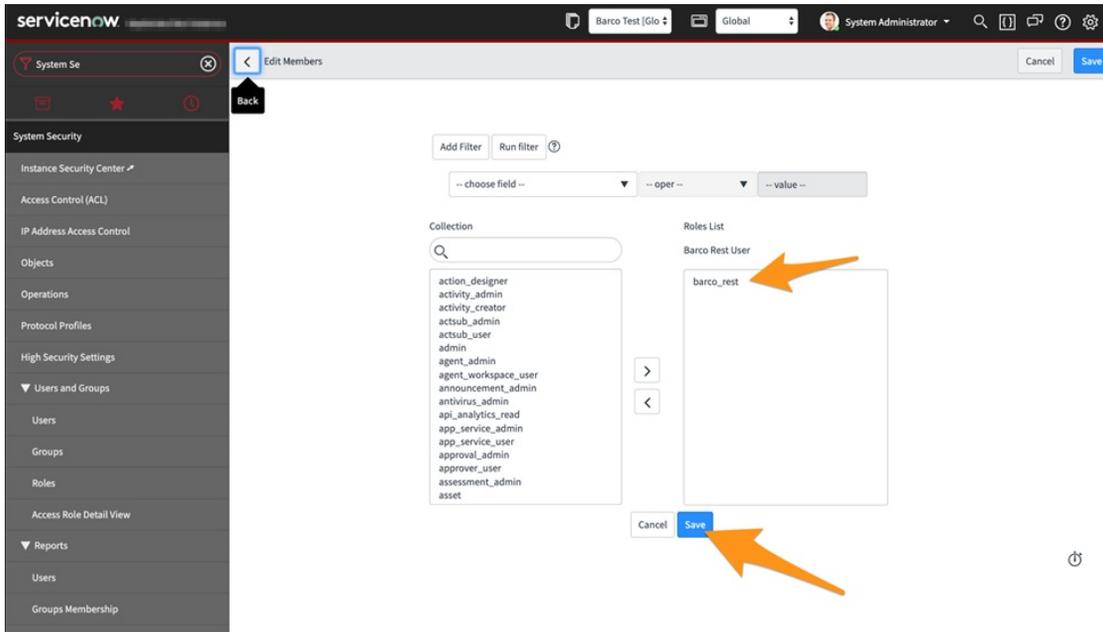
Related Links
View Subscriptions



15.5.3.3 Add role to user

In the record for the Barco user, find the Roles tab in the bottom section of the form. Click on Edit to add the Barco role to the Barco user. Search for the Barco role and add it to the Role list column. Click on save once done. Validate that 2 roles are now in the list under the Roles tab. One for the Barco role and one for the [personalize_choices](#).

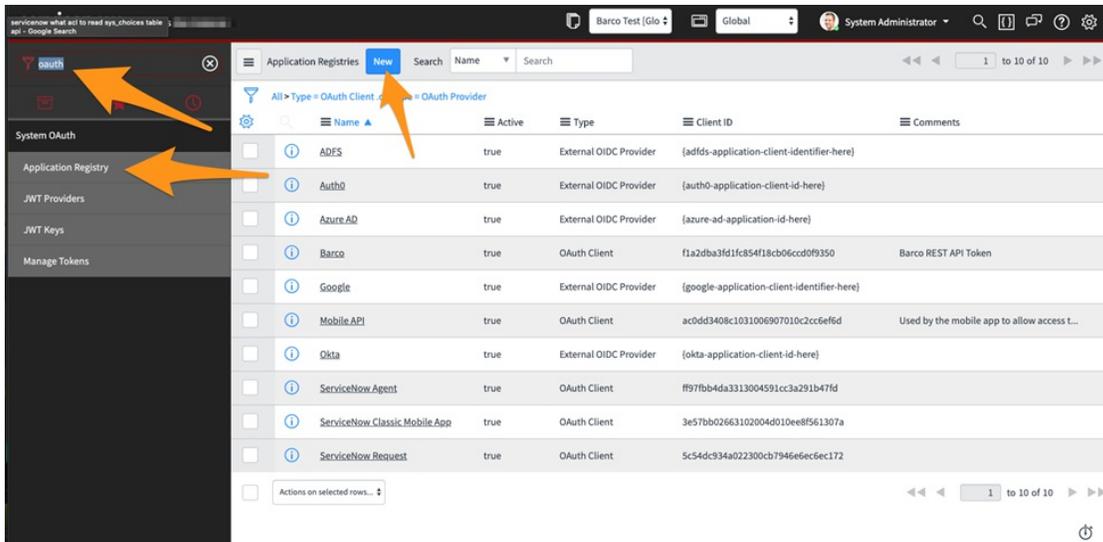




15.5.4 oAUTH Configuration

15.5.4.1 Create an application registry

To be able to use oAuth v 2.0 with the Table API, you need to create an Application Registry. Look for oAuth in the navigation filter, select application registry in the menu and click New to create a new one. Entered the data as shown and write down the Client ID and Client Secret for the REST API calls, click submit to create the entry. The refresh and token lifespans are in seconds.



The screenshot shows the ServiceNow interface for configuring an OAuth client application. The page title is 'Application Registries Barco'. A blue information box at the top provides details: 'Name: A unique name.', 'Client ID: Client ID automatically generated by ServiceNow OAuth server.', 'Client Secret: Client secret for the OAuth application. Leave it empty for auto-generation.', 'Refresh Token Lifespan: Time in seconds the Refresh Token will be valid.', 'Access Token Lifespan: Time in seconds the Access Token will be valid.', and 'Redirect URL: The redirect URLs authorization server redirect to. They must be absolute URLs and they are comma separated.' Below this, the configuration form includes fields for Name (Barco), Client ID (fla2dba3fd1fc854f18cb06ccd0f9350), Client Secret (masked), Redirect URL, Logo URL, PKCE required (unchecked), Refresh Token Lifespan (8,640,000), Access Token Lifespan (1,800), and Comments (Barco REST API Token). Buttons for 'Update' and 'Delete' are at the bottom left. A sidebar on the left contains navigation options like 'System OAuth', 'Application Registry', 'JWT Providers', 'JWT Keys', and 'Manage Tokens'. Two orange arrows point to the 'Name' and 'Client ID' fields.

