OSRAM

Light is OSRAM

Our Brand

e:cue

e:cue SYMPHOLIGHT 5

User Manual

Table of Contents

1   Preface

1.1   About this manual

With version 5, SYMPHOLIGHT has become a very powerful and full-featured tool. As with every complex tool, every documentation has to decide if it wants to show any microscopic feature and function, or to guide the user through the workflow and give him or her a basic understanding. We decided to take the second approach. To make yourself familiar with SYMPHOLIGHT, either install and discover what SYMPHOLIGHT does and is able to provide. Help text and explanations, as well as mouse-over help will tell you what functions are available. The other way is to read the manual and trying the various features. The best way will be a way in between.
This manual is not intended to explain every drop-down menu and every context menu. Instead, it explains the different parts and windows in SYMPHOLIGHT, and there is always a chance to find even more features. It is a basic guide line for SYMPHOLIGHT to help you to understand the main features and what to do with them. If you have questions or comments, please visit our forums on our website to get additional support or help.

1.2   How to read this manual

The manual is divided into three main sections. The first section gives a general overview how to install SYMPHOLIGHT and prerequisites as an overview. The second section shows basic principles. The third section describes in detail the single areas and feature sets.

1.3   Special markup

Preconditions and critical functions are shown like this (example):

> ⚠ Change the Windows password within six weeks after installation.
> Otherwise the user account will be locked and cannot be unlocked.

Useful hints, short-cuts and functions are shown like this (example):

> 💡 You can zoom in and out with the scroll wheel of your mouse. Keeping the
> mouse wheel pressed, will zoom in steps of 10 percent.

In the whole manual, all keywords describing key elements of SYMPHOLIGHT will be written with capitals. Examples: Properties Window, Workflow Designer or Media Library.

1.4   Special notes

> 💡 Throughout the manual, the term "multi-select" is used. This means the possibility in Windows to select groups or sequences of elements by keeping the Ctrl or SHIFT key pressed to select elements in a list. See the Windows documentation for a description.

> 💡 There is also a term "long-click" used. A long-click are two mouse clicks on an object on two different positions. This makes the object editable, e. g. to change the name of an element or a value.

1.5   Online help functions

When editing properties in the Properties Window, a help text and a description is displayed in the lower part of the Properties window.

Mutual Exclude Collection ID	0
Priority	1
Start Position	+0.0:0:0.0

Priority
(Integer)

Render Priority. Sequences with higher priority are overwrite values of sequences with lower priority.

1.6   Common time format in SYMPHOLIGHT

SYMPHOLIGHT uses time values in the format dd.hh:mm:ss:ms, where dd = days, hh = hours, mm = minutes, ss = seconds and ms = milliseconds.

2   Introduction

2.1   Digital Lighting Control

Over the last decades digital systems replaced conventional analog lighting control instruments. However, this is not the only change in lighting technology. The story began with single RGB elements like pars and flood lights, developed over moving heads and more complex fixture types. Today, media LED solutions with thousands, if not ten-thousands of RGB pixels became a standard.

e:cue released its first version of the Lighting Application Suite (in short: LAS) in 2002. The LAS was an application suite to setup a configuration with engines, fixtures and everything else to have a physical representation of the installation. Later you can create illuminations, light projects with effects and much more to run on your selection of fixtures. The LAS was the central application for creating light projects and controlling

complex lighting projects, and to control your network and all devices in it. In its life cycle the LAS began in the world of some cuelists with triggers and actions and grew into the world of media displays and complex effects.

The overall development of lighting went from fixture-oriented to media-oriented applications, as mentioned before with high resolution media walls and very complex solutions. Today configurations with hundreds of DMX universes are no exception today, they are daily business for Traxon and e:cue. For this reason, the LAS needed a successor that reflects these new world of lighting control. This is e:cue SYMPHOLIGHT1.

2.2   About SYMPHOLIGHT

SYMPHOLIGHT is a software application to design, test and program medium to very complex lighting designs and lighting applications.
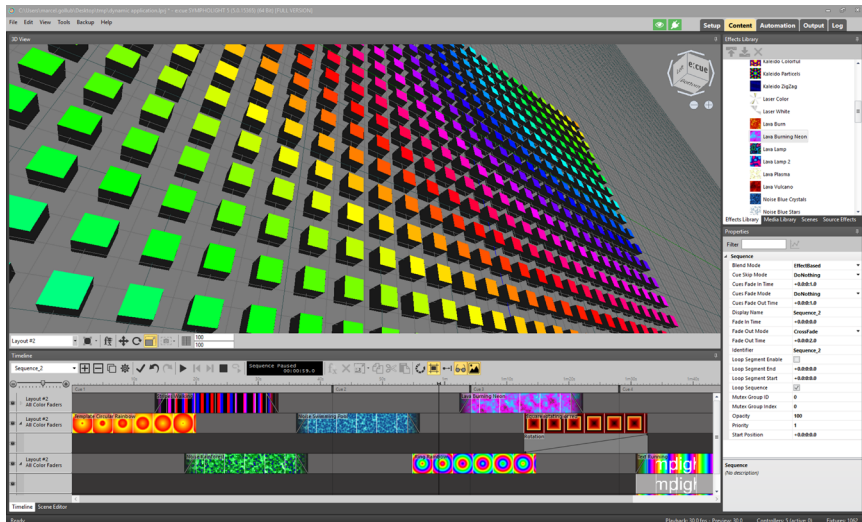
SYMPHOLIGHT creates a single application for the complete cycle of a project, from setting up fixtures and interfaces ("patching"), design and test the project and execute and control the project in real-time.

Main features include:

— Dynamic lighting with media and effects and static lighting with Scenes.
— Timeline design.
— Layer concept for media like images and videos, and effects.
— Built-in effect library with color and intensity effects including fades, channel effects and more.
— Built-in fixture library with generic and vendor-specific fixtures.

1 Whenever SYMPHOLIGHT is written, e:cue SYMPHOLIGHT is meant.

— A flexible and powerful Workflow Designer for automation and control.

— Separation between design and execution state for flexibility and stability.

— Separated preview of projects in design and execution.

— Extendable with C# programming language.

3   System Prerequisites

For running SYMPHOLIGHT the system must meet the following specifications:

Minimum configuration for all sizes:

    Microsoft® Windows® 10 32 Bit

    Internet Explorer 10 or better

    8 GB RAM

    A minimum of 20 GB hard disk space

    Dual-Core processor or better. Quad-Core processor is recommended.

    DirectX 9.0c supported graphics adapter

    A minimum of one Ethernet interface for connections to engines and interfaces

Recommend configuration for small to medium projects (~50.000 pixels)

    Quad-Core desktop processor (e.g., Intel® Core i5)

    nVidia® GeForce® GTX 760 class graphics adapter

    Second Ethernet adapter dedicated to e:net

Recommend configuration for large projects (200.000+ pixels)

    Microsoft® Windows® 10 64 Bit

    16 GB RAM

    Quad-Core high-end desktop processor (e.g., Intel Core i7)

    nVidia GeForce GTX 780 class graphics adapter

    Second Ethernet adapter dedicated to e:net

4   Installation

On the e:cue Lighting Control Engine 2 SYMPHOLIGHT is pre-installed.

> Please make sure the installed SYMPHOLIGHT Version on your Lighting Control Engine 2 is the latest version. Otherwise proceed as follows below.

On all other systems proceed as follows:

— You must have administrator rights to install e:cue SYMPHOLIGHT on Windows.

— Insert a CD/DVD with the SYMPHOLIGHT installer into the drive or plug in an USB stick with the SYMPHOLIGHT installer.

— Open the Windows Explorer, navigate to the CD/DVD drive or USB drive.
— Double-click on the installer, the installer will ask for permission and acceptance of the copyright conditions.
— Select the installation language.
— After the installer has finished, click OK.

The installer places an optional short-cut to SYMPHOLIGHT on the desktop. Double-click the icon to start SYMPHOLIGHT. If application updates are available, SYMPHOLIGHT will download and install updates.

⚠ The Windows Activation Code for the LCE2 is on the housing behind the front panel. After starting the LCE2, do not shut down the system until the desktop shows. Change the Windows password within six weeks after installation. Otherwise the user account will be locked and cannot be unlocked!

4.1   Trial and full version mode

SYMPHOLIGHT requires a protection dongle to run in full mode. This dongle is part of the delivery when you purchase SYMPHOLIGHT as separate software. In the Lighting Control Engines, the dongle is part of the server system and already built-in on the main board.
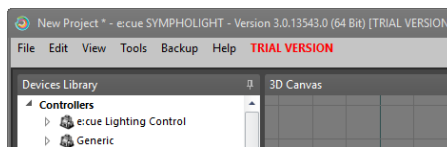
Without a dongle, SYMPHOLIGHT runs in trial mode with full functionality, but without generating any output, neither DMX or for relay or serial Nodes. If you forgot to plug in the protection dongle while stating SYMPHOLIGHT, plug it in while SYMPHOLIGHT runs and it will switch to full mode with generated output data.
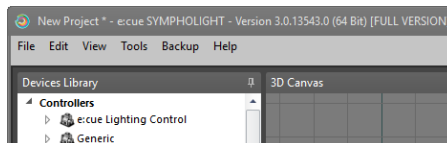
All existing LAS Programmer Enterprise users with a LAS Enterprise dongle can use SYMPHOLIGHT`s full version without additional costs. The SYMPL Core S can also be used as a license key for the SYMPHOLIGHT software. Please refer to the Core S documentation for more information [Chapter 14: SYMPL Core engines].

4.2   Automatic updates

On start-up, SYMPHOLIGHT is able to find and install updates automatically. This requires an active Internet connection during start-up. The *check for updates* can be enabled or disabled in the application settings from the main menu.
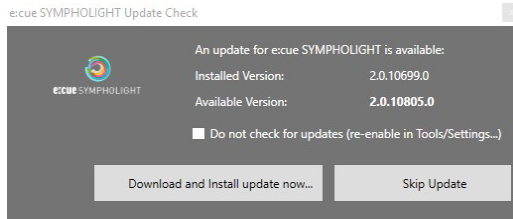
After downloading the update, the update will be installed. This again requires administrator permissions on the system.
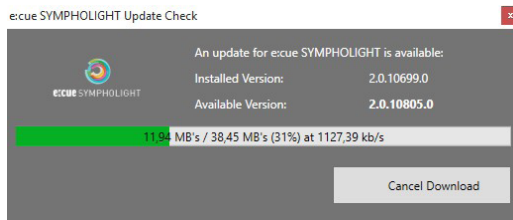
4.3   Keyboard shortcuts

As many users like to work with keyboard shortcuts instead of clicking menu entries, SYMPHOLIGHT supports a predefined set of keyboard commands. To add keyboard shortcuts or edit the existing definitions, use Tools | Configure keyboard shortcuts from the main menu.
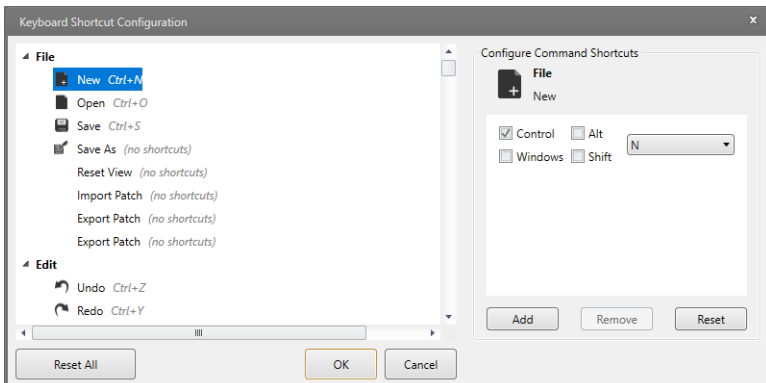
5   Release Notes

This is an index for changes in SYMPHOLIGHT 5 compared to the SYMPHOLIGHT 3.0
Service Release 6.

5.1   General
— Initial support for Chinese localization.
— The 3D canvas can be disabled (muted) automatically when it is not needed, e.g.
   when the commissioning is done.
— New top-level buttons to mute the e:net communication (see below) and to prevent
   the interruption of the live DMX output from the Output tab, when the user switches to
   the Content tab or Setup tab.
— The integrated REST Server allows controlling and monitoring SYMPHOLIGHT
   instances over a secure HTTPS connection from a remote system.
— The REST API is browsable via the integrated Swagger web application. Detailed
   documentation for each REST request is available there.
— The Scripting API has been enhanced and simplified. More documentation is
   available, both offline as well as when typing. Debugging the scripts is easier.
   Devices, RDM, custom DMX, Content Management, Workflow blocks and system
   information can now be accessed from Scripting.
— Scripts can now consist of more than one class and each class can consist of more
   than one method accessible from the Workflow Designer.
— Added support for the new e:cue LCE3.
— Added a button to clear the log view.
— e:bus installations can now be monitored for troubleshooting.
— Added support for the new test mode of SYMPL nodes.
— The Fixture Library has been cleaned up. Commonly used devices like generic "RGB"
   now have clearer names and obsolete devices have been removed.
— The 3D canvas no longer shows a confusing depth when working with planar 2D
   layouts.
— Improved the performance of the DMX output (tested with up to 2100 DMX universes).

5.2   Setup
— Separated the views for: Controllers, SYMPHOLIGHT instances and layout elements
   like fixtures. This gives a better overview in bigger projects.
— A new dialog for adding and managing controllers offers more clarity and
   performance for commissioning in small and big projects. Users can add multiple
   controllers at once and many management tasks are made easily accessible.
— Mute the e:net access to controllers for any SYMPHOLIGHT or SYMPL Core S
   instances. This allows easy handover of controllers from one SYMPHOLIGHT or
   SYMPL Core S to another.

— Log files from a SYMPL Core S can be downloaded using the context menu in the SYMPHOLIGHT instances view.
— The enhanced SYMPL Core S project upload dialog allows a direct project upload and handover of e:net controllers in one step.

5.2.1   DALI

— Scanning and addressing of devices no longer happens automatically and unpredictably. Instead, the user has more control when maintaining the DALI installation.
— New global commands "Scan All", "Delete Device with Address Collisions and Rescan" for resolving collisions, "Address only unaddressed" for added devices and "Readdress all Devices".
— New per device commands "Check Device Address Collision", "Delete device" and "Modify Short Address".
— New status icons in the DALI view indicate busy and error conditions.
— Added settings to control the DALI bus in case the connection to the SYMPL dali node is interrupted.

5.3   Content

— Videos can now be imported in any resolution to provide higher quality needed in bigger projects or smaller videos e.g. for the SYMPL Core S.
— The new black level threshold sets the output to zero for pixels with a value below the threshold.

5.4   Automation

5.4.1   Workflow Designer

— More logical sorting and naming of Workflow blocks. This follows the basic flow "Triggers / Inputs -> Processing -> Output / Action". The property grid has been simplified as well.
— A new "Observe controller" block to observe e:net controllers allows maintenance and tracking e:net controller data e.g. on the ActionPad.
— The new system status block "Observe System" gives live access to many aspects of the running SYMPHOLIGHT instance. This is useful for maintenance, visualization on the ActionPad or reporting to a central server.
— The new blocks "Send to Network" and "Receive from Network" allow transferring data and events from configurable ports over the network. This saves the user from implementing boilerplate code in Scripting for the same task.
— Fading from one value to another is now implemented in a dedicated "Fade" block. This allows smooth and easily configurable fading. The "Fade" block supports different flavors of inputs like buttons, switches or Glass Touch devices.

— The new "Permit Event" block can disable Workflow events when a condition is satisfied.

— New "Build RDM command" and "Send RDM" blocks are implemented for building and sending many RDM commands. Possible targets are all devices (broadcast), a single universe or a single device. One out of many use cases is changing the personality at runtime.

— The new "Generic Control Sequence" block can start, pause or stop sequences by their Id. Also, start / pause / stop events from all sequences can be consumed at one place. Easy access to the Grand Master is also offered.

— The "Event To Value" block allows configuring multiple named events. When triggered, the named value is pushed to its output.

— The "Periodic Trigger" block now provides an interval resolution of 100ms. The smallest interval is now 100ms.

— The ActionPad page navigation in the Workflow Designer is now name based and no longer index based.

5.4.2 ActionPad Designer

— The ActionPad now provides a clear modern design, by default. The properties of the controls have been simplified.

— More customization options for the fader control allow detailed styling and different shapes.

6 Principles

6.1 Operational sequence

The procedures with SYMPHOLIGHT for most projects will follow these steps:
— Setup your DMX fixture configuration and addressing scheme. Distribute DMX universes and start addressing to the target settings of your DMX output engines.
— Place your DALI ballasts and input devices, e.g. sensors.
— This configuration scheme is called a patch. You can also import patch source files created with the Patchelor of the Lighting Application Suite from version 5.x and higher.
— Insert and setup your external engines and interfaces, e. g. SYMPL Core S, SYMPL dmx Nodes or SYMPL dali Nodes.
— Setup DMX universe distribution and scan your DALI lines, assign DALI addresses and create DALI groups.
— Create your show under Content. Import videos or images in various formats. All imported media will be converted to a final format and placed in the Media Library. This media can be dragged into the sequence. Or:
— Apply color and intensity effects as main content and drag them into the sequence. Create fades, overlays and other visual effects. The applied content can be previewed in real-time.
— Create automation and control in the Automation area with the Workflow Designer.
— Create an end-user control interface with the graphical user interface Action Pad.
— Publish your show and execute it.

6.2 Migration from Lighting Application Suite to SYMPHOLIGHT

If you are familiar with the e:cue Lighting Application Suite (LAS), SYMPHOLIGHT will represent a completely new interface and way of programming. The LAS was mainly cuelist-oriented.

SYMPHOLIGHT works with a timeline concept similar to the Sequencer and Emotion FX. In general it is possible to work with the concept of cues. This will be explained in a later chapter "Cue creation".

Setting up the fixture configuration is similar to the Patchelor. You can even import patch files (.pdoc) from the Patchelor into SYMPHOLIGHT and reuse your configuration. There are even more similarities with the LAS, like the Media Pool or the Fixture Library. It is not possible to use or import shows from the Programmer, as the concepts and formats are completely different.

An important change is the layer concept. The Sequencer of the LAS did not support a layer concept, the tracks were played in parallel. Also the Live FX window is now replaced by effects from the effects library directly. Also, the Workflow Designer of

SYMPHOLIGHT is much more powerful and flexible compared to the Programmer's Trigger Engine.

Scripting in the Programmer of the LAS was done with a C-like dialect. SYMPHOLIGHT also allows scripting, but now in C#. C# in SYMPHOLIGHT runs in a .NET environment, so SYMPHOLIGHT scripting is far more powerful than the C scripting in the Programmer of the Lighting Application Suite. Use the standard C# and .NET documentation to learn more about C#.

7 Application Concept

7.1 Setup

The first step is setting up your lighting configuration with fixtures and all other components and elements. You can select engines and interfaces like SYMPL Nodes or Butlers and add them to the project. As the next step, you select fixtures from the list on the left side. You can place single fixtures or arrays of fixtures like pixel boards. To represent your later configuration, fixtures can be bound to DMX universes and addresses.
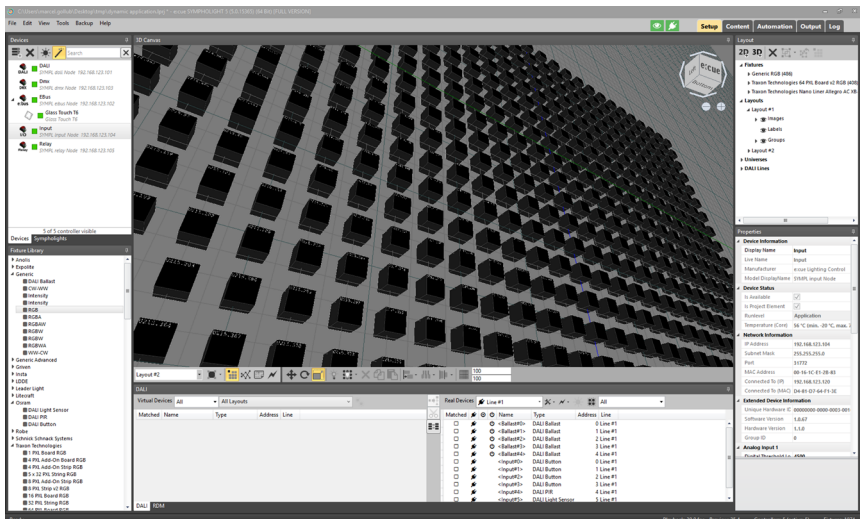
Also, fixtures can be grouped. You can put fixtures in one or more than one group. Grouping has a direct effect on the later content creation, as fixtures in one group have their own Layer in the Timeline. If you do not create any specific groups, all placed fixtures will belong to one Layer. SYMPHOLIGHT supports a 2D and a 3D model for

creating the Setup. The Setup Window also contains the RDM section, where RDM is supported.

7.2   Content

In the Content step, the show with media, effects and Scenes are created. This happens in the Content Window. Understanding the features and principles is important to use SYMPHOLIGHT effectively and efficient. Content is built from one or more Layers. Each Layer is comparable to layers in graphic or image editors. SYMPHOLIGHT works with a timeline and layers as the main control element. Nevertheless, static content is still possible as Scenes and will be shown in detail later. At the beginning, there are layers generated for all Layouts in the Setup. This is the collection of all color channels of all fixtures that have RGB channels, or a Layer for fixtures with intensity channels only. When placing fixtures in the setup step, fixtures will be assigned to the respective Layer. Building more than one group during setup will create additional Layers. In these Layers you can place media like images or videos, or effects from the Effect Library.
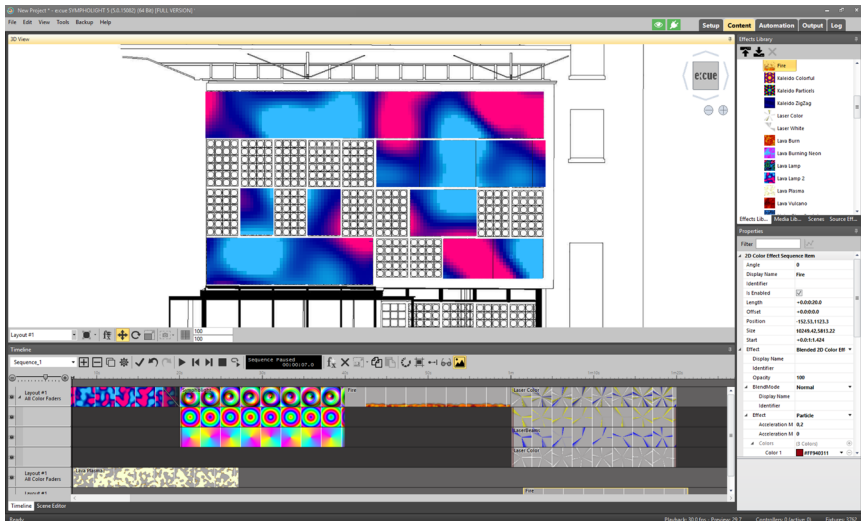
Layers higher up in the layer hierarchy have a higher priority in visibility. Effects can be color or intensity effects. Be aware, that all fixtures are assigned to the two initial layers for color faders and intensity faders automatically, depending on the data in the Fixture Library during setup of the fixtures. To place media or effects from the windows in the upper right, drag-and-drop them from Effects or Media Library into the Layer in the Timeline. After placing elements in the Layer, properties for the element can be adjusted in the lower right Properties Window.

One set of Layers and content builds a Sequence. You can have as many Sequences as you need. Each Sequence realizes a part of the show or application. You can add or delete Sequences as necessary. Later, when adding Automation, Sequences are the main target of operation.

7.3   Automation

In the Automation Window, you will define a Workflow for control and user interfaces. With Automation you prepare the show how it will be executed live and how to react on external signaling. After the fixtures and content have been created, you can setup automation. The Automation Window constists of the Workflow Designer, the Scripting Window and the GUI Editor.
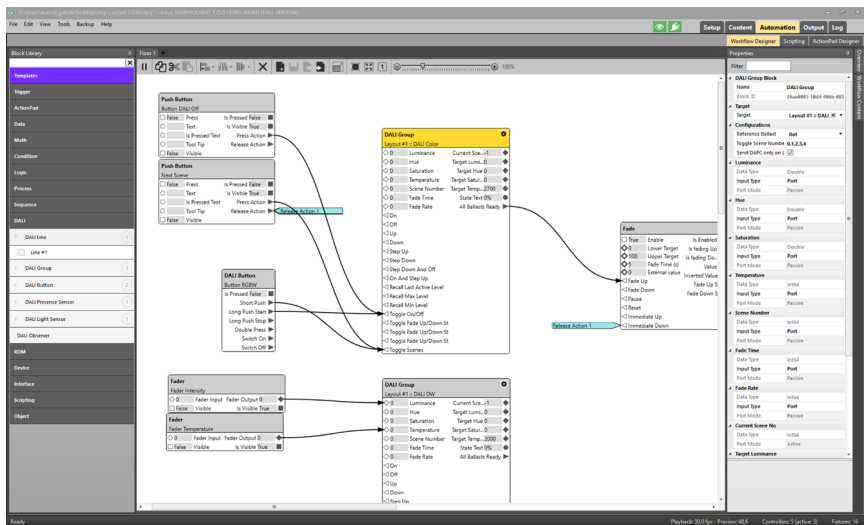
The Automation Window contains all functions that are necessary to make your show really complete. This window contains a graphical interface for the end user, so the user has only access to functions he needs and not the interface of the original SYMPHOLIGHT during development of a show. Automation also defines the reactions to external signaling, time-controlled execution and necessary calculations.

7.4   Output

In the Output Window you can control the Sequences and monitor the output of SYMPHOLIGHT.
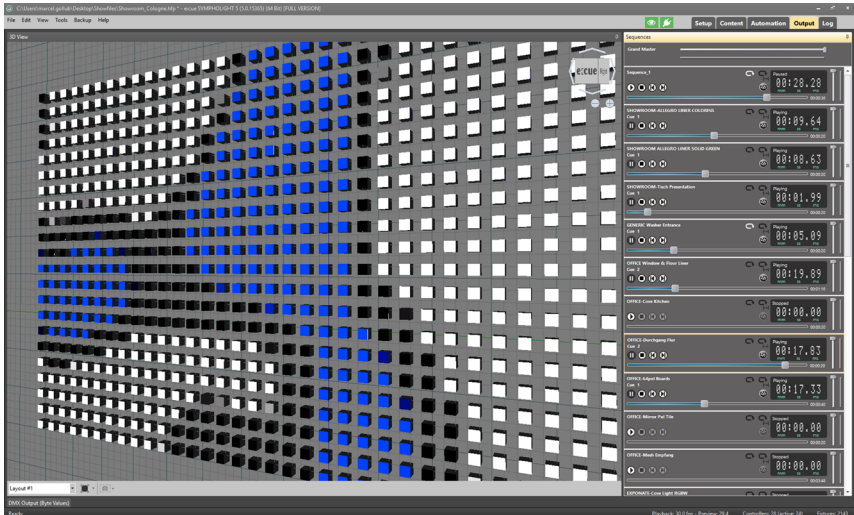
From the Output Window, you can start and stop sequences. The Output Window shows what is really sent to the fixtures. If you edit the content in the Content Window, the changes are not valid until you make the changes valid with the check symbol. Editing happens in the so called blind mode, except the parameter _Auto save changes to content_ is set in the Application Settings.

7.5   Log

The Log is the last tab of the main window of SYMPHOLIGHT. Here error messages and warnings are collected from the whole system.
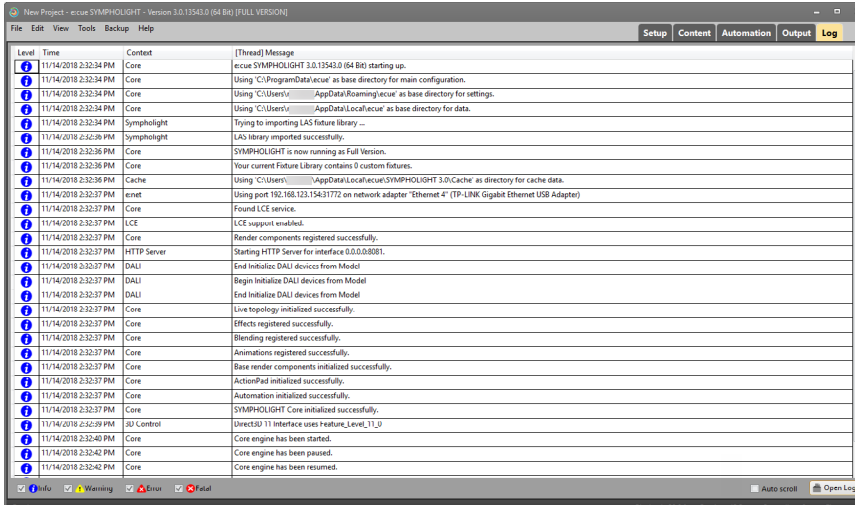
The Log Window has a selection for the levels of entries and an Autoscroll feature now.
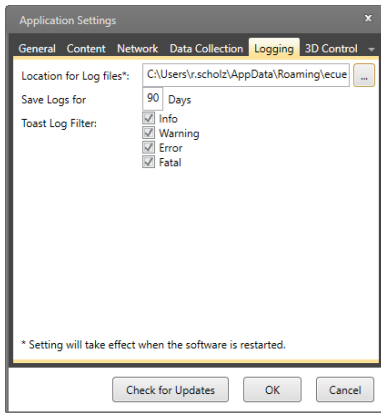
In SYMPHOLIGHT main settings, via the main menu, you can also select which toaster (in Window's taskbar) messages are logged to the global logging file.

8 User interface

8.1 Main menu

In the upper left corner you find the main window menu for SYMPHOLIGHT. Here is a list of selectable commands and their functions.
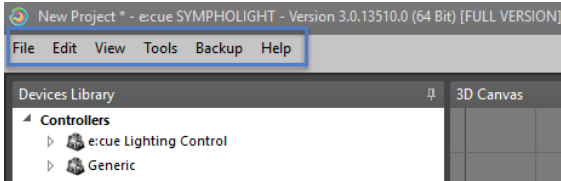
File	
New	Create a new project.
Open	Load an existing project.
Save	Save the current project.
Save As ...	Save the current project under a new name.
Import Patch ...	Import a SYMPHOLIGHT (.slpatch) or Patchelor (.pdoc) patch files.
Export Patch ...	Export current setup as SYMPHOLIGHT patch file (.slpatch).
Export Project with Media	Export project to file with used media (.hfp).
Recently Used Project Files	A list of the most recent used projects.
Exit	Leave SYMPHOLIGHT.
Edit	
Undo	Reverse the last change.

Redo	Repeat the last undone action.
View	
Reset View	Reset the window distribution to the default values.
Tools	
Application Settings	Define application options like paths, network interface and logging. For details see page 32.
Bulk Change	Change properties for all media or effects.
Configure Keyboard Shortcuts	Assign keys to functions.
Backup	
Configure Backup	Enable Backup and define role. For details see page 30.
Help	
System Manual	See the Manual.
What's New?	Change Log.
About	Application information.

8.2   Usability

In the upper right corner, left of the concept Windows, you find the two buttons to handle general behavior of SYMPHOLIGHT.

👁	Define which content is provided to the fixtures. Important when running a live operation. Green: Actions of every shown Window provide content for the fixtures (Patching test mode and Content preview included). Helpful during commissioning phase. Red: Blind Mode activated. Actions in the Setup and Content Window are ignored when shown - only actions in the Automation and Output Window provide content to the fixtures (Patching test mode and Content preview excluded). Recommendend during live operation.
🔌	Define e:net (network) communciation to hand over the control of devices to other SYMPHOLIGHT instances, e.g. SYMPL Core S. Green: e:net communication active. This very SYMPHOLIGHT has the control over devices. Red: e:net communication disabled (muted). Another SYMPHOLIGHT instance can take over control over controller.

8.3   Hidden menus

When sub windows are resized, so that not all buttons can be displayed, an additional drop-down mark is displayed on the right border of the window:
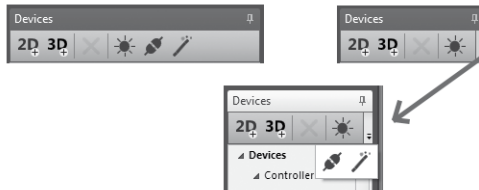
Click on the small triangle in this mark to display the hidden drop-down lists.

8.4   Saving project files

SYMPHOLIGHT project files use the extension .lprj. You can open or save project files as with any other Windows application. SYMPHOLIGHT remembers the last recently used project files, the list size of the most recently used files can be set in the application settings. Set the default location for projects in the application settings. There are two types of project files. Select the type while saving the project file.
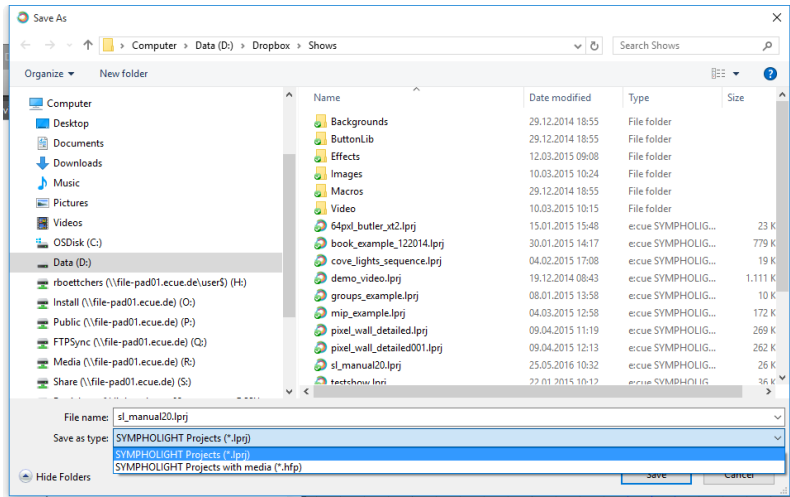
Use the .lprj version to save only setup, automation and content data. Use the .hfp version to save the project including media in the project file. This way, you are independent from your Media Library when using the project on another system. The .hfp file also includes customized fixtures and user defined effect templates.

8.5   Examples

SYMPHOLIGHT provides a couple of Example files. They are stored at C:\ProgramData\ecue\SYMPHOLIGHT 5.0\Examples.

9   Settings

9.1   Configure Backup
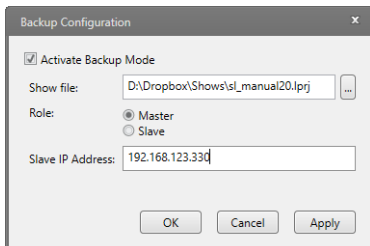
With *Backup | Configure Backup* you define a master/slave configuration for fail-safe operation. If the Master server fails, a Slave system takes over control automatically.

At first, make sure that the web server is activated on the Slave and that it has the same credentials as the Master. Run both servers simultaneously only after the configuration is completed.

Set the check mark for *Activate Backup Mode*. Select the project file used as the Backup project file. Define the role of this server, if it is the Master or the Slave.

Define the IP address for the Master server and the Slave server. Master and Slave server must be in the same network segment.
The Backup status is displayed in the title bar of SYMPHOLIGHT and in the Web UI at Settings. A restart of the servers might be necessary after the configuration is completed. Alternatively use the *Reset backup system* as described below. The backup project file and the currently used SYMPHOLIGHT project do not need to be identical.

Statuses of the Backup system:

Status	Description
Disabled	The Backup is not enabled.
Connecting	The server is waiting for the other partner to get online.
Connected	The connection between the Master and the Slave is successfully established.
Upload Pending	The upload of the project file from the Master to the Slave is pending because the Slave is started with a delay of 5 seconds. Appears only on a Master server.
Uploading	The upload of the project file is in process.
Error	An error occurred. Details are entered in the Log. The Backup system remains in this status until SYMPHOLIGHT is restarted.
Timeout	Gets the Master offline for more than 60 seconds, the Slave changes to Timeout status for 30 seconds and waits for the Master to come back online.
Backup	Has the Timeout period of 30 seconds elapsed without the Master coming back online, the Slave changes to the Backup status. The Slave loads its project in SYMPHOLIGHT and takes over control. The Slave stays in the Backup mode until either its Backup feature gets disabled or the Slave is changed to Master. All controllers will be connected to the Slave even if the original Master comes back online.

To control the Backup system, two functions are available at _Backup_ and in the Web UI: Use _Give control back_ for the Slave to check whether a Master server is available. If a Master is available, the Slave hands over control to the Master.
Use _Reset backup system_ to reconnect the Backup system after an error occurred and got fixed.

— You can reupload a Backup project file on the Slave with _Backup | Reupload Project_.
— Can a Slave not establish a connection to the Master, the Slave changes into Timeout status after 60 seconds. After another 30 seconds, the Slave loads its project in SYMPHOLIGHT and takes over control.
— When you encounter upload problems of a project file, try a manual project upload to discover network problems like firewall or configuration reasons.
— If you use any UDP Block in the Workflow and use the same project on both devices, set the IP address of the UDP Block to 0.0.0.0.
— Please note that local serial ports are not supported. In this case use a SYMPL serial Node.
— Is the Master server to be replaced, use the same IP address of the previous Master and the same software version. Please configure the new Master in a discrete network first.
— When you use custom fixture, please match the Fixture Library on both devices.
— A Master Core S will not automatically upload projects to the Slave. On a Slave Core S you have to upload the Backup project file manually.
— A Slave is not allowed to load a project in SYMPHOLIGHT on starting. A log message will be created.

Device / Master		Can be backed up by / Slave		
		Core S	PC / LCE2MX	LCE2 / LCE2FX
	Core S	x		
	PC LCE2MX		x	x
	LCE2 LCE2FX			x

9.2　Application Settings

With _Tools | Application Settings_, you define the general application settings for SYMPHOLIGHT. The dialog for Application Settings is divided into five tabs: General, Content, Network, Logging and 3D Control. Additionally you can now check for updates for SYMPHOLIGHT with a button in the lower part of the window.

General

Location for Project files	This is the default location where projects and project files are stored.
Location for User Storage	The location where user managed data is stored. For example when using the FileDownload feature of REST API or Scripting, this directory is the download location.
Load last saved Project on startup	If this is checked, SYMPHOLIGHT will load the last project after SYMPHOLIGHT was closed.
Ask the user to save changed projects	If SYMPHOLIGHT gets closed and this mark is checked, and there are changes in the projects, SYMPHOLIGHT will ask to save these changed projects. Default = on.
Number of recently used files	In the File menu, the last used projects are shown. This is the maximum number of shown project files. Default = 10.
Check for updates at start	Check the e:cue website if updates for SYMPHOLIGHT are available. Default = on.
Start with Windows	Start SYMPHOLIGHT after Windows has booted. Default = off.
Speed of Test Mode / Running Highlight	The speed of the "Running Highlight" type of the Test Mode in the Setup Window (see page 50).
Default start up tab	SYMPHOLIGHT will show the selected tab after start of the application. Default = Setup Tab.
Engine maximum frames per second	This is maximum frame rate used in rendering the content. Default = 60. Devices that do not support this frame rate are fed with 30 fps.
Language	Select a language for SYMPHOLIGHT. Setting will take effect after a software restart.
Content	

Fading of effects enabled on creation	Add a fade-in and fade-out automatically when placing effects. Default = on.
Default fade times for effects	Default automatic fade time in and out.
Fading of media enabled on creation	Add a fade-in and fade-out automatically when placing media. Default = on.
Default fade times for media	Default automatic fade time in and out.
Auto save changes to content.	When changing Content, the changes are saved to live output automatically. Default = on.
Default height for Tracks	The height of the Layers in the Content Tab.
Network	
Interfaces for e:net	Select the IP address of the network interface that is used for connections to engines and interfaces. Default = Windows user area.
Name	System name pattern. Default = %u (%h), this is *username (hostname)*.
Enable Web Server	Check to enable the SYMPHOLIGHT web server.
Network Interface	The IP address for the web connection of the Action Pad.
Port	The port for the web connection. Default = 8081.
User Name	Default: admin.
Password	The access password for webserver configuration. Initially, the password is a random value to prevent unauthorized access to the WebUI configuration, which is similar to the Core S and is also available on Windows.

Used ports	Specifies whether the REST API is available via HTTPS (Port 5001) or via HTTP (Port 5000). Setting will take effect after a software restart.
Use own SSL certificate	Specifies whether the user provides a custom SSL certificate. Otherwise a self-signed certificate is provided by SYMPHOLIGHT. Self-signed certificates can lead to problems in environments with high security requirements. Setting will take effect after a software restart.
Path to SSL certificate	Specifies the path to the custom SSL certificate. A *.pfx file is required here. Setting will take effect after a software restart.
Certificate Password	The password defined when the certificate was created. Setting will take effect after a software restart.
Enabled	Enables the static file hosting feature. When enabled, a "PublicHtml" directory is created within the directory specified as "Location for User Storage". All files and directories in "PublicHtml" are accessible from the network. The access is protected by Basic-Authentication which has to be configured with below "Username" and "Password". The files will be available at https://YourIp:5002/StaticFileAccess/FileName afterwards. Setting will take effect after a software restart.
Port	The port used to access the hosted files. Since a port can only be bound to one application, the ports used for the REST API and Webserver are not valid here. Setting will take effect after a software restart.
Username	The username for the basic authentication when accessing the statically hosted files. Setting will take effect after a software restart.

| Password | The password for the basic authentication when accessing the statically hosted files. Setting will take effect after a software restart. |

In Windows 7 and later a user without administrative rights is not allowed to change the system configuration. To access the web server your system administrator can set an exception rule, so users without administrator rights may start the web server and access it externally. Ask your administrator to enter credentials for an administrator login.

Logging	
Location for Log files	Select the path for the SYMPHOLIGHT log files. Default = C:\Users\[username]\AppData\Roaming\ecue\ Sympholight\Logs. Setting will take effect after a software restart.
Save Logs for	Time frame for which the logs are kept. Older Logs will be deleted. Default = 90.
Toast Log Filter	Which information will be logged. Default = Info off.
Log e:bus events	Logs all e:bus events for debugging. Please enable only in case of Glass Touch problems.
3D Control	
Control Layout	The used Layout mode, SYMPHOLIGHT, SYMPHOLIGHT Classic, Blender.
Smooth LEDs	Use a constant light value or fading of LEDs in the Preview.
Element Borders	Show the borders of elements in Setup, e. g. for luminaires.
Grid Size	The grid size in Setup for X, Y and Z axes. Default = 25.000.

Cell Size	The raster size of the grid. Default = 250.
Mute Visualization	Static / disabled display in Workspace (Preview). Saves engine performance. Auto: set the idle time at Mute Visualization After. Off: no muted visualization. On: immediate muted visualization; remains despite of working activities.
Mute Visualization After	The idle time after which the Workspace gets muted when Mute Visualization is set to Auto.

10 Setup

10.1   Setup Window

In the Setup Window, you patch your fixture configuration with all other components like engines, interfaces and fixtures. You also assign DMX universes and addresses as well as DALI lines to fixtures and interfaces.

Devices, and Sympholights		Layout
	Workspace 2D or 3D canvas	**Layout** Project's Fixtures, layouts, universes, and DALI lines
Fixture Library Drag and drop them into the Workspace to add them to the Setup.	**Mode Selection** Select Build Mode, Patch Mode, Background Mode and Test Mode.	**Properties** Edit device properties like DMX address, universe, and DALI device type.
	RDM/DALI Window Watch and filter RDM events and messages, or show DALI configuration	

10.2   Adding interfaces

⚠️   SYMPHOLIGHT supports interfaces and engines of the e:cue SYMPL range and recent e:cue Butler engines. The ETH2DALI and the e:bus Input Modules are only supported by the Lighting Application Suite.

To use SYMPL Nodes as interfaces in SYMPHOLIGHT, connect the Nodes via an Ethernet switch to the SYMPHOLIGHT server. Read the Setup Manual for the device and take care that the given installation requirements are met.
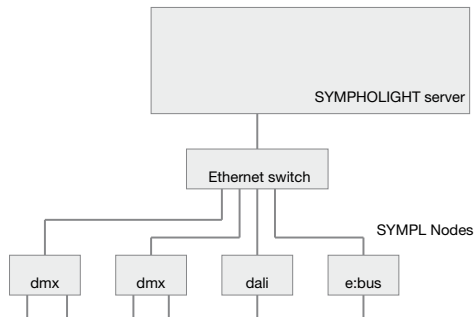
⚠️   Before adding interfaces, select the proper network interface for e:net in SYMPHOLIGHT's Application Settings. Use the preferred local network range 192.168.123.xxx with a system-internal network card.

Open the Device Manager (▤) to add or remove devices in the project:
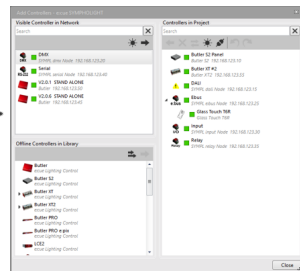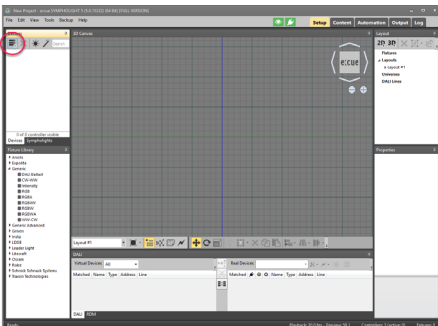
Connected Nodes in this network become available in the Devices Manager as visible devices in the network. The list of devices in the project is also shown when the Device Manager is closed in the Devices Tab.
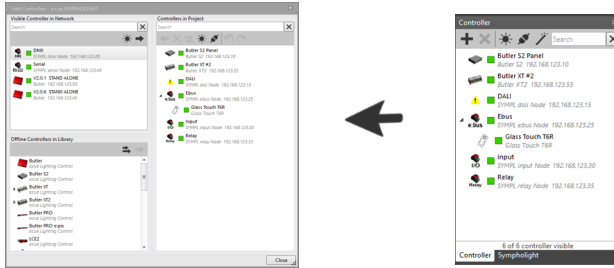
To add or remove a device, use Drag & Drop of visible devices or offline devices to the Devices in the project window. You can also use the context menu of the device or these buttons:

➡	Add selected device to the project.
⇉	Add multiple devices to the project.
✕	Remove selected device(s) from the project.
⬅	Unassign selected device(s). The real device becomes a visible device, while an unassigned device remains in he project.
⇄	Replace online device: To assign a real device to an unassigned virtual device, select an unassigned device in the project and an visible device in the network of the same type.
⚡	Enable/disable to add interfaces to show automatically: press the hardware button to add when enabled.

Glass Touches are available at the small triangle ▶ in front of Butler XT, Butler XT2, and SYMPL e:bus Node.
Further functions are available using the following buttons:

✳	Enable/disable interfaces identify mode: make Error LED and e:net LED on the device blink to identify the device.
🪄	Enable/disable replace interfaces automatically: after finishing system setup, activate this icon. If a SYMPL Node fails, you can replace it and SYMPHOLIGHT will reconfigure the new device to the current settings.

The Identify icon (✳) determines how the Identify button on the SYMPLs is handled.
A press of the Identify button makes the device blink in red in the Device window.
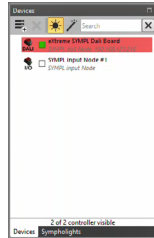
> 💡 The e:cue SYMPL Core S has a separate chapter in this manual. See „14 SYMPL Core engines" on page 291.

After the device was added to the project, you can change the SYMPHOLIGHT internal name of the device, or the network data for the device, if a manual configuration is needed. Click into the name field of the device to change the name, or change properties in the Properties Window. The following colors are assigned to the devices:

🟩	An interface marked with a green box is available and can be added or is already added to the project.
⬜	A white box indicates an unassigned devices. No real device is assigned to this interface.
🟨	If an interface has a yellow box, it is a) in test mode or b) in use, e.g. by a SYMPL Core S or another SYMPHOLIGHT instance. To get access to a used interface, in case of a) exit the test mode and in case of b) deactivate the e.g. Core S.
⬛	A virtual interface (device is offline in the show) has a grey box. Or the project is muted.
🟥	If an interface has a red box, it is in bootloader, e.g. during an update.

> ⚠ When running the Programmer of the LAS and SYMPHOLIGHT on the same e:cue LCE2 server, the IO ports of the LCE2 will be disabled in SYMPHOLIGHT. The LCE display is still available.

Select the Sympholights Tab to see all SYMPHOLIGHT instances in the network.
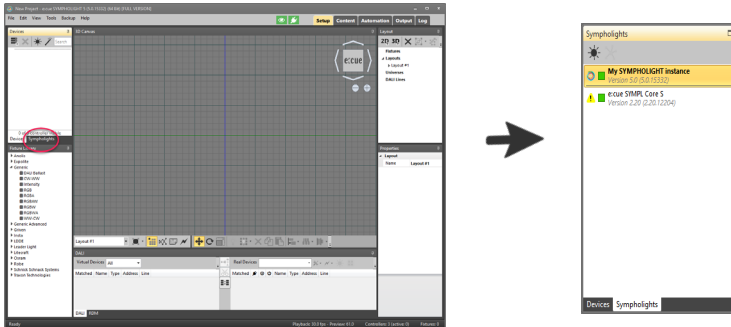
Click into the name field of the instance to see its properties in the Properties Window. The following colors are assigned to the devices:

	An instance marked with a green box has all devices in the project available.
	If an instance has a yellow box, it has at least one unassigned device in the project.
	An instance that is muted has a grey box.
	If an instance has a red box, no devices are connected.

The context menu of an instance allows you to manage your own and other instances. You can for example mute your own detection or other instances, open the web interface, share the license, upload a show, trigger a system update, or open the REST API web interface.

10.2.1 Using Butler Classic, XT, XT2 and S2

In general, these e:cue Butlers are supported by SYMPHOLIGHT as DMX interfaces and with dry contacts for Butler XT and XT2. The e:bus interfaces of Butler XT and XT2 are supported. Also the RDM functionality of the Butler S2 and XT2 is available.

⚠️   e:cue Butler devices must be configured with the Programmer/Patchelor from the Lighting Application Suite. IP addresses must be set manually. Take care for possible IP address conflicts.

⚠️   Please be aware that SYMPHOLIGHT doesn't support the standalone mode of this devices.

10.2.2 Network addresses

There are two ways to assign a network address to a device. As a default, the device gets an IP address assigned by the DHCP (Dynamic Host Configuration Protocol) server. This assures that only unique IP addresses are assigned. If you want the device to have a fixed IP address, add the device to the project, right-click on the newly added device and select _Configure Network_. In the configuration window you can change the IP Address, the Subnet Mask, the Gateway and in case you've selected a Core S, the Network Mode and the password to write the new settings to the device.
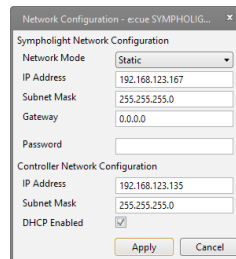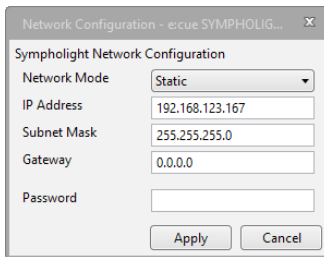
The left picture shows the Network Configuration Dialog for the SYMPL Core S. Please adjust the settings and enter the correct Password in order to transfer the configuration onto the SYMPL Core S. Please make sure the device is configured properly, otherwise the new settings won't be applied. See „14.2 Configuring the SYMPL Core S" on page 292 for more information. The right picture shows the dialog if some interfaces and Core S have been selected in the Device Window at once. The Sympholight Network Configuration will set the address for the SYMPL Core S, while the Controller Network Configuration will readdress, e.g. the SYMPL Node devices.

> ⚠ Make sure that all devices including the SYMPHOLIGHT server have a unique IP address. Duplicated IP addresses will cause malfunctions or a complete fail of the network.

Compared to the Programmer in the former Lighting Application Suite, SYMPHOLIGHT does not identify interfaces by their IP address, but by their unique hardware ID. This means that a change of the IP address does not change the DMX universe mapping. For this reason, SYMPHOLIGHT can use DHCP with changing IP addresses without problems.

10.2.3   Device properties

The protocol-specific properties of the devices are also defined in the Properties Window. Only the network address parts for the device can be changed, and the device name of the element.

For example you can assign DMX universes to the SYMPL dmx Node outputs. After adding the Node to the project, new DMX universes are assigned. Click on the plus-symbol in the Address Mapping to assign a new DMX universe, or click the down-arrow to assign an already used DMX universe, which was created while placing fixtures. Click the x-symbol to remove the assigned universe.

In the same way, you can select DALI lines for the SYMPL dali Nodes, or any other functional property for the rest of the SYMPL Node family.

Properties shown in grey are not editable in SYMPHOLIGHT, they are read-only values like the MAC address, model numbers or unique hardware IDs given during production of the device. The Device Status in the Properties Window shows the status of the device, if it is online or added to the project. Even the internal temperature of the device is displayed.

See „10.15 SYMPL Device Properties" on page 59 for a description of device properties and their meaning.
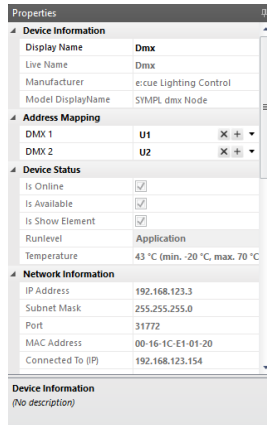
> ⚠ Do not change properties if you are not fully aware of the results, for example parameters for the DMX protocol. Changing parameters in an uncontrolled way can make the device fail or cause spooky results.

10.2.4   Additional functions

A right-click on the device in the Device Window opens a context menu, where you can remove the device from the project, reset the device and enter or leave the bootloader for uploading a new firmware to SYMPL Node interfaces. SYMPHOLIGHT comes with the necessary firmware versions for SYMPLs built-in, so you could upload a firmware without loading it from the web site.
The command Unassign in the context menu removes the device from the list and replaces it with a virtual device of the same type. This is useful when replacing a device by another device of the same type. The port assignments are kept.

10.2.5   Virtual interfaces and devices

You can also drag-and-drop interfaces and devices from the Device Library into the project, like SYMPL Nodes. These virtual interfaces can be used like real physical interfaces. This is helpful when setting up configurations without the real devices present. Later, you can replace the virtual devices with real devices. The same is valid when assigning DMX universes. You can place fixtures in the Workspace and delete/ add/assign DMX addresses and universes, when still no interfaces are present. Later, you can add interfaces and map the outputs accordingly.

10.2.6   User terminals

SYMPHOLIGHT supports the e:cue Glass Touch T6, T6R and T12 user terminals via e:bus. They can be used and configured in the Workflow Designer in the Automation. See „12.2 Workflow Designer" on page 72 for more.

10.3   Icons in the Mode Selection

These are the first five functions on the left of the Setup Mode, the Mode Selection:

	Optimize view, reset to default view or define as new default view, select with the small triangle. Zooms to an optimum view of the fixture set.
	Switch to Build Mode: place fixtures, add interfaces.See „10.4 Build Mode" on page 46.
	Switch to Patch Mode: set wiring and distribution. See „10.10 Patch Mode for DMX and e:pix" on page 50.
	Switch to Background Mode to set text and backgrounds. See „10.12 Background Mode" on page 57.

⚡	Switch to Test Mode: test fixtures and patching. See „10.13 Test Mode" on page 57.

10.4　Build Mode

In Build Mode, you place, address and arrange all fixtures and devices.
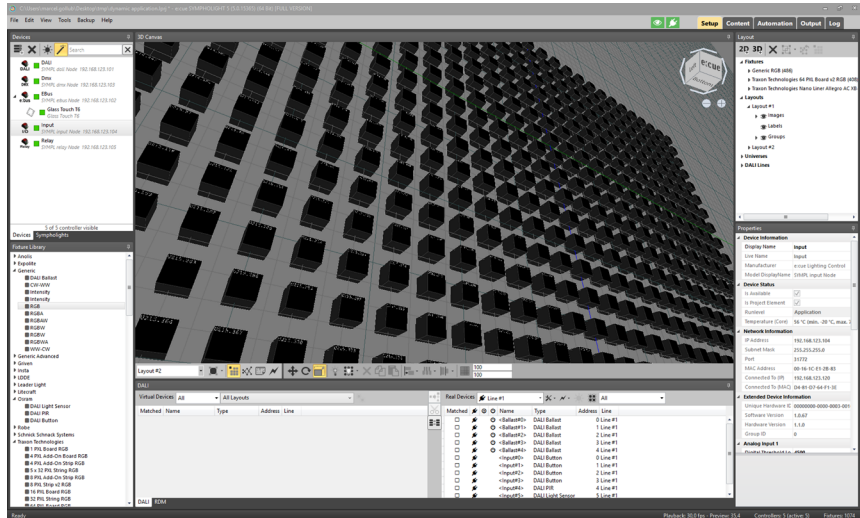
The following functions are available in Build Mode:

✥	Move tool: move fixtures. Or change the coordinates in the Properties.
↻	Rotate tool: rotate fixture. You can also rotate by setting the angle value in the Properties.
⛶	Scale tool: scale a group of fixtures.
💡	Create fixture: after selecting a fixture from the library, places a single instance of this fixture in the Layout.
⣿	Create fixture matrix: create a fixture matrix with left mouse button pressed. You can use an offset for X and Y with the drop-down menu under the small triangle to keep space between the fixtures..
✕	Delete the selected fixture(s).

	Copy selected element(s) to clipboard
	Paste the clipboard contents
	Align selected fixtures left.
	Space selected fixtures evenly.
	Distribute selected fixtures left edge.
	Create a group for the selected fixtures.
	Enable or disable snap to grid, change grid spacing in the right fields.

There are three ways to add and place devices.

— In any case switch to Build Mode.
— Select a fixture from the Device Library and drag-and-drop it into the workspace.

Or

— Select a fixture from the Device Library.
— In Build Mode, select Create Fixture Matrix and draw a matrix in the Workspace keeping the left mouse button pressed.

Or

— Select a fixture from the Device Library.
— In Build Mode, select the Add Single Fixture and click into the workspace. A new fixture will be added to the Workspace with every new click.

10.4.1   Navigating elements

Before you can move elements, you have to select them. This is done in a similar way as in other Windows applications.

— Click a single fixture to select it.
— With the left mouse button pressed, draw a rectangle around fixtures. These fixtures get selected.
— Keep the Ctrl key pressed when selecting fixtures. These fixtures are added or removed from the selection.
— Keep the Shift key pressed. There fixtures are added to selection, not removed.

The selected fixtures get a blue frame. You can now click and move the fixture with the blue frame or drag the arrows. Placement can be aligned to a grid.

If the grid is enabled (click *Snap to grid icon* to change), you will see grid markers in the background of the Workspace. In the Mode Selection, there is a grid icon, click it to enable the grid or to disable it. Click on the grid size values to edit.

To zoom in and out for the workspace, there are two circular icon buttons in the upper right corner of the Workspace. You can also zoom in and out with the mouse wheel. The four cursor-like elements in the right of the workspace move the view of the workspace. To change properties of a fixture, click the fixture in the Workspace. In the Properties Window, you now can change values like assigned DMX universe and DMX address. Setting the DMX universe requires available interfaces.

Fixtures can be moved after they are selected. If Snap To Grid in the menu bar below is active, every move snaps to the underlying grid. If disabled, the fixtures can be moved freely. Use the pressed left mouse button or the cursor keys to move. Pressing the Ctrl key and using the cursor keys shifts the view port of the workspace.

10.5   Layouts, Layers and Groups

A Layout contains all fixtures and devices for a project. You can have one or many Layouts, representing parts or subdivisions of the whole project. SYMPHOLIGHT supports two Layout models, 2D Layouts and 3D Layouts.

For every Layout with DMX fixtures, SYMPHOLIGHT creates an associated Layer in the Timeline in the Content Window. Depending on the fixture types, these Layers are used for color or for intensity only, if the used fixtures only support intensity control. For every Layer, you can define as many Sequences as necessary. There will be at least one Layout and one Layer in the Content Window, if you place fixtures in one Layout. If you want to address only certain fixtures in a Layout, you can group fixtures.

In the Layout, you can select or multi-select fixtures by keeping the Ctrl key pressed and clicking fixtures, or keeping the left mouse button pressed and drawing a rectangle around fixtures. Click the Create Group icon in the lower icon menu to create a new group. You can add new fixtures to already created Groups or merge different Groups by using the context menu of each Group in the Device Window. If you select a fixture or a selection of fixtures, SYMPHOLIGHT will show the Groups they belong to, in the Properties Window.

Later, in the Content Window, Groups get a separate Layer. You can place effects or media in the Layer and they will only be mapped to the fixtures in this group.

10.6   Patch files

While all parts of a show, Content, Setup, and Automation, are stored in the project file, you can also save and load the Setup results in a separate file. These files are called Patch files and contain only the fixture and interface configuration. To save or load a Patch file, use *File | Import Patch* or *Export Patch*. With *File | Export Patch with Media* also images and Labels are included.

10.7   Using custom fixtures in SYMPHOLIGHT

It is possible to integrate custom fixtures. In order to create a custom fixture, the Library Editor of the Lighting Application Suite can be used. Once the fixture is defined, SYMPHOLIGHT will update the Devices Library after a restart of SYMPHOLIGHT.

⚠️ Custom fixtures may only contain color and intensity channels. All fixtures containing different channel types will not be loaded on start-up.

⚠️ Please make sure to export a *.hfp file onto a SYMPL Core S if your show contains custom fixtures.

10.8   Placing DALI devices

DALI devices are also part of the Devices Library and are placed and arranged the same way as DMX devices.

There are two groups in the Devices Window for DALI. In the group Generic, you will find DALI devices that are supported vendor-independently, e. g. ballasts. SYMPHOLIGHT can control all DALI ballasts that conform to the DALI Standard. In the group OSRAM, DALI buttons and sensors are available.

> ⚠ SYMPHOLIGHT supports only sensors and buttons from the OSRAM DALI Professional range. Sensors and buttons from other vendors are unsupported, due to the specific protocol versions in DALI.

A Layout contains all fixtures and elements of a show, DMX devices as well as DALI devices. While DMX fixtures need a two parameter address only, DALI devices are configured in a more complex way. See „10.11 Addressing DALI devices" on page 51 for addressing DALI devices.

10.9   Non-DMX and non-DALI devices

Other devices and interfaces like serial interfaces, UDP interfaces, SYMPL input Nodes or SYMPL relay Nodes are not related to the content as media. They are added and configured in the Setup Window, but they require basic configurations like system interfaces, IP addresses and names only. Adding these interfaces to the Setup works the same way as fixtures. Either select them from the Devices Library and drag-and-drop the into the Workspace, or add them like the SYMPL dmx and dali Nodes, e. g. via _Add to Show_ in the context menu. They are then visible in the right-side Devices list, you can select and configure them in the Properties Window.

10.10   Patch Mode for DMX and e:pix

In Patch Mode for DMX devices, you select the wiring of fixtures in the DMX universes. Usually, when placing fixtures in the Workspace, SYMPHOLIGHT assigns DMX universes and addresses according to the number of addresses the fixtures cover in linear mode.
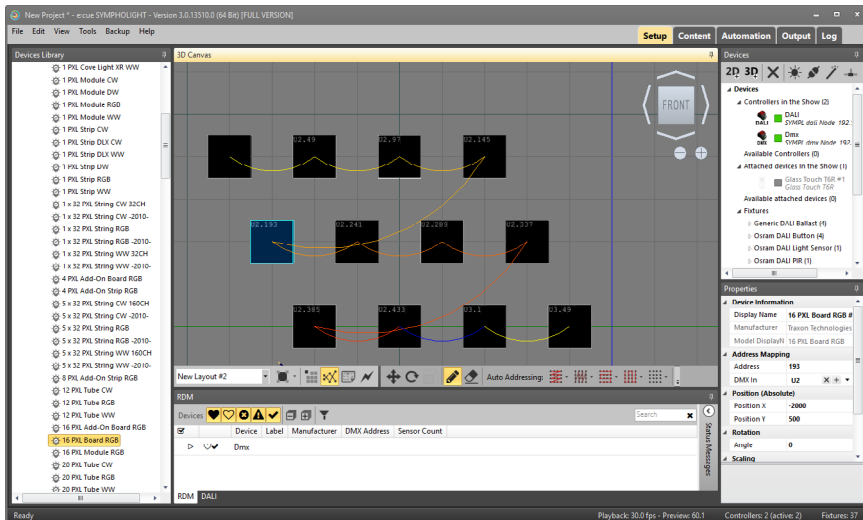
Mode selection in Patch Mode has the following icons with drop-down menus behind:

✏️	Create a single patch line.
🩹	Remove a single patch line.
≋	Address selected fixtures row-wise.
〰️	Address selected fixtures column-wise.
⬚	Address selected fixtures row-wise in snake mode.
⫼	Address selected fixtures column-wise in snake mode.
⋮⋮⋮	Remove all connections/from selected fixtures/in Layout.

You can change the wiring by clicking on a connection, keeping the mouse button pressed and redirecting the connection to another fixture. Use the dots in the middle of a fixture as start or end point of a connection. SYMPHOLIGHT can also do an automatic addressing with the four icons in the center of the Mode Selection. Select fixtures with dragging a rectangle over all fixtures to select. Click on an auto-addressing icon to execute automatic address assignment.

If fixtures are added and removed again, the now unused DMX universes will stay in the configuration. To remove the redundant universes, select or multi-select the universes in the Devices Window and delete them manually.

10.11   Addressing DALI devices

Using DALI requires at least one SYMPL dali Node. Add it to the project as usual. A DALI line will be assigned automatically to the Node. Now:

— Connect the SYMPL dali Node to the actual DALI equipment.
— Place the expected DALI devices in the Layout, e.g. based on a picture of a floor plan.
— In the RDM/DALI Window, select the DALI tab.
— The DALI Window is split into two different areas: the left area is called Virtual Devices and is showing all devices that have been added to the 3D view before. The area on the right hand side is called Real Devices and shows all DALI devices that have been automatically discovered on the selected DALI Line.
— In the DALI subwindow, select a DALI Line on the right part (Real Devices). The SYMPL dali Node will scan all devices.

— All detected and configured DALI devices are visible on the right side of the DALI Window, all DALI elements in the Layout are available on the left side.

> ⚠ The current SYMPHOLIGHT release will not support reading DALI device parameters back into the software but will only push the device configuration from patched devices onto the connected DALI devices.
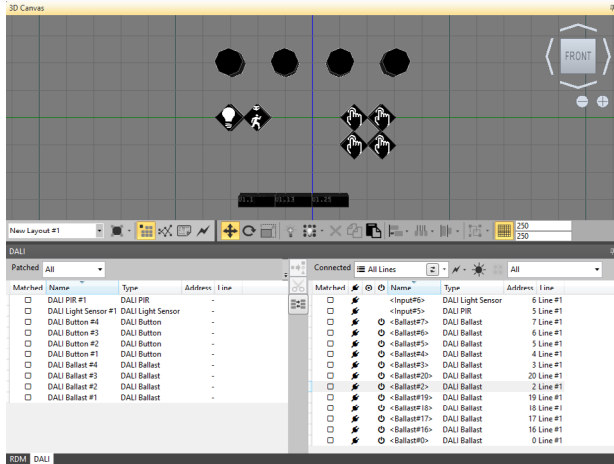
Now map the patched to the connected devices, i.d. the virtual to the real devices. Click on a device in the left list for a patched DALI element, or select the device in the workspace. Then select in the right list a detected DALI device. The left DALI device will stay selected in light grey. Click on the _Match virtual device to real device_ icon located between the two lists to match the two devices. Or drag-and-drop a patched device from the left list to the connected device on the right list.
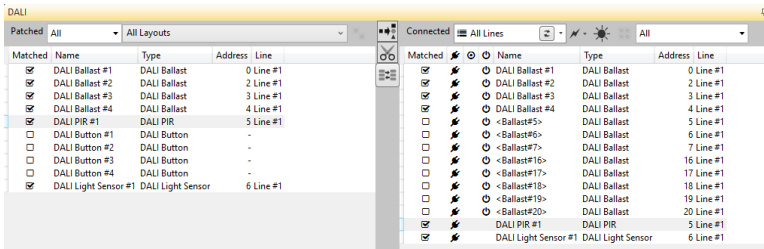
Here are the columns displayed in the DALI Manager:

Virtual DALI Devices

Matched	If marked, the patched device is mapped to a connected device.
Name	The name of the patched device.
Type	The DALI type of the device.
Address	The short address of the device.
Line	The matched DALI line.
Real DALI Devices	
Matched	If marked, the connected device is mapped to a patched device.
⚡	The device is connected.
⚠	An exclamation mark is shown, if the device is patched but not connected
⚡	A flash is shown when two or more devices have the same short address.
◉	Signals an event for a device, e. g. a button being pressed or a sensor signaling a change.
⏻	Switches the DALI device on / off.
Name	The name of the device.
Type	The DALI type of the device.
Address	The assigned DALI short address.
Line	The used DALI line for the device.

There is no explicit Patch Mode for DALI devices like for DMX devices. Set the properties of a DALI device either in Patch Mode or in Build Mode. Select a DALI device from the left list and change the properties of the device in the Properties Window, e. g. Fade Rate or Fade Time.

If there are unaddressed devices at the line available, this icon 💬 indicates this situation.

In the list of patched and connected DALI devices, there is a filter function in the top menu bar, to select which devices are displayed or hidden. If the subwindow is too small to show these drop-down lists, click the small triangle on the right border of the window.

10.11.1   DALI grouping

SYMPHOLIGHT handles DALI ballasts always in groups. To create a group of DALI ballasts, select the ballasts in the Workspace, either by drawing a frame with pressed left mouse button, or by clicking ballasts keeping the Ctrl key pressed. Click the Group icon in the Mode Selection and a new DALI group will be created in the Devices list. Rename the group as required. If you want to control a single DALI ballast, create a DALI group with this ballast as single member.
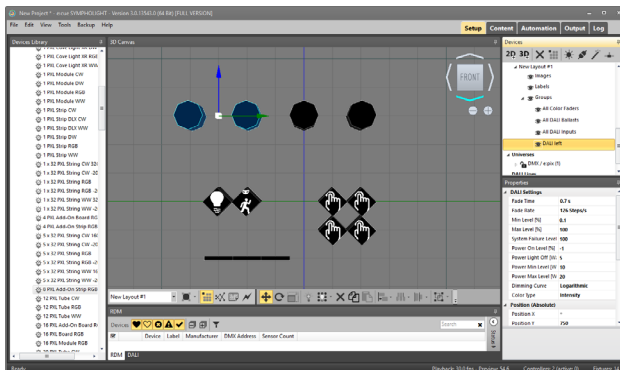
Selecting the DALI groups in the Devices list will highlight the included ballasts. Also selecting the All DALI Ballasts and All DALI Inputs will highlight the included DALI devices.

When selecting a DALI device, you can edit the device properties in the Properties Window. This includes the usual DALI parameters like Min and Max Value, Fade Rate and Fade Time, Power On Level etc. When more than one ballast is in a group, you have

to define a Reference Ballast in the DALI Group Block in the Workflow Designer. See "„12.2 Workflow Designer" on page 72" for more details.

10.11.2   Additional DALI functions and features

There are some further functions in SYMPHOLIGHT for DALI. For the left-hand list of patched DALI devices, the drop-down list at the top provides a filter function, for which devices are displayed.
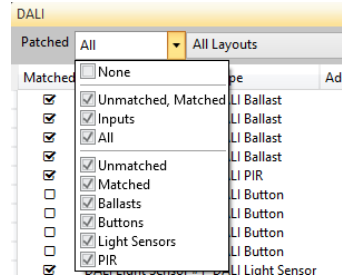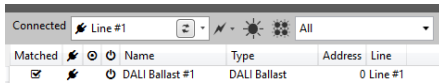
In case the line is scanning, addressing or initializing, you'll see this by the indicator in the line combo box
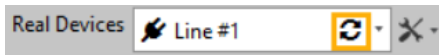
To test or verify your DALI connections, there are two test functions available in the top menu bar of the list with connected devices. To use them, select the DALI line to test, here Line #1. Do not use _All Lines_! Click the single light ☀ and select a ballast. The selected ballast will blink. Click the light matrix and the blink commands will be broadcasted to all connected DALI ballasts, even though they are not addressed yet. For sensors like PIRs or for buttons, the Event icon in the list of connected devices will blink. This way, you can identify buttons or sensors by watching the list of connected devices.

Test a whole DALI Line using Broadcast by clicking the lightning icon ⚡ :

Fader	Use the fader to set the current value (0 - 254) of the slected DALI Line. Consider fade time.
⏻	The power button toggles to the last ON value (1 - 255) ot to 0. Consider fade time.
Min	The Min button sets the DALI Line to its physically minimum value 1 (never 0). Without fade time.
Max	The Max button sets the DALI Line to its physically maximum value 254. Without fade time.

The following DALI Line Tools are available at ✖ :

Scan All	This scans the complete line for ballasts and OSRAM inputs
Delete Devices with Address Collisions and Rescan	First, all collision ballast and OSRAM input devices will be deleted, then we start a new scan on all devices
Address only unaddressed	Only devices without Short Address will be addressed
Readdress all Devices	First all Devices will be deleted and then the complete line will be addressed. This destroys the complete previous address settings!

The context menu of a real devices offers the following functions:

Check Device Address Collision	Checks a single device for addressing conflicts
Delete device	First, it unmatches the device, then it deletes the device's short address and removes the device from the line
Modify Short Address	First, it unmatches the device, then its sets the short address to new short address.Internally, it destroys the old device from the line engine and then it scans the new device address on the line. If the new device is found, it will create the newly found device. This can now match to a virtual device.

For an interrupted connection (Network problem) from SYMPL to Sympholight (PC/CoreS), you can define the DALI bus behavior. Set the settings in the Properties of the controller (SYMPL dali Node):

Bus Power On Start	Set the DALI bus power on startup [on] or [off].
Bus Power Off Timeout (s)	The time in seconds after the DALI bus power will be switched off in offline mode.

> ⚠ SYMPHOLIGHT will automatically keep scanning the DALI line and addressing new devices if the DALI window is visible in the Setup tab. Please be aware that a constant scanning of the DALI bus might slow down the execution of other DALI relevant commands.

10.12 Background Mode

In Background Mode you can place background images or labels with text in the Workspace.
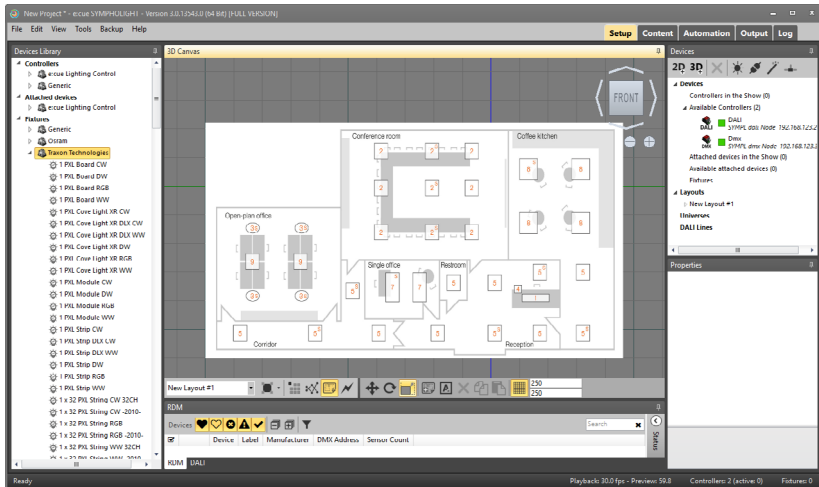
The Background Mode has only two functions and icons.

🗒	Image Mode: Add an image to the workspace.
🅰	Label Mode: Add a label (text box) to the workspace.

Click the picture or the text block icon in the Mode Selection. Select an element to add or remove the current selected element. Use the Properties Window to change scaling or size, color, font or other parameters.

10.13 Test Mode

In Test Mode, you can test and check your DMX setup and patching.
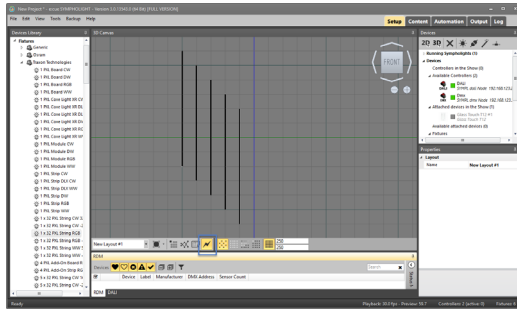
The Test Mode supports four test scenarios for the connected fixtures.

	Blink: all channels in the selected fixture blink.
	Highlight: all channels in the selected fixture are set to full intensity,
	Running Highlight: A pattern runs from lowest to highest channel though the selected fixture. You can define the speed in the Application Settings.
	Tile Orientation: The lowest channel in the selected fixture is at full intensity. The other channels are dimmed. The other unselected fixtures are dimmed in blue.

Click a fixture, select the test pattern. For the Running Highlight, make sure that the direction in the preview in the workspace and on the fixture is the same. Otherwise rotate the fixture.

10.14   The RDM Window
In the RDM Window, SYMPHOLIGHT collects for all DMX devices and their subdevices data like sensor values or functionality indicators.
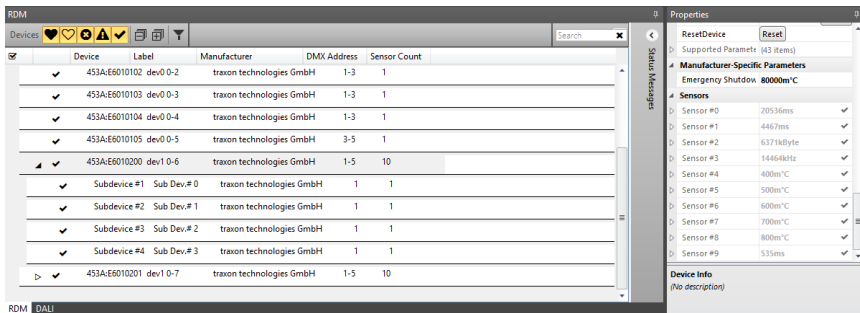
By default, RDM is not enabled for interfaces. Right-click on a port of a DMX device in the RDM Window and select Enable RDM:
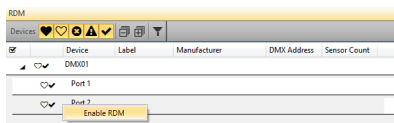
In the top icon row of the RDM Window, you can filter RDM events by their level or you can search for specific RDM entries. Clicking a single device or a subdevice, you can edit values not greyed-out in the Properties Window, if the device supports this.

10.15   SYMPL Device Properties

Properties that are read-only and cannot be modified, are shown in grey.

Properties of all SYMPL Nodes	
Name	The logical device name, e. g. DMX01.
IP Address	The assigned IP address, e. g. 192.168.123.11
Subnet Mask	The subnet mask, default = 255.255.255.0
Port	The UDP port number, default = 31772, read-only
MAC Address	The hardware MAC address, read-only.
Connected To (IP)	The IP address of the SYMPHOLIGHT server the Node is connected to, read-only.
Connected To (MAC)	The MAC address of the SYMPHOLIGHT server the Node is connected to, read-only.
Manufacturer	e:cue Lighting Control, read-only.
Model Name	The type of the device.
Unique Hardware ID	The internal hardware ID of the device. This allows the device to be addressed uniquely, read-only.
Is Online	Shows the device as online or offline, read-only.

Is Available	Shows if the device is available, read-only.
Is Show Element	Shows if this device is used in the project, read-only.
Runlevel	Current state of the device, e. g. while updating or resetting, default = Application.
Temperature	The internal temperature of the device, read-only.
Unique Hardware ID	A generated unique ID to make the device addressable without relation to the IP address, read-only.
Software Version	The version of the internal firmware, read-only.
Hardware Version	The production release of the hardware, read-only.
Properties of SYMPL dmx Node	
DMX 1	The DMX universe number port 1 of the Node sends to. Assign a used universe or create a new one.
DMX 2	The DMX universe number port 2 of the Node sends to. Assign a used universe or create a new one.
Direction	Define this port as DMX input or output port. In case the direction is set to input, a DMX port will be added in the Workflow Designer for further processing and automation. Default = Output.
DMX Receive Length	The block size after which a received DMX block is seen as complete, default = 512.
DMX Break Length	The Break resets the line, signaling a new DMX Packet, default = 100 us.
DMX MAB Length	Length of Mark After Break, MAB signals the receiver to begin reading data default = 17 us.

RDM Break Length	The Break resets the line, signaling a new RDM Packet, default = 180 us.
RDM MAB Length	Length of Mark After Break, MAB signals the receiver to begin reading data default = 17 us.
RDM switch delay	The delay between sending a DMX and an RDM block, default = 176 us.
Startup Behavior	State of the outputs after startup, default = HighImpedance.
Offline Behavior	State of the outputs when going offline, default = HoldLastFrame. You can define offline content in the Content Window and use this content as offline content. See „11.2 Window structure" on page 53.
Properties of SYMPL e:pix Node	
e:pix 1	The e:pix universe number port 1 of the Node sends to. Assign a used universe or create a new one.
e:pix 2	The e:pix universe number port 2 of the Node sends to. Assign a used universe or create a new one.
	The other properties are the same as for the SYMPL dmx Node.
Properties of SYMPL input Node	
Digital Threshold Low	Input voltage for digital signaling off, this is also the level for switching the LED, default = 4000 mV.
Digital Threshold High	Input voltage for digital signaling on, this is also the level for switching the LED, default = 6000 mV.
Event Delta	Voltage range that is ignored between the two above thresholds, default = 100 mV.

Properties of SYMPL relay Node	
Initialize relays on show load	After loading the project, the relays are initialized to a state, NC or No, default = not initialized.
Initial Relay 1	State to be initialized to, default = not initialized.
Initial Relay 1	State to be initialized to, default = not initialized.
Initial Relay 1	State to be initialized to, default = not initialized.
Properties of SYMPL e:bus Node	
	The SYMPL e:bus Node has no properties to be set or modified.
Properties of SYMPL serial Node	
Baudrate	The data rate of the interface, default = 115200.
Data Bits	Length of one data block, default = 8.
Parity	Parity check bit, default = None.
Stop Bits	Number of stop bits, default = One.
Properties of SYMPL dali Node	
Line A	The DALI line in the setup this Node is used for. Select an already used one or create a new line.

11 Content Window

11.1 Principles

The Content Window of SYMPHOLIGHT is the central area where you create the project, its content and how this content is handled, including fades and control structures. SYMPHOLIGHT works with a timeline like a sequencer. All functions and control is based on this timeline.

SYMPHOLIGHT is a media-oriented system with focus on media and effects, at first sight. These videos, images and effects are placed in the Layers of a Sequence. But, it is also possible to control even single LEDs in a fixture. Even if a timeline is often seen as not so exact and controllable as, for example, cuelists, this is not true. In SYMPHOLIGHT, all components of a Sequence can be adjusted down to milliseconds. The snap mechanism in SYMPHOLIGHT allows very precise adjustment even for fade times and cues for run control. The advantage of a timeline is the clear graphical user interface, which leads to a more intuitive work flow. You always see what you will get, compared to other concepts, like cuelists.

The content of a project is organized in Sequences, these Sequences contain Layers. The basic Layers are created automatically during the Setup process. Depending on the types of fixtures placed in Layouts, a Sequence contains different Layers. If the fixtures in a Layout are RGB-capable, the corresponding Layers will be set for color channels. If the fixtures only support intensities, the Layers will be intensity channels. DALI ballasts are handled in a separate way and do not have Layers. While the basic Layers are created while defining the Setup, you can add additional Layers by placing media content or effects in the timeline.

> This chapter deals in general with DMX-oriented functions in SYMPHOLIGHT. DALI functionality is explained in more detail in the chapter Automation, even though some few DALI functions are part of the Content Window, e. g. DALI Scenes (see „11.17 DALI Scenes" on page 80).

11.2   Window structure
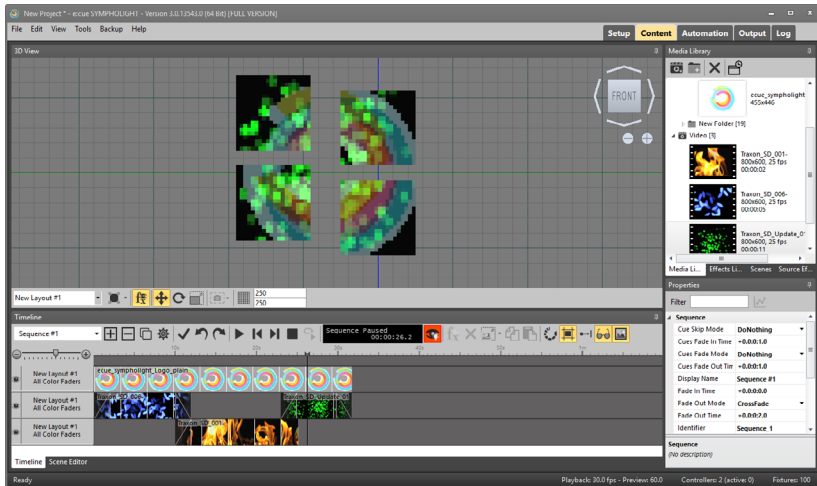
Preview Window	Media, Effects, Scenes
Control Bar Play/Pause/Stop/Skip, select Sequence handling.	Properties Window Edit properties of timeline components, Effects or videos.
Timeline and Layers in Sequence Sequences with Effects and Media	

11.3   Preview Window

In the Preview Window, you see the simulation of the output that is sent to the fixtures.
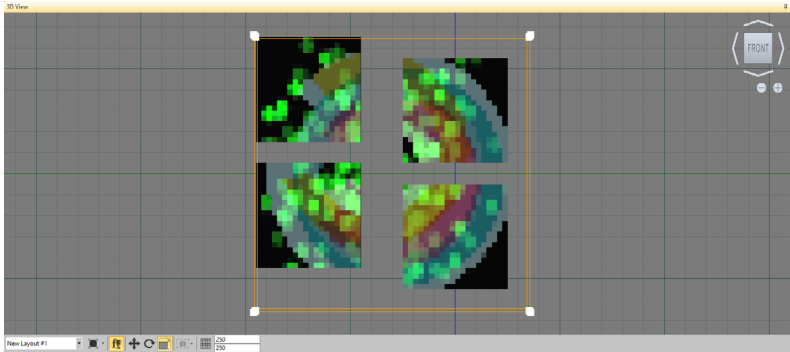
The Layout, that is displayed, can be selected by the drop-down box at the bottom of the Preview Window. Zooming in and out is done with the control elements in the Preview Window or with the mouse wheel. The yellow frame is the area on which an Effect or a media is mapped: the Effect Rectangle. Additional functions are available with the icons below. The Effect Rectangle frame is the view port and defines, how the content of the Sequence is mapped on the fixtures. So the view port can be smaller than the fixture area, where only a part of the content is mapped.

	Optimize view for current window size.
	Hide or show the Effect Rectangle.
	Move Effect Rectangle.
	Rotate Effect Rectangle.
	Scale Effect Rectangle.
	Current content as offline image: run the Project, pause it at a position and use this function to set an output image for SYMPL dmx and e:pix Node for offline situation. They will only receive that part of the output they are responsible for.
	Set the grid size for the Preview Window and the View Port.

Use the icons in the top right corner or _alt + pressed mouse_ click to rotate the canvas.
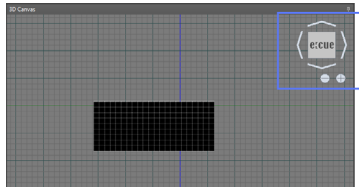
11.4   Control Bar

The Control Bar provides a selector for the currently working Sequence, an icon to create a new Sequence or to delete a Sequence as well as control functions.
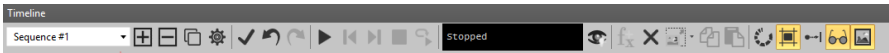
Changes in the Sequence can be saved or undone. Additionally, information about the running Sequence is displayed and you can add or delete elements in a Layer.

	Create a new Sequence.
	Delete the current Sequence.
	Change the Properties of the Sequence in the Properties Window. You can also click into an empty area of the Sequence to edit the Sequence Properties.
	Save the Sequence to the Live View. If the option _Auto save changes to content_ in the main settings is set, this is done automatically if the Tab is changed or another Sequence is selected.
	Revert to original Sequence, undo the last changes that have been saved with a Tab switch or before the Sequence has been changed.
	Play the Sequence from current cursor position.
	Pause the Sequence, click again to return to playing.
	Skip to previous Cue.
	Skip to next Cue.

■	Stop Sequence.
↻▶	Play selected Effect and loop.
👁	Live Preview on Stage: if this mode is enabled, the Content preview is sent as real DMX output to the fixtures. If this mode is disabled, no DMX output is generated and you edit in blind mode. This mode is only applied if the content tab is active! The status will be saved in the project file.
f_x	Create Effect Template: creates a new Effect template from the current selected Effect and its properties, adds a new template to the Effects library.
✖	Delete Effect/Track: deletes the selected Effect or Track in the Sequence.
⬚🖼	Resizes the Effect Rectangle to cover all fixtures in the current Layout, if you resized or moved the effect mapping to fixtures.
🗐	Copy the current selected Effect or media to the clipboard.
📋	Paste the clipboard content into the current selected Layer. The item will be pasted at the position of the time cursor on the selected track.
⟳	Auto Preview Selected Effects: toggles preview of the selected Effect only.
⊟	Enable/Disable Snapping on Timeline: toggles snapping of media, Effects and Cues in the Layer in the Timeline.
•⟶|	Follow Cursor on Timeline: if enabled, the view on the Timeline always follows the cursor while playing. If disabled, the view on the Timeline stays fixed.
👓	Show/Hide tracks from other Layouts: if enabled, all Layers from all Layouts are visible. If disabled, only Layers for the current Layout are visible.
🖼	Enable and disable thumbnails. If activated, the previewed content in the Layer is rendered as the real content will be played. Otherwise only thumbnails are displayed.

11.5 Layers

The Layers contain the real content of the project. In the Layers, media content or effects are placed. To place media or effects, select Media Library or Effects in the upper right window and drag it keeping the left mouse button pressed into the Layer of the Sequence. For the selected media or effect in the Layer, you can change properties in the Properties Window. Looping can be assigned to a media element or an effect by selecting the element and dragging the end of the element.
To create a new Layer for a Layout in a Sequence, drag-and-drop an element of the Media Library, the Effects Library, or the Source Effects on top of or below an existing Layer.
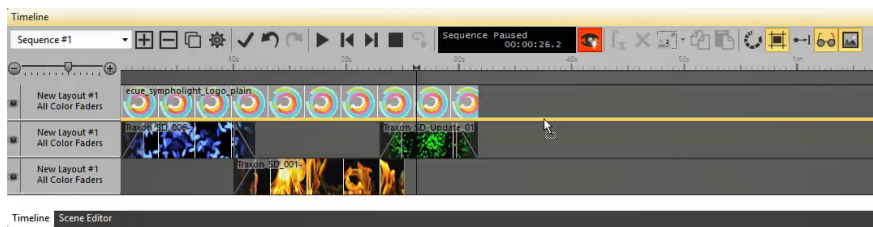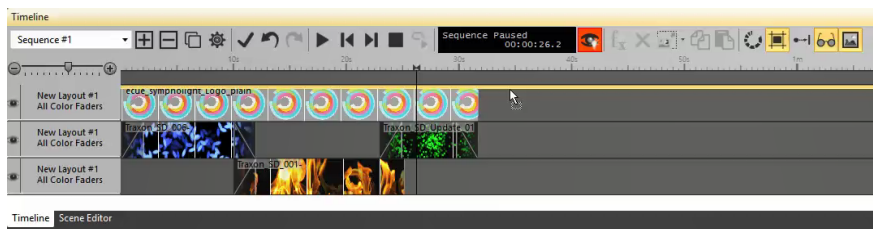
You can create new Layers by dragging media or Effects from the libraries into the Timeline, but you can also create new Layers by dragging content elements from one Layer to the upper or lower border of a Layer.
Next, you can rearrange the Layers by keeping the header of a Layer clicked and moving the Layer above or below another Layer. Every Layer has two small buttons in the Layer header. With the M icon you can mute (switch off) a Layer. With the S icon you can solo the Layer, so it becomes the only active Layer.

11.6 Media Library

To add media to the Media Library, select the chapter (Image or Video) and use the clap board icon to add a file.
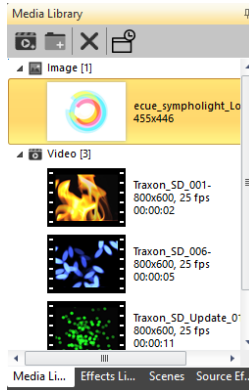
In the opened dialog, select media to be added to the Media Library. You can create subfolders in the Media Library with the folder icon in the top menu of the Media Library.

Videos will be converted to an internal format. Several video formats are supported and will be converted to a certain format in the Media Library. Define and customize the resolution of the imported videos and the bitrate.
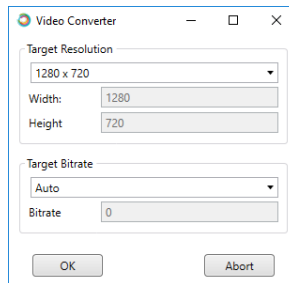
💡 The SYMPL Core S supports resolutions with 720p and lower. It does not support HD resolution.

For videos, SYMPHOLIGHT supports a tool for frame rate interpolation for videos. The tool is available via the link in the Media Library. This tool re-samples videos for slow motion or high-speed playback to avoid hard frame rate delays. Download the Free Video FPS Converter from the site and install it in your system. When the tool is installed, the message will be replaced by an icon for the tool.
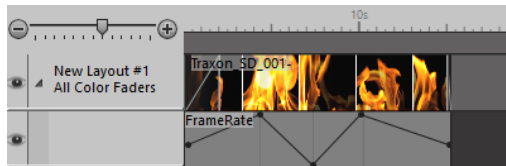
SYMPHOLIGHT will detect the tool automatically. Place a video in a Layer in the Sequence, select the video, right-click in the Properties window of the video on Frame Rate and select _Add Keyframing_. Double-click in the Keyframes below the video to place key frames and change the frame rates to create slow-mo or hi-speed effects.

Only videos imported (or re-imported) after installation of this tool can be used with frame rate animation.

11.7   Properties Window

When an Effect or a media element is selected in a Layer, you can edit the properties of this element in the Properties Window.

Click the item in the Layer to select it. There are many properties for Effects and Media to edit. Most properties and the names are obvious. Videos and images are always used as a 2D Color Effect Sequence. When you adjust an Effect with properties different from the basic settings, you can save this Effect in the Effects Library above the Properties Window. Give the Effect a new, unique name and save it with the Create New Effect Template icon.
There are some more control elements in the main menu of the Timeline Window: You can set up an Effect Template, delete Effects or Layers, enable or disable timeline snapping and much more. Use the filter option to select only certain content.
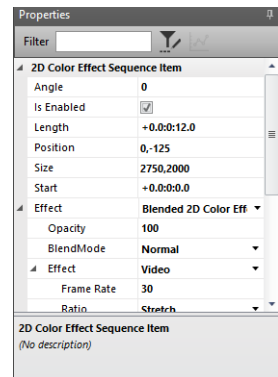
11.8　Adjust mapping to fixtures
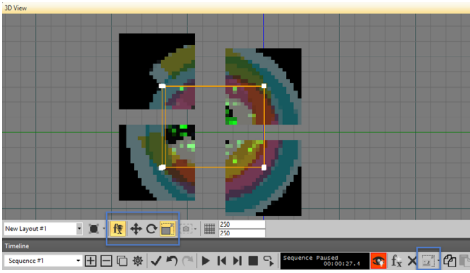
By default, an image, video or Effect will be mapped to the full Layout size of the fixtures. Adjust the mapping by selecting the object in the timeline and change the mapping. Use the icons below the preview to move, rotate or resize the Effect Rectangle.

Click the icon in the lower right to reset the mapping to the complete Layout

11.9　Cues

You can add Cues to the Timeline. Cues are control elements for fades or to control playing of content. Cues control fade ins and outs as well as playing states.

> If you know Cues from the Programmer of the LAS, do not be confused. Cues in SYMPHOLIGHT are control elements, not lighting scenes!
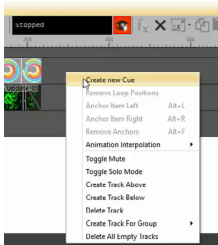
To add a Cue, right-click in a free area of a Layer, or right-click in the Cue track (the grey band between time markers and Layouts) and select *Create new Cue*, or double-click into the Cue track. A new Cue will be created. Click on the Cue and adjust its properties in the Properties window. By default Cues refer to the Cue settings of the Sequence.

When executing the project, you can use functions to jump to Cues forward or backward, or to continue from a definite point in the Timeline. An additional control element in Cues are Trigger Points. To add Trigger Points, right-click into a Cue and select Add Trigger Point. Later, in the Workflow Designer, you can react on such a Trigger Point.
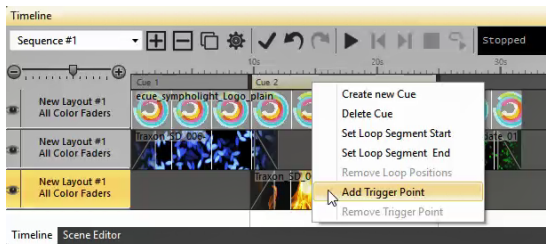
You can change the length of a Cue by dragging its end or start. There are three modes to do this. Without additional keys pressed, the length of a Cue changes and other Cues move. Keeping the Ctrl key pressed, the length of the affected Cues are changed, whereas the other Cues do not move. Keeping the Alt key pressed, also the Effects of the affected Layers in this Cue are moved. You may combine the Alt and Ctrl key.

11.10   Sequence and Cue Properties

To set the Sequence properties, click in an empty area of the Timeline or click the cogwheel icon in the Control Bar:
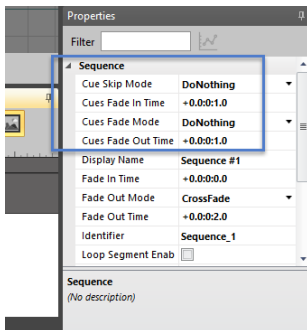
In the Sequence properties, the default behavior for all Cues is defined. This is the Skip Mode (DoNothing, CrossFade or FadeFromLastFrame), the fade in and fade out times and more. These default Cue properties are used if not defined otherwise. To change the Cue properties for a single Cue, click the Cue to select and change the properties in the Properties Window.

Important parameters for a Cue are *Play Mode* (Continue: continue playing, Wait: pause on end of Cue, Repeat: loop this Cue) and *Regular Fade Out Mode*, e. g. to realize a cross fade between two content elements in a Layer. Uncheck the box *Use Sequence Fade Times* to set fade times different from the times defined in the Sequence properties.

On the top of this window a small filter icon is shown. If selected, the properties window will show all properties relevant for this item. If deactivated, only the most important properties are shown.

11.11   Fading

Fading can be used in SYMPHOLIGHT for all elements in a Layer. There are various sources for fading in SYMPHOLIGHT. Here are the control possibilities for fades.

— Element-specific: Fade-In and Fade-Out are directly defined in the properties of the element.

— Fade-configuration: this is defined in the applications main settings: *Tools | Settings | Content*. This is the default fade time for all Effect and media Layer elements that get inserted. It is also the default setting after installation and enabled.

— Manual edits: at the beginning and the end of an element, there is a ramp. Keep the left mouse button pressed over the border and move the border. This will change the fade time, but will not change the applications settings.

11.12   Scenes and the Scene Editor

Scenes are the tool for use when you want to send DMX output to non-fixtures like DMX switches or if you want to define static DMX output to few fixtures. To create static content as Scene the Scene Editor is used. Scenes can be used in the All Color Faders/ Intensity Faders Layer like any other content. We show Scenes for the All Color Faders Layers here. Intensity Scenes work the same way, but with only one or few channels. To create a Scene in the Content window, click _Scenes_ in the bottom of the Media Library. As the first step you have to select the fixtures the Scene will be created for. Equivalent to the Setup procedure, click on a fixture to select it, use the Ctrl key while selecting to add or remove fixtures in an existing selection or draw a rectangle with pressed left mouse button.
Select the Scenen Editor at the bottom of the Timeline Window. While working with this Scene, the following commands are available.

	Create a new Scene for the selected fixtures. Rename the Scene in the name field. You can still modify the selection of fixtures in the Preview Window to add or remove fixtures to the selection.
	Save the Sequence.
	Revert to original Scene, discard changes.
%	If enabled, values are percent values, if disabled, values are Byte values.
	Live Preview on Stage: if this mode is enabled, the Content preview is sent as real DMX output to the fixtures. If this mode is disabled, no DMX output is generated and you edit in blind mode. This mode is only applied if the content tab is active! The status will be saved in the project file.
	Enable or disable the color picker to select colors.
	Show all fixtures in Scene. This overwrites to previous selection and includes all touched fixtures in the selection.

	Copy the current selected values to the clipboard.
	Paste the values from the clipboard to the selected channels.

You now have the Scene Editor with the selected fixtures for the Scene and you can assign color or intensity values to the complete fixture or to single pixels in the fixture. To open or close a fixture with more than one channel, click on the arrow left of the top fixture line.

You can remove (untouch) fixtures from the Scene with the cross icon on the left side of a fixture in the list. To place it again in the selection, click the fixture again in the Preview Window. Or click the _Show All_ icon to show all touched fixtures of that Scene.

You can copy and paste value settings from one element to another, or for several elements. When copying multiple elements, there may be no gap between them.
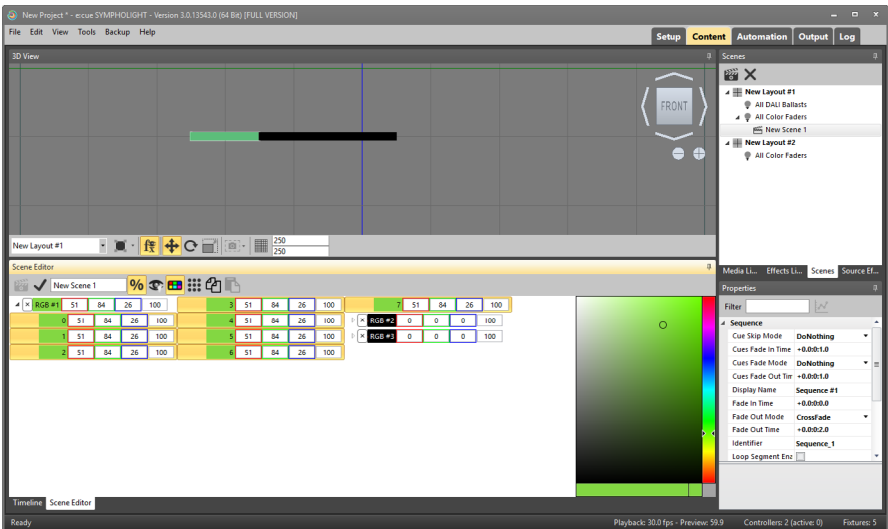
Select a complete fixture or a single channel, or use multi-select with a pressed Ctrl key. Now you can either assign colors to the complete fixture or to the channel with the color picker, or enter values for fixture or channel directly. After assigning value, click the check mark to save the Scene. Fixtures, which are not included in the Scene, receive their channel values from the content defined in the Sequence. Only the fixtures and channels used in the Scene receive their content from the Scene. The assigned colors are displayed in the Preview window and on the fixtures, if _Live Preview On Stage_ is activated.
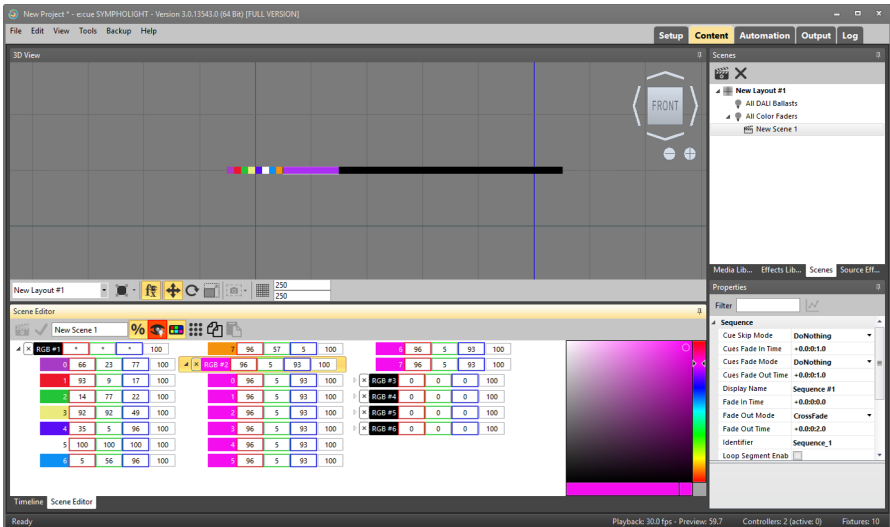
After saving the Scene, you can drag-and drop the Scene from the Scenes tab onto a Layer in the Timeline window like any other Layer content. Playing the content now shows the color settings of this Scene as content if the Scene is placed in an All Color Faders Layer.
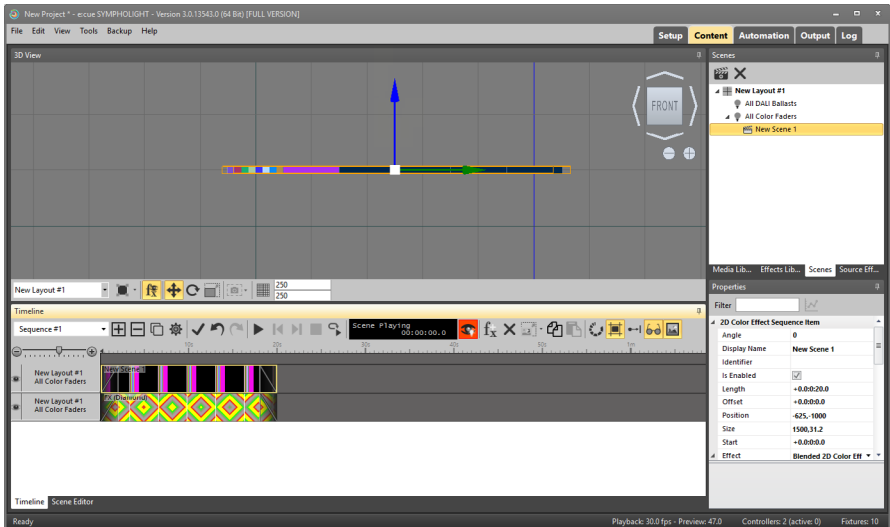
Use Cues for play control and fades as required. For Scenes in RGB Layouts, you can set the color channels and opacity. For Scenes in Intensity Layouts, you can set intensity

and Opacity. The Opacity defines how much of the Layers below is visible. A zero means full transparency, 255 or 100% means no transparency.

11.13   Keyframing

For certain properties of an effect or a media element you make properties dynamic.

This way, a property of an element can change over time. Not for all properties Key Framing is possible. Right-click on the property name, in the context menu either _Add keyframing_ is disabled (greyed out) or enabled. Select _Add keyframing_.
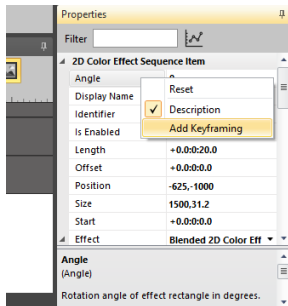
A sublayer below the selected Layer will be added, where the animation is defined. Two keyframes were already created: one for frame 0 and one keyframe for the last frame. Double-click into the keyframe sublayer to create a new keyframe. Click the new keyframe to edit its properties, time position and value, in the Properties Window. Right-click in the keyframe sublayer Right-click into the Keyframe Layer to select the animation mode. The context menu contains the following options:

— _Animation Interpolation | Step Interpolation_ will set the selected property directly in one step.
— _Animation Interpolation | Linear Interpolation_ will calculate intermediate steps for a continuous property change.
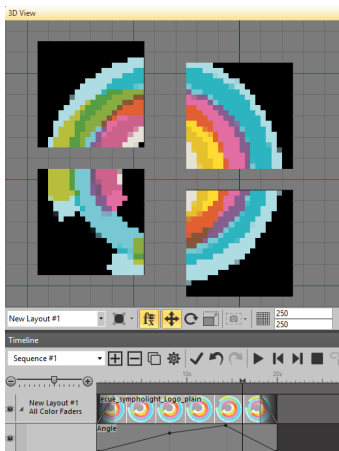
For angle animations, positive values are counterclockwise rotation, negative values are clockwise rotation.

The keyframe layer also has a mute button in the Layer header, to disable or enable the animation.

11.14   Working with Gradients

In many places in SYMPHOLIGHT, you can select Gradients, for example for Effects or for backgrounds or colors in the GUI Editor. Gradients are not a single color, but a span

of one or more colors over an area. A single pure color is handled as a special case of a Gradient.

11.14.1   The Gradient Editor
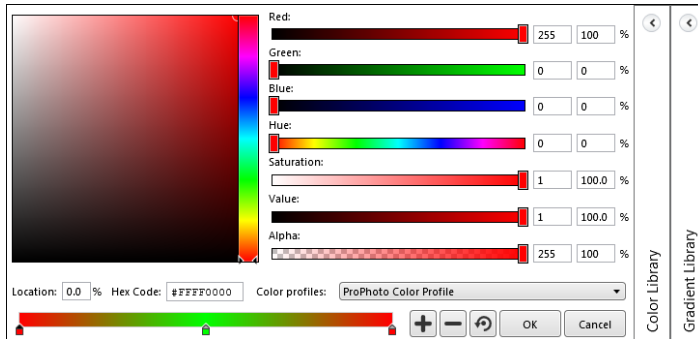
A single color can be defined either by picking a color in the left color picker, or editing the color and intensity values with faders on the right. The line at the bottom of the editor is the color span, indicating the Gradient Stops and is called the Gradient Preview.
The Gradient Stops define the points where color changes are placed. The faders have the following functions:

— Red/Green/Blue: the three color channels of a color.
— Hue: the color type.
— Saturation: deepness of the color.
— Value: intensity of the color.
— Alpha: amount of transparency.

11.14.2   Single color

Use a single Gradient Stop, use the minus icon right of the Gradient Preview to remove additional Gradient Stops. The position of the single Gradient Stop has no meaning now. Define the color now. Open the Color Library on the right to save this color as color template.

11.14.3   Gradient markers

Use at least two Gradient Stops. If Gradient Stops are not placed at the beginning or the end, the area in front or after will wear the same color as the Gradient Stop.
Add Gradient Stops with the plus icon right of the Gradient Preview. Click a Gradient Stop to select it, move it and set the color properties. Click another Gradient Stop to edit this one. Save this Gradient as a template in the Gradient Library.

11.15   Effects

Effects are an important part of SYMPHOLIGHT as a media application.

Effects are available as color or intensity versions and can be adapted by changing the properties of an Effect. SYMPHOLIGHT comes with a rich Effects Library of predefined, ready to use Effects. Also these Effects can be adapted by changing the properties of the Effect. You can create your own new Effects and save them for later use in your project. In the Effects Library, user defined and factory Effects are stored in separate trees. Preview icons give an idea what Effect is behind.

Under Source Effects there are the building blocks for creating new effects. Nat all Effects from the Source effects can be used in a Layer directly. Effects are saved in your system and can be exported and imported. User created Effects can be deleted, factory Effects cannot. To delete an Effect, right-click on the Effect and select _Delete Effect Template_. When new Effects are created or properties or a factory Effect are changed, save them as user-defined Effects. SYMPHOLIGHT will create a new preview icon when the Effect is saved. When placing Factory Effects in a Layer, nothing else is required except adapting the Effect properties.
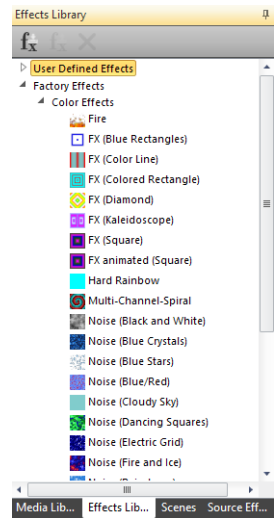
11.16   Creating new Effects

You can create new Effects from the tab Source Effects at the bottom of the Effects Library. To do this, you have to be aware of the way Effects are structured and composed in SYMPHOLIGHT. The two basic building blocks of Effects are Render Effects and Transformation Effects. Every Effect in Source Effects belongs either to Render Effects or Transformation Effects. Render Effects provide colors or intensities. Transformation Effects modify or structure colors or intensities.
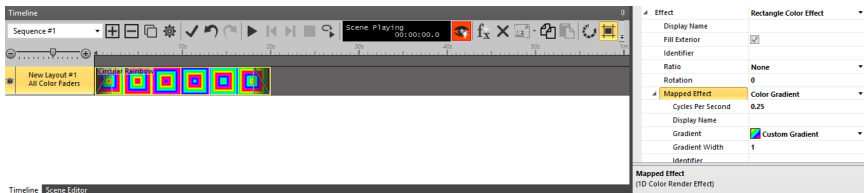
Most Effects consist of a Render and a Transformation Effect. In a few cases only a Render Effect can be enough, for example if you only want to use a constant color or a color chase. Both Effect types have properties, which can be changed. This has an influence on how the color rendering happens. The Transformation Effect Particle has

many properties like Fade Factor, Lifetime and Movement values. This modifies the transforming Effect. After creating or changing an Effect, you can save this Effect as a user defined Effect, using the Create Effect Template icon in the Timeline menu.

11.16.1   Example of a new Effect: Transformation plus Render
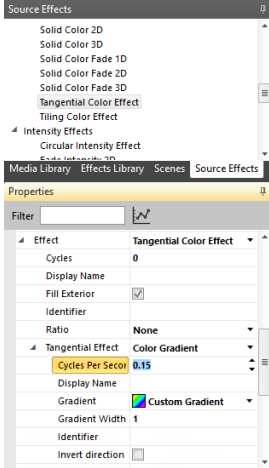
We will now create a new Effect.

Place the Multi-Channel-Spiral Effect from the Effects Library in a Layer. Use the Tangential Color Effect as Effect, the Tangential Effect in the Properties is shown as Color Gradient. Opening the Tangential Effect drop down menu provides the other possible Render Effects.

We use Color Gradient as Render Effect. Now the new Effect is complete, the preview of the Effect is shown in the Layer of the Timeline. You can edit now the Gradient properties by clicking the Gradient property for the Color Gradient Editor. While you change the Gradient, the Preview Window already shows how the final Effect will look like. Save the new Effect via the Timeline menu. Here we use Purple Spiral.

Additionally, the Effect property Cycles is set to 2.5, the Cycles Per Second to 0.15, giving more lines of the spiral.
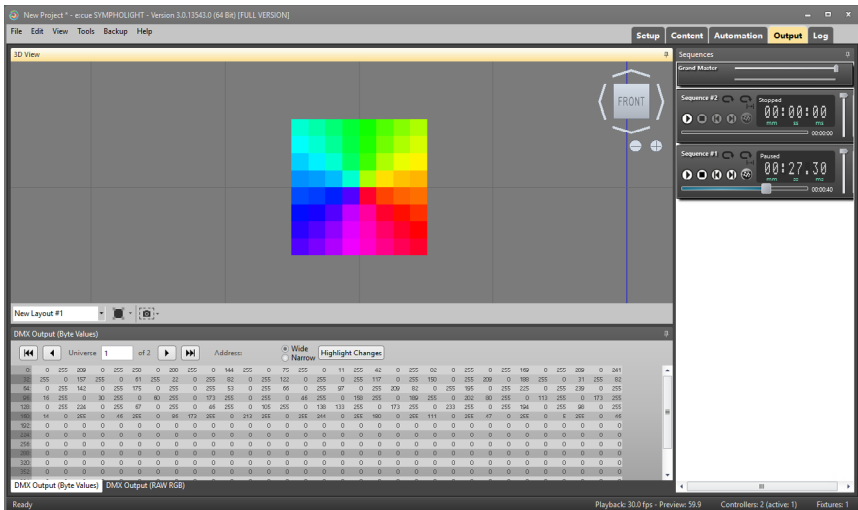
11.17   DALI Scenes

Also part of the Content Window is the Scene Editor for DALI Scenes. Select Scenes in the Media/Effects/Scenes Window on the top right and choose a device groupyou want to work on. Then switch to the Scene Editor in the lower part of the main window. Here you now can define DALI Scenes for the selected DALI group.
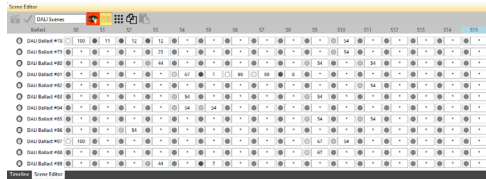
With the percent icon in the top left part of the Scene Editor, you can enter percent values or absolute values. When entering percent values, SYMPHOLIGHT recalculates these values to absolute values and the entered percent value may change to binary representation. Calling DALI Scenes for a DALI group is part of the Workflow Designer in the Automation Window.

> ⚠ The values for each DALI scene is stored inside the DALI ballast itself and due to the DALI standard limited to 16 scenes only. Therefore it is not possible to create more than 16 scenes per ballast.

11.18   Properties of content in Layers

The first six properties are common for all objects.

Angle	Rotate the effect(rectangle) in degrees.
Display Name	The name displayed as object name.
Is Enabled	Show the object in Preview and Live view. If disabled, the object is hidden, but still present.
Length	Run length in the Layer
Offset	An offset used at the beginning of media. Either enter a value or move the beginning of the media with a pressed mouse and Shift key.
Position	Position of center
Size	Size of the object in mm
Start	Start time in the Layer.

All other properties are specific to the content types like images, Effects or Cues.

11.19   Sequence properties

The behavior of a Sequence is mainly controlled with the Sequence properties. Also properties of Cues can be taken from the Sequence properties, or can be defined separately. Click into an empty area of the Sequence to set the properties of the Sequence or simply click the Sequence Properties icon (the cog wheel next to Delete Sequence) in the control bar.

Cue Skip Mode	Fade mode when using the skip functions. CrossFade: perform cross fade when entering next Cue. FadeFromLastFrame: use the last frame for fade.
Cues Fade In Time	The fade-in time for Cues, default = 1 second.

Cues Fade Mode	Fade mode in standard play. CrossFade: perform cross fade when entering next Cue. FadeFromLastFrame: use the last frame for fade.
Cues Fade Out Time	The fade-out time for Cues, default = 1 second.
Display Name	The name displayed as Sequence name.
Identifier	Unique identifier for this Sequence.
Loop Segment Begin	Define a loop segment begin position. Marked in the Cue track in the Timeline with a green marker.
Loop Segment Enable	Enable looping the defined loop segment.
Loop Segment End	Define the loop segment end. Marked in the Cue track in the Timeline with a red marker.
Loop Sequence	Loop the whole Sequence.
Mutual Exclude Collection ID	If not zero, a mutex group ID where only one Sequence may play at one time. If zero, no mutex grouping happens. Mutex is mutual exclusion, only one Sequence in a group may run at a time.
Priority	Render Priority: Sequences with higher priority overwrite values from Sequences with lower priority.
Start Position	Define the start position of the Sequence when triggered from the Automation or the Output Window.

11.20   Cue properties

Display Name	The name displayed as Cue name.
Fade In Time	The fade-in time.

Fade Out Time	The fade-out time.
Identifier	Unique identifier for this Cue.
Is Skip Stop Position	This is a stop position when using the skip functions.
Length	Length of the Cue.
Play Mode	Behavior during play mode. Continue: continue playing. Wait: wait at the end of the cue. Repeat: repeat this Cue.
Regular Fade Out Mode	Fade mode during regular play mode. CrossFade: perform cross fade when entering next Cue. FadeFromLastFrame: use the last frame for fade.
Skip Fade Out Mode	Fade more when using skip functions. CrossFade: perform cross fade when entering next Cue. FadeFromLastFrame: use the last frame for fade.
Use Sequence Fade Times	Ignore fade times from Cue, use fade times from Sequence properties.

11.21   Special function keys

[SPACE]	Using the space tab, you can toggle between play mode and paused.
Ctrl + C	Copy elements in the Timeline to the clipboard.
Ctrl + V	Paste elements from the clipboard to the Timeline.

12 Automation

12.1   Principles

In the Automation Window, you will define a Workflow for control and user interfaces. With Automation you prepare the project how it will be executed live and define how to react to external signaling, time-controlled execution, and necessary calculations. The Automation tab contains the Workflow Designer, the Scripting window, and the ActionPad Designer. With the Workflow Designer you can automate SYMPHOLIGHT regarding control, events, external signaling and status changes. With Scripting you can add functions and features of SYMPHOLIGHT by programmed code in C#. The ActionPad Designer is used to create graphical user interfaces for mobile devices or external web browsers.

As one of the main tabs, the Automaton tab is located at the top right corner with the Workflow Designer, the Scripting window, and the ActionPad Designer beneath.
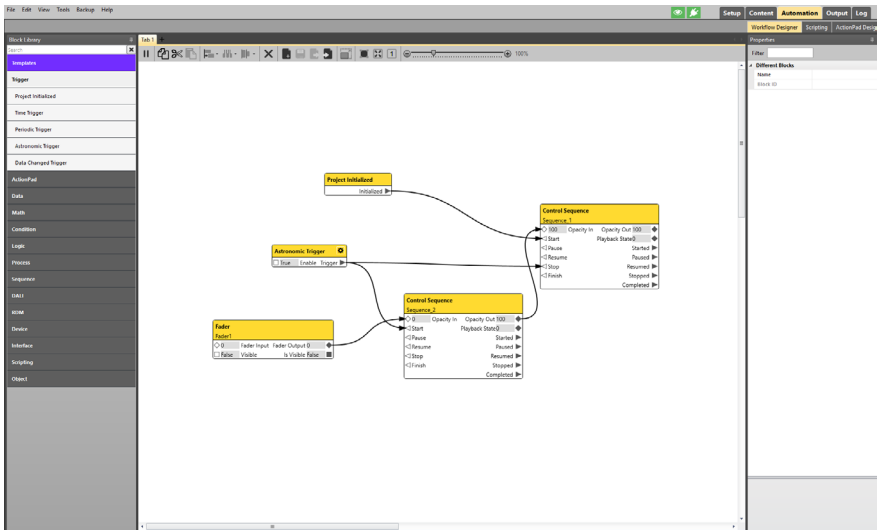
12.2   Workflow Designer

In the Automation window, you will define a Workflow for control and user interfaces. With Automation you prepare the project for how it will be executed live. Automation defines the reactions to external signalling, time-controlled execution, and necessary calculations.

Here is a simple example of a Workflow for controlling two Sequences in SYMPHOLIGHT with two astronomical triggers and one fader:
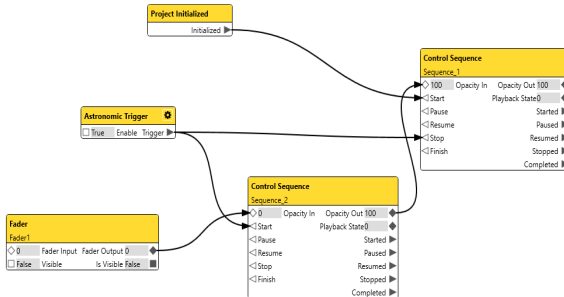
Sequence #1 gets started by a Trigger configured as "TwilightAm". Another Trigger is configured as "TwilightPm" and stops Sequence #1 and starts Sequence #2". Both Sequences are controlled with a Fader on the input Opacity, which is Brightness here.

12.2.1   User interface of the Workflow Designer
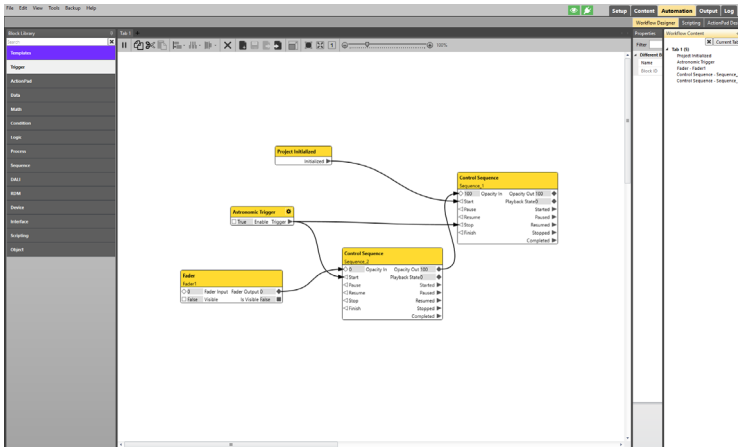
Block Library	Tab Control to structure the Workspace	Workflow Content	Overview of the Workspace
	Workspace with SYMPHOLIGHT Blocks		Properties

— The Workspace contains the Workflow. You arrange Blocks in the Workspace to
   create and edit the Workflow. Connect Blocks with a Link to create causalities. All
   causalities together form the Workflow. In the Workspace, you can move a selected
   Block. Multi-select is possible to move and edit more than one Block in the
   Workspace. Either draw a frame with the left mouse button pressed. Or if you keep
   the Ctrl key pressed while drawing a rectangle with the left mouse button in the
   Workspace, you select all included Blocks. In the Workspace, delete Blocks and
   Links.

— The Block Library contains all elements that can be placed in the Workspace, for
   example triggers, logical functions, Action Pad contents, or functions of devices like
   SYMPL dmx or dali Nodes. Blocks, that are already in use and can only have a single
   instance, like Action Pad buttons, Sequences or devices, have a check mark in front.

— In the Tab Control bar, you can create new tabs, create Templates, adapt the viewport
   with Fit or Fill, or you can zoom in and out manually. You can create as many Tabs
   and Templates as needed to have different areas for DALI, Sequences or functions. If

not one Block is selected, Fit zooms to the best possible view port. If one or several Blocks are selected, Fit zooms to the selected part of the Workspace. Fill always shows the complete Workspace.

— The Overview of the Workflow shows the current view port of the Workspace related to the complete tab. You can move the view port by dragging the yellow area. A right-click of the mouse opens a context menu where you can zoom in and out (Fit, Fill, 100%).

— The Workflow Content has all Blocks that are currently contained in the Workflow. The Blocks are sorted by the Tab (bold entry) to which they belong. Clicking an element here, or in the Workspace, makes the Block editable in its properties in the Properties Window. You can search in the Workflow Content to select only special Blocks in the list.

— The Properties Window provides the settings of selected Blocks for configuration.

Tab Control bar

❚❚	Suspend the currently running Workflow execution.
▶	Continue the currently running Workflow execution.
⎙	Copy selected Blocks, including connected Links. (Ctrl + C)
✂	Cut selected Blocks, including connected Links. (Ctrl + X)
⎙	Paste the copied or cut Blocks into the Workflow, including connected Links. (Ctrl + V)
▥	Align selected Blocks. Choose from the drop-down menu the direction of alignment: left, right, top, bottom, horizontal center, vertical center.
�ⅲ	Space selected Blocks evenly. Choose from the drop-down menu the axis of spacing: horizontal, vertical.

	Distribute selected Blocks. Choose from the drop-down menu the reference point of distribution: left edge, right edge, top edge, bottom edge, horizontal center, vertical center.
	Delete selected Blocks or selected Links.
	Create a new Template.
	Save a Template.
	Export a Template.
	Import a Template.
	Fit the canvas to the content. Any excess area is clipped.
	Optimize the viewport. The content of the Workspace is fit to the viewport.
	Fill the entire Workspace to the viewport.
	Scale the content of the Workspace to 100%.
	Zoom in and out of the Workspace (min 10%, max 400%).

12.2.2   Creating the Workflow

1. To create a Workflow, drag and drop a Block from the Block Library into the Workspace. Blocks are the main objects off which a workflow is made.
2. Configure the settings of the Block in the Properties Window.
3. Add Blocks to the Workspace as required.
4. Connect the Blocks with Links defining their causality, conditional statements and interdependencies. To create a connection, drag a Link: left-click on a PortOut

interface, keep the mouse button pressed and move the mouse cursor to the target Output.

5. Add Links as required.

The Workflow is executable at every state without an explicit command.

12.2.3   Connecting Blocks with Links

You connect the Blocks with Links defining their causality, conditional statements and interdependencies.

— To create a connection, left-click on a Port or Event interface, keep the mouse button pressed and move the mouse cursor to the target Port or Event. While doing this, rectangle indicates if this connection is allowed or prohibited. You can connect Ports of the same Port types indicated by icons of the same shape. When connecting Ports of Blocks, the data type and the value condition must be compatible. In some cases links between different port types are possible. Then an automatic conversion will be done.

	The green rectangle shows that the connection is allowed and ok.
	The connection is prohibited, because the Port types are incompatible .
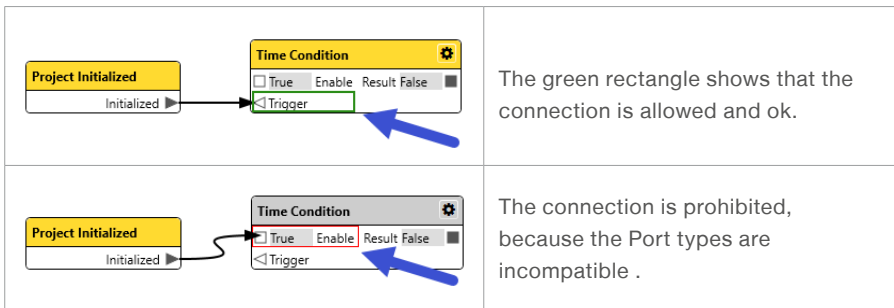
— You can change the color of a Link between ports. Click on the Link to see the properties of the Link in the Properties Window. Change the color of this Link in the Properties Window.

— To delete a Link, click on the Link to select it and press the Delete key.

12.2.4   Signal Flags

Signal Flags are Links that link different Block Ports in different Tabs. They can also be used instead of Links for a cleaner signal routing (better visual clarity) inside the same Tab. Signal Flags can connect one PortOut with more than one PortIn. You can either assign Signal Flags to a Port or Links, but not both types.

1. To create a Signal Flag, click on a PortOut. The PortOut is marked red.

2. Now press the Space bar on your keyboard. An text entry field opens with an auto-generated name. You can either change the name or press the Enter button and a Signal Flag gets created.

   Press the Enter key instead of the Space bar and the text entry field is empty. Enter a name and press the Enter button. A Signal Flag gets created.

Signal Flag names are unique. You can only create Signal Flag with a new, unused name.

3. Assign the Signal Flag to a PortIn: click on the PortIn. The Block can be in a different Tab.
4. Press the Enter key or the Space bar and a list of valid, already existing Signal Flags will be shown.
5. Select a corresponding Signal Flag and press Enter.
6. Assign the Signal Flag to more than one PortIns if needed.

The connection between PortOut and PortIn is established.
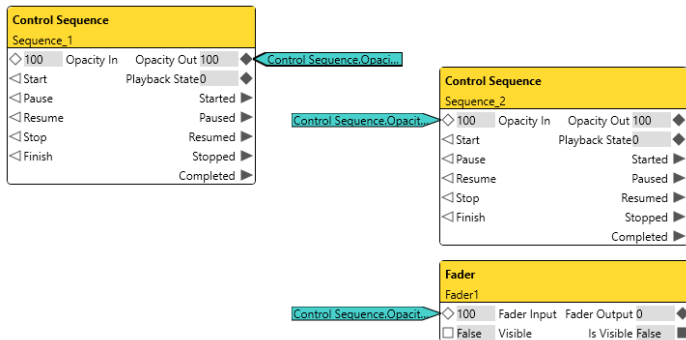
— You can edit the color and the name of a Signal Flag. To set the properties of a Signal Flag, select the Signal Flag with a mouse click. The properties of the Signal Flag are shown in the Properties Window and the selected Signal Flag flashes.
— Jump to the location of the counterpart Block of a Signal Flag. Click on a Signal Flag and a list with the connected counterpart Blocks appears. Double-click onto an entry of the list to jump to the connected Block.
— To show the full name of an abbreviated Signal Flag name, move the mouse cursor over the Flag.
— Signal Flags with a red border have no counterpart Blocks, means a Signal Flag has not been assigned to a PortIn.

⚠️ You can either assign Signal Flags or Links to an individual PortIn / PortOut.
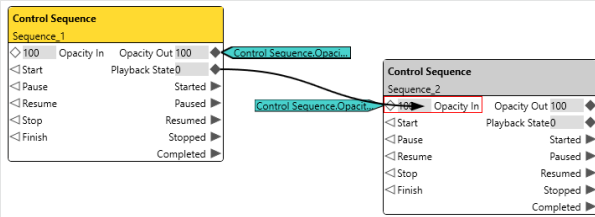
⚠️ Signal Flags disappear after changing the data type of one connected Block (e.g. from string to bool).

12.2.5   Working with Blocks

Typically, one Block offers one functionality as indicated by its name. The general structure of Blocks is the following:

Display name of the Block
Main setting

PortIns
(no filling)
receiving a value

PortOuts
(filled icons)
sending a value
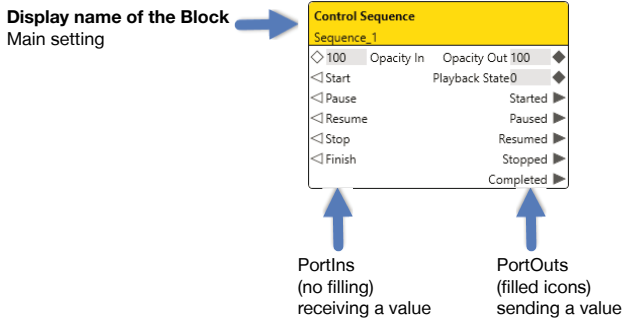
To connect Blocks, a Block needs at least one PortOut to forward its data as well as at least one PortIn to receive data. PortOuts are at the right side, the interfaces indicated by filled icons. PortIns are at the left side, the interfaces indicated by icons with no filling. Between receiving and forwarding data, the Block operates according to its settings.

Each interface of a PortIn and a PortOut is indicated by an icon. An icon states the property settings of the interface.

For example, the following Sequence Play Control Block has six PortIns and seven PortOuts:

A Port interface:
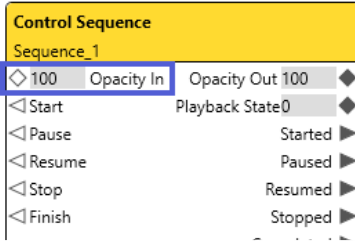
Icon   Data value   Port name
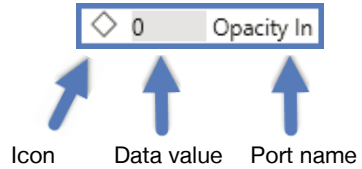
Additional buttons in Block headers:

⚙	Opens a dialogue for setting parameters of a Block.	
✏	Opens a tab for editing the Template Block.	

— To create a Block for the Workflow, drag and drop a Block from the Block Library into the Workspace.
Example of adding an Astronomic Trigger Block to the Workspace:
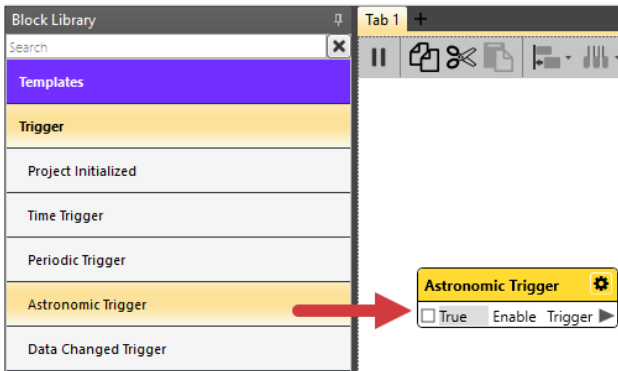
— Configure and adjust the settings of a Block for the Automation in your Workflow. See „12.2.6 Configuring the Settings of Blocks" on page 93.
— Rename a Block by changing its Display Name in the Properties window or double-click on the Block header.

— To delete a Block, click on the Block to select it and press the Delete key. Or click the Cut button in the Tab Control bar: ✂.

Target Based Blocks

Blocks that are target based can only have one bound instance per target. Targets are hardware devices and control elements in SYMPHOLIGHT like groups, Sequences, or ActionPad elements.

In the Block Library, a circled number at the right side indicates the amount of available instances for target based Blocks when the related target is added to the project (Setup Tab):

Example for three existing Sequences:
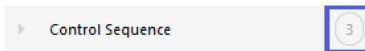
The Block Library also contains additional Blocks for each of the targets when the related target is added to the project (Setup Tab).

Example: A Glass Touch T12 is added to the project:

1 Glass Touch T12 in show

Bound block

Add either the target based Block to the project or an unbound Block. Unbound, orphan Blocks can be added without restriction and pre-configured to some degree, but are not executed. Orphan Blocks are indicated by the grayed out header.

Example of a DALI Group block when not one DALI groups exist in the project:

Bind an unbound Block to a target by defining the Target Value in the Properties Window of the Block. A device has to be in the project (Devices Window in the Setup Tab - "Attached devices in the project") to be available as Target Value. When in Trial Mode, attached devices cannot be added to the project.

To undo the binding, delete the Target Value in the Properties Window by clicking the cross button or selecting "<not set>" in the drop-down menu.

The name of Target Values are always the Identifier of the targets.

Please note that the PortIn values of target based Blocks are set by default: actual settings within the objects/ targets are not retrieved, not displayed in the Block and might differ from the values set by default.

12.2.6   Configuring the Settings of Blocks

Configure the settings of a Block to adjust the Block for the Automation in your Workflow:

— Select the Block to access its settings in the Properties Window. To select a Block, left-click on the Block in the Workspace or in the Workflow Content.
— Adjust the settings of the Block in the Properties Window.

Multi-selection of Blocks is possible to set identical properties at one blow:
— To multi-select, press the Crtl key and left-click on the Blocks. Or draw a frame with the left mouse button pressed. The Blocks included in the frame get selected. Both ways are available in the Workspace as well as in the Workflow Content.
— Adjust the identical settings of the selected Blocks in the Properties Window.
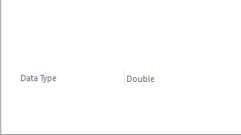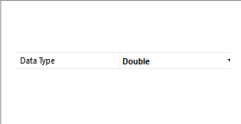
There are generic properties and Block specific properties. The generic properties are listed below. For a description of each Block and its properties see „12.3 Workflow Blocks in Detail" on page 113.

List of Generic Block Properties

Display Name	Name of the Block. Displayed in the top row of the Block's header.
Block Position X	X-Position of the Block in the Workspace.
Block Position Y	Y-Position of the Block in the Workspace.

Accessability of Properties

Properties can be either fixed or selectable. Fixed properties are grayed out, selectable properties are bold:

Fixed property	Grayed-out	Not changeable. This is a Port that has been given a constant value.	Data Type Double
Selectable property	Bold	Use drop-down menu, counting arrows or type to change the value.	Data Type **Double**

A red framed Block or Property entry indicates an invalid setting. The block cannot be executed. Please adjust the property settings.

Example: The formula of the Math Formula block (Advanced Math) is incomplete:
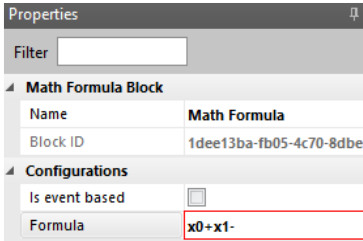
Please note: To apply defined values press the Enter key on the keyboard or select another property in the Properties Window.

12.2.7   Setting Port Types and Modes

Each Port has types and modes assigned. These are:
— Data types (see „12.2.8 Data Types for PortIn and PortOut Values" on page 96)
— Input types (see „12.2.10 Setting Input Types for PortIn Values" on page 97)
— Trigger (see „12.2.11 Adding an Event Based Trigger" on page 98)
— Port Modes (see „12.2.12 Setting Port modes" on page 99).

The assigned types and the Trigger are graphically indicated by the Port icons:

Data type	Shape of the icon.	E.g.
Input type: Property	Bold framed icon.	E.g.
Trigger	Triangle icon.	

12.2.8   Data Types for PortIn and PortOut Values

The following data types are supported in SYMPHOLIGHT's Workflow Designer:

Data type	Port Icon (PortIn / PortOut)	Explanation: "example".
Bool		A bool value, equals True or False: "True".
Double		A 64 bit float value.
DateTime		A value representing date and time with high granularity.
Dynamic		Any data type is compatible.
HdrIntensity		(High-dynamic-range Intensity) Tuple of Intensity and Alpha values with 32 bit float precision. Used for Intensity Effects.
Int64		A 64 bit signed integer value.
LogLevel		Values: Warning, Debug, Message, Error. Used in the Write Logbook Message Block.
ProPhotoColor		Tuple of RGB and Alpha values with 32 bit float precision. Used for Color Effects.

String		A string of UTF-16 characters: "Text".

When connecting Ports of Blocks, the data types must be compatible. A rectangle indicates if a connection is allowed or prohibited.

12.2.9   Setting Data Types of Ports

To configure the data types of Ports of a Block, select the Block. In the Properties Window, adjust the data types of the Ports. If data types of Ports are fixed, this is visible when the Properties in the Property Window are grayed-out.

12.2.10   Setting Input Types for PortIn Values

A special type of PortIn is the Property Port. A Property Port does not receive a value, but has a fixed value. As such, a PortOut cannot be connected to a Property Port with a Link. A Property Port is indicated with a bold frame.

Input type: Port	Regular icon.	E.g.
Input type: Property	Bold framed icon.	E.g.

For some PortIns in some Blocks, you can switch from Port to Property:
— Look for the property "Input Type" in the Properties Window.
— At the right column, select from the drop-down-list or type "Property".
  Now, the Port is marked with a bold circle, the data type is adjustable, and the property "Port Mode" is replaced by "Value" for setting a fixed value.
— Configure the data type and set the value as desired.
The PortIn has now a fixed value that is used for the Block's calculation.

Example:
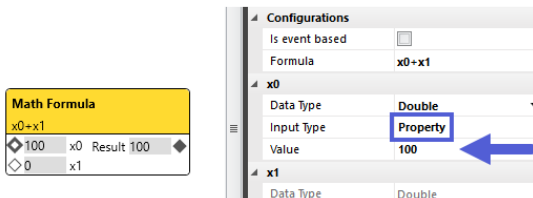
Here for the Math Formula Block, the value x0 is set to a constant of 100. The result will be 100 + x1, like the Formula says.

A Port with a "Property" input type is accompanied by a Value property instead of a Port mode property.

12.2.11   Adding an Event Based Trigger

A special type of Port is the event based Port. Instead of a specific value, a Port can send or receive a trigger. An EventIn receives a trigger and not a value. An EventOut sends a trigger and not a value.

	An EventIn input. Receives a trigger or an event from another operation.
	An EventOut output. Sends a trigger or the event from an operation.

If Is event based is checked, the Block gets an additional EventIn and creates the result for the output only when the block gets a trigger on the "Run" input. Example: Math Block without (left) and with (right) checked Is event based.
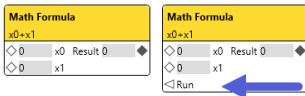
For some Blocks, you can add an event based Port to trigger the execution of the Block:

— At the property "Is event based", check the box for "Value" to enable triggering. An event based PortIn (EventIn) is added to the Block.
Example of Value Block without the Event based Trigger:
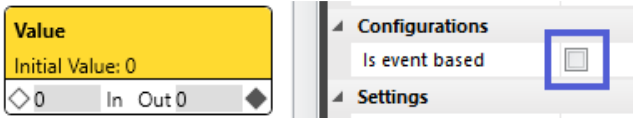
Value Block with the enabled "Is event based" property and the additional EventIn:
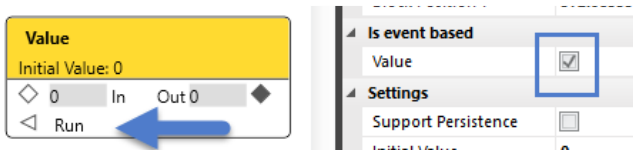
When connecting Ports of Blocks, the value condition must be compatible.
Setting a PortIn to event based switches the Port mode from passive to active.
Is a Block event based, the Block only executes on an external trigger event.

— You can manually trigger EventOut Ports. Right-click on an EventOut and click the "Fire Event" context menu entry.
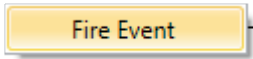
The defined Trigger is sent instantly to the connected EventIn.

> You can use this method to test a part of the workflow independently of preconditions.

12.2.12   Setting Port modes

A Port can be either active or passive. In general, "active" means that the Port refreshes its data values constantly, or when input values change, regardless of what the receiving Port expects whereas "passive" means that the data only gets updated if explicitly ordered to do so. A Port is passive, when it requires a request from its counterpart and then creates the result.
Example:
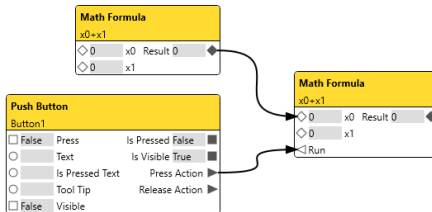
The upper Math Block has a Passive PortOut, the lower Math Block has an Active PortIn. The lower Block is also set to Is Event Based. If the button on the Action Pad is pressed, the lower Math Block will now request the result from the upper Math Block, calculate its own result and send it to its PortOut. Set the sending Port as active and the receiving Port as passive. This mechanism is helpful when you want to avoid permanent load from the Blocks.

Important is the combination of linked Port modes and to consider the executing block:

PortOut of block A - block A is executing	PortIn of block B	Description
active	active	Data value is forwarded, block B is not triggered.
active	passive	Data value is forwarded, block B gets triggered to execute.
passive	active	Data value stays in the PortOut of block A until PortIn of block B fetches the value.
passive	passive	Do not use: No data flow.

PortOut of block A	PortIn of block B - block B is executing	Description
active	active	No action since data value is up-to-date.
active	passive	Block B executes with existing data value.
passive	active	Block B triggers block A to run again and forward the new data value.
passive	passive	Do not use: No data flow.

— To change the Port mode, go to the property "Port Mode", select from the drop-down-list or type "Active" respectively "Passive".
— Set a Port mode to Passive to avoid permanent evaluation of blocks.

By default, PortIns are passive and PortOuts are active. Setting a PortIn to event based, switches the Port mode from passive to active.
Take care to only connect a single Link to an active PortIn. Otherwise you cannot distinguish with which incoming value the block calculates.
If Port modes are fixed, this is visible when the Properties in the Property Window are grayed-out.

12.2.13   Enabling and Disabling Blocks

Many Blocks have a PortIn named Enable and a PortOut named Is Enabled. By default, these Ports are set to TRUE: Any Block is enabled and executed when placed in the Workflow. Sending the bool value "FALSE" from a linked counterpart Block to the Enable PortIn of a Block disables the Block. The Is Enabled PortOut will forward the value "FALSE" to a connected block.

Example: A Write Logbook block creates entries in the Log as its initial Enable value is "TRUE". A Value Block can prohibit the Write Logbook block to create entries in the Log. Set the data type of the Value Block to Bool. The default initial value of the Block becomes FALSE. Connecting the Out PortOut of the Value Block to the Enable PortIn of the Write Logbook block disables the Write Logbook Message. No Log entry is created any longer.
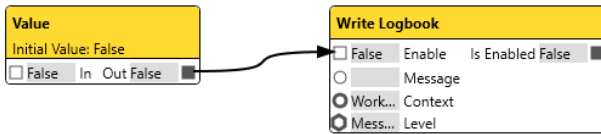
12.2.14   Adding Tabs, Deleting, and Renaming

A Tab contains one Workspace. You can create as many Tabs as needed to have different areas for DALI, Sequences or functions. The least number of Tabs is one. All Tabs together build the Workflow. You can connect Blocks of different Tabs using Signal Flags (see „12.2.4 Signal Flags" on page 89).

— To create a new Tab, click the Plus button above the Tab Control bar: ✚ . By
default, on opening the Workflow Designer one Tab is already present:
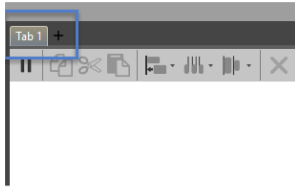
— To delete a Tab, right-click on the Tab. In the context menu click "Delete":
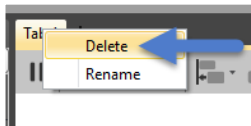
To rename a Tab, right-click on the Tab. In the context menu click "Rename". Enter
the new Tab name into the entry field and click the "Ok" button.

12.2.15   Adjusting the View

You can adjust the view of the Workspace to your needs.
— Zoom in and out using the Tab Control bar:

	Click to fit the canvas to the content. Any excess area is clipped.
	Click to optimize the viewport. The content of the Workspace is fit to the viewport.
	Click to fill the entire Workspace to the viewport.
	Click to scale the content of the Workspace to 100%.
	Zoom in and out of the Workspace using the fader (min 10%, max 400%).

— Move the viewport of the Workspace using the Overview of the Workflow: drag the yellow rectangle of the viewport pressing the left mouse key to the desired area and release the left mouse key.

12.2.16   Adjust arrangement

You can rearrange the Blocks of the Workspace at any time for a better visual clarity.
— Align and space evenly and distribute selected Blocks using the Tab Control bar:

	Click to align selected Blocks. Choose from the drop-down menu the direction of alignment: left, right, top, bottom, horizontal center, vertical center.
	Click to space selected Blocks evenly. Choose from the drop-down menu the axis of spacing: horizontal, vertical.
	Click to distribute selected Blocks. Choose from the drop-down menu the reference point of distribution: left edge, right edge, top edge, bottom edge, horizontal center, vertical center.

12.2.17   Search for Blocks

You can search the Workflow Designer for Blocks. Use the Block Library to search the whole range of provided Block items. Use the Workflow Content to search for Blocks utilized in the Workflow, independently of the Tab.
— Using the Block Library for Block items, enter a search text into the entry field at the top of the Block Library window. You can search for Blocks and for Block categories. The Block Library displays the Block categories of the matches. Clear the search text by clicking the button at the right of the entry field:
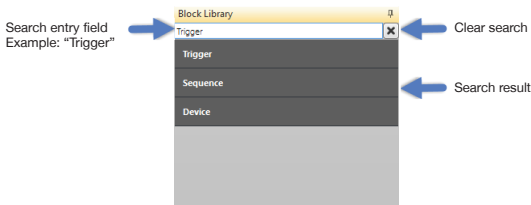
— Using the Workflow Content for utilized Blocks, enter a search text into the entry field at the top of the Workflow Content. The Workflow Content displays the matching Blocks. Clear the search text by clicking the button at the right of the entry field:

— Filter the displayed Blocks of the Workflow Content to display only the Blocks of the currently shown Tab in the Workspace. Click the "Current Tab" button at the right of the Search entry field.

Search entry field   Current Tab filter

12.2.18   The Block Library

The Block Library is separated into various function areas.
A quick overview is presented in this chapter. For more information on each Block see the following chapter „12.3 Workflow Blocks in Detail".

> 💡 When editing properties of Blocks, details and more information for each property are shown in the help window below the Property Window..

Templates	
User Defined Templates	See „12.3.1 Adding Templates" on page 113 for creating and using Template blocks.
Input	Use this Block if you would like add input ports to your template. Available in Template Tab only.
Output	Use this Block if you would like add output ports to your template. Available in Template Tab only.
Trigger	
Project Initialized (p.115)	Use this Block to execute actions as soon as the project file has been loaded, e.g. Use this trigger to start a sequence if the system is started.

Time Trigger (p.116)	Use this Block to react on time events, like every working day at 7am.
Periodic Trigger (p.118)	Use this Block to trigger actions periodically, e.g. to execute a script periodically.
Astronomic Trigger (p.119)	Use this Block to react on astronomic events, like Dawn, Dusk, Sunrise and Sunset.
Data Changed Trigger (p.123)	Use this Block to react on any changes that are related to a dedicated port, e.g. if a Sequence Opacity changes.
ActionPad	
Push Button (p.124)	Use this Block to use button press and release events to trigger elements in the Workflow Designer, e.g. if you want to start a Sequence from an Action Pad Button.
Toggle Button (p.128)	Use this Block to use toggle on and off events to trigger elements in the Workflow Designer, e.g. if you want to start and stop a Sequence from Toggle Button.
Image (p.132)	Use this block to set the image item.
Text Label (p.138)	Use this Block if you want to change the text of Action Pad labels from the Workflow Designer, e.g. if you want to display different messages on the Action Pad depending on the daytime.
Fader (p.136)	Use this Block if you want to set or receive values of Action Pad Faders in the Workflow Designer, e.g. if you want to control the Opacity of a Sequence.
Inline Frame (p.138)	Use this block if you want to change the URL for the Frame content.
Color Box (p.140)	Use this block to set color.
Color Wheel (p.140)	Use this block to set color.

Page Navigation (p.143)	Use this Block if you would like to switch to a certain Action Pad Page on all devices that are actually showing the corresponding Action Pad, e.g. if you want to show a specific page in case an emergency comes up.
Data	
Value (p.292)	Use this Block to add a constant value to your workflow, e.g. you want to trigger a scene of a DALI Group.
State to Event (p.148)	Use this Block to react on bool state changes, e.g. if a value changes from TRUE to FALSE.
Event to Value (p.149)	Use this Block if you want to output statically defined values on an event.
Combine Text (p.151)	Use this Block to combine two or more texts.
Value to Event / State (p.153)	Use this Block to compare Strings and Values that have been received from e.g. external sources and trigger actions based on the input received.
Converter (p.156)	Use this Block to convert data types. Since v3.0 this block is obsolete as the Workflow Designer Blocks are converting data types automatically.
Permit Event (p.158)	Use this Block to decide whether to pass an incoming event or to block it.
Math	
Math Formula (p.159)	Use this Block for advanced mathematical functionalities, e.g. combine arithmetic calculations.
Addition (p.177)	Use this Block to add two or more values.
Subtraction (p.163)	Use this Block to subtract two or more values.

Multiplication (p.165)	Use this Block to multiply two or more values.
Division (p.167)	Use this Block to divide two values.
Moving Average (p.169)	Use this Block if you want to create a moving average value based on a number of received values or based on values that have been received within a certain time.

Condition	
Time Condition (p.171)	Use this Block to react on time events. Use this Block if you want to limit the execution of other triggers to a specific time range, e.g. only allow to switch on the lights if it is after 7am.
Compare Condition (p.173)	Use this Block to compare to values with each other.
State Condition (p.175)	Use this Block if you would like to switch between two different scenarios depending on one single value, e.g. change the text of a button if another button has been pressed.

Logic	
AND (p.177)	Use this Block if you want to combine two or more conditions: only if all inputs are TRUE, the output will be TRUE.
OR (p.179)	Use this Block if you want to combine two or more conditions: if at least one input is TRUE, the output will be TRUE.
NOT (p.181)	Use this Block to negate a boolean value: if the input is TRUE, the output will be FALSE and vice versa.

Process	
Threshold (p.182)	Use this Block if you want to turn off the light if the light sensor value is above a threshold level and to turn on the light if the sensor value is below.
Fade (p.184)	Use this block to fade between two values.

Measure Compare Adjust (PID) (p.188)	Use this Block if you want to adjust the fixture light level depending on a certain sensor target value, e.g. in Daylight Harvesting scenarios.
Human Centric Lighting (p.192)	Use this Block to create human centric lighting features like defining a color temperature curve depending on the time of day.
Toggle (p.197)	Use this Block to achieve a toggle functionality, e.g. first button press: light on, second button press: light off.
Single Delay (p.198)	Use this Block to react on an event a certain time after it has been triggered.
Multi Delay (p.199)	Use this Block to react on an event a certain time after it has been triggered. Multiple delays can be specified.
Sequence	
Control Sequence (p.204)	Use this Block to control the opacity and play state of a sequence or to react on play state changes, e.g. to start, stop or pause a Sequence.
Adjust Sequence (p.206)	Use this Block to control some advanced parameters of a Sequence, e.g. the priority value of a Sequence.
Control Cue (p.208)	Use this Block to control Sequences based on cues, e.g. skip to the next or previous cue or react to completed cues of a Sequence.
Trigger Point (p.210)	Use this Block to react on Trigger Points that have been assigned to a specific Sequence. Whenever the Sequence play time is passing a trigger point an event will be fired and the number of crossings is available.
Generic Control Sequence (p.211)	Use this Block to control multiple Sequences at once and get notifications of them at one place.
Mutex Group (p.213)	Use this Block to control Mutex Groups, e.g. to jump to the next Sequence within the same Mutex Group.

DALI	
DALI Line (p.216)	Use this block to control DALI ballasts of a line in the direct way by ballast address, group address, and Broadcast. Requires no ballast matching in the Setup Tab.
DALI Group (p.225)	Use this Block to control your DALI Groups. You can also configure Daylight Harvesting.
DALI Button (p.238)	Use this Block to include DALI buttons in the Workflow Designer, e.g. to switch on/off lights based on button presses.
DALI Presence Sensor (p.240)	Use this Block to include DALI presence detection sensors in the Workflow Designer, e.g. to switch on/off lights based on presence information.
DALI Light Sensor (p.241)	Use this Block to process DALI light sensor values in the Workflow Designer, e.g. for Daylight Harvesting.
DALI Observer (p.242)	Use this Block to process DALI information that can be collected from DALI fixtures.
RDM	
Build RDM Command (p.308)	Use this Block to build simple RDM commands that can be send to RDM devices using the Send RDM Block or scripting.
Send RDM (p.248)	Use this Block to send RDM commands to specified destination remote devices.
Observe RDM (p.250)	Use this Block to process RDM information that can be collected from RDM fixtures, e.g. temperatures.
Device	
Analog / Digital Contact (p.255)	Use this Block to react on the values of analog inputs of e.g. the SYMPL input Node and process the values in the Workflow Designer.

Dry Contact (p.259)	Use this Block to receive dry contact triggers from, e.g. the LCE2 or the Butler XT2.
Relay (p.262)	Use this Block to control the relay outputs of e:cue devices, e.g. the SYMPL relay Node.
Observe Controller (p.264)	Use this Block to control and fetch information from e:net devices.
Observe System (p.265)	Use this Block to retrieve various information about the running instance and the project.
Receive DMX (p.266)	Use this Block to receive DMX values from an external system, like a SYMPL dmx Node or a SYMPL e:pix Node, and process them in the Workflow Designer, e.g. to map a DMX channel value to a Sequence Opacity.
LCE2 Display (p.268)	Use this Block to show a specific message on the display of the e:cue Lighting Control Engine 2 devicem or the LCE3.
Glass Touch T6 (p.270)	Use this Block to control your installation with a Glass Touch T6 User Terminal.
Glass Touch T6R (p.274)	Use this Block to control your installation with a Glass Touch T6R User Terminal.
Glass Touch T12 (p.281)	Use this Block to control your installation with a Glass Touch T12 User Terminal.
Glass Touch IR (p.282)	Use this Block to process IR commands from a remote control that are received by any Glass Touch User Terminal. Make sure the IR Remote supports RC5.
SymplTrace Controller	Use this Block to control a SymplTrace powered luminaire.
SymplTrace Sensor	Use this Block to retrieve values from a SymplTrace Sensor.
SymplTrace Tag Trigger	Use this Block to receive events if a Bluetooth LE Tag is in range or out of range.

Interface	
Send E-mail (p.286)	Use this Block if you want to send emails with customizable messages.
Send UDP (p.315)	Use this Block to send UDP network triggers to a given IP address of external systems, e.g. with a Constant Value Block.
Receive UDP (p.292)	Use this Block to receive external UDP network triggers that can be processed in other blocks, e.g. the Value to Event / State Block.
Send & Receive Serial (p.295)	Use this Block to communicate with external systems using serial ports, like the SYMPL serial Node or a Local Serial Port.
Write Logbook (p.289)	Use this Block to write a Logbook Message to the Log Tab.
Send to Network (p.315)	Use this Block to send data from configured ports to the network.
Receive from Network (p.304)	Use this Block to receive data from the network and dispatch it to configured ports.
Scripting	
Script (p.306)	Use this Block to add Scripts to the Workflow Designer, e.g. if you want to start Scripts based on Action Pad button presses.
Object	
Set Property (p.307)	Use this Block if you want to change the value of an object property. An object can for example be an opacity value of an effect or the master fader. Example: assign a fader on an Action Pad page to the Grand Master Fader (Target Object: ProjectInformation, Target Property: Opacity).

Get Property (p.308)	Use this Block if you want to get the value of an object. The Block gets a property from an object and provides it as value. An object can for example be an opacity value of an effect or the master fader. Example: fetch the color selection from a Color Picker on an Action Pad page and provide the color value as ProPhotoColor value.
Check Property (p.312)	Use this Block if you want to compare the value of an object property. The Block compares a property of an object against a value. It provides a bool value as result from the comparison. An object can for example be an opacity value of an effect or the master fader.
Call Method (p.314)	Use this Block to call a function of an object. Example: Play/Pause/Stop of a Sequence, or Cue Control for the Sequence.

12.3   Workflow Blocks in Detail

12.3.1   Adding Templates

A special type of Blocks are Templates. With Templates you can design your own customized Block. You can also use Templates to create a complex Workflow logic once (Master Template) and then multiply it effortlessly (Template instances). A template Block consists, as a usual Block, of PortIns, EventIns, PortOuts, and EventOuts. Additionally, a Templates has its own internal logic that is created by an own Workflow, consisting of interconnected Blocks from the Block Library.

Create a Template (Master Template)

Create a Template in the Workflow Designer:

1. Click the button in the toolbar of the Workflow Designer to create a Template. A new tab with an empty Workspace opens. Newly created Templates are always Master Templates. You can rename the Template over the context menu.
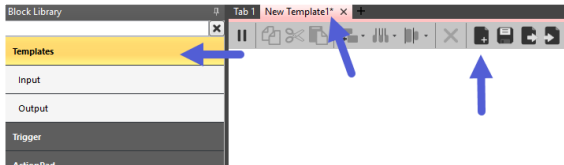
2. Go to the Templates category in the Block Library. You can add Inputs and Outputs in the shape of Blocks to the Template Workflow as desired which will act as Ports in the Template instances. Configure the Input and Output blocks as you need them as Ports.

3. Add Blocks from the Block Library as desired and link them to design the internal logic of the Template.
   Please note that target based Blocks are always orphan in Master Templates. Target values can only be assigned in Template instances, using either the Properties window of the selected Template instance or the editing tab.

4. Click the button in the toolbar of the Workflow Designer to save the Template.
When switching from the Template tab to an ordinary tab, the Block Library now contains the newly created Template (category "Templates").

You can also import ( ) and export ( ) Templates. The file format for Templates is .*slwt (SYMPHOLIGHT Workflow Template).

Use a Template (Template instance)

To use a Template in the Workflow after creating the Master Template, an Template instance is automatically created:

1. Go to the Templates category in the Block Library of the Workflow Designer.
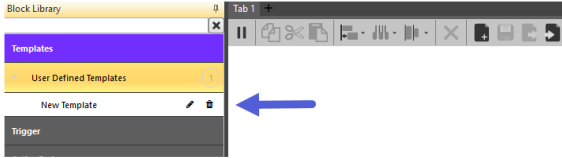
2. Drag and drop a Template into the Workspace like any other Block. The Block created in the Workflow is an instance of the Template. Assign target values to target based Blocks where present.
3. Integrate the Template into your Workflow by defining its properties and linking it with counterpart Blocks.

Edit Templates (Master or instances)

You can edit either a Master Template or a Template instance itself. Changes on an instance only apply to this very Template block as it is deployed in the Workflow. Changes on a Master Template apply to all henceforth deployed Template instances of this Master Template - but not retroactive.

— Master Template editing: access over the ✏ icon in the Block Library.
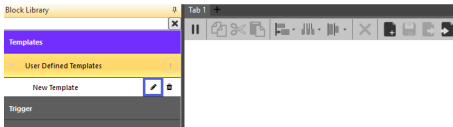
The Template content of the Master is opened in another tab of reddish color, with the Input and Output blocks again available in the Block Library. You need to save the changes by clicking the 💾 button in the toolbar.

— Template instance editing: access over the ✏ icon in the Template block in the Workspace.
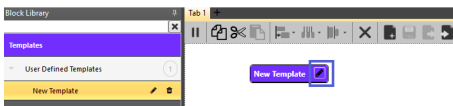
The Template content of the instance is opened in another tab of purple color, with the Input and Output blocks again available in the Block Library. The changes are applied instantly without saving them.

12.3.2   Project Initialized

Use this Block to execute actions as soon as the project file has been loaded, e.g. use this trigger to start a sequence when the system is started.

The Block has one EventOut for the trigger when the project is initialized.

— No configuration needed.

12.3.3   Time Trigger
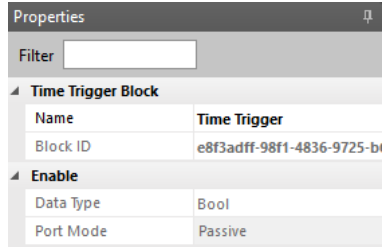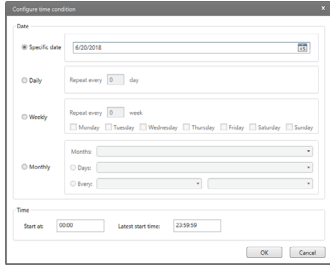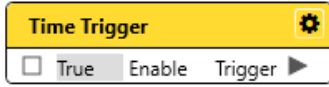
Use this Block to react on time events, like every working day at 7am.

The Block has one EventOut for the trigger and an Enable PortIn. Extended configuration is accessible at the cogwheel icon. The trigger configuration is displayed as main setting in the Block's header.

— Define date and time of the time triggers in the extended configuration window.

Category	Description
Enable	A passively received bool value to dis- or enable the Block.

Defining date and time triggers in the extended configuration window
Define at what date and at what time the trigger is scheduled in the extended configuration window:
1. Click the cogwheel icon at the right top corner of the Block to open the extended configuration window: .
2. Select one of the four date options at the Date frame:

   Specific date: Use the calender at the icon or enter manually a particular day for a trigger.
   Daily: Enter a number for the daily interval in which you want to send a trigger, e.g. enter "2" for triggering every other day. Compared to the "0" with immediate effect, when entering a "1", the day count starts with the day after configuring the Time Trigger block.

Weekly: Enter a number for the weekly interval in which you want to send a trigger, e.g. enter "1" for triggering every week. Compared to the "0" with immediate effect, when entering a "1", the week count starts with the Monday of the next week at the time of configuring the Time Trigger block. Further select the weekdays.
Monthly: Select one or multiple months. Also enter specific days or weekdays. The Trigger configuration is of immediate effect. You will have to set at least the days in the month, otherwise no event will be fired.

3. Define a time span at the Time frame. SYMPHOLIGHT scans in the given time span if the configured trigger has been sent, e.g. at a belated project start. If not, the trigger will be sent immediately.
   Latest start time: The latest time that the trigger will fire if the Enable EventIn changes in this time span from False to True.

4. Click the "OK" button to save the configuration and close the window.

The Time Trigger block has now the trigger configuration displayed as main setting in the Block's header. It displays the next date and time of sending a trigger.

12.3.4 Periodic Trigger
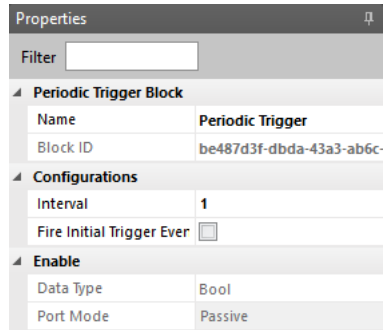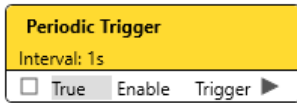
Use this Block to trigger actions periodically, e.g. to execute a script at a regularly occurring interval.

The Block has one EventOut for the trigger and an Enable PortIn. The trigger configuration is displayed as main setting in the Block's header.

— Define the interval of the triggering.

Category	Description
Enable	A passively received bool value to dis- or enable the Block.
Interval	Enter the trigger interval in seconds.
Fire Initial Trigger Event	Enable to fire an initial start trigger without waiting for the first elapsed time period.

12.3.5   Astronomic Trigger
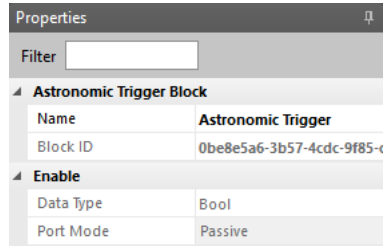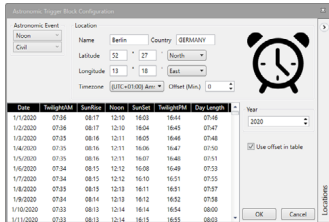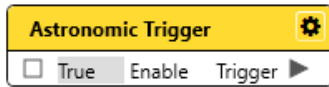
Use this Block to react on astronomic events, like Dawn, Dusk, Sunrise and Sunset.

The Block has one EventOut for the trigger and an Enable PortIn. Extended configuration is accessible at the cogwheel icon. The trigger configuration is displayed as main setting in the Block's header.

— Define the astronomic event and the reference location in the extended configuration window.

Category	Description
Enable	A passively received bool value to dis- or enable the Block.

Defining the astronomic event and the reference location in the extended configuration window

Define the astronomic event and the reference location in the extended configuration window:

1. Click the cogwheel icon at the right top corner of the Block to open the extended configuration window: ⚙ .

Astronomic event

Twilight definition

Table of all astronomic events each day of a year for the chosen twilight definition and location

Location search

List of provided locations

Buttons to add, delete, and rename locations

2. Select an astronomic event at the top drop-down menu on the left side. Available events are: TwilightAM, TwilightPM, Sunrise, Sunset, and Noon.

3. Select a twilight definition at the bottom drop-down menu on the left side to define the referenced time for all astronomic event.
   Available twilight definitions are: Civil, Nautical, and Astronomic.
   Civil: Civil twilight is defined when the sun is 6 degrees below the horizon. Illumination is sufficient, under good weather conditions, for terrestrial objects to be clearly distinguished.
   Nautical: Nautical twilight is defined when the sun is 12 degrees below the horizon. General outlines of ground objects may be distinguishable, but detailed outdoor operations are not possible, and the horizon is indistinct.
   Astronomic: Astronomical twilight is defined when the sun is 18 degrees below the horizon. For a considerable interval after the beginning of morning twilight and before the end of evening twilight, sky illumination is so faint that it is practically imperceptible.

4. Change the reference location for the event either by clicking a provided location from the list on the right side.

5. Or manually enter a location in the middle section. Click the ➕ button to add the location to the provided list. You can delete the entry (✖) or rename it (**A**). If you have offset, choose whether to use the offset in the table by checking the box "Use offset in table".

6. Click the "OK" button to save the configuration and close the window.

The Astronomic Trigger block has now the trigger configuration displayed as main setting in the Block's header. It displays the event's next date and time of sending a trigger.

— Search for provided locations using the search entry field at the top of the locations list.

Considered Settings

The computation of the astronomic trigger depends on three settings:

1. The time zone specified in the Web-UI of SYMPHOLIGHT (local time of SYMPHOLIGHT)
2. The location specified in the dialog (with latitude and longitude)
3. The time zone specified in the dialog

The meaning of these settings regarding the computation of the astronomic trigger is the following:

1. Any version of SYMPHOLIGHT, no matter if Core S or Windows application, allows to set a time zone via the Web UI. This time zone is used for the calculation of the local time inside of SYMPHOLIGHT. The default time zone setting for the Core S is UTC (Universal Time Coordinated). The default time zone for SYMPHOLIGHT on Windows is the time zone configured in the underlying Windows OS.
2. The location where the times of astronomic events will be calculated for. This is in fact a UTC time.
3. The time zone setting inside the dialog defines how the UTC time calculated based on the location is visualized. This means, the UTC time of an astronomic event is converted to the time equivalent in the specified time zone.

— When does SYMPHOLIGHT raise the astronomic trigger?

SYMPHOLIGHT evaluates the time you can see in the table generated when configuring the astronomic trigger and compares this value with the local time of SYMPHOLIGHT. The local time of SYMPHOLIGHT depends on the time zone setting made in the Web UI or the default time zone value as described in 1. . If this local time equals the time in the dialog, the trigger will be fired.

For productive use, the local time zone of SYMPHOLIGHT should match the time zone specified in the dialog.

— Examples

Local astronomic event

Imagine having an installation in Berlin demonstrating the day-night cycle of Berlin.

We want our SYMPHOLIGHT located in Berlin to fire when it is sunset in Berlin. We are having a SYMPHOLIGHT configured with the time zone of "(UTC+1) Berlin, Amsterdam, Wien" via Web-UI. We have specified the location Berlin inside of the dialog with the correct coordinates. In the dialog we specified the time zone to be "(UTC+1) Berlin, Amsterdam, Wien". In this scenario we will have the trigger firing when we see the sunset out of our Berlin office windows.

Distant astronomic event

Imagine having an installation in Berlin demonstrating the day-night cycle of Auckland. We want our SYMPHOLIGHT located in Berlin to fire when it is sunset in Auckland, New-Zealand. We are having a SYMPHOLIGHT configured with the time zone of "(UTC+1) Berlin, Amsterdam, Wien" via Web-UI. We have specified the location Auckland inside of the dialog with the correct coordinates. In the dialog we specified the time zone to be "(UTC+1) Berlin, Amsterdam, Wien". In this scenario we will have the trigger firing when the sun sets in Auckland. This could be at the early morning in Berlin, but in Auckland it is sunset time.

Testing a local astronomic event at a different place on the world

Imagine having an installation in Berlin demonstrating the day-night cycle of Berlin but designing the project in Auckland, New Zealand. We want to test it.
We want our SYMPHOLIGHT located in Berlin to fire when it is sunset in Berlin. We assume our SYMPHOLIGHT has the time zone "(UTC+12) Auckland" configured in the Web-UI. The test approach is setting the time zone of SYMPHOLIGHT to "(UTC+1) Berlin, Amsterdam, Wien" via Web-UI. The trigger will now be firing when the sun sets in Berlin. This might be some time in the night in Auckland time.

To avoid waiting for the trigger to fire while testing, you can modify the offset. The offset will be directly applied to the times of astronomic events in the time table.

Attention: Changing the offset will modify the project. After testing the offset should be reverted.

12.3.6   Data Changed Trigger

Use this Block to react on any changes that are related to a dedicated port, e.g. if a Sequence Opacity changes. Creates an Event, if a value of any data type including string changed.

The Block has one PortIn for the incoming value and one EventOut for the trigger.

Category	Description
Data	By default, the value is passively caught from the counterpart Block.

12.3.7   Push Button

Use this Block to use button press and release events to trigger elements in the Workflow Designer, e.g. if you want to start a Sequence from an Action Pad Button.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has five PortIns to control the Button and its visibility, two PortOuts for the current Button state and its state of visibility, and two EventOuts of the Button press state.
The Block can be bound to a Button from the Action Pad.

— Define the Target Property.
— Define the configuration of the PortIns.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to a Push Button from the Action Pad.
Press		Configure the Press state (pressed / not pressed) of the Push Button. By default, the bool value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value. The Value property sets the value of the PortIn to TRUE (pressed) if checked. An unchecked Value sets the PortIn to FALSE (not pressed).
Text		Define the Text on the Push Button when unpressed. By default, the string value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value.
Is Pressed Text		Define the Text on the Push Button when pressed. By default, the string value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value.
Tool Tip		Define the Tool Tip for the Push Button when the cursor hovers over the Push Button. By default, the string value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value.

Visible		Configure the visibility of the Push Button. By default, the bool value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value. The Value property sets the value of the PortIn to TRUE (visible) if checked. An unchecked Value sets the PortIn to FALSE (invisible).
Is Pressed		An actively sent bool value whether the Push Button is pressed in the Action Pad or not.
Is Visible		An actively sent bool value whether the Push Button is visible (displayed) in the Action Pad or not.

— The properties "Text", "Is Pressed Text", "Tool Tip", and "Visible" relate to the same properties of the target value that are configured in the ActionPad Designer. When differing, the Properties configuration of the ActionPad is temporarily overruled by the Workflow.

12.3.8   Toggle Button

Use this Block to use toggle on and off events to trigger elements in the Workflow Designer, e.g. if you want to start and stop a Sequence from Toggle Button.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has five PortIns to control the Toggle Button and its visibility, two PortOuts for the current Toggle Button state and its state of visibility, and two EventOuts of the Toggle state.
The Block can be bound to a Toggle Button from the Action Pad.

— Define the Target Property.
— Define the configuration of the PortIns.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to a Toggle Button from the Action Pad.
Press		Configure the Toggle state (pressed / not pressed) of the Toggle Button. By default, the bool value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value. The Value property sets the value of the PortIn to TRUE (pressed) if checked. An unchecked Value sets the PortIn to FALSE (not pressed).
Text		Define the Text on the Toggle Button when unpressed. By default, the string value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value.
Is Pressed Text		Define the Text on the Toggle Button when pressed. By default, the string value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value.
Tool Tip		Define the Tool Tip for the Toggle Button when the cursor hovers over the Toggle Button. By default, the string value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value.

Visible		Configure the visibility of the Toggle Button. By default, the bool value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value. The Value property sets the value of the PortIn to TRUE (visible) if checked. An unchecked Value sets the PortIn to FALSE (invisible).
Is Pressed		An actively sent bool value whether the Toggle Button is pressed in the Action Pad or not.
Is Visible		An actively sent bool value whether the Toggle Button is visible (displayed) in the Action Pad or not.

— The properties "Text", "Is Pressed Text", "Tool Tip", and "Visible" relate to the same properties of the target value that are configured in the ActionPad Designer. When differing, the Properties configuration of the ActionPad is temporarily overruled by the Workflow.

12.3.9   Image
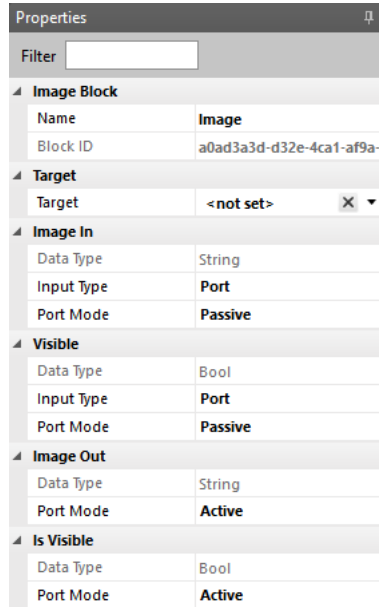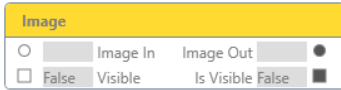
Use this block to set the image item.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has two PortIns to control the Image and the visibility and two PortOuts for the current Image and the state of visibility.
The Block can be bound to an Image from the Action Pad.

— Define the Target Property.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to an Image from the Action Pad.

Image In		Define the Image by stating the name of an Image as it is called in the Media Library (Content Tab). By default, the string value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value.
Visible		Configure the visibility of the Image. By default, the bool value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value. The Value property sets the value of the PortIn to TRUE (visible) if checked. An unchecked Value sets the PortIn to FALSE (invisible).
Image Out		An actively sent string value of the Image.
Is Visible		An actively sent bool value whether the Image is visible (displayed) in the Action Pad or not.

— The property "Visible" relates to the same property of the target value that is configured in the ActionPad Designer. When differing, the Property configuration of the ActionPad is temporarily overruled by the Workflow.

12.3.10   Text Label

Use this Block if you want to change the text of Action Pad labels from the Workflow Designer, e.g. if you want to display different messages on the Action Pad depending on the daytime.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has two PortIns to control the Text and the visibility and two PortOuts for the current Text and the state of visibility.
The Block can be bound to a Label from the Action Pad.

— Define the Target Property.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to a Label from the Action Pad.

Text In		Configure the incoming Text of the Label. By default, the string value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value.
Visible		Configure the visibility of the Label. By default, the bool value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value. The Value property sets the value of the PortIn to TRUE (visible) if checked. An unchecked Value sets the PortIn to FALSE (invisible).
Text Out		An actively sent string value of the Text from the Label.
Is Visible		An actively sent bool value whether the Label is visible (displayed) in the Action Pad or not.

— The properties "Text" and "Visible" relate to the same properties of the target value that are configured in the ActionPad Designer. When differing, the Properties configuration of the ActionPad is temporarily overruled by the Workflow.

12.3.11 Fader
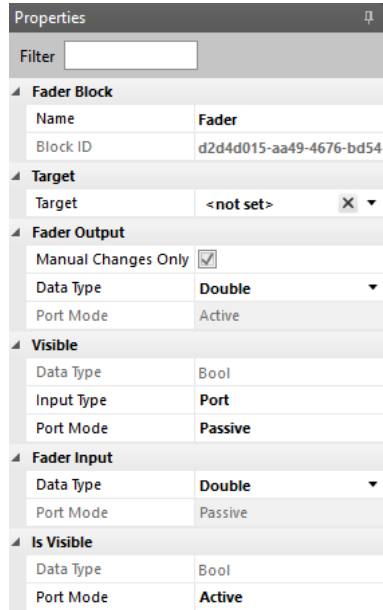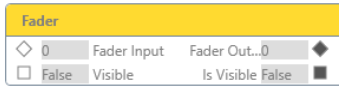
Use this Block if you want to set or receive values of Action Pad Faders in the Workflow Designer, e.g. if you want to control the Opacity of a Sequence.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has two PortIns to control the Fader value and its visibility and two PortOuts for the current Fader value and state of visibility.
The Block can be bound to a Fader from the Action Pad.

— Define the Target Property.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to a Fader from the Action Pad.

Fader Output		The configuration of the Fader value that is forwarded from the Block.
	Manual Changes Only	Enabled. The Block changes the Fader Output only on fader modification in the Action Pad and not when the Fader Input of the Block changes.
	Data Type	Define the data type of the PortOut. The Fader value can either be forwarded as Int64 or as Double.
Visible		Configure the visibility of the Fader. By default, the bool value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value. The Value property sets the value of the PortIn to TRUE (visible) if checked. An unchecked Value sets the PortIn to FALSE (invisible).
Fader Input		Define the data type of the PortIn. The Fader value can either be forwarded as Int64 or as Double.
Is Visible		An actively sent bool value whether the Fader is visible (displayed) in the Action Pad or not.

— The property "Visible" relates to the same property of the target value that is configured in the ActionPad Designer. When differing, the Properties configuration of the ActionPad is temporarily overruled by the Workflow.

— Use the "Manual Changes Only" option when multiple faders control

12.3.12   Inline Frame

Use this block if you want to change the URL for the Frame content.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has two PortIns to control the URL and the visibility and two PortOuts for the current URL and the state of visibility.
The Block can be bound to an Inline Frame from the Action Pad.

— Define the Target Property.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to an Inline Frame from the Action Pad.
URL In		Define the URL by stating the full format, e.g. "http://www.ecue.com". By default, the string value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value.
Visible		Configure the visibility of the Inline Frame. By default, the bool value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value. The Value property sets the value of the PortIn to TRUE (visible) if checked. An unchecked Value sets the PortIn to FALSE (invisible).
URL Out		An actively sent string value of the URL.
Is Visible		An actively sent bool value whether the Inline Frame is visible (displayed) in the Action Pad or not.

— The properties "URL In" and "Visible" relate to the same properties of the target value that are configured in the ActionPad Designer. When differing, the Properties configuration of the ActionPad is temporarily overruled by the Workflow.

12.3.13   Color Box / Color Wheel

Use the Color Box block or the Color Wheel block to set color.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has six PortIns to control the color and the visibility and six PortOuts for the current color and the state of visibility.
The Color Box block can be bound to a Color Box from the Action Pad.
The Color Wheel block can be bound to a Color Wheel from the Action Pad.

— Define the Target Property.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to a Color Box or a Color Wheel from the Action Pad.
Visible		Configure the visibility of the Color Box / Wheel. By default, the bool value is passively received from the counterpart Block. Set the Input Type to Property to manually define a static value. The Value property sets the value of the PortIn to TRUE (visible) if checked. An unchecked Value sets the PortIn to FALSE (invisible).
Color In		A passively received ProPhotoColor value for the color. You can change the Port Mode.
Red In Green In Blue In Alpha In		A passively received Int64 value (between 0 and 255) for the color channel. You can change the Port Mode.
Color Out		An actively sent ProPhotoColor value of the color. You can change the Port Mode.

Red Out Green Out Blue Out Alpha Out		An actively sent Int64 value of the color channel. You can change the Port Mode.
Is Visible		An actively sent bool value whether the Color Box / Wheel is visible (displayed) in the Action Pad or not.

— The property "Visible" relates to the same property of the target value that is configured in the ActionPad Designer. When differing, the Properties configuration of the ActionPad is temporarily overruled by the Workflow.

12.3.14   Page Navigation

Use this Block if you would like to switch to a certain Action Pad Page on all devices that are actually showing the corresponding Action Pad, e.g. if you want to show a specific page in case an emergency comes up.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has one PortIn and four EventIns to navigate around the Action Pad Pages.
The Block can be bound to a Target Device (Desktop, iPad, Android) from the Action Pad.

— Define the Target Property.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to a Target Device from the Action Pad.
Page Identifier		Define the target Page to which you want to switch. By default, the string value is set to the Input Type "Property" to manually define a static value. Set the Input Type to Port to passively receive a value from the counterpart Block.

— The First Page EventIn switches to the Page tab with the index 0.
— The Previous Page EventIn switches to the Page tab with the preceding index.
— The Next Page EventIn switches to the Page tab with the following index.
— The Last Page EventIn switches to the Page tab with the highest index.
— Connect a Button block to the Page Navigation block to switch Action Pad Pages upon a Button press. You can also connect different block types to the Page Navigation block.
— All instances of one Target Device's Action Pad that are run in browsers switch to the same triggered target Page, independently of their current Page.
— In comparison to the manual selection of Pages using the Action Pad, Workflow actions concern every open Action Pad on all devices that are actually showing the corresponding Action Pad.

12.3.15   Value
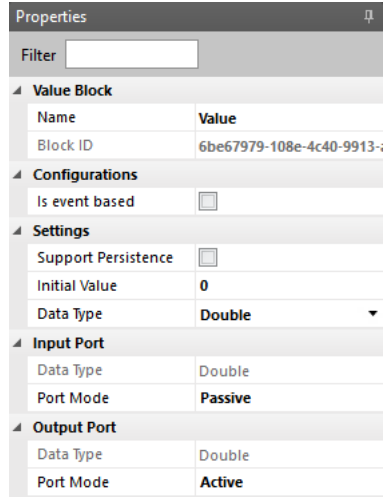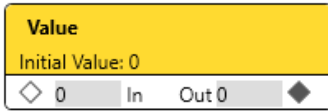
Use this Block to add a constant value to your workflow, e.g. you want to trigger a scene of a DALI Group. The Value block can be configured to be triggered manually.

The Block has one PortIn and one PortOut for the value. The Initial Value is displayed as main setting in the Block's header.

— Define the Data Type.
— Define the Initial Value.

Category	Property	Description
Is event based		Enables this block to be executed on event trigger.
Settings		Configure the constant value and whether to cache it or not.
	Support Persistence	Enable to store the value locally to the persistent memory. Note that the memory is not saved inside SYMPHOLIGHT. A project transferred to another device does not contain the persistent memory data.

	Initial Value	The initial value of the PortOut. For data type Bool, check the initial value to be TRUE. If "Store Every Value Change" is enabled, the initial value is the last stored value on loading the project.
	Data Type	The data type for the PortIn and PortOut.
Store Every Value Change		Only for "Support Persistence": Enable to save the last value change to the persistent memory.
Input Port		By default, the value is passively caught from the counterpart Block.
Output Port		The value, by default sent actively.

— A possible replacement for a Property PortIn: Connect a Value block to the Port PortIn.
— You can use this Block to cache data and to continue a project from previous data state. Enable "Support Persistence". The Block gets four additional EventIns:
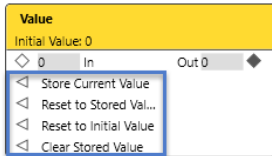
Store Current Value: When triggered, saves the current value of the Block to the persistent memory.
Reset to Stored Value: When triggered, sets the PortOut to the last stored value of the persistent memory.
Reset to Initial Value: When triggered, sets the PortOut to the Initial Value.
Clear Stored Value: When triggered, clears the persistent memory of its value data.

— Is the Out PortOut already connected to a counterpart Block and you then set an initial value, the value will be forwarded immediately.
— Use the Converter block if you want e.g. to do a DateTime to double conversion.

> Every Link that is connected to the Block will be dismissed on changing the data type.

12.3.16   State To Event
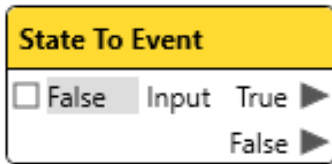
Use this Block to react on bool state changes, e.g. if a value changes from TRUE to FALSE. Creates an Event, if a bool value changed.

The Block has one PortIn for the incoming value and two EventOuts for each value state.

Category	Description
Fire Only On Change	Enable to fire events only on bool state changes, e.g. if a value changes from TRUE to FALSE. Disabled, events are fired every time a bool value was received.
Input	By default, the value is passively caught from the counterpart Block.

12.3.17 Event to Value
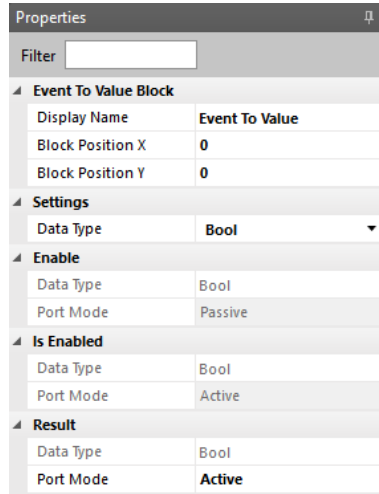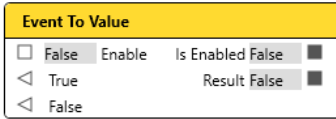
Use this Block if you want to output statically defined values on an event.

The Block has initially two EventIns for the incoming events, an Enable PortIn, one PortOut for the value and an Is Enabled PortOut.

— Define the data types of the incoming Ports.

Category	Property	Description
Settings		Define the EventIns for the incoming events.
	Data Type	Define the source data type of the incoming values. Choose between bool, double, and string.
	Number of Inputs	Define the amount of EventIns to create (max. 32).
Output Values		Define each value that is forwarded from the PortOut when the corresponding EventIn is triggered.

Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.
Result		The latest incoming EventIn as a string value, by default sent actively.

12.3.18   Combine Text
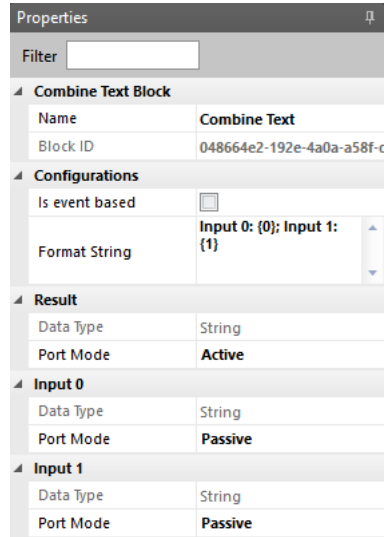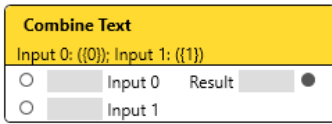
Use this block to combine two or more texts.

The Block has initially two PortIns for the incoming text parts and one PortOut for the combined text. The Format String is displayed as main setting in the Block's header.

— Define how the combined text is constituted. The text is built with a format string (Format String), where {0} and {1}, etc. are the input values.

Category	Description
Is event based	Enables this block to be executed on event trigger.
Format String	The format string to be created. It can contain free text as well as numbers in curly brackets (e.g. {0}). Each number in curly brackets defines a PortIn for a text part. The same number can occur multiple times. Example: "Here are the values of port0: {0} and port1: {1} and port0 again: {0}."
Result	The combined text, by default sent actively.

Text part (e.g. Input 0)	Each text part has its own PortIn. The number of Ports is defined in the format string. You can change the Port mode for each Input PortIn.

— You can use string in the Format String e.g. to declare units.
  Example: "{0} Watt".
— Use the keys Shift + Return in the Format String to insert a line break.

12.3.19   Value to Event / State
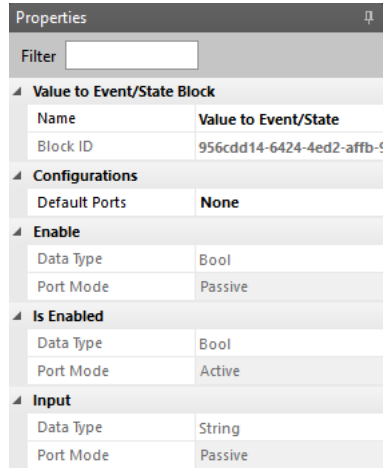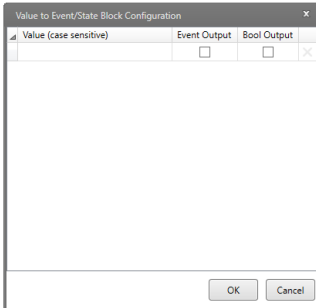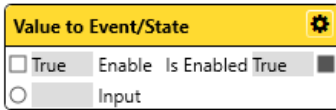
Use this Block to catch defined incoming Strings and to send a notification about that event.

The Block has one PortIn for the incoming string, an Enable PortIn, and an Is Enabled PortOut. Extended configuration is accessible at the cogwheel icon.

— Define string values in the extended configuration window and how the event of a matching string value is forwarded. A trigger EventOut and a bool PortOut are available for each string value.
— Configure the handling of non-matching strings.

Category	Property	Description
Default Ports		Define how to handle non-matching strings.
	Value	Configure which type of default PortOut(s) shall be available. The default PortOut(s) is activated whenever none of the input values is matching the conditions. **None:** No action for non-matching strings. **Bool Out:** Creates the bool PortOut "Default" for all non-matching strings. FALSE: the incoming string matches one condition. TRUE: the incoming string does not match any condition. **Event Out:** Creates the EventOut "Default" for all non-matching strings. Whenever an incoming string does not match any condition, an trigger event is fired. **Event & Bool Out:** Creates the EventOut "Default" and the bool PortOut "Default".
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not.
Input		The PortIn that passively receives the string value. The Port mode of the counterpart Block should be set to "active".

Defining string values in the extended configuration window

Define string values in the extended configuration window and how the event of a matching string value is forwarded:

1. Click the cogwheel icon at the right top corner of the Block to open the extended configuration window: ⚙ .
2. Enter the string value you want to catch in the column "Match Expression".

3. Either check the Event Output to create an EventOut, or check the Bool Output to create a bool PortOut, or check both options. At least one box needs to be checked. The EventOut will send a trigger event whenever the incoming string matches the entered string value in "Match Expression".
   The bool PortOut will send a bool data: FALSE when the incoming string does not match the entered string value; TRUE when the incoming string matches the entered string value.
4. Click the "OK" button to save the configuration and close the window.

The Value to Event / State block has now additional PortOuts according to the extended configuration and named with the string values.

— To add another string value condition, double-click into the column "Match Expression" and press Enter. Alternatively, press the Tab key five times.
— Multi-select of rows is possible. You can also select all rows by clicking the triangle icon at the left top corner of the extended configuration window.
— To delete a string value condition, click the cross button of the row on the right.

Example: The three string values "ExampleText" with a bool PortOut, "AnotherExample" with both EventOut and bool PortOut, and "Sequence#1Play" with an EventOut are defined. Non-matching string values are not reported (Default Ports Value: None):
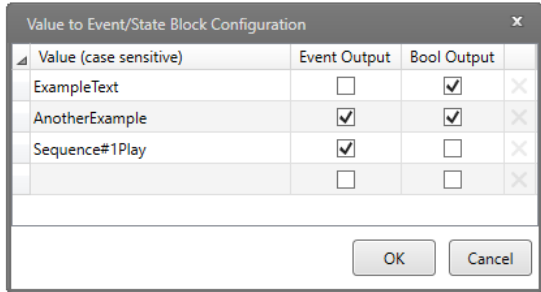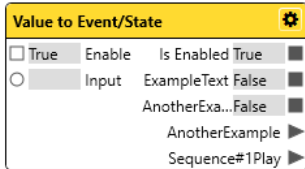
12.3.20 Converter
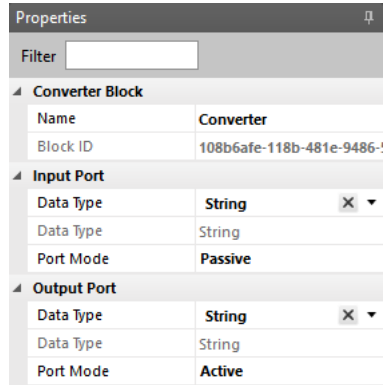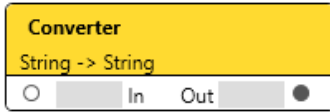
Use this Block to convert data types.
Since v3.0 this block is obsolete as the Workflow Designer Blocks are converting data types automatically.

The Block has one PortIn for the incoming value and one PortOut for the converted value. The Data Type Conversion defines the title of the Block.

— Define the data types of the incoming and of the outgoing Ports.

Category	Property	Description
Input Port		An initially passively received value to be converted into another data type.
	Data Type	Define the source data type of the incoming value.
	Port Mode	Define the Port mode of the Input Port.

Output Port		An initially actively sent value of the conversion's result.
	Data Type	Define the target data type of the outgoing value.
	Port Mode	Define the Port mode of the Output Port.

— The availability of data types depends on the counterpart Port. E.g. after choosing a data type for the PortIn, the PortOut offers a changing set of data types (same goes when first setting the PortOut data type.). To retrieve the full list of (all) data types, you have to select "String" as data type on either Port. Then the other Port provides all data types again.

12.3.21 Permit Event
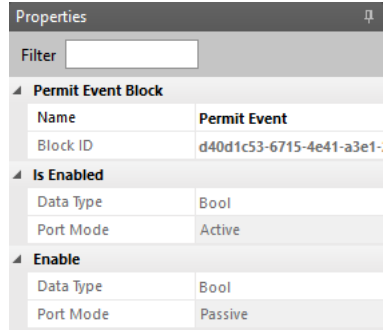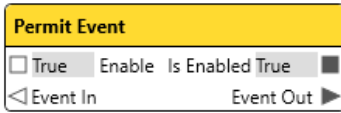
Use this Block to decide whether to pass an incoming event or to block it.

The Block has one EventIn for the incoming event, an Enable PortIn, one EventOut for the permitted event and an Is Enabled PortOut.

Category	Description
Is Enabled	An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.
Enable	A passively received bool value to dis- or enable the Block.

12.3.22 Math Formula

Use this Block for advanced mathematical functionalities, e.g. combine arithmetic calculations.

The Block has PortIns for the variables of the formula and one PortOut for the result of the calculation. The equation is displayed as main setting in the Block's header.

— Define a mathematical formula.

Category	Description
Is event based	Enables this block to be executed on event trigger.
Formula	Enter the mathematical function as a string. Any formula is possible (all math methods such as "pow(x,y)": the number x raised to the power y) and any variable names (any string separated by an operator is a new variable / Port).

Variable (e.g. x0)	By default, the content is passively caught from the counterpart Block. Set the Input Type to Property to manually define a static value.
Result	The actively sent result from the calculation. You can change the Port Mode.

— Available mathematical functions to use in the Formula are:

abs

cos

cosh

arccos

sin

sinh

arcsin

tan

tanh

artan

sqrt

rem

root

pow

exp

log

round

truncate

floor

sign

12.3.23 Addition

Use this Block to add two or more values.

The Block has initially two PortIns for the incoming values, an Enable PortIn, and one PortOut for the result. The equation is displayed as main setting in the Block's header.

— Define the numbers of terms (Input Count).

Category	Description
Is event based	Enables this block to be executed on event trigger.
Input Count	Count of available PortIns / terms (min 2 values, max 20 values).
PortIn (e.g. Input 2)	By default, the value is passively caught from the counterpart Block. Set the Input Type to Property to manually define a static value.

Enable	A passively received bool value to dis- or enable the Block.
Input 1	A received value. You can set the Port Mode.
Result	The result of the addition, by default sent actively.

12.3.24 Subtraction

Use this Block to subtract two or more values.

The Block has initially two PortIns for the incoming values, an Enable PortIn, and one PortOut for the result. The equation is displayed as main setting in the Block's header.

— Define the numbers of terms (Input Count).

Category	Description
Is event based	Enables this block to be executed on event trigger.
Input Count	Count of available PortIns / terms (min 2 values, max 20 values).
PortIn (e.g. Input 2)	By default, the value is passively caught from the counterpart Block. Set the Input Type to Property to manually define a static value.

Enable	A passively received bool value to dis- or enable the Block.
Input 1	A received value. You can set the Port Mode.
Result	The result of the subtraction, by default sent actively.

12.3.25 Multiplication

Use this Block to multiply two or more values.

The Block has initially two PortIns for the incoming values, an Enable PortIn, and one PortOut for the result. The equation is displayed as main setting in the Block's header.

— Define the numbers of factors (Input Count).

Category	Description
Is event based	Enables this block to be executed on event trigger.
Input Count	Count of available PortIns / factors (min 2 values, max 20 values).
PortIn (e.g. Input 2)	By default, the value is passively caught from the counterpart Block. Set the Input Type to Property to manually define a static value.

Enable	A passively received bool value to dis- or enable the Block.
Input 1	A received value. You can set the Port Mode.
Result	The result of the multiplication, by default sent actively.

12.3.26   Division

Use this Block to divide two values.

The Block has two PortIns for the incoming values, an Enable PortIn, and one PortOut for the result. The equation is displayed as main setting in the Block's header.

— Define the numbers of terms (Input Count).

Category	Description
Is event based	Enables this block to be executed on event trigger.
Input 2	The divisor value. You can set the Port Mode. By default it is a Property PortIn. The value 0 is invalid.
Enable	A passively received bool value to dis- or enable the Block.
Input 1	The dividend value. You can set the Port Mode.
Result	The result of the division, by default sent actively.

12.3.27   Moving Average
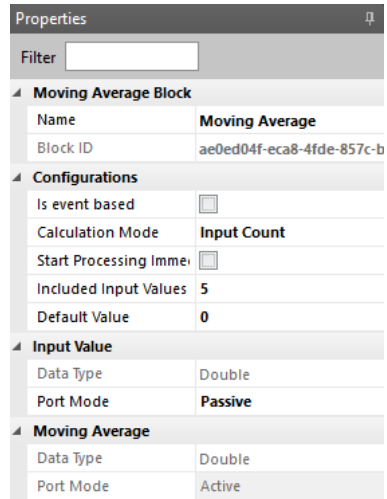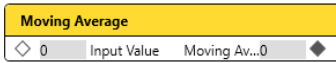
Use this Block if you want to create a moving average value based on a number of received values or based on values that have been received within a certain time.

The Block has one PortIn for the received values and one PortOut for the result of the moving average.

— Define the Calculation Mode.
— Define the usage of values.

Category	Description
Is event based	Enables this block to be executed on event trigger.
Calculation Mode	The work mode of the block. **Time:** Calculates the moving average based on the PortIn values received within the specified time. **Input Count:** Calculates the moving average based on the specified number of lastly received values.
Start Processing Immediately	When enabled, calculates the moving average also when less values than specified have been received or less time than specified has past.

Included Input Values	For Input Count as Calculation Mode only: Specifies the count of lastly received values which is used to calculate the moving average. Default: 5 values.
Covered Timespan	For Time as Calculation Mode only: Specifies the elapsed timespan for which the received PortIn values will be used to calculate the moving average (in seconds).
Default Value	The value which is used in the Moving Average PortOut while not enough PortIn values have been received.
Input Value	The passively received values to calculate the moving average from.
Moving Average	The actively sent moving average from the calculation.

Please note that while assembling your Workflow, the Moving Average resets with every newly added connection to the "Input Value" PortIn. This does not lead to any loss of functionality when running a project. Recommendation: Connect blocks (e.g. Value Blocks) to the PortIns first and then define or change the values of the connected blocks (e.g. Value Blocks).

12.3.28   Time Condition
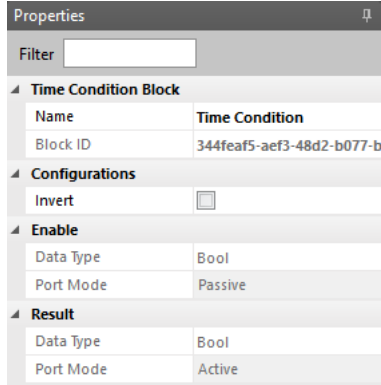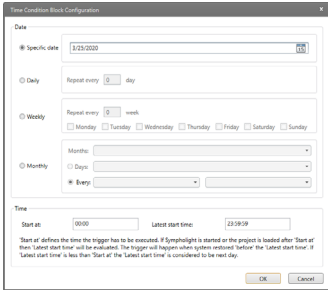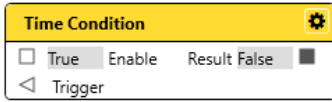
Use this Block to react on time events, but providing a bool value instead of a trigger. Use this Block if you want to limit the execution of other triggers to a specific time range, e.g. only allow to switch on the lights if it is after 7am.

The Block has one EventIn for the incoming trigger, an Enable PortIn, and a bool PortOut. Extended configuration is accessible at the cogwheel icon. The condition configuration is displayed as main setting in the Block's header.

— Define date and time of the time condition in the extended configuration window.
— Connect a trigger Link to the EventIn.

Category	Description
Invert	Inverts the bool value of the PortOut. Use this setting to get a FALSE PortOut value whenever the time condition is met, e.g. to disable a button for a certain time span.
Enable	A passively received bool value to dis- or enable the Block.
Result	An actively sent bool value whether the time condition is met or not.

Defining date and time in the extended configuration window

Define date and time in the extended configuration window:

1. Click the cogwheel icon at the right top corner of the Block to open the extended configuration window: ⚙ .
2. Select one of the four date options at the Date frame:

   Specific date: Use the calender at the icon or enter manually a particular day.
   Daily: Enter a number for the daily interval, e.g. enter "2" for triggering every other day. Compared to the "0" with immediate effect, when entering a "1", the day count starts with the day after configuring the block.
   Weekly: Enter a number for the weekly interval, e.g. enter "1" for triggering every week. Compared to the "0" with immediate effect, when entering a "1", the week count starts with the Monday of the next week at the time of configuring the block. Further select the weekdays.
   Monthly: Select one or multiple months. Also enter specific days or weekdays. The configuration is of immediate effect.

3. Define a time span at the Time frame. During the time span the condition is met (TRUE PortOut value).
4. Click the "OK" button to save the configuration and close the window.

The Time Condition block has now the condition configuration displayed as main setting in the Block's header. It displays the next date and time of being TRUE.

12.3.29   Compare Condition
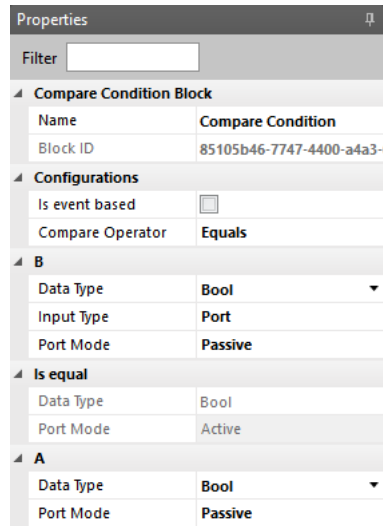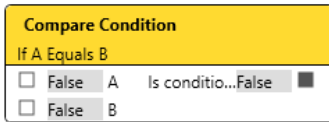
Use this Block to compare two values with each other.

The Block has two PortIns for the values that are to be compared, and a bool PortOut. The condition configuration is displayed as main setting in the Block's header.

— Define how to compare two values with the Compare Operator.
— Define the two PortIns for the values you want to compare.

Category	Property	Description
Is event based		Enables this block to be executed on event trigger.
Compare Operator		The operator the block needs to compare the two values.
	Value	**Equals:** The two values match. **SmallerThan:** When A is smaller than B. **GreaterThan:** When A is greater than B. **SmallerThanOrEquals:** When A is smaller than or matches B. **GreaterThanOrEquals:** When A is greater than or matches B.
B		B is the value to be compared to. Adjust the data type, input type and port mode for the B PortIn according to your needs. You can set the B PortIn as a Property with a fixed value.
Is condition met		An actively sent bool value whether the compare condition is met or not.
A		Adjust the data type and the Port mode for the A PortIn according to your needs.

12.3.30   State Condition

Use this Block if you would like to switch between two different scenarios depending on one single value, e.g. change the text of a button if another button has been pressed. Depending on the Condition PortIn either the value of the Case True PortIn or of the Case False PortIn is routed to the Result PortOut.

The Block has three PortIns for the state and the cases that are to be caught, and a dynamic PortOut. The condition configuration is displayed as main setting in the Block's header.

— Define the two Cases (Case True and Case False)

Category	Description
Case True	The value which is forwarded at the PortOut if the value of the bool Condition PortIn is TRUE. As a dynamic data type Port, any PortOut is connectable. By default, the content is actively caught from the counterpart Block. Set the Input Type to Property to manually define a static value. The Value property sets the value of the PortIn to TRUE if checked. An unchecked Value sets the PortIn to FALSE.
Case False	The value which is forwarded at the PortOut if the value of the bool Condition PortIn is FALSE. As a dynamic data type Port, any PortOut is connectable. By default, the content is actively caught from the counterpart Block. Set the Input Type to Property to manually define a static value. The Value property sets the value of the PortIn to TRUE if checked. An unchecked Value sets the PortIn to FALSE.
Result	The value of the Case which matches the Condition as a dynamic PortOut. You can change the Port Mode.
Condition	The bool value that decides which Case's value is forwarded in the Result PortOut. You can change the Port Mode.

12.3.31   AND
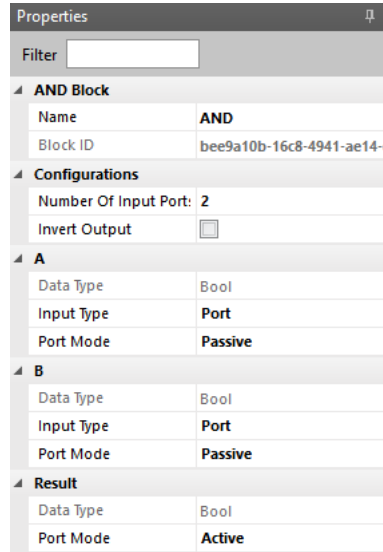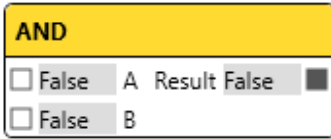
Use this Block if you want to combine two or more conditions: only if all inputs are TRUE, the output will be TRUE. A logical AND function with two or more inputs.

The Block has initially two PortIns for the incoming values and one PortOut for the result.

— Define the numbers of PortIns (Number of Input Ports).

Category	Description
Number of Input Ports	Count of available PortIns (min 2 values).
Invert Output	Inverts the bool value of the PortOut. Use this setting to get a FALSE PortOut value whenever the AND condition is met. Enable to use the block as AND gate (output which is FALSE only if all its inputs are TRUE).

PortIn (e.g. A)	By default, the value is passively caught from the counterpart Block. Set the Input Type to Property to manually define a static value.
Result	The result of the combination, by default sent actively.

12.3.32 OR

Use this Block if you want to combine two or more conditions: only if one or more inputs are TRUE, the output will be TRUE. A logical OR function with two or more inputs.

The Block has initially two PortIns for the incoming values and one PortOut for the result.

— Define the numbers of PortIns (Number of Input Ports).

Category	Description
Number of Input Ports	Count of available PortIns (min 2 values).
Invert Output	Inverts the bool value of the PortOut. Use this setting to get a FALSE PortOut value whenever the OR condition is met. Enable to use the block as NOR gate (output which is TRUE only if all its inputs are FALSE).

PortIn (e.g. A)	By default, the value is passively caught from the counterpart Block. Set the Input Type to Property to manually define a static value.
Result	The result of the combination, by default sent actively.

12.3.33 NOT

Use this Block to negate a boolean value: if the input is TRUE, the output will be FALSE and vice versa. A logical NOT function for one value.

The Block has one PortIn for the incoming value and one PortOut for the result.

Category	Description
In	By default, the value is passively caught from the counterpart Block.
Result	The result of the combination, by default sent actively.

12.3.34 Threshold

Use this Block if you want to turn off the light if the light sensor value is above a threshold level and to turn on the light if the sensor value is below, using triggers.

The Block has an Enable PortIn, two PortIns for the light sensor value and the threshold, two EventOuts to switch the light on / off, and an Is Enabled PortOut.

— Define the Target Light Level.
— Define Tolerance, Turn On and Off Delay, and Initial State where applicable.

Category	Description
Initial State	Define whether to send a trigger upon initializing the SYMPHOLIGHT project: **SetLightOn**: The Light On EventOut is fired. **SetLightOff**: The Light Off EventOut is fired. **DoNothing**: No trigger is fired.

Tolerance	Define the value range around the Target Light Level that is ignored for sending triggers.
Turn On Delay	Define the time in seconds to elapse between receiving a sensor value below the threshold Target Light Level value and firing the Light On EventOut trigger. During the delay time, incoming sensor values are ignored.
Turn Off Delay	Define the time in seconds to elapse between receiving a sensor value above the threshold Target Light Level value and firing the Light Off EventOut trigger. During the delay time, incoming sensor values are ignored.
Target Light Level	The threshold value to which the incoming sensor value is compared. Define whether the threshold value is a fixed value (Input Type "Property") or a received Input value (Input Type "Port").
Enable	A passively received bool value to dis- or enable the Block.
Is Enabled	An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.
Sensor Value	A passively received Int64 value of the light sensor.

12.3.35 Fade

Fade	
☐ True	Enable
◇ 0	Lower Target
◇ 100	Upper Target
◇ 5	Fade Time (s)
◇ 0	External value
◁ Fade Up	
◁ Fade Down	
◁ Pause	
◁ Reset	
◁ Immediate Up	
◁ Immediate Down	

Is Enabled True ■
Is fading Up False ■
Is fading Do...False ■
Value 0 ◆
Inverted Value 100 ◆
Fade Up Stopped ▶
Fade Down Stopped ▶

Properties 🗕

Filter [_____]

▲ **Fade Block**	
Name	**Fade**
Block ID	d6aa5f1e-652a-4ed0-8334-...
▲ **Behavior**	
Trigger ports	**Dedicated event ports**
Retrigger Behavior	**Continue fading**
Opposite direction tri	**Pause current fade**
▲ **Lower Target**	
Data Type	Double
Input Type	**Property**
Value	**0**
▲ **Upper Target**	
Data Type	Double
Input Type	**Property**
Value	**100**
▲ **Fade Time (s)**	
Data Type	Double
Input Type	**Property**
Value	**5**
▲ **External value**	
Data Type	Double
Input Type	**Property**
Value	**0**
▲ **Enable**	
Data Type	Bool
Port Mode	Passive
▲ **Is Enabled**	
Data Type	Bool
Port Mode	Active
▲ **Is fading Up**	
Data Type	Bool
Port Mode	**Active**
▲ **Is fading Down**	
Data Type	Bool
Port Mode	**Active**
▲ **Value**	
Data Type	Double
Port Mode	**Active**
▲ **Inverted Value**	
Data Type	Double
Port Mode	**Active**

Use this Block to fade between two values. E.g. fade the opacity of a Sequence or the luminance of a DALI group.

The Block has initially four PortIns for the opacity, six EventIns to control the play state of the Sequence, an Enable PortIn, four PortOuts for opacity and playback, two EventOuts of the play state, and an Is Enabled PortOut.

Category	Property	Description
Behavior		
	Trigger Ports	Choose a set of ports which trigger fadings.
	Retrigger Behavior	Define the behavior of the block when a fade in the same direction is triggered while the block is already fading.
	Opposite Direction Trigger	Define the behavior of the block when a fade in the opposite direction is triggered while the block is already fading in one direction.
Lower Target		Defines the minimum value where the fade down ends.
	Input Type	Define whether the target value is a fixed value (Input Type "Property") or a received Input value (Input Type "Port").
	Value	Define the target value of the lower target.
Upper Target		Defines the maximum value where the fade up ends.

	Input Type	Define whether the target value is a fixed value (Input Type "Property") or a received Input value (Input Type "Port").
	Value	Define the target value of the upper target.
Fade Time		The maximum duration of the fade from Lower Target to Upper Target.
	Input Type	Define whether the target value is a fixed value (Input Type "Property") or a received Input value (Input Type "Port").
	Value	Define the target value of the lower target.
External Value		Define an initially set or externally set value. Also used for the Reset Trigger event.
	Input Type	Define whether the target value is a fixed value (Input Type "Property") or a received Input value (Input Type "Port").
	Value	Define the target value of the lower target.
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.

Is Fading Up		A bool value about whether the block is currently fading up. You can define the Port Mode.
Is Fading Down		A bool value about whether the block is currently fading down. You can define the Port Mode.
Value		A double value of the fading value. You can define the Port Mode.
Inverted Value		A double value that inverts the fading value. You can define the Port Mode.

— External value: Helps, when a target value can be controlled from multiple sources (Fade block, ActionPad Fader, etc.). Other sources can be connected to this port of the Fade block instead of the target directly, and there should not be much conflict between fading and other sources.
— Immediate up/down events: E.g. set a fade to a Glass Touch button short push and the immediate jump to the double click or long press. For users, who are in a hurry.
— Fade up/down stopped events: To be notified on a fade stop and completion.
— Fade time can be a port. Changing the fade time during a fade is also supported. E.g. if someone wants to trigger a "fast forward".

12.3.36   Measure Compare Adjust (PID)

Use this Block if you want to control the fixture light level depending on a certain sensor target value, e.g. in Daylight Harvesting scenarios.
The Output Value is calculated in accordance to the Calculation Settings and controls the light level of fixtures. The Output Value is adjusted until the light level that is produced in assistance of the fixtures and measured by the sensor equals the Target Value.

The Block has an Enable PortIn, two PortIns for the light sensor value and the sensor target value, one PortOut of the current light value, and an Is Enabled PortOut.

- Define the light adjustment method and the forwarded light value limits at the Block Configuration.
- Define how to calculate the light adjustment at the Calculation Settings.
- Define the Target Value of the light to be achieved.

Category	Property	Description
Block Configuration		Define the light limits of the forwarded light value and the light adjustment method.
	Min. Target Level	Define the minimum value the forwarded light value (Output Value) can be.
	Max. Target Level	Define the maximum value the forwarded light value (Output Value) can be.
	Enable Expert Mode	Define the light adjustment method: Enable to change the Calculation Settings to the three PID terms K_p, K_i, and K_d using the formula $$u(t) = K_p e(t) + K_i \int_0^t e(r)dr + K_d \frac{de(t)}{dt}$$. When disabled, the simple mode is activated with K_p as Responsiveness, K_i as Step, and K_d=0.

Calculation Settings		The adjustment calculation is based on the terms of the Calculation Settings.
	Step	Define the size of the adjustment step between 1 as low impact and 100 as high impact (high risk of oversteering). K_i = Step / 100.
	Responsiveness	Define the speed of the adjustment between 1 as slowest and 100 as fastest. K_p = Responsiveness * 0.01.
	K_p	The proportional gain of the adjustment.
	K_i	The integral gain of the adjustment.
	K_d	The derivative gain of the adjustment.
Target Value		The value to which the incoming sensor value is compared. The Output Value is adjusted until the light level that is produced in assistance of the fixtures and measured by the sensor equals the Target Value.
	Input Type	Define whether the target value is a fixed value (Input Type "Property") or a received Input value (Input Type "Port"). Use Port for defined scenarios like a connected Multi Delay block or Human Centric Lighting block.
	Value	Define the target value of light. The target value is limited by the Min. Target Level and the Max. Target Level.
Enable		A passively received bool value to dis- or enable the Block.

Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.
Sensor Value		A passively received double value of the light sensor.
Output Value		A double value that is calculated to control the fixtures in order to achieve the Target Value measured by the sensor. Connect to fixtures. You can define the Port Mode. The Output value is limited by the Min. Target Level and the Max. Target Level.

— The Block is designed for the use case of sensor and fixtures being placed in one area.
— When switching between Expert Mode and Simple Mode in the Block Configuration, the Calculation Settings are reset.
— The configuration of the Measure Compare Adjust (PID) block is subject to the sensors and fixtures in use as well as the location of the devices. Test the installation in the field and for Daylight Harvesting settings under various scenarios to determine whether the sensor is eligible and to define the Calculation Settings. Use a second sensor to check the light value of the target space, e.g. table surface.
— To determine the Min. Target Level for Daylight Harvesting scenarios, test the installation on a very bright day. Control the light value until the target value of the second sensor is achieved. Measure the light value of the fixtures at that moment. Set this value as Min. Target Level.
— To determine the Max. Target Level for Daylight Harvesting scenarios, test the installation at a very dark situation, e.g. at night. Turn all fixtures off. Measure the light value of the fixtures at that moment. Set this value as Max. Target Level.

12.3.37   Human Centric Lighting

What is Human Centric Lighting - An Introduction

Human centric lighting is the art of creating lighting that mimics the natural daylight that drives our bodily functions by changing the color temperature and intensity to stimulate natural light colors. It enhances human performance, comfort, health and wellbeing. Human Centric Lighting solutions essentially contribute to levels of well-being.

The most obvious effect of light on humans, is vision. It enables us to identify brightness, shapes, colors, images, and perceive information and contrast. But light also impacts our biology; it affects our hormones, alertness, attention and fatigue, and also determines our body clock and circadian rhythm. Human centric lighting takes these effects into consideration to provide a holistic and application oriented approach to lighting for humans. It balances visual, emotional and biological needs of humans in lighting application.

Dynamic white light with brightness levels and light colours modified according to the course of day support our natural biological rhythm to increase our well-being and performance capability, as well as achieving optimum synchronisation with the outdoor environment.

To implement an adequate Human Centric Lighting solution install a time-controlled lighting system with a circadian-friendly light sequence, meaning the luminaires adapt their light colour and intensity during the course of the day.

Based upon time, date and geographical location, the Human Centric Lighting block of SYMPHOLIGHT can calculate sunrise and sunset for the given location at any time of the year. Transition values are calculated to continuously and accordingly adapt the color temperature and intensity.

The Human Centric Lighting block is user configurable, the default configuration is: At 7 AM, the Block starts adapting the light temperature setting from 2700 K to its maximum of 6500 K over the course of four hours. The light will stay at a fixed temperature level of 6500 K until the 2 PM. Then, the Block starts the inverse transition process, adapting the light temperature back to 2700 K, reaching 2700 K at 7 PM.

Use this Block to create human centric lighting features like defining a color temperature curve depending on the time of day.

The Block has an Enabled PortIn, two EventIns to control the Preview, an Is Enabled PortOut, and two PortOuts for the values of the human centric lighting features. Extended configuration is accessible at the cogwheel icon.

— Define the human centric lighting features in the extended configuration window.
— Define the Execution Interval.

Category	Properties	Description
Execution Interval		Define the interval in seconds in which the block calculates its outputs. Minimum time is 0.1 seconds.
Enable		A passively received bool value to dis- or enable the Block.

Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.
Color Temperature		The color temperature value that is calculated by the block according to the human centric lighting settings. You can change the Port Mode.
Intensity		The light intensity value that is calculated by the block according to the human centric lighting settings. You can change the Port Mode.

Defining the human centric lighting features in the extended configuration window

Define the human centric lighting features by setting the location and the desired lighting features in the extended configuration window:

1. Click the cogwheel icon at the right top corner of the Block to open the extended configuration window: ⚙ .
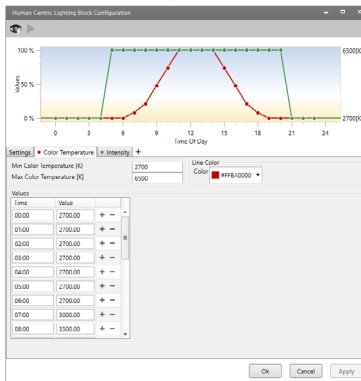
The graph represents the configuration of the tabs. The scale at the right displays the minimum and maximum values of the selected lighting feature (tab). The scale at the left displays the percent with the minimum value as 0% and the maximum value as 100%. The background color displays the color gradient between the minimum and the maximum value of the Color Temperature. Use the calender at the top of the graph to display the astronomic settings for a certain day.

2. Define the location in the Visual Settings tab to adjust the astronomic times as reference of the astronomic events on site. You can select the astronomic events that you want to be displayed in the graph and define offset times.

3. Define the human centric lighting features. Each tab represents a lighting feature. By default, the two features Color Temperature in Kelvin and Intensity in percent are predefined.
   You can define the minimum and maximum limits for the values that can be used in the table, a line color for the graph, and values for specific times. The number of time spots is also adjustable, by default one dot every hour. Each time spot with its value is represented by a colored dot in graph.
   You can change the name of the lighting feature that is also displayed as PortOut name in the block by using the context menu of the tab.
4. You can manually add (double-click in the graph) and adjust (drag and drop) dots for a selected lighting feature (tab) in the graph for rough settings.
5. Click the "OK" button to save the configuration and close the window. Click "Apply" to save the changes, but continue working in the configuration window.

The Human Centric Lighting block has now the lighting features configured. All features are implemented with Ports in the Block.

Preview the Human Centric Lighting configuration

Use the Preview for testing the human centric lighting configuration. The Preview is a fast run-through of one day, displayed on your fixtures. Define the duration of one run-through and the interval for value changes during this time in the extended configuration window, Visual Settings tab. You can run the Preview in loop and define a time frame. You can either start the Preview from inside the extended configuration window, enabling the Preview with the 👁 icon and clicking the play icon ▶ . Or you can start and control the Preview using the EventIns of the Block. When using the 👁 icon, the icon turns red to indicate the Preview mode of the block. Additionally, a time bar appears in the graph for visual feedback in the graph when the Preview is running. This time bar can be dragged to manually change the starting point of the Preview. When a Preview is running or enabled from inside the extended configuration window, you cannot change any configuration in the extended configuration window.

— The transition values between the defined time spots are linear calculated / interpolated.
— Due to the high amount of data, the light values on the fixtures might vary slightly during Preview from the live execution values.
— Hover over a dot in the graph for information on its values.
— Time values will be sorted automatically after re-selecting the tab if entered in a "wrong" chronological order.
— When changing the minimum and maximum limits of a lighting feature in the extended configuration window and values exceed these limits, these values are automatically adjusted to the new limits. E.g. you have value of 6500 K used in the Color

Temperature feature. Now you set the maximum limit for Color Temperature to 6000 K; the value is automatically changed to 6000 K.
— Each additional tab for a new lighting attribute creates a corresponding PortOut in the Block.
— Combine the Human Centric Lighting block with Daylight Threshold and sensor blocks to achieve a flawless energy balance while avoiding energy input.

12.3.38 Toggle

Use this Block to achieve a toggle functionality, e.g. first button press: light on, second button press: light off.

The Block has a Trigger EventIn, two EventOuts to switch the light on / off, and a State PortOut.

— No configuration is needed.

Category	Description
Initial State	Enable to fire the Triggered EventOut on project start.
State	An actively sent bool value that toggles on every incoming trigger, alternating between TRUE and FALSE.

— The Triggered EventOut is fired when the State is changed to TRUE.
— The Released EventOut is fired when the State is changed to FALSE.

12.3.39   Single Delay
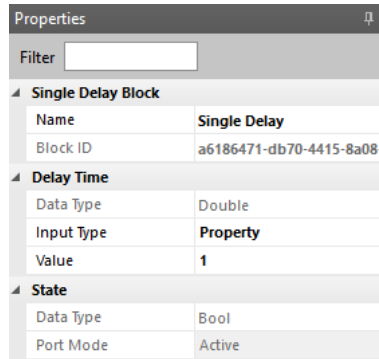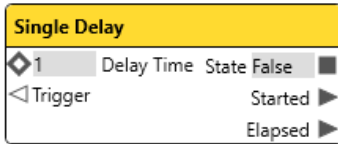
Use this Block to react on an event a certain time after it has been triggered.

The Block has a Trigger EventIn, a PortIn for the delay time, two EventOuts to switch the light on / off, and a State PortOut.

— Define the Delay Time.

Category	Description
Delay Time	Define the time after which the State value will be reset to FALSE after it has been triggered to TRUE.
State	An actively sent bool value that becomes TRUE on receiving the incoming trigger and turns back to FALSE after the defined Delay Time.

— The Triggered EventOut is fired as soon as a Trigger EventIn is received; the State is TRUE.
— The Released EventOut is fired after the defined Delay Time and when the State is changed to FALSE. If another Trigger EventIn is received before the Delay Time has passed, the Released EventOut will not be fired and the State will not change to FALSE.

12.3.40   Multi Delay

Use this Block to react on an event a certain time after it has been triggered. Multiple delays can be specified.

The Block has an Enabled PortIn, three Delay and three Value PortIns for the delay configuration, an Is Enabled PortOut, a Trigger EventIn, a Value PortOut, three State PortOuts, an Elapsed EventOut for the completed execution, and three Started EventOuts. Extended configuration is accessible at the cogwheel icon.

— Define the multiple trigger sequence in the extended configuration window.
— Connect a Link to the Trigger EventIn.

Category	Property	Description
Retrigger Behavior		Define how the Block should react when the Block is triggered while the trigger sequence is still running: Restart From Begin, Restart Last Trigger, Restart One After End, Ignore Trigger.
Delay (e.g. T0)		Define the Delay Time. Each trigger has its own Delay Time.
	Input Type	Define whether the Delay Time is a fixed value (Input Type "Property") or a received Input value (Input Type "Port").
	Value	Define the time after which the State value of a trigger will be reset to FALSE after it has been triggered to TRUE.

Value (e.g.T0)		Define the value that is added to a trigger. Each trigger has its own value.
	Date Type	The data type for the PortIn. Please note that all Value properties, PortIns and PortOut, are automatically set to the same Data Type.
	Input Type	Define whether the added trigger value is a fixed value (Input Type "Property") or a received Input value (Input Type "Port").
	Value	Define the value that is sent along with the trigger at the Value PortOut.
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.
Value		The value that is sent when its event has been triggered. You can change the Port Mode. You can change the Data Type. Please note that all Value properties, PortIns and PortOut, are automatically set to the same Data Type.
State (e.g. T0)		An actively sent bool value that becomes TRUE when the trigger is activated and turns back to FALSE after the defined Delay Time.

Defining the multiple trigger sequence in the extended configuration window

Define the multiple trigger sequence with the number of triggers, their order, and each trigger's delay time and value, in the extended configuration window:

1. Click the cogwheel icon at the right top corner of the Block to open the extended configuration window: ⚙ .
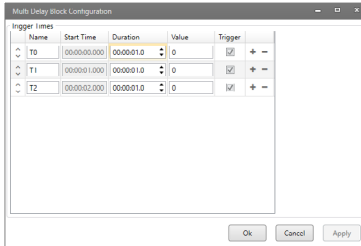
2. Define the number of triggers with the icons ➕ for adding and ➖ for removal. You can rename each trigger.

3. Define the delay time (Duration) and the Value of each trigger.

4. Define the sequence of the triggers: The order of their entries in the configuration window equals the chronology of the triggers. The top most trigger is fired first, the second row is fired afterwards and so on. Use the arrows at the front to change the sequence: ⌄ .
   The Start Time value indicates the absolute amount of time between the block is triggered and the specific trigger will be fired.

5. Dis- or enable whether to have the Started EventOut for a trigger displayed in the block. Uncheck the "Trigger" box to remove the Started EventOut.

6. Click the "OK" button to save the configuration and close the window. Click "Apply" to save the changes, but continue working in the configuration window.

The Multi Delay block has now the trigger sequence configured. All settings are implemented with Ports in the Block.

Example:
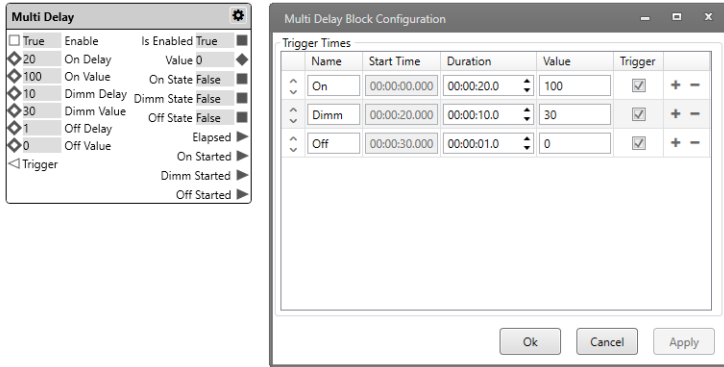
Here the Multi Delay block is used for a stepwise dimming of fixtures. The Value PortOut provides the light level for fixtures. First "On" is triggered with its Value of 100. For 20 seconds, the light is on at its maximum capacity. During this period, the On State PortOut is TRUE. After the Delay Time of 20 seconds has passed, "Dimm" is triggered. The light level is reduced to 30%. The On State value turns to FALSE and the Dimm State value becomes TRUE. And after the "Dimm" Delay Time of 10 seconds has passed, "Off" is triggered at last that turns the light off. Only Off State is TRUE for 1 second.

— The trigger properties Delay Time and Value can be defined in the Properties window of the Workflow Designer as well as in the extended configuration window of the Multi Delay block when set to the Input Type "Property". Properties of the Input Type "Port" are not editable in the extended configuration window.

12.3.41   Control Sequence
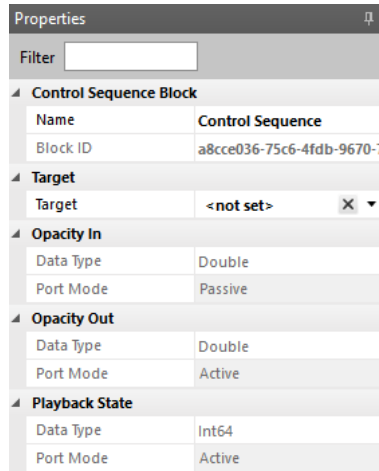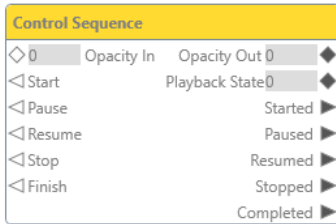
Use this Block to control the opacity and play state of a sequence or to react on play state changes, e.g. to start, stop or pause a Sequence.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has a PortIn for the opacity, five EventIns to control the play state of the Sequence, two PortOuts for opacity and playback, and five EventOuts of the play state.
The Block can be bound to a Sequence from the Content Tab.

— Define the Target Property.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to a Sequence.
Opacity In		A passively received double value of the opacity for the Sequence.
Opacity Out		An actively sent double value of the opacity of the Sequence.

Playback State		An actively sent Int64 value of the play state of the Sequence: 0: Sequence does not run (use trigger Stop or Finish) 1: Sequence is running (use trigger Start or Resume) 2: Sequence is paused (use trigger Pause).

12.3.42   Adjust Sequence
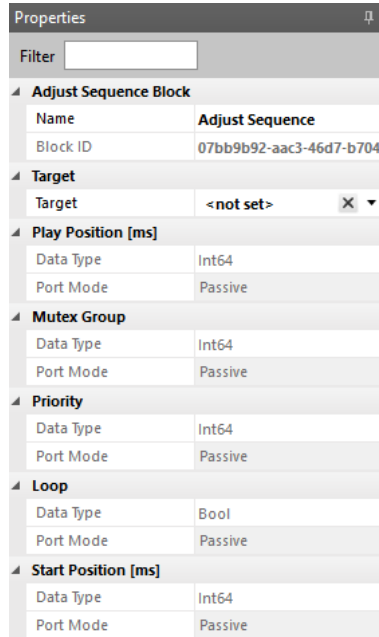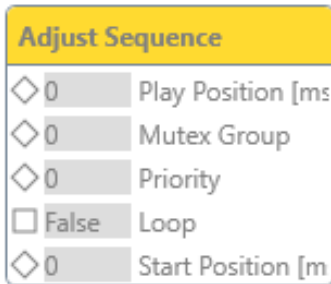
Use this Block to control some advanced parameters of a Sequence, e.g. the priority value of a Sequence.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has five PortIns to control the advanced parameters of the Sequence.
The Block can be bound to a Sequence from the Content Tab.

— Define the Target Property.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to a Sequence.

Timecode		A passively received Int64 value to jump to a specific time point of the running Sequence in milliseconds. Refers to the total playing time, e.g. looping time included.
MutexGroup		A passively received Int64 value to change the Mutex Group of the Sequence. This change does not affect the Sequence configuration in the Content Tab.
Priority		A passively received Int64 value to change the Priority for the Sequence.
Loop		A passively received bool value to dis- or enable looping the Sequence. If FALSE, but Loop Segment property in Sequence Properties is enabled, then the Sequence loops nonetheless.
StartPosition		A passively received Int64 value for the Start Position of the Sequence in milliseconds.

— All parameters are Properties of the Sequence that are configured in the Content Tab. When differing, the Properties configuration of the Content Tab is temporarily overruled by the Workflow.

12.3.43   Control Cue
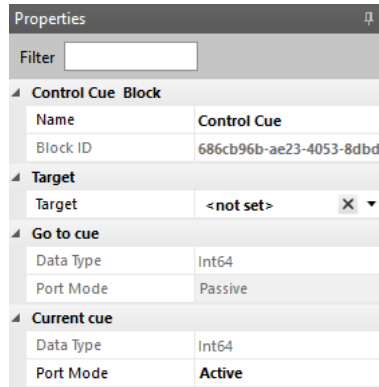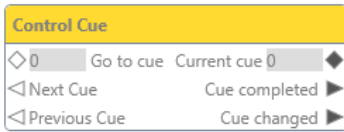
Use this Block to control Sequences based on Cues, e.g. skip to the next or previous Cue or react to completed cues of a Sequence.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has one PortIn and two EventIns to control the played Cue of the Sequence, one PortOut for the currently played Cue, and two EventOuts of Cues events.
The Block can be bound to a Sequence from the Content Tab.

— Define the Target Property.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to a Sequence.
Go to cue		A passively received Int64 value to jump to a specific Cue of the Sequence.

Current cue		An actively sent Int64 value of the current Cue. You can change the Port Mode. When the Sequence is completed, the value 1 is sent.

— The EventOut "Cue completed" is also fired when a Sequence without any Cues is completed.
— The Next Cue EventIn jumps to the next Cue. When the Sequence is configured to loop, it restarts with the first Cue after the last Cue.
— The Previous Cue EventIn jumps to the previous Cue. Even when the Sequence is configured to loop, it jumps to the first Cue at most - it does not continue with the last Cue.

12.3.44 Trigger Point
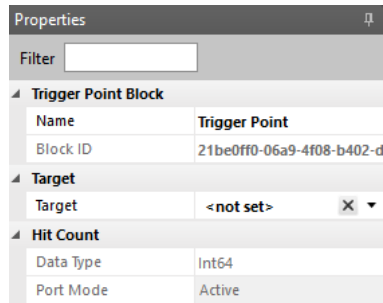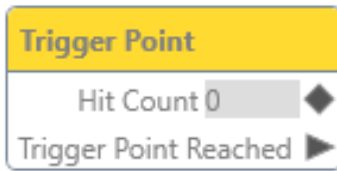
Use this Block to react on Trigger Points that have been assigned to a specific Sequence. Whenever the Sequence play time is passing a trigger point an event will be fired and the number of crossings is available.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has one PortOut for the Hit Count and one EventOut of the Trigger Point passing.
The Block can be bound to a Trigger Point of a Sequence from the Content Tab.

— Define the Target Property.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to a Trigger Point of a Sequence.
Hit Count		An actively sent Int64 value of the number of Trigger Point crossings.

12.3.45   Generic Control Sequence

Use this Block to control multiple Sequences at once and get notifications of them at one place.

The Block has four six PortIns and one EventIns to control the Sequences, an Enable PortIn, six PortOuts to get notifications of the Sequences' activities, and an Is Enabled PortOut.

— The following controls are available: start, pause, resume, stop, and finish any Sequence by sending the Sequence Identifier (string value) to the block. Set the opacity of the Grand Master in the Output Tab to a specific value with the Grand Master In PortIn. Stop all running Sequences with the Stop All EventIn.
— The following notifications are available: the last started, the last paused, the last resumed, the last stopped, and the last completed Sequence as Sequence Identifier (string value). Get the opacity of the Grand Master in the Output Tab with the Grand Master Out PortOut.
— Use this Block when you work with many Sequences to reduce the scale of the Workflow Designer.
— Use this Block to trigger Sequences with the network, e.g. connecting the Receive from Network block.

12.3.46   Mutex Group
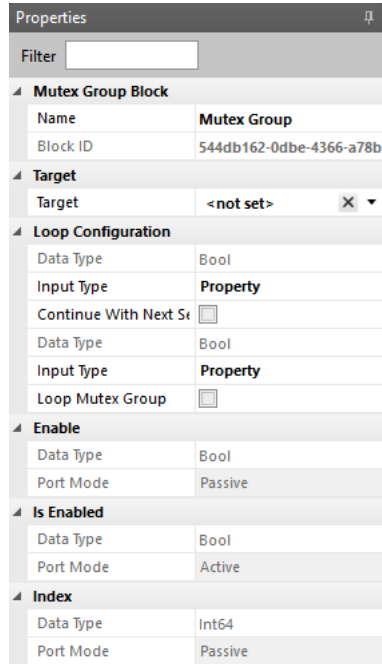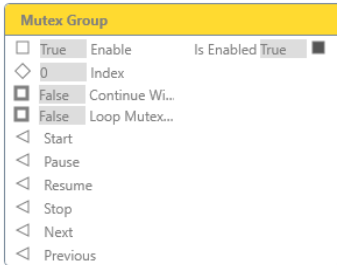
Use this Block to control Mutex Groups, e.g. to jump to the next Sequence within the same Mutex Group.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has an Enable PortIn, three PortIns to control the Mutex Group, six EventIns to control the play state of the Mutex Group, and an Is Enabled PortOut.
The Block can be bound to a Mutex Group. A Mutex Group is the abbreviation for Mutual Exclude Collection ID which is defined in the Properties of a Sequence in the Content Tab. With Mutex you can easily define a set of Sequences of which only one can run at a time. Assign the same Mutual Exclude Collection ID to all the Sequences you want to group. If the ID is zero, no mutex grouping happens.

— Define the Target Property.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to a Mutex Group.
Loop Configuration		Define the behavior of the Mutex Group.
	Input Type	Define whether the Continue With Next Sequence value is a fixed bool value (Input Type "Property") or a received bool Input value (Input Type "Port").
	Continue With Next Sequence	Enable to continue with the next Sequence when the current Sequence has finished. Continues until the last Sequence of the Mutex Group.
	Input Type	Define whether the Loop Mutex Group value is a fixed bool value (Input Type "Property") or a received bool Input value (Input Type "Port").
	Loop Mutex Group	Enable to loop the entire Mutex Group when the last Sequence has finished. Works only when "Continue With Next Sequence" is enabled.
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.

Index		A passively received Int64 value to control the Sequence within the Mutex Group that is to be played. You can find the Index value in the Content Tab: All Sequences of SYMPHOLIGHT are sorted by Index in the Sequence drop-down menu of the Timeline. The position in the menu is equivalent to the Index value. E.g. the Sequence at the very top position has the Index value of 1.

— When a Sequence has the Property "Loop Sequence" in the Content Tab enabled, the Mutex Group does not go to the next Mutex Group Sequence even though "Continue With Next Sequence" is enabled. The Mutex Group loops the enabled Sequence.

— Please note that all Sequence Properties, such as "Fade In Time", "Fade Out Time", and "Start Position" are taken into account when running a Sequence from within the Mutex Group block.

— When the Index is 0, the first or the last paused Sequence of this Mutex Group starts.

— When an Index value (Sequence) does not belong to the Mutex Group, the next belonging Index value (Sequence) is used.

12.3.47   DALI Line

DALI Line	
◇ 0	Luminance
◇ 0	Hue
◇ 0	Saturation
◇ 0	Temperature
◇ 0	Scene Number
◇ 0	Fade Time
◇ 0	Fade Rate
⬡ Color...	Color Control Mo
⬡ Logar...	Dimming Curve
⬡ Broa...	Address Type
◇ 0	Address
◁ Off	
◁ Up	
◁ Down	
◁ Step Up	
◁ Step Down	
◁ Step Down And Off	
◁ On And Step Up	
◁ Recall Last Active Level, w	
◁ Recall Min Level	
◁ Recall Max Level	

Properties	�
Filter	

DALI Line Block	
Name	DALI Line
Block ID	61e93176-dce5-48cc-8ac3-a
Target	
Target	<not set> ✕ ▾
Color Control Mode	
Data Type	DaliBallastColorControlMo
Input Type	Property
Value	RGB
Dimming Curve	
Data Type	DaliBallastDimmingCurveTy
Input Type	Property
Value	Logarithmic
Address Type	
Data Type	DaliBallastAddressType
Input Type	Property
Value	Broadcast
Address	
Data Type	Int64
Input Type	Port
Port Mode	Passive
Luminance	
Data Type	Double
Port Mode	Passive
Hue	
Data Type	Double
Port Mode	Passive
Saturation	
Data Type	Double
Port Mode	Passive
Temperature	
Data Type	Int64
Port Mode	Passive
Scene Number	
Data Type	Int64
Port Mode	Passive
Fade Time	
Data Type	Int64
Port Mode	Passive
Fade Rate	
Data Type	Int64
Port Mode	Passive

Use this block to control DALI ballasts of a line in the direct way by ballast short address, group address, and Broadcast. No matching of ballasts in the Setup Tab required.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has double PortIns for incoming properties of light and EventIns to trigger light values.
The Block can be bound to a DALI Line from the Setup Tab.

— Define the Target.

Category	Property	Description
Target		Define the target to bind the Block to a DALI Line.
Color Control Mode		Define the color control mode. By default, the value is set to the Input Type "Property" to manually define a static value. Set the Input Type to Port to passively receive an Int64 value from the counterpart Block. The respective valid numbers are described in the related Property "Value" below.
	Value	Select the color control mode: **RGB**: Generates RGB color DALI commands . When the Dimming Curve is set to Port, invoked by "0". **RGBW**: Generates RGBW color DALI commands. When the Dimming Curve is set to Port, invoked by "1".

Dimming Curve		Define the dimming mode for fading calculations. By default, the value is set to the Input Type "Property" to manually define a static value. Set the Input Type to Port to passively receive an Int64 value from the counterpart Block. The respective valid numbers are described in the related Property "Value" below.
	Value	Select the dimming mode: **Logarithmic**: . When the Dimming Curve is set to Port, invoked by "0". **Linear**: . When the Dimming Curve is set to Port, invoked by "1".
Address Type		Define the address mode of the Block. By default, the value is set to the Input Type "Property" to manually define a static value. Set the Input Type to Port to passively receive an Int64 value from the counterpart Block. The respective valid numbers are described in the related Property "Value" below.
	Value	Select the address mode: **Single**: A single ballast is addressed. When the Address Type is set to Port, invoked by "0". **Group**: A group of ballasts is addressed. When the Address Type is set to Port, invoked by "1". **Broadcast**: All ballasts on the Line are addressed. When the Address Type is set to Port, invoked by "2".

Address		A passively received Int64 value for the address of the ballast or the group. When defined as Property, configure a value.
	Value	Define an address. Dependent on the selected Address Type, enter the address for either a single ballast or a group: For the Address Type "Single", enter a value between 0 and 63. For the Address Type "Group", enter a value between 0 and 15.
Luminance		A passively received double value for the light intensity in percent. When defined as Property, configure a value between 0 and 100.
Hue		A passively received double value for the hue in degrees When defined as Property, configure a value between 0 and 360. Only relevant for DT8 or a group containing a Channel R ballast, a Channel G ballast, and a Channel B ballast (Setup Tab - Properties of a virtual ballast: Color Type).
Saturation		A passively received double value for the saturation in percent. When defined as Property, configure a value between 0 and 100. Only relevant for DT8 or a group containing a Channel R ballast, a Channel G ballast, and a Channel B ballast (Setup Tab - Properties of a virtual ballast: Color Type).

Temperature		A passively received Int64 value for the temperature in Kelvin. When defined as Property, configure a value between 2,000 and 20,000 in Kelvin. Only relevant for DT8 or a group containing a Channel Warm ballast and a Channel Cold ballast (Setup Tab - Properties of a virtual ballast: Color Type).
Scene Number		A passively received Int64 value for the scene number. The scenes are defined in the Content tab - Scene Editor, a maximum of 16 scenes with 0 as the first scene and 15 as the last. When defined as Property, configure a value between 0 and 15.

Fade Time		A passively received Int64 value for the fade time in seconds.

Seconds	Representative value
0	0
0.7	1
1	2
1.4	3
2	4
2.8	5
4	6
5.7	7
8	8
11.3	9
16	10
22.6	11
32	12
45.3	13
64	14
90.5	15

Please note: Using this PortIn overrules the fade time of the internal ballast settings (Setup Tab - Properties of the virtual ballast).
When defined as Property, configure a representative value. Only set to Property when all ballasts of the group belong to this group only.

Fade Rate		A passively received Int64 value for the fade rate in steps per second.

	Steps per seconds	Representative value
	358	0
	253	1
	179	2
	126	3
	89	4
	63	5
	45	6
	32	7
	22	8
	16	9
	11	10
	8	11
	6	12
	4	13
	2.8	14

Please note: Using this PortIn overrules the fade rate of the internal ballast settings (Setup Tab - Properties of the virtual ballast).
When defined as Property, configure a representative value. Only set to Property when all ballasts of the group belong to this group only.

The EventIns of the Block in detail:
Please note that the "light value" includes the luminance, hue, saturation, and temperature as applicable to the device type of the ballasts.

Off	Turns the group off. Please note that the group can not be turned on again using Toggle Fade triggers.

Up	Increases the light value up to the maximum value (Setup Tab - Properties of the virtual ballast: Max Level), honoring the Fade Rate of the group.
Down	Decreases the light value down to the minimum value (Setup Tab - Properties of the virtual ballast: Min Level), honoring the Fade Rate of the group. The group does not turn off as the minimum value cannot be zero.
Step Up	A fade up, honoring the Fade Rate starts for 300 ms. If retriggered during this period, another 300 ms fade continues. The maximum value is the Max Level. For continuously fading please use the Toggle Fade Up/Down by Limit EventIn instead.
Step Down	A fade down, honoring the Fade Rate starts for 300 ms. If retriggered during this period, another 300 ms fade continues. The minimum value is the Min Level. For continuously fading please use the Toggle Fade Up/Down by Limit EventIn instead.
Step Down And Off	A fade down, honoring the Fade Rate starts for 300 ms. If retriggered during this period, another 300 ms fade continues. Following the minimum value, the group is turned off. For continuously fading please use the Toggle Fade Up/Down by Limit EventIn instead.
On And Step Up	The group is turned on and a fade up, honoring the Fade Rate starts for 300 ms. If retriggered during this period, another 300 ms fade continues. For continuously fading please use the Toggle Fade Up/Down by Limit EventIn instead.
Recall Last Active Level	The group is switched on to the last active light value (value when the Off was triggered), respecting the Fade Time. When triggered after a system start, the maximum value (Setup Tab - Properties of the virtual ballast: Max Level) of the reference ballast is used. Works only on DALI 2 ballasts!

Recall Min Level	The group is switched to the Min Level value (Setup Tab - Properties of the virtual ballast: Min Level), respecting the Fade Time.
Recall Max Level	The group is switched to the Max Level value (Setup Tab - Properties of the virtual ballast: Max Level), respecting the Fade Time.

— Please note that for linear dimming purposes DALI ballasts of DT6 with linear dimming curve support are required.
— Please note that for color control mode support DALI ballasts of DT8 Color Temperature are required. Or when changing Hue, Saturation or Temperature.
— Please use the DALI Line block only for DALI Lines that contains unmatched ballasts only to avoid controlling conflicts.
— Please note that short address or group address of unmatched ballasts will only be displayed in a later release of SYMPHOLIGHT.
— For Hue values, relate to the following gradient:

12.3.48   DALI Group and Daylight Harvesting

Use this Block to control your DALI Groups. You can also configure Daylight Harvesting.

This Block is target based. The Target Value is displayed as main setting in the Block's header.

When orphan, the Block has all the properties of a bound Block available.

The Block has double PortIns and PortOuts for incoming and forwarded properties of light and EventIns to trigger light values. Extended configuration is accessible at the cogwheel icon.

The Block can be bound to a DALI Ballasts group as defined in the Setup Tab.

— Define the Target.
— Define the Reference Ballast that act as representative for DALI Settings. Important when one of these properties differ within the group's ballasts.
— Define dynamic scenes for Daylight Harvesting in the extended configuration window.

In the following list, the properties without a Port are explained:

Category	Properties	Description
Target		Define the target to bind the Block to a DALI group.
Reference Ballast		Define a ballast whose Fade Time and Fade Rate settings (Setup Tab - Properties of a virtual ballast: DALI Settings) are attributed to the whole group.
Toggle Scene Numbers		Define the scenes between which you want to switch when triggering the Toggles Scenes EventIn. The scenes are defined in the Content tab - Scene Editor, a maximum of 16 scenes with 0 as the first scene. Use the numbers of the scenes, separated by commas. Valid are numbers between -1 and 15; the number "-1" switches the group off. The order of the numbers defines the sequence of the scenes. Example: "0, 1, 2, -1". Go to scene 0, go to scene 1, go to scene 2 and switch the group off.

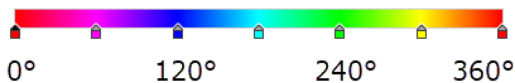In the following list, all PortIns, PortOuts EventIns and EventOuts are explained. Please note that the "light value" includes the luminance, hue, saturation, and temperature as applicable to the device type of the group. Please note as well that most of the EventIns are only applicable for DALI DT8 devices:

Port	Description
Luminance	A passively received double value for the light intensity in percent. When defined as Property, configure a value between 0 and 100.
Hue	A passively received double value for the hue in degrees. When defined as Property, configure a value between 0 and 360. Only relevant for DT8 or a group containing a Channel R ballast, a Channel G ballast, and a Channel B ballast (Setup Tab - Properties of a virtual ballast: Color Type).
Saturation	A passively received double value for the saturation in percent. When defined as Property, configure a value between 0 and 100. Only relevant for DT8 or a group containing a Channel R ballast, a Channel G ballast, and a Channel B ballast (Setup Tab - Properties of a virtual ballast: Color Type).
Temperature	A passively received Int64 value for the temperature in Kelvin. When defined as Property, configure a value between 2,000 and 20,000 in Kelvin. Only relevant for DT8 or a group containing a Channel Warm ballast and a Channel Cold ballast (Setup Tab - Properties of a virtual ballast: Color Type).
Scene Number	A passively received Int64 value for the scene number. The scenes are defined in the Content tab - Scene Editor, a maximum of 16 scenes with 0 as the first scene and 15 as the last. When defined as Property, configure a value between 0 and 15.

| Fade Time | A passively received Int64 value for the fade time in seconds. A valid representative value needs to be received for the Block to execute the according fade time in seconds. |

Seconds	Representative value
0	0
0.7	1
1	2
1.4	3
2	4
2.8	5
4	6
5.7	7
8	8
11.3	9
16	10
22.6	11
32	12
45.3	13
64	14
90.5	15
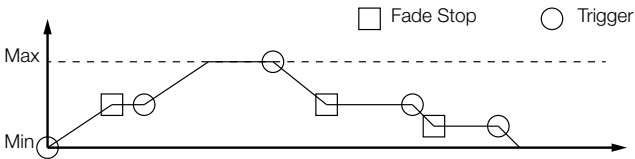
Please note: Using this PortIn overrules the fade time of the internal ballast settings (Setup Tab - Properties of the virtual ballast).
When defined as Property, configure a representative value. Only set to Property when all ballasts of the group belong to this group only.

Fade Rate	A passively received Int64 value for the fade rate in steps per second. A valid representative value needs to be received for the Block to execute the according fade rate in seconds.

Steps per seconds	Representative value
358	0
253	1
179	2
126	3
89	4
63	5
45	6
32	7
22	8
16	9
11	10
8	11
6	12
4	13
2.8	14

Please note: Using this PortIn overrules the fade rate of the internal ballast settings (Setup Tab - Properties of the virtual ballast).
When defined as Property, configure a representative value. Only set to Property when all ballasts of the group belong to this group only

On	The group is switched on to the last active light value (value when the Off was triggered), respecting the Fade Time. When triggered after a system start, the maximum value (Setup Tab - Properties of the virtual ballast: Max Level) of the reference ballast is used. Use ONLY with DALI DT8 devices.

Off	Turns the group off. Please note that the group can not be turned on again using Toggle Fade triggers. Use ONLY with DALI DT8 devices.
Up	Increases the light value up to the maximum value (Setup Tab - Properties of the virtual ballast: Max Level), honoring the Fade Rate of the group. Use ONLY with DALI DT8 devices.
Down	Decreases the light value down to the minimum value (Setup Tab - Properties of the virtual ballast: Min Level), honoring the Fade Rate of the group. The group does not turn off as the minimum value cannot be zero. Use ONLY with DALI DT8 devices.
Step Up	A fade up, honoring the Fade Rate starts for 300 ms. If retriggered during this period, another 300 ms fade continues. The maximum value is the Max Level. For continuously fading please use the Toggle Fade Up/Down by Limit EventIn instead. Use ONLY with DALI DT8 devices.
Step Down	A fade down, honoring the Fade Rate starts for 300 ms. If retriggered during this period, another 300 ms fade continues. The minimum value is the Min Level. For continuously fading please use the Toggle Fade Up/Down by Limit EventIn instead.
Step Down And Off	A fade down, honoring the Fade Rate starts for 300 ms. If retriggered during this period, another 300 ms fade continues. Following the minimum value, the group is turned off. For continuously fading please use the Toggle Fade Up/Down by Limit EventIn instead. Use ONLY with DALI DT8 devices.

On And Step Up	The group is turned on and a fade up, honoring the Fade Rate starts for 300 ms. If retriggered during this period, another 300 ms fade continues. For continuously fading please use the Toggle Fade Up/Down by Limit EventIn instead. Use ONLY with DALI DT8 devices.
Recall Last Active Level	The group is switched on to the last active light value (value when the Off was triggered), respecting the Fade Time. When triggered after a system start, the maximum value (Setup Tab - Properties of the virtual ballast: Max Level) of the reference ballast is used. Use ONLY with DALI DT8 devices.
Recall Max Level	The group is switched to the Max Level value (Setup Tab - Properties of the virtual ballast: Max Level), respecting the Fade Time. Use ONLY with DALI DT8 devices.
Recall Min Level	The group is switched to the Min Level value (Setup Tab - Properties of the virtual ballast: Min Level), respecting the Fade Time. Use ONLY with DALI DT8 devices.
Toggle On / Off	Switches between turning the group on and off. Use ONLY with DALI DT8 devices.
Toggle Fade Up / Down Start by Direction	 The group fades until a fade stop (Toggle Fade Up / Down Stop) happens or Min / Max Level is reached. After another trigger, the fading changes direction until a fade stop comes or Min / Max Level is reached. Use ONLY with DALI DT8 devices.

Toggle Fade Up / Down Start by Limit	The group fades until a fade stop (Toggle Fade Up / Down Stop) happens or Min / Max Level is reached. After another trigger, the fading continues until Mix / Max Level or another fade stop is reached. The direction does not change. Use ONLY with DALI DT8 devices.
Toggle Fade Up / Down Stop	Stops the fading of Toggle Fade Up / Down Start by Direction or by Limit. Use ONLY with DALI DT8 devices.
Toggle Scenes	Switches between the scenes that are defined in the Toggle Scene Numbers property.
Current Scene No	An actively sent Int64 value of the currently running scene number. The scenes are defined in the Content tab - Scene Editor, a maximum of 16 scenes with 0 as the first scene and 15 as the last. When light values are changed by another Port than Toggle Scenes or Scene Numb., e.g. Step Up, the current scene number is always "-1".
Target Luminance	An actively sent Int64 value of the target luminance value (in percent) to be reached or already in place by the reference ballast. Fades are not respected. 0 is off.
Target Hue	An actively sent Int64 value of the target hue value (between 0 and 360) to be reached or already in place. Fades are not respected.
Target Saturation	An actively sent Int64 value of the target saturation value (in percent) to be reached or already in place. Fades are not respected.

Target Temperature	An actively sent Int64 value of the target temperature value in Kelvin to be reached or already in place. Fades are not respected.
State Text	An actively sent string value of the group's state. When a scene is running, the state text equals the Display Name of the running scene number as defined in the extended configuration window. When the Current Scene No is "-1", the Target Luminance percentage is used.
All Ballasts Ready	Sends a trigger when the initializing is completed after a system start. Fires only once.

Defining scenes for Daylight Harvesting in the extended configuration window
SYMPHOLIGHT supports Daylight Harvesting for DALI devices. Daylight harvesting systems use daylight to reduce the amount of electric lighting needed to properly light a space, in order to reduce energy consumption. You will need a light sensor to measure the amount of natural light, the received value is used to control the intensity for the DALI Group. The formulas to calculate the target light values are to be defined in the extended configuration window behind the cogwheel icon of the DALI Group block. The scenes defined are dynamic and replace the static scenes from the Content Tab.
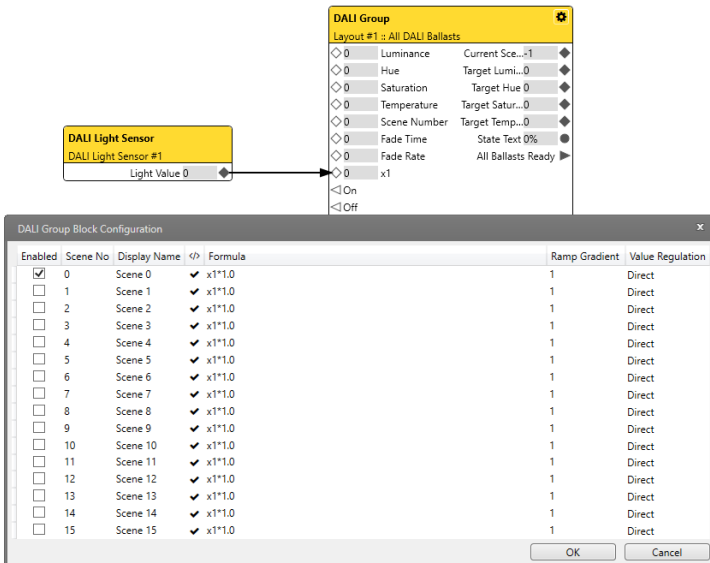
Define scenes for Daylight Harvesting in the extended configuration window:

Column	Description
Enabled	Enable to make the scene available when called. When the scene is already defined in the Scene Editor of the Content Tab, the configuration for Daylight Harvesting overwrites this definition.
Scene No	The scene number, ranging between 0 and 15. The scenes are otherwise defined in the Scene Editor of the Content Tab.
Display Name	Define a name for the Daylight Harvesting scene. The Display Name is forwarded in the State Text PortOut when the scene is running.
</>	Gives information about the Formula: ✔ : The Formula is valid. PortOuts for the variables of the Formula are created and the scene with the Formula are executed when selected. ⚠: The Formula is erroneous. PortOuts for the variables of the Formula are not created. When enabled and the scene selected, the Formula is not executed.
Formula	Define the mathematical function for Daylight Harvesting. All mathematical functions, like sin(x1), are valid. Every variable has to start with "x" followed by a number, e.g. x1. Further letters are possible, e.g. x1Ceiling. The same variable can be used in multiple scenes.
Ramp Gradient	When "RampByGradient" in Value Regulation is selected, define a luminance value (percent) per second. The higher the Ramp Gradient value, the faster the target light values will be reached. Please be aware that this might stress the DALI bus.

Value Regulation	Define the way of dimming the Daylight Harvesting scene: **Direct**: The target light values will be set with the Fade Time configured in the ballasts of the group. **RampByGradient**: A fading time longer than the maximum duration of 90 ms is possible using RampByGradient. SYMPHOLIGHT is calculating immediate steps, considering the Ramp Gradient in (percent) values per second, until the target light values are reached.

1. Click the cogwheel icon at the right top corner of the Block to open the extended configuration window: ⚙.
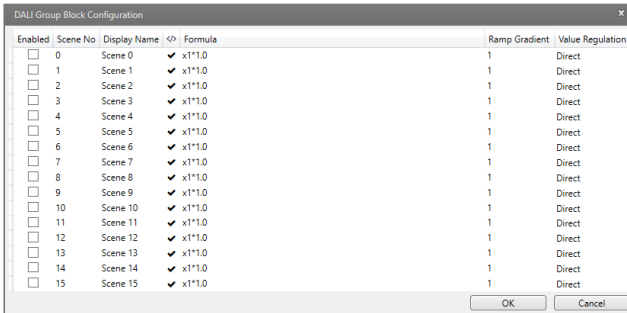
2. Define the Formulas. Use variables representing PortIns. Variable names have to start with x and a following number. If not correct, a warning icon instead of a check mark appears. The scene is still enablable but is neither executed nor new PortIns created.
3. Define the Value Regulation as RampByGradient for a fading longer than 90 ms. If the Value Regulation is defined as RampByGradient, define the Ramp Gradient.
4. Enable scenes as desired by clicking the Enabled check box. For each enabled variable a double PortIn is created.
5. Click the "OK" button to save the configuration and close the window.

The DALI Group block has now the Daylight Harvesting scenes configured and the PortIns for each enabled variable accordingly available:

Category	Description

x1	Additional PortIn for controlling the light value, created in the extended configuration window for Daylight Harvesting. Use variables starting with x and a following number in the formulas to create additional inputs. When defined as Property, configure the data type (double or Int64) and the value.

— The PortOuts for light values are not updated as long as the group consists of only virtual devices and no real devices are matched.

— SYMPHOLIGHT only processes changes in values. Has the same value been incoming, a second calculation or processing does not take place; the first processed value is still valid.

— The Target Luminance might vary for decimal places from the Luminance PortIn depending on the chosen DALI settings. The Target Luminance is the exact technical value the DALI system has produced at the Reference Ballast.

— For Hue values, relate to the following gradient:
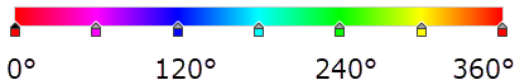
— In Daylight Harvesting, the same variable can be used in multiple scenes. The Current Scene No determines which formula the variable is allocated to and the result is accordingly calculated.

— Please configure all DALI ballast of one group identical at the following properties (Setup Tab - Properties of the virtual ballasts: DALI Settings): Fade Rate, Fade Time, Min Level, Max Level, and Dimming Curve. Otherwise irregular light values within the group will occur. Further take care to only group ballast of compatible types: Intensity and DT8; Channel Warm, Channel Cold, and Color Temperature DT8; Channel Red, Channel Green, Channel Blue, Color RGB DT8, and Color RGBW DT8.

— Please consider that the fading process is regulated by the reference ballast.

— Imitate an RGB DT8 ballast by configuring three ballasts with Channel Red, Channel Green, and Channel Blue respectively.

— Imitate a Temperature DT8 ballast by configuring two ballasts with Channel Cold and Channel Warm respectively.

— When ballasts are configured for DT8 or its equivalent by imitation, the PortOuts for Target Hue, Target Saturation, and / or Target Temperature are set to the respective minimum values.

— Please consider that enabled scenes from the extended configuration window of the DALI Group block overrule scene values defined in the Content Tab or in the ballasts of the fixtures.

— Please note that the Reference Ballast has to be online for the Block to be fully testable.

Implementing Daylight Harvesting behavior:

— Add a light sensor to the project.
— Add a DALI Light Sensor block to the Workflow.
— Add a real and a virtual DALI ballast to the project (necessary: License dongle, SYMPL dali Node) and match them.
— Add a DALI Group block to the Workflow.
— Define a scene in the extended configuration window for Daylight Harvesting, enable it, and click OK.
— Connect the sensor's Light Value PortOut to the new PortIn for the Daylight Harvesting scene, here x1.

The implementation of Daylight Harvesting is completed.

12.3.49   DALI Button
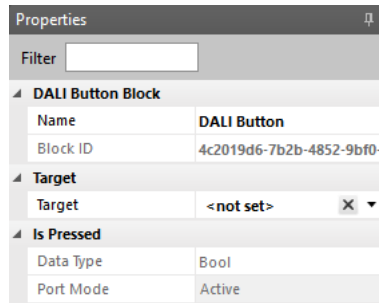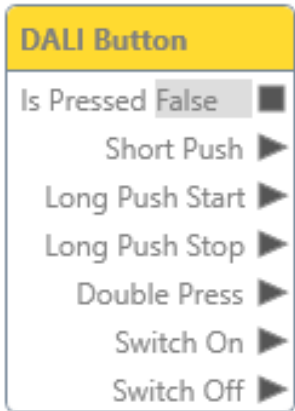
Use this Block to include DALI buttons in the Workflow Designer, e.g. to switch on/off lights based on button presses.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has an Is Pressed PortOut and six EventOuts.
The Block can be bound to a DALI Button (Devices Library in Setup Tab: Fixtures - Osram).

— Define the Target.
— Configure the DALI Button in the Setup Tab.

Category	Description
Target	Define the target to bind the Block to a DALI Button.
Is Pressed	An actively sent bool value whether the bound target is being pressed or not.

— Switch On fires when the state is TRUE. Switch Off fires when the state is FALSE.
— You can configure the DALI Button and its Block either for push buttons sending Push and Press commands OR for switches using Switch behavior. The Block always shows all EventOuts, although some are idle due to this configuration. Define the configuration in the Setup Tab: In the Properties of the DALI Button, disable "Plug

and Play". You can now select three different Press events for push buttons OR the Switch behavior. SYMPHOLIGHT prevents you to select from both types.

— Connect the DALI Button block to the Toggle, a Single Delay or a Multi Delay block to further elaborate the handling of DALI Button events.

12.3.50   DALI Presence Sensor
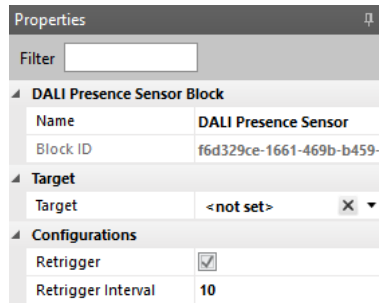
Use this Block to include DALI presence detection sensors (PIR) in the Workflow Designer, e.g. to switch on/off lights based on presence information.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has two EventOuts for motion start and end points.
The Block can be bound to a DALI PIR (Devices Library in Setup Tab: Fixtures - Osram).

— Define the Target.

Category	Description
Target	Define the target to bind the Block to a DALI PIR.
Retrigger	Enable to trigger a Motion End event if no new Motion Start event occurs within the Retrigger Interval.
Retrigger Interval	Define the interval in seconds for the Retrigger property after which a Motion End event is sent if no new Motion Start event occurred.

— Motion Start fires when a motion is detected. Motion End fires when no more motion is detected in the Retrigger Interval and Retrigger is active.

12.3.51   DALI Light Sensor
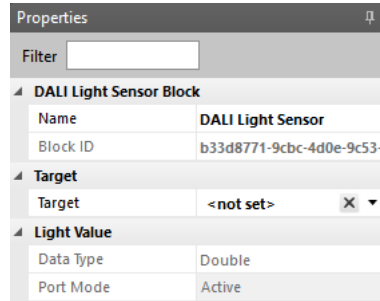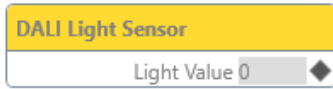
Use this Block to process DALI light sensor values in the Workflow Designer, e.g. for Daylight Harvesting.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has one PortOut for the light value in percent.
The Block can be bound to a DALI Light Sensor (Devices Library in Setup Tab: Fixtures - Osram).

— Define the Target.

Category	Description
Target	Define the target to bind the Block to a DALI Light Sensor.
Light Value	An actively sent double value of the light value in percent (0 - 100). The light value in lux is defined in the DALI Settings of the DALI Light Sensor's properties in the Setup Tab.

— Define the DALI settings as the sensor values in lux and the send intervals of the DALI Light Sensor in the Properties, in the Setup Tab.

12.3.52   DALI Observer

Use this Block to process DALI information that can be collected from DALI fixtures.

The Block has one Enable PortIn, four PortOuts for the DALI information and one EventOut for changed status triggers.
The Block can be assigned to a target. Targets are virtual DALI fixtures.

— Define the DALI devices to listen to at the Block Target.
— Define the Event Configuration.

Category	Property	Description
Event Configuration		Define what type of information you want to retrieve from the DALI fixture. Set an Event Category and, where applicable, the information type.
	Event Category	Define the event type to which this Block should react to. The PortOut "Event Category" sends the chosen type: **Any**: All available event types are considered. **Connection Type**: Events of scanning the system. **Warning Type**: Events of minor negative impact. **Error Type**: Events of major negative impact.
	Connection Type	Define the information type for connection events. "Any" considers all available information types.
	Warning Type	Define the information type for warning events. "Any" considers all available information types.
	Error Type	Define the information type for error events. "Any" considers all available information types.

Device Configuration		Define the DALI targets.
	Block Target	Define the selection mechanism for DALI fixtures and specify the DALI fixtures. **All DALI Devices**: All connected DALI fixtures. **DALI Device**: Listen to one single DALI fixture. **DALI Group**: Listen to a group of DALI fixtures.
	DALI Device	Block Target is DALI Device: Select from the drop-down list of available DALI fixtures.
	Layout	Block Target is DALI Group: Select from the drop-down list of available Layouts.
	DALI Group	Block Target is DALI Group: Select from the drop-down list of available Groups.
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not.
Source		An actively sent string value of the DALI ballast's Display Name. The Display Name is configured in the Setup Tab - Properties of the virtual ballast: Device Information.
Event Category		An actively sent string value of the event category.
Event Type		An actively sent string value of the event type.

Port	Description
Status Changed	Sends a trigger when any of the PortOuts value has changed, that is when the defined information occurs.

— Please note that most information types are only discovered after a sent request of the DALI system. The timing of sending requests depends on the traffic and the amount of ballasts in the DALI system. It might take 10 minutes until a request is sent and an event information is received after happening. Missing LAN connectivity is, as exception, discovered and received immediately (Connection event - Device Lost).

— Use a Combine Text block to attach all string PortOuts and forward the combined information.

12.3.53 Build RDM Command
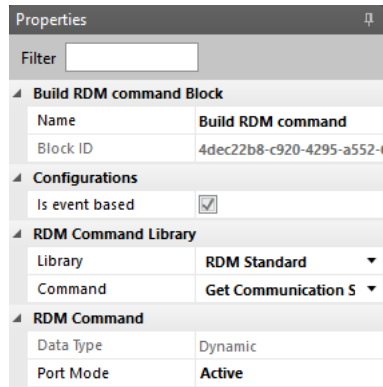
Use this Block to process RDM information that can be collected from RDM fixtures, e.g. temperatures.

Please consider the following preconditions:

— Before working with the RDM system in the Workflow Designer, do not forget to enable RDM for the used DMX port in the RDM subwindow at the button of the Setup window. See „10.14 The RDM Window" on page 51 for details.

Use this Block to build simple RDM commands that can be send to RDM devices using the Send RDM Block or scripting.

The Block has initially one EventIn and one PortOut for the command.

— Define the Library of RDM commands.
— Define the Command.

Category	Property	Description
Is event based		Enables this block to be executed on event trigger.
RDM Command Library		Define the library and the specific RDM command from within the chosen library.
	Library	Select the library to choose the RDM command from.

	Command	Select the RDM command to create.
RDM Command		The actively sent dynamic command.

— For detail information about RDM and using RDM in SYMPHOLIGHT, please see „15 RDM" on page 343.

12.3.54   Send RDM
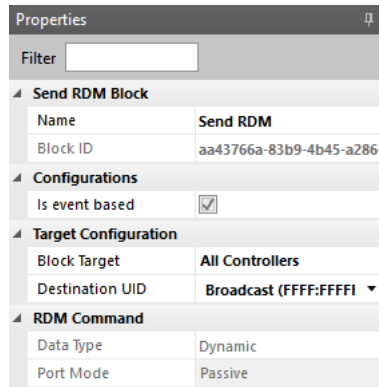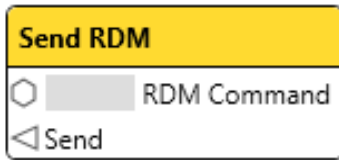
Use this Block to process RDM information that can be collected from RDM fixtures, e.g. temperatures.

Please consider the following preconditions:

— Before working with the RDM system in the Workflow Designer, do not forget to enable RDM for the used DMX port in the RDM subwindow at the button of the Setup window. See „10.14 The RDM Window" on page 51 for details.

The Send RDM block is only functional if RDM is enabled in Setup.

Use this Block to send RDM commands to specified destination remote devices.

The Block has initially one EventIn and one PortIn for receiving the RDM command.

— Define the Target Configuration.

> ⚠ While working with the Send RDM block, we strongly suggest to configure the Block while the real fixture(s) is/are connected to the DMX port. While doing so, the vendor specific parameters are scanned from the fixture(s) directly and the RDM Parameters are filled with valid entries only.

Category	Property	Description
Is event based		Enables this block to be executed on event trigger.
Target Configuration		Define the library and the specific RDM command from within the chosen library.

	Block Target	Select the target type to send the RDM command.
	Destination UID	The identifier of the device to be handled. Format: ManufacturerId:DeviceId:SubDeviceIndex E.g.: 45a3:1234567:1 or 45a3:1234567:65535 to perform SUB_DEVICE_ALL_CALL
	Controller	Select the RDM controller to be used for sending of RDM requests.
	Port	Select the RDM controller port to be used for sending of RDM requests.
RDM Command		The passively received dynamic command.

— For detail information about RDM and using RDM in SYMPHOLIGHT, please see „15 RDM" on page 343.

12.3.55 Observe RDM

Use this Block to process RDM information that can be collected from RDM fixtures, e.g. temperatures.

Please consider the following preconditions:

— Before working with the RDM system in the Workflow Designer, do not forget to enable RDM for the used DMX port in the RDM subwindow at the button of the Setup window. See „10.14 The RDM Window" on page 51 for details.

The Observe RDM block is only functional if RDM is enabled in Setup.

The Block has initially three PortOuts for the RDM information.

The Block can be assigned to a target. Targets are real RDM fixtures that are connected to the project (see "Implementing RDM fixtures" the end of the chapter). When assigned to a target, more configuration options are available.

— Define the RDM fixtures to listen to at the Block Target.
— Define the Parameters.

While working with the Observe RDM block, we strongly suggest to configure the Block while the real fixture(s) is/are connected to the DMX port. While doing so, the vendor specific parameters are scanned from the fixture(s) directly and the RDM Parameters are filled with valid entries only.

Category	Property	Description
Block Configuration		Define the RDM targets and the number of available information represented by PortOuts.
	Number of Parameters	Define the number of parameters for the targets of the Block. Each information type is an attribute of a parameter.
	Block Target	Define the selection mechanism for RDM fixtures. Specify the RDM fixtures at Device Configuration. **Any**: All connected RDM fixtures. **Single Device**: Listen to one single RDM fixture. **Layout**: Listen to a group of RDM fixtures.
Device Configuration		Depending on the Block Target settings, either both Layout properties are to be defined or the Single Device Identifier. Specify the RDM fixtures to listen to.
	Layout	Block Target is Layout: Select the Layout that contains the desired RDM fixtures. Prerequisite: real RDM fixtures have to be matched to virtual fixtures.
	Layout Group	Block Target is Layout: Select a group within the Layout. Prerequisite: real RDM fixtures have to be matched to virtual fixtures.
	Single Device Identifier	Block Target is Single Device: Select from the drop-down list of available RDM fixtures. Prerequisite: RDM has to be enabled in the Setup Tab.

Parameter Configuration		Each type of information is an attribute of a parameter. Set the Parameter Identifier first, then the Attribute Identifier.
	Parameter Text	The Port description in the block. By default the Parameter Identifier name.
	Parameter Identifier	Select the identifier of the RDM parameter from the drop-down list. Listed from the RDM scan of the devices. The list is also displayed in the Setup Tab: select the RDM fixture in the RDM pane. In the Properties window, Supported Parameters (Standard Parameters) lists the RDM parameters.
	Attribute Identifier	Select the identifier of the attribute from the drop-down list. The list varies according to the selected parameter.
Condition Configuration		Some attributes need a minimum value and a maximum value to set a range for valid values. The unit of the values depend on the selected attribute. When defined, two additional EventOuts for the value limits are created.
	Minimum Value	Define a minimum value. When the received value from the fixture is smaller, the Underflow EventOut is triggered.
	Maximum Value	Define a maximum value. When the received value from the fixture is larger, the Overflow EventOut is triggered.

Source		An actively sent string value of the RDM device name. The device name is displayed in the Setup Tab - RDM pane - Device; or Properties of the virtual device: Device Info - Device ID. Always the latest fixture that had sent a changed information.
Value 1		An actively sent value of the data type dynamic containing the information about the RDM fixture as defined in the Parameter Configuration. The PortOut is named as defined at Parameter Text.

Port	Description
Device Found	Sends a trigger when a new RDM fixture is discovered.
Device Lost	Sends a trigger when an RDM fixture has gone missing.

— It is recommended to preconfigure the Block with real devices connected.
— Please note that the available identifiers are the union of all RDM fixtures' parameters when the Block Target is "Any" or "Layout". To ensure identical information behavior, the RDM fixtures should be of the same device type, the same firmware version, and of the same RDM Personality.
— The Parameter Identifier are based on the RDM standard. For further information on the topic please refer to the standard. Also consult the manual of the RDM fixtures.

Implementing RDM fixtures:
— You need a License dongle and a SYMPHOLIGHT project running in Full Mode.
— Add an RDM capable controller to the project: Butler S2, Butler XT2, Butler Pro, SYMPL dmx Node or SYMPL e:pix Node. Define the DMX universes at the Properties window - Address Mapping (Universe = Port).
— Add virtual RDM fixtures to the project (customized fixtures) and map DMX universes and addresses to them at their Properties.
— Connect the real RDM fixtures to the controller where applicable.
— Go to the RDM pane in the RDM / DALI window. Enable RDM for Ports of the controller in its context menu (right-click). A full heart icon indicates the enabled RDM. The scanning of RDM capable devices and retrieving of information from the fixtures starts (stethoscope icon). A list of discovered fixtures is added to each Port.

— The Device column displays the Device ID needed for Single Device target assigning of the Observe RDM block. You can change the Label (DeviceLabel) for additional identification. The label is stated before the device name in the Single Device Identifier of the Observe RDM block.

— Map DMX addresses to the RDM fixtures: Select an RDM fixture from the list. In the Properties window, define the DmxStartAddress. Use the Port / DMX universe and the DMX address of the virtual fixture you want to map the RDM fixture to. This step is necessary for SYMPHOLIGHT when using the Layout target assignment of the Observe RDM block.

— Define groups of RDM fixtures in the 3D canvas as desired; these groups are available for Layout target assignment. Consider the implications for parameter configuration when creating a heterogeneous group.

The implementation of RDM fixtures is completed.

— For more detail information about RDM and using RDM in SYMPHOLIGHT, please see „15 RDM" on page 343.

12.3.56 Analog / Digital Contact
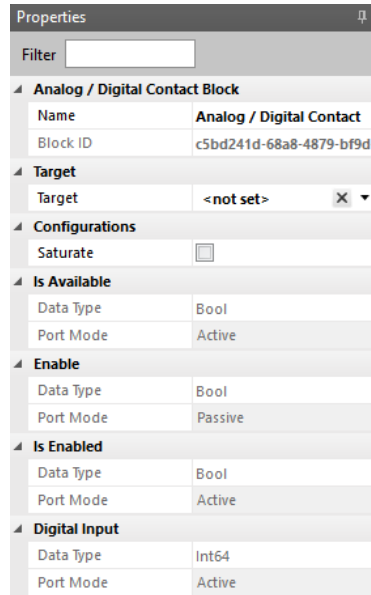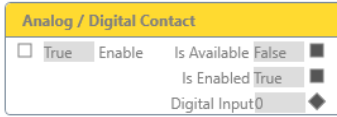
Use this Block to react on the values of analog inputs of e.g. the SYMPL input Node and process these values in the Workflow Designer.

This Block is target based. The Target Value is displayed as main setting in the Block's header and its type defines the Block title.
When initially orphan, the Block has an Enable PortIn, an Is Available PortOut, an Is Enabled PortOut, and one PortOut for a received digital input value. When orphan after being target based once, the Block keeps the target based Ports facilitating an exchange of targets.
The Block can be bound to a SYMPL input Node.

— Define the Target.

Category	Description
Target	Define the target to bind the Block to a SYMPL input Node.

Saturate	Enable to saturate underflow / overflow values to the Target value range. The Port values will be limited so that overflow and underflow values cannot exceed the Target value range.
Is Available	An actively sent bool value whether the bound target is available or not. For SYMPL nodes, the condition is automatically TRUE when a real Node is matched.
Enable	A passively received bool value to dis- or enable the Block.
Is Enabled	An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.
Digital Input	An actively sent Int64 value that is composed from the digital representation of all input values. Base are the threshold settings in the input Nodes' properties in the Setup Tab.

— For each target value, the Analog Input properties in the Setup Tab need to be configured. Please define for every available Analog Input port the Digital Threshold Low, the Digital Threshold High and the Event Delta. Please consider that this configuration has influence on the Digital Input port of the Block as well as the visualization of the port LEDs on the SYMPL input Node.

SYMPL input Node based:

In addition to the properties of an orphan Analog / Digital Contact block, the SYMPL input Node based Block has eight PortOuts for the received input values and 16 EventOuts for overflow and underflow triggers.

In addition to the properties of an orphan block:
— Define the Source Value range.
— Define the Target Value range.

Category	Property	Description
Input		An actively sent double value that is received from the connected sensor.
	Input Minimum Source Value	The minimum allowed source value in millivolt. Any value below will fire an underflow event.
	Input Maximum Source Value	The maximum allowed source value in millivolt. Any value above will fire an overflow event.
	Input Minimum Target Value	The minimum allowed value in unit of the sensor. The minimum source value will be mapped to this value.
	Input Maximum Target Value	The maximum allowed value in the unit of the sensor. The maximum source value will be mapped to this value.

12.3.57   Dry Contact
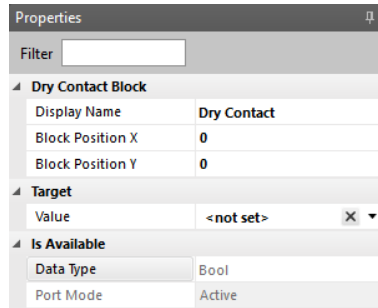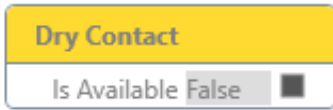
Use this Block to receive dry contact triggers from, e.g. the LCE2.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has an Is Available PortOut.
The Block can be bound to an LCE2 or LCE3 (see „LCE2 and LCE3 based:" on page 260), a Butler XT, or a Butler XT2 (see „Butler XT and Butler XT2 based:" on page 261).

— Define the Target.

Category	Description
Target	Define the target to bind the Block to: an LCE2, an LCE3, a Butler XT, or a Butler XT2.
Is Available	An actively sent bool value whether the bound target is available or not.

LCE2 and LCE3 based:
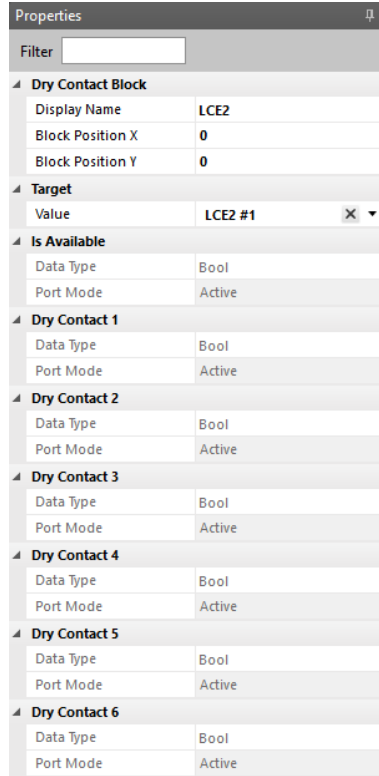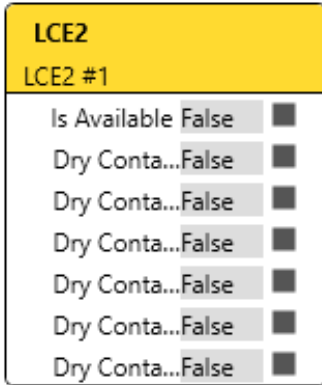
In addition to the properties of an orphan Dry Contact block, the LCE2 / LCE3 based Block has six PortOuts for six dry contacts.

— Connect the PortOuts to counterpart Blocks.

Category	Description
Dry Contact	An actively sent bool value of the state of the dry contact (FALSE: low voltage or open; TRUE: high voltage).

Butler XT and Butler XT2 based:

The Butler XT based Block and the Butler XT2 based Block are identical to the LCE2 based Block, but with eight PortOuts for eight dry contacts.

— Connect the PortOuts to counterpart Blocks.

12.3.58   Relay
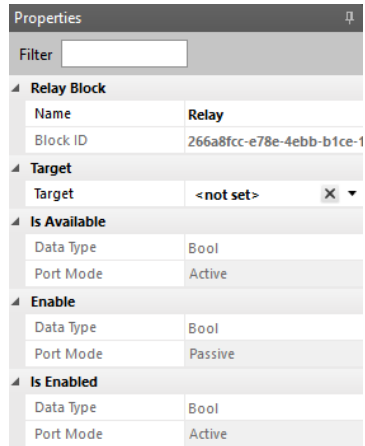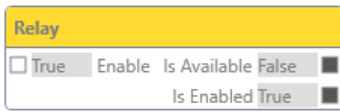
Use this Block to control the relay outputs of e:cue devices, e.g. the SYMPL relay Node.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has an Enable PortIn, an Is Available PortOut, and an Is Enabled PortOut.
The Block can be bound to an LCE2, an LCE3, or a SYMPL relay Node. When target based, the Block has additionally EventIns for the relay controls:
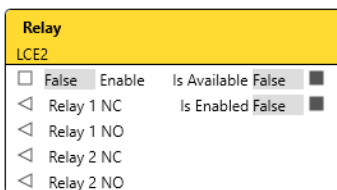
— Define the Target Property.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to an LCE2, an LCE3, or a SYMPL relay Node.

Is Available		An actively sent bool value whether the bound target is available or not.
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.

Trigger the NC event to close the NC contact and trigger the NO event to close the NO contact.

12.3.59   Observe Controller
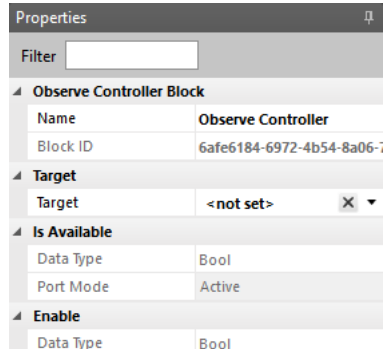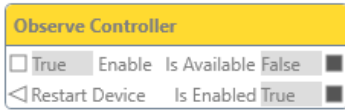
Use this Block to control and fetch information from e:net devices.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has an Enable PortIn, one EventIn, an Is Available PortOut and an Is Enabled PortOut.
The Block can be bound to a controller.

— Define the Target.

Category	Property	Description
Target		Define the target controller to bind the Block to.
Is Available		An actively sent bool value whether the bound target is available or not.
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.

— For each target controller the Block is bound to different PortOuts are available. They do not need configuration.

12.3.60   Observe System

Use this Block to retrieve various information about the running instance and the project.

The Block has 20 PortOuts. All of them are fixed.

12.3.61 Receive DMX
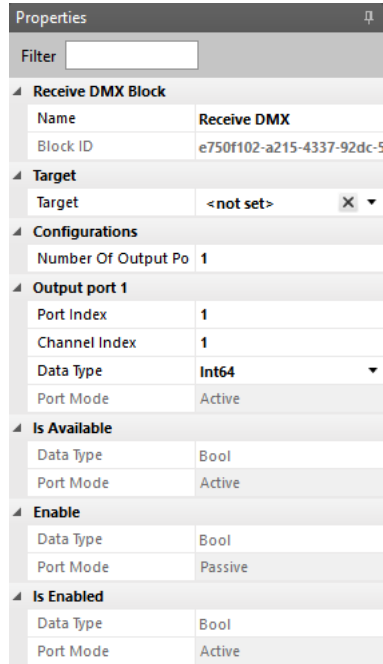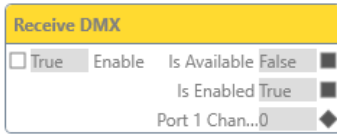
Use this Block to receive DMX values from an external system, like a SYMPL dmx Node, and process them in the Workflow Designer, e.g. to map a DMX channel value to a Sequence Opacity.

This Block is target based. The Target Value is displayed as main setting in the Block's header and its type defines the Block title.
When orphan, the Block has all the properties of a bound Block available. The Block has an Enable PortIn, an Is Available PortOut, an Is Enabled PortOut, and initially one PortOut for the received DMX value.
The Block can be bound to a SYMPL dmx Node or a SYMPL e:pix Node.

— Add a (virtual) SYMPL dmx or e:pix Node to the project. In the Setup Tab, set at least one DMX "Direction" property of the SYMPL node to "Input". The SYMPL node is now available as a target value for the DMX Input block.
— Define the Target.
— Define the number Output Port(s).

Category	Property	Description
Target		Define the target to bind the Block to: a SYMPL dmx Node or a SYMPL e:pix Node.
Number of Output Ports		Define the number of PortOuts to be available at the block. Each DMX channel of each DMX input universe has its own PortOut assigned. Define the configuration and mapping in the dedicated "Output port" properties.
Output port (e.g. 1)		
	Port Index	Define the DMX Port (universe) of the Target Value to listen to.
	Channel Index	Define the DMX channel of the selected universe to listen to. The Output Ports are per default consecutively mapped to DMX channels.
	Data Type	Define the data type of the PortOut. The DMX value can either be forwarded as Int64 or as Double.
Is Available		An actively sent bool value whether the bound target is available or not. For SYMPL nodes, the condition is automatically TRUE when a real Node is matched.
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.

— You can completely preconfigure the Receive DMX block without an attached SYMPL node in the project.

12.3.62   LCE2 Display
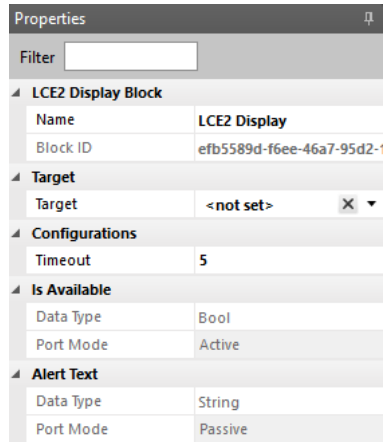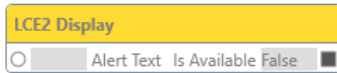
Use this Block to show a specific message on the display of the e:cue Lighting Control Engine 2 device, or LCE3.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has a PortIn for the Alert Text on the LCE2 display and an Is Available PortOut.
The Block can be bound to an LCE2 or an LCE3.

— Define the Target Property.
— Define the Timeout.

Category	Property	Description
Target Value	Value	Define the target to bind the Block to a LCE2 or an LCE3.
Timeout		Define the timeout of the alert in seconds. After the timeout, the LCE2 display returns to display its system pages

Is Available		An actively sent bool value whether the bound target is available or not.
Alert Text		A passively received string value of the alert text to be displayed on the LCE2.

— While an alert text is displayed on the LCE2, the OK button of the cursor control keys on the LCE2 is flashing. Press the OK button to dismiss the alert text before the timeout as an act of acknowledgement.

— To display alert text over more than one row / display line, use the Combine Text block. Put each curly bracket / PortIn in a new line in the Format String property of the Combine Text block and connect the alert text as parts to the PortIns.

— Please note that the LCE2 display supports the standard characters from ASCII 32 to ASCII 126 (Hex 20 to Hex 7E).

12.3.63   Glass Touch T6
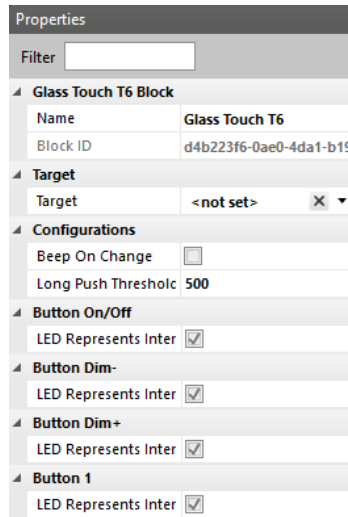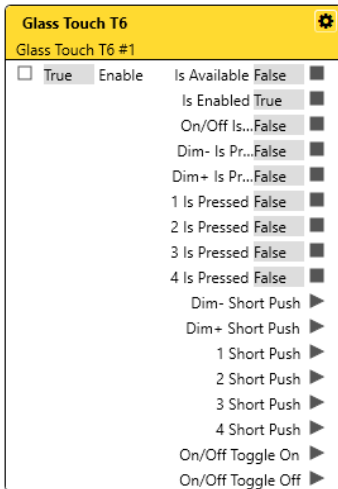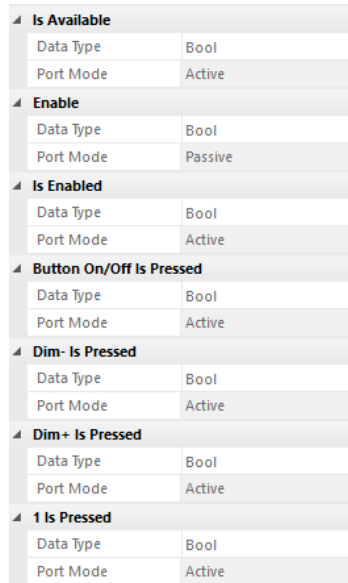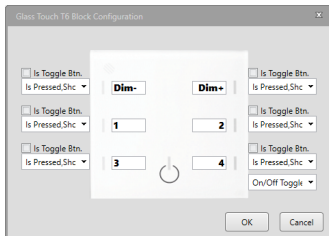
Use this Block to control your installation with a Glass Touch T6 User Terminal.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has bool PortOuts and EventOuts for each button as defined in the extended configuration window. Extended configuration is accessible at the cogwheel icon.
The Block can be bound to a Glass Touch T6.

— Define the Target.
— Define the available functionalities as PortOuts of each button in the extended configuration window.

Category	Description
Target	Define the target to bind the Block to a Glass Touch T6.
Beep on Change	Enable to get an acoustical feedback of the Glass Touch on every event. The Glass Touch will beep on pushing the button and on releasing the button.
Button (e.g. On / Off, Dim+/-, 1)	Enable to display the internal state of the button with the button's LED. The LED is on when the button is being pressed or is toggled on. The LED is off when the button is not being pressed or is toggled off. Disable to create a passive Int64 PortIn named "LED (e.g. On / Off, Dim+/-, 1)". This PortIn controls the LED of the button: 0: off 1: on 3: flashing.
Is Available	An actively sent bool value whether the bound target is available or not.
Enable	A passively received bool value to dis- or enable the Block.
Is Enabled	An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.

Button Is Pressed (e.g. On / Off, Dim+/-, 1)	An actively sent bool value whether the button is being pressed or not.

Defining the available functionalities of each button in the extended configuration window

Define the available functionalities of each button in the extended configuration window:

1. Click the cogwheel icon at the right top corner of the Block to open the extended configuration window: ⚙ .

Configuration of each button

Button names

Ok to apply changes

Cancel to discard changes

2. Configure each button. Select which functionalities are available:
First, define whether to handle the button as a toggle button. Check the box "Is Toggle Button" at the top of the drop-down menu to enable as toggle button.
Second, enable functionalities in the drop-down menu. Each functionality will have a separate PortOut in the Glass Touch block.

Is NOT Toggle Button:

Button not as toggle button defined

Click to open drop-down menu

Enabled functionalities and their PortOuts

Is Pressed: The boolean TRUE is forwarded for the duration of the button pressing.
Short Push: A trigger is sent when the button is pressed for a short time.

Long Push Start: A trigger is sent at the beginning of a longer button press.
Long Push Stop: A trigger is sent at the end of a longer button press.
Double Press: A trigger is sent when the button is pressed shortly twice.

Is Toggle Button:

Is Pressed: The boolean TRUE is forwarded for the duration of the button pressing.
Toggle On: A trigger is sent when the button is pressed to switch to the state "Toggle On". If "LED Represents Internal State" is enabled for the Button On / Off, the LED is on at the Toggle On state.
Toggle Off: A trigger is sent when the button is pressed to switch to the state "Toggle Off". If "LED Represents Internal State" is enabled for the Button On / Off, the LED is off at the Toggle Off state.

3. Click the "OK" button to save the configuration and close the window.

The Glass Touch T6 block has now the functionalities of the buttons configured and the PortOuts accordingly available.

— Please do not use the Glass Touch button as a toggle button if more than one button is connected to the same functionality. The Glass Touch toggle button has no back channel to remotely change its state.
— The On / Off Toggle button does not represent a power switch. The Power button of the Glass Touch does not need to be pressed on for the other buttons to be executed.
— You can rename buttons in the extended configuration window. Note that the customized name is used for the PortOut names only, not at the Properties window.
— You can completely preconfigure the Glass Touch block without an attached Glass Touch user terminal in the project and in Trial Mode.
— Please note that a Glass Touch is only responsive (LED indication, sound) after adding its target based Workflow block to the Workspace.

12.3.64   Glass Touch T6R

Use this Block to control your installation with a Glass Touch T6R User Terminal.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has bool PortOuts and EventOuts for each button as defined in the extended configuration window. Extended configuration is accessible at the cogwheel icon.
The Block can be bound to a Glass Touch T6R.

— Define the Target.
— Define the available functionalities as PortOuts of each button in the extended configuration window.

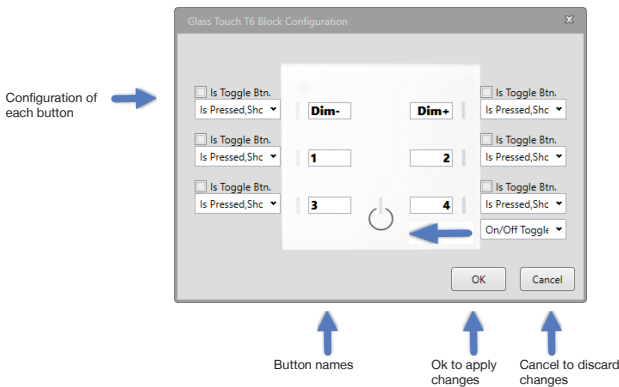| Category	Property	Description
Target		Define the target to bind the Block to a Glass Touch T6R.
Beep on Change		Enable to get an acoustical feedback of the Glass Touch on every event. The Glass Touch will beep on pushing the button and on releasing the button.

Jog Increment Abs		An actively sent double value of the jog position movement. The Jog Increment Abs value is the absolute increment of the jog position of all motions, from the very first motion to the last position of the latest motion. Clockwise motion adds to the value, counterclockwise motion subtracts from the value. One turn equals the value 100.
	Enable Increment Offset	Enable to create a PortIn which will be taken as start value for Jog Increment Abs.
	Min Value	Define the minimum value the Jog Increment Abs can have. E.g. when -300, three full turns counterclockwise on the jog wheel are needed to reach the minimum value starting from zero.
	Max Value	Define the maximum value the Jog Increment Abs can have. E.g. when 300, three full turns clockwise on the jog wheel are needed to reach the maximum value starting from zero.
Button (e.g. On / Off, 1)	LED Represents Internal State	Enable to display the state of the button with the LED of the button. The LED is on when the button is being pressed or is toggled on. The LED is off when the button is not being pressed or is toggled off. Disable to create a passive Int64 PortIn named "LED (e.g. On / Off, 1)". This PortIn controls the LED of the button: 0: off 1: on 3: flashing.

Is Available		An actively sent bool value whether the bound target is available or not.
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.
Button Is Pressed (e.g. On / Off, Play / Pause, 1)		An actively sent bool value whether the button is being pressed or not.
Jog Position		An actively sent double value of the place where the jog wheel is touched. The Jog Position value is "0" at the 12 o'clock position and increases clockwise up to a maximum value of "100" at 360°.
Jog Increment Rel		An actively sent double value of the jog position movement. The Jog Increment Rel value is the difference of the jog position from one motion, from the starting point to the end point of a motion.

Defining the available functionalities of each button in the extended configuration window

Define the available functionalities of each button in the extended configuration window:

1. Click the cogwheel icon at the right top corner of the Block to open the extended configuration window: ⚙ .

Jog wheel

Configuration of each button

Button names
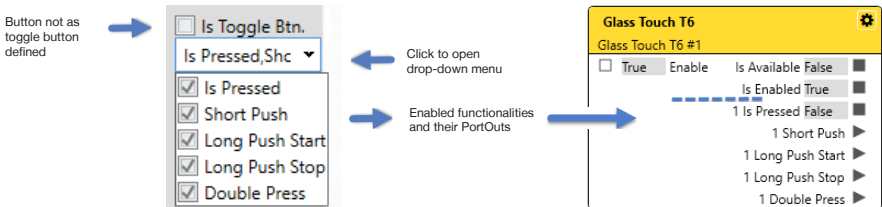
Ok to apply changes

Cancel to discard changes

2. Configure each button. Select which functionalities are available:
   First, define whether to handle the button as a toggle button. Check the box "Is Toggle Button" at the top of the drop-down menu to enable as toggle button.
   Second, enable functionalities in the drop-down menu. Each functionality will have a separate PortOut in the Glass Touch block.

   Is NOT Toggle Button:

Button not as toggle button defined

Click to open drop-down menu

Enabled functionalities and their PortOuts

Is Pressed: The boolean TRUE is forwarded for the duration of the button pressing.
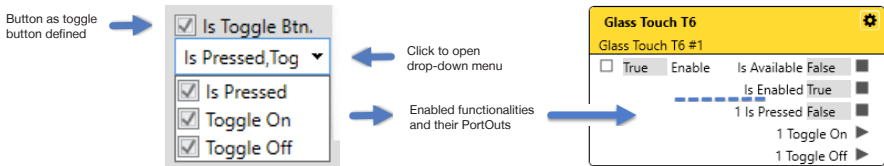**Short Push**: A trigger is sent when the button is pressed for a short time.

Long Push Start: A trigger is sent at the beginning of a longer button press.
Long Push Stop: A trigger is sent at the end of a longer button press.
Double Press: A trigger is sent when the button is pressed shortly twice.

Is Toggle Button:

Button as toggle button defined

Click to open drop-down menu

Enabled functionalities and their PortOuts

Is Pressed: The boolean TRUE is forwarded for the duration of the button pressing.
Toggle On: A trigger is sent when the button is pressed to switch to the state "Toggle On". If "LED Represents Internal State" is enabled for the Button On / Off, the LED is on at the Toggle On state.
Toggle Off: A trigger is sent when the button is pressed to switch to the state "Toggle Off". If "LED Represents Internal State" is enabled for the Button On / Off, the LED is off at the Toggle Off state.

3. Click the "OK" button to save the configuration and close the window.

The Glass Touch T6R block has now the functionalities of the buttons configured and the PortOuts accordingly available.

— The On / Off Toggle button does not represent a power switch. The Power button of the Glass Touch does not need to be pressed on for the other buttons to be executed.
— You can rename a button in the extended configuration window. Note that the customized name is used for the PortOut names, not at the Properties window.
— Use the jog wheel PortOuts for dimming purposes, e.g. connect an ActionPad fader block or a DALI - Group Control block. Use it also to control Key Framings of Effects by connecting a Change Object Property block. Use the Jog Increment Abs PortOut for dimming purposes with more fine-tuning: increase the limits of the Min and Max value to enable fine adjustment when calibrating. It is not possible to control hue. Use the Jog Push PortOut for triggers that are sent upon a short push on the jog wheel.
— You can completely preconfigure the Glass Touch block without an attached Glass Touch user terminal in the project and in Trial Mode.
— Please note that a Glass Touch is only responsive (LED indication, sound) after adding its Workflow block to the Workspace ().

Implementing classic jog wheel behavior

— To avoid a jump from 100% to 0% in opacity when crossing the 12 o'clock position on the jog wheel, use the Jog Increment Abs PortOut.

— If you want to use several Glass Touches T6R to control the same sequence, you need to pass the jog value through to all Glass Touches T6R. Enable the "Enable Increment Offset" of the blocks and connect the opacity of the sequence to the Increment Offset PortIn of the Glass Touch blocks. The offset will be considered by the Jog Increment Abs PortOut.

— To get the classic behavior of the jog wheel, use the following workflow:
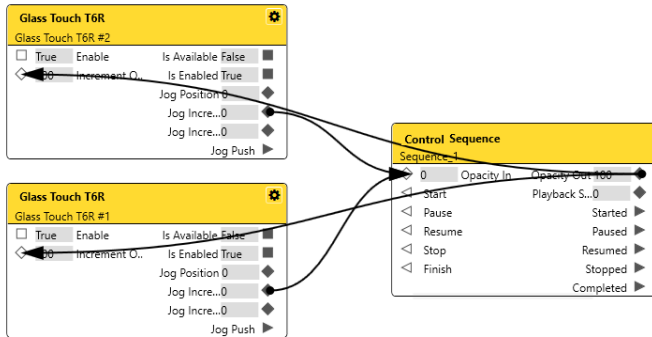
12.3.65   Glass Touch T12

The Glass Touch T12 based Block is identical to the Glass Touch T6 based Block, but with 12 buttons and one Button On / Off.
See „12.3.63 Glass Touch T6" on page 270 for details on the Block.
The Block can be bound to a Glass Touch T12.

12.3.66   Glass Touch IR
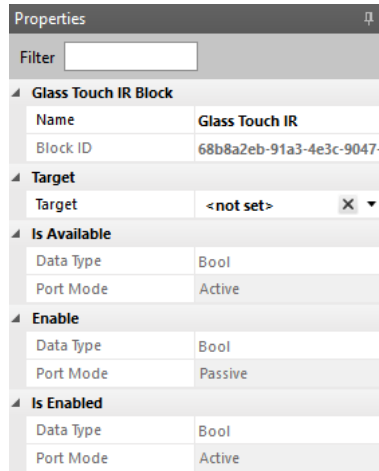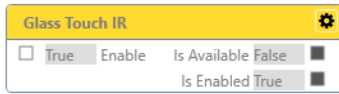
Use this Block to process IR commands from a remote control that are received by any Glass Touch User Terminal. Make sure the IR-Remote supports RC5.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has all the properties of a bound Block available.
The Block has bool PortOuts and EventOuts for each IR command as defined in the extended configuration window. Extended configuration is accessible at the cogwheel icon.
The Block can be bound to a Glass Touch T6, Glass Touch T6R and Glass Touch T12.

— Define the Target.
— Define the available EventOuts of IR commands in the extended configuration window.

Category	Description
Target	Define the target to bind the Block to a Glass Touch.
Is Available	An actively sent bool value whether the bound target is available or not.
Enable	A passively received bool value to dis- or enable the Block.
Is Enabled	An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.

Defining the available EventOuts of IR commands in the extended configuration window

Define the available EventOuts of IR commands in the extended configuration window:

1. Click the cogwheel icon at the right top corner of the Block to open the extended configuration window: ⚙ .
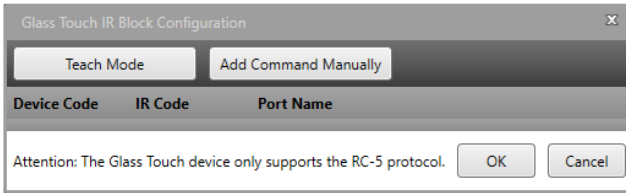
The configuration of an IR command consists of three elements:

Device Code: A selectable identification number for the remote control device which sends the IR command. A device with a device code "0" is not being progressed.

IR Code: The IR code sent by the remote control.

Port Name: Define a name for the EventOut that will be created for the IR command.

2. Add an IR command using either the Teach Mode or the manually adding option.

Teach Mode: Configure IR commands using remote controls.

Click the "Teach Mode" button. When activated, the button is highlighted:
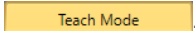
Point the remote control towards the Glass Touch while pressing the button on the remote control you want to configure. When the IR frequency is received by the Glass Touch, enter a name for the EventOut in the pop up entry field.

Click "OK" to add the IR command to the configuration. A new line is added to the extended configuration window that consists of the automatically entered device code, IR code and the user defined Port name.

To add further IR commands, press another button on the remote control.

Add Command Manually: Configure IR commands without using remote controls. You need to know the IR codes to be able to use this option.

Click the "Add Command Manually" button to add a new line to the extended configuration window:
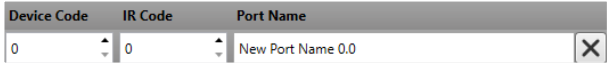
Define the three elements of the IR command.

To add further IR commands, click the button again.

3. Click the "OK" button to save the configuration and close the window.

The Glass Touch IR block has now the IR commands configured and the EventOuts accordingly available.

— You can preconfigure the Glass Touch block for IR commands while in Trial Mode and without an attached Glass Touch user terminal in the project as long as you configure the commands manually.
— You can manually change and rename any IR command.
— To delete an IR command, click the cross button at the end of the line: .
— To configure identical IR commands for multiple Glass Touch IR blocks, first configure one target based Block. Secondly, delete the target value of the Block. Copy and paste the now orphan Block. Set the target values for the copied Blocks as desired. All configured IR commands are multiplied and available.

Example: Two IR commands for pressing button 1 and the Power button on the remote control are defined using Teach Mode in the extended configuration window. Two EventOuts for the IR commands are created in the Glass Touch IR block:
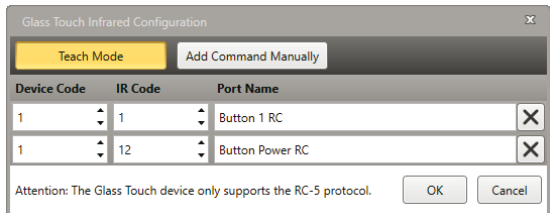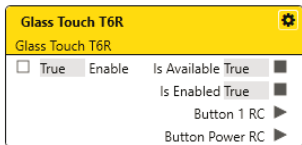
Implementing classic dimming behavior using remote control
— To dim a sequence using a remote control, create the two IR commands Up and Down.
— Connect an Add Numbers block to the Up EventOut, setting the Block to "Is event based" and Input 2 to Property with a value of 10. Using values lower than ten slows

down dimming, greater than ten makes dimming faster (depending on the repeat rate of the used IR control).

— Connect the Result to the Opacity In PortIn of the Sequence Play Control block and the Opacity Out PortOut of the sequence to the Input 1 PortIn of the Add Numbers block.

— Connect a Subtract Numbers block to the Down EventOut and repeat all settings from the Add Numbers block.
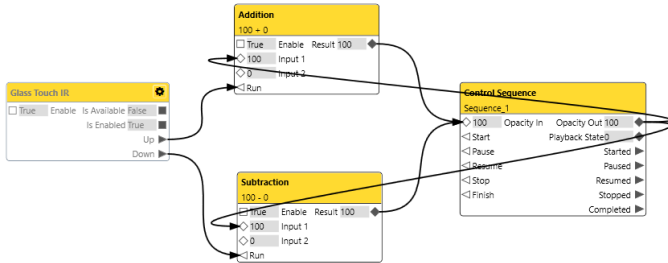
12.3.67   Send E-Mail
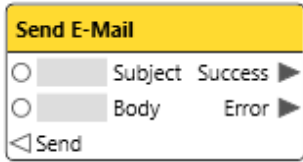
Use this Block if you want to send emails with customizable messages.

The Block has two PortIns for the content of the email, is event triggered by default, and has two EventOuts to give feedback on its execution.

— Define an SMTP server data (Host, Encryption, and Port).
— Enter sender address (From, Password) and recipient address (To). Optionally, add recipients in CC and BCC.
— Define the content of the Subject and the Body of the emails.

Category	Property	Description
Is event based		Enabled. This Block is executed on event trigger.

Mail Settings		Define the mail settings.
	Encryption	Choose between None, SSL, STARTTLS according to your email provider specifications. Required.
	Host	Enter the SMPT server address according to your email provider specifications. Required.
	Port	Enter the SMPT port according to your email provider specifications. Required.
	Password	Enter the password of your email account. Required.
	From	Enter your email address. Required.
	To	Enter the recipient address. Required.
	CC	Optional.
	BCC	Optional.
Subject		Configure how the content for the subject of the email message is constituted. By default, the content is actively caught from the counterpart Block. Set the Input Type to Property to manually define a static value.

Body		Configure how the content of the body of the email message is constituted. By default, the content is actively caught from the counterpart Block. Set the Input Type to Property to manually define a static value.

12.3.68   Send UDP

Use this Block to send UDP network triggers to a given IP address of external systems, e.g. with a Constant Value Block.

The Block has one PortIn for the message to be sent, an Enable PortIn, an Is Enabled PortOut, and an EventOut for feedback that the message was sent.

— Define the UDP Settings.

Category	Property	Description
UDP Settings		Define the receiver / destination and the encoding of the UDP packets.
	Encoding	The encoding how to interpret the Packet Content. Choose between ANSI, ASCII, UTF8, and Binary. Using Binary, you can combine hex chars, decimal chars and text for input, output and filtering in any way: 13 10 "Hello" 0x0D 0x0A $D $A means \r\nHello\r\n\r\n. E.g. specify each character one by one and combine it with ANSI text in quotes. Use \" to have a quote sign inside the quotes.
	IP Address	The destination IP Address of the receiver of the UDP packets to send to. Default: 127.0.0.1 which is the IP address of the localhost (= the computer running SYMPHOLIGHT); the computer itself receives the sent message.
	Port	The destination Port of the receiver that receives the UDP packets sent from the SYMPHOLIGHT instance (from 0 to 65535).
	Sender Port	If -1, a random port is chosen from which SYMPHOLIGHT is sending. If not -1, define a port from 0 to 65535 to ensure that the peer receives the packet from that port.
Enable		A passively received bool value to dis- or enable the Block.

Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.
Packet Content		The content of the UDP packet, a string value, is actively forwarded to the counterpart Block.

— For mutual, bidirectional communication with an external system, use the Send UDP and the Receive UDP blocks. The IP Addresses in both UDP Settings have to be identical.

— Please make sure to avoid well known ports. Only use unassigned UDP ports.

12.3.69   Receive UDP

Use this Block to receive external UDP network triggers that can be processed in other blocks, e.g. the Value to Event / State Block. Creates an event and provides the content of a received UDP packet.

The Block has an Enable PortIn, one PortOut for the content, one EventOut and an Is Enabled PortOut. The UDP Settings are displayed as main setting in the Block's header.

— Define the UDP settings.

Category	Property	Description
Packet Filter		Define if only UDP packets with a specified value are to be received. When empty, all UDP packets are received.
UDP Settings		Define the sender and the encoding of the UDP packets.
	Encoding	The encoding how to interpret the Packet Content. Choose between ANSI, ASCII, UTF8, and Binary. Using Binary, you can combine hex chars, decimal chars and text for input, output and filtering in any way: 13 10 "Hello" 0x0D 0x0A $D $A means \r\nHello\r\n\r\n. E.g. specify each character one by one and combine it with ANSI text in quotes. Use \" to have a quote sign inside the quotes.
	IP Address	The IP Address of the sender from which it is allowed to send UDP packets to the SYMPHOLIGHT instance. Use 0.0.0.0. if all senders are allowed to send UDP packets.
	Port	The Port of the receiver / destination (= SYMPHOLIGHT) of the UDP packets to bind to (from 0 to 65535).
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.

Packet Content		The content of the UDP packet, a string value, is actively received from the network (e.g. a Send UDP block).

— For mutual, bidirectional communication with an external system, use the Send UDP and the Receive UDP blocks. The IP Addresses in both UDP Settings have to be identical.

— Please make sure to avoid well known ports. Only use unassigned UDP ports.

12.3.70   Send & Receive Serial
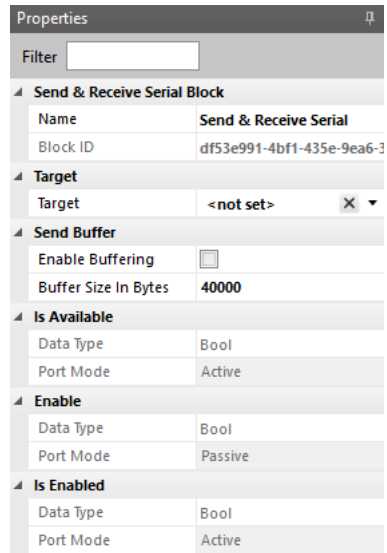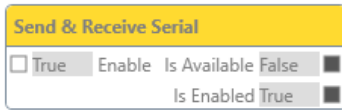
Use this Block to communicate with external systems using serial ports, like the SYMPL serial Node or a Local Serial Port. Use this Block to send data to the serial ports and to receive data from the serial ports.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has an Enable PortIn, an Is Available PortOut and an Is Enabled PortOut.
The Block can be bound to a Local Serial Port with RS232 interface (see „Local Serial Port based:" on page 297) or a SYMPL serial Node with RS232 and RS485 interfaces (see „SYMPL serial Node based:" on page 299).

— Define the Target.

Category	Property	Description
Target		Define the target serial port to bind the Block to a SYMPL serial Node or a Local Serial Port.

Send Buffer		Define whether to buffer data that is to be sent to the target serial port.
	Enable Buffering	Enable to buffer the data of the project that is to be sent to the target serial port if the target is currently unavailable. An additional EventIn named "Clear Send Buffer" is created. **Clear Send Buffer:** When triggered, clears the buffer of its value data.
	Buffer Size in Bytes	Only for enabled "Enable Buffering": Define the size of the send buffer in bytes.
Is Available		An actively sent bool value whether the bound target is available or not. For a Local Serial Port, configure Serial Settings properties in the Setup Tab. For SYMPL Nodes, the condition is automatically TRUE when a real Node is matched.
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.

— For each target serial port, the Serial Settings properties in the Setup Tab need to be configured. Please define the Baudrate, DataBits, Parity, and Stop Bits properties in accordance to the serial ports. Conformity of the sender and the receiver ports is essential.

Local Serial Port based:
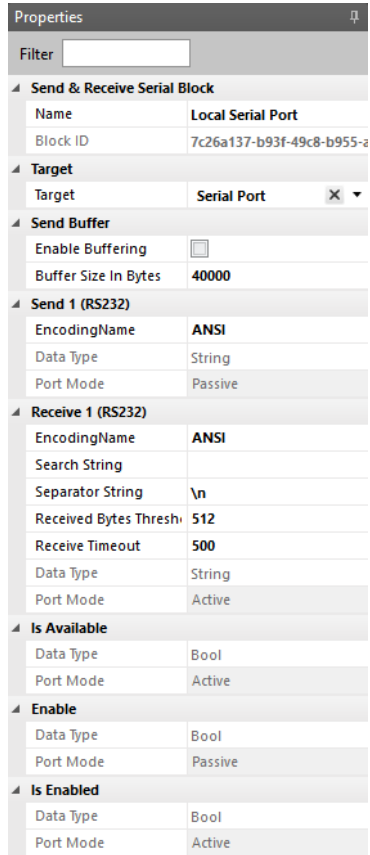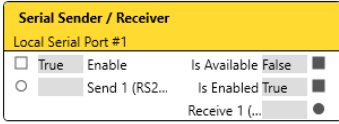
In addition to the properties of an orphan Send & Receive Serial block, the Local Serial Port based Block has one PortIn for the content to be sent to the target serial port and one PortOut for the received message.

In addition to the properties of an orphan Send & Receive Serial block:
— Define the encoding of the outgoing messages (Send 1 (RS232)).
— Define the settings for the received PortOut messages (Receiver 1 (RS232)).

Category	Property	Description
Send 1 (RS232)		A passively received string value that is sent to the target serial port (RS232 interface).
	Encoding Name	Select an encoding scheme for the outgoing string value: ANSI, ASCII, UTF8, or Binary.
Receive 1 (RS232)		An actively sent string value that was received from the target serial port (RS232 interface).
	Encoding Name	Select an encoding scheme for the received string value: ANSI, ASCII, UTF8, or Binary. Use the encoding scheme of the target serial port where specified.
	Search String	Define the string value that defines whether a message is forwarded or not. Only messages that contain the search string (multiple entries allowed) are processed.
	Separator String	Define a string value. When the specified string has been received, the message will be processed. Not part of the forwarded content. Programming language, e.g. line feed "\n", is not supported.
	Received Bytes Threshold	Define a number in bytes. When the specified number has been received, the message will be processed. Largest possible received number should be encompassed as minimum size.
	Receive Timeout	Define a time in milliseconds (ms). When the specified period has elapsed, the message will be processed. The time metering starts on the first received value. A message is only created when any value has been received in the specified time.

— A message is processed when the condition of the separator string OR of the bytes threshold OR of the timeout is met. The search string does not have any influence on the length of a message.
— The threshold and the timeout start over after a message is processed.
— Please take the serial port specifications into account when configuring the message processing scheme in SYMPHOLIGHT.
— The encoding scheme can be predetermined by the target serial port. Select this encoding scheme in the "Encoding Name" properties.
— The string value can be predetermined by the target serial port.
— Please note that SYMPHOLIGHT has an internal clock generator of 100 ms. A receive timeout might deviate for about 100 ms or for about 5 characters.
— To set the "Is Available" property to TRUE, go to the Setup Tab properties of the Local Serial Port controller. Enable the "Is Active" property and define a "Serial Port".
— For the Encoding Name properties: choose between ANSI, ASCII, UTF8, and Binary. Using Binary, you can combine hex chars, decimal chars and text for input, output and filtering in any way:
13 10 "Hello" 0x0D 0x0A $D $A means \r\nHello\r\n\r\n.
E.g. specify each character one by one and combine it with ANSI text in quotes. Use \" to have a quote sign inside the quotes.

SYMPL serial Node based:

The SYMPL serial Node based Block is identical to the Local Serial Port based Block, but with an additional RS485 PortIn and an additional RS485 PortOut.

In addition to the properties of an orphan Send & Receive Serial block:
— Define the encoding of the outgoing messages (Send 1 (RS232) and Send 2 (RS485)). See the Local Serial Port based RS232 properties for Send 1 details. They also apply for the Send 2 (RS485) interface.
— Define the settings for the received PortOut messages (Receiver 1 (RS232) and Receiver 2 (RS485)). See the Local Serial Port based RS232 properties for Receive 1 details. They also apply for the Receive 2 (RS485) interface.

— The "Is Available" property is TRUE when the target serial port is matched to a real SYMPL serial Node.

12.3.71 Write Logbook
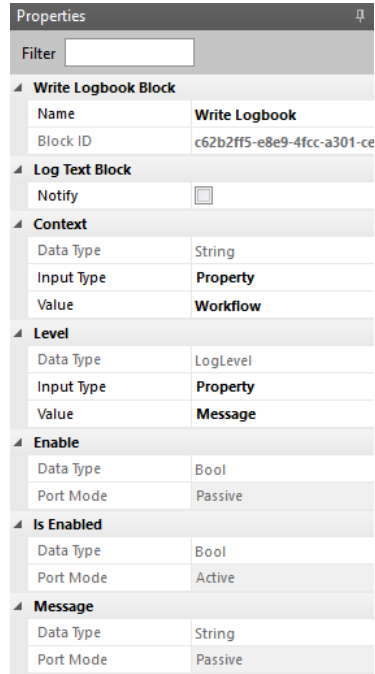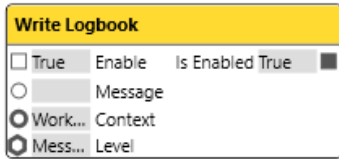
Use this Block to write a logbook message to the Log Tab, e.g. to check if the triggers work.

The Block has three PortIns for the content of the message, an Enable PortIn, and an Is Enabled PortOut.

— Define the message, the context and level.

Category	Description
Log Text Block	Enable to additionally display the message in a toast box.
Context	Define with which context the message is entered in the logbook (third column). If the Input type is set to Port, a Link at the PortIn Context is required for the Block to execute.

Level	Define with which level the message is entered in the logbook (first column). The visibility of the according level has to be enabled in the Log Tab for the messages to be displayed. Define the Input Type. Set as Property, choose a Value: **Debug:** Use for Support only. The message will not be visible in the Log Tab, only written in the log file. Text Block notifications are not displayed. **Message:** Uses the info icon . **Warning:** Uses the Warning icon . **Error:** Uses the Error icon .
Enable	A passively received bool value to dis- or enable the Block.
Is Enabled	An actively sent bool value whether the Block is enabled or not.
Message	The content of the logbook message, a string value, is passively received from the counterpart Block.

12.3.72   Send to Network

Use this Block to send data from configured ports to the network.

The Block has initially two PortIns for the target network, an Enable PortIn, and an Is Enabled PortOut. Extended configuration is accessible at the cogwheel icon.

— Define Target IP Address(es).
— Define the Port.
— You can configure additional PortIns for values and EventIns (uncheck "Contains Data"), group them, and import and export CSV sheets in the extended cogwheel window.

Category	Property	Description
Configurations	Encryption Key	A key to encrypt the transferred data. Sender and receiver must use the same key for successful transfer. Leave empty to transfer the data unencrypted.
Target IP Address(es)		Define the target IP address(es) to communicate with.
	Input Type	Define whether the Target IP address(es) value is a fixed bool value (Input Type "Property") or a received bool Input value (Input Type "Port").
	Value	Set one or more (comma-separated without space) IP addresses here to unicast to these addresses. Leave empty to use broadcast.
Port		
	Input Type	Define whether the port value is a fixed bool value (Input Type "Property") or a received bool Input value (Input Type "Port").
	Value	Define the port to send to. The port value must be the same port value as configured at the receiver.
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.

12.3.73   Receive from Network

Use this Block to receive data from the network and dispatch it to configured ports.

The Block has initially one PortIn for the target network, an Enable PortIn, and an Is Enabled PortOut. Extended configuration is accessible at the cogwheel icon.

— Define the Port.
— You can configure additional PortOuts for values and EventOuts (uncheck "Contains Data"), group them, and import and export CSV sheets in the extended cogwheel window.

Category	Property	Description
Configurations	Encryption Key	A key to encrypt the transferred data. Sender and receiver must use the same key for successful transfer. Leave empty to transfer the data unencrypted.
Port		
	Input Type	Define whether the port value is a fixed bool value (Input Type "Property") or a received bool Input value (Input Type "Port").
	Value	Define the port to receive from. The port value must be the same port value as configured at the sender.
Enable		A passively received bool value to dis- or enable the Block.
Is Enabled		An actively sent bool value whether the Block is enabled or not at the "Enable" PortIn.

12.3.74   Script
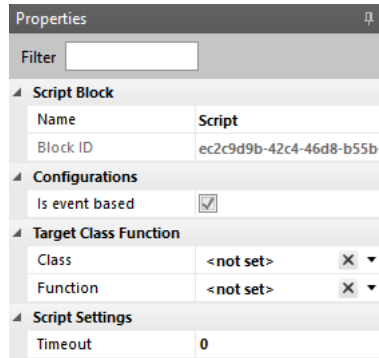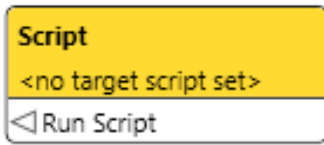
Use this Block to add Scripts to the Workflow Designer, e.g. if you want to start Scripts based on Action Pad button presses.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has no Ports. After defining the Target Settings, the Block has Ports as defined in the script.
The Block can be bound to a Script from the Scripting tab.

— Define the Target.

Category	Property	Description
Target		Define the target to bind the Block to a script.
Is event based		Enabled. This Block is executed on event trigger.
Script Settings	Timeout	Define the maximum run time of the script for one execution after triggering in milliseconds. 0 means disabled.

— Define further properties as designed in each script.

12.3.75   Set Property
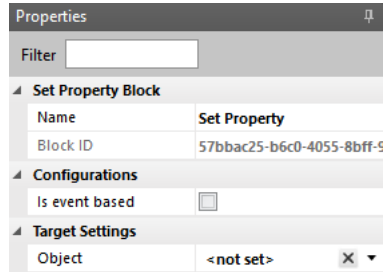
Set Property

Use this Block if you want to change the value of an object property. An object can for example be an opacity value of an effect or the master fader.
Example: assign a fader on an Action Pad page to the Grand Master Fader (Target Object: ProjectInformation, Target Property: Opacity).

This Block is target based. The Target Value is displayed as main setting in the Block's header.
After defining the Target Settings, the Block has one PortIn for the new value to which the defined property of the object is to be changed to.
The Block can be bound to a Timeline element (Content), an Action Pad element or a general property.

— Define the target (Property "Object").
— Define the Target Property.
— Define the new value.

In addition, select whether the new value is a fixed value (Input Type "Property") or a received Input value (Input Type "Port"). When set to Input Type "Property", the Block gets the additional EventIn "Perform Change" so that the change can be applied.

— To be able to set property information of an element, the element needs a defined Identifier name for the Block to be able to bind to the element. Most elements automatically have a defined Identifier. Effect components do not have a defined Identifier. See the „Example: Getting Effect Opacity Value" on page 309 on how to bind Blocks to a property of an element with undefined Identifier.

12.3.76 Get Property
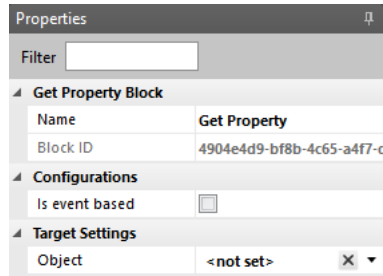
Use this Block if you want to get the value of an object. The Block gets a property from an object and provides it as value. An object can for example be an opacity value of an effect or the master fader.
Example: fetch the color selection from a Color Picker on an Action Pad page and provide the color value as ProPhotoColor value.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has no Ports. After defining the Target Settings, the Block has one PortOut for the defined property.
The Block can be bound to a Timeline element (Content), an Action Pad element or a general property.

— Define the target (Property "Object").
— Define the Target Property.

Category	Property	Description
Target Settings		Define the target to bind the Block to (object) and the property you want to get from the target.
	Object	Define the target to bind the Block to. Objects are all editable properties of the currently running SYMPHOLIGHT project.
	Target Property	Define the property of the selected target to receive. Depending on the selected type of object, the available list of properties varies.

Output Value		The data type and Port mode of the created PortOut, defined by SYMPHOLIGHT.

— To be able to get property information of an element, the element needs a defined Identifier name for the Block to be able to bind to the element. Most elements automatically have a defined Identifier. Effect components do not have a defined Identifier. See the example below on how to bind Blocks to a property of an element with undefined Identifier:

Example: Getting Effect Opacity Value

Get the opacity of an Effect, e.g. Circular Rainbow, by defining the Identifier of the Effect in the Content Tab first to make the Effect available as an Object. Now bind the Get Property block to the property of the Effect and you receive the current opacity value:

1. Go to the Content Tab and select the Effect of which you want to get the opacity value. In this example, select the Circular Rainbow effect in Sequence #3:
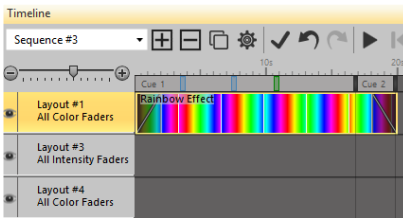
2. In the Properties window of the Effect, go to the Identifier property of the first Effect (Blended 2D Color Effect) and define the name / value. Here, the given Identifier value is RainbowEffectBasicEffect:
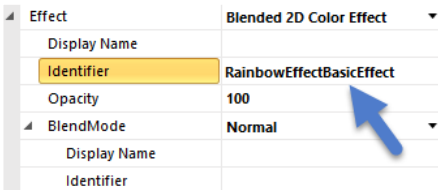
3. Go to the Workflow Designer, Automation Tab and add the Get Property block to the Workspace.

4. In the Properties window of the Block, select the RainbowEffectBasicEffect in the Object property:
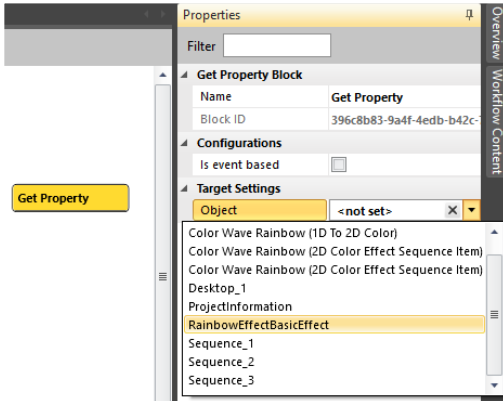
Now the Block is bound to this Content element and provides a list of available properties at Target Property:
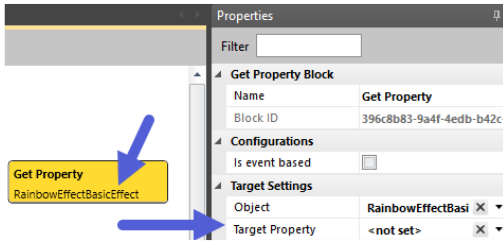
5. Select Opacity as Target Property. Now the Block has a PortOut that provides the current opacity value of the Rainbow Effect:
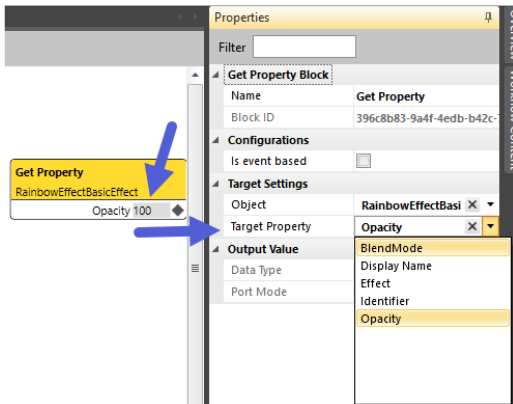
The Get Property block now provides the current opacity value of the Rainbow Effect. To change this value, use the Change Object Property block and to check the value by comparison with another value, use the Check Object Property block.

12.3.77 Check Property
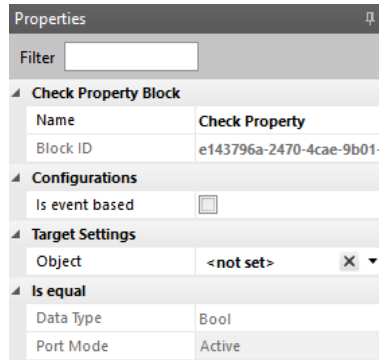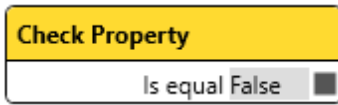
Use this Block if you want to compare the value of an object property. The Block compares a property of an object against a value. It provides a bool value as result from the comparison. An object can for example be an opacity value of an effect or the master fader.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
When orphan, the Block has no Ports. After defining the Target Settings, the Block has one PortIn for the value to which the defined property of the object is to be compared and one bool PortOut for the result of the comparison.
The Block can be bound to a Timeline element (Content), an Action Pad element or a general property.

— Define the target (Property "Object").
— Define the Target Property.
— Define the value to which the defined property of the object is to be compared (Category "Comparand").

In addition, select whether the comparand is a fixed value (Input Type "Property") or a received Input value (Input Type "Port").

Category	Property	Description
Is event based		Enable to execute this Block on event triggers.

— The Is condition met PortOut forwards the result of the comparison: TRUE when the condition of the comparand gets identical to the Target Property.

— To be able to get property information of an element, the element needs a defined Identifier name for the Block to be able to bind to the element. Most elements automatically have a defined Identifier. Effect components do not have a defined Identifier. See the „Example: Getting Effect Opacity Value" on page 309 on how to bind Blocks to a property of an element with undefined Identifier.

12.3.78   Call Method
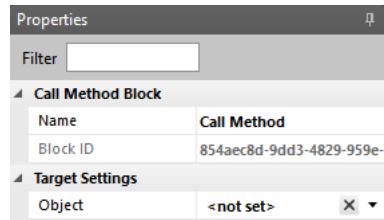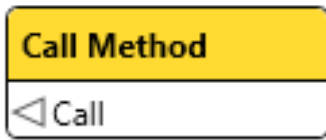
Use this Block to call a function of an object.
Example: Play/Pause/Stop of a Sequence, or Cue Control for the Sequence.

This Block is target based. The Target Value is displayed as main setting in the Block's header.
After defining the Target Settings, the Block has where applicable one PortIn for the value of the called method and one EventIn to trigger the method call.
The Block can be bound to a Sequence or to a Mutex Group.

— Define the target (Property "Object").
— Define the Target Method.
— Define the new parameter's PortIn value where applicable.

— Please note: To be able to call a method of an element, the element needs a defined Identifier name for the Block to be able to bind to the element. Most elements automatically have a defined Identifier. Effect components do not have a defined Identifier. See the on how to bind Blocks to a property of an element with undefined Identifier.

12.4   Scripting

Using Rest API and Scripting requires prior knowledge of these topics.

In the Scripting tab, you can create macros in the C# language. The macros are compiled and can be executed in the Workflow Designer. For example allows the Scripting RDM access, video import or sending custom DMX commands / values to fixtures, overwriting Content / Output Windows' values. In general, the full C# language scope can be used. There are some limitations for Scripts in SYMPHOLIGHT regarding the use and runtime behavior, compared to standalone C# applications in Windows:

— Do not use threading. You cannot control threads that are created by your Script.
— Do not create never ending program types (like for(;;)), as they may consume too much memory and there is no real control over your Script.

The best method to create Scripts in SYMPHOLIGHT is to use the concept of state machines.

A Script starts to execute and does only the necessary things. Then it stops until it is called again to do the next things. This minimizes the runtime for the Script and allows better control over your Script.
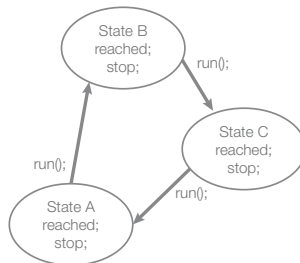
To create and edit Scripts, use the Scripting tab in the Automation Window.

To create a new script, click *New*.
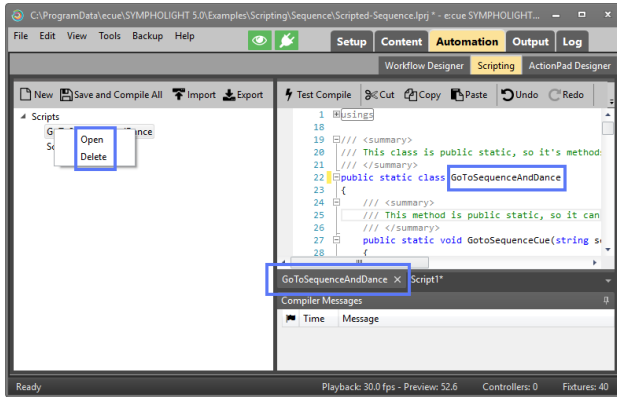
To edit a Script, double-click it in the list of Scripts, select *Open* in the context menu, or select its tab. The tabs for the opened scripts are at the bottom of the editor window, you can also close a script with the cross in the tab line.
You can also import Scripts and export Scripts. Click *Import* or *Export* in the tool bar.
Select *Delete* in the context menu to delete a script.

One Script may contain more than one class. You can also define classes in other Scripts and refer to them:
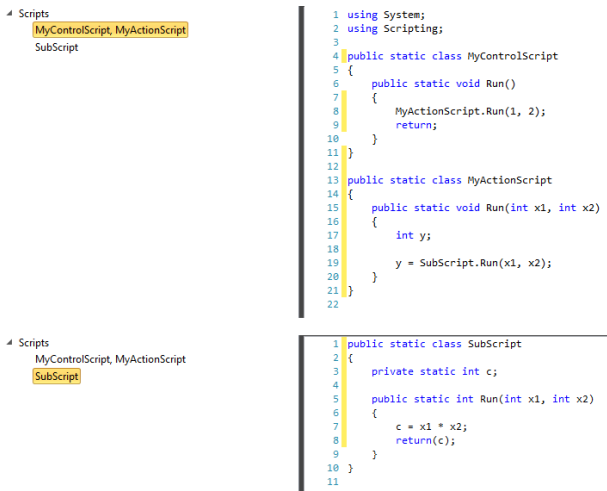
To test a Script for errors (syntax check of the script), click ⚡ _Test Compile_ in the top menu. In the Compiler Messages you find the debugging feedback. The Script is not applied to the project yet.
To compile, save, and apply a Script, click 💾 _Save and Compile All_. After a Script compiled successfully, the Script is available in the Workflow Designer, where the defined input and output parameters are used as PortIns and PortOuts.
At first, the name of the Script will be system-defined. Change the name of the class inside the Script to a new unique name. After a first 💾 _Save and Compile All_, the name will be changed in the list of Scripts also.

Access to the objects of SYMPHOLIGHT happens via interface classes. While you type, you will get an overview of objects, methods or devices that are available at this point. Press _Ctrl + Space_ to get a help overview.
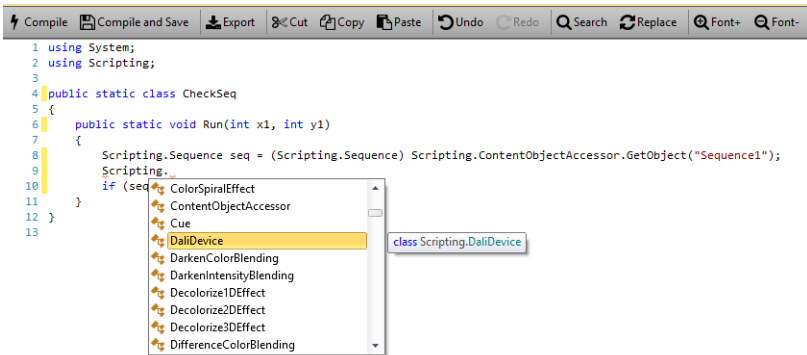
The Help Reference is available at F1 from within the editor or at any time at Scripting Help under Help in the main menu.
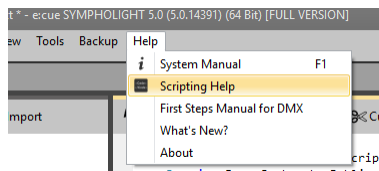
Run a script, and a method also, directly from the Workflow Designer by using the Script bound Scripting blocks.

> For older Scripting versions, simply add the using Ecue.Sympholight.Scripting.Api.

12.5   Examples for SYMPHOLIGHT Scripting

Example files of Scripting are stored at
　　C:\ProgramData\ecue\SYMPHOLIGHT 5.0\Examples\Scripting.
Check also the download area of the e:cue website.

> When you copy and paste the examples shown below into a source code editor, you need to replace the quotation marks.

12.5.1   Adding an entry in the SYMPHOLIGHT Logs

This example is the default code in every newly generated script.
Create a new script by clicking _New_ in the left subwindow of the Scripting Editor.
Rename the `public static class Script1` to `public static class DebugTest`.
This is the name with which this script will be available in the Workflow Editor later. Use the following code:

```
// using Scripting; // The deprecated scripting API necessary to ac-
cess ActionPad and Effects.
using Ecue.Sympholight.Scripting.Api;
using System;

/// <summary>
/// This class is public static, so it's methods can be accessed from
the workflow designer.
/// </summary>
public static class DebugTest
{
    /// <summary>
    /// This method is public static, so it can be accessed from the
workflow designer.
    /// </summary>
    public static void Run()
    {
        // Use "ecue." to access SYMPHOLIGHT scripting functionality
        ecue.Logging.Info("This is a test log message generated by
your script.");
    }
}
```

In the Workflow Editor, insert a Block of a Periodic Trigger and the Block for your script. Set the time interval of the Periodic Trigger to five seconds, connect the EventOut of the Trigger to the EventIn in your script.
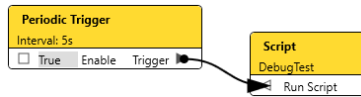
Head over to the SYMPHOLIGHT Logs now to see the Log entries.

12.5.2 Automation of an Effect
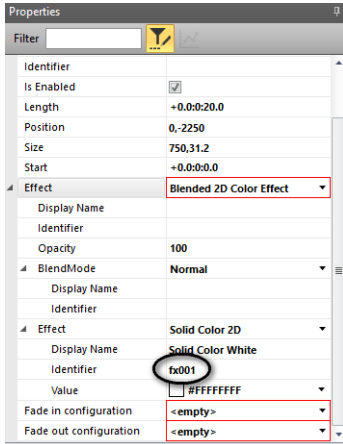
This script sets the color of an Effect to random values. The Content contains a 2D Color Effect with constant color, the Effect needs an identifier, here fx001. Using this ID, we can address the property of the Effect.

```
using System;
using Scripting;
using Ecue.Scripting.Compatibility;

public static class ColorFx
{
        public static void Run()
        {
                SolidColor2DEffect effect =
                        ContentObjectAccessor.GetObject("fx001")
                        as SolidColor2DEffect;
                if (effect != null) // if successful
                {
                        Random rgbval = new Random();
                        // RGBA, floats between 0 and 1
                        effect.Value = new ProPhotoColor(
                                (float)rgbval.NextDouble(),
                                (float)rgbval.NextDouble(),
                                (float)rgbval.NextDouble(),
                                1.0f); // -> alpha is constant
                }
        }
}
```

12.5.3   Handling UDP messages

This script receives a command via a UDP message and controls a Sequence. See comments for details.

```
using Ecue.Contracts;
using Ecue.Contracts.Workflow;
using Ecue.Contracts.UserAccess;
using Ecue.Contracts.Scheduler;
using Ecue.Contracts.Rdm;
using Ecue.Contracts.Logging;
using Ecue.Contracts.Dmx;
using Ecue.Contracts.Exceptions;
using Ecue.Contracts.Devices;
using Ecue.Contracts.Content;
using Ecue.Contracts.Content.Effects;
using Ecue.Sympholight.Scripting.Api;
using System;

public static class UDP_Handler
{
   // Evaluates strings received
   // Received strings must have the following structure:
   // Action-SequenceIdentifier,
   // where "Action" is a Sequence action (Play, Pause, Stop)
   // and SequenceIdentifier is a valid Sequence ID
   //
   // Further error handling (e.g., handling of wrong or invalid
   // parameters) is recommended but not part of this script sample
   public static string Run(String UDP_Content)
   {
      string retVal = "";
      if (UDP_Content == "")
      // check if submitted string is empty,
      // if empty a warning is returned.
      // if message is valid, further processing will be done
      {
         retVal = "An empty message has been received. Please
         check.";
      } else
      {
         string[] splittedString = UDP_Content.Split('-');
         // Split string at each "-", the string before "-"
         // is saved in splittedString[0] and the other string is
         // saved in splittedString[1]
         if (splittedString.Length < 2)
         {
            //without 2 Strings we cannot proceed
            retVal = "Could not split the string \"" + UDP_Content +
```

```
           "\" into two parts. Please check.";
        } else
        {
            if (splittedString.Length > 2)
            {
                // we can proceed, but something unexpected happened.
                retVal = "The string \"" + UDP_Content + "\" has been
                splitted into " +  splittedString.Length + " parts.
                \n\r" +
                "Proceeding with the script. Please check.";
            }
                var seq = ecue.Sequences.GetSequenceByIdentifier
                (splittedString[1]);
            if (seq == null)
            {
                // we have to check, if there is a Sequence with the
                passed name available
                retVal += "could not find Sequence \"" + splitted
                String[1] + "\". Please check.";
            }
            else
            {
                switch(splittedString[0])
                // check and perform the action that has been
                // submitted in the string
                {
                    case "Start":
                        seq.Play();
                        retVal += splittedString[1] + " started";
                        break;
                    case "Pause":
                        seq.Pause();
                        retVal += splittedString[1] + " paused";
                        break;
                    case "Stop":
                        seq.Stop();
                        retVal += splittedString[1] + " stopped";
                        break;
                    default:
                        retVal += "invalid string submitted";
                        break;
                }
            }
        }
    }
    return retVal;
  }
}
```

This is the configuration in the Workflow Designer:
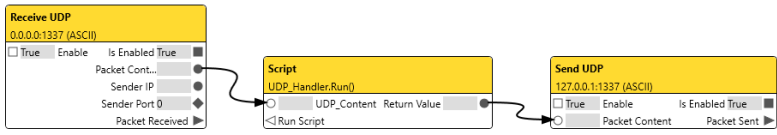
12.6   Supported function param types

The following list contains all supported function parameter types for the Scripting Run(...) method.

```
using System;
using Scripting;

/// <summary>
/// This script test all supported function param types for scripting Run(...) method
/// </summary>

public static class TestInOut
{
    public static bool Run(
                bool vBool,
                out bool ovBool,
        ref bool rvBool,

        byte vByte,
        out byte ovByte,
        ref byte rvByte,

                char vChar,
                out char ovChar,
                ref Char rvChar,

                DateTime vDateTime,
                out DateTime ovDateTime,
                ref DateTime rvDateTime,

                float vFloat,
                out float ovFloat,
                ref float rvFloat,

                Int32 vInt32,
                out Int32 ovInt32,
                ref Int32 rvInt32,

                Int64 vInt64,
                out Int64 ovInt64,
                ref Int64 rvInt64,
```

```
            double vDouble,
            out double ovDouble,
            ref double rvDouble,

            string vString,
            out string ovString,
            ref string rvString)
{
            vBool = !vBool;
            ovBool = vBool;
            rvBool = vBool;

            vByte++;
            ovByte = vByte;
            rvByte = vByte;

            vChar++;
            ovChar = vChar;
            rvChar = vChar;

            vDateTime = DateTime.Now;
            ovDateTime = vDateTime;
            rvDateTime = vDateTime;

            vFloat++;
            ovFloat = vFloat;
            rvFloat = vFloat;

            vInt32++;
            ovInt32 = vInt32;
            rvInt32 = vInt32;

            vInt64++;
            ovInt64 = vInt64;
            rvInt64 = vInt64;

            vDouble++;
            ovDouble = vDouble;
            rvDouble = vDouble;
```

```
            vString += "1";
            ovString = vString;
            rvString = vString;

            Logger.Info("TestInOut:"+vString);
            return true;
    }
}
```

12.7   The ActionPad Designer

With the ActionPad Designer, you can build a graphical user interface for SYMPHOLIGHT, called the Action Pad. You can connect to the SYMPHOLIGHT via the build-in HTTP server and control SYMPHOLIGHT over a LAN or WiFi connection.
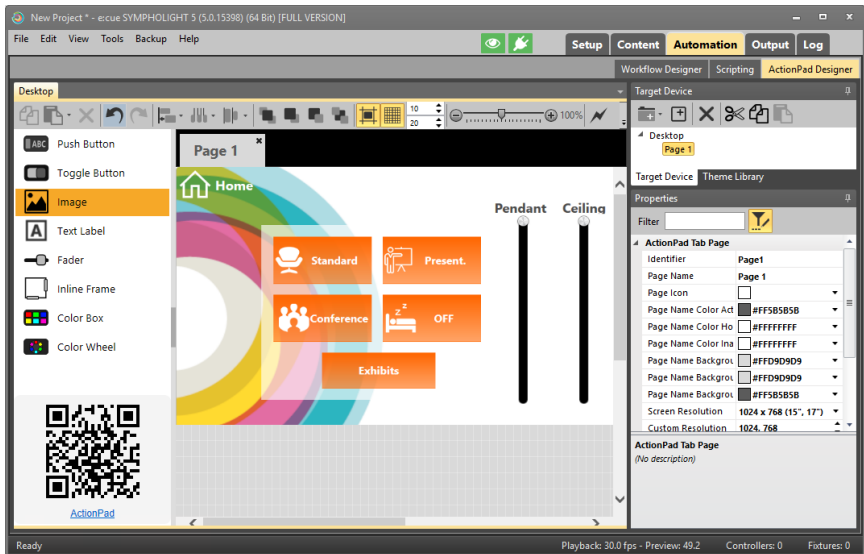
To create a new page for the ActionPad Designer, click on the folder icon left in the Target Device Window or on the plus tab in the page tab. Select the target display size in the Properties Window below the Target Device Window. You can now set all other properties for the page here, like background color or grid usage. Depending on the selected element in the ActionPad Designer, the Properties Window will reflect the properties of the selected element.

To place control elements on the page, drag-and-drop elements from the left window onto the page. Possible elements are Push Buttons, Toggle Buttons, Images, Text Labels, Faders and the Color Pickers. Place the element on the page and resize it to fit your needs. Also adjust the Properties as desired.

A special element is the Inline Frame. After placing it on the page and selecting it, enter an URL in the Properties for the Inline Frame. The content of the web page will be displayed in the frame.

In the top menu of the ActionPad Designer, you can align selected elements or copy and paste elements. Test the look and feel of your Action Pad within SYMPHOLIGHT by clicking the flash icon in the top toolbar. Clicking on the earth icon, your standard browser will be opened and redirected to this Action Pad page.

> ⚠️ Make sure that the web server is enabled in the Tools menu of SYMPHOLIGHT! Also adjust the network parameters in the main menu of SYMPHOLIGHT in Tools - Application Settings.

These elements are available in the ActionPad Designer:

Push Button	A button that sends an event in the Workflow Designer when pressed and when released.
Toggle Button	A button that locks and releases when pressed. Sends an Toggle On and Toggle Off event.
Image	An image from the Media Library. You can upload a new image in the dialogue when selecting the Path.
Text Label	A text element.

Fader	A fader element, horizontal or vertical orientation. It has a PortIn to set the fader, and a PortOut to provide the fader position (only sent on change).
Inline Frame	An area on the desktop of the page. Content is fetched from a web site with the URL defined in its properties.
Color Box	A color picker as square. Select with Mapping in the properties the target of the Color Box, usually an SYMPHOLIGHT Solid Color effect, e.g. Solid Color White from the Factory Effects Library.
Color Wheel	Like above, but as a wheel.

The Theme Library, sharing the place with the Target Devices, offers predefined design themes of ActionPad properties and the possibility to create your own designs.

12.7.1   Element properties

All properties regarding the look and feel, like colors, font types and sizes, orientation and more are self-explaining due to the property names. Additionally, there are help text comments in the small window below the properties. As the Action Pad uses HTML5 to create the browser content, there are many options for styles and behavior. Some properties are not self-explaining, so here is more information about these special properties.

Text End Ellipsis	If the text in a button is to long, the name will be shortened and ends with "...".
Tool Tip	A text that gets displayed in the web browser as help when the user hovers over a button.
Range Step	Property for faders, allows to increase the positions in defined steps only.

When editing text like button text, _Shift + Enter_ will create a line break. _Enter_ alone will commit the change.

12.7.2   Assigning actions to GUI elements

After placing and defining the Action Pad pages with buttons and faders, you have to assign an action to the elements in the page. For buttons and faders, this is done in the

Workflow Designer. See the previous chapter about the Workflow Designer. Use the "State configuration" at the"IsPressed" property to link buttons to objects.
The Color Pickers do not create events. In the Properties Window for a Color Picker, use the Mapping property to assign the Color Picker to a Solid Color Effect, e.g. Solid Color White from the Factory Effects Library in the Timeline of a Sequence. For Color Pickers and Color Picker Wheels, the Workflow Designer provides also Workflow Blocks.
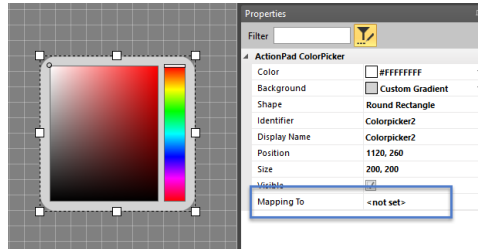
12.7.3   Accessing the Action Pad on mobile devices

In the lower left part of the ActionPad Designer, there is a QR field with the URL of the Action Pad on this server.

Scan the QR code with an iOS or Android application and the mobile device can open the Action Pad by the given URL.
If you already plan to use a specific device for remote control, select the right desktop sizes in the Target Device list of the ActionPad Designer. There are lots of predefined desktop sizes in the ActionPad Designer.
If more than one device project has been created, the individual devices can be accessed by using the correct URL. The scheme is as follows:
http://[IP]:[Port]/ActionPad2/ActionPad2.html?idx=0
Where [IP] is in the following format: 192.168.123.101, [Port] is the same value as configured in the SYMPHOLIGHT Settings and the value for idx (zero based) is matching the position of the device in the Target Device Window of the ActionPad Designer Window.

12.8   SYMPHOLIGHT Remote Console

You can also access SYMPHOLIGHT and the Action Pad via the Remote Console. Use the values defined in SYMPHOLIGHT's system settings.
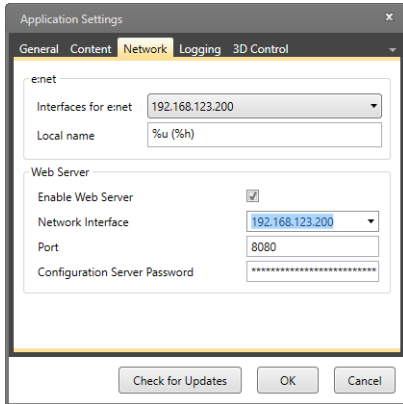
With your web browser, use the system's IP address and port as URL and enter it in the browser's address line. Here:
http://192.168.123.200:8080
The Server Password defined in the settings is required to access the Remote Console. The default setting is "ecue".
Log-in to the Remote Console with user = admin and the given password.
Next the Remote Console gets shown.
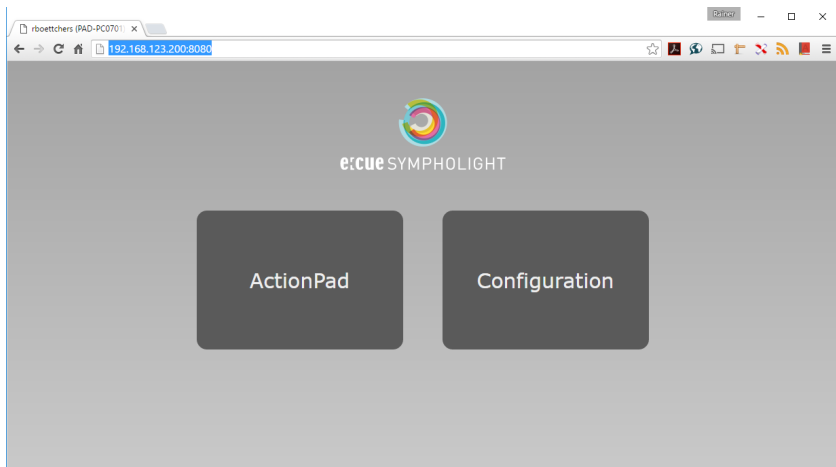
From here, you can call the Action Pad, or set some system settings, or start and stop Sequences. This is in general the same user interface as for the SYMPL Core S. See „14 SYMPL Core engines" on page 296 for more details.

13 Output Window

13.1   Principles

The Output window shows the real results being sent to the Layouts, which are the fixtures respectively.
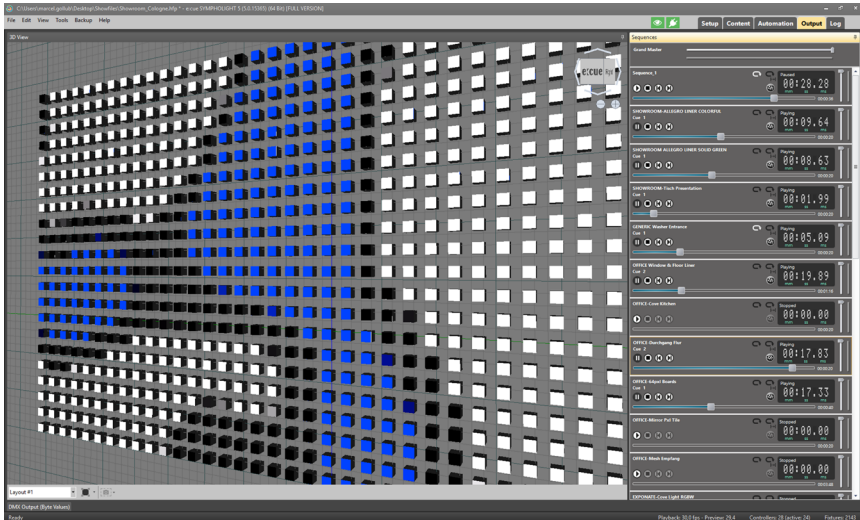
In the center, there is a preview of the currently running and viewed Layout. If there are several Layouts, you can select the one to view. In the Preview area, you can zoom in and out with the mouse wheel. Below the Output view, the DMX data is displayed. You can select either byte values or raw RGB. Selection is made with the two tabbed areas below the DMX Output window.

On the right side window, there is the Play Control Window. You can start, pause or stop the Sequences. This allows checking and testing of shows and configurations. Additionally you can set Opacity in every run control with the fader on the right. Above all play controls, the grand master sets intensity for all fixtures and Sequences.

You can rearrange the player fields by keeping a player clicked and moving it to another place. A yellow bar shows the resulting position.

The Loop icons on the right side of the players allows looping. There is one icon for Sequence Loop and another icon for Sequence Segment Loop. Both icons are linked with the Sequence Properties of the individual sequences. If you want to use the Loop Segment feature, please make sure the Loop Segment is enabled and defined in the respective Sequence Property. You can access the Sequence Property in the Content Tab. Both Loop Properties are saved in and loaded from the Show File.

The Finish icon disables all loops in the Sequence until the Sequence reaches its end and stops.

13.2   Player window icons, commands and functions

1	Play controls, play/pause, stop, skip to previous Cue, skip to next Cue.
2	Current position in Sequence, move to change position.
3	Total length of Sequence.
4	End loop when reaching end of Sequence.
5	Enable Sequence Loop mode.
6	Enable Loop Sequence Segment.
7	Submaster with visualization for this Sequence. The visualization is showing the live value of the Submaster, for example if it is influenced by the Workflow Designer and/or a fader on in the GUI Editor.
8	Grand Master with visualization. The visualization is showing the live value of the Grand Master, for example if it is influenced by the Workflow Designer and/or a fader on in the GUI Editor.
9	Current play position of the Sequence. The time format will be adapted automatically for Sequences longer than 60 minutes and longer.

13.3   Z-position and priority

The position of a player determines its priority. The higher the position of the player, the higher the priority of the generated output.

13.4   DMX Output window

When viewing byte values, you can see the real byte values sent as DMX values. Select the currently viewed addresses and universe with the option left above the window. Select a wide or narrow display of data. In narrow display, the rows are limited to 16 byte ranges. You can also highlight changed values by displaying them in red.

14 SYMPL Core engines

SYMPL Core engines are SYMPHOLIGHT instances which run the same SYMPHOLIGHT application, i.e. project, as a server but in a different environment. They cannot be used to develop a project, but run a completed project in a runtime-only environment. The main use of SYMPL Cores are applications that do not require the performance of a PC-based SYMPHOLIGHT server but the same functionality. As there is no user interface like on a SYMPHOLIGHT server, the SYMPL Cores are controlled via the Action Pad or the Remote Console.

14.1   System overview

For development, connect the SYMPL Core S to the network of the SYMPHOLIGHT server.
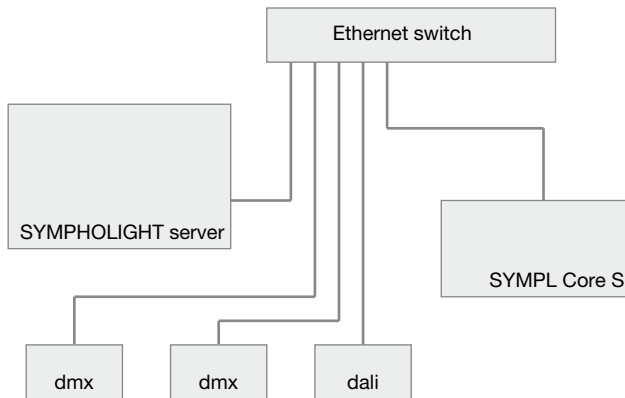
SYMPL Nodes

After booting, the Core S will become visible in the device list of SYMPHOLIGHT. The Core S is not listed as an interface device like the other SYMPL Nodes and Butlers,

but has an own chapter in the device list as Available SYMPHOLIGHTs.
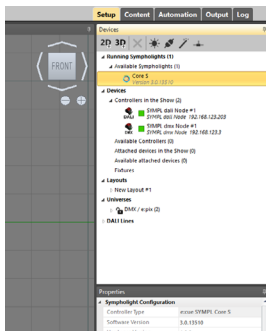
Like for any other device, the main properties for the SYMPL Core S are editable in the Properties Window. Only values in black can be changed, like the name and network properties. Also the SYMPL Core S uses DHCP as default. IP addresses can be set to manual configuration, in this case be aware of possible IP address conflicts. Please refer to „10.2.2 Network addresses" on page 32 for more information.

A right-click on the SYMPL Core S in the device list opens a context menu for additional function for the Core S.
Before assigning a project and setting all properties of the Core S, you have to configure it completely, e.g. assign a Password. This can be done with the context menu to open a browser for this purpose.

> ⚠️ Before assigning a project and setting all properties of the Core S, you have to configure it completely, e.g. assign a Password. With the context menu of the Core S in the Device Window of the Setup tab open a browser for this purpose.

Over the context menu of the SYMPL Core S the following functions are available:

Upload project	Upload a project to the Core S. Set a password in the Web Interface beforehand, otherwise the upload will not run. Click the Handover & Close button when you want to immediately start the project on the Core S and its devices.
Delete currently loaded project	Deletes the currently loaded project from the Core S.
Mute controller detection	Select to become invisible for other SYMPHOLIGHT instances in the network. Click again to unmute.
Share License	Use the Core S as License Key. For details see page 338.
Restart controller	Restart the Core S, e. g. in case of project problems.
Trigger system update	Enable an update for the firmware of the Core S.
Reset controller to factory defaults	Reset all settings and configuration to the factory defaults. This will also delete the project.
Download log files	Download the log files of the Core S via a web browser.

Open Remote Control in Browser	Configure and control the Core S in a web browser.
Open Rest API (Swagger) in Browser	Configure and control the Core S in a Rest API (Swagger).
Configure Network	Configure the network settings of the Core S.

Before using the Core S in full functionality, select the last entry to start the default web browser on the SYMPHOLIGHT server.

14.2   Configuring the SYMPL Core S

With a properly configured network, you will see the start screen. Select Dashboard from the left menu.
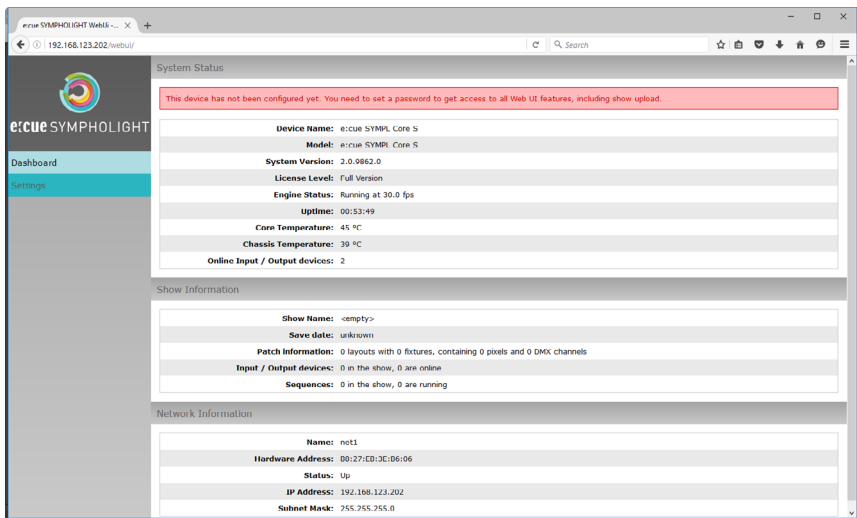
In the Dashboard, you see an overview about the SYMPL Core S and the uploaded project. As no project is loaded when initially accessing the Dashboard, this part is empty. As the SYMPL Core S is fresh out of the box, the system reminds you on assigning a password to this Core S. Click on Settings to do so.
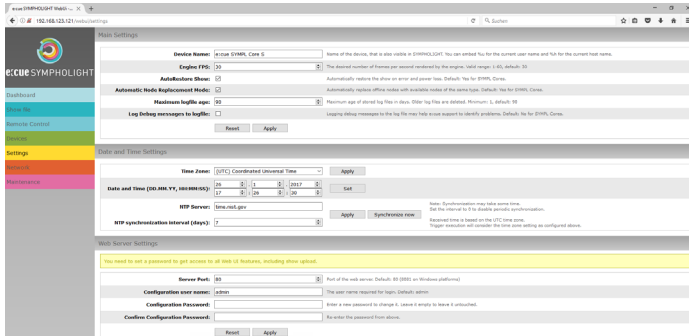
The editable parameters in the Settings window all have a description on the right side. Edit the basic system settings and click _Apply_ to commit the changes. Assign a password to this SYMPL Core S and Click _Apply_. The system has now its basic configuration and the complete configuration is possible.
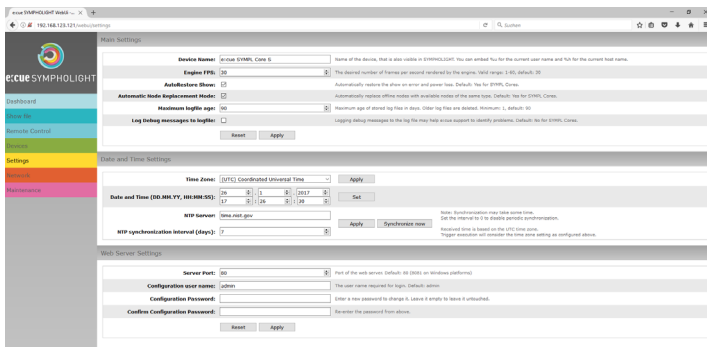
The settings and parameters in the Settings area should be self-explaining like IP address, time zone or device overviews.
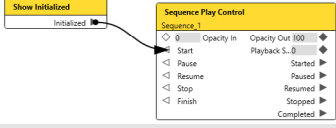
Dashboard	Overview of the SYMPL Core S, firmware version, uptime, project details, input and output devices and more.
Project file	Upload a project to this SYMPL Core S by drag-and-drop or upload. *.lprj and *hfp formats are supported. By uploading the project from any source, no SYMPHOLIGHT server is necessary. Use an USB memory stick.

Remote Control	Other SYMPHOLIGHT instances from which remote control is possible. You can execute basic operations from Remote Control instances, for example to start and stop Sequences.
Devices	SYMPHOLIGHT instances and devices, available or included.
Settings	Basic device and network settings.
Network	Network configuration and network information. You can select automatic or static address distribution.
Maintenance	Reset the SYMPL Core S or run system reset.

After assigning a SYMPHOLIGHT project to the SYMPL Core S via the web interface, the Core S will begin executing the project. As long as the SYMPHOLIGHT server runs, the Core S cannot get SYMPL Nodes or other interfaces. As soon as the SYMPHOLIGHT server releases the devices, the SYMPL Core S will take over the devices and integrate them as configured in the running project.

> If you want the Core S or SYMPHOLIGHT server to start a Sequence or do anything else after system start, do not forget to include the Project Initialized Block in the Workflow Designer.

To control the Core S, use the Action Pad or the Remote Control in a web browser.

14.3 Using the SYMPL Core S as a License Key for SYMPHOLIGHT

The SYMPL Core S can be used as License Key for the SYMPHOLIGHT Software. In order to transfer the License to another SYMPHOLIGHT instance, go to the Setup Tab of the SYMPHOLIGHT the License shall be transfered to. Open the context menu of the SYMPL Core S in the Devices Window. Select Share License and enter the password for the SYMPL Core S you've configured before. If you did not set a password for the SYMPL Core S before, please refer to chapter 14.2 for further instructions and set a password.

Once the License has been transferred successfully, the red "Trial Version" label will be removed. This can take a few moments.

⚠ Please be aware that the Share License feature is not meant for permanent operation, but is limited for a certain period of time (4 hours). Once the License is expired, it can be renewed by selecting Share License from the context menu again.

14.4   Selecting the Action Pad

Look up the IP address of the SYMPL Core S in the Properties Window in the Setup tab of SYMPHOLIGHT, or call the Remote Control from the SYMPHOLIGHT server's device list. Use the plain IP address in the web browser and you can select between the Action Pad and the Configuration.

For later direct access, bookmark the Action Pad's IP address on this Core S.

14.5   Update SYMPHOLIGHT in the SYMPL Core S

In some cases, e. g. if a new SYMPHOLIGHT version is available, an update of the SYMPL Core S firmware is necessary. The firmware version in which a project is created and the version with which this project is executed have to be the same. An exclamation mark ⚠ in front of the Core S SYMPHOLIGHT instance in the Setup Window informs you about a pending update.

In this case, download the SYMPL Core S Updater from the e:cue website and save it on your local drive. Double-click to execute the Updater.

Choose between using an USB memory stick or an SD card:

HowTo update SYMPL Core S with an USB memory stick (recommended)
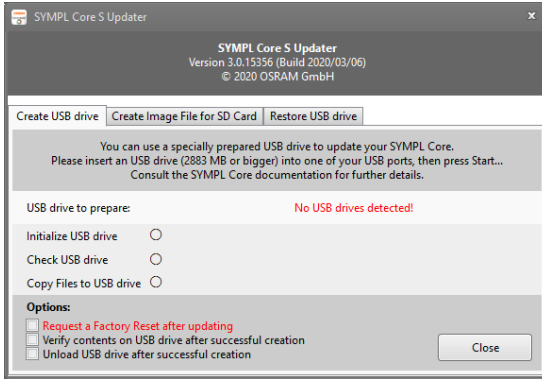
You will need a USB memory stick with at least 8 Gbytes capacity.

Plug the USB memory stick into your PC. If the Updater found no USB memory stick, it will show an error message. If it found one or more USB memory sticks, you have a list to select the appropriate USB memory stick.

You can even plug in the USB memory stick while the Updater runs.

Set the check marks, if you want the contents of the USB memory stick to be verified, and if you want the system to remove the USB memory stick after the process. After the Updater has finished the process, the USB memory stick has a complete image of the SYMPL Core S firmware.

— Please backup the Project File from the SYMPL Core S WebUI before starting the update process in case any issue comes up during the update process.
— Disconnect the SYMPL Core S from power.
— Plug in the USB memory stick into the SYMPL Core S.
— Keep the Identify button pressed and reconnect the power supply.
— The Master/Slave LED will start to blink.
— Release the button immediately.

After the update, the SYMPL Core S will restart and return to normal operation.

Note:

If you update from a firmware version smaller than v5, the Updater will restart the Core S once during the update process. Do not disconnect the USB memory stick!

HowTo update SYMPL Core S image directly to SD card

Follow these steps in order:

1. Open file SymplCoreUpdater_*.exe.
2. Switch to tab "Create Image File for SD Card".
3. Select a location on your local disk and start the process.
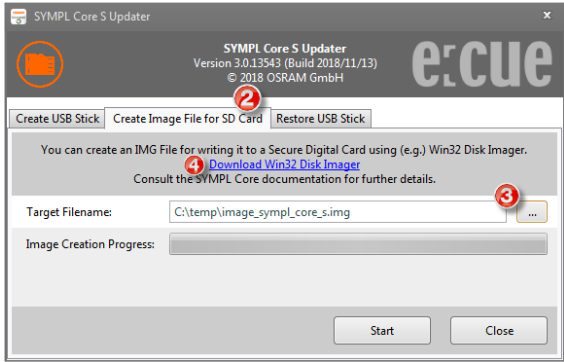
4. Download and install Win32 Disk Imager (or another image-writing tool). See link in dialog.
5. Open Win32 Disk Imager, choose the image file which you created under 4.
6. Select the drive letter of the SD card.

ATTENTION: This will unrecoverably wipe all data on that drive. So double (better triple!) check to choose the drive letter matching your SD card!
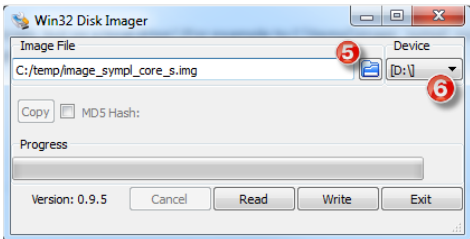
7. Check again if the drive letter corresponds to the SD card. If so, confirm overwriting.
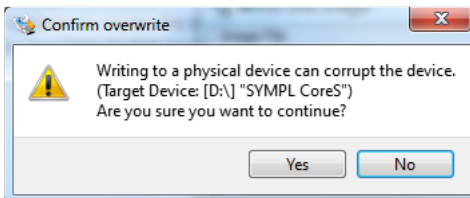
8. When done, power off the SYMPL Core S. Insert the SD card and power it on.

Be patient on the first boot, it might take longer up to ~3 minutes.

When Master/Slave LED of SYMPL Core S lights in blue, the device has booted and is operational.

Hint: Don't forget to set a password on the website of the SYMPL Core S > Configuration > Settings.

14.6   Resetting the SYMPL Core S to factory default

The Identify button can also be used to reset the device to factory state:
— Keep the button pressed while powering up until Master/Slave and Error/Status LEDs blink alternating (5s), then blink together (5s).
— Releasing the button now resets the device to its factory settings.
— Keeping the button pressed further on, the device proceeds to normal operation. No changes apply.

14.7   Accessing the SYMPL Core S Logfiles

Please access the following URL in the browser: http://[IP Address]/logs/.

14.8   Maximum number of SYMPHOLIGHT instances

There should not be more than ten instances of SYMPHOLIGHT in a network. Using more than ten instances will decrease the performance of the overall configuration.

14.9   Using Video with the SYMPL Core S

Please be aware that the SYMPL Core S is capable of decoding a maximum of two Videos in parallel.

15 RDM

This is an explanatory chapter of RDM and using RDM in SYMPHOLIGHT.

15.1   RDM Glossary

Request

A network message send from a SYMPHOLIGHT to a RDM device.

Response

A network message send from a RDM device to SYMPHOLIGHT as reaction to a request.

Command

In the RDM context a command is a type of a message which is send in a request and response.

Parameter

A parameter is a dataset stored on the RDM device. E.g.: Sensor Value

Attribute

An attribute is a value from within the parameter dataset. E.g.: Sensor Value -> Present Value

15.2   Remote Device Management (RDM)

RDM permits intelligent bi-directional communication between devices from multiple manufacturers utilizing a modified DMX512 data link. RDM permits a controlling device to discover and then configure, monitor and manage intermediate and end-devices connected through a DMX512 network. SYMPHOLIGHT provides the functionality of such a controlling device.

15.2.1   Topology

SYMPHOLIGHT works with a topology containing controllers (like Butler S2, SYMPL dmx Node …) having different amounts of DMX512 ports. RDM extends this topology with so called root-devices. This can be a luminaire, ballast or similar. Root-devices can have child elements, those devices are called sub-devices. The total topology visualized:
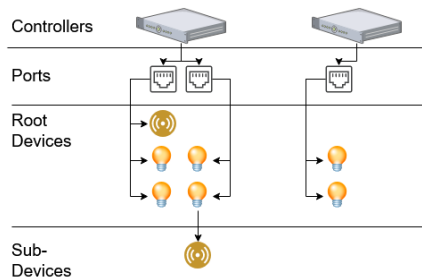
15.2.2   Addressing

RDM devices (root- and sub-devices) have a unique identifier. The UID consists of three parts: 1) The manufacturer ID is unique for each manufacturer of RDM capable devices and is assigned to the manufacturer by the ESTA. 2) The device ID is assigned by the manufacturer within the production process. The combination of manufacturer ID and device ID is globally unique to one root-device. 3) If a root-device has sub-devices, those will have a sub-device index in range 1 - 255.

Name	Size	Range	Note
ESTA Manufacturer ID	16Bit	0x0001 – 0x7FFF	Unique ID per manufacturer (managed by ESTA)
Device ID	32 Bit	1 - 4.294.967.295	Unique ID per root device of the manufacturer
Sub-device Index	16 Bit	0 - 255	Zero for root-devices, otherwise within range

15.2.3   Discovery

To make it possible to communicate with root- or sub-devices, the devices have to be known by the controlling entity (SYMPHOLIGHT). To realize that, the discovery process of the entity will send requests using a binary search algorithm on the addresses. The discovery process proceeds until every root- and sub-device has been found and all the information the controller requests is provided.

15.2.4   Update discovery

To make sure SYMPHOLIGHT recognizes when a device goes offline, a provided value of a device changes, or new devices are added, the discovery takes place in the background all the time.

15.3   RDM view in Setup tab

The complete topology of the discovered RDM devices is visualized in the RDM view. The RDM view is located at the bottom of the Setup tab: choose "RDM" at the right side of "DALI". An example of a controller (Butler S2 12/12) with two ports:
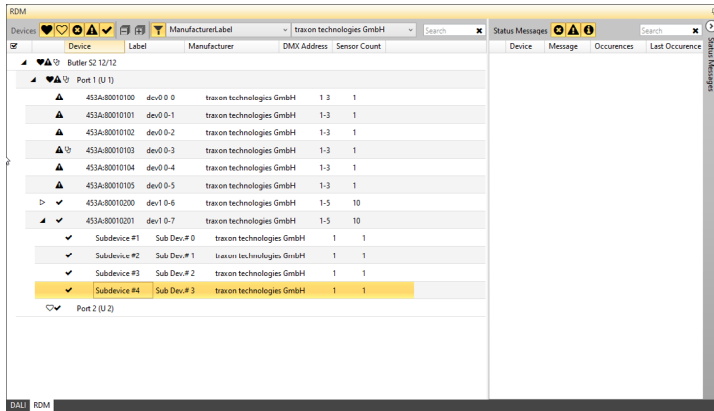
Only Port 1 has RDM enabled. This is visualized with the filled heart icon next to Port 1. Port 2 has RDM disabled, the heart icon is not filled. There are eight root-devices found at Port 1. The last two root-devices do also have sub-devices. The root-devices 1 to 6 have reported a warning which is visualized with the exclamation mark (this happens e.g. when a sensor of the device is not inside its valid range). The stethoscope visualizes that the system is currently updating the attributes of the device. With the toolbar above it is possible to filter the devices based on their status and attribute values. The status messages which are provided by some RDM devices are visualized at the right side

15.3.1   Update optimization

To improve the performance of the discovery time, SYMPHOLIGHT does not request every information a device provides every round. Some information is not supposed to change every round (e.g. the manufacturer label of a fixture will not change from one round to another). The priority of the information request is pre-configured for all default attributes defined in the RDM specification.

15.3.2   Adjust attribute update priority

Sometimes it is necessary to change the update interval for a specific attribute. This can be done within the property grid of SYMPHOLIGHT:
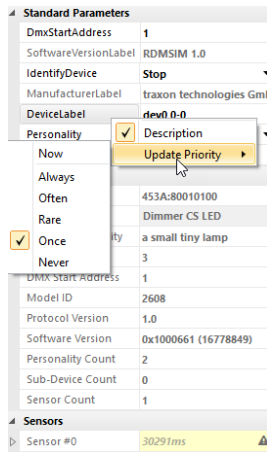
A right-click on the attributes opens the context menu. The context menu provides the possibility to choose between different update intervals. After changing the interval, the selected attribute will be requested one time (as fast as possible). The update priority will then continue with the chosen interval.

15.4   RDM in Workflow-Designer

For monitoring and handling a huge amount of RDM devices, the Workflow-Designer provides blocks which can be used for the automation and monitoring of RDM devices.

15.4.1   Observe RDM

The Observe RDM block provides the possibility to listen for RDM messages received by SYMPHOLIGHT. The simplest use case of the block is the monitoring of device found and device lost events. This is the default configuration.
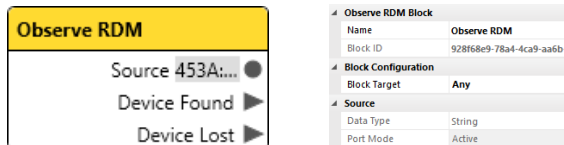
The Source PortOut of the block will automatically be set to the RDM UID of the device which was found or lost every time when such an event takes place. Since the Block Target is set to Any in the property grid, the block reacts on every device lost / found notification independent from the controller or port where the lost / found device was located.

A more advanced use case is when the Observe RDM block is configured to listen for a single root-device (see Block Target):
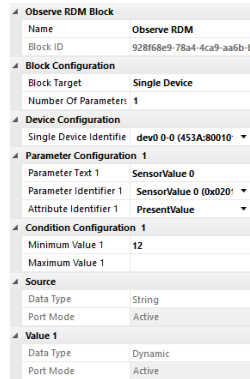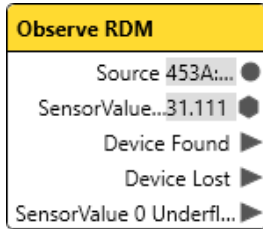
Define the root-device with its UID at Device Configuration. Furthermore, the block listens only for changes of one attribute from one parameter (see Number Of Parameters). In Parameter Identifier 1, the block is set to listen for changes of the SensorValue 0. Since there is multiple information regarding a sensor (recorded value, lowest value, …), the Attribute Identifier 1 has to be specified too. Set it to here to "PresentValue". The present sensor value is now available and will update every time the present value is received from the RDM device in the background. To make the block send a trigger e.g. when the present sensor value falls below 12, set the Minimum Value 1 to "12". The event port for the underflow is visible at the block and is evaluated every time the sensor values changes.

15.4.2   Build RDM Command

The Build RDM Command block provides a simple possibility to build RDM commands which can afterwards be send by using the Send RDM block.
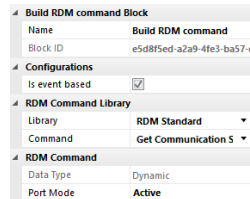
To specify the command which shall be build, it is necessary to choose the library where the command is located in. All "default" commands are located in the RDM Standard library. Additionally, SYMPHOLIGHT provides an Allegro library which contains custom commands used by Allegro fixtures.

Once a library is selected, the Command drop-down menu contains all commands of chosen the library. To search in the drop-down menu, type in the command property. Some commands are Get commands, others are Set commands. When sending a Get command, use an Observe RDM block and listen for the according attribute of the RDM device. Some Get commands and all Set commands require the specification of some values in order to build a valid RDM command. Specify those values as usual by PortIn values or property configuration.

> **Advanced:**
> An RDM device manufacturer may define own commands. To facilitate the usage of those commands in SYMPHOLIGHT, it is possible to define those commands in a own library file as it was done for the Allegro fixtures. To do so, create a *.slrt file complying with the XML format which is defined in the RdmCommandTemplateSchema.xsd. Place the *.slrt file in the „%appdata%\ecue\SYMPHOLIGHT <Version_Number>\RDM Command Templates" directory. The schema file is located at the same directory.

15.4.3   Send RDM

The Send RDM block forwards commands to RDM devices.
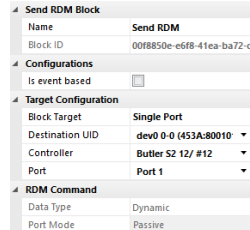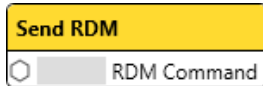
For example, RDM commands which are created by the Build RDM Command block can be given to the Send RDM block. In this block, configure the destination to which the command shall be send. The Block Target property offers All Controllers, Controller Ports and Single Port. In case of All Controllers, it is only possible to send commands to broadcast addresses. In case of Controller Ports, specify a controller which will send those commands. Like the All Controllers setting, the Controller Ports setting allows only broadcast addresses.

The Single Port setting allows the specification of a controller, port and either the UID of a specific RDM device or a broadcast address.

15.5   RDM in scripting

The management of a big number of RDM devices or the handling of complex rule chains with the workflow designer is best handled with scripts for RDM. The following chapter gives a slight overview about the functions available in scripting. Focus is on current limitations which can be solved by combining scripting with the Workflow Designer. A basic example can be found in the examples folder which is per default installed with SYMPHOLIGHT on the computer. The folder has the name Examples and is located in the installation directory of SYMPHOLIGHT.

15.5.1   Finding RDM devices

RDM works on basis of the devices described in „15.2.1 Topology" on page 343. To find specific devices, there are different possibilities. A) Go through the hierarchy of the devices and try to find the desired device. B) Use existing methods and properties providing simple ways to find a device, for example as shown below.

In case a specific RDM root-device is searched for, but the controller and the port where it is connected are unknown. Only the UID of the root-device is known. In that case, use the following code:

```
var rootDevice = ecue.Rdm.GetRootDeviceByUid(0x453A, 0x80020101);
```

The code consist of the following parts: The first word is the keyword var. It creates a new variable which gets the name rootDevice with the next word. Then the return value of the method ecue.Rdm.GetRootDeviceByUid is assigned to the rootDevice variable. This method requires two parameters. The first one is 0x453A. It is the manufacturer ID of Traxon. The second parameter 0x80020101 is the device ID of the wanted root-device. While typing in the editor, the code completion tool assists with proposing possible functionality. For example, to access a functionality of SYMPHOLIGHT, begin with „ecue.", then suggestions for the categories which exist are shown. Currently this would be DMX, RDM, Logging, Sequences and many more. Selecting RDM would extend the written code to ecue.RDM while typing a dot would now suggest all the methods and properties accessible for RDM in SYMPHOLIGHT.

15.5.2   Working with RDM devices

To demonstrate how to work with RDM devices, the example from the previous chapter is taken. Now an RDM command shall be sent which sets a DMX address to the earlier defined root-device. To realize that in a very simple way, a combination of the functionalities from the Workflow Designer and scripting is recommended. The start is scripting part; for example with the following code:

```
/// <summary>
/// This class is public static, so it's methods can be accessed from the workflow designer.
/// </summary>
public static class RdmTest01
{
    /// <summary>
    /// This method is public static, so it can be accessed from the workflow designer.
    /// </summary>
    /// <param name="command">RDM command to be send to root device.</param>
    public static void SendCommand(IRdmCommand command)
    {
// Find the RDM root device based on the specified manufacturer ID and device ID
        var rootDevice = ecue.Rdm.GetRootDeviceByUid(0x453A, 0x80020101);

// Send the command we got as method parameter e.g. from the workflow designer  // to the root
Device we found in the previous line of code.
        rootDevice.Send(command);
    }
}
```

This example has a class called RdmTest01 which is having a method called SendCommand. This method expects one method parameter. The parameter is required to be of the type IRdmCommand and has the name command. The next line of code is known from the previous chapter. In the following line, the rootDevice is used to call the Send method for this rootDevice. Like the SendCommand method, the Send method expects a method parameter of type IRdmCommand. Give the parameter command of the SendCommand method into the Send method as parameter.

Now the questions is "Who is providing the command method parameter of the SendCommand method?" That's where the Workflow Designer comes in. As shown below, the Set DMX Start Address command is built in the Workflow Designer:
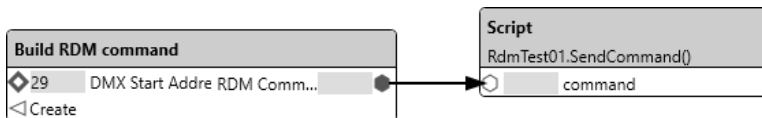
The resulting RDM command has the exact type as expected parameter in the script method (IRdmCommand). Call the script method by using a Script block targeting the SendCommand method and class RdmTest01. Now triggering the Create EventIn of the Build RDM Command block creates the DMX Start Address RDM command. This RDM command is forwarded to the script method. The method sends the RDM command to the root-device. Now the RDM root-device has set the DMX Start Address to the received value (here 29).

It is possible to create those commands also directly in scripting. This is also demonstrated in the RDM example project located in the Examples folder of the SYMPHOLIGHT installation directory.

16 REST API

For an open and secure interaction with other systems (3rd party, cloud, etc.), SYMPHOLIGHT provides a SYMPHOLIGHT (REST) Web API. You can establish a secure connection with the new encrypted (SSL) webserver and send commands or receive status information. This will give you the possibility to handle several SYMPHOLIGHT instances from one central system (PC as well as Core S).

⚠️ The SYMPL Core S does not support SSL encryption.

💡 Using Rest API and Scripting requires prior knowledge of these topics.

16.1   REST API setup

Set the network user name and password in Tools - Application Settings - Network - Web Server. They are the same for the Rest API authentication.
Also configure the Rest API settings at this location. It is recommended to generate an own SSL PFX certificate. The Rest API settings will only take effect after a software restart.

16.2   SYMPHOLIGHT Web API

With the SYMPHOLIGHT Rest API it is possible to write your own client that can control a Sympholight instance. The Rest API adheres to the Swagger specification that means that you can use the Swagger tools to generate clients for different programming languages like C#, TypeScript, JavaScript, etc.

To execute the methods and get responses you need to log in:
— To open the Rest API documentation (Swagger) in a browser, open the context menu of the SYMPHOLIGHT instance in the Setup window:
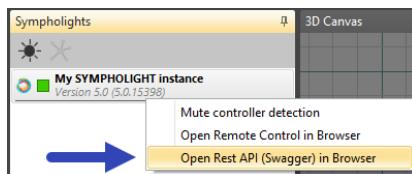
The Swagger web page displays all available methods of the SYMPHOLIGHT Rest API:
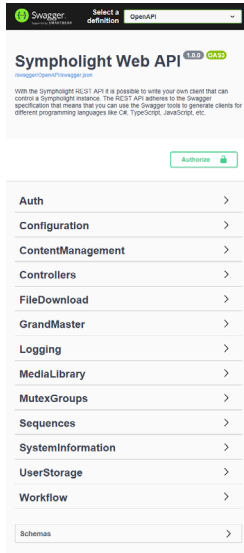
Auth	Authentication.
Configuration	Gets the Configuration that contains the configuration.
ContentManagement	Get, Change, and Interact with Media Effects (Video and Image). Requires additional license.
Controllers	Gets all controllers that are used in the current project.
FileDownload	Get and Interact with downloads.
GrandMaster	Get and Change the Grand Master.
Logging	Get log files (also filtered).
MediaLibrary	Get and Interact with the Media Library.
MutexGroups	Get and Interact (Play, Pause, next…) with Mutex Groups.

Sequences	Get and Interact with Sequences (incl. opacity) and Cues.
SystemInformation	Get System Status.
UserStorage	Get and Interact with files and directories in the User Data directory.
Workflow	Get all Blocks and Write (or get) Values from the "Value Block".

— Select at Auth the Post method /login:
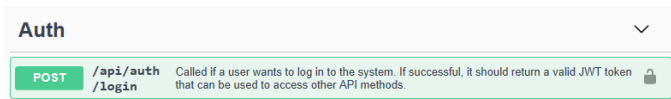
— Click the Try it out button:
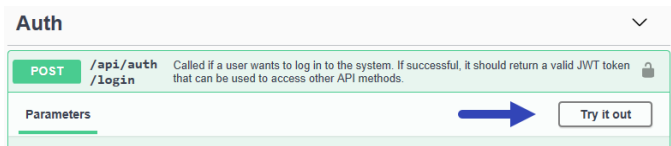
— In the Request body, replace the text String in the username row with your user name as defined in the network settings of SYMPHOLIGHT. Also replace the text String in the password row with your password as defined in the network settings:
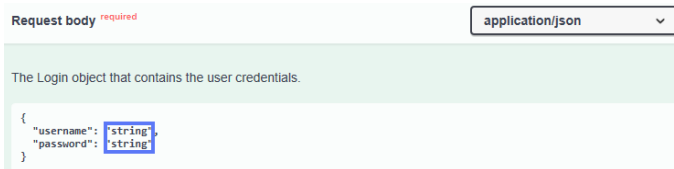
— Click the Execute button.
— In Responses - Code Details - Response body, copy the token value (excl. the quotation marks):

— Click the Authorize button at the top of the methods list:

— Enter the following text into the Value field and add right behind the text your copied token value:

Bearer +[token value]

Do not omit the space character. Do not enter the square brackets.

— Click the Authorize button to login.

The login is completed. You can now execute the methods.

16.3   Examples

Examples files are stored at
C:\ProgramData\ecue\SYMPHOLIGHT 5.0\Examples\REST API.

The Media Exchange example shows the authentication in Python language.
The Text Input example offers text input fields in the ActionPad, usable also in the
Workflow Designer. It uses the Static File Hosting.

17 Glossary

Object   Any element in SYMPHOLIGHT, a Sequence, an Effect, a button in the Action Pad. Objects in SYMPHOLIGHT can be accessed via Scripting or with the Workflow Designer, to influence a Property or to call a function of this object.

Property   An attribute of an object. Properties can be e. g. colors and gradients, text, orientation, size, address / ID.

Sequence   The complete content that creates a "project". A Sequence contains Layers, Layers contain Effects or media, Cues and Triggers, Scenes and other content.

Layer   A track in a Sequence. All Layers in a Sequence contain the final content. Layers in SYMPHOLIGHT are organized in hierarchical order: content of Layers below another Layer are only visible on the fixture if the upper Layer has transparent pixels or an opacity < 100%.

Cue   A control element in SYMPHOLIGHT. Cues in a Sequence can be addressed as time markers. You can skip to Cues to control the execution of a Sequence.

Triggers   a) Triggers in a Cue are control objects, they cause an event in the Workflow Designer which can be caught and lead to an action.
b) An output of a Block in the Workflow Designer. In case of an event, e. g. a time and date, receiving an external signal, or pressing a DALI button, the output sends a Trigger to the following Block, causing the execution of an action or to calculate a result.

Block   A function element in the Workflow Designer. A Block consists of data inputs and outputs and/or Trigger inputs and outputs. The internal functions of a Block execute an action, calculate a result or send data.

Method   A function of an object. Derived from class-oriented programming, executing a function or an action.

Layout   The configuration and placement of fixtures and devices in a SYMPHOLIGHT project. The Layout is created in the Setup tab of SYMPHOLIGHT, providing input and output channels to fixtures, interfaces or system functions.

Effect   An algorithm creating content in a Layer. In general, Effects provide colors, intensities, media or text content that is placed in a Layer.

Fixture   General expression for a lighting system, usually equipped with LEDs, but also laser driven. Like LED liners, head washers or video matrix panels.

DMX   DMX512 (Digital Multiplex) is a standard for digital communication networks to control stage lighting and effects such as fog machines and moving lights. DMX512 employs EIA-485 differential signalling at its physical layer, in conjunction with a variable-size, packet based communication protocol at 250kBit/s. It is unidirectional and does not include automatic error checking and correction. DMX is the most used connection type in lighting control.

DALI   Digital Addressable Lighting Interface (DALI) is a standard for control lighting in buildings. It was established as a successor for 0-10 V lighting control systems, and as an open standard alternative to Digital Signal Interface (DSI), on which it is based. The DALI standard, which is specified in the IEC 60929 standard for fluorescent lamp ballasts, encompasses the communications protocol and electrical interface for lighting control networks.

KNX   KNX is a standardised (EN 50090, ISO/IEC 14543), OSI-based network communications protocol for intelligent building automation. KNX is the successor to, and convergence of, three previous standards: the European Home Systems Protocol (EHS), BatiBUS, and the European Installation Bus (EIB or Instabus). The KNX standard is administered by the KNX Association.

RDM   Remote Device Management, a protocol based on DMX512 with bidirectional communication capability between a lighting controller and RDM-capable lighting fixtures or devices. In this way, information from fixtures can be given back to the controller, a feature that standard DMX does not support.

e:net   An Ethernet-based e:cue protocol used as the backbone communication standard between most e:cue engines and interfaces.

e:bus   The e:bus is a 2-wire, polarity-independent bus system for communication between lighting components and engines. Due to the free topology design and capability to wear data communication and power supply voltage at the same time, it simplifies cabling and installation costs. e:bus has a maximum cable length of up to 400 m.

e:pix   e:pix is a protocol based on DMX, developed by e:cue, for an improved communication between the VMC and Traxon e:pix-ready LED products.

Engine   An engine is a standalone-capable device that controls fixtures, receives signals and reacts on input from external devices like terminals or switches.

Interface   A device that converts interfaces or protocols, e.g. from e:net to DMX, from e:net to DALI, or from e:net to relay contacts. In general, a device that connects other devices to SYMPHOLIGHT.

Terminal   Any device that transfers user interaction or commands into a lighting system, e.g. Glass Touches.

Universe   Means DMX universe. A DMX universe are all devices connected to one DMX line, so a maximum of 512 addresses.

18 Notes

Downloads and more information at:
www.ecue.com and www.traxontechnologies.com

OSRAM GmbH
Sales Operations
Karl-Schurz-Str. 38
33100 Paderborn, Germany
+49 5251 54648-0
support@ecue.com
www.ecue.com

Head office / Zentrale:
Marcel-Breuer-Strasse 6
80807 Munich, Germany
Phone +49 89 6213-0
Fax +49 89 6213-2020
www.osram.com

Light is OSRAM

Our Brand

e:cue

OSRAM