



Pixel Processing Compere 1.0-5

User Guide



Compere 1.0-5 : User Guide

Trademark Information

The 7thsense logo, and various hardware and software product names are trademarks of 7thSense Design Ltd. Product or company names that may be mentioned in 7thSense publications are tradenames or trademarks of their respective owners, and such trademarks may also be registered in their respective countries. Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Copyright Information

All Rights Reserved. This document is copyrighted © by 7thSense Design Ltd and shall not be reproduced or copied without express written authorisation from 7thSense Design Ltd.

The information in this document is subject to change without notice. 7thSense Design Ltd assumes no responsibility for errors, and/or omissions contained in this information.

Printed: April 2023

This edition is for software version Compere 1.0 build 5
Document ref.: M512-5

E: info@7thsense.one
W: 7thsense.one

7thSense Design Ltd
2 The Courtyard, Shoreham Road
Upper Beeding
Steyning
West Sussex
BN44 3TN
UK

T: +44 (0) 1903 812299

7thSense Design LLC
4207 Vineland Rd
Suite M1
Orlando, FL 32811
USA

T: +1 407 505 5200



Introduction	6
Compere for Juggler Systems	7
Workflow: Juggler in Compere	9
Juggler Software and Firmware	11
Compere UI for Jugglers	13
Installing Compere on a Control PC	15
Terms and Definitions in Compere	16
Keyboard Shortcuts	18
The Juggler View Layout	20
Using Compere Layouts	27
Main Menu	29
Debug Menu	33
Status Bar	35
Task Tracker Panel	36
Settings Panel	37
Preferences	37
Juggler Settings	40
3D (Visualise) Settings	40
Canvas Panel Settings	40
Nodes Panel Settings	41
Compere Projects	42
Project Panel	43
Assembly Panel Context Menus	45
Compere MinGUI	47
Network Configuration	49
Create a Project Group	56
Configure a Display	60
Canvas Output Setup	63
Canvas Window Setup	67
Create and Recall Canvas Arrangements	70
Display Setup via MPCDI	77

Connecting Juggler Systems	79
Juggler Unit Properties	81
Internal Processor	84
Defaults	85
GPIO	85
System Monitor	86
Adding Jugglers as Nodes	86
Video Channels Properties	92
IntelligentSource	96
Video Measure	97
Video Channel Tx	99
Video Rx / Tx Port Interfaces	99
Video Interface Timing	102
Bus Connections in a Juggler System	102
Bus Properties	105
Bridging Two 8-Lane Buses	106
Output Canvas Bus	110
OCB Wizard	116
Connecting Inputs and Outputs	119
EDID Spoofing	121
Nodes: Processing Nodes	123
Presets and Clones	129
Ports Used by Juggler and Compere	131
ASCII / XML External Control	132
Wildcards	136
Message Formats	137
System Status	141
Juggler Status	141
Bus Status	142
Bus Reset	142
Genlock	142

Hotplug Detect	144
EDID Lists	144
Resolution	145
MPCDI	146
Project Controls	147
Presets and Clones	148
Shutdown or Reboot Jugglers	149
Logs and Diagnostics	150
Command Line Arguments	152
Troubleshooting a Juggler System	153
Document Information	154
Index	155

Introduction

This guide

This guide relates to Compere 1.0 build 5, the software used to configure systems using Juggler 1, the first member of the Juggler family in a 1U rack configuration with 4 video channels. It describes Juggler 1 and how it works, how to connect it to media inputs and device outputs, network Jugglers together, redirect inputs and process them to outputs, and create the Projects that store all the configurations.

Juggler and Compere

Juggler™ and Compere™ go together. Juggler is the hardware that processes the pixels and Compere, like an event manager, is the control software and user interface that tells one or more Jugglers what to do. Together, any range of digital inputs can be transformed and directed to an any range of outputs, because Juggler modules can be daisy-chained on a data bus to expand inputs, outputs and bandwidth as required. Compere does much more besides in its Visualise role (separate licence), but this guide focuses on Juggler system set up.

Juggler is a highly modular FPGA-based pixel processor, providing the highest hardware performance and efficiency, combined with ultimate connectivity. It is specified according to application, with a very wide choice of input and output options: DisplayPort™ 1.2, HDMI™ 2.0, SDI, and with Video over IP in development. The Juggler system can process a data bandwidth of approximately 14 GB/s. As an example, that equates to a canvas size of 16K × 4K at 60 fps 10-bit 4:4:4.

The configuration of a particular Juggler module will optimise use of the available bandwidth according to requirement, and the Juggler concept is one of tremendous flexibility. Modules can be daisy-chained via a high-speed box-to-box fibre-optic link, to expand I/O type and capacity. Juggler can also display all inputs via an NDI thumbnail matrix on the front panel.

Compere for Juggler Systems

Juggler

Juggler is not just a standalone rack module. We imagine it as stage performer in an event, where as many Jugglers and other performers as you want can join, take part, or leave, all under the direction of a Compere, or stage manager that holds the event together.

Juggler is a modular hardware base for full high bit-depth pixel-based colour processing, warp and blend. Its main component board is a field-programmable gate array (FPGA), which makes it extremely versatile for performing highly specialised processing tasks. It accepts a scalable number of inputs and delivers a scalable number of outputs, in many protocols. 2D, passive and active stereo 3D are fully supported.

With latency reduced to an absolute minimum, Juggler is ideal for live performance. All sources can be genlocked for frame accuracy, with instant system failover across all connected devices via frame-data monitoring (an API is available for third-party sources).

➤ See the user guide: [Operating Juggler 1 Pixel Processor](#)

Compere

Compere (pronounced 'KOM pair') is the software system that manages all Juggler connections in a 'stage set' (we call it the Project Group), defines inputs, outputs and manages displays. A version of Compere runs in every Juggler on a Linux OS that drives the front panel touch display and communicates with the FPGA.

Compere provides the user interface (UI) for a Juggler system. Here you create Project Groups of Jugglers and configure all the inputs and outputs of each Juggler involved. All group activity can then be managed via external control from multiple network points.

Compere does much more, but for configuring your Juggler system with its media inputs and display outputs, you need to become familiar with the Juggler side of Compere.

Connect up your hardware before you venture into Compere. Once that is done, connect into the system from a workstation control PC with Compere running, and every part of the system will be recognised and identified, along with all their available ports, in order to create the relationships between each Juggler, their input sources and output devices.

Projects and Groups

Projects are a bit like stage sets, with scenery and equipment ready for a play or show. When you create or see a Project in Compere, you will see a list of what belongs together for the 'stage set':

Jugglers, displays, media devices, projectors, 3D models and so on, that are assigned as a Project Group.

All the details and settings are held in the Project file, which governs how every member of the Project Group behaves. Whichever instance of the Compere software takes control (the one assigned as server), like a stage manager, coordinates and distributes this set of stage instructions to all instances of Compere, which includes every Juggler in the Group. If any of these members, as clients, wants to make changes at any time, the changes are handed back to the server to redistribute around the Group.

Any client joining the Project Group is handed the common current Project file and performs as directed. In this way any PC running Compere on the network can join the Project Group, become the server or stay as a client and interact with the rest of the Group. This might be to make static changes in the 'stage set' or to move things around dynamically, such as picture-in-picture elements.

If network connection is lost by any Juggler in a Project Group, the local copy of the Project will keep running until it can reconnect and re-sync with the Group, at which point any changes to the Project file will be received from the assigned (or a reassigned) server.

Simultaneous Editing of a Project

Any instance of Compere can edit the Project configuration, save it locally and then redistribute it to all Compere clients. This is a powerful feature of Compere, giving multiple users the potential to accomplish large-scale tasks much faster. A good example of this would be one user using an instance of Compere to warp one side of a projection surface while another, using a separate instance connected to the same project, warps the other side.

The wider network

There can be more than one Project Group, each with an assigned server to coordinate it. They can all be on the same network, and any member can be moved from one Group to another as required. If you need other Groups, they can be created and named, and available members assigned to them instead of the initial default (unnamed) Group.

For more about the system architecture of Juggler systems, see:

➤ [Operating Juggler 1 Pixel Processor](#)

Workflow: Juggler in Compere

This is a generalised workflow, typical for [input bus type](#) ⁽¹⁰²⁾ systems. For output canvas bus systems, see [Output Canvas Bus](#) ⁽¹¹⁰⁾ and using the setup wizard.

➤ See [Terms and Definitions in Compere](#) ⁽¹⁶⁾

The precise workflow sequence will depend on the requirements of the overall system of Juggler modules, but this outline gives you the sequence of steps.

Assemble your Project Group in Compere

1. With Compere installed on a [remote \('control'\) PC](#) ⁽¹⁵⁾, log into the network.
2. Open Compere and make sure it is in [Juggler Layout View](#) ⁽²⁰⁾ (top left icon), go to *Configuration > Network Configuration* ⁽⁴⁹⁾. You should see all Jugglers and any other devices running Compere.
3. In Network Configuration, ensure that all the devices running Compere that are involved in your Project, belong to the same [Project Group](#) ⁽⁵⁶⁾. Create your Project Group(s) here as necessary.
4. Assign one member (Juggler) of the Project Group (any) as server, with a binding NIC, and all the rest as clients.
5. All Jugglers in the same Project Group as you as the Control PC, will now be visible in the Project panel on the left of the Compere screen.
6. Drag each of these Jugglers from the Project Panel into the [Nodes assembly panel](#) ⁽⁷⁹⁾ and arrange.
7. Connect the [bus nodes between the Jugglers](#) ⁽¹⁰²⁾ in Compere.

Assemble your display output

Note that if your Juggler configuration is for MPCDI display input, Canvas, Outputs and Windows will be imported with the data and do not require intervention.

8. In the Compere [Canvas assembly panel](#) ⁽⁶⁰⁾, create your overall canvas to size. The Canvas components all need to be created here or, if available as MPCDI, this file can simply be dragged from the Project Panel into the Canvas assembly panel.
9. In the Canvas panel, add and arrange your [Canvas Outputs](#) ⁽⁶³⁾. These will adopt the resolution of the Juggler Tx ports to which you attach them in the Nodes panel.
10. Assign the Canvas for the Project Group to each Juggler.

Configure your Juggler input and output connections

11. For each [Rx connection](#) ⁽⁹⁹⁾ available per Juggler (the number is the channel, 1 to 4 per Juggler), make sure the correct EDID is present for the graphics card feeding each Juggler input.

12. For each [Tx connection](#)⁽⁹⁹⁾ available per Juggler, the EDID should be autodetected from the output device. If not going directly to an output device, select and apply an EDID from the connection's Properties panel.
13. 'Windows' in the Canvas Outputs are areas in the whole Canvas to which Juggler inputs will be sent. [Add and configure](#)⁽⁶⁷⁾ your Windows, and name them for easy identification.
14. Make connections between Juggler [Inputs to Canvas Outputs and Windows](#)⁽¹¹⁹⁾ in the Nodes area, either directly through a Juggler, or from the input on one Juggler to the output of another via the data bus, and/or if using MPCDI data, introducing processing with warps, blends and colour convergence.

Juggler Software and Firmware

Compere is already installed for the internal operation of each Juggler on Linux. It handles all system communications including the front panel display. To configure your system over the Juggler network, install Compere on a local PC on the same network.

Versions

Juggler projects require compatible versions of the Compere software on all units in a project. If incompatible versions are detected in a Project Group, these will show in the Network Discovery Panel.

The current version of Compere shows in several places: on a Juggler front panel display, in the Compere UI title bar, and in *Help > Show About*. To see the version of a remote Juggler on the same network, select it in a Compere Project Panel and find 'Compere version' in the Properties panel.

Upgrading Compere on Jugglers

You may need to install or update a (Linux) version of Compere on a Juggler or many Jugglers in a Project Group. This can be done remotely from a control PC to distribute the installer. In Compere on the control PC, select the required Jugglers in the Nodes panel. Right-click on one and select *Admin > Upgrade Compere* (or Ctrl+Alt+C). Browse to the Linux installer on the control PC and click on 'Open'. This will distribute the file to all the Juggler internal controllers. Progress can be seen in the Compere Task Tracker icon or in the tracker panel (select main menu *Panels > Task Tracker*). Once all tasks have completed 100%, the selected Jugglers can be rebooted remotely in the Nodes Panel from the same right-click menu (*Power > Reboot*) and will be upgraded.

Juggler Firmware

Juggler is a highly configurable item, yours will carry firmware with some specific original configurations to match your requirements. Firmware updates may be available from time to time to improve performance, not in Compere but in the FPGA processor. The update process is very similar to upgrading Compere, but the file you will have been provided with will be called BOOT.bin. Progress of firmware uploading can also be seen in each Juggler:



Wait for all tasks to be complete 100% before rebooting.

System maintenance

Several operations are available from a control PC running Compere, by selecting a Juggler or Jugglers in the [Nodes](#)⁽⁷⁹⁾ Panel of Compere and using the right-click context menu. The same three options are available on right-click of a Juggler in the Project panel.

Uploads

Upload EDID (Shift+E)

Upload an EDID file to become available to this Juggler.

Upload Script (Ctrl+Shift+E)

Example: IntelligentSource (Python) script.

Admin

Show Juggler Console (Shift+C)

Admin access to the Juggler TCP/UDP (service/support) console.

Update Firmware (Shift+F)

Juggler firmware determines the configuration and functionality of each unit. Improved versions may be required, or older versions brought into line with newer units.

Juggler Console Mode [Disabled, Enabled, Enabled (errors only)]

Normally disabled by default. Reduces performance when enabled. Admin / developer use only via the UART port.

Upgrade Compere (Ctrl+Alt+C)

Distribute and install a Linux version of Compere to selected Jugglers.

Power

Reset

Temporarily remove a Juggler from the Project, reboot and and reset it to clean state.

Reboot

Restart the Juggler.

Shutdown

Turn the Juggler unit off. (Cannot be subsequently turned on remotely.)

Compere UI for Jugglers

Once Jugglers have been physically connected to each other, to the network and to related devices (see [Connect a Juggler Data Bus](#) and [Connecting a System](#))⁽¹¹⁹⁾ they need connecting internally, from inputs to outputs via whatever processes are required. This is all achieved using Compere's UI on a control PC, to create a complete 'stage-management' script that contains the connections, configuration properties, positions and arrangements of all the equipment and players that comprise what is called a 'Project'.

- See overview: [Juggler and Compere](#)⁽⁷⁾
- Important for new users: [Terms and Definitions in Compere](#)⁽¹⁶⁾
- [Keyboard Shortcuts](#)⁽¹⁸⁾

Project Group, Server, Clients

Active instances of Compere that have a shared involvement in a Project, must belong to a [Project Group](#)⁽⁴²⁾. This will include all the Jugglers, plus any control PC that plays a part in managing the Project. One, and only one, instance of Compere in a Project Group has the role of server to the rest as clients. This role includes owning the Project script, receiving any changes to this from clients, and redistributing any changes to the rest of the Project Group.

It does not matter which instance of Compere is the server, and the role can be handed to any Group member. If a server leaves the Group for any reason, the Project will keep running, but intercommunication stops and changes will not be redistributed until there is a Group server again. A group Juggler is always the preferred choice – a control PC does not need ever to play the server role. The function of the control PC is to be able to use the graphical interface to view and edit the Project Group and all its members and their roles, and/or to be used for external control.

- Special case: [PC as Project Group server](#)⁽⁴⁷⁾

The function of the Jugglers in a Project is to assemble one or more shared visual canvases between them, i.e. a combined total pixel space. These pixels may be displayed anywhere in the real world, and you will place individual inputs into the system onto any number of locations in this overall space. Each Juggler in a system has the advantage of handling a range of input and output connection standards, including DisplayPort™ 1.2, HDMI® 2.0 and 12G SDI, and when put on a common fibre-optic data bus they can share a single very large canvas.



Names: Almost every element of a Juggler system has a name in its properties. Plan a logical scheme of naming so that everything is easily referenced. Names are used as parameters in [External Control](#).

Network

To start, the network that the Juggler system and associated devices live on needs to be set up in Compere, grouping devices for a Project (see [Network Configuration](#)⁽⁴⁹⁾).

Canvas

You will now configure where the received (input) pixels will be displayed within the total output canvas 'map' available to the Juggler system. This is done in Compere's Juggler view, in the [Canvas assembly panel](#)⁽⁶⁰⁾. In the overall canvas you will define canvas outputs delivered by the output devices. Inputs are assigned to canvas windows anywhere within these areas. It won't matter if the window is spread over multiple outputs; all pixels in the total canvas are available to the Juggler system.

Juggler systems can accommodate multiple canvases.

Calibrated canvases can be imported as MPCDI files.

Nodes

The next task is to describe the connections of the Jugglers so that source devices address the appropriate inputs, and outputs are assigned and configured to the right display devices. You have wired the outside world; the inside world of the system is 'wired' in Compere. This is all done in Compere's Juggler view, in the [Nodes assembly panel](#)⁽⁷⁹⁾. Here you will add representations of all the Jugglers in your system, connect up any data bus that the system shares (see [Connect a Juggler Data Bus](#)), and connect media streams from anywhere in the system, insert media processes, through to display devices and areas anywhere in the system.

Media throughput: NDI® Monitoring

When everything looks right, you want real media to play onto your canvas. You might want to use simple colour or numbered media per channel to start with, to check all your flows. Compere also provides NDI Monitoring (an option in the properties of each Juggler). This enables you to visually trace media input, through outputs and into canvas windows in the Compere assembly panels. These should correspond with the real world displays.

External Control

Everything in a Juggler system, constructed using Compere, has its own unique label (UUID). Any of its components or properties, including position, size, opacity, visibility, projector beams etc. can therefore be addressed remotely using [External Control](#) commands and a show controller.

Installing Compere on a Control PC

Compere installs to C:\7thSense\compere.



Minimum requirements

A control PC to run Juggler need not normally be of particularly high spec., but memory and number of cores govern performance. For system configuration:

- Minimum quad-core, Intel® Core™ i5 processor
- 16 GB RAM (recommended)
- a dedicated GPU
- OpenGL 3.3 core support
- Vulkan 1.1
- Windows 10

As redundancy, a lightweight networked PC can hold the Project dataset and communicate with a Juggler system.

Terms and Definitions in Compere

We use a lot of terms that are specific to the Juggler-Compere world, so it's worth being familiar with them.



Tooltips are implemented throughout Compere, with longer descriptions for labels in panels and toolbars.

Term	Definition
Assembly panel	There are three panels for assembling systems on a grid: Canvas for the display, Nodes to connect Jugglers and processes, and Visualise to create 3D theatre spaces. Objects of any kind in an assembly panel are assistive representations; removing them from an assembly panel does not alter them or the way they interact.
Bus	The Juggler fibre-optic data bus connects Jugglers together to direct any input through any process to any other Juggler output in the system. The bus can carry individual media streams (input bus) or the whole system canvas (output canvas bus), and its overall bandwidth is dependent on the unit configuration.
Bypass	A media stream input is directed in the same Juggler to a matched-type and resolution output, requiring no processing.
Canvas	An array of pixels that is attributed to the combined resolution of number of physical outputs.
Canvas panel	The assembly panel containing the total available video space. This panel can contain multiple display canvases on which Canvas Outputs can be positioned.
Canvas Output	A section of a composited canvas that is delivered to an output (Tx) port.
Canvas Window	A source of video that can be positioned on a Canvas across one or more Canvas Outputs.
Clone	xml descriptor for a project entity, e.g. canvas+outputs+windows; projectors; Juggler configuration; 3D model arrangements. Contains all properties and can be saved and reused by loading into in any project. Unlike Presets, clones are saved without UUIDs.
Compere	The project management software and UI that unites Jugglers, Actors, output and input devices, media management, and timeline building and control.
Connector	A video interface point for a device displayed in Node view. May be of any type (e.g. DisplayPort, HDMI, SDI, or bus), input or output.
External Control	A system of commands to address and message any element in a Project.
Grid	A background scaling and positioning guide used in the Assembly panels: Canvas, Nodes and Visualise.
IntelligentSource™	7thSense automatic failover system.
Juggler	A Juggler pixel processor unit of any configuration of inputs and outputs. Represented in Compere as a Node featuring external connections and internal processing.
Juggler Group	All Jugglers on a common network available to a Project, whether assigned to a Project or not.
Layout	The selection of dockable panels within which a Juggler project is addressed. Users can save the Layout they create and prefer, or reset the UI to the default Layout. 'Views' are default task-oriented Layouts.
Logic Node	A type of node used to create logical processes between system elements.

Term	Definition
Mini Map	A small preview representing the current viewing position and enabling navigation, amongst all nodes available in the Nodes panel.
Network Client	Any network device running an instance of Compere that is not a Network Server.
Network Discovery	The tab in the Network Configuration panel that lists and profiles all available Compere hosts on the network.
Network Server	Network device assigned as communication 'project controller' across all devices.
Node	A connectable object in Compere that is used for creating a system with interactive processes. Jugglers, for example, are nodes, and there are source nodes, value nodes, operator nodes and more, that can be combined to create logical processes in a Juggler system.
Nodes panel	An assembly panel into which a collections of nodes are introduced, and connections and processes built.
Output Canvas Bus	(OCB) An alternative bus mode. The canvas with its outputs and windows is carried around the bus, rather than being assembled from independent Juggler inputs that individually can be carried on the bus.
Panel	A movable, dockable panel providing: <ul style="list-style-type: none"> • assembly (e.g. Nodes; Canvas; Visualise; Timeline) • a contents list (e.g. Project; Resources) • configuration options (e.g. the Properties panel populated by selection from a list; the Network Discovery panel) • information (e.g. logs; system config table; task tracking)
Panel Settings	Properties that define how the contents of an assembly panel are seen. Project render settings are shared by all project users, local render settings are independent.
Path	In Compere, a path is the essential route to any property of any component allowing it to be addressed, in order to obtain its value or to change it.
Pin	An internal connection on a Node to link it to another similar pin.
Preset	Presets capture arrangements and settings for components in a Project. When recalled to the same Project, nothing is added to the Project, but dimensions and locations etc. of existing project components will be set again to those of the saved Preset. <i>Example of use:</i> multiple arrangements of windows in a canvas or canvases.
Project	A single dataset of information for a system of Jugglers, connected devices and display configurations, shared and used by all Compere nodes. (Think stage, lighting and cast of performers with various roles.) The project dataset includes connections and configurations of all associated devices and assets. Every element in a Project has its own universally unique identifier (UUID). All project data can be stored and recalled, in a .prj format.
Project Group	A network Project Group comprises Jugglers and devices on the same network assigned to work together in a specific Project.
Selection Group	Project panel: associate project panel members you want to often select together, to see or edit properties common to all. For example, to edit common resolutions and EDIDs, control beams, test patterns, sync, etc.
UUID	Every element in a Compere Project is assigned a Universally Unique Identifier so that it can be addressed by any other part of the system, for example by external control commands.
Video Channel	Transport route through a Juggler. There are up to 4 video channels per Juggler.
View	The main toolbar View buttons present default Layouts that are task oriented. 'Juggler' is a useful Layout for setting up hardware input, output and data bus connections through to the output canvas. 'Visualise' is a useful Layout for setting up a theatre scene in 3D space.

Keyboard Shortcuts

Keyboard shortcuts are shown in menus and button dropdown lists, but some that you might use most often may be worth learning.

General

Mouse / key	Function
Left mouse + drag	Pan a whole grid and everything on it.
Left mouse + Shift + drag	Reposition and resize objects on a grid.
Right mouse click	Raise a context menu. There may be additional items relevant to the area or object clicked (shows a green border around the selected object).
Left mouse click	Select an item in a list (list panel type, menu, drop-down selector). In the Canvas panel, selects an object according to interaction mode.
Left mouse double-click	Select an object or part (shows an orange bounding box, populates the Properties panel).
Mouse wheel	Zoom in/out about the cursor point
=/- keys (main keyboard)	Zoom in/out
V	Bring all objects in an assembly panel into view.
Ctrl+S	Save the project
Ctrl+J	Select all Jugglers
Alt+F4	Quit Compere (standard Windows function)

Nodes panel

Key	Function
Shift+I	Import an MPCDI file
Ctrl+Alt+D	Clear temporary MPCDI data
Ctrl+J	Select all Jugglers
Ctrl+Shift+Alt+J	Calculate (red) bus links (same as toolbar button to show all bus links). Note that connections require a minimum zoom level to show, and must first be calculated.
H	Cycle display around revealed Juggler bus connections: All; Selected Jugglers only; None.
K	Cycle display around processing node links: All; Selected nodes only; None.
Shift+Alt+B	Reset the bus
Ctrl+X	Show / hide all text in Juggler nodes
Delete/Backspace	Remove link from selected pin

Canvas panel elements

Key	Function
M	Toggle interaction mode (None/Canvas, Outputs, Windows)
N	Toggle NDI in Windows
Alt+T	Toggle NDI transparent
Alt+M	Toggle the minimap
Shift+drag	Outputs and Windows: snap to edges
Shift+F	Toggle a Window to fill the parent Canvas
Ctrl + cursor left	Move selected Windows to align with left edge of Canvas
Ctrl + cursor right	Move selected Windows to align with right edge of Canvas
Ctrl + cursor up	Move selected Windows to align with top edge of Canvas
Ctrl + cursor down	Move selected Windows to align with bottom edge of Canvas
Shift + C	Centre a Window in the canvas
Cursor left	Move selected Window/Output/Canvas left one unit
Shift + cursor left	Move selected Window/Output/Canvas left 10 units
Ctrl + Shift + cursor left	Move selected Window/Output/Canvas left 100 units
Cursor right	Move selected Window/Output/Canvas right
Shift + cursor right	Move selected Window/Output/Canvas right 10
Ctrl + Shift + cursor right	Move selected Window/Output/Canvas right 100
Cursor up	Move selected Window/Output/Canvas up one unit
Shift + cursor up	Move selected Window/Output/Canvas up 10 units
Ctrl + Shift + cursor up	Move selected Window/Output/Canvas up 100 units
Cursor down	Move selected Window/Output/Canvas down one unit
Shift + cursor down	Move selected Window/Output/Canvas down 10 units
Ctrl + Shift + cursor down	Move selected Window/Output/Canvas down 100 units

Juggler admin

Key	Function
Shift+C	Show Juggler console window
Ctrl+Alt+C	Upgrade Compere version on a Juggler or Jugglers

Key	Function
Ctrl+Alt+E	Upload an EDID to a Juggler or Jugglers
Ctrl+Alt+F	Update firmware on a Juggler or Jugglers
Ctrl+Shift+E	Upload a script to a Juggler or Jugglers

Beam control

Key	Function
Individual	Off
Ctrl + R	Red beam off
Ctrl + G	Green beam off
Ctrl + B	Blue beam off
Ctrl + W	RGB beams off
All	Off
Ctrl + Shift + R	All Red beam off
Ctrl + Shift + G	All Green beams off
Ctrl + Shift + B	All Blue beams off
Ctrl + Shift + W	All RGB beams off

Key	Function
Individual	Off
Ctrl + Alt + R	Red beam on
Ctrl + Alt + G	Green beam on
Ctrl + Alt + B	Blue beam on
Ctrl + Alt + W	RGB beams on
All	Off
Shift + R	All Red beams on
Shift + G	All Green beams on
Shift + B	All Blue beams on
Shift + W	All RGB beams on

The Juggler View Layout

Switch on each part of your system of inputs, outputs and Jugglers in turn.

Open Compere on a Control PC and you will be in a new empty Project. We shall learn more about [Compere Projects](#) ⁽⁴²⁾ later; for now, get to know how Compere is put together.

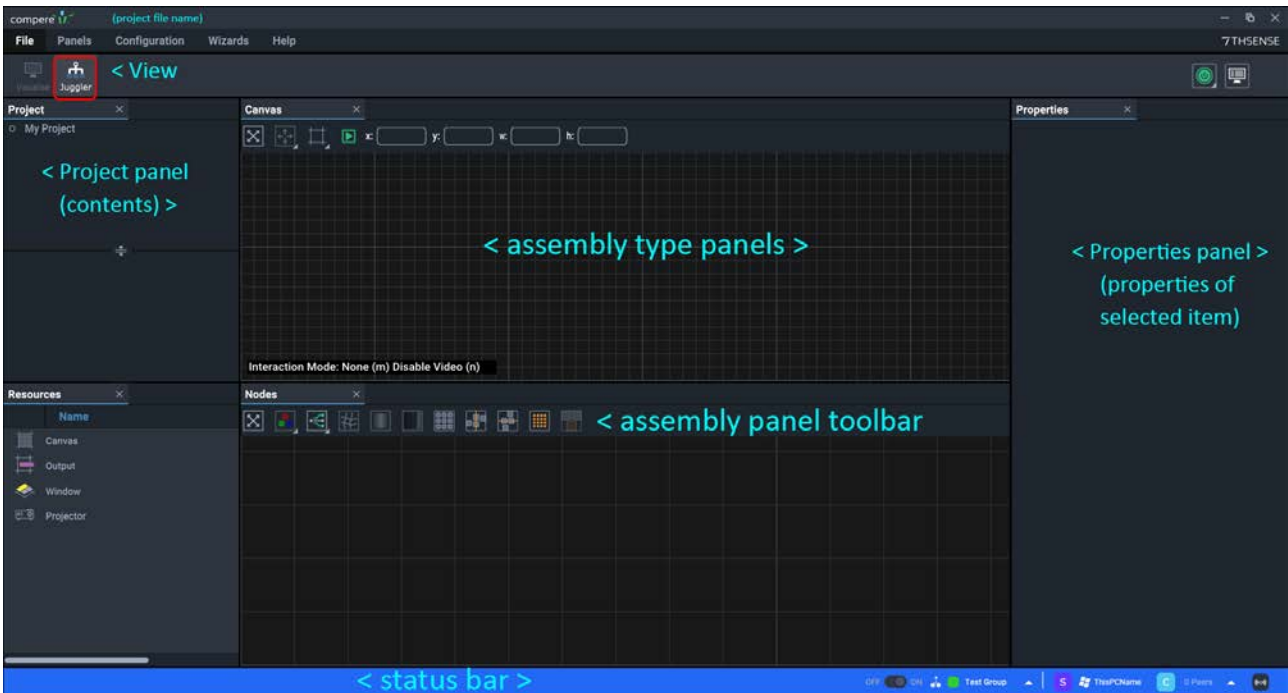
This guide will take you through the Juggler View of a project. If not in Juggler View, select the Juggler icon:



The alternative default View is 'Visualise', which introduces a 3D scene assembly panel for modelling a 3D space, with interactive projectors, displays and surfaces to configure. This is described separately.

Layout

The Juggler View is just a default Layout, or arrangement of panels in the Compere workspace. Note the default Nodes and Canvas assembly panels, and the sidebar panels. You may want to maximise the screen with the default overall Layout:



The initial default 'Juggler' Layout comprises:

- the main (top) menu
- the main toolbar area
- the 'Project' panel (default: upper left), listing all configurable elements

- the 'Properties' panel (default: right) for any selected element in a Project
- the 'Resources' panel (default: lower left), showing a range of asset types that can be added to a Project
- a grid 'assembly panel' with a tab marked 'Canvas'
- a grid 'assembly panel' with a tab marked 'Nodes'
- a coloured status bar.

If any of the panels is not present (you can close any element) they can be reopened from the top menu item *Panels*, or by restoring the default view, size and screen location from *Panels > Reset UI*. If you rearrange the parts as you prefer for different tasks, you can save and retrieve custom Layouts.

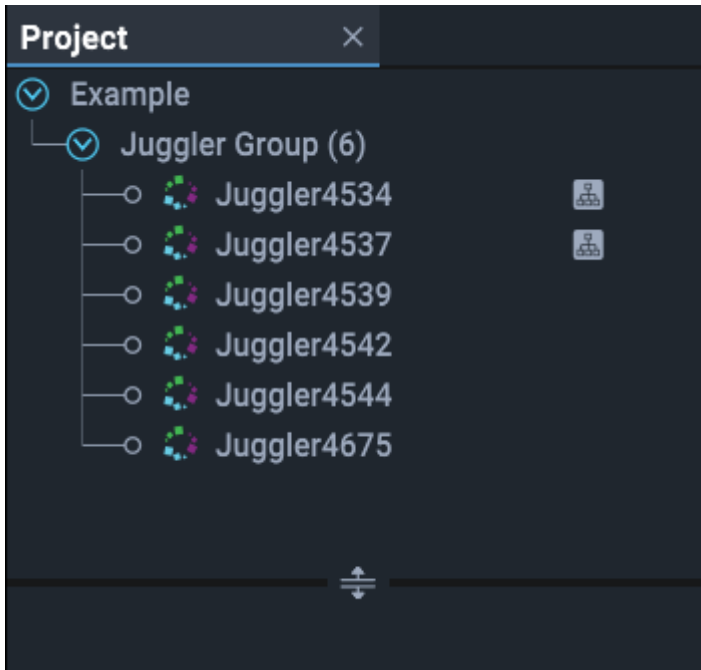
Project panel

The Project panel is a branching structure of all elements connected in a system, including Jugglers, input sources and output devices, 3D models, canvases and display surfaces. These can be dragged into assembly panels as appropriate.

When you start Compere for the first time, you will find a new empty Project (default name 'My Project'). You can use this, change the name, and save it. Projects are held as extensive xml files, by default stored in C:\Users\[username]\AppData\Local\Compere\Projects, with the extension .prj. This is configurable in *Compere > Preferences*.

The Project is where you will assemble everything required to 'set the stage'. Your Jugglers will appear in here when present in the same Project Group, as will your canvases, outputs, projectors, 3D models and any other component as you add them in.

When you join a network with the Compere UI on a PC, all Jugglers available on that network will be represented in the Project panel on the left side. They will be in a group under a title that shows how many are present.



This is just the list of what is available. Each can be named, examined for the properties of all its connections.

From here, you will select and drag Juggler names into the Nodes panel to further configure and connect them (see [Nodes: Connecting Juggler Systems](#)⁽⁷⁹⁾). Only Jugglers in the same [Project Group](#)⁽⁴⁹⁾ can be added to the Nodes panel, and those that have been added are identified by an icon to their right (as shown above).

When a Juggler is physically connected to display outputs and is switched on, a test pattern can be displayed. Until media inputs, internal pathways and outputs are defined, these will be the reassurance that your systems are connected. How these fill the screen is subject to application of EDIDs on the Juggler outputs.

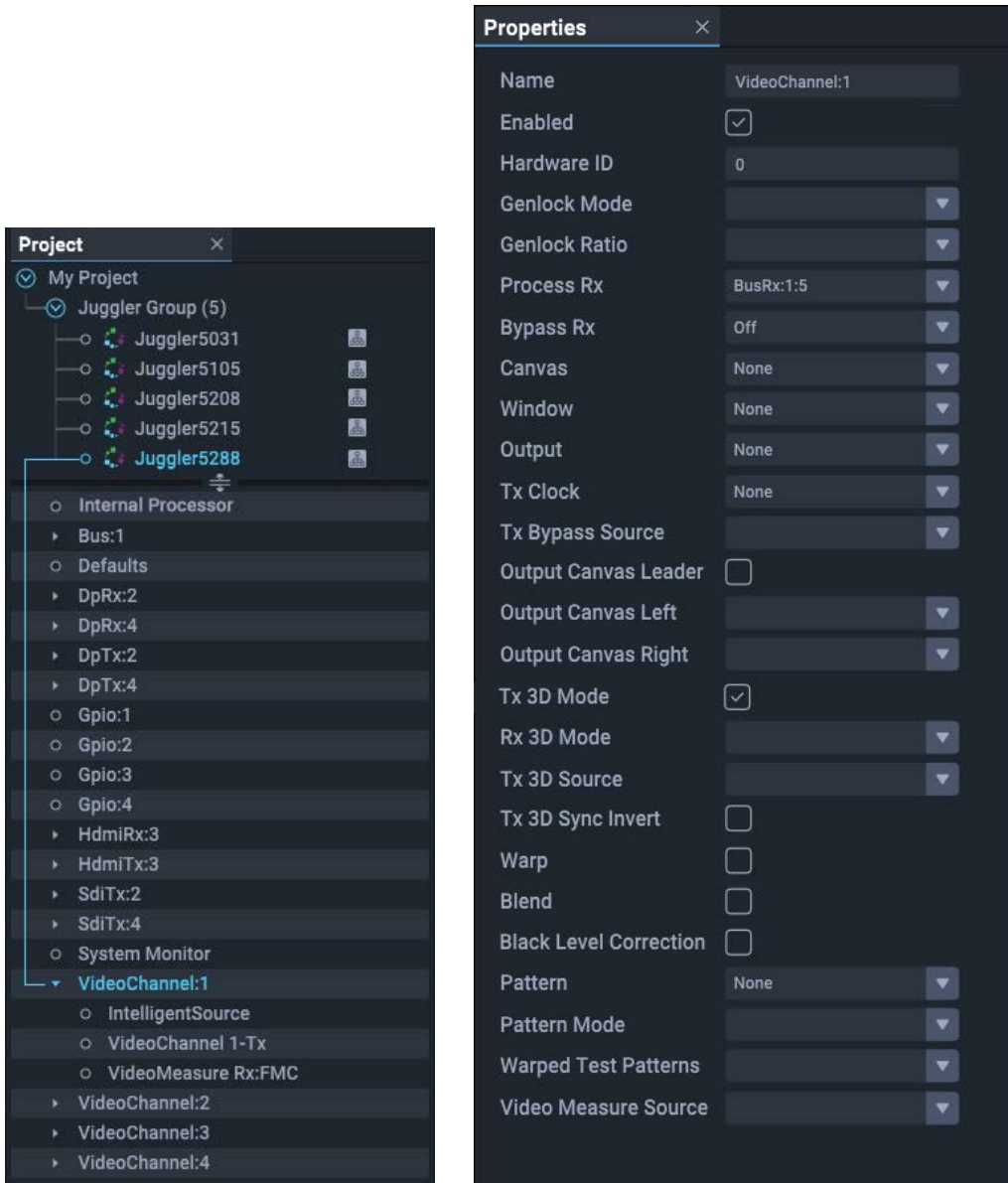
An instance of Compere will hold its own Project only when offline. As a Group client it will inherit the Group project from the Group server. As Group server it will distribute and assimilate changes made within the Group. When offline a project can be saved locally.

➤ For full details, see [Compere Projects](#)⁽⁴²⁾

Properties panel

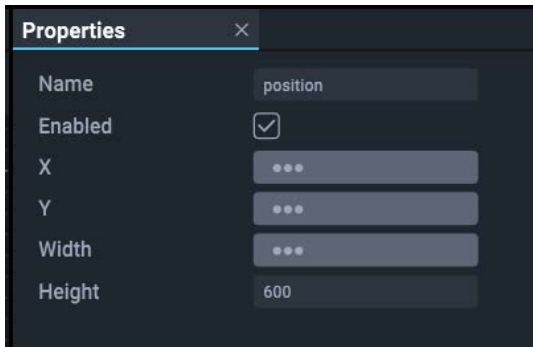
The Properties panel reveals all the features for any element in the Project. The Properties panel will always be showing the properties of the item or items highlighted as selected in the Project panel.

Members in the Project panel have their properties broken down by elements, for example a Juggler is broken down to each of its components and connections. Select a branch to see and edit the properties in the properties panel. For example, the properties of Video Channel:1.



Note: multiple items, such as Jugglers or Canvas Outputs, can be selected in the Project panel, but only shared common properties and values will be shown. Unshared properties will be greyed out and unshared values will show ellipsis (...).

Example: multiple windows that only have a shared height value will only show that value, which can be edited for all:

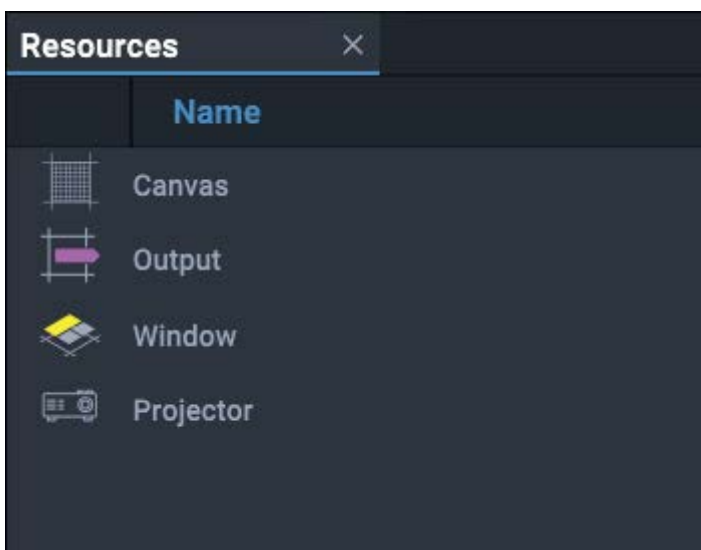


Addressing properties remotely

Hover over a property label in the Properties panel to see its full address. Shift+Right-click to copy this path string for use in [External Control](#)¹³².

Resources panel

The Resources panel contains internal Compere objects that can be inserted into the project, including Canvas, Window, Juggler node components, processing nodes and default projectors.



Using Resources: Add, Select, Edit, Remove, Delete

- Objects can be dragged from the Resources panel onto an assembly panel. This simultaneously adds them as branches of the Project panel.
- Objects shown in the Project panel, such as Jugglers and imported objects, can be dragged into an assembly panel.
- Objects are selected within an assembly panel to position, size or edit them, either by double-clicking them directly, or selecting them by name from the Project panel.

- When any object in an assembly panel is selected it will show an orange border, and its name highlighted in blue in the Project panel. Its properties are now shown in the Properties panel.
- To delete any asset, select it in the Project panel and press delete, or select it in the assembly panel and press Ctrl+Delete.
- To remove any asset from an assembly panel, select 'Remove' from its right-click context menu.
- Links between nodes and/or pins are deleted by selecting the receiving connector, and pressing delete.

Summary

- ❖ **Assembly** panels are a visual representation to aid system configuration.
- ❖ The **Project** panel holds all the members of the project with their individual configurations and links, whether displayed in assembly panels or not.
- ❖ Resources are **added** to a project by being dragged into an assembly panel.
- ❖ Resources are **taken out** of a project by being deleted from the Project panel.
- ❖ Resources can be **removed** from an assembly panel, but they remain fully **configured and interactive** within the project.
- ❖ Resources that have been removed can be dragged back from the Project panel into an assembly panel, where their configuration, links, etc. will be correctly displayed.

Canvas assembly panel

The **Canvas assembly panel** represents the total pixels being processed by an entire Juggler Group. In this overall space, Canvas Outputs (channels) are defined. Media inputs (Canvas Windows) are then placed anywhere in the overall canvas. The Canvas assembly panel appears in the default Layout for the Juggler View.

➤ [Display Configuration: Canvas Setup](#) ⁶⁰

Nodes assembly panel

This assembly panel creates all the throughput connections and processes within the Juggler system. Representations of each Juggler are shown here, with each video interface connector (bus and AV inputs and outputs), along with information on external connections and internal routing. Nodes is the assembly panel where you configure everything that comes into and goes out from each Juggler, and how these are connected and processed internally. NDI feeds are added and managed here. The Nodes assembly panel appears in the default Layout for the Juggler View.

➤ [Nodes: Connecting Juggler Systems](#) ⁷⁹

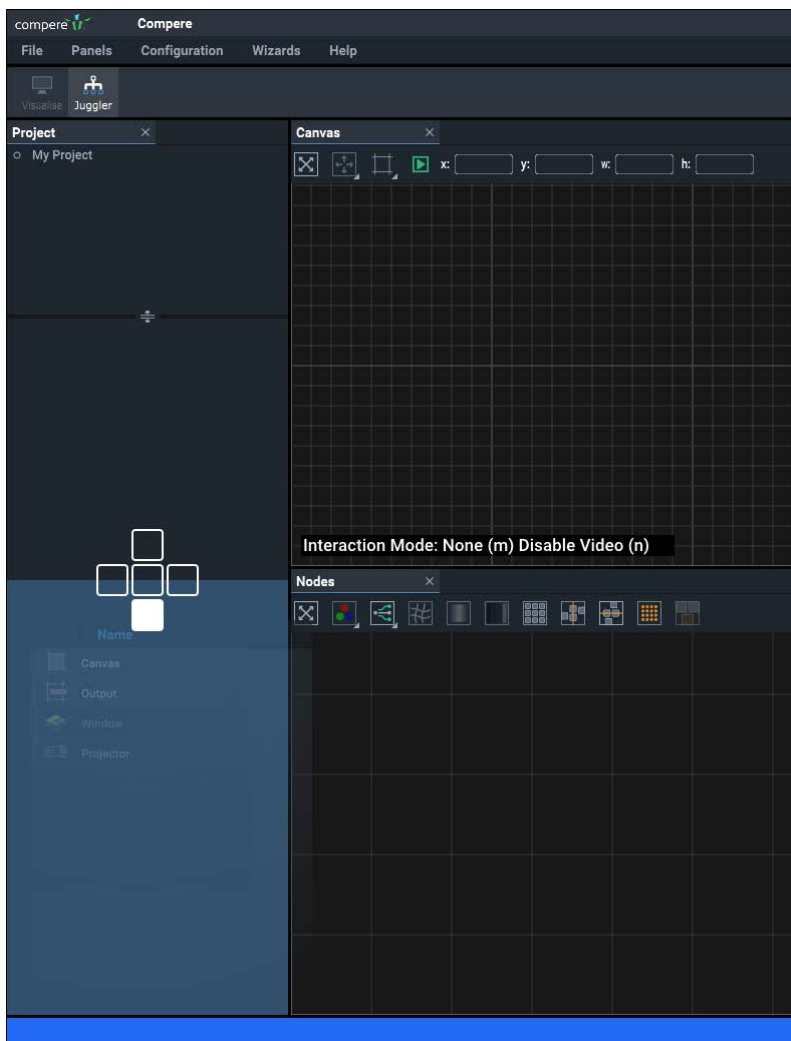
Using Compere Layouts

The Compere user interface is completely flexible. Panels can be resized, docked and nested however you want, or detached and floating, to work on multiple screens. They are docked in rows, columns or tabs (layered). Any item can be a docking space for others.

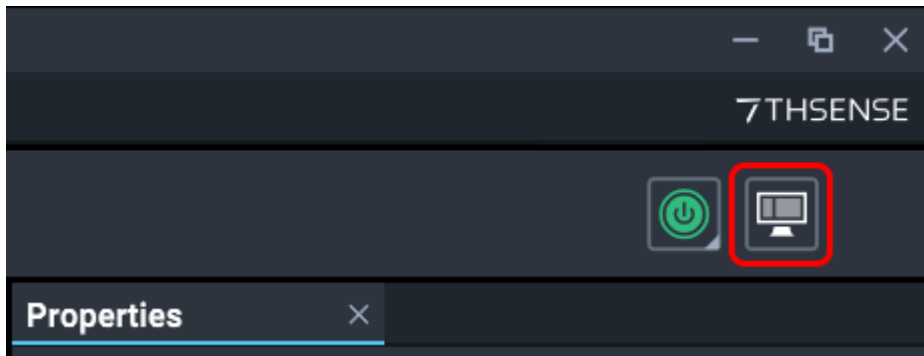
Any element of the Layout can be resized. When the whole space is resized, assembly panels (those with a grid) resize proportionately, whilst other panels do not. When more than one panel is in a row or column, they resize together.

Elements are positioned by dragging their title tabs.

- Click and drag a panel boundary to resize it.
- Click and drag a panel title to rearrange it. Dragging will bring up a cross pattern of white squares. As you drag across the screen, this pattern will relate to the vertical divisions of the default docked Compere Layout. Panels can be arranged horizontally or vertically in these zones, or layered (centre square). As the panel title is dragged over the pattern, the squares light up, and where the panel will be is highlighted. Drag free of the squares (no highlight) to float a panel. Here the Resources panel is being moved to dock below on the-left:



- Any panel can be closed, and redisplayed from the *Panels* menu item. If any panel is duplicated, these are just multiple instances, but can display different areas of the panel simultaneously. For example you may want to see different parts of a complex Project next to each other, or fill a separate display screen with a single assembly panel.
- Menu *Panels > Reset UI* will restore the default fully-docked arrangement of panels to the default Layout relevant to either Juggler or to Visualise View.
- To save your own Layout to use again, use the top right toolbar:



- Assembly panel grids can be dragged around to reposition the view, and zoomed with the mouse wheel. The Visualise assembly panel grid operates in a 3D space and can be rotated about 3 axes.

General

- To add items from a Resource panel, drag them into assembly panels or onto items already there.
- To remove items from assembly panels, right-click and select 'Remove'. Note that 'Remove' is not 'Delete', and that removing leaves configuration of properties intact. Deletion from the Project panel also deletes properties.
- To edit the properties of any resource, first select it in the Project panel to list its properties in the Properties panel.
- To delete a link in the Juggler Nodes assembly panel, select an end point of the node and press Delete.
- Assembly panels have their own toolbars, and right-clicking between and on items will raise a context-specific menu.

Main Menu

File Menu

Projects

New

Start a new Project; opens a dialog to name the Project. This is the Project name not the file name, which is given when saving a Project file.

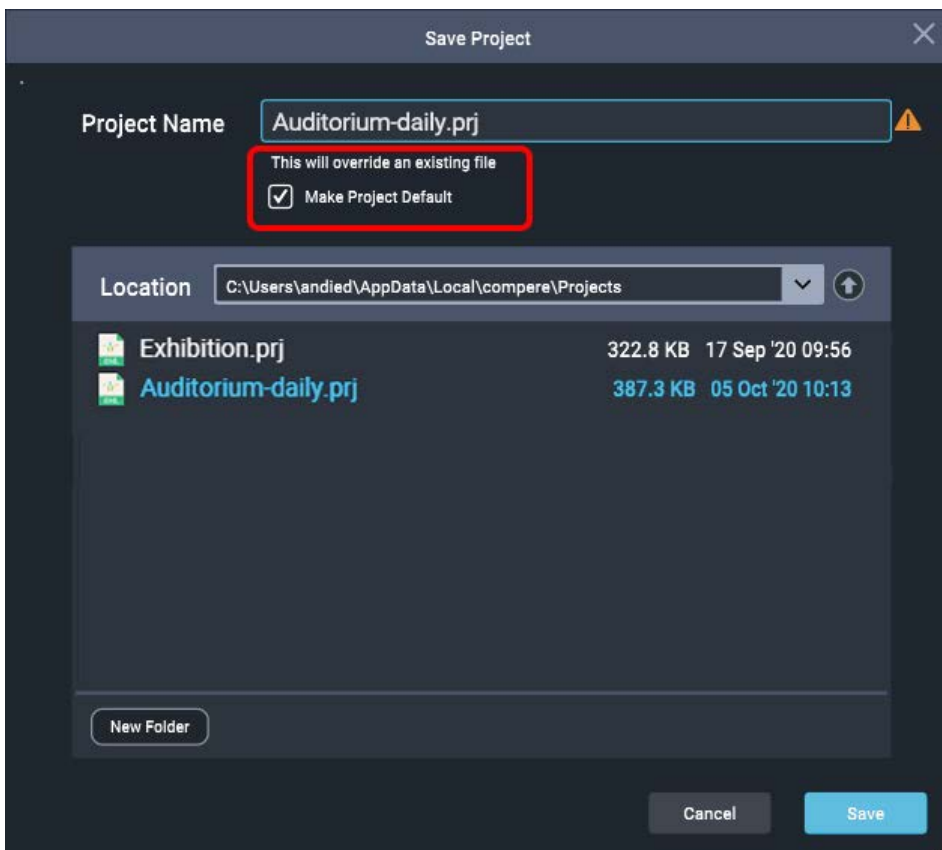
Open

Open a saved Project file (file extension: .prj). This will refer to a Juggler Group, members of which have been assigned to the Project Group, and to display configurations and all assets of the Project. Opening a Project file as a client, will receive the latest version from the Project server. Opening as the server of a Project will distribute the Project as the latest version to all clients.

Save As / Save

Enter a filename for your Project, and/or if you are the server, override the default location. For a client, the saved file will go to the Project Group server for redistribution.

Save As also allows you to assign this Project as default, so that Compere will open with this Project each time:



Saving to an existing file name raises a warning.

If you save a different Project as default, this will replace your previous selection. This is only really relevant for a Project Group server, or an offline instance of Compere UI (for example) assembling parts of a project remotely for integration later (see Save/Load Clone below).

Export

For a client, or a client not attached to a server, it can be useful to save a copy of a project to a different location, for example to send for support diagnosis.

Set Default Project

Allocate any existing project as default, to open with Compere.

Clear Default Project

If, having set a default project (as above under Save) you want Compere to open without a project running, this will remove the setting you made.

Presets

Recall / Save As / Export

Record multiple arrangements for the same project, typically arrangements of windows on and off a canvas for different display purposes. Properties for all or selected UUIDs are saved and recalled. Recall will replace property values for all matching UUIDs. The Export option saves the Preset file locally, not in the Juggler network. (file extension: .pre)

Save (any) selected elements only, or save the contents of the Canvas panel.

Apply Preset to Selection

Typically for configuring multiple Jugglers of the same kind from the saved properties of just one.

➤ See [Clones and Presets](#) ¹²⁹

Clones

Load / Save As / Export

Any entity in a Project assembly will typically be configured for the situation. All its properties can be saved for reuse. So if you are configuring a Juggler, or a projector, for example, select it in the Project panel and save it as a Clone. Similarly a canvas and its outputs can be dimensioned and arranged as a standard configuration. You can then load the same Clone file into another project where it will be added to the Project panel. The item can then be dragged, preconfigured, into your Project assembly panel (Nodes, Canvas, Visualise). The Export option saves the Clone file locally, not in the Juggler network. (file extension: .clo)

Save (any) selected elements only, or save the contents of the Canvas panel.

Quit

Close Compere (Project files are not auto-saved).

Panels Menu

Selected items can be used multiple times and are added undocked. Closing or undocking any these elements in a Layout does not lose any content. They can be used independently, for example on separate displays.

➤ See [Compere Layouts](#) ⁽²⁷⁾

Nodes

Add a Notes assembly panel to the Layout

➤ See [Node View: Connecting Juggler Systems](#) ⁽⁷⁹⁾

Canvas

Add a Canvas assembly panel to the Layout.

➤ See [Display Configuration: Canvas Setup](#) ⁽⁶⁰⁾

Project

Add a Project Panel to the Layout

Properties

Add a Properties Panel to the Layout

Logs

Open live log messages. See [Logs and Diagnostics](#) ⁽¹⁵⁰⁾.

Resources

Add a Resources Panel to the Layout

Task Tracker

Open a process progress [Task Tracker Panel](#) ⁽³⁶⁾.

Visualise

Add a 3D Visualise assembly panel to the Layout. This is a feature of Compere not covered by this Guide.

Reset UI

Restore the default Compere Layout (panel contents, arrangement and size) for the current View (i.e. Juggler or Visualise).

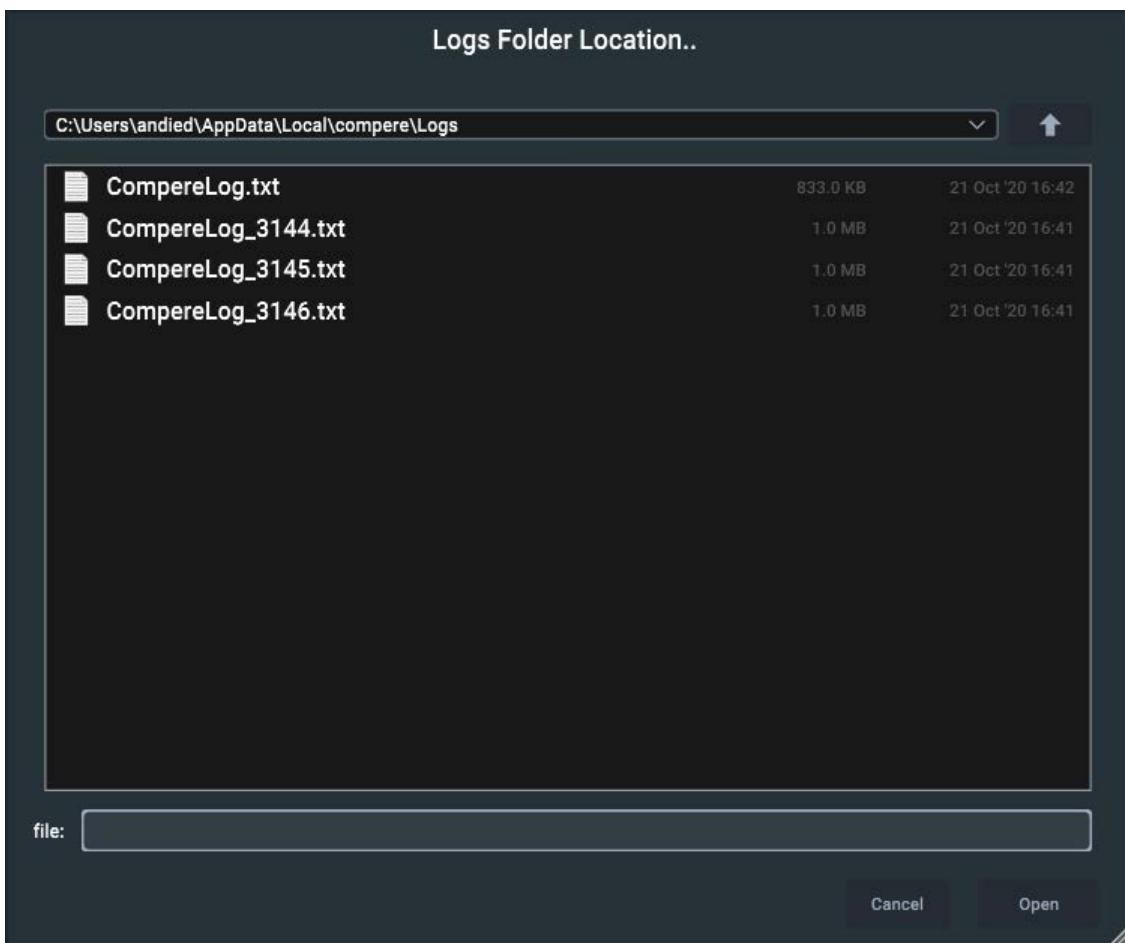
Configuration Menu

System Config Table

Displays a table of all Jugglers with their IP addresses, versions of software and firmware, port characteristics, and capabilities.

Gather Project Group Data

Lists logs per Juggler in the Project Group. Click 'Open' to create a zip file in this location of all logs and the local Compere Preference.pref file.



Network Configuration

Opens the Network Discovery and Project Group panel, for configuring a network of Compere devices.

➤ See [Network Configuration](#) ⁽⁴⁹⁾

Settings

Opens the Settings panel, containing Preferences for the current instance of Compere, and render settings for the appearance of each assembly panel (Canvas, Nodes and 3D Visualise).

➤ See [Settings Panel](#) ⁽³⁷⁾

Wizards Menu

Output Canvas Bus Wizard

The step-through assistant for creating all the essential settings for an Output Canvas Bus system.

➤ See [OCB Wizard](#) ⁽¹¹⁶⁾

Help Menu

External Control Protocol

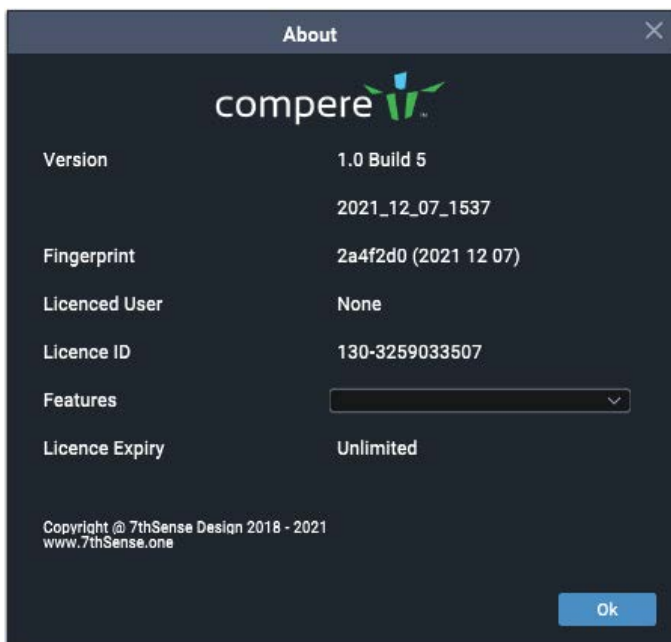
List [External Control](#) ⁽¹³²⁾ commands that can be sent to Compere to control it in various ways.

Licence Management

Shows permissions for licensed modules and users and allows licences to be ingested.

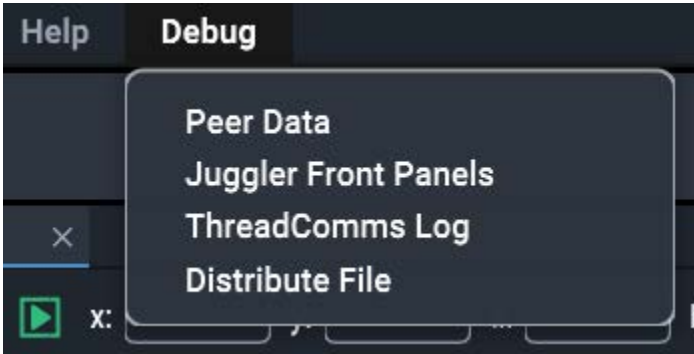
About

Shows the version of Compere on this device, included features, and licence details.



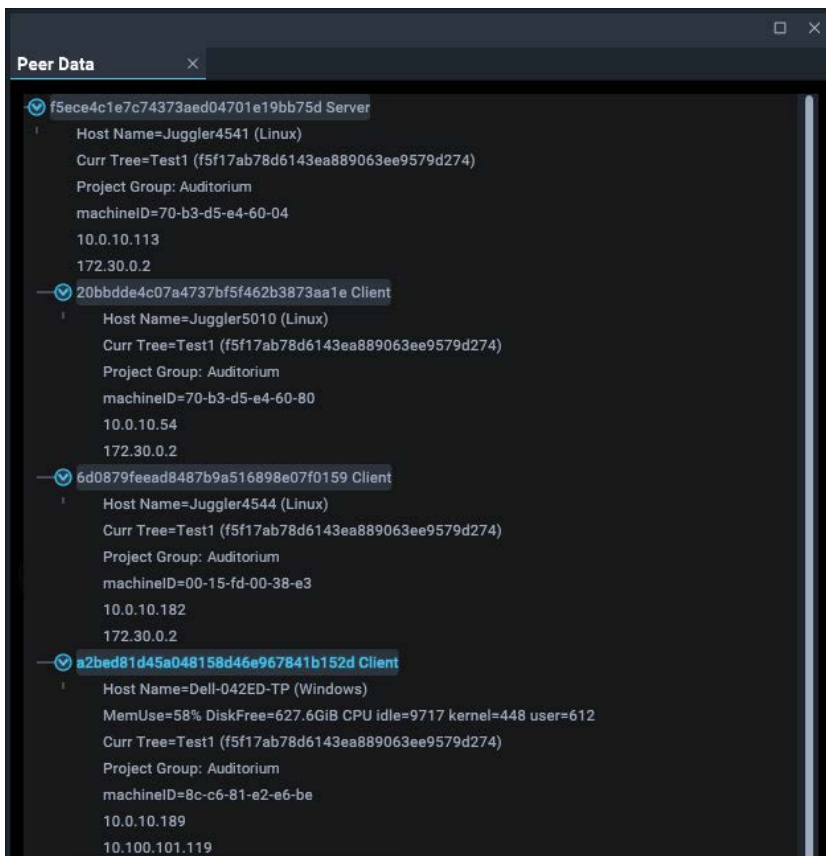
Debug Menu

By ticking 'Enable Debug UI Elements' in the *Configuration > Settings > Preferences* Panel, this additional menu is added. Its contents are for diagnosis and maintenance only.



Peer Data

Server-client data for all instances of Compere in the Project Group, including disk and memory usage, machine Ids and IP addresses.



Note the server (Juggler), clients, and the Windows PC.

Juggler Front Panels

Replicates a front panel display of the current Compere host.

ThreadComms Log

Extended system logging information that can be cleared or saved.

Distribute File

Used to select certain systems files to be delivered to all Jugglers in the Project Group.

Status Bar

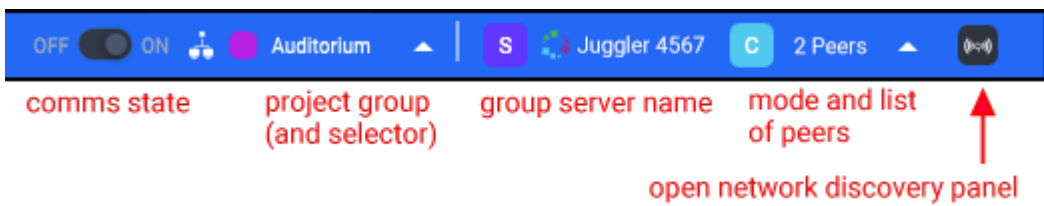
The status bar carries information about the current instance of Compere on the network. It has three main colour states:

grey: offline. Since it is disconnected, any information may be residual from the last online state.

red: online as a client but with no Project Group server present.

blue: online and in a Project Group with a server.

Project and Group information



In this example, Compere running this UI is in the Project Group 'Auditorium', with Juggler 4567 acting as group server to 2 client peers. The list available in 'Peers' shows all their names and IP addresses.

The symbol next to the server shows its operating system icon: Juggler (Linux), Windows or MacOS.

purple: A fourth colour state of purple indicates that a duplicate element (duplicate UUID) has been introduced. This would be a very rare occurrence, but is essential to resolve immediately and is an important diagnostic report to make. If this occurs, please contact support@7thsense.one.

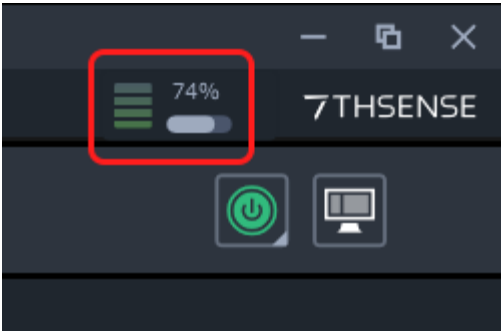
Status information

On the left side of the status bar, alerts, warnings and information can be found:



Task Tracker Panel

Whenever a system task takes place (the example here is importing an MPCDI file, but it could be a system file distribution, restore or update etc.), a small progress meter will appear on the right side of the main menu bar.



Its colour (red through yellow to green) will reflect current load, and by clicking on it, the Task Tracker panel will open, showing in progress, failed, and completed tasks. This panel is also available from the top menu *Panels > Task Tracker*.



There is a progress bar for each task and each subtask. Blue progress will change to green on completion. For longer tasks, click the arrow (here next to 'MPCDI') to expand the list for progress of subtasks.

When all tasks are complete, the list will clear and the task progress icon will disappear.

Failed tasks can be cleared by clicking the red button next to 'Overall Progress'.

Settings Panel

This non-docking panel is where local and system settings and preferences are stored. Panel render settings are the display properties of the Compere assembly panels themselves. This might be to see, or to set, how or from where the panel contents are viewed, what detail is included, or the zoom level. You can also choose whether to see the text labels of objects or not, to declutter a complex view. 'Render Settings' names can be changed if you want, which can be useful when using local settings.

These settings are shared by all project users, so if one user changes the view, it changes for all observers. Adopting local render settings enables the user to choose and keep an independent view of all the assembly panels *for this particular project*.

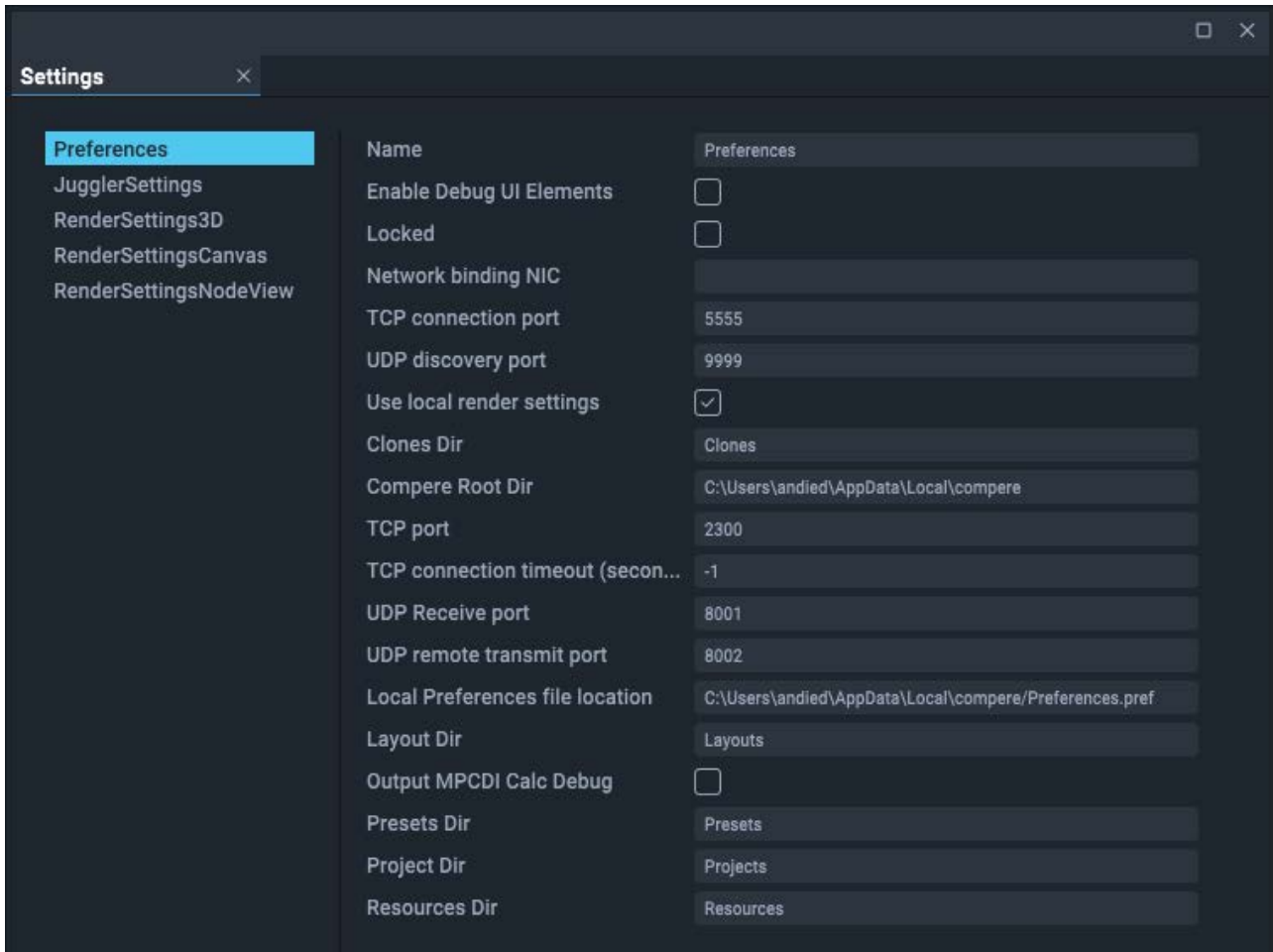
Shared settings are stored with the project, local settings are stored in the local Compere root, under \LocalRenderSettings.

To use local render settings, first check the option in *Configuration > Settings > Preferences* and then assert your own settings. Uncheck to revert to shared render settings, and check again to pick up your local settings.

- [Preferences](#) ³⁷
- [Juggler Settings](#) ⁴⁰
- [3D \(Visualise\) Settings](#) ⁴⁰
- [Canvas Panel Settings](#) ⁴⁰
- [Nodes Panel Settings](#) ⁴¹

Preferences

Menu: *Configuration > Settings > Preferences*



Use Local Render Settings

Compere allows users to retain all their own settings for how assembly panels are rendered, identified by the project, and stored in their Compere root folder, under \LocalRenderSettings.

UDP Remote Config Port

Remote discovery port: default 9103, configurable.

UDP remote Tx port

UDP port number to which external control will transmit.

UDP receive port

UDP port number external control socket will listen on.

TCP port

TCP port number external control socket will listen on

TCP connection timeout

Timeout on an existing TCP connection in seconds in which it'll be closed if no data is received. A negative value means there will be no expiry time.

Network binding NIC

Shows the IP address of the current instance of Compere.

UDP discovery port

UDP port used when discovering other Compere applications on the network (default 9999).

TCP connection port

TCP port on which connections will be made between instances of Compere for Project updates (default 5555).

External Control Adapter TCP Port

Port to address Jugglers on, default 5584, configurable.

Locked

Check to avoid accidental changes in the panel.

Enable Debug UI Elements

Implements an additional [Debug Menu](#)⁽³³⁾ for diagnostics.

➤ See also: [Ports Used by Juggler and Compere](#)⁽¹³¹⁾

File Locations**Compere Root Dir**

Compere data is stored in: C:\Users\[UserName]\AppData\Local\Compere

Project Dir

Compere Project (.prj) files are stored by default on the Compere server device in: [Compere Root Dir]\Projects

Resources Dir

Compere Project Resources are stored by default in: [Compere Root Dir]\Resources. Add subfolders as required.

Layout Dir

Compere Layout (.lay) files are the arrangement of panels saved by the user. Their default location is: [Compere Root Dir]\Layouts.

Presets

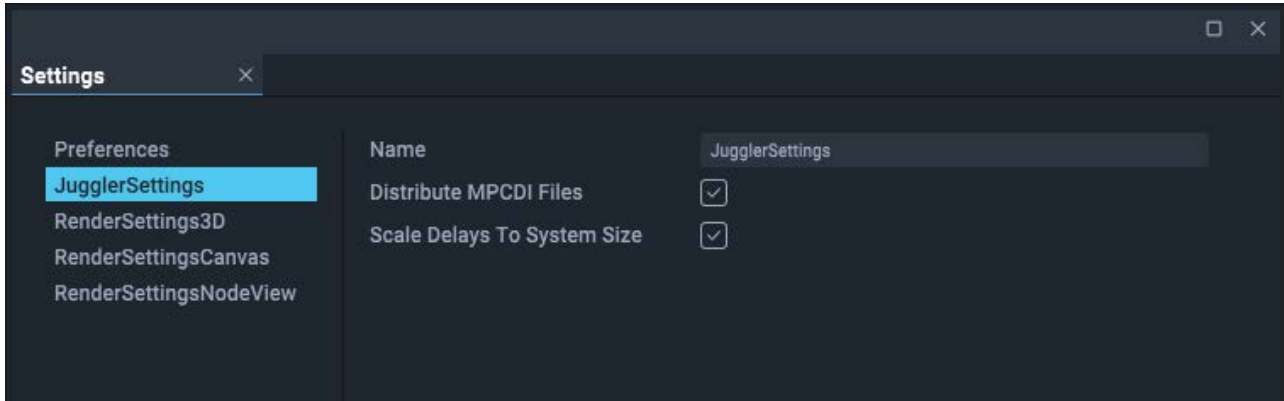
Preset (.pre) files are saved to: [Compere Root Dir]\Presets.

Clones

Clone (.clo) files are saved to: [Compere Root Dir]\Clones.

Juggler Settings

Menu: *Configuration > Settings > Juggler*



Distribute MPCDI Files

uncheck to prevent distribution of .mpcdi compressed source files by default (see note in [Display Setup via MPCDI](#))⁽⁷⁷⁾. Content extracted from the file will be distributed, just not the original single source.

Scale Delays to System Size

When creating an [Output Canvas Bus](#)⁽¹¹⁰⁾ using the wizard, it takes a few seconds to propagate all the settings. This increases with the size of the juggler system. For large systems, scaling the delay introduces a more appropriate wait time.

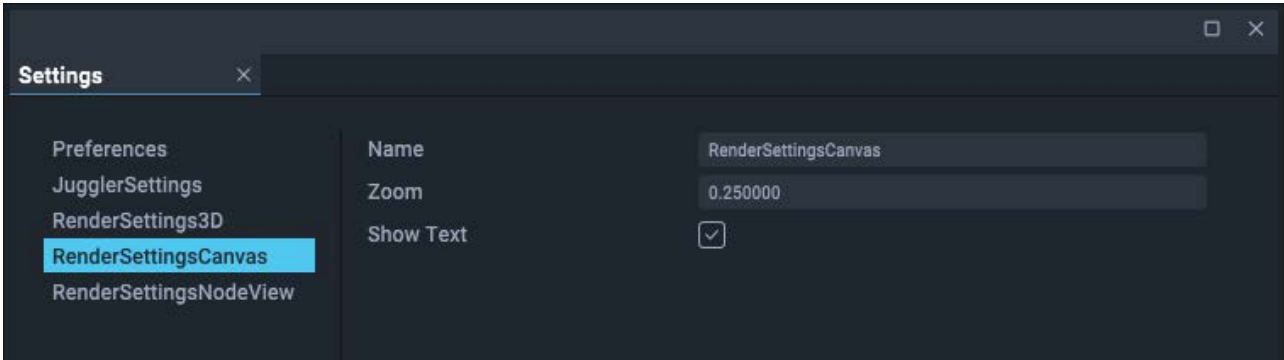
3D (Visualise) Settings

Menu: *Configuration > Settings > Render Settings 3D*

These settings relate to Compere Visualise, not covered in this guide.

Canvas Panel Settings

Panel settings determine the preferred appearance and/or behaviours when opened in the local instance of Compere.



Zoom

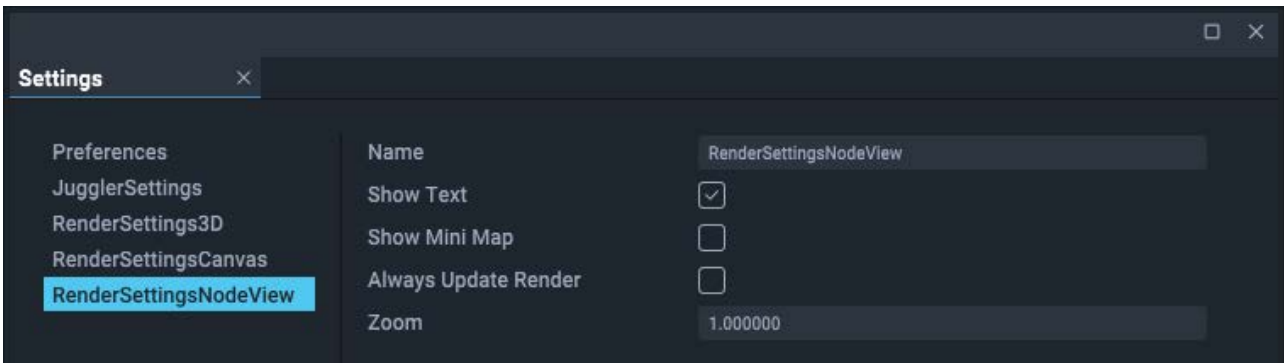
Shows the zoom ratio. Editable.

Show Text

Show or hide the text names of all Canvas objects

Nodes Panel Settings

Panel settings determine the preferred appearance and/or behaviours when opened in the local instance of Compere.



Mini Map is one feature that helps you locate the current viewing position within a large system of Jugglers. Also available from the toolbar button. Check / uncheck this box in the Nodes render settings (Show Mini Map):



The orange box contains all objects. The lighter area is the current view scope and can be dragged around to view the required Juggler(s) without having to zoom out or in to find your bearings.

Compere Projects

There is always a Project

Once Compere runs, on a control PC or as a Juggler, a Project is always present. This is true whether offline, or in the role of client or server.

- If this is an unconnected instance, Compere will, on opening, start a new empty project, or run a default project if one has been assigned.
- If the instance is a client or a server, is online, but is not a member of a [Project Group](#)⁵⁶ with a server, it will open as if unconnected.
- If the instance is a client, is online, and joins a Project Group with an assigned server, it will receive and run the Project of that server.
- If the instance is a server, is online, and is a member of a Project Group, whatever Project is running (new, empty or default) will persist and be handed to all online Project Group clients present or joining.

As the Project is assembled, all changes made from any instance of Compere are referred back to the Project Group server to redistribute to all clients. Changes to a Project when made offline as a server

are distributed on going online. If a client from a Group goes offline, edits the Project and then rejoins, its changes will be lost.

Saving and sharing, projects and parts

You can create and save as many Projects as you want in Compere. Parts of a Project (e.g. a configured Projector, Window, Canvas) can also be saved as Clones or Presets for reuse in Projects, so clear naming is important. Project (.xml) files can be saved locally, but can only be distributed among a Project Group by a server.

Edits to a Project by any member of a Project Group, whether in a client or server role, are collected and distributed to all others via the server.

Similarly, any device running Compere that joins the Project Group as a client will be given the Project file, displacing any other, so that it follows the same script as the 'stage manager' (server).

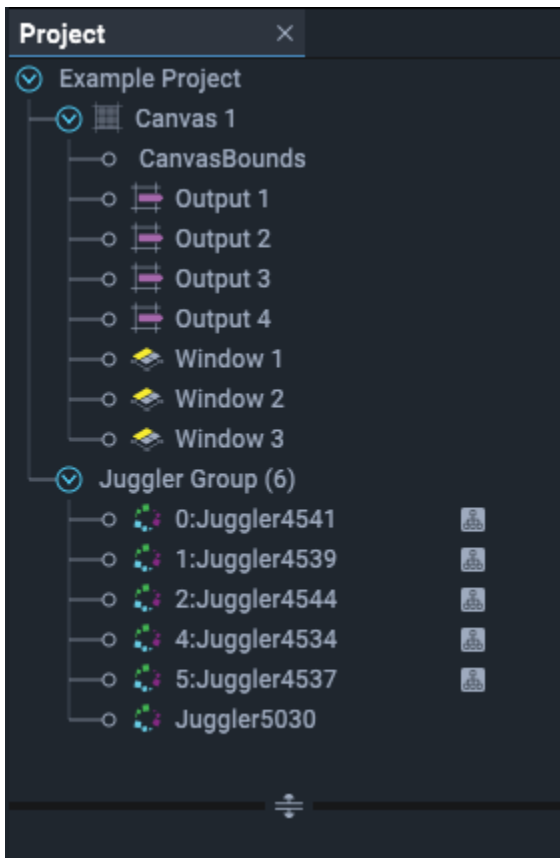
Server determines the project

Distributed files reside on all instances of Compere on the network (this includes every Juggler and control PC), so that every one has its own directions, and any one can be assigned the role of server. The server coordinates the distribution of the current Project to all clients, receiving edits from any client and distributing all revisions.

Compere can be used offline – as a client but switched to offline – or as a client without a Project Group server present. Project edits while offline will not be saved or distributed, but can be exported locally.

Project Panel

The Project panel on the left lists everything that takes part in a Project. Many items will have elements with their own properties, for example a Juggler can be expanded in the Project 'tree' into its branches of channels, inputs and output ports, and a Canvas can be expanded into its Canvas Outputs and Canvas Windows. When any branch of the Project is selected, it is highlighted blue and its top-level properties are shown in the Properties panel. All the internal parts of the selected object will be expanded below the sliding divider, from where they too can be selected for their properties.



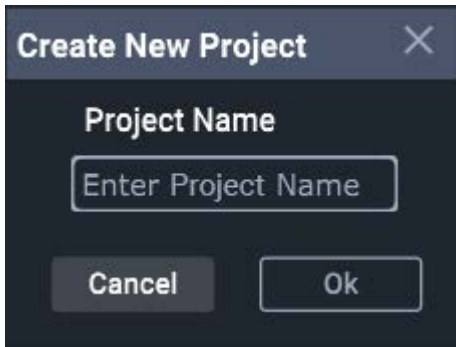
Major groupings (above, Canvas and Juggler Group) and the items in them, are arranged alphabetically, so they can be re-ordered by editing the names (above by adding numbers before Juggler names). Name is a property of any object.

Multiple objects can be selected in the usual way using Shift or Ctrl, and the properties panel will only show shared properties of common values. Unshared properties will be greyed out and unshared values will show ellipsis (...). However, other operations can be performed on multiple objects at once.

Project Properties: Name

The Name property is the only editable property of a Project and represents how the user knows the Project. This is not the same as the project file name. If you *File > Save As*, the project filename will change but the Project name does not.

When you start a new Project with *File > New*, you need to add a Project name:



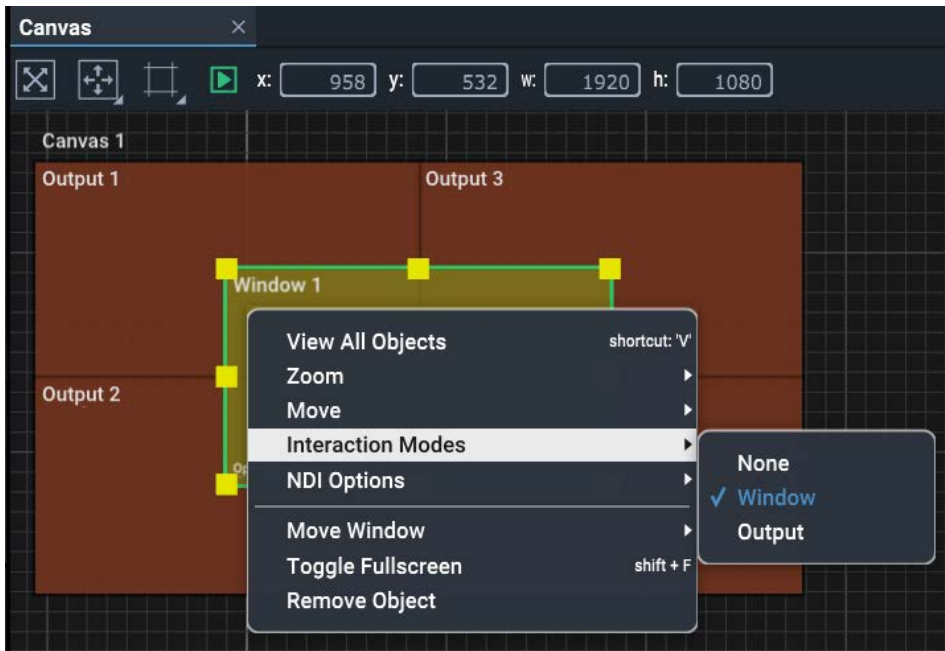
This will then show as the Project name in the Project panel.

The name can be subsequently edited in the Properties panel of the Project.

Assembly Panel Context Menus

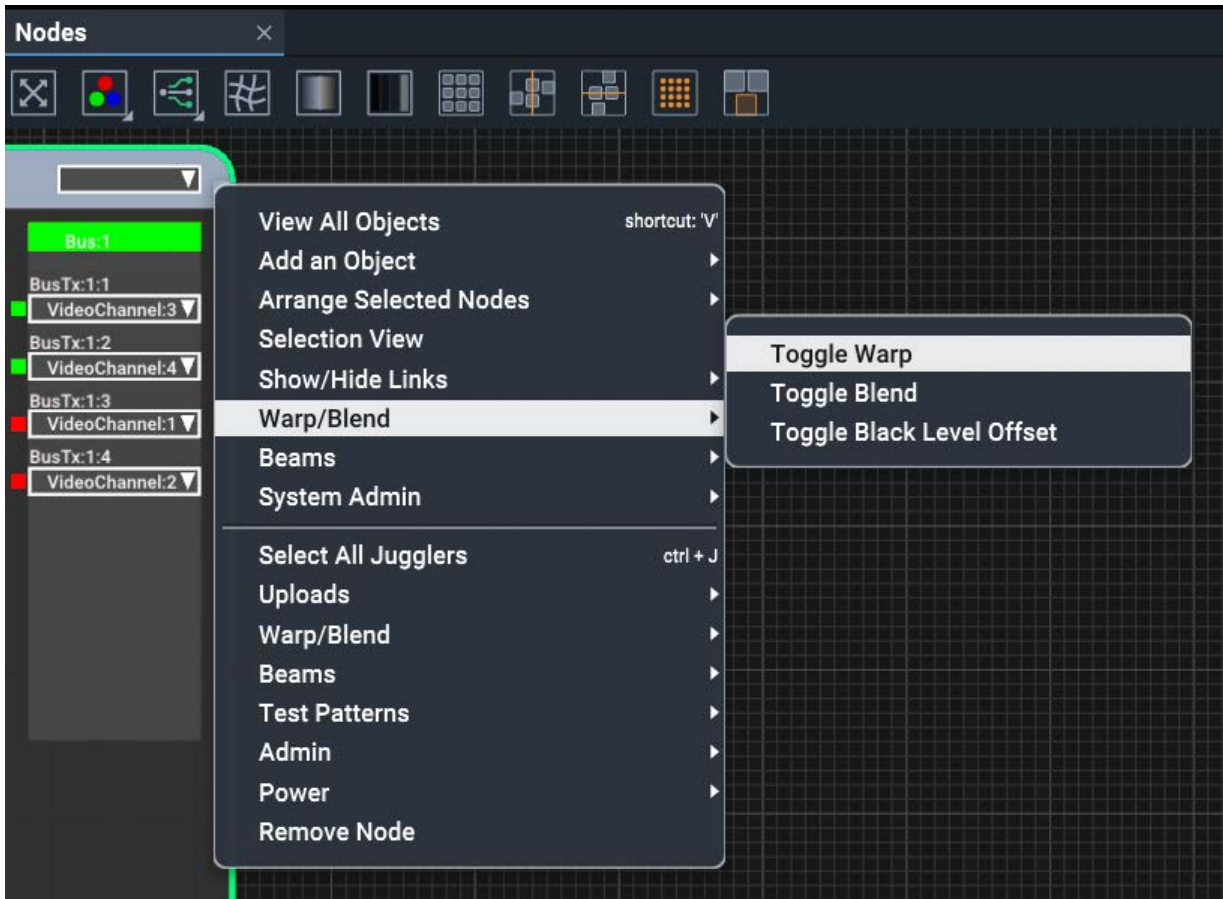
Right-clicking on an assembly panel background, or on a resource in an assembly panel, will open a contextual menu, many with shortcut hints. The object selected will change its border to green.

Example: a Canvas Window in the Canvas assembly panel, showing a submenu:



Contextual menus typically have two sections when called for an object rather than the containing panel. The upper section relates to the panel in general, the lower part to the selected object. In this example, the interaction mode is 'Window', so the two-part right-click menu contains generalised options such as Zoom, above object-specific options such as 'Move Window', which in this case has a further option menu. These last shortcuts are alternatives to using this menu, and are replicated as toolbar options.

Shortcut keys are themselves contextual to each assembly panel, though some, such as 'V' to view all objects, will be common. Context menus are also sensitive to which part of an item you right-click. Here is another example – Right-click a selected Juggler's title bar in the Nodes panel:



Some context menu items repeat the toolbar button actions. When the menu is divided, the upper section applies to all Jugglers in the assembly panel, and the lower to the selected Juggler only.

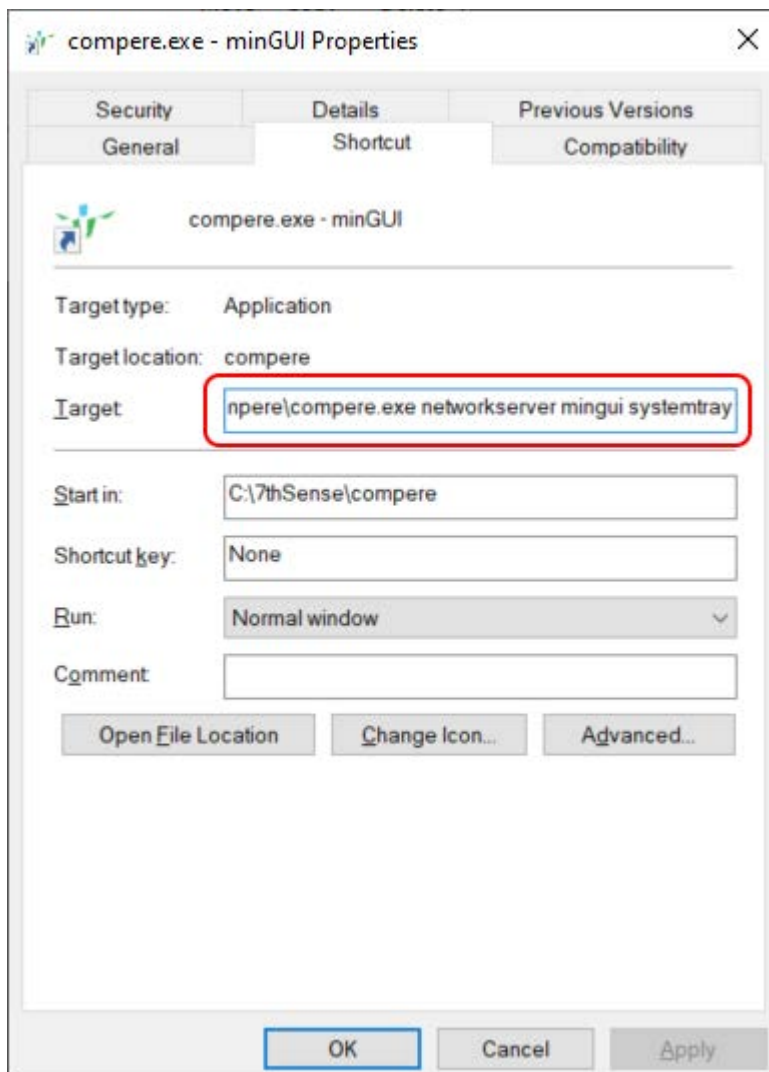
Compere MinGUI

Running Compere with the full graphical user interface and controls is CPU intensive. The Linux CPU in each Juggler runs a version of Compere without the UI, and also drives the front panel display for information and with a limited set of touch controls.

There can be a case for using a PC with a lower than recommended specification as a Project Group server, but not as a control PC with the full UI. The PC becomes part of the Project Group and is addressable by external control commands, just as any Juggler.

By running Compere with these additional arguments, it will run from the PC system tray. (Create a shortcut and add these arguments after the .exe command.)

C:\7thSense\compere\compere.exe [networkserver mingui systemtray](#)



Add a shortcut key or run at startup as required. Compere in its reduced form will be run from the system tray, right-click the icon to quit. There is no further user interaction.

Installation of Compere is otherwise the same as for the full user interface for a control PC:

- [Installing Compere on a Control PC](#) ¹⁵

Network Configuration

Compere Roles: Server and Client

Every Juggler is already running its own instance of Compere. Any PC added to the network, running Compere, provides a user interface (UI) into the network and can join a Project Group, but it does not need to remain present. Every Project Group member shares the same Project file and edits are sent back to the Project Group server to distribute. Any instance of Compere in the Project Group can be assigned the role of server and become the holder and distributor of the Project (as 'stage manager'). Any instance of Compere joining a Project Group will receive the server's Project file. Any edits made to the Project file whilst offline will be lost.

Each network Project Group looks after itself, so that if the server instance of Compere drops out another can be assigned the role. For this reason a control PC running Compere UI should not be assigned as server.

Jugglers or PCs running Compere in the Project Group can also be specifically addressed using external control commands sent from another PC to the Project Group server.

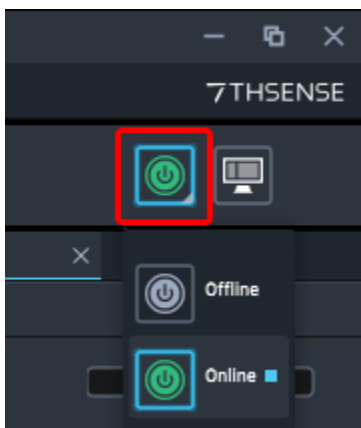
Project Group Server

In this role, Compere can connect to any number of clients but does not connect to other servers directly. A server shares its projects with clients and there can only be one server per Project Group.

Project Group Client

In this role, Compere can connect to any number of servers but does not connect to other clients directly. A client does not share its own projects, so any Project opened on a client will be dropped on joining a Group with a server.

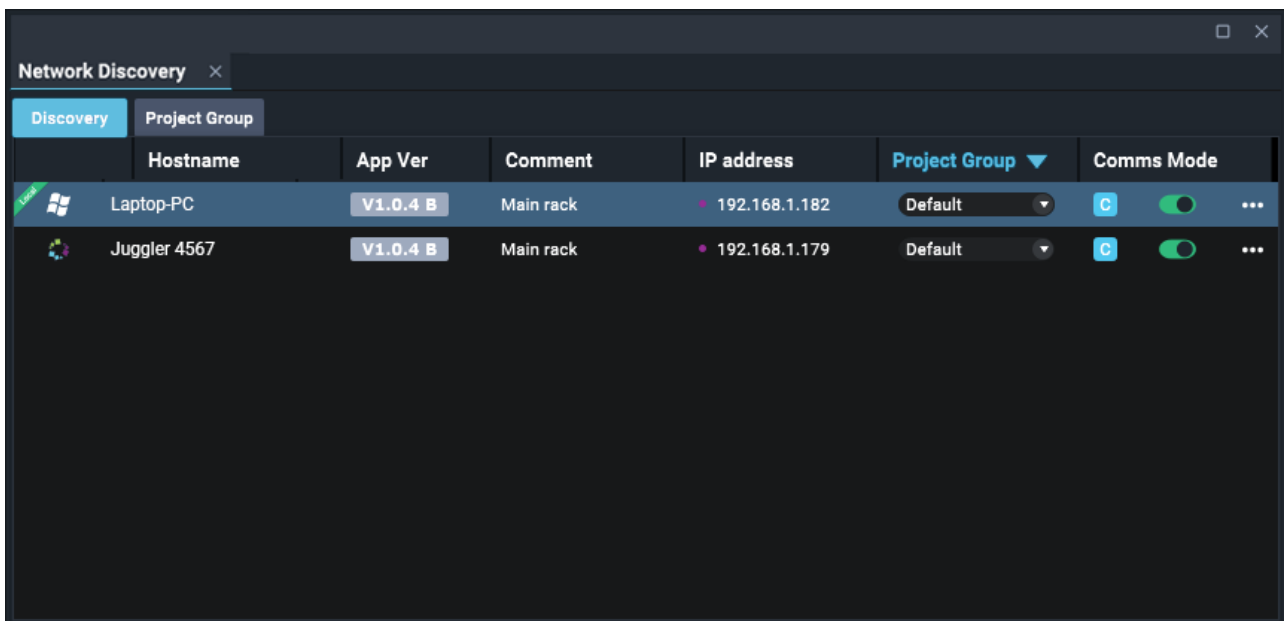
Every instance of Compere can be either online or offline. If online it can be acting as a client or as a server. This toolbar button/switch shows online status:



One instance of Compere (only) is assigned as Project Group server, all others are clients. If the server fails, a different client can be configured as the Project Group server. A control PC (e.g. a laptop) is able to assign any Juggler as server. The control PC can therefore be taken away or reinserted once the system is configured and running. Alternatively, redundancy or control bandwidth can be built into a system by dedicating a permanent control PC ready as server.

Configure the Network

Go to *Configuration > Network Configuration* where, in the 'Discovery' tab you should see all Compere devices (Control PC and all Jugglers) on the local network:



The screenshot shows a 'Network Discovery' window with two tabs: 'Discovery' and 'Project Group'. The 'Discovery' tab is active, displaying a table of discovered devices. The table has columns for Hostname, App Ver, Comment, IP address, Project Group, and Comms Mode. Two devices are listed: 'Laptop-PC' and 'Juggler 4567'. The 'Laptop-PC' row has a green 'Local' flash in the Hostname column. Both devices have 'V1.0.4 B' in the App Ver column, 'Main rack' in the Comment column, and IP addresses 192.168.1.182 and 192.168.1.179 respectively. The Project Group is set to 'Default' and Comms Mode is 'C' with a green toggle switch.

Discovery	Project Group	Hostname	App Ver	Comment	IP address	Project Group	Comms Mode
Local		Laptop-PC	V1.0.4 B	Main rack	192.168.1.182	Default	C
		Juggler 4567	V1.0.4 B	Main rack	192.168.1.179	Default	C

Your instance of Compere has the green flash across it marked 'Local'.

Discovery tab

Columns 'Hostname', 'IP Address', and 'Project Group' in this tab can be sorted by clicking on the column heads.

Hostname

This is the name that has been given to Juggler units (or PC). Initially these will be the names assigned when they were built.

App Ver

Different versions of Compere might conflict within a Project Group. This column shows the version on each instance of Compere in grey. If there are different versions of Compere running this may not present a problem, but a blue alert is given indicating that different versions are in play. (Note: if [Debug](#)³³ mode is enabled, the Compere version information is the more specific fingerprint.)

Comment

Edit this for clarification of location, role, etc., for your own use.

IP address

Shows the current IP of each unit.

Note: The internal IP range for all Jugglers is 172.30.n.n so do not use this range for your network of Jugglers and other devices.

Project Group

Which devices are going to share the Project file ('stage manager script')? Jugglers communicate by being assigned to a Project (Network) Group. Initially, all instances of Compere available on the network appear under a Group label 'Default', which is not a communications group. Project Groups are created in the other tab of this panel. Project Group is directly selectable here from the list of Project Groups you set up. Once in a named Group any instance of Compere can be put back into the Default pool.

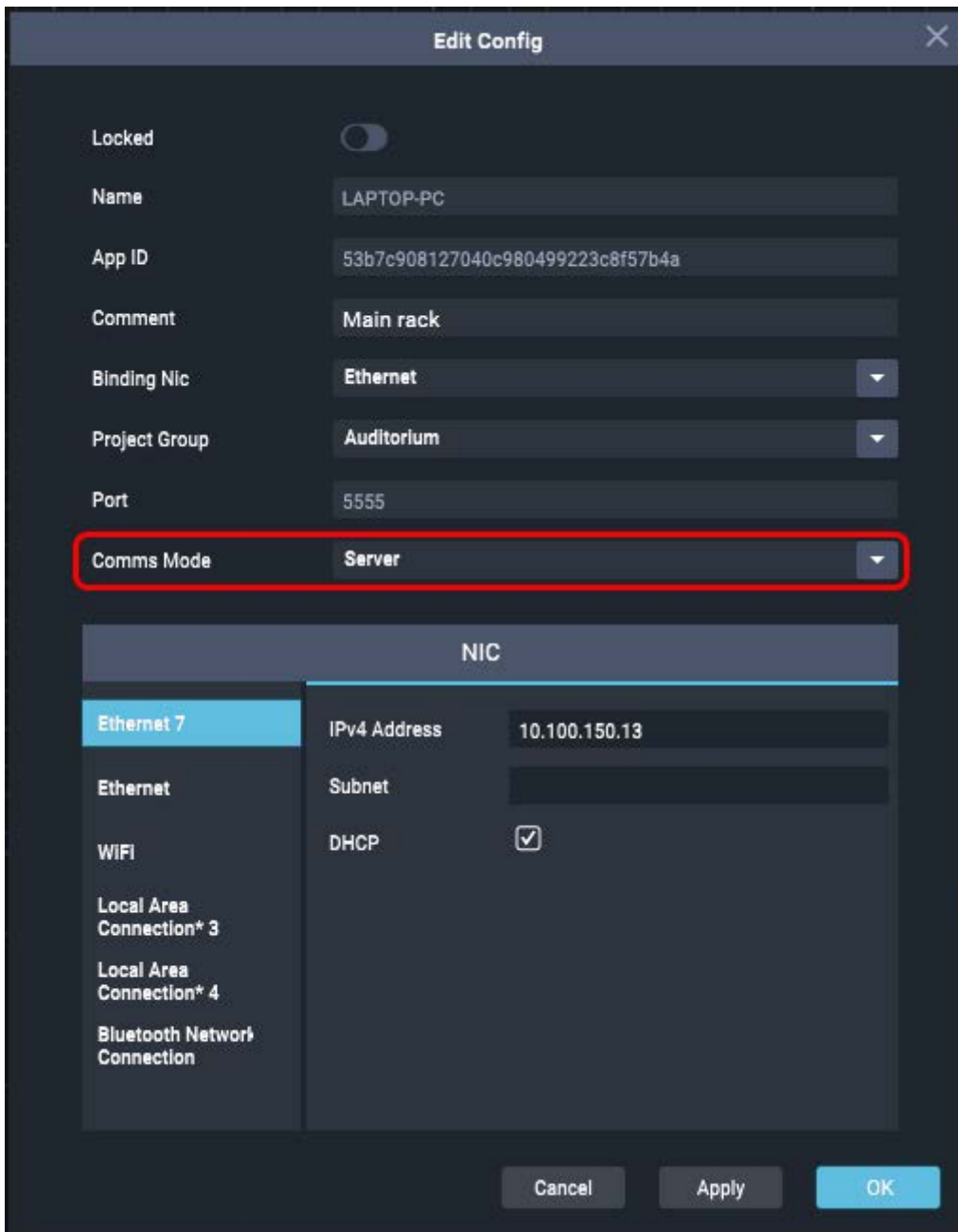
You must also belong to the same to the same Group as the Jugglers with which you want to share a Project.

Comms Mode

Offline, or online as client or server. The adjacent marker shows if online.

Click 'Apply' before any other changes, such as adding a Binding NIC. You may see an orange bar with the alert that 'Network Configuration is in a temporary state'.

Each Compere host shown here requires configuring, so click on the three dots to open the 'Edit Config' dialog. Here you need to edit the Comms Mode to assign a role (Off, Server or Client):



The screenshot shows the 'Edit Config' dialog box with the following fields:

- Locked:
- Name: LAPTOP-PC
- App ID: 53b7c908127040c980499223c8f57b4a
- Comment: Main rack
- Binding Nic: Ethernet
- Project Group: Auditorium
- Port: 5555
- Comms Mode: Server (highlighted with a red box)

The 'NIC' section shows the following details for 'Ethernet 7':

NIC	IPv4 Address	Subnet	DHCP
Ethernet 7	10.100.150.13		<input checked="" type="checkbox"/>
Ethernet			
WIFI			
Local Area Connection* 3			
Local Area Connection* 4			
Bluetooth Network Connection			

Buttons: Cancel, Apply, OK

You must have one instance of Compere running as server per Project Group. Where the role is server, there must be a binding NIC: in the lower panel you can see details of available NICs. If joining a Group with no server, there will be a warning message to alert you here, and in the Discovery list:

Edit Config

Warning: No available servers in your selected Project Group

Locked

Name: LAPTOP-PC

App ID: 53b7c908127040c980499223c8f57b4a

Comment: Main rack

Binding Nic: Ethernet

Binding nic not applicable unless in server mode

Project Group: Auditorium

Port: 5555

Comms Mode: Client

NIC

Ethernet7	IPv4 Address	10.100.150.13
Ethernet	Subnet	
WIFI	DHCP	<input checked="" type="checkbox"/>
Local Area Connection* 3		
Local Area Connection* 4		

Cancel Apply OK

Discovery	Project Group	Hostname	App Ver	Comment	IP address	Project Group	Comms Mode
		Laptop-PC	V1.0.4 B	Main rack	192.168.1.182	Auditorium	C
		Juggler 4567	V1.0.4 B	Main rack	192.168.1.179	Auditorium	S

You cannot have two servers on binding NICs in the same range, and an orange warning will appear to indicate any conflict. You cannot have two servers in the same Project Group. Trying to add a second will be prevented, with a red warning message displayed.

Locked

Secure your settings by checking this box.

App ID

Not editable: identifies this instance of Compere.

Comment

Adds a brief description to the Discover list.

Port

Default Compere comms port is 5555; edit another free port as required.

Do I need to create Project Groups?

No, if you do not need multiple Project Groups. The initial default is that unassigned devices will communicate with each other in an uneditable group called 'Default'. If you do not need multiple groups of devices, Default will be your Group. If you do create Project Group(s), make sure all Compere instances are assigned to created Groups, not the Default Group, and with one server role in each Group.

➤ [Create a Project Group](#) ⁵⁶

Once you have configured one instance of Compere as server, configure all the others as clients. The assigned role will remain with the instance of Compere, wherever it goes, until reconfigured.

The server-assigned Compere will hold the initial configuration for the Project. The Project file can then be saved locally and as server, it will be distributed to all other instances of Compere in the Project Group. In the event of losing a server connection, any Juggler can subsequently be handed the role of server and redistribute the project.

This example shows instances of Compere set up using Project Groups:

Discovery	Project Group	Hostname	App Ver	Comment	IP address	Project Group	Comms Mode
		Laptop-PC	V1.0.4 B	Main rack	192.168.1.182	Auditorium	C
		Juggler 4567	V1.0.4 B	Main rack	192.168.1.179	Auditorium	S

The Project Group members, and only the Group members, are shown in the Compere Project panel. Jugglers not in this Group will not be seen, and do not participate in the Project running in this Group.

Changing to another Project Group

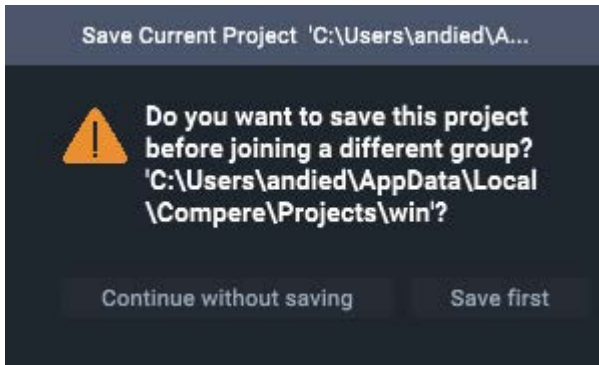
Role takes precedence.

- If the instance of Compere being moved to a Project Group is a client, it will adopt the Project being distributed by the Group server.
- If the instance of Compere is a server and is being moved into a Group without a current server, it will distribute the Project that it is currently running.

Before moving an instance of Compere into a Group, check first how you want to join – as server or client.

... as an online client

Your instance of Compere will be given the Project file of the Project Group by the server in that Group, on joining. You will see the alert regarding your current Project:



... as an online server

If you are joining a Project Group that has no current server (maybe it has failed or been withdrawn), you will be asked if you want to create a new Project. If not, then as a server your current Project will be distributed to the client Group, which will be reconnected.

You cannot join a Project Group that has a server, if your role is also server.

Lost Network Connection

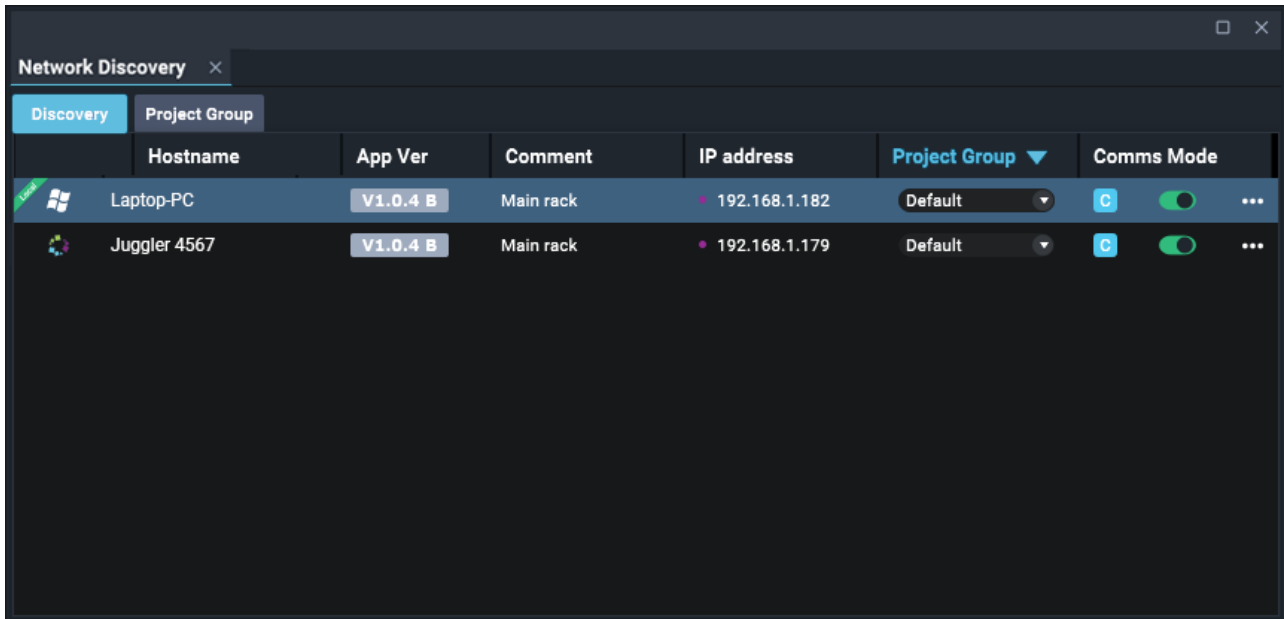
There must be an instance of Compere running as server in a Juggler system. If the server is disconnected or fails for any reason, the status bar of Compere on an editing PC will be red:



Clients continue to run with the Project but are no longer synchronised and no changes are exchanged.

Create a Project Group

Jugglers and devices on a shared network will be discovered by Compere running on the same network. In Compere on your control PC, go to *Configuration > Network Configuration*. This will bring up the Network Configuration Panel, which has two tabs: 'Discovery', and 'Project Group':



The screenshot shows a 'Network Discovery' window with a 'Project Group' tab selected. The window displays a table of discovered devices. The table has columns for Hostname, App Ver, Comment, IP address, Project Group, and Comms Mode. Two devices are listed: 'Laptop-PC' and 'Juggler 4567'. Both are running 'V1.0.4 B' and are in the 'Main rack'. The 'Laptop-PC' has IP address 192.168.1.182 and the 'Juggler 4567' has IP address 192.168.1.179. Both are assigned to the 'Default' Project Group and have their Comms Mode turned on.

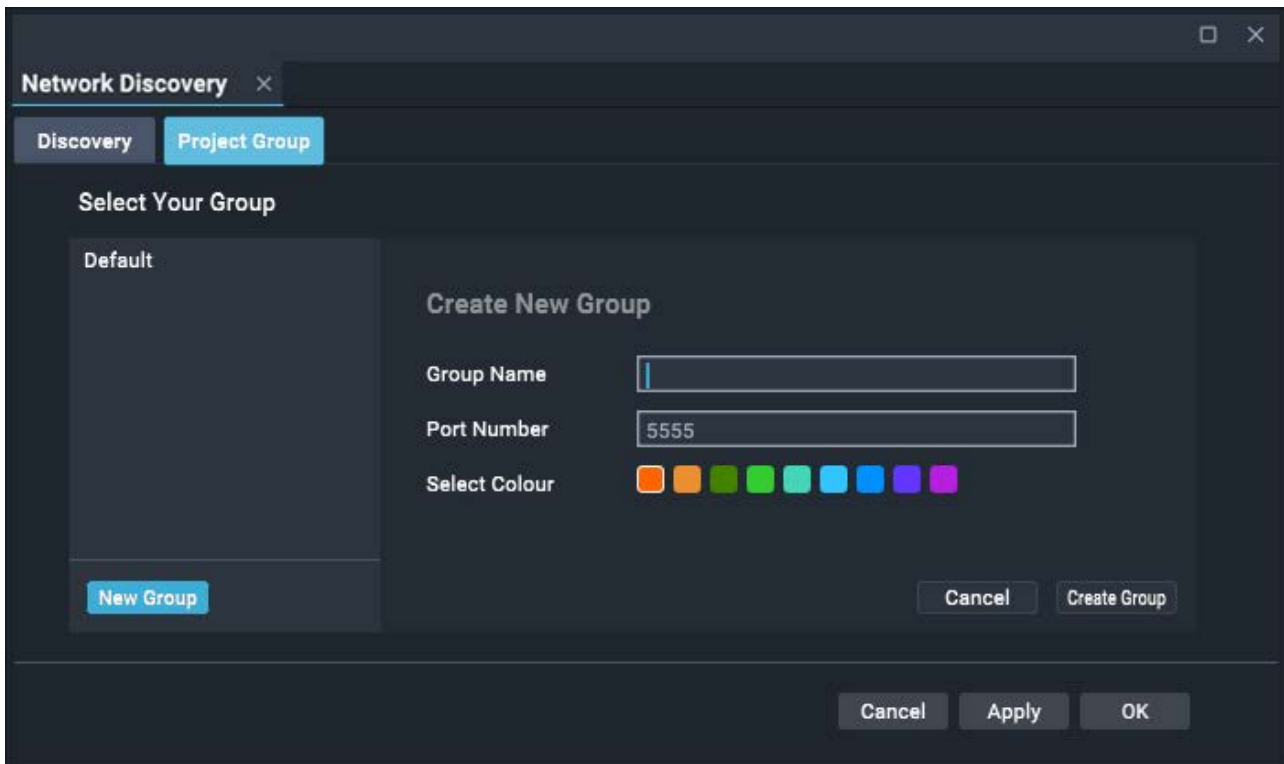
Discovery	Project Group	Hostname	App Ver	Comment	IP address	Project Group	Comms Mode
		Laptop-PC	V1.0.4 B	Main rack	192.168.1.182	Default	
		Juggler 4567	V1.0.4 B	Main rack	192.168.1.179	Default	

What is a Project Group?

All instances of Jugglers and PCs running Compere that wish to share a Project must belong to the same Project Group.

You do not need to create a new Project Group unless you want more than one, or wish to name it. Each Juggler on the network belongs to the 'Default' Group, or can be assigned to your own Group(s). If you need only two Groups, create two new named Groups, rather than using the Default Group plus one named Group.

Click the 'New Group' button and enter a name and the port on which this Group will communicate (both are required), and click 'Create Group':



If you go back and edit these details, click 'Apply' and 'OK'.

Group Name

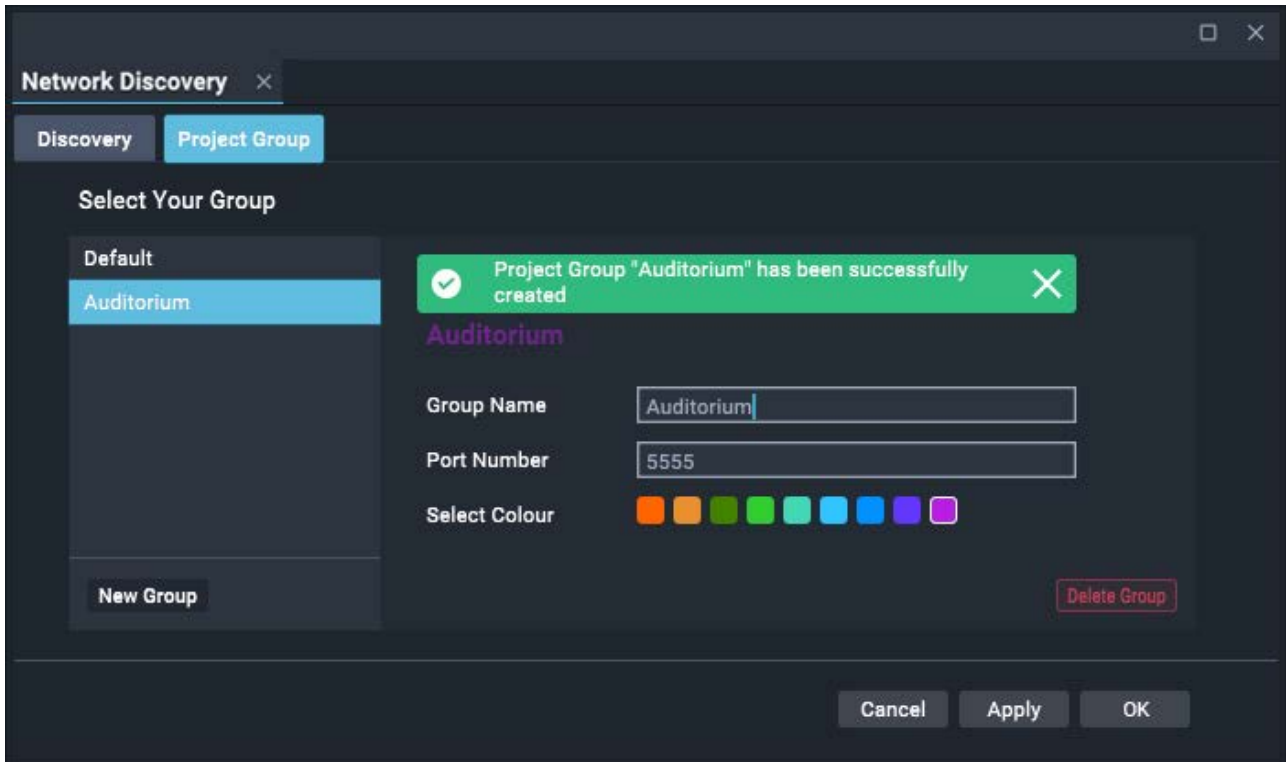
Provide a concise name for this group of Jugglers. Duplicate Group names are disallowed.

Port Number

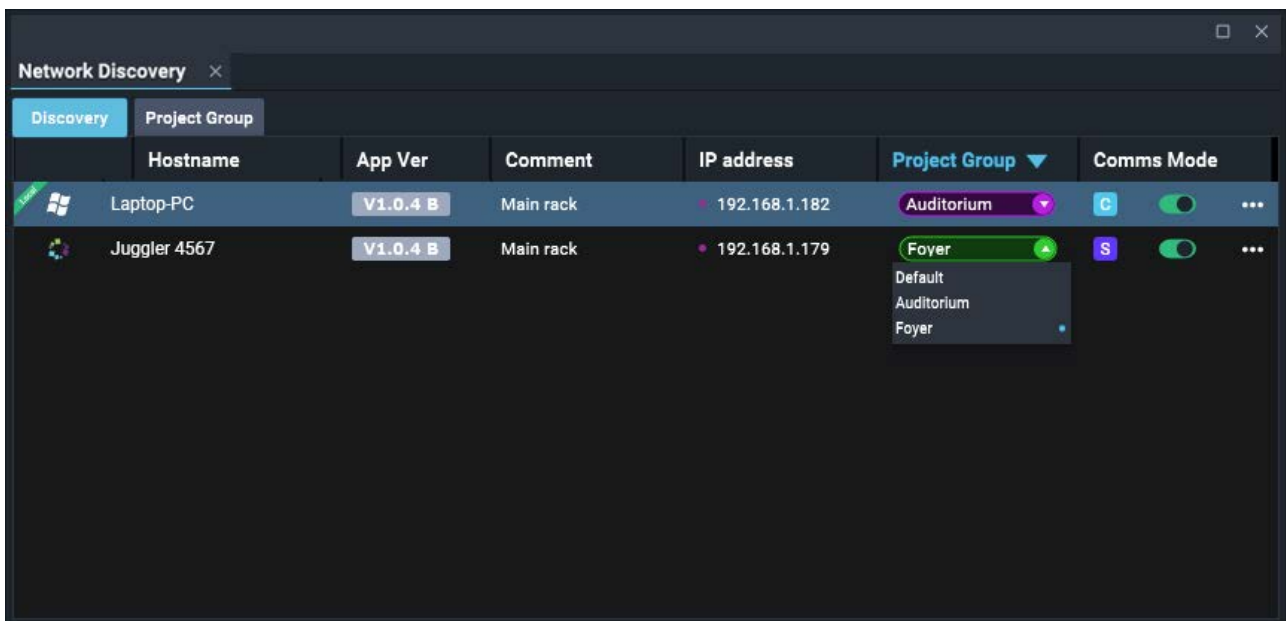
The port you wish to use for this Group. Groups are separate and do not communicate with each other. The default Compere port is 5555, but can be any available port.

Select Colour

Choose a colour to characterise this Project Group. This will help in the Discovery tab to identify members of different groups.



Add as many Groups as you need. Groups can also be deleted and any members will revert to the Default group. The colours are used in the Discovery tab:



Configure a Display

A **Canvas** in a Juggler system comprises all available pixels in a Juggler system as a whole. Within this, **Outputs** to the resolution of available devices are positioned. Depending on the firmware type for your Juggler, direct one-to-one feeds can be made to these Outputs, or to **Windows** as areas anywhere in the Canvas, even across output boundaries. Which type of Juggler you have – Warp/Blend or Windowing – is defined in its factory build (you can see which, in *Juggler > Properties*, under Internal Processor, Capabilities).

The following description is for Windowing Jugglers.

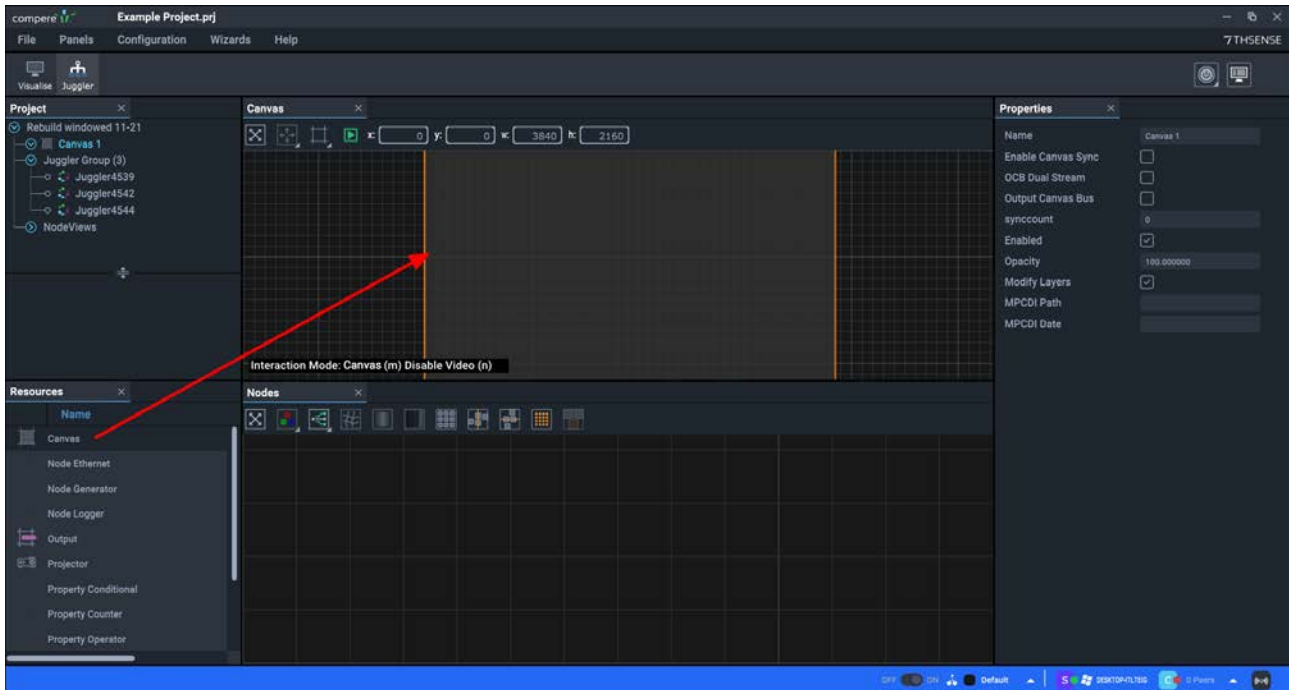
See also:

- [Display Setup via MPCDI](#)⁽⁷⁷⁾ – canvas and outputs are pre-configured, defined and uploaded along with warp, blend and black level files
- [Output Canvas Bus](#)⁽¹¹⁰⁾ – a configuration for maximising numbers of inputs directed to windows moved anywhere on very large canvases

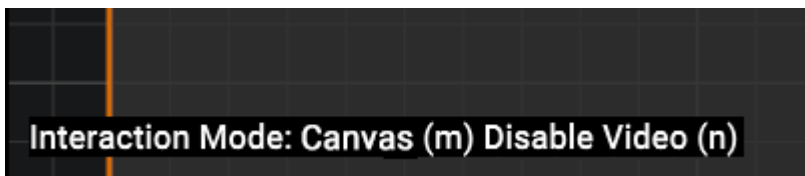
Canvas, Canvas Outputs and Canvas Windows

The Canvas, the Outputs on it, and Windows in them are all laid out in the Canvas assembly panel, simply dragged there from the Resources panel. Each of these will be connected by Compere into the Juggler system, using the Nodes assembly panel. As each element is dragged in, it is added to the Projects panel, its properties become available, including how the element is named.

- Drag a canvas from the Resources panel into the Canvas assembly panel. This will be added at default coordinates (0,0) but can be moved.
- You will now have the grey overall canvas maximised to full view in the panel. Since it is selected, it will have an orange bounding box. The initial default dimensions are 3840 × 2160 (4K).
- The Project panel will now contain a branch for this canvas, and the Properties panel is populated for the canvas.
- You may only need one canvas.



Below the canvas, the **Interaction Mode** is shown:



This mode switches the selection layer between the Canvas, Outputs, and Windows.

Pressing 'm' switches between the three types of canvas objects so they can be selected and moved independently. Once set up you will not need to move the canvas, but you will add and position Canvas Outputs and Canvas Windows within it.

In 'None' mode, canvases can be dragged within the grid to position them using Shift+left mouse, but not to resize by dragging.

Mode selection is also available on the toolbar button or on the right-click menu as 'Interaction Mode'.

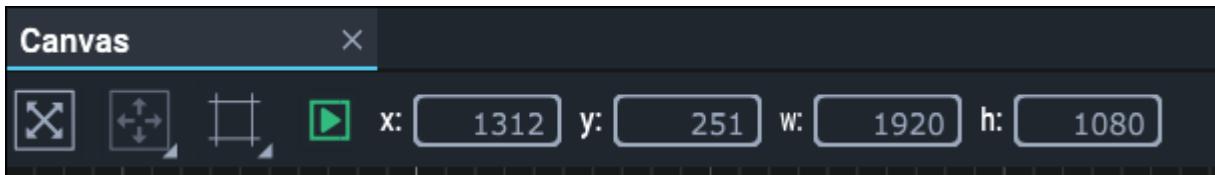
Disable Video: press 'n' to toggle between the outlines of Outputs and Windows and any connected video input.

Menus, moving, sizing

Right-click context menus are available everywhere in the Canvas assembly panel. Many of these options also have keyboard shortcuts and mouse operations as in other panels.

➤ See [Keyboard Shortcuts](#) ¹⁸

Canvas Panel Toolbar



- The first button operates the same as pressing the 'v' key and will bring into scope everything in the panel.
- The second button refers only to Canvas Windows, and provides a list of common positions (top, bottom, left, right, centre and fullscreen).
- The third 'frame' icon operates the same as pressing the 'm' key, switching between object types for selection.
- The green / grey arrow button icon operates the same as pressing the 'n' key, to enable or disable NDI preview in the Canvas panel.

With the canvas selected in the Project panel, configure its properties in the Properties panel.

Canvas Properties

Name

Name this canvas clearly. Multiple Canvases are by default numbered sequentially. It will be listed in your project and selectable for connection to Jugglers. Names can also be changed in the Project panel by double-clicking the name there.

Opacity

Set the opacity of all Windows on this Canvas. This will not be applied to any bypassed video channels.

Modify Layers

Uncheck to fix the layering of Canvas Windows. Check so that selecting any window brings it to the top.

Output Canvas Bus

On OCB firmware-enabled systems: switch bus mode to place the canvas onto the bus. See [Output Canvas Bus](#)⁽¹¹⁰⁾.

OCB Dual Stream

Split inputs in 4-lane output canvas bus systems.

MPCDI Path

On warp and blend firmware systems: if using MPCDI data, enter the relative folder path where you choose to keep MPCDI data files.

MPCDI Date

Date of the MPCDI source being used.

Canvas > Position**X and Y, width and height**

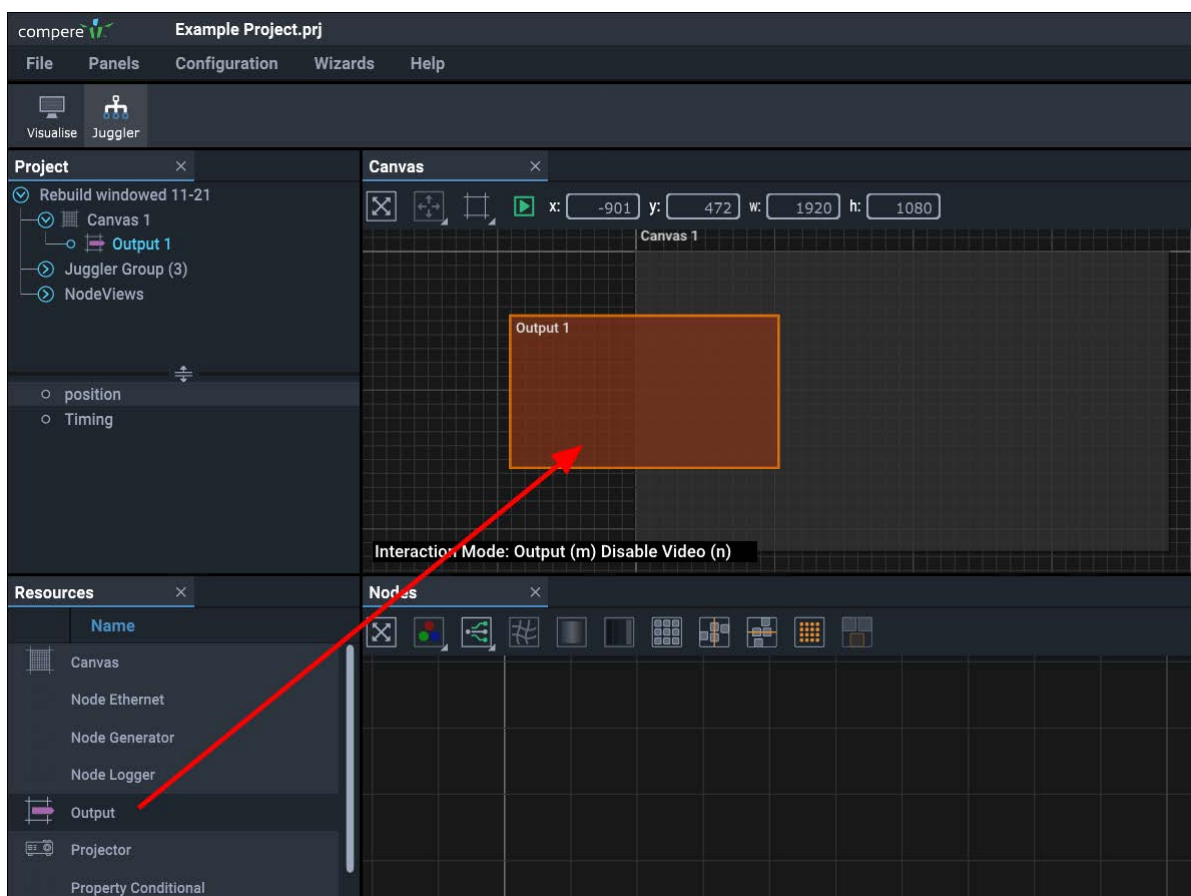
Specify the dimensions of a Canvas.* If using more than one Canvas, these are all inserted at 0,0 X,Y coordinates, so set the X and Y offsets to displace a second Canvas.

* **The resolution** (Width and Height in properties) of the Canvas for a Project is set to at least the maximum overall requirement. It does not necessarily represent contiguous physical space, only the number of pixels available to be claimed by outputs (see [Output Setup](#)⁶³).

Canvas Output Setup

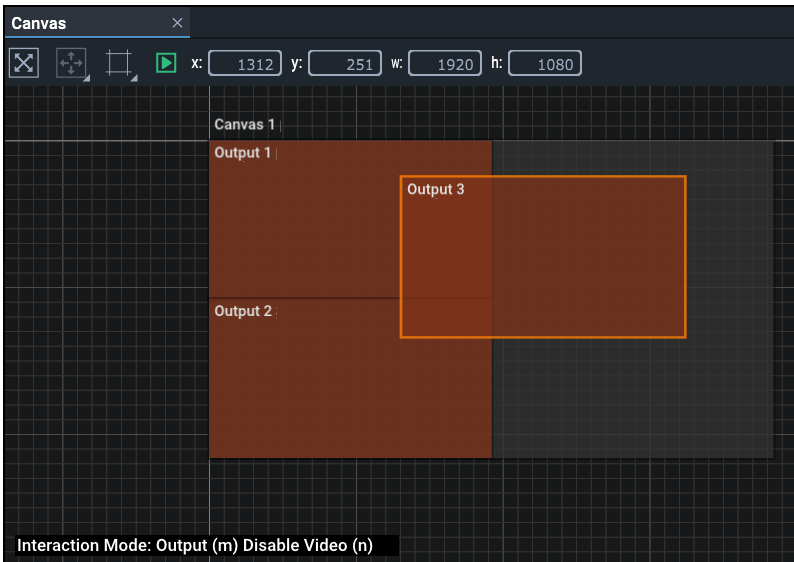
Canvas Outputs are the channels driven from Juggler transmit (Tx) ports that together will comprise your display area(s). They are sized to the physical pixel resolution of the output device. The EDID of Tx ports are set in the Juggler using Nodes View.

- Drag an Output from the Resources panel onto the Canvas.



Again, since it is selected it will have an orange bounding box.

The Output is now visible as a branch in the Project panel as a child to the canvas it was dragged onto. By selecting an Output in the Project panel, you can configure it in the Properties panel. Add all your Outputs in the same way:



Note how the position and dimensions of the selected object is shown in the toolbar.

- Change the **Interaction Mode** to 'Output' by pressing the 'm' key while in the Canvas assembly panel until it reads 'Output'. The same toggle is available on the toolbar button or right-click menu as 'Interaction Mode'. This enables direct selection by double-clicking, otherwise Outputs are selected in the Project panel as branches of the Canvas.

Move an Output around the assembly panel by selecting it, and holding the Shift key while dragging it. Hold Shift+Alt to snap the Output to Canvas edges or to other items in the panel. Alternatively, type in the position, with the dimensions, into the Output > position properties, or the toolbar fields.

Right click Canvas objects for the shortcut menu item 'Move', offering predefined positions and incremental adjustments:

Move Left	cursor left
Move Left 10	shift + cursor left
Move Left 100	ctrl + shift + cursor left
Move Right	cursor right
Move Right 10	shift + cursor right
Move Right 100	ctrl + shift + cursor right
Move Top	cursor up
Move Top 10	shift + cursor up
Move Top 100	ctrl + shift + cursor up
Move Bottom	cursor down
Move Bottom 10	shift + cursor down
Move Bottom 100	ctrl + shift + cursor down

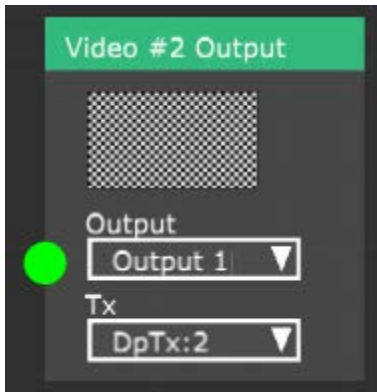
- The Up/Down/Left/Right arrow keys move the object towards that boundary of the canvas, 1 px at a time.
- Add Shift to move in 10 px increments.
- Add Ctrl+Shift to move in 100 px increments.
- Just add Ctrl to move straight to the boundary.

Window objects only:

- Shift+C will centre the item in the canvas.
- Shift+F will make the window fill the canvas.

When you move a canvas around in this panel (you can have more than one canvas), all the child outputs and windows move with it.

Apply an Output EDID. The properties of an output will adopt the EDID of the physical Juggler port to which it is assigned. When, in the [Nodes assembly panel](#)⁽⁷⁹⁾, you select a Canvas Output for a video channel output that has been assigned to a physical Juggler port, the Output *here in the Canvas* will be dynamically resized for you. In this next example in the Nodes panel, the **Tx** (physical output port) for channel 2 has been assigned to DisplayPort DpTx:2. By selecting 'Output 1' in the **Output** selector, the EDID for DpTx:2 will be applied to Canvas 'Output 1', and the Output in the Canvas panel will adopt the correct dimensions.



Edit the properties of an Output by selecting it in the Project panel to populate the Properties panel. Outputs and Windows are branches of the Canvas, which can be expanded or collapsed.

Output Properties

Name

Name the output logically and clearly. Multiple outputs are by default numbered sequentially. Names can also be changed in the Project panel by double-clicking the name there.

X Resolution / Y Resolution

Populated when using MPCDI.

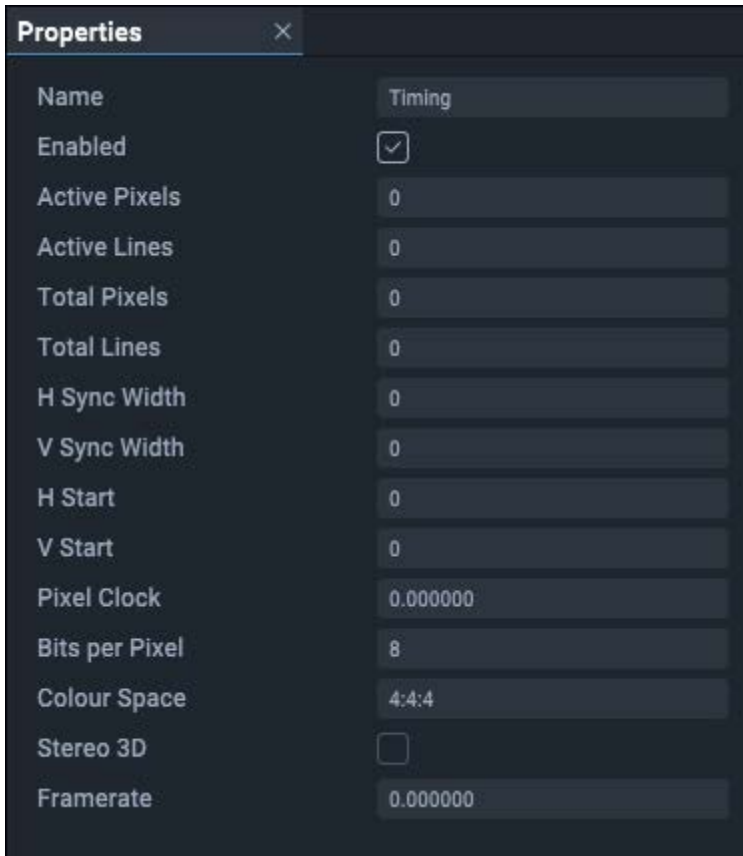
Output > Position

X/Y, Width/Height

X and Y are the top-left co-ordinates in the Canvas. Width and height relate to the physical output device. Precisely position and size this Output in the Canvas.

Output > Timing

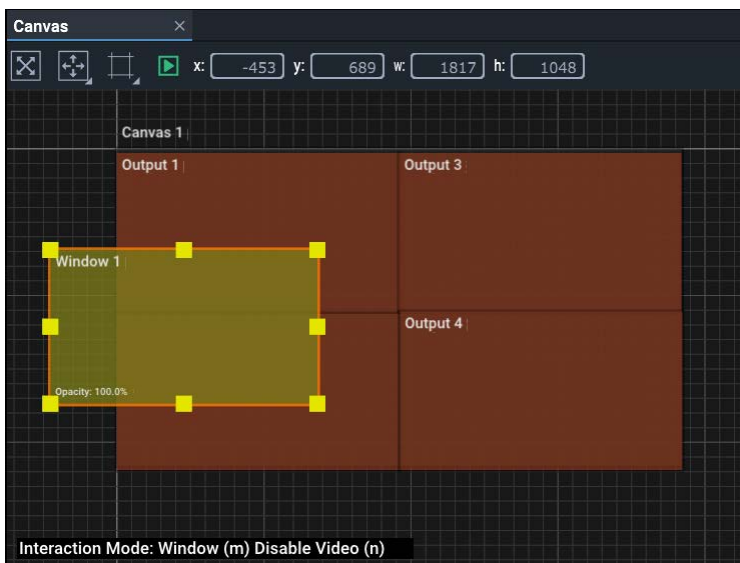
Read only: provides information about this Canvas Output.



Canvas Window Setup

Having set up the output areas, you can now place input sources (media feeds) on them. The placed input source is called a Canvas Window and can be positioned and scaled anywhere in the Canvas area, on or off the Canvas itself.

- Drag a yellow Window from the Resources panel into the Canvas assembly panel:



As with the Canvas Output, an unassigned asset adopts a default resolution. The Canvas Window has been added as a branch to the Canvas in the Project panel. Select it there and its properties become available in the Properties panel.

Interaction Mode: Use the 'm' key toggle to 'Window', to enable dragging and also sizing of Windows. Select Windows in the Project panel as a branch to the Canvas.

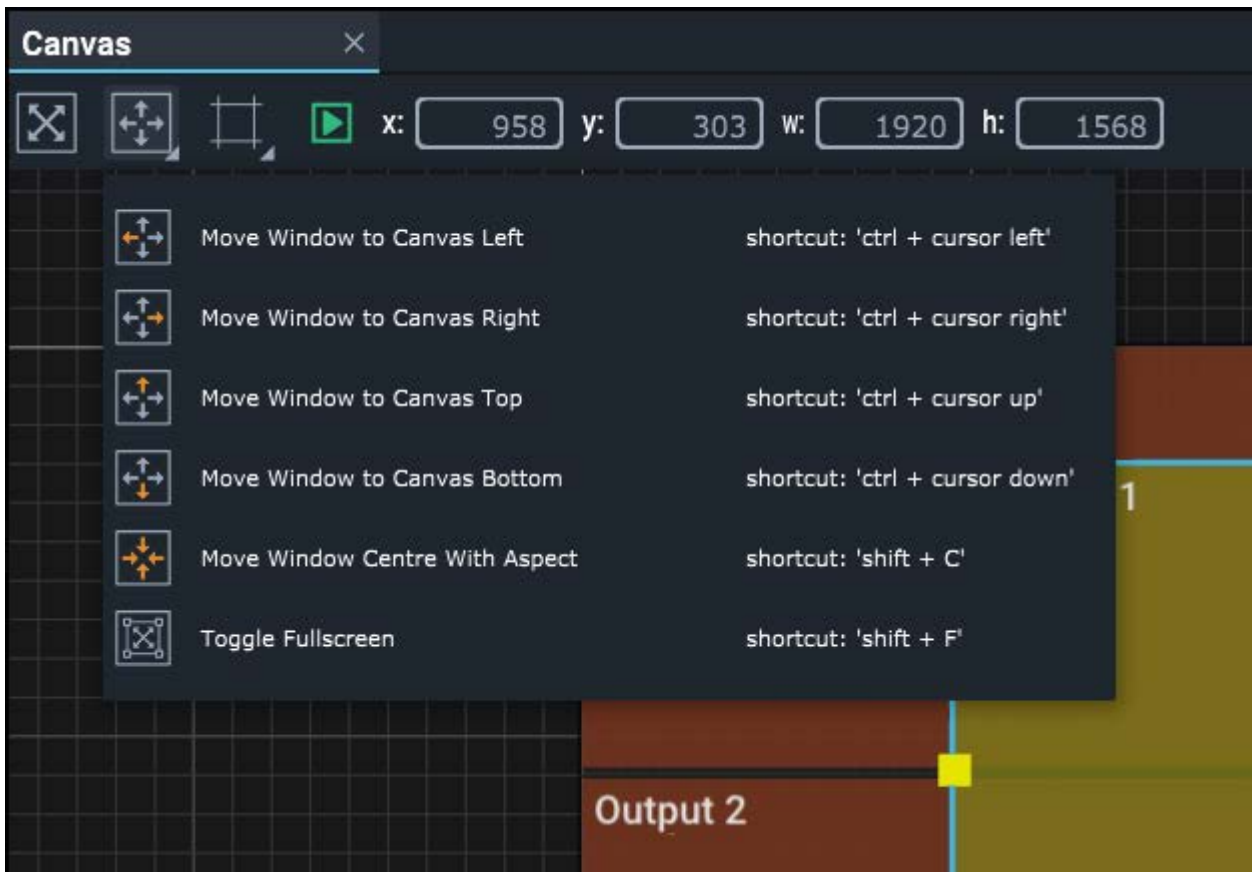
If any window appears to be inaccessible for dragging and resizing, for example, obscured by other windows, select it by name in the Project panel and it comes into focus.

Position the Window anywhere in the Canvas assembly panel using Shift+left mouse. Add Alt+ to drag to snap to other edges. Its size reflects a media source sent to it, scaled as required, and is not bound by any Canvas Output dimensions.

When within the Canvas bounds, the area of a Window that overlaps an Output on the Canvas will be displayed/projected. Any part of the media feed to this Canvas Window that lies outside an Output will not be seen. In this way streaming feeds can be moved around, on or off the Canvas display dynamically, either by interacting directly in this Compere interface, or by [External Control](#)⁽¹³²⁾.

Resize a Canvas Window with the mouse using Shift+ the grab handles. Add Shift+Alt+ to snap to other object edges. Add Shift+Ctrl when sizing from a corner to maintain aspect ratio. Alternatively enter values in the Canvas toolbar or the Window properties.

The other toolbar button is now active to move selected Canvas Windows around to common positions. Incremental 'Move' positions are also available in the right-click context menu for a Window. The Window can also be toggled to fill the Canvas and return to its set size. Shortcuts are listed in the toolbar menu.



Window Properties

Name

Name Canvas Windows logically and clearly for their position or purpose. Names can also be changed in the Project panel by double-clicking the name there.

Enabled

Media will only be processed through a Canvas Window when it is enabled. When this is unchecked, the Window object remains in the Canvas panel shown at 25% opacity. If you do not want a Window to display, and do not want to remove it, it can either be moved off the Canvas, or un-enabled.

Fullscreen

A toggle switch, to make this Window fit and fill the Canvas. Shift+F also has this function, which is on the Window's Move toolbar menu.

Window > Position

X/Y, Width/Height

X and Y are the top-left co-ordinates in the Canvas. Set Height and Width. The input (media feed) will be scaled to this size. Unlike Canvas Outputs, nothing forces the size of a Canvas Window. Like position, resizing can be performed dynamically and by [External Control](#)⁽¹³²⁾.

Window Layering (z-order)

The Canvas property 'Modify Layers' allows selection of any single Canvas Window to be brought to the top (front). Use this to achieve the correct layering of Windows and then fix the layering by unchecking the Canvas 'Modify Layers' property.

Create and Recall Canvas Arrangements

How can you quickly rearrange your canvas setup? For many venues, there will be a requirement for a number of flat-screen scenarios for different times of days or events, or just to reset at the end of a session. It is, of course, quite possible to create multiple Compere projects, but setting up, closing and reopening these would be a slow and cumbersome way of managing the total assets of a venue. In fact we don't actually *need* to replace or rebuild everything.

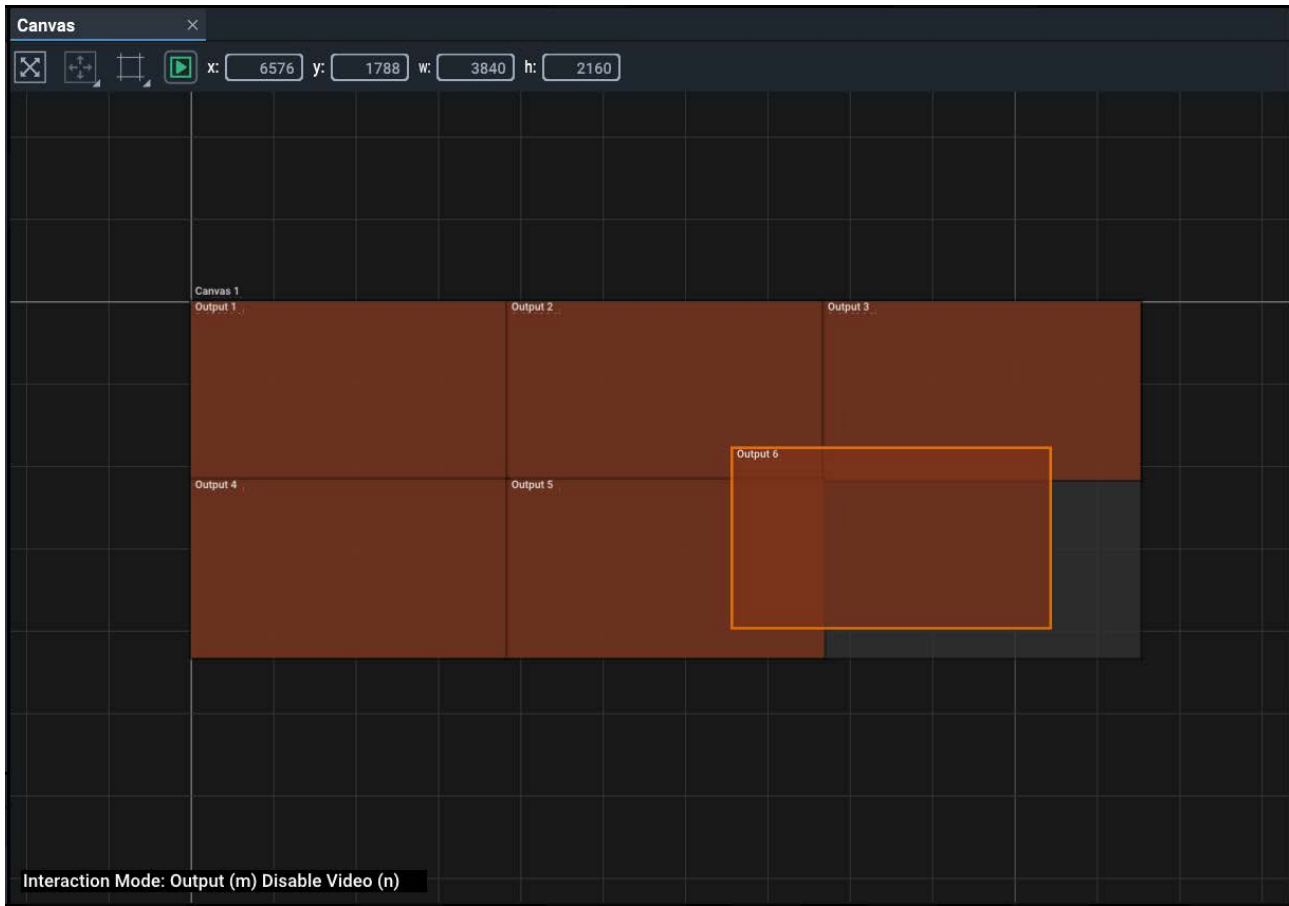
Presets are designed to capture the arrangements and settings for components in the same project. When recalled, dimensions and locations etc. of existing canvas objects will be set again to those of the saved preset.

➤ See [Clones and Presets](#) ¹²⁹

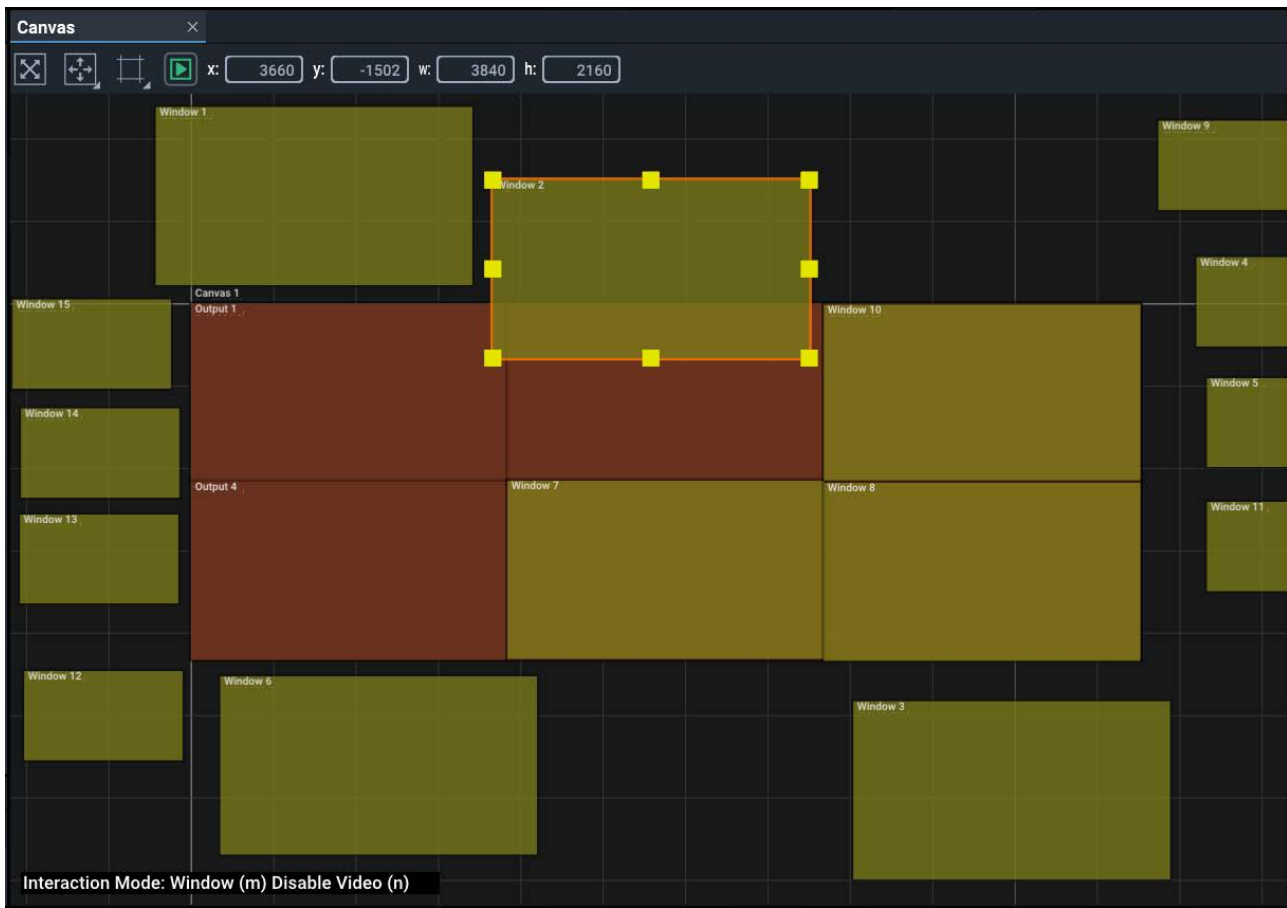
Example of use

1: Assemble all canvas elements

In this example we create a display with 6 × UHD outputs, and video channels running to 15 canvas windows. So to begin, the [canvas](#) ⁶⁰ is added to the project, followed by the [outputs](#) ⁶³, sized and positioned as described above:



Into this, all the [windows](#)⁽⁶⁷⁾ can be added and given dimensions. Once released within the canvas bounds, windows can be moved off into the space outside the canvas bounds, where they will not appear on the actual display. Only those required for a particular arrangement are placed inside the canvas:



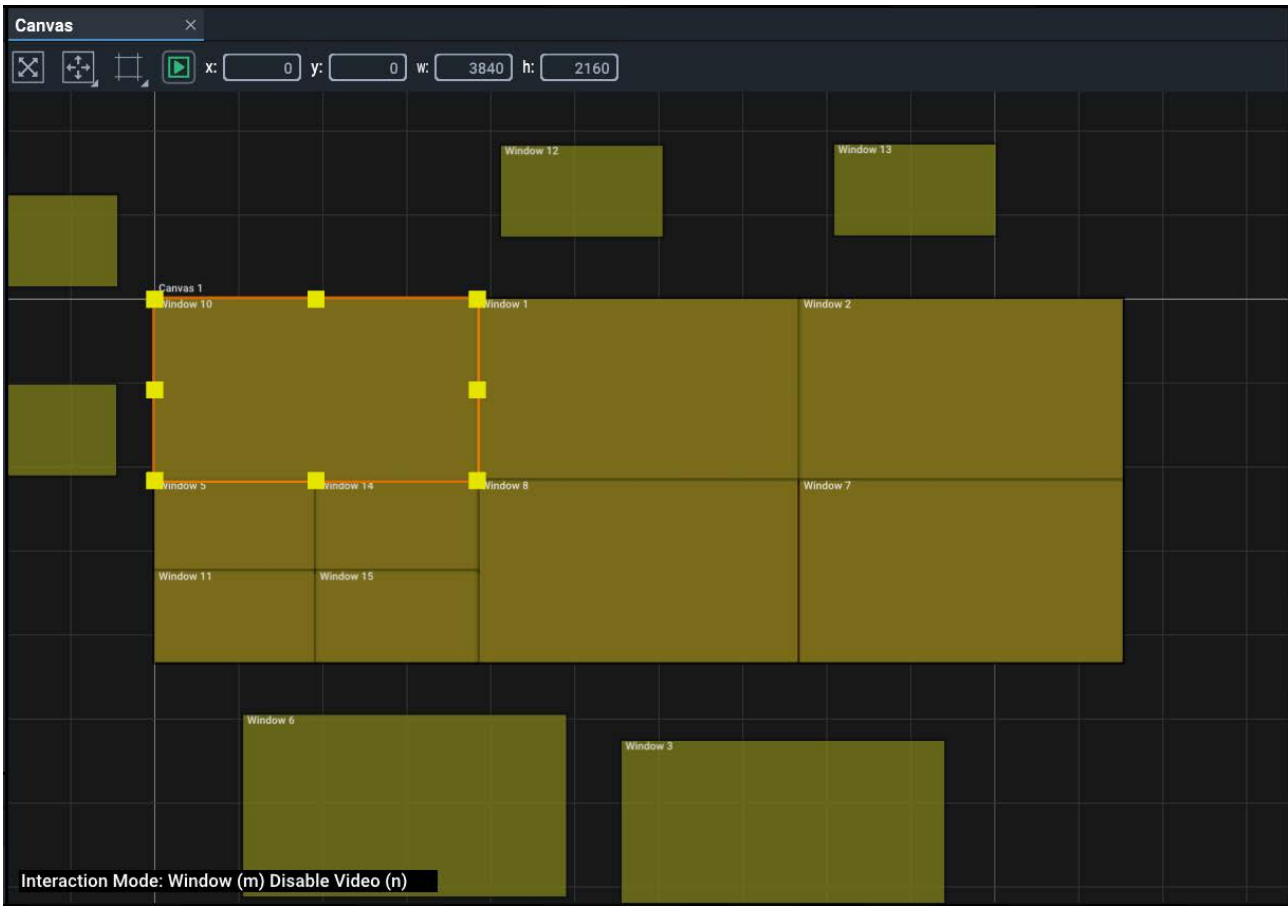
2: Create window arrangement Presets

Once set up, the arrangement can be captured as a Preset. It does not matter where you park windows that are not required for an arrangement. However, with many windows it can be easy to lose track of them.

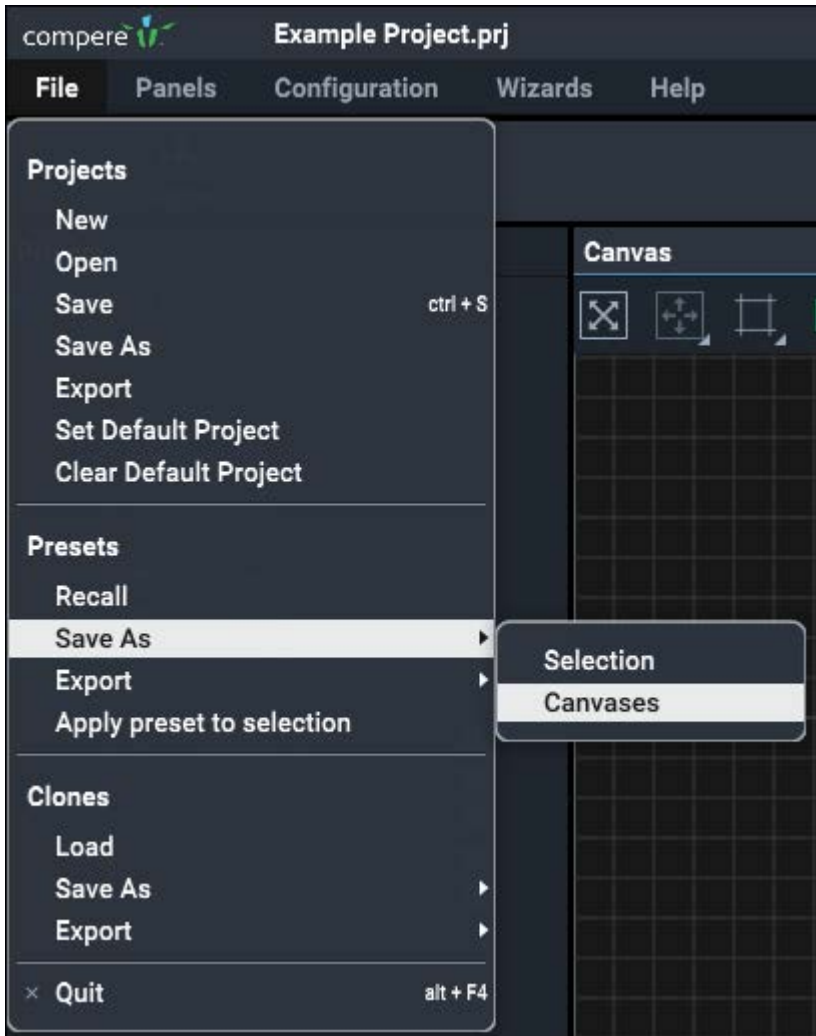


While you are assembling different arrangements, keep associated windows together. NDI feeds may provide sufficient identifiers, but clear naming of windows is important. For example, where a group of windows forms a 'side panel', or a position is used as a 'news feed'.

The Preset will register the positions of windows off the canvas as well as those that are on. Once set up, Presets are recalled by either menu selection, or a single external control command.

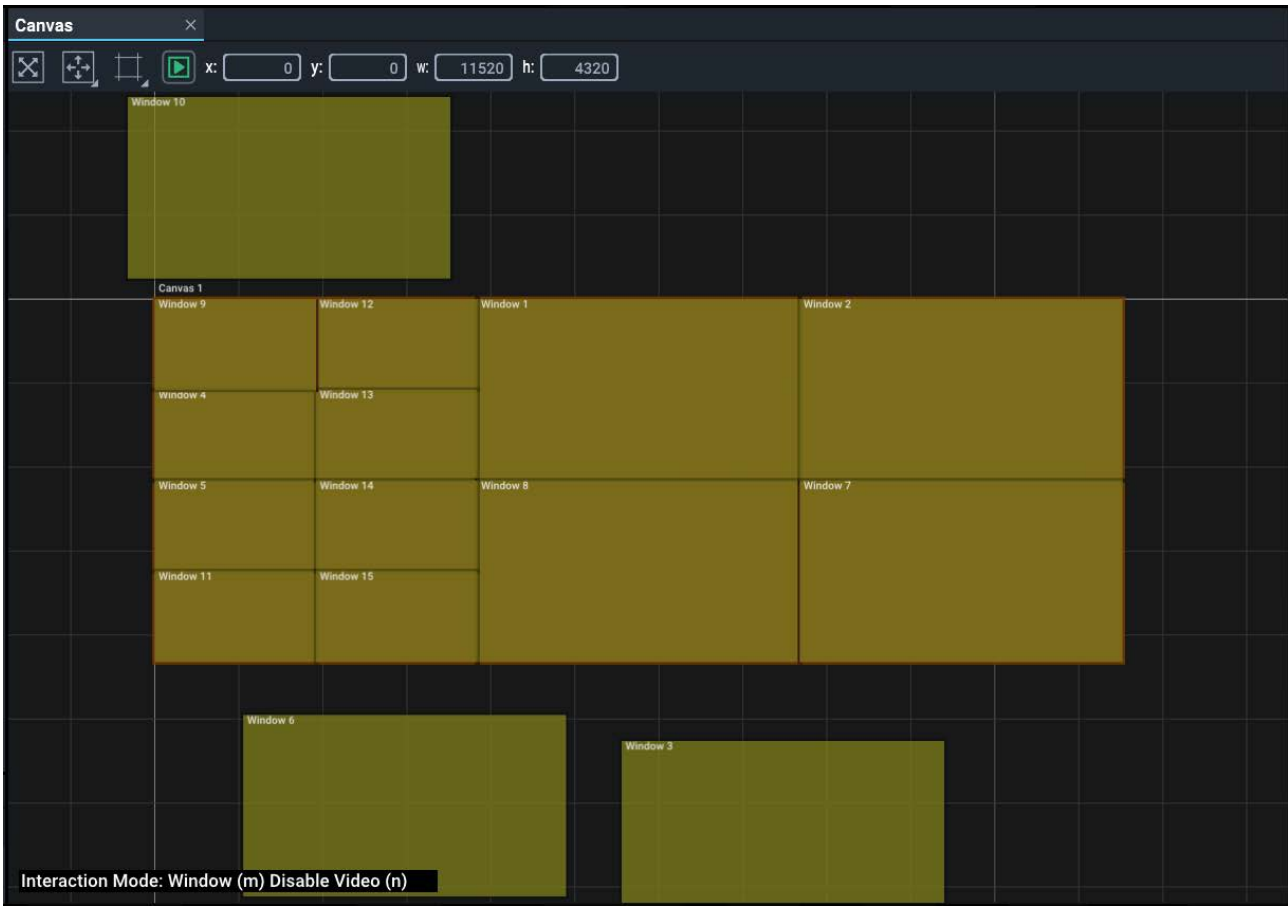


Select *File > Presets > Save As > Canvases*

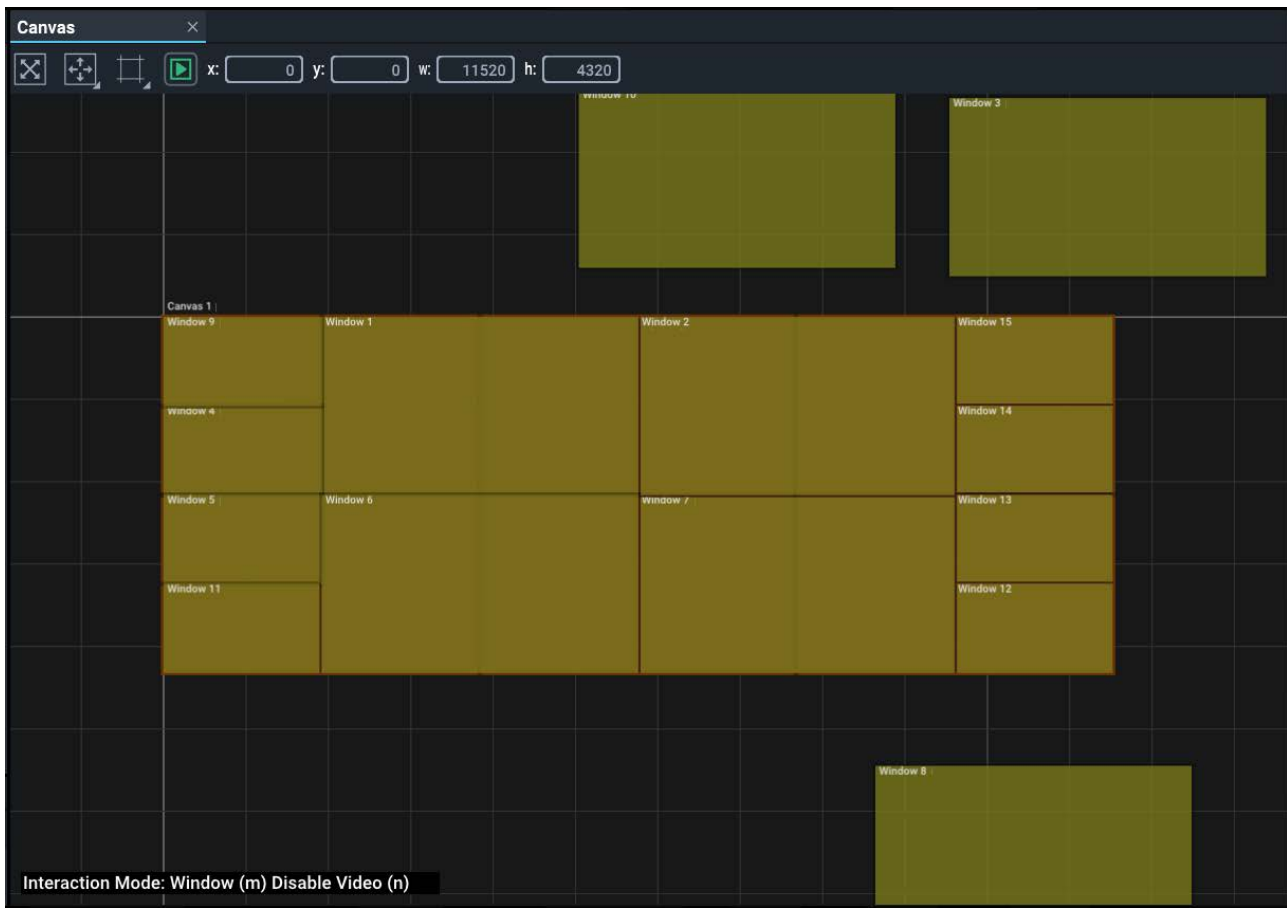


The default file location for Presets is set in *Configuration > Preferences*. Name the preset.

Now rearrange the windows again as required for the next Preset:



Save this as another Preset with a new name. Create further arrangements as required.



3: Recall Presets

From this example, the only changes will be the window positions and/or sizes.

Method 1: From *File > Presets > Recall*, select and apply the arrangement required. As the values for each window are inserted, they will be repositioned in the Canvas panel and the actual display.

Method 2: Send an [External Control](#) ⁽¹³²⁾ command to recall the Preset. A single command will select the Preset file and apply all its values. Example:

```
<command cmd="recallpreset" file="C:[Compere Root Path]\Presets\[file].pre"/>
```

A note on Canvas Window layering (z-order)

Presets are not aware of any other elements on a canvas and so cannot relate to overall arrangements that have changed since they were saved. For this reason, recalling a preset or loading a clone will not necessarily assert the z-order of windows that you require. These can only be changed with 'Modify Layers' enabled in the Canvas, and by selecting Windows to bring them to the top.

Point of comparison: Clones

There are two ways to capture parts of a project setup such as a display configuration: Clones and Presets. Unlike Presets, Clones are designed to create reusable parts for other configurations or projects. When loaded into a project they are instead added as new branches of the project.

Example: a clone would be useful for multiple LED walls of 6 × UHD panels, or very different projects using the same display.

Display Setup via MPCDI

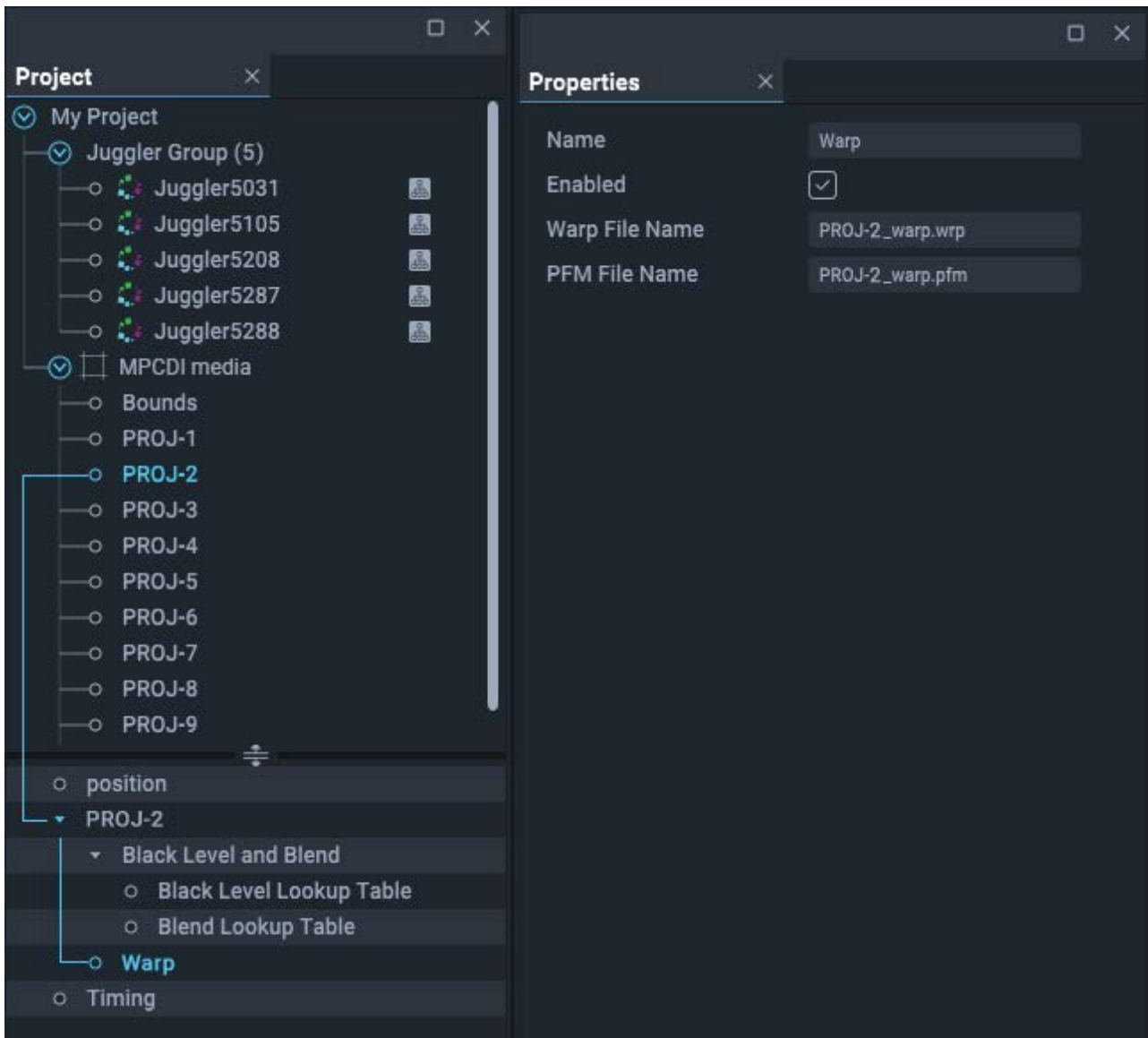
The non-Windowing (Warp/Blend) version of Juggler firmware instead allows for the placement of MPCDI data to define media location, warp, blend and black level correction for a projection system.

In File Explorer, browse to the MPCDI file (*.mpcdi format) and simply drag it into the upper Canvas assembly panel. It may take a few moments to ingest the data, depending on its complexity, before a complete populated canvas appears with the data. .mpcdi files compress and collate a number of other files, including warp, blend and black level correction files, and the mpcdi.xml file itself.

By default, when MPCDI files are imported, Compere automatically distributes the imported (unprocessed) file to all Juggler devices within the project. This can block UI operation for an unacceptable time, and is not necessary. To avoid this, in the Project panel, make sure Juggler Settings > Distribute MPCDI Files is unchecked. The imported file will still be processed and incorporated into the project, which will be distributed.

The MPCDI name will now appear as a branch in the Project panel. The name of the source (here 'MPCDI media') equates to the Canvas. Branches below these (here named 'PROJ-#') equate to Canvas Outputs. These carry the properties for black level, blends and warps.

Names for the Canvas and Canvas Outputs are derived from the imported MPCDI file.



All properties appear on the right as in the windowing setup, but since these are from the MPCDI file, there is no user interaction as when a canvas is assembled from scratch in Compere. All the required canvas outputs will be represented, and connections to outputs are made in the Nodes panel.

On import, the MPCDI information is directed to the Project Group server, which distributes it to reside on every Juggler in the group.

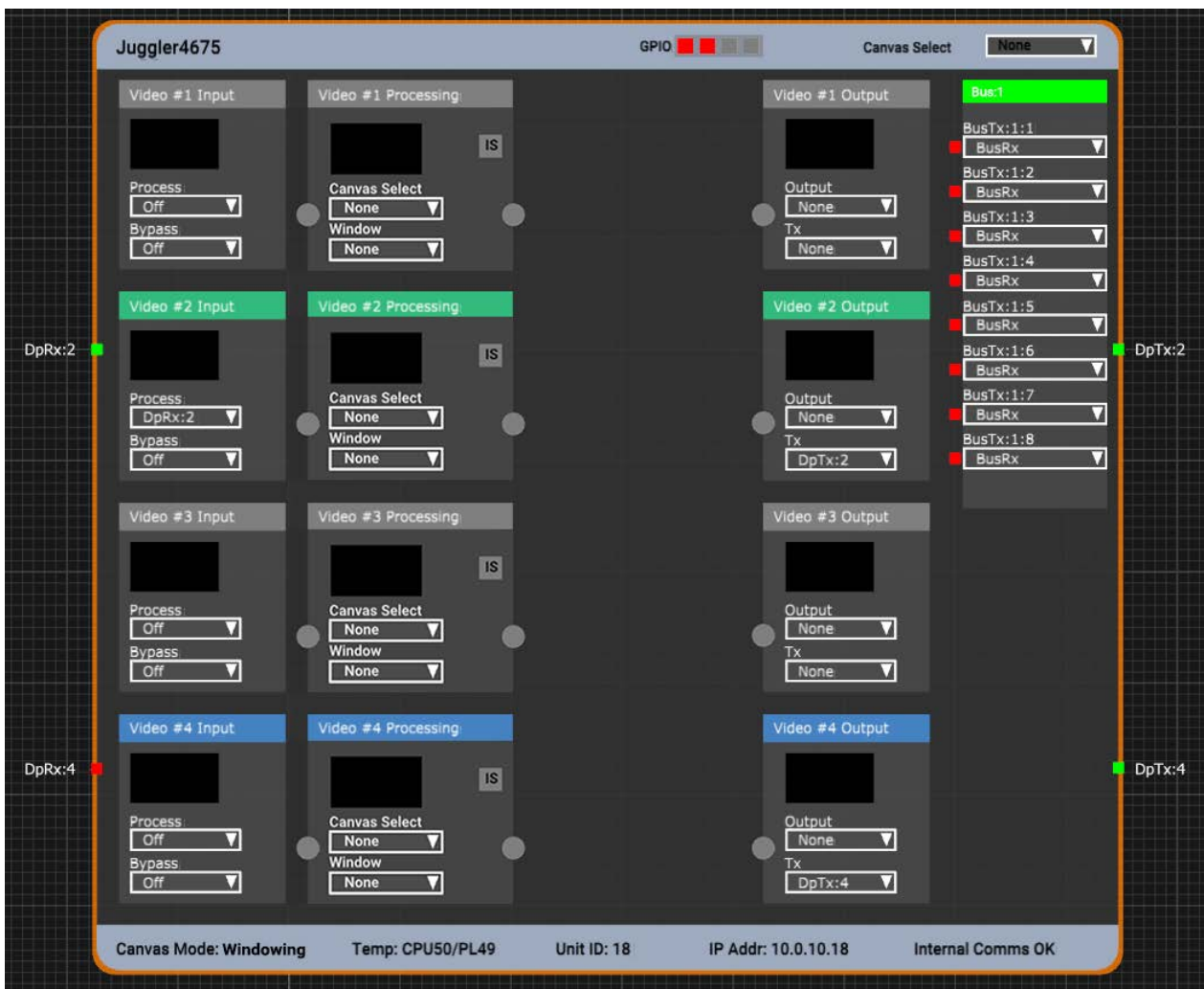
Black Level, Blend and Warp filenames will be shown in the properties for these branches.

A note on MPCDI 'Canvas' naming

Where an imported MPCDI file lies in the Project panel depends on its name (the xml display id) because the panel is alpha sorted. Its name can be edited, but if the MPCDI is reimported, it will not be identified as the same file and will duplicate the data.

Connecting Juggler Systems

Each Juggler in a Project has its build configuration of available ports recognised by Compere, and this is exposed in the Properties and Project panels. When a Juggler is added from the Project panel into a [Nodes](#) ⁽⁷⁹⁾ panel, the build options of the Juggler are represented as connector pins: square for external and round for internal. These are used to make connections within and between Jugglers, media sources and output devices. This is the full zoom level of detail for a Juggler node. As the Nodes panel is zoomed out, levels of detail are progressively reduced and ultimately replaced by simple colour blocks. This is a typical Juggler node:



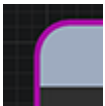
Colours in the Nodes Panel



Hover over any node and it will be outlined in yellow.

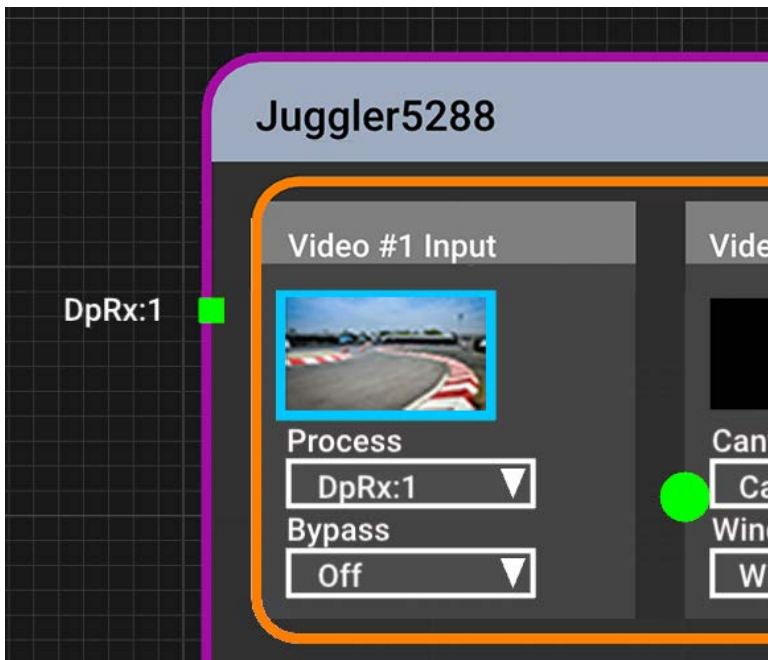


Double-click to select it and it will be outlined in bright orange.



Select an element of a node and the element will be outlined orange and the parent in magenta.

Example, selecting a video channel in a Juggler:



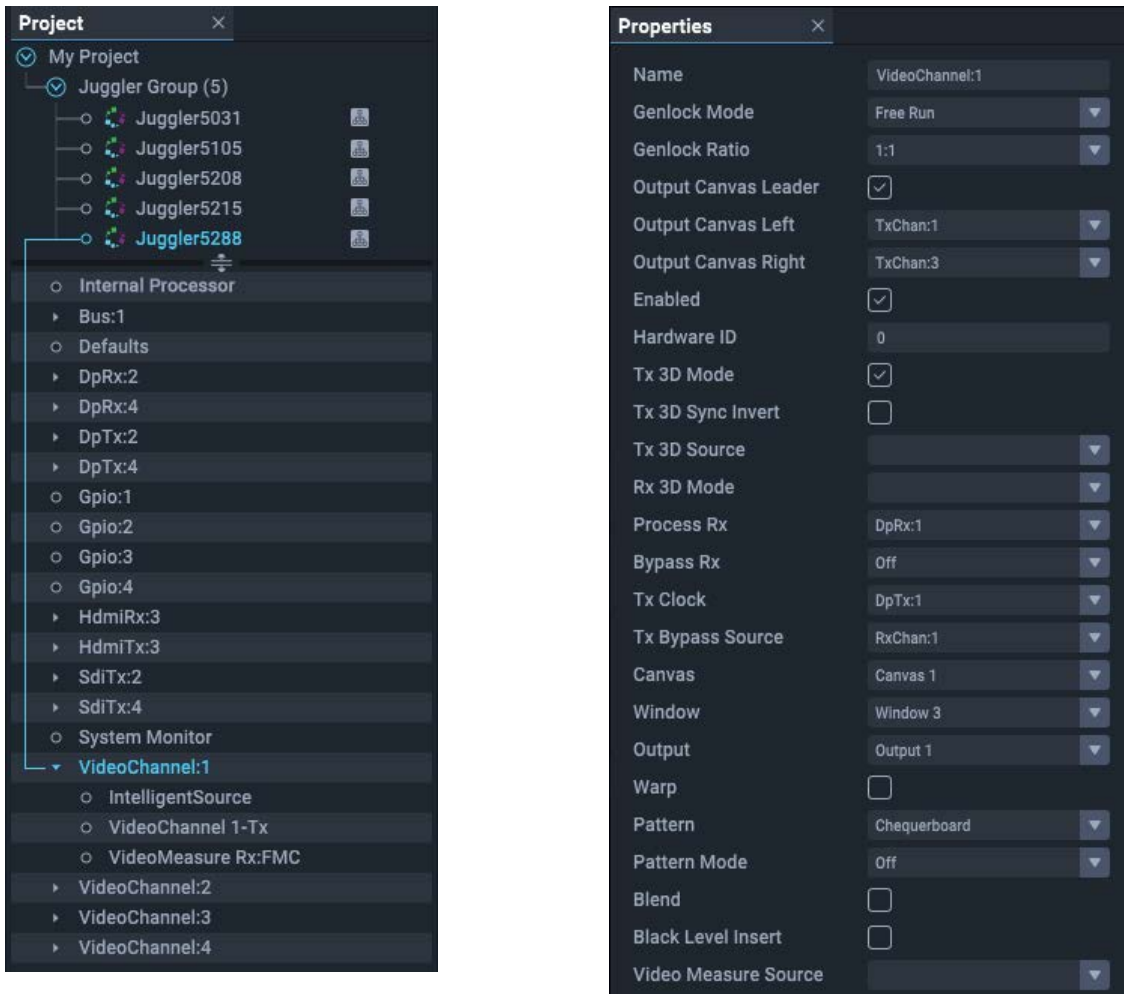
The square pins on the left and right margins represent the physical input and output connectors. When a signal is present they are green, and when disconnected they are red.

An unexpected red pin is a good indicator of a bad physical connector or cable.

Properties

Jugglers in the network are visible in the Project panel as a group. An icon indicates that a Juggler is currently included in the Nodes panel. Each Juggler entry in the Project panel can be expanded into its

parts, ports and channels. Select any element (left, below – highlights in blue) to examine and edit its properties in the Properties panel (right, below).



The Nodes assembly panel is where representations of Jugglers and other parts of the system can be connected: to each other, to outputs, to other devices, and to processes. Here you decide how the receive (Rx) ports of Jugglers – whichever type is present – accept media sources, transform them, and put them out to display via the transmit (Tx) ports – whichever type is present – or elsewhere in the system via the system bus.

Juggler Unit Properties

In the Project panel, select the part of a Juggler for which you want to see the properties. Some of these properties relate to the unit as a whole.

Select an individual Juggler name in the Project panel for its module level properties.

Property	Value
Name	Juggler5288
Enabled	<input checked="" type="checkbox"/>
IP Address	10.0.10.11
NDI	<input checked="" type="checkbox"/>
Auto Identify	<input checked="" type="checkbox"/>
Unit ID	11
Canvas	None
Warp Mode	Warp-1x4
Warp Zone Pixels	3840
Warp Zone Lines	2160
Uptime	0d:00h:02m
Auto Bus Monitor	<input type="checkbox"/>
System Pixel Format	RGB-4:4:4
System Bits Per Pixel	10
HDCP Key Manager State	

Name

By default, the Juggler serial number. Rename, perhaps to reflect real-world tagging or purpose in your project.

Enabled

Normally this should be checked. Remember that in most cases, Jugglers are connected as part of a system, including being part of a bus loop.

IP Address

This unit's IPv4 Address (see [Network Configuration](#)⁴⁹).

NDI

Check this to introduce NDI confidence monitoring on Juggler I/O connectors and in the Canvas assembly panel. In the latter, all video can separately be switched off.

Auto Identify

When enabled, the outputs from this Juggler will throw a sequence of RGB colours in turn onto an output to identify it, when an output is assigned to a video channel.

Unit Id

Last octet digits of the Juggler module's IP address, to provide an identifier within the Project Group. Unit Id is appended to connector identifiers and their type, e.g. when one Juggler AV connection is receiving from an AV connection on another.

Canvas

Select from list of Canvas names created (there may be only 1). Applies to all video channels of the Juggler (same setting as in the Juggler title bar in the Nodes panel).

Warp Mode (warp and blend firmware only)

Depending on the firmware of the Juggler, the unit may be configured for warp and blend rather than flat display windowing. For warp and blend units there are 3 warp modes: 1×4 , 2×2 , 4×1 (this X-Y orientation setting has to match supplied warp files). Warp Mode describes how the four input channels into the particular Juggler unit should be interpreted. Defining the arrangement of the source channels allows an output to span across multiple inputs.

Warp Zone Pixels / Warp Zone Lines (warp and blend firmware only)

Provides a mechanism to override the size of the input Warp Zones (i.e. the video feed into each Video Channel). If this is left at zero then Juggler will use the incoming video resolution of Video Channel:1 to determine the correct size – however, this can cause issues if the video is not present when the warp is being calculated.

Note: All Warp Zones must be the same size.

Warped Test Patterns (warp and blend firmware only)

Select from: Grid, None, X-Ramp, Y-Ramp

Uptime

Total time continuous running of the Juggler FPGA since last restart.

Auto Bus Monitor

Recalculates any bus routing drawn in the Nodes panel, whenever Jugglers report a change.

System Pixel Format

Select as appropriate, e.g. RGB 4:4:4.

System Bits Per Pixel

Select as appropriate 8- or 10-bit.

HDCP Key Manager State

For users of HDCP (content protection). See also [Video Interface Properties](#)⁽¹⁰²⁾.

Values:

Initialising

Read Key File

Failed to Read Key File

Key File Decrypted

Failed To Decrypt Key File

Keys Written to Flash

Failed to write keys to flash

Read Keys From Flash

Failed to read keys from flash

Ready
Disabled

Properties of the component parts of a Juggler

- [Internal Processor](#) ⁸⁴
- [Defaults](#) ⁸⁵
- [GPIO](#) ⁸⁵
- [System Monitor](#) ⁸⁶

Properties of the connections and channels are described in context.

- [Video Channels Properties](#) ⁹²
- [Video Interface Properties](#) ¹⁰²
- [Bus Properties](#) ¹⁰⁵

You can select the same feature in more than one Juggler, and edit shared properties as one. See [Project Panel](#) ⁴³.

Internal Processor

This is the CPU in each Juggler that runs a reduced Linux version of Compere, and communicates with the main FPGA board. In Juggler 1, this is an Intel Atom single board computer. The properties listed under this are largely for reference only, including software and firmware configuration versions installed on the Juggler unit.

Name

Internal Processor

Juggler Connected

Should be checked.

The other properties are for reference:

- Juggler low level SW version
- Juggler FW version
- DisplayPort version (for DP configured units)
- HDMI version (for HDMI configured units)
- SDI version (for SDI configured units)
- Bus version

- Capabilities (e.g Warp & Blend; Scaling/Windowing)
- Compere version
- PIC version
- Front panel default (Info, Rx video input channels, Tx video output channels)

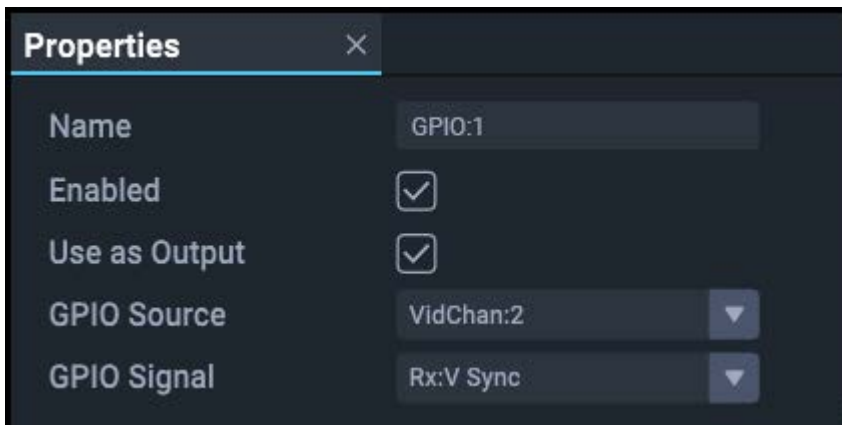
Defaults

Defaults

Default properties relate to Juggler properties held not in the Project, but in the Juggler module's firmware on its internal SD card. They include internal IP and MAC addresses, port numbers, the Tx/Rx EDID on each connector, and pixel formats.

Note: If edited, you must click save data and reboot to effect any changes.

GPIO



Name

Identifies the physical GPIO.

Enabled

For GPIO ports to be live, they must be enabled. If you really need to disable the port, do that here.

Use as Output

If not an output, an enabled port is assigned to receive a signal from a source.

GPIO Source (when used as an output)

To output a signal requires a source. This could be a video channel media source sync, or another feed via another GPIO port.

GPIO Signal (when used as an output)

Select an available signal from the selector. Example: an input (Rx) as a sync source.

System Monitor**System Monitor**

Includes internal temperature monitoring of the Juggler module.

Name

There is no need to edit this.

Current Temp CPU

Shows the current temperature of the internal Linux processor.

FPGA Temp

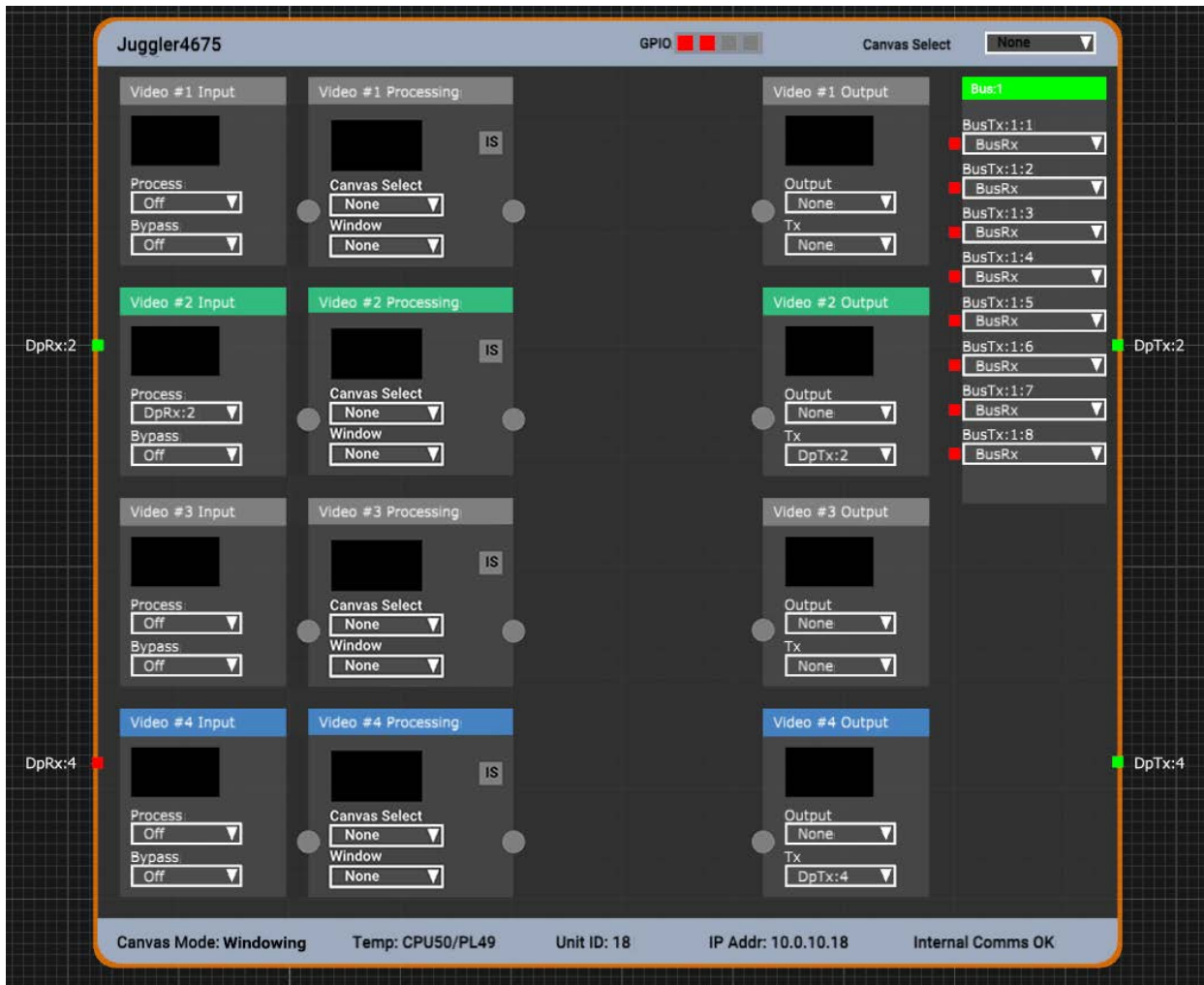
Shows the current temperature of the FPGA processor.

Temperatures are also shown in each Juggler status bar in the Nodes panel, and on the front panel display of each unit. The text will turn orange and then red as temperatures rise to levels requiring attention. Check Juggler filters regularly to optimise cooling fan performance.

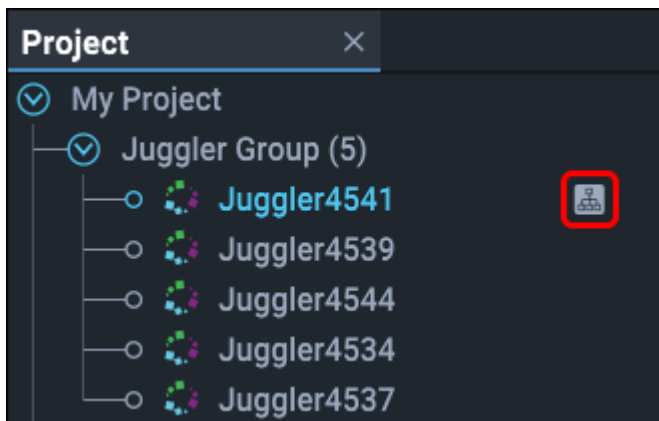
Adding Jugglers as Nodes

When Jugglers are added into the network and assigned to Network Project Groups, they are recognised and added into the Project running for that Network Project Group, becoming visible in the Project panel.

- Drag a Juggler or Jugglers from the Project panel into a Nodes assembly panel. If necessary expand the Nodes assembly panel and press V to view all, or zoom in to reveal more detail and text. Drag further Jugglers in, and position them on the grid as best fits your purpose.



- These Jugglers are now nodes. A Juggler that has been added to the Nodes panel has an indicator against it in the Project panel:



Naming and arrangement

With many Jugglers in various roles, you are probably working to a diagram of how they are connected and their functions. It is useful to name and arrange your Jugglers in the Nodes panel (and/or use the alignment tools – see below) to best represent your diagram.

Juggler title bar

Identifies the Juggler name and allows the Canvas to be selected for all video channels in the Juggler.

GPIO

These four indicators correspond to the GPIO HD-BNC ports on the back panel 1-4 left to right.

Grey: not enabled (in Juggler properties)

Red: enabled but no signal present

Green: enabled and connected

Juggler node status bar information

Canvas Mode

Reflects the type of Juggler firmware being used: Windowing, or Warp and Blend (MPCDI).

Temperature

For internal monitoring of the Juggler.

Unit ID

Last 2 digits of the Juggler module's IP address, to provide an identifier within the Project Group.

IP Address

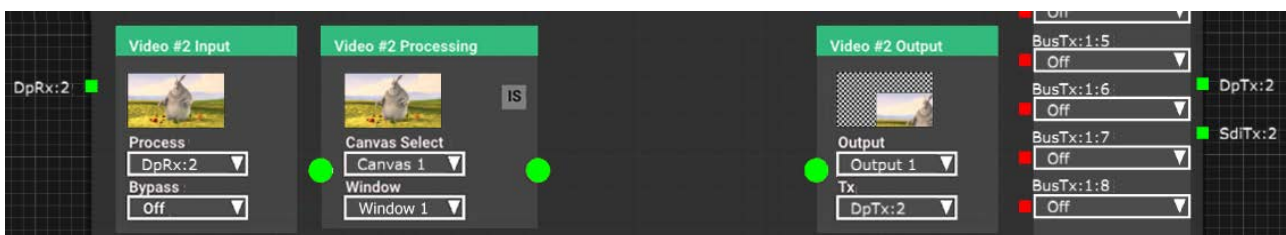
Full IP address of the Juggler module.

Internal Comms

Indicates the status of communication within the Juggler module between the Linux processor running Compere and the main board processing the pixels.

The four Juggler video channels

Every Juggler has four channels of processing: 4 × Rx, 4 × Tx. Available processing channels are arranged left to right and coloured; unavailable channels (e.g. being utilised for the bus) are grey. Here are the essential parts of channel 2:



On the left is the square green Juggler module input pin, here DisplayPort #2 (DpRx : 2). It is showing green because a valid media feed has been detected on it.

Hint: Click any input or output pin to show its properties in the Properties panel.

➤ See [Video Channels Properties](#) ⁹²

Video Input

- This block accepts either the input to this video channel, and/or an input from the bus.
- With NDI enabled, any video appearing here is that arriving at the adjacent physical input for the channel.
- It can connect an input directly to a matched output, **bypassing** any further processing, or it can **process** an input in order to position it on the output canvas (i.e. in a named canvas window).
- An adjacent input pin does not have to be handled by this video channel. It could instead be **selected in the bus** for this Juggler, in order to pass it on to a different Juggler to handle (process to a window, or bypass to a matched output).

Process: this selector chooses the source for this channel to process. Select 'Off' if nothing to process. In the example shown here, DpRx : 2 has been selected as an input to process. This has connected it automatically to the Video Processing block, adding the green connecting link. You might equally want to take a media stream from the bus as an input for this channel to process.

Bypass: media can, if required, bypass all processing and go straight to the same-numbered channel output. Select 'Off' if nothing to bypass. This direct routing reduces bandwidth if no processing is required. A warning is given if the resolutions do not match and the pin will be yellow.

You can bypass one input and simultaneously process another in the same video channel.

Video Processing

If you *process* an input, you are locating it in a window on a canvas.

Canvas Select: select a named Canvas within which this channel will be displayed. The list is populated from Canvases added to the Project, named and placed in the Canvas assembly panel. This can be set from the title bar to apply to all available video channels per Juggler.

Window: select a Window of the selected Canvas, in which the channel media will be displayed (if required). The list is populated when the Canvas Windows are added and named.

Video Output

Output: select a Canvas Output for this channel. The list is populated when the Canvas Outputs are added and named.

Tx: select the port out from Juggler module to a display (square pin on the right-hand side of the Juggler node in Compere).



Note that when **Tx** here is assigned here to a Juggler output, the EDID of that output port will pass to the Canvas Output selected in the **Output** selector. The Canvas Output will adopt size and resolution of the EDID. This is a quick way of setting the properties of a Canvas Output, which can then be positioned in the Canvas panel.

See also:

➤ Media connectors: [Connecting Inputs and Outputs](#) ¹¹⁹

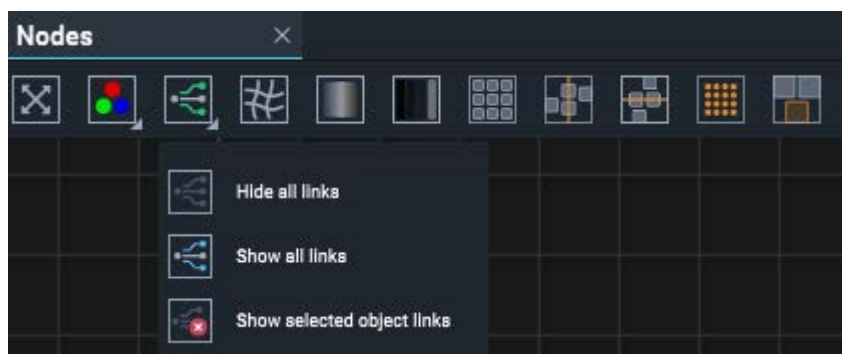
Colours, pins and links

Channel headers, pins and links are grey if unavailable (for example channels dedicated in a Juggler module to a bus). Available channel headers will be coloured when available. Pins and links will be red if disconnected, and green if connected. A yellow pin indicates an incompatibility.

The small square pins on the left and right edges of a Juggler in Compere, are labelled as channel inputs (Rx, left) and outputs (Tx, right).

The larger round pins inside a Juggler in Compere represent connection points for processing media through a channel.

Links are drawn between pins to show these connections, and naturally can become difficult to trace and distinguish with many units. The Nodes toolbar button allows some clarification:



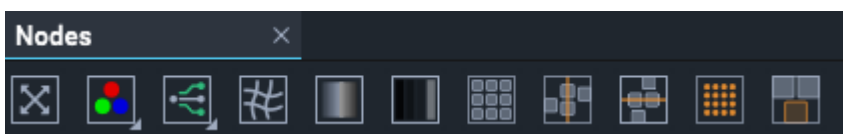
'Show selected object links', for example, will show links to, from and inside a selected Juggler only.












The square pins in the Bus block (upper right of each Juggler node) are the bus connections. These pins will show green when valid media is available on the bus lane.

Multiple selection

Whilst properties cannot be displayed for more than one object, you may want to move or arrange multiple Jugglers. Select a node by double-clicking on it, or click on its name in the Project panel. To select several nodes at once use Ctrl+select.

Nodes panel toolbar



-  Bring all objects into view (shortcut key: 'V').
-  Select colour beams on or off for all outputs, all Jugglers in the Project. The context menu (right-click on a Juggler in Nodes) also offers beam controls for the outputs of the specific Juggler.
-  Display links between nodes selectively.
-  Enable/disable warp.
-  Enable/disable blends.
-  Enable/disable black-level offset.
-  Arrange selected nodes (e.g. Jugglers) in a grid pattern.
-  Arrange selected nodes in a single column.
-  Arrange selected nodes in a single row.
-  Show a miniature map of all nodes and the position of the current view.
-  Make the position of a selected node the viewpoint for any node then selected in the Project panel.
This toggle button is not enabled until one and only one node is selected. When the position of that node is pinned, any node then selected centres all nodes around that point. If more than one node is selected, the viewpoint is disabled again.

The Juggler bus options

On the side of the Juggler node is the bus block. Jugglers can have one of two optical fibre bus options: 4-lane (2 × Rx and 2 × Tx video channels) and 8-lane (4 × Rx and 4 × Tx video channels). The pins on the left show green when the bus pin is connected. This example shows a fully connected 8-lane Juggler. Its green title bar shows it is active. An inactive bus is red.

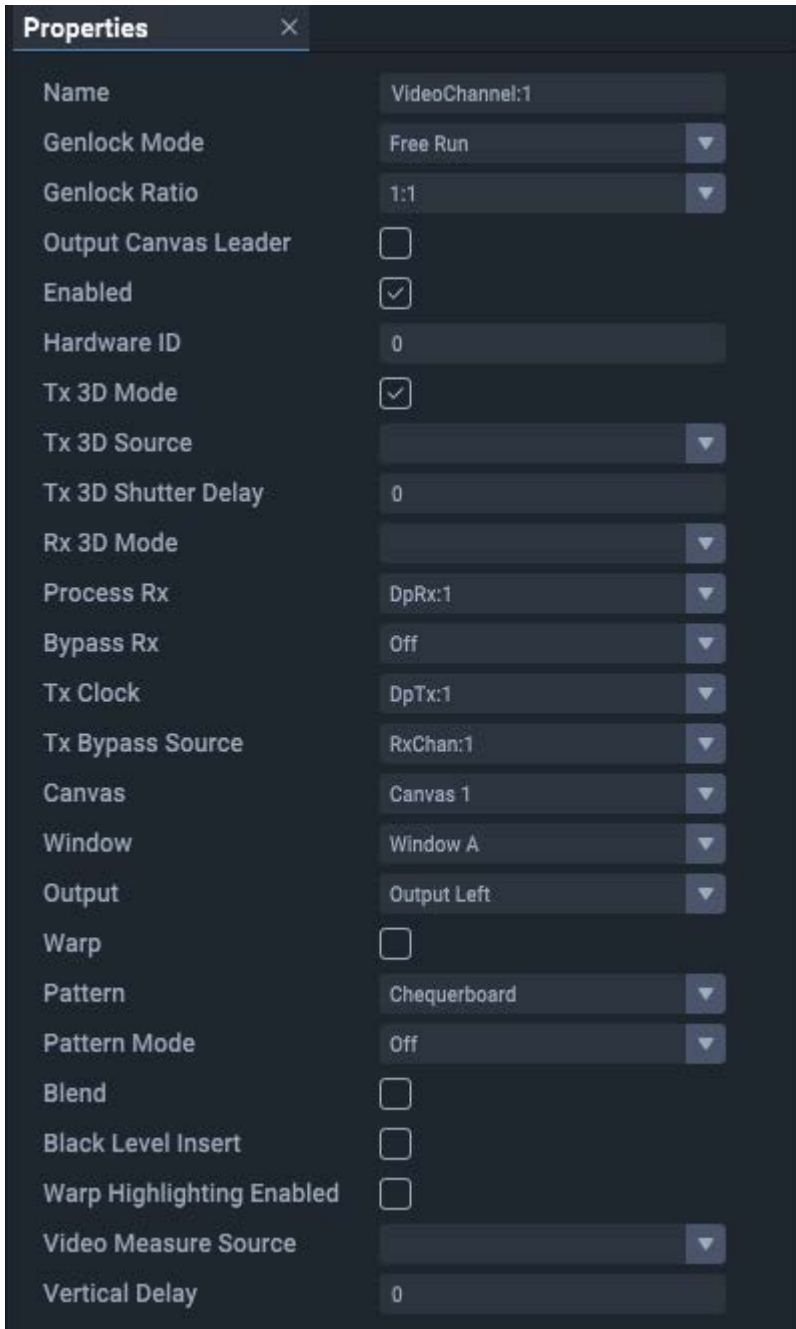


Wider bus options with 12 and 16 lanes are being developed.

For how the bus connections correspond to the physical cabling, see [Bus Connections in a Juggler System](#)⁽¹⁰²⁾.

Video Channels Properties

Up to four video channels are available per Juggler. Select each video channel branch for its properties.



Property	Value
Name	VideoChannel:1
Genlock Mode	Free Run
Genlock Ratio	1:1
Output Canvas Leader	<input type="checkbox"/>
Enabled	<input checked="" type="checkbox"/>
Hardware ID	0
Tx 3D Mode	<input checked="" type="checkbox"/>
Tx 3D Source	[Dropdown]
Tx 3D Shutter Delay	0
Rx 3D Mode	[Dropdown]
Process Rx	DpRx:1
Bypass Rx	Off
Tx Clock	DpTx:1
Tx Bypass Source	RxChan:1
Canvas	Canvas 1
Window	Window A
Output	Output Left
Warp	<input type="checkbox"/>
Pattern	Chequerboard
Pattern Mode	Off
Blend	<input type="checkbox"/>
Black Level Insert	<input type="checkbox"/>
Warp Highlighting Enabled	<input type="checkbox"/>
Video Measure Source	[Dropdown]
Vertical Delay	0

Name

The default name of video channels is VideoChannel:# (1-4)

Genlock Mode

Free run or lock to a source (see: [GPIO Genlock Example](#) in the 'Operating Juggler 1' user guide).

Genlock Ratio

This is ratio of input to output frame rate:

1:1 For normal operation (e.g. 60 fps in to 60 fps out)

1:2 (e.g. 60 fps in to 120 fps out) for passive to active stereo
0 = 1:1, 1 = 1:2, 2 = 2:1

Output Canvas Leader

Applies to [Output Canvas Bus](#) ⁽¹¹⁰⁾ mode only. Check for this Juggler to adopt the lead *for this channel* (i.e. send this video channel to a display device).

Output Canvas Left (if Leader for this channel)

Applies to Output Canvas Bus mode only. Select the Tx pin on this Juggler to carry the left-side part of the output pair.

Output Canvas Right (if Leader for this channel)

Applies to Output Canvas Bus mode only. Select the Tx pin on this Juggler to carry the right-side part of the output pair.

Hardware ID

Information only. The internal recognition identifier in the respective Juggler. Be sure to understand the correlation with the physical interface panel identification.

Connectors in Compere project are identified by type and number, as follows:

DisplayPort: base 000 (so numbered 001, 002, 003, 004)

SDI: base 100 (so numbered 101, 102, 103, 104)

HDMI: base 200 (so numbered 201, 202, 203, 204)

BUS: base 300 (so numbered 301, 302, 303, 304)

Tx 3D Mode

Select to enable 3D out.

Tx 3D Source

Select the required video source.

Tx 3D Shutter Delay

Select angle in degrees (0 to 359).

Rx 3D Mode

Select from: Mono, Left Eye only, Left Eye (Right=Other), Right Eye (Left=Other), or Right Eye, with Left Eye from Channel 2, 3 or 4)

Process Rx

Select a Bus channel to receive media in or place media out, or none.

Bypass Rx

Select to direct a signal from the Bus to the corresponding channel Tx (must be matched resolution).

Tx Clock

Sync this output port to an available option.

Tx Bypass Source

Select the input to send through bypass.

Canvas

Name of the canvas containing the output that this channel feeds.

Window

Name of a window on this canvas to which media is sent via this video channel.

Output

Name of the output on this canvas to which the TX port connects.

Warp (Warp & Blend configurations only)

Enable/disable warp.

Test Pattern

Select a test pattern to display (None; 4 quadrant colour Y ramp; Black; Chequerboard; Colour Chequerboard; Flat; Grid; X ramp; Y ramp).

Test Pattern Mode

Select On, Off, or Mix to overlay the pattern on media.

Warped Test Patterns (Warp & Blend configurations only)

Apply warps to test patterns.

Blend (Warp & Blend configurations only)

Enable/disable blends

Black Level Correction (Warp & Blend configurations only)

Enable/disable black level correction

Warped Highlight Enabled (Warp & Blend configurations only)

Display green fill where video is warped away from rectangular bounds.

Video Measure Source

Select the video source for which you want to see details, listed under the Juggler in the Project panel. See [Video Measure](#) ⁹⁷.

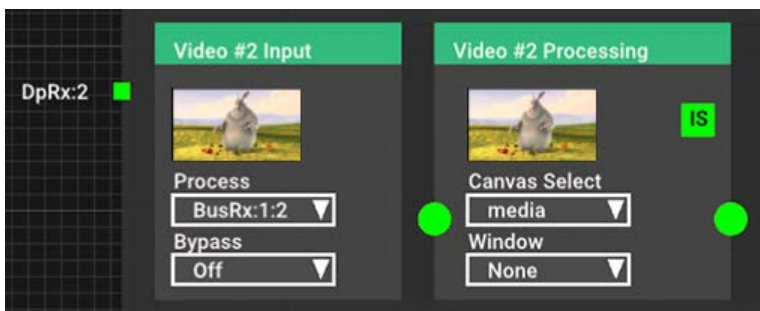
Vertical Delay

Adds a fractional amount of vertical delay to an output, specifically with the aim to minimize throughput delay to the bare minimum.

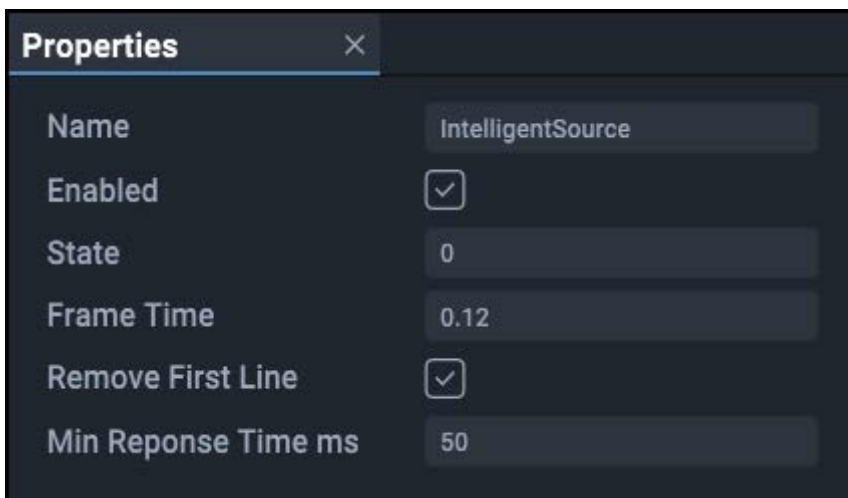
IntelligentSource

7thSense IntelligentSource™ is a method of inserting data into the top line of source frames in order to detect frame sequence breaks and trigger same-frame failover to a backup stream. This can work across Juggler modules to provide either an automatic or manual loop option for source failover. Use of the 7thSense IntelligentSource API in conjunction with this allows both sync and video data monitoring.

IntelligentSource detection is seen in the selected video channel as an 'IS' icon:



The 'IS' icon will be grey until IntelligentSource is enabled. It is enabled, per required video channels for a Juggler, in the VideoChannel Properties panel for that channel of the Juggler:



- Frame Count and Frame Time will be incrementing when all is well.
- If using a stream with IntelligentSource data, check 'Remove First Line' to strip off the additional row of pixels used to convey the frame data.

The IS icon will be green when enabled and good IntelligentSource data is flowing, and red if enabled but no IntelligentSource data is being received. If IntelligentSource framecount/time data stops incrementing, the IS icon for the channel will be yellow.

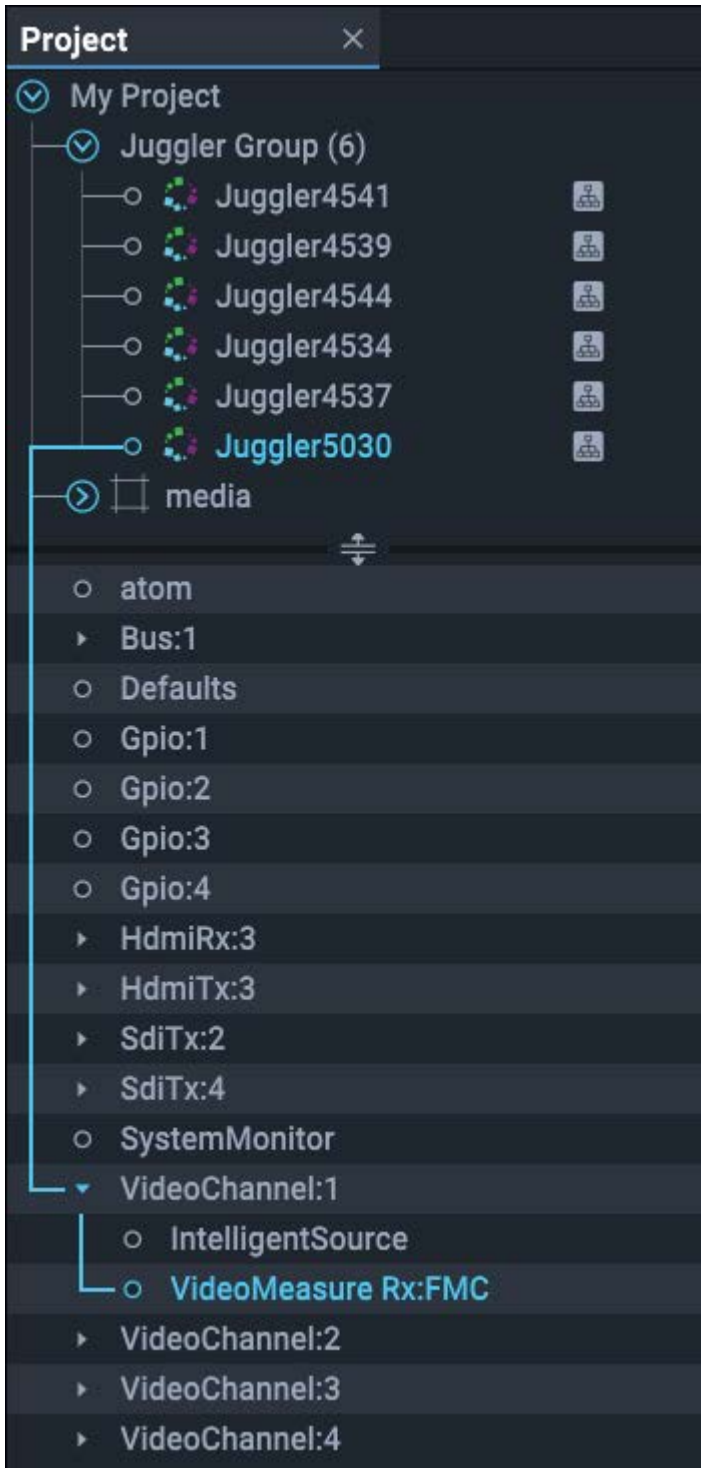
In the event of signal failure, a double-banked Juggler system will switch instantly to the second system.

- To implement IntelligentSource with Delta 2.7 software, document **M450: IntelligentSource API** is available from 7thSense.

Video Measure

Video Measure is a feature that profiles video sources coming into and emerging from a Juggler video channel. These can be via rear panel Interface 1 and 2 (listed here as FMC), or the Unit HDMI (listed here as OnBoard) or via the modules (listed here as SFP).

Select which you want to see the properties for, by selecting from the dropdown in the main Video Channel property, 'Video Measure Source'. This will then be added to the Video Channel branch of the Juggler in the Project panel:



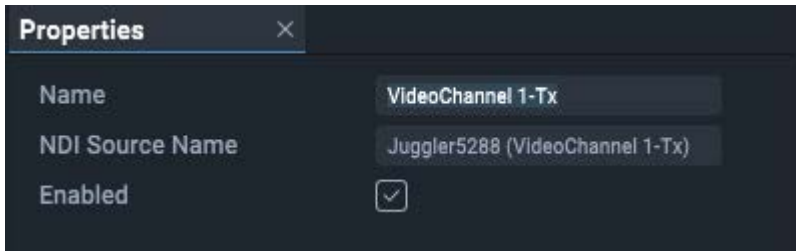
All properties except 'Refresh' are read-only, and give a complete profile of the source.

Refresh Video Measure

This checkbox triggers a refresh of the video measure data when checked. You need to check it after selecting the source in the main Video Channel properties. In response, the Juggler will immediately reset the refresh control after the data is refreshed.

Video Channel Tx

For each Video Channel of each Juggler, the Rx NDI source name is shown (read only):



The Rx NDI source belongs to the adjacent Juggler connector on the left, and shows in Video # Input.

Video Rx / Tx Port Interfaces

Each Juggler module is delivered with firmware capable of the hardware functionality purchased. The firmware determines how each module will be seen in Compere and can be programmed. The Compere software manages what functionality can be configured, per Juggler module.

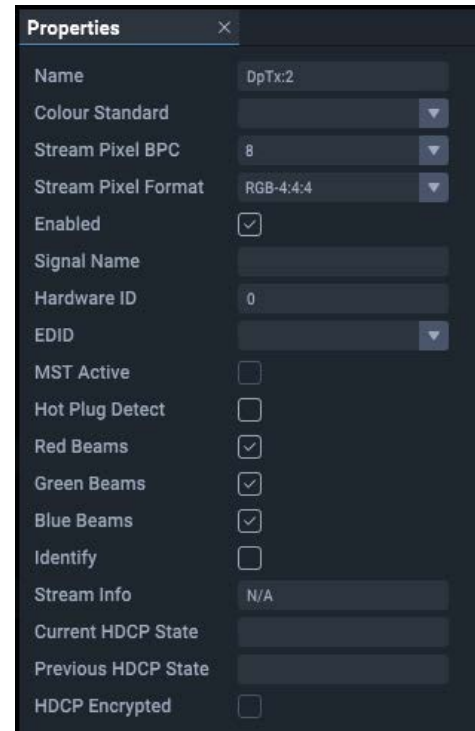
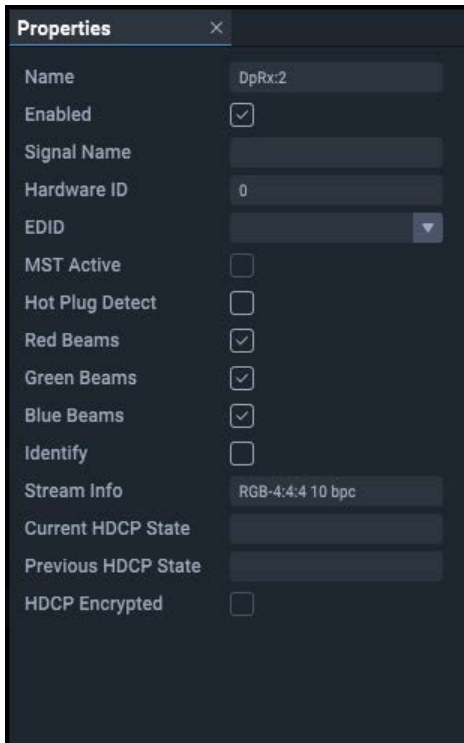
Select a Juggler in the Project and expanded its branch to list its individual elements. Each can be selected for its properties. Alternatively, select the Juggler and expand and collapse its parts in the Properties panel.

Channel Input/Output Properties

There are a total of four Channels, numbered 1-4, in every Juggler. For example, two of these may be allocated as an 8-lane bus, and two as HDMI. The number and type of inputs and outputs on each Juggler module is detected and listed in the Juggler's properties in Compere.

- DisplayPort connectors marked DPRx and DPTx, numbered 1 to 4
- HDMI connectors marked HDMIRx and HDMITx, numbered 1 to 4
- SDI connectors marked SDIRx and SDITx, numbered 1 to 4.
- Data bus connectors marked BusRx and BusTx, numbered 1 to 4

Note that in the case of a 4-channel HDMI unit, video channel 3 occupies the Unit HDMI on the rear left, not an interface panel on the rear right.

Example: DisplayPort Rx and Tx properties**Name**

Type and number. Example default: DP Rx:2, meaning DisplayPort input, video channel 2.

Colour Standard (Tx)

Colour gamuts such as BT.709/Rec. 709 may require scaling the full colour range to a limited range. Options are BT.709 or BT.2020, with BT.709 being the default.

Colour Range

Full / Extended, or Limited range.

Stream Pixel BPC (Tx)

Set your Tx colour bit depth: 8- or 10-bit.

While SDI 12G is always 12-bit for the channel pipe, the video data might only be 8- or 10- bit, and the end device needs to know this.

Stream Pixel Format (Tx)

Select from: RGB-4:4:4, YCbCr-4:4:4, YCbCr-4:2:2.

Signal Name

Optional user-friendly name for the signal.

Hardware ID

The internal recognition identifier in the respective Juggler. Be sure to understand the

correlation with the physical interface panel identification.

Connectors in Compere project are identified by type and number, as follows:

DisplayPort: base 000 (so numbered 001, 002, 003, 004)

SDI: base 100 (so numbered 101, 102, 103, 104)

HDMI: base 200 (so numbered 201, 202, 203, 204)

EDID

Select the EDID from the available list (see [EDID Spoofing](#)⁽¹²¹⁾).

MST Active

Relates to DisplayPort Tx: check if using DisplayPort multi stream transport on this AV connection.

Hot Plug Detect

For DisplayPort and HDMI connectors: bring a connection back to life if lost, e.g. a monitor or a capture source.

Red / Green / Blue Beams

Enable / disable colour beams on this connector (can help in identification).

Identify

Check this to throw a sequence of RGB colours in turn onto an input or an output to identify it. Self-cancelling.

Stream Info

Shows the active video format on the Rx or Tx, for example, RGB 8/10 or YUV 4:2:2 8/10 etc. On the Tx side the format is defined in the Tx itself (or the defaults) This directly impacts the format which is negotiated with the display. Notice the options for 'Stream Pixel BPC and Stream Pixel format' when clicking on a Tx pin.

Current HDCP State

For HDMI HDCP-enabled systems only.

Previous HDCP State

For HDMI HDCP-enabled systems only.

HDCP Encrypted

For HDMI HDCP-enabled systems only.

The combination of Name, Hardware Id, and Channel identify the Connector within the Juggler. Use the Identify checkbox to reveal it in the media display.

Video Interface Timing

These properties are common to all video interfaces. This area of the properties of a video interface relates to the EDID being used on this input or output. It is not editable.

- Name
- Enabled
- Active Pixels
- Active Lines
- Total Pixels
- Total Lines
- H Sync Width
- V Syn Width
- H Start
- V Start
- Pixel Clock
- Bits per Pixel
- Colour Space
- Interlaced
- Stereo 3D
- Framerate

Bus Connections in a Juggler System

There are two modes of use for the Juggler bus: the 'Input Bus' and the 'Output Canvas Bus'. The mode of your system will be established in its firmware. This section relates mainly to the former, and describes the bus in general.

The physical Juggler bus can be configured in two ways: as a 4-lane 'half' bus via SFP modules, or as an 8-lane 'full' bus with MPO connectors (see: [Connect a Juggler Data Bus](#)). In Compere this makes no difference to configuration, and the bus is automatically detected.

A further option is a high-bandwidth configuration of the 8-lane bus, in which lanes 4 and 8 share their capacity with the other six.

Juggler bus bandwidth is UHD (3840 × 2160) at 60fps 10bpc 444 capable (or equivalent bandwidth) per bus lane. The high-bandwidth option (6-lane) is capable of 4K.

A juggler System is a group of 2 or more Jugglers on a data bus. In Compere, any input can be sent onto the bus and redirected to another Juggler module to be picked up there as an input, to process and direct onto a display. This configuration is stored in the Project file.

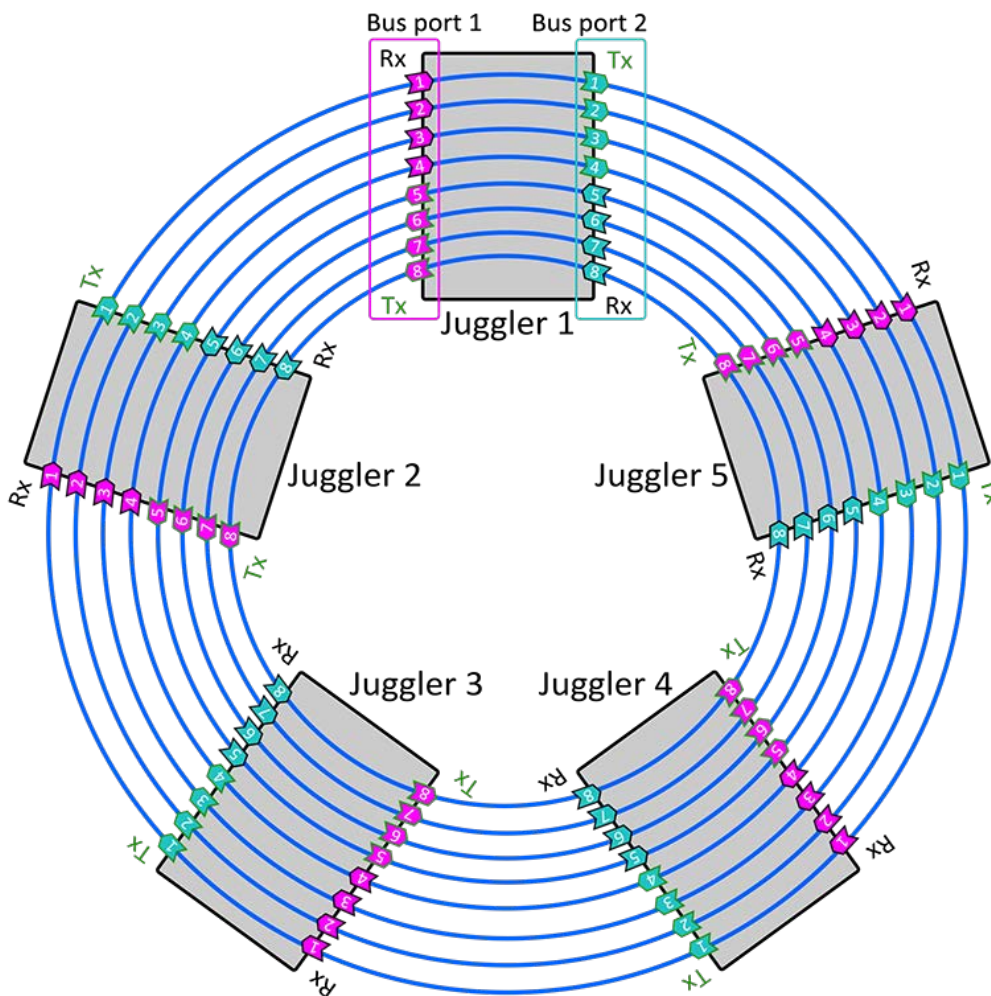
The Bus connection block is represented inside the Juggler in the Nodes panel of Compere.

➤ See also: Physical bus connectors, [Connecting a System in Compere](#) ⁽¹⁰²⁾

8-Lane Bus

24 or 30 bpp – (30 bpp limited to UHD at 60 fps or equivalent)

There will typically be only one bus in a Juggler, but two are possible (see [Bridging Two 8-Lane Buses](#) ⁽¹⁰⁶⁾). The number of pins in the bus are those available lanes on the installed bus. Any Juggler can then place up to four video channels on the bus, and receive up to four as inputs. See in the diagram below how Jugglers interconnect, and the bidirectional flow of the bus itself, indicated by the pointers of the green pins.

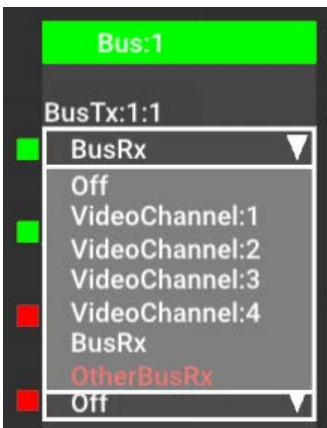


Any Juggler can therefore place media onto the bus, from where any other(s) can stream it off as an input.

In Compere Nodes panel

- The red/green markers on the left of the Bus nodes indicate where a bus input (Rx) exists (green).
- The drop-down list indicates what this bus node is to send out (Tx). This could be nothing (Off), an unprocessed video channel input, or whatever is being received on the bus from the previous Juggler in line (an Rx in the list).

The drop-down selector in the Compere Juggler Bus offers these options:



BusRx

Selecting 'BusRx' here will simply pass on this Juggler's received bus, so Juggler 1 BusRx:1:1 will connect internally to Juggler 1 BusTx:1:1 and be passed on to Juggler 2 BusRx:1:1 and so on, as in the diagram above.

Video Channels

The 4 video channels listed are the media channels of this Juggler. By selecting an available channel that has an input (green), this input will be sent out onto the bus and can be streamed off as an input anywhere else along the bus.

OtherBusRx

A single Juggler can have two 8-lane bus interfaces. One example of use would be to link two racks of Jugglers in an input failover (IntelligentSource) configuration. See [Bridging Two 8-Lane Buses](#)⁽¹⁰⁶⁾.

Example:

For Juggler 1 in the rack, select an input source for its Video #2 Input. Now, in the Bus block, select Video Channel 2 for BusTx:1:1. On Juggler 2 (next on the rack bus), the pin for BusTx:1:1 will change to green to show a valid media source is now present on that pin.

This media is now available on this lane of the data bus, from where the channel can be selected as an input by another Juggler to process.

Example continued:

On Juggler 2, select BusTx:1:1 as the input to Video #4 Input. Media from Juggler 1 channel #2, will now pass to Juggler 2 channel #4, from where it can go to a Canvas, an Output, and a Window. If NDI is enabled, the media will show present on this video channel in the Nodes panel.

Connect the bus as required for your data. You do not need to make unnecessary connections.

4-Lane Bus

This option has half the capacity of the 8-Lane bus, operating through Juggler SFP modules. The arrangement is similar to the 8-Lane diagram, but can be imagined as pins 1 to 4 only.

High Bandwidth Bus

To increase bandwidth, lanes 4 and 8 can share their capacity with lanes 1-3 and 5-7. Effectively this becomes a 6-Lane bus, with lanes 4 and 8 unavailable to separate video channels. This is enabled in the main bus properties (see below).

Bus Properties

Bus availability and type will vary according to the Juggler build.

Name

It can be useful to name a bus by function if there is more than one.

Bus Number

In most cases there will only be one bus in a system.

Bus Reset

A debug function only, checking this will switch the bus off and on again, and will therefore interrupt any video flow around the bus.

Health Test

A debug function only, checking this will request a health check be performed on the Bus. The box will be instantly unchecked again and the '**Health**' parameter below will report some text about the health. If all is well, the text will be 'OK', otherwise it will be one of a number of things that to assist with debugging the Bus.

Health

Should be reading OK. Anything else indicates a possible problem with bus connectivity.

High Bandwidth

Check if a 6-lane high bandwidth bus is implemented. See [Bus Connections in a Juggler System](#)⁽¹⁰²⁾.

Output Canvas Bus

Enable output (rather than video channel) busing mode. See [Output Canvas Bus](#)⁽¹¹⁰⁾.

Bus Rx and TX pins

Each Bus element of a Juggler can be further expanded in the Project panel, into its available receive (Rx) and transmit (Tx) pins. The Bus is a Video Interface and shares some of the same property types as Video Rx/Tx pins.

Name:

BusRx 1:1 indicates bus 1, receiving lane 1.

Hardware ID

The internal recognition identifier in the respective Juggler. Be sure to understand the correlation with the physical interface panel identification.

Connectors in Compere project are identified by type and number:

Bus: base 300 (so numbered 301, 302, 303, 304)

Other properties here identify the bus, the type, and the bus lanes in use.

Tx Source

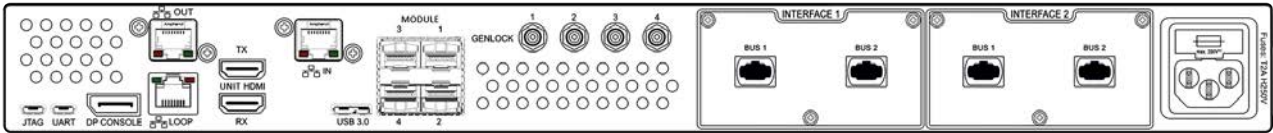
As selected in the bus selector in Nodes panel.

Video Timing

This part of the properties relates to the EDID being used on this input or output. For information; it is not editable.

Bridging Two 8-Lane Buses

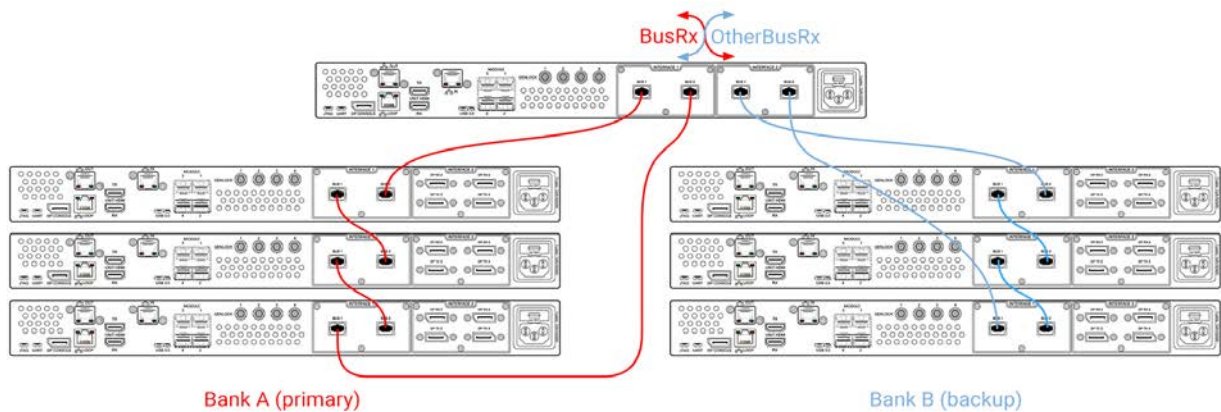
Juggler systems can process many inputs and drive complex displays. Compere is in control, but what happens if a video input fails? Compere has the IntelligentSource failover feature, which can be used to instantly redirect synchronised inputs from a second system into the primary bus instead. Such a system uses a specific Juggler configuration with two bus interfaces (dual bus) instead of media inputs of its own.



This allows for two separate banks of channels to be routed through the same Juggler, acting as a bridge between the two. Compere is used to assign and distribute any combination of the available 16 channels* to any of the two 8-channel bus banks, to be processed for outputs. (*A high bandwidth configuration dual bus would comprise 2 × 6 channels.)

Example of use: redundancy failover

In this example, a primary bank (A) of jugglers can assign 8 input channels to 8 bus channels and use those bus lanes to support specific outputs. A second (or backup) bank of Jugglers (B) with matched inputs is placed on a second set of bus lanes. If input to primary bank A fails, the backup bank's inputs take the place of those on the primary.



This switch would happen within the dual bus juggler where the 'BusRx' assignment would be switched to 'OtherBusRx':



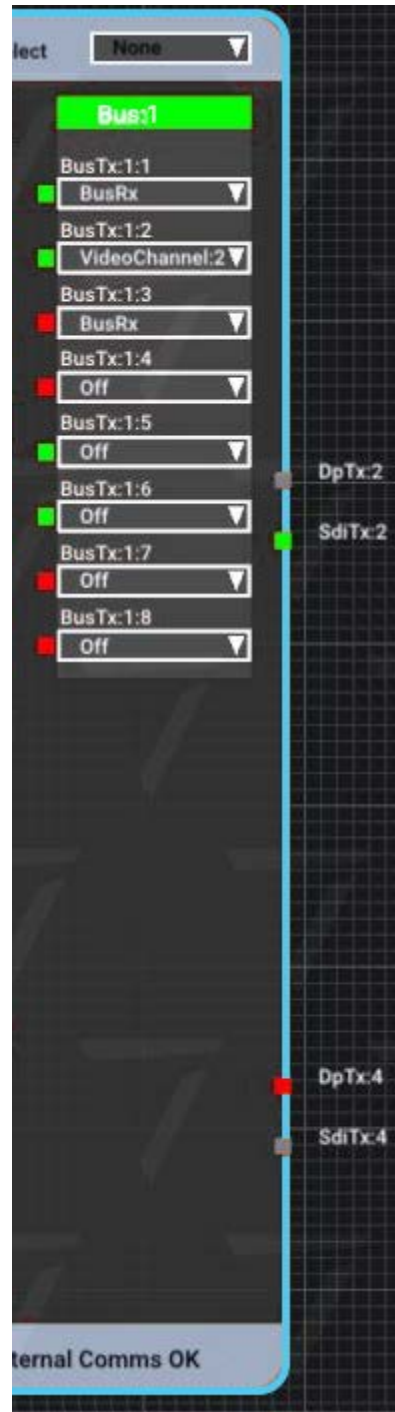
'BusRx' refers to the channels placed on the local bus, and the 'OtherBusRx' refers to the channels from the second bus. This switch would reassign the backup channels to the bus, in place of the primary channels. After this switch, within each juggler that is supporting a physical output, the processing channel is selected from the bus that is now carrying the corresponding backup input channel.

This is the first Juggler in Bank A.

Incoming video from a local DP input (e.g. VideoChannel:2) is placed onto BusTx:1:1.

This is the second Juggler in Bank A.

Incoming video from a local DP input (e.g. VideoChannel:2) is placed onto BusTx:1:2.



This is the dual-bus Juggler.

On the first bus (Bus:1) this Juggler will receive the inputs that were assigned within the Bank A Jugglers. It also receives the inputs on the second bus (Bus:2) from the Bank B Jugglers.

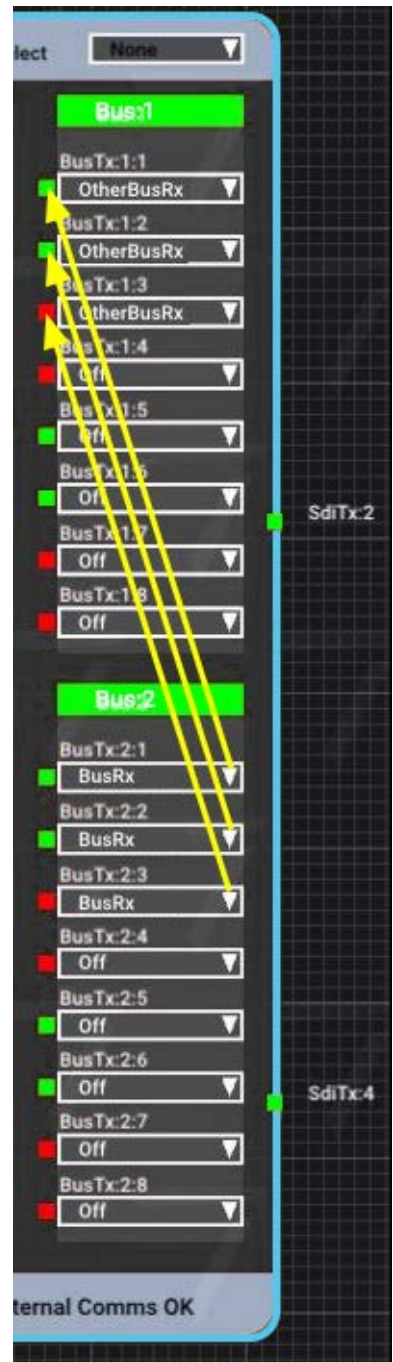
Note that it simply receives each BusRx, per bus lane, and routes to the equivalent OtherBusRx.

Bank A Jugglers are the primary units, whilst Bank B is connected as backup. IntelligentSource can be used to trigger instant switching.

The Bank A Juggler inputs are being assigned to the Bank B bus by the assignment 'OtherBusRX'. This will allow the inputs from Bank A to be available in Bank B.

The routing can be switched, so that the Bank B inputs (as BusRx) instead become available to the Bank A bus (as OtherBusRx).

This will allow the inputs from Bank B to be available in Bank A.



This applies to whatever inputs are placed on the bus of each bank, per channel process inputs, bypass streams or output canvas.

Output Canvas Bus

So far we have looked at using the bus as a transport system for taking an **input** from one Juggler and sending it to another to handle the display output. The canvas is effectively assembled at the end of the process. This has its limits as display sizes increase and the number of inserted windows multiplies.

Whilst every Juggler has four video channels, and up to four input ports and four output ports, there isn't always a match between the number of inputs required and number of outputs. For example, you may well want many input options to many simultaneous canvas windows, on a very large overall canvas, comprising a much smaller number of display devices. This will require many input signals to be processed and sent to many windows, each of which can be **anywhere** on the overall canvas, on any canvas output, at any time, whereas the physical displays need the output connectors of only a few of the Jugglers.

The OCB Difference

This is done by operating the Juggler bus system in a different way. On an [Input bus](#)¹⁰², Jugglers are described as selectively processing 'Input > Process > (Bus) > Output' – *across each Juggler video channel*. (left to right through the Juggler in the Nodes panel). In other words, inputs can be selectively and individually placed on and taken from the bus before deciding their display destination. It also means there can only be a maximum four Windows on a Canvas.

In an output canvas bus (OCB) setup, the four Juggler 'Video # Processes' (including their canvas location) are diverted automatically onto four *bus video channels*, which carry the whole canvas assembly (Canvas, with any number of Outputs and Windows) around. In an 8-lane bus, two bus lanes are assigned to each of these bus video channels*, which then feed two output connectors each. One Juggler is assigned to each bus video channel to send these outputs to its Tx ports to display.

* On a 4-lane bus, these can do the same by enabling 'dual-stream' channels.

In this mode, any number of windows (media input) can be moved anywhere on the canvas (overall display) regardless of which Juggler hosts the input media, and which is driving the actual display output.

It is essential to have an understanding of how an output canvas bus works, but to make it easier to set up, there is the [OCB Wizard](#)¹¹⁶.

Video Channel Leaders

- Allocated Jugglers take one bus video channel (only) onto two of their Tx ports.
- A juggler that takes a bus video channel output and places it on its Tx ports ‘takes the lead’ for that video channel. In that sense, a Juggler that provides the output connection for video channel 1 is a ‘leader’, one that provides the output for video channel 2 is also a ‘leader’, and so on.
- Any Juggler that provides no physical output is referred to as a ‘follower’. In reality it just has no channel lead role.
- Every ‘Video # Output’ in the system uses the same canvas output as named by the leader of that channel.

Example: Juggler A has its VideoChannel:1 property set to ‘Output Canvas Leader’ and its Video #1 Output is set to an Output named ‘Top Left’. Juggler B, C, D ... will all set their Video #1 Output also to Output ‘Top Left’.

Populating the Canvas (Canvas Panel)

A canvas comprises up to four video channels, carried over the 8 bus lanes of a ‘full bus’ system (or 4 dual streams of a 4-lane system). Each bus video channel is mapped onto two lanes, providing left and right parts for that canvas output to be delivered by a Juggler taking the lead for that bus video channel.

These paired elements of the overall display can be arranged as required. Here are three common linear arrangements, but the pairs don’t have to be aligned to each other. Names for the canvas outputs are just suggestions that will help in constructing the flows.

8 × 1

Output ‘Far Left’		Output ‘Mid Left’		Output ‘Mid Right’		Output ‘Far Right’	
VC1:Left	VC1:Right	VC2:Left	VC2:Right	VC3:Left	VC3:Right	VC4:Left	VC4:Right
B1:1	B1:2	B1:3	B1:4	B1:5	B1:6	B1:7	B1:8

4 × 2

Output ‘Top Left’		Output ‘Top Right’	
VC1:Left	VC1:Right	VC2:Left	VC2:Right
B1:1	B1:2	B1:3	B1:4
VC3:Left	VC3:Right	VC4:Left	VC4:Right
B1:5	B1:6	B1:7	B1:8
Output ‘Bottom Left’		Output ‘Bottom Right’	

2 × 4

VC1:Left	VC1:Right	Output ‘Row 1’
B1:1	B1:2	
VC2:Left	VC2:Right	Output ‘Row 2’
B1:3	B1:4	
VC3:Left	VC3:Right	Output ‘Row 3’
B1:5	B1:6	
VC4:Left	VC4:Right	Output ‘Row 4’
B1:7	B1:8	

When a canvas output is now added onto the canvas in the Compere Canvas panel, and then linked to one of these dual video channels in a Juggler node, it will represent the two bus lanes – i.e. as wide

as two outputs. The left half of this will go to one Tx port of the lead for the video channel, the right half to another Tx port.

Example: the 4 × 2 arrangement above:



A similar output canvas mode is available to 4-lane bus systems, but instead of each video channel connecting to two bus lanes, they each connect to one. In effect this is as above, but with only a single output. By applying a further 'dual stream' mode, this also can be split into two outputs.

Output Canvas Bus Workflow (manual method)

To set up an OCB system in this way, requires a different workflow from the [Input bus](#)⁽¹⁰²⁾. A schematic diagram is important to work from, with systematic naming of Jugglers, canvas elements and output devices. In this way, between diagram, physical racking and Compere, you will more easily see the sequential arrangement of all the components.

Add the Jugglers to the Nodes panel

- Drag all required Jugglers in the network from the Project panel into the Nodes panel.
- Name the Jugglers for their roles, e.g. Juggler3456 Ldr1 (this Juggler is leader for video channel 1), Juggler3457 F (this Juggler is a follower only).
- Arrange the Jugglers logically. For example, you may want to lay out the leader Jugglers at the top or group them separately.

Connect all inputs

- Connect each input pin on every Juggler to its corresponding video channel. *Example:* DpRx1 to Video #1 Input, Process = DpRx1:



Establish the Canvas

- In Compere, drag a Canvas from Resources into the Canvas panel.
- Size the Canvas in its 'Canvas Bounds' properties for the overall pixel resolution of the whole display area.
- In the Canvas main properties, check 'Output Canvas Bus'. This identifies all its child Outputs and Windows as coming from the bus.
- For 4-lane bus configurations with the same left/right output pairing, select 'OCB Dual Stream'. Otherwise all the bus video channel will go to a single rather than paired output.
- From Resources, add the required number of Outputs onto the Canvas to fill/fit the Canvas, bearing in mind that each bus video channel will provide 2 Tx feeds, 'Left' and 'Right' (i.e. double width).

Do not manually size these outputs, but it can help to name them for their positions (e.g. 'Output top left' rather than default 'Output 1' – see the illustration, above).

- Paired outputs? An odd number of display devices means that one half of one of the outputs will be used. This unused part will simply lie off the canvas.

Set each bus mode to 'Output Canvas'

- For all these Jugglers, and every bus on each (you can multi-select them all in the Project panel) set the bus properties to 'Output Canvas Bus'. This will change the way the whole bus is connected and operates.
- The bus drop-down menus will now be **auto-populated** with their set channels. *Example:* for an 8-lane bus system:

BusTx:1:1 and BusTx:1:2 carry Video Channel 1 Output

BusTx:1:3 and BusTx:1:4 carry Video Channel 2 Output

BusTx:1:5 and BusTx:1:6 carry Video Channel 3 Output

BusTx:1:7 and BusTx:1:8 carry Video Channel 4 Output

Do not change any of these auto settings.

Which Jugglers will be providing the physical output connections?

One Juggler must be assigned to lead each bus video channel. *Example:* select VideoChannel:1 from Juggler leader1 in the project panel. In its properties, check 'Output Canvas Leader'.

- Refer to your schematic drawing for which Leader Juggler is feeding each display (e.g. from left to right)
- On the Leader Jugglers, assign one video channel that will be connected to an output device, to 'Output Canvas Leader' in the properties.

Example:

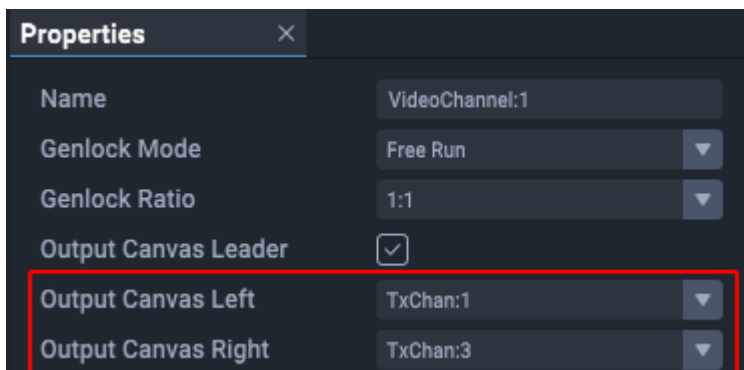
Juggler Ldr1 claims Video Channel 1 as Leader, taking Bus lanes 1 & 2 (other channels available as Follower)

Juggler Ldr2 claims Video Channel 2 as Leader, taking Bus lanes 3 & 4 (other channels available as Follower)

Juggler Ldr3 claims Video Channel 3 as Leader, taking Bus lanes 5 & 6 (other channels available as Follower)

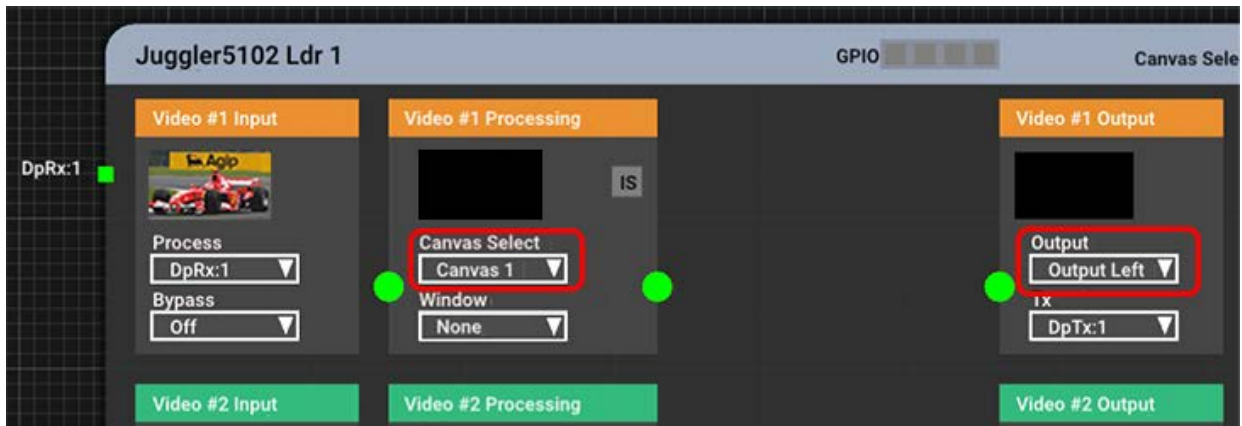
Juggler Ldr4 claims Video Channel 4 as Leader, taking Bus lanes 7 & 8 (other channels available as Follower)

- Set up/check the EDID of all these Juggler leader output connectors. These may be custom-supplied to suit the particular configuration.
- In the properties for each of these channel leads, assign 'Output Canvas Left' and 'Output Canvas Right' in their drop-downs, to the output connectors for that Juggler (e.g. TxChan:1 and TxChan:3):



Route the lead Juggler outputs to the canvas outputs

- Find Juggler video channel 1 leader in Nodes, and in Video #1 Processing assign the Canvas, and select the required canvas output name in Video #1 Output. (Ignore the Tx box.)



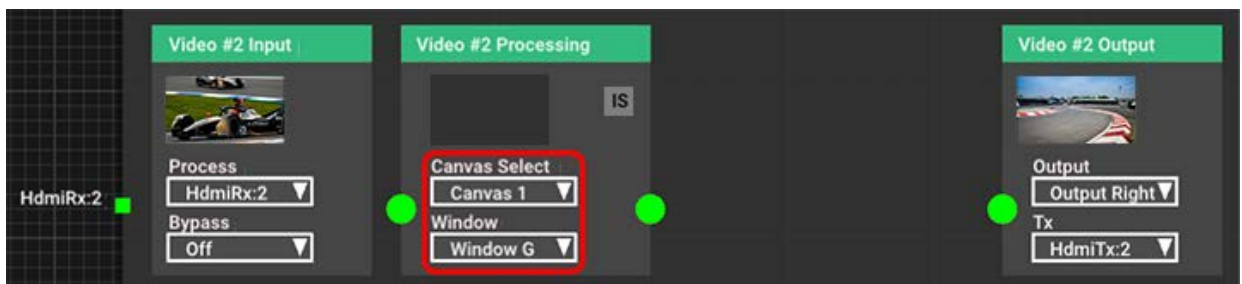
- The canvas output in the Canvas panel will now have assumed its dual (left/right) dimensions, so now locate it correctly in the canvas.
- Do the same for the Juggler leading video channel 2, and assign Video #2 Processing to the Canvas and Video #2 Output to its required output name.
- The canvas output will now have assumed its dual (left/right) dimensions, so now located this correctly in the canvas.
- Repeat through all leader Jugglers until all video channels are covered.

Route all other Juggler non-lead outputs

- In all Jugglers, each remaining video channel needs to be assigned to the canvas. Do this irrespective of any active input on that Juggler’s video channels.
- In every Juggler where the Juggler is not managing (leading) that channel, set ‘Video # Output’ in the output drop-down to the same output name that the leader for that channel number is managing. *Example:* channel lead 1 is addressing ‘Output Top Left’, so in the ‘Video #1 Output’ block for all other Jugglers, also select ‘Output Top Left’ from the Output drop-down.

Add the media windows to the canvas

- Add Windows to the canvas as required. Name, size and position these as in any Compere system. These reflect the all the inputs you wish to display. They may have preset positions or groups, but each can now be moved seamlessly anywhere on the entire canvas.
- Assign these named windows as required to Juggler inputs (under Video # Processing):



- Windows can be moved on or off the Canvas; when off, their content will not appear on the display. (See also: [Create and Recall Canvas Arrangements](#)⁽⁷⁰⁾.)

Since the whole canvas is on the bus, all window information is carried around with it, so unlike the Input bus mode, the media is not going to the same Juggler's output (on the right in the node), but straight onto the bus.

OCB Wizard

When you have understood how and why the [Output Canvas Bus](#)⁽¹¹⁰⁾ works, the Wizard can take you through all required settings for a new project.

The workflow for an OCB system is similar to that for an input bus system (see [Workflow: Create a Juggler System](#)⁽⁹⁾) but the wizard will set up the canvas and make the bus and output connections for you. It will also automatically switch you into OCB mode.

Preparation Checklist

- ✓ The Juggler system, inputs, outputs and bus, is fully wired.
- ✓ You will already have configured your network, Project Group membership and Project Group server.
- ✓ All required Jugglers in the Project Group appear in the Project panel.
- ✓ You have added all these Jugglers to the Nodes panel, set up their Rx and Tx EDIDs, connected video inputs to sources, and (optionally) renamed them in some way to easily identify their roles, especially those you want to be video channel leaders.

Step 1: The Canvas

Context: see [Canvas: Configure a Display](#)⁽⁶⁰⁾.

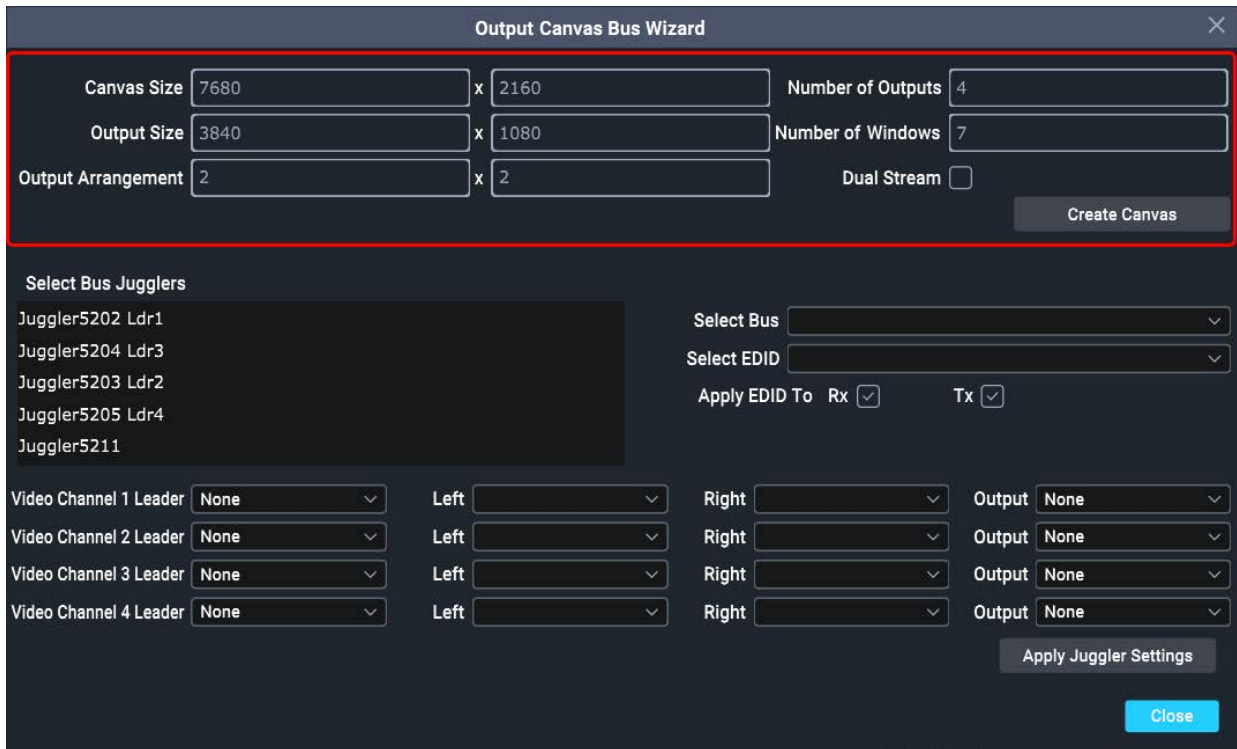
This part of the wizard will create your basic display configuration in the Project, and requires that there is no Canvas already in the Project panel. It will not delete, reset or edit an existing canvas.

Once created, the normal advantages of naming your canvas outputs and windows apply, so that you can easily identify all the elements and match them to any system schematic drawing.

Enter the overall canvas size, the number of canvas outputs (paired, so double width) and their arrangement (e.g. 3 × 3, 4 × 2, 1 × 4).

- Enter the common resolution for all the canvas outputs (double the width to accommodate the pairing on the bus).
- The number of windows is less important; these can be added and changed afterwards. This will simply add a number of unconfigured canvas windows, or none.
- If your bus is a 4-lane ('thin') bus, check 'Dual Stream' to create the paired outputs. This is unnecessary for 8-lane ('full') bus systems.

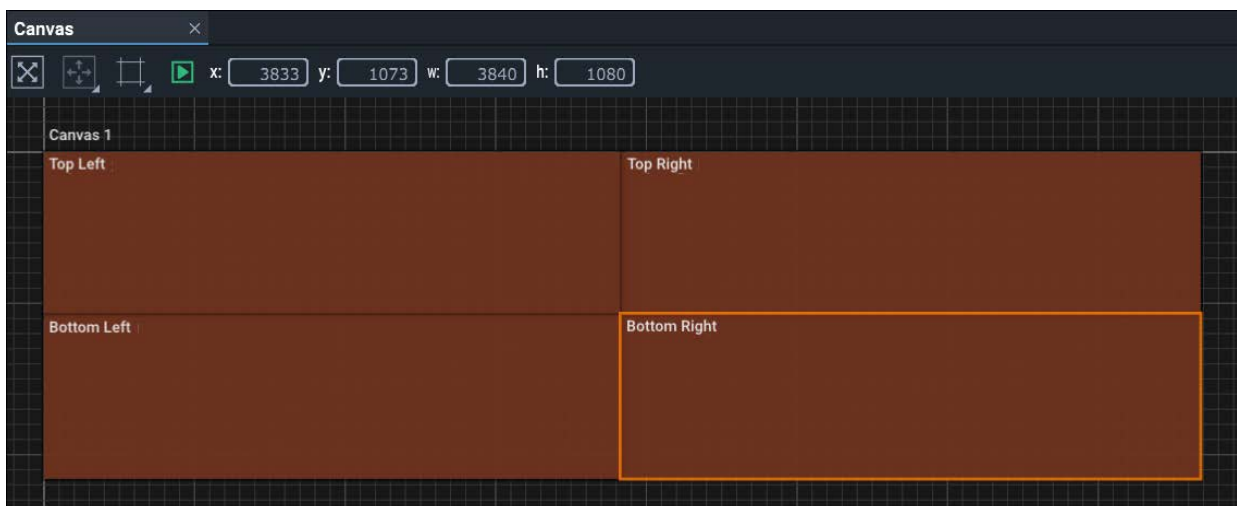
- Click 'Create Canvas'.



Notes:

Each OCB output is *per video channel* and comprises a pair of 'Left' and 'Right' parts. The above example has a 2 x 2 arrangement.

- You can close the Wizard for now.
- Drag the new **Canvas** into the Canvas panel.

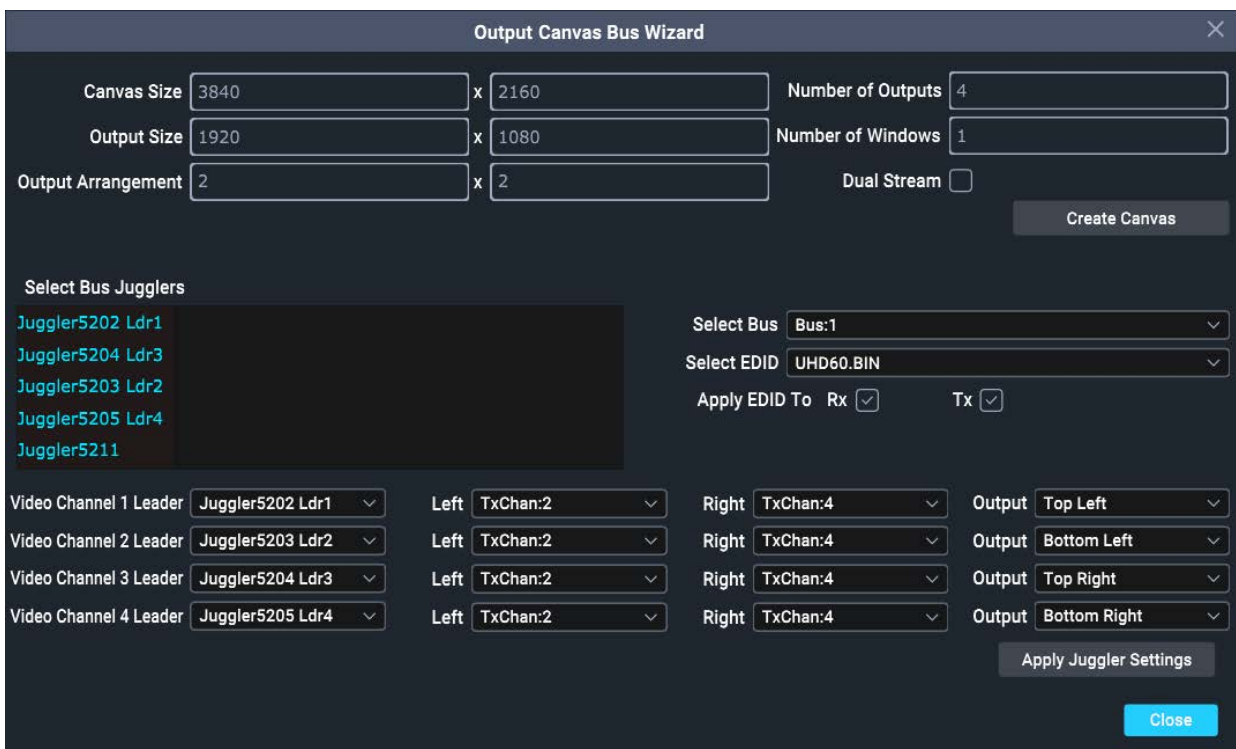


- At this point, it can be very helpful to rename your **canvas outputs**, for example by their location (Top Left, Top Middle etc.).
- Add, name, position and resize any **canvas windows** now if you wish, but this can be done at any time later. Windows are not invoked by the OCB wizard.

Step 2: Connecting Jugglers to the bus

This part of the wizard connects all Jugglers connected to a canvas, and assigns up to four Jugglers to provide the Tx ports out for each canvas output.

- Reopen the wizard.
- Ignore any default numbers appearing in the canvas section – you will not be using this again.
- Select all Jugglers that are on a common bus.
- Select the bus they belong to (typically this will be Bus:1).
- If necessary, select the EDID for the Tx and Rx ports that the Leader Jugglers use.
- Now assign a Leader Juggler to each active video channel, up to four. If you have named these Jugglers already (as in this example) this will help.
- Select the named canvas output to attach to each of the bus video channels. Every non-lead Juggler will be given the same association between canvas output name and Video # Output. (In the illustration below, every Juggler will have 'Top Left' in its Video#1 Output, 'Bottom Left' in its Video#2 Output, and so on.)
- Select the two Tx ports on each Video Channel Leader that connect the left/right pair of the canvas output to output devices:



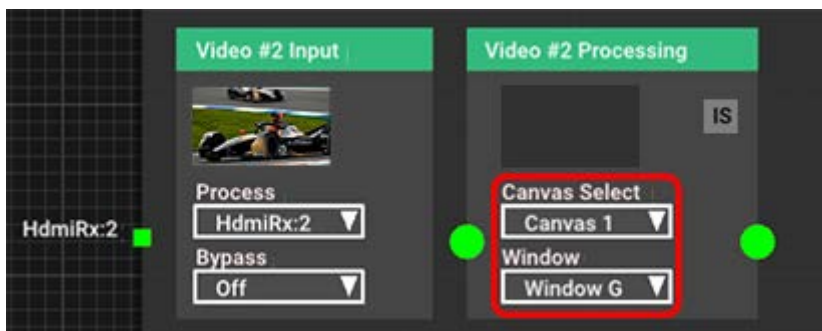
- Click 'Apply Juggler Settings'. Updating Jugglers is not instant: you will see an alert to give a moment before proceeding further.
- Close the wizard.

- All Jugglers are now addressing the same Canvas, and their video channel outputs correspond to those of the Leaders.

Step 2: Add, configure, assign windows

This step is the same as in the final manual workflow step in [Output Canvas Bus](#)⁽¹¹⁰⁾, above.

- Whether you added windows via the wizard or not, now make sure you have all the windows you need for the project, name them, size and locate them.
- For each Juggler video channel with an input, in Video # Processing select the window on which to display the media:



Connecting Inputs and Outputs

In a windowing Juggler system, you will have set up your [display Canvas](#)⁽⁶⁰⁾, with a number of Canvas Outputs and Canvas Windows. The Outputs can be named or numbered to identify the area of Canvas on which you may want to display media. Any number of Windows can be positioned and scaled dynamically anywhere on the Canvas, across the boundaries of the Canvas Outputs, to which media streams can be assigned.

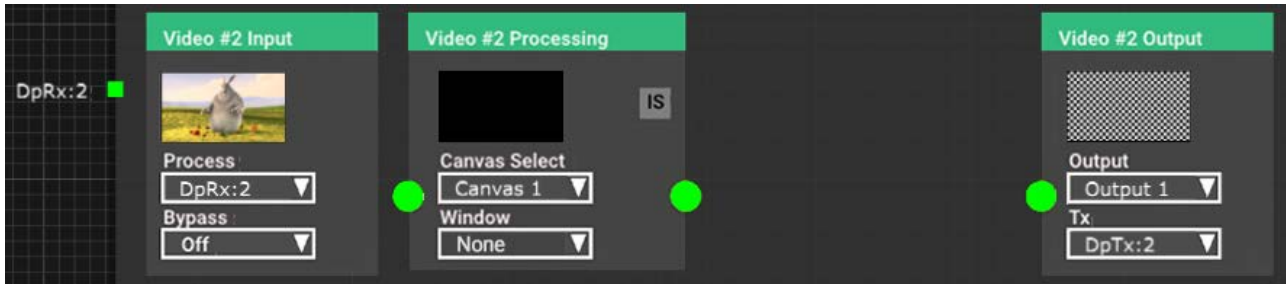
Throughput for a single Juggler

In the top bar of the Juggler, select the Canvas name that this Juggler will address. This list is populated when Canvases are created and named. Selecting here will populate 'Canvas Select' in all available video channels for the Juggler.

Juggler input pins, on the left edge of the Juggler, are named by type, e.g. **DpRx : 2**, meaning a DisplayPort input on channel 2 of the Juggler. If no valid media are present on that physical port, it will be red. If valid media is present, it will be green.

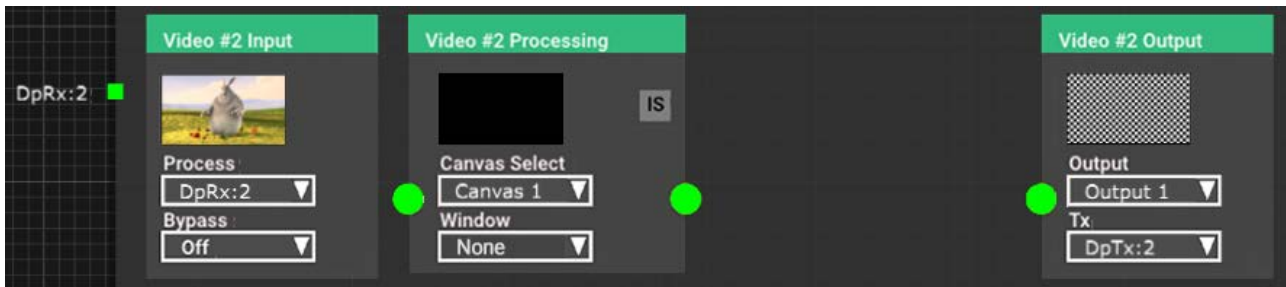
From the list of available feeds in the **Video #2 Input** 'Process' selector, add an input. The Process block will now be linked to the **Video #2 Processing** block. If valid media is present, pins and links will be green. If you have 'Enable NDI Confidence Monitors' selected in the Properties of each Juggler, this will help verify that a media stream is connecting correctly. NDI monitors can be seen in the video channel input and output blocks of a Juggler in Compere, and in the Canvas assembly panel. This

illustration confirms that video is being received from the named input and video is being presented to the listed Tx pin – in this case the canvas as a whole has a test pattern.

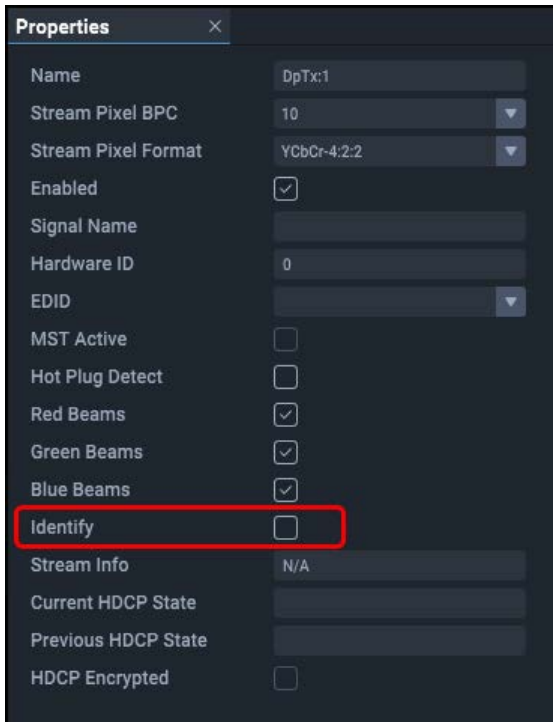


In the Processing block, you decide where this is to go next. There are other processes you can add, but for now, just ensure you have the canvas you want to use.

In the Video #2 Output block for this channel, ensure the **Tx** has the appropriate destination port selected. Using the Output selector, choose which Canvas Output area is required. The list is populated as Canvases are added to the Canvas assembly panel. This will remove the direct link to the Output block because in order to display the media we need to say where on the overall Canvas it is to appear, i.e. in a Canvas Window. You can use a Juggler test pattern as a background of the output when media is not present:

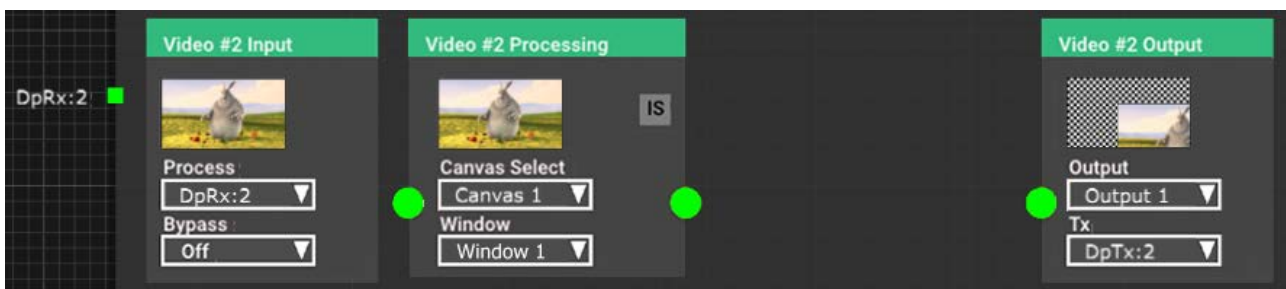


Reminder: You can identify a Juggler output in the real world by checking the Identify box in the Juggler output properties:



This checkbox is 'push-to-show', and will temporarily cause the output device to flash red and then green.

Now assign the video channel to a [Canvas Window](#)⁶⁷ in the Video Processing block. The selector list is populated by the addition of Windows in the Canvas assembly panel. Canvas windows can be positioned anywhere in the Canvas, so what you see on the output display will require this Window to be positioned somewhere over the selected Canvas Output for this video channel, as here:



Now we have NDI showing an input signal is being received, that it is being presented to a named window, and where it is being displayed on the named output.

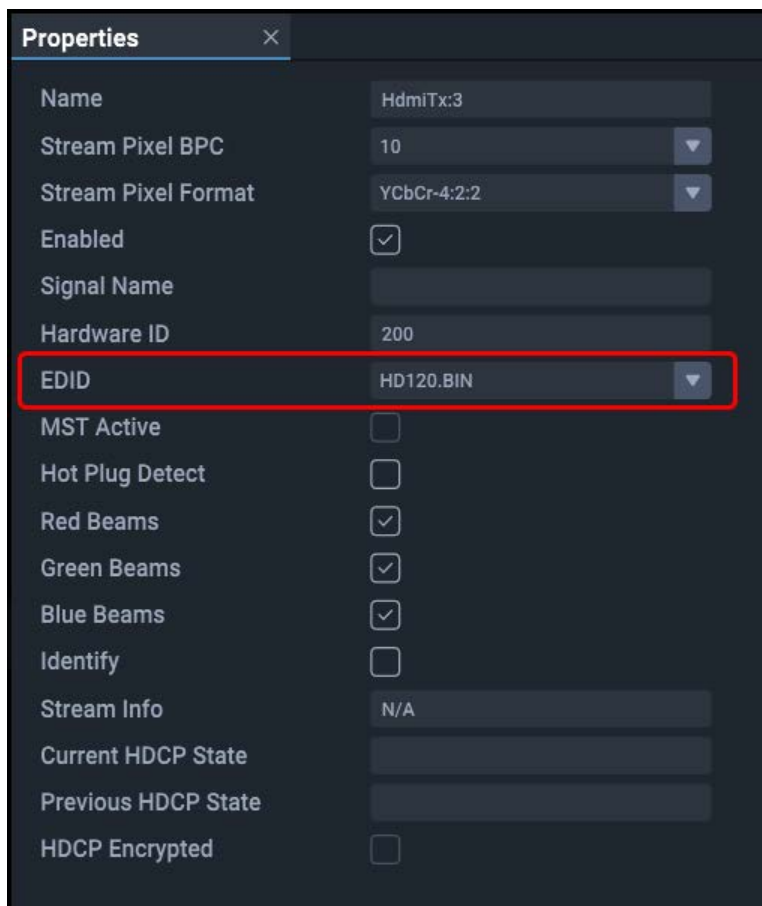
EDID Spoofing

When an Input or Output is assigned to a Juggler video connector, it adopts the EDID of that connector.

Juggler supports spoofing of EDIDs onto DisplayPort and HDMI inputs and outputs via the 7thSense Compere software.

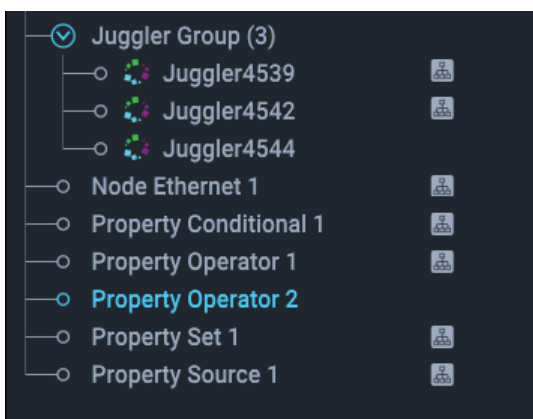
Each Juggler comes with a list of standard EDIDs. Additional EDIDs can be loaded into the Juggler Module(s) via the Compere interface.

For each Juggler output in the Nodes assembly panel (described 'AVout#' per Juggler) select the appropriate Tx EDID from The Properties panel > EDID list:

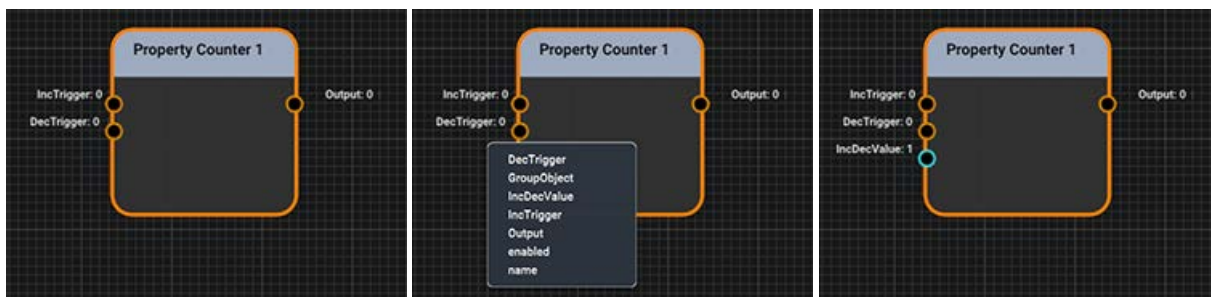


Nodes: Processing Nodes

Processing nodes include sources and values, and logic processes. Flexible and powerful, they can be combined to create more complex interactions in Juggler systems including automated events such as IntelligentSource™ failover. They are dragged from the Resources panel into a Nodes panel and arranged or selected in the same way as Juggler nodes. They become listed to the Project panel, and when selected, Properties are shown for edit in the Properties panel. Nodes in the Nodes panel are marked, just as Jugglers are, by the icon. This does not mean they are inactive, since a node removed from the Nodes panel remains in the project with all its properties.



- Nodes can be **removed** from (Right-click menu) from the Nodes panel but remain an interactive part of the Project, losing the 'displayed' icon in the Project panel. They can only be **deleted** from the Project panel.
- **Pins** available to a processing node can be added to what is otherwise a minimalist representation, as required. Select the node, Shift+Right click for properties to add, and select the pin to add:



- Nodes have default **names** denoting their function but can be named for their purpose. The name appears in the title bar of the node.
- Those **prefixed** 'Node' initiate things, those prefixed 'Property' relate things via a process.
- Pins shown in orange take **integer** values, those taking **strings** are green, and blue indicated **double** values.
- Pins in a Nodes panel are **linked** by Shift+Left mouse dragging from source pin to target pin. In this way, values are communicated, transformed, used as triggers for other events, and so on.
- Links are **deleted** by hovering over the target pin and pressing Delete.

- Pin value **colour**: pins that exert control show yellow in the Properties panel, those that are being controlled show green.
- [Presets and Clones](#)¹²⁹ can be created from selected nodes.

Nodes that do things

Node Ethernet

Description

An Ethernet receiver or transmitter. Used for external control messaging within a Compere system, over UDP or TCP.

In client mode this is a message sender, in server mode it is a message receiver.

Properties

- Name: defaults to node type, rename for function as required
- Connected: can be switched on or off by giving the value 1 = yes.0 = no
- Host name: name or IP address of source to use
- Input: as server, string message received
- Output: as client, string to send by UDP/TCP
- Port: default: 23
- Type: UDP, or TCP Client (send a string) or TCP Server (receive a string)
- Input – remove end of line chars from input strings
- Output – remove end of line chars from output strings
- UDP read port: default 7781

Node Generator

Description

Generates a waveform output. Constantly changing property values might include a position, colour or intensity, for example. The timing source is that of the server Juggler.

Properties

- Name: defaults to node type, rename for function as required
- Amplitude (relative units)

- Frequency (Hz)
- Mode: sine or square
- Phase: shift in degrees (180 is inverted)
- Time:
- Output: amplitude value

Node Logger

Description

Receives an input to append events to a named log file in a named location.

Properties

- Name: defaults to node type, rename for function as required
- Directory: log file destination
- Filename: log file name
- Input: link to a node output being tracked for changes

Nodes that relate and process values of other things

Property Conditional

Description

Connects to a configurable number of integer inputs, any one of which can be selected and redirected to an output, or overridden by a set value.

Properties

- Name: defaults to node type, rename for function as required
- Number of Inputs: add number required
- Switch Index: the number of the input to watch (**output** becomes the value of this input)

Alternative mode

- Priority Mode: output **value** becomes the **number** of lowest number input to be non-zero.
- Priority Latch: **number** of the input to latch onto as output **value**, even if lower order inputs become valid again.

Property Counter

Description

Receives triggers that match set values to increment or decrement the output by a step value.

Properties

- Name: defaults to node type, rename for function as required
- Increment Trigger: set the value to compare with an input, and when it matches, increment the output
- Decrement Trigger: set the value to compare with an input, and when it matches, decrement the output
- Increment / Decrement Value: set the value by which to increment or decrement the current output when triggered.

Property Operator

Description

Configurable integer logic operator based on a flexible number of inputs with selectable mode to generate a numeric output value.

Properties

- Name: defaults to node type, rename for function as required
- Number of Inputs: add number required
- Input: auto-labelled Input_# as generated
- Mode: AND, OR, NOR, NOT, INVERT, ADD, SUBTRACT
- Output: integer value to be output

Examples:

AND : 3 inputs all with the same value, output = 1, else output = 0

OR : 3 inputs with different values, output = the greatest value, or use as boolean operator on values of 0 and 1

NOR : 3 inputs, only when all inputs are 0 is output value = 1, else output = 0

NOT : 2 inputs (only) when both are 0, output = 1, else output = 0

INVERT : 1 input (only) changes the sign of any value in input 1

ADD : output = sum of all inputs

SUBTRACT : output = sum of all other inputs subtracted from the value of input 1

Property Set

Description

Apply values to destinations (paths) when triggered by a set value.

Properties

- Name: defaults to node type, rename for function as required
- Input Trigger: receives an input and is set to state 0 unless the input value matches the trigger value, when it switches to state 1.
- Trigger Value: enter the value at which the trigger will switch.
- Number of Outputs: add number required. Generates a [Path] and [Output value] pair for each.
- Path #: auto-labelled Path_# as generated. Enter the property paths required. Paired with:
- Output #: auto-labelled Output_# as generated. Sends this value to the paired path when trigger value is 1.

Property Source

Description

Creates string outputs with values derived from paths. The path of any property in Compere can be copied using Shift+Right click on the property label in the Properties panel.

Properties

- Name: defaults to node type, rename for function as required
- Number of Outputs: add number required. Generates a Path and Output pair.
- Path #: auto-labelled Path_# as generated, paired with:
- Output #: auto-labelled Output_# as generated; adopts the value from the paired path.

Presets and Clones

All Compere project information is held in blocks of xml. This means that you can save, recall and distribute a complete show setup with all its players and all their parts. It also means that you can save just the parts, which might mean the way you set up a canvas populated with just outputs, or just a set of windows with their sizes and positions, or a group of Jugglers configured in a particular way, and so on.

Elements in presets and clones are selected copied from the Project panel, not from what is placed and visible in the assembly panels.

Presets: file type .pre

If you want to use these same things in the same project, you can use these xml blocks to recall the values, which can be particularly useful in switching between various preset window layouts ([Create and Recall Canvas Arrangements](#)⁽⁷⁰⁾).

Presets capture arrangements and settings for components in a specific Project. When recalled to the same Project, nothing is added to the Project, but dimensions and locations etc. of existing project components will be set again to those of the saved Preset.

Presets cannot be used in 'foreign' projects.

Applying presets to a selection

Example use: to save repetitious work in configuring many Jugglers, this option allows the core configuration properties of one Juggler of a kind to be saved and recalled into many selected Jugglers of the same type in the Nodes panel, not just to Jugglers of matched UUIDs. Non-transferrable properties such as Name are not copied over.

Clones: file type .clo

If instead you want to create standardised components, for example a display wall of a number of UHD panels, with or without windows, these can be saved as clones. They are no longer a member of any project, but can be loaded as new elements into a different project, from where they can be added into assembly panels and incorporated and repurposed as necessary.

Unlike presets, clones are saved without their project UUIDs. This is why they are 'loaded' (new), not 'recalled' (remembered).

Options

Save As: Selection

This option in both types will save out just those items selected in the Project panel. If the selected items require a parent element (e.g. a Window or Output) a default parent will be inserted into the clone file. Child elements cannot be transferred to a different canvas when reloaded. When using canvas clones, it is best to 'save as canvas'.

The exception would be that if you already have more than one canvas, you may want a clone of just one of them.

Save As: Canvasses

This option will save out every canvas parent and every child output and window. No selection is needed.

Export

Whilst the save options will direct you to the preset and clone default locations in the Compere root location (set in Preferences), export will direct you to local locations on the PC using the Compere UI at the time.

Ports Used by Juggler and Compere

Function	Configurable	Port
Compere UDP	Default, editable via comms.ini or in Preferences	9999
Compere TCP	Default, editable via comms.ini or in Preferences	5555 5553 (early versions of Compere)
Broadcast Logging UDP	Default, editable via comms.ini or in Preferences	5596
Juggler 1 Console TCP	Fixed	9013
Juggler 2 TCP receive	Fixed	9016
Juggler 2 TCP transmit	Fixed	9017
Multicast discovery UDP	Fixed	5353
External control comms, default UDP receive port	Yes, in Preferences	8001
External control comms, default UDP transmit port	Yes, in Preferences	8002
External control adapter TCP port	Yes, in Preferences	5584
UDP Console Rx	Reserved	9004
UDP Console Tx	Reserved	9005
Compere Discovery (Rx)	Fixed	9101
Compere Discovery (Tx)	Fixed	9102
UDP Remote Discovery	Yes, in Preferences	9103
Time Server	Default, editable in Preferences	18888
Front Panel touch display	Fixed	5000

It is important to make sure the UDP and TCP ports match between all instance of Compere in a system that need to communicate with one another.

ASCII / XML External Control

This guide covers the ASCII / XML protocol of external control introduced with the original Juggler pixel processor.

Principle

Every element of a Compere Project (the branching tree you see in the Project panel) has a unique identifier (UUID) and a common text name that you give to it. The same is true for every parameter that you can see, and may want to change, in the Properties panel for that element.

TCP and UDP ports are defined in *Settings > Preferences*.

- Use a **remote terminal client** e.g. PuTTY, or Tera Term, on a raw (or raw ASCII over TCP) connection, to address Compere on the **server** for a project group.
- The **IP of a target Juggler** can be found in *Configuration > Network Configuration > [Discovery]*.
- For TCP, use Port 2300.
- For UDP, first enable 'Use UDP Console Connection' in *Settings > Juggler*. UDP receive port is 8001, UDP Remote Transmit port is 8002.

Expressions in this guide

Elements shown in [\[SquareBrackets\]](#) indicate that the string is replaced by a user string, e.g. [\\[JugglerName\]\](#) might typically be the serial name, [\Juggler5678\](#).

Similarly, [C:\\[Compere Root Path\]\](#) is almost always [C:\Users\\[UserName\]\AppData\Local\compere\](#).

Protocol

In the Project Panel you see the structure of the Project file that contains all the instructions for the entire setup of a Juggler system.

From this information you can see that there is a route to the value of everything. In the same way as we use paths to files in folders on devices on a network, you can send commands, to find or change any value. As an example we might want, in rough terms, to do this:

set the value to '10' for the x-axis position of a projector in this project

which as a command would look something like this:

```
set path="/GroupSet/Group/Projector 1/position/x" value="10" (in plain text)
```

```
<command cmd="set" path="/GroupSet/Group/Projector 1/position/x" value="10"/> (in xml)
```

The message format (plain text or xml) is detected and the response is returned in the same format. All recognised commands respond immediately with: `success=true/false`.

For commands that do not have an instant response (e.g. `open`, or `"Import MPCDI"`) the command must be followed by a system status request.

These commands or controls can be sent in either plain text or XML to the IP address of the **server Juggler** in a project group, or to a Windows control PC as client or server.

Compere supports commands sent via TCP. Use the Compere Preferences Panel to configure IP addresses and transmit/receive ports, including the connection timeout.

Commands and their parameters are case insensitive. Each parameter's actual *value* is case sensitive. Individual commands are activated in text strings by line ending characters `\n`, or in xml, by opening and closing tags `<command> </command>` or, for empty tags, `<command />`.

Grouping Commands

It can be more efficient to encapsulate many SET commands in a single group using the `<commands></commands>` tag.

For example to position two projectors in a single control message:

```
<commands>
<command cmd="set" path="/GroupSet/Group/proj1/position/x" value="0.2"/>
<command cmd="set" path="/GroupSet/Group/proj1/position/y" value="0.3"/>
<command cmd="set" path="/GroupSet/Group/proj1/position/z" value="0.4"/>
<command cmd="set" path="/GroupSet/Group/proj1/heading/az" value="0.5"/>
<command cmd="set" path="/GroupSet/Group/proj1/heading/el" value="0.6"/>
<command cmd="set" path="/GroupSet/Group/proj1/heading/roll" value="0.7"/>
<command cmd="set" path="/GroupSet/Group/proj2/position/x" value="0.2"/>
<command cmd="set" path="/GroupSet/Group/proj2/position/y" value="0.3"/>
<command cmd="set" path="/GroupSet/Group/proj2/position/z" value="0.4"/>
<command cmd="set" path="/GroupSet/Group/proj2/heading/az" value="0.5"/>
<command cmd="set" path="/GroupSet/Group/proj2/heading/el" value="0.6"/>
<command cmd="set" path="/GroupSet/Group/proj2/heading/roll" value="0.7"/>
</commands>
```

The reply would be:

```
<group success="true" numberOfCommands="12" atomic="true" commandDuration="8111us"/>
```

Note: 'atomic' in this reply indicates that the **commands** group was not to be distributed to the other members of the Compere Project Group until all commands had been received by the instance of Compere addressed.

In this example, projector movement will therefore be smoother. The attribute `atomic="true"` is implicit, but `atomic="false"` can be used if you want commands to be distributed one at a time as received, i.e. while other commands are still being received.

Explicit example:

```
<commands atomic="true">
<command cmd="set" path="/GroupSet/Group/proj1/position/x" value="0.2"/>
...
</commands>
```

The plain text equivalent would be like this:

```
group commands="GroupSet/Group/proj1/position/x=0.2;
[...];GroupSet/Group/proj2/heading/roll=0.2"
```

Scope of external control

With complex multi-Juggler systems, External Control is the route to both systems management, and dynamic interaction with all the elements. Whether used directly or through show controllers, it removes the need to use Compere for complex daily interaction with devices, models, projectors, sources, processes and interconnections.

Static and Dynamic commands

There are two types of external control commands, **static** and **dynamic**.

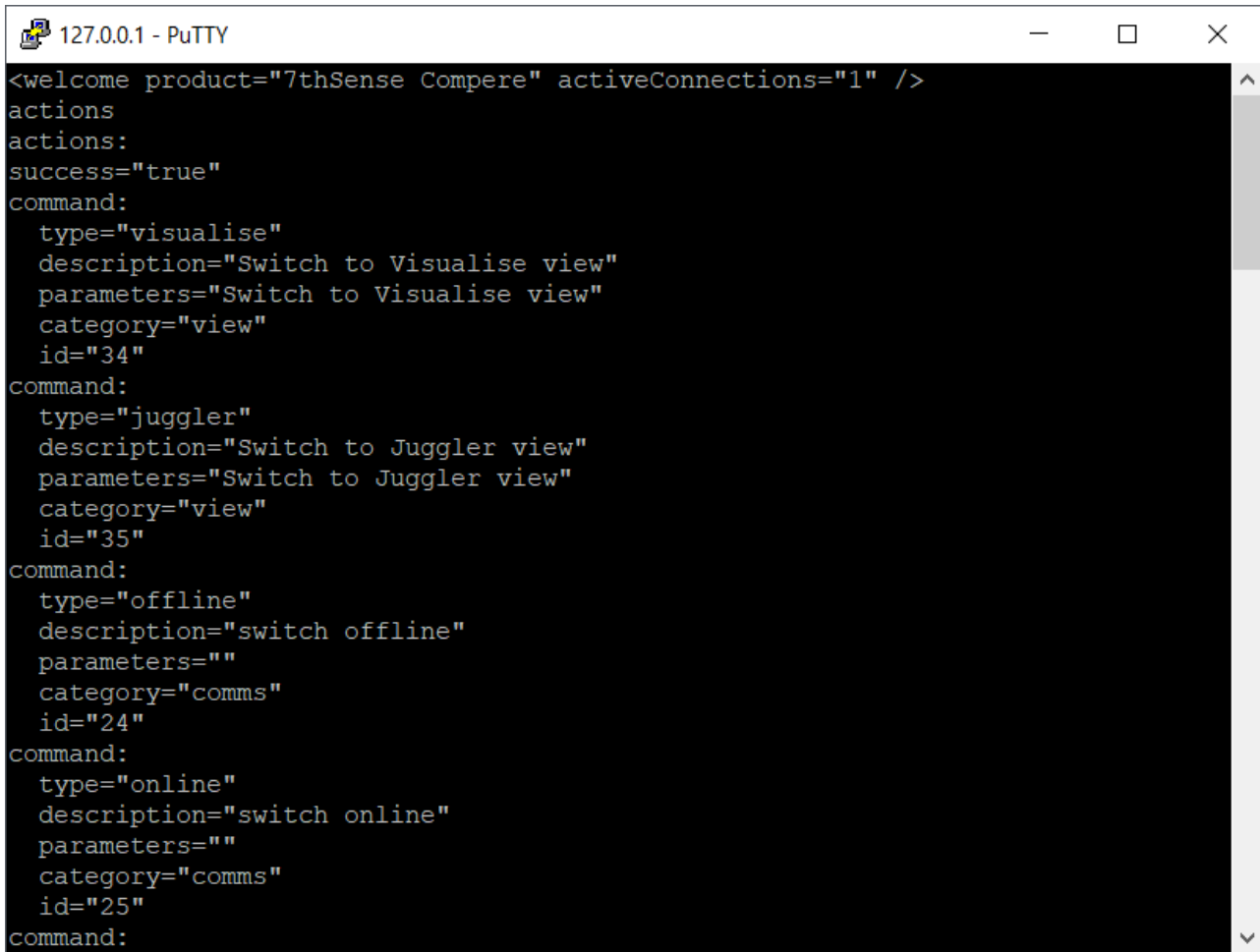
Static, or core, commands interact with a Compere Project, load and save Project files, and request specific notifications (registers). For example to `set` a value of an object property.

- The full list of core commands is available within Compere, *Help > External Control Protocol*.
- For the latest version, on this Portal, see [ExternalControlHelp.html](#)

Dynamic (action) commands are less generic and go beyond changing properties in the Project in order to do something. These are detailed with examples in the following pages of this guide.

Available actions

Type "actions" to list all actions available for the target instance of Compere in its current state. This will return a series of actions of name `type=` under headings of `command`:

A screenshot of a PuTTY terminal window titled "127.0.0.1 - PuTTY". The terminal displays XML configuration for actions. The root element is <welcome product="7thSense Compere" activeConnections="1" />. Below it, there are four <actions> elements, each with a <command> sub-element. The first action has type="visualise", description="Switch to Visualise view", parameters="Switch to Visualise view", category="view", and id="34". The second action has type="juggler", description="Switch to Juggler view", parameters="Switch to Juggler view", category="view", and id="35". The third action has type="offline", description="switch offline", parameters="", category="comms", and id="24". The fourth action has type="online", description="switch online", parameters="", category="comms", and id="25". The terminal text is as follows:

```
<welcome product="7thSense Compere" activeConnections="1" />
actions
actions:
success="true"
command:
  type="visualise"
  description="Switch to Visualise view"
  parameters="Switch to Visualise view"
  category="view"
  id="34"
command:
  type="juggler"
  description="Switch to Juggler view"
  parameters="Switch to Juggler view"
  category="view"
  id="35"
command:
  type="offline"
  description="switch offline"
  parameters=""
  category="comms"
  id="24"
command:
  type="online"
  description="switch online"
  parameters=""
  category="comms"
  id="25"
command:
```

To use an action:

`actions type="[name]"` enacts a specific action of this name

Example

Import an MPCDI file from a given location:

`actions type="import MPCDI" file="C:[MyPath]\MPCDI\example.mpcdi"`

Real use case example:

`actions type="Import MPCDI" file="C:\Users\[UserName]\Downloads\MyFile01.mpcdi"`

Options

Any of the commands below could be used to set the x position of a Canvas Window value to 800:

Using names

```
<command cmd="set" path="/GroupSet/Group/Canvas 1/Window 1/canvasposition/x" value="800"/>
```

Using class names (i.e. non-specific)

The element `path="juggler"`, for example, addresses all objects of type 'juggler'.

```
<command cmd="set" path="/GroupSet/Group/juggler"/>
```

Using a mixture of [wildcards](#)¹³⁶ (*), names (specific) and class names (non-specific)

```
<command cmd="set" path="*/*/*/Window 1/canvasposition/x" value="800" />
```

Wildcards

Matching a name between slashes

Command

```
<command cmd="get" path="[JugglerName]*/health"/>
```

Reply

```
<replies success="true" path="[JugglerName]*/health" numberofreplies="1"
  commandDuration="123us">
```

```
<health value="~RxLane:1-LaneDown~RxLane:2-LaneDown~RxLane:3-
  LaneDown~RxLane:4-LaneDown~" fullpath="GroupSet/Group/
  [JugglerName]/Bus:1/health"/>
```

```
</replies>
```

Absolute versus relative paths

Relative : `<command cmd="get" path="[JugglerName]/DpRx:2/VideoTiming"/>`

Absolute: `<command cmd="get" path="/GroupSet/Group/[JugglerName]/DpRx:2/VideoTiming"/>`

It is possible to use a relative path, offering a shortcut for an absolute path. Compere will perform an exhaustive search so there is a performance overhead depending on the extent of the Juggler system. Additionally, multiple results may be returned: searching for "name" will return every absolute path that matches.

The full path is returned in replies, and in commands, the leading / (/GroupSet) is preferred.

Command

```
<command cmd="get" path="health"/>
```

Reply

```
<replies success="true" path="health" numberofreplies="3" commandDuration="123us">  
<health value="~RxLane:1-LaneDown~RxLane:2-LaneDown~RxLane:3-  
LaneDown~RxLane:4-LaneDown~"  
fullpath="GroupSet/Group/JugglerName1/Bus:1/health"/>  
<health value="~RxLane:1-LaneDown~RxLane:2-LaneDown~RxLane:3-  
LaneDown~RxLane:4-LaneDown~"  
fullpath="GroupSet/Group/JugglerName2/Bus:1/health"/>  
<health value="~RxLane:1-LaneDown~RxLane:2-LaneDown~RxLane:3-  
LaneDown~RxLane:4-LaneDown~"  
fullpath="GroupSet/Group/JugglerName3/Bus:1/health"/>  
</replies>
```

Message Formats

Message formats: core and extended

Messages may be sent in either plain text or XML format – see below for the plain text summary – and Compere will respond in the same format as the incoming message. All XML attributes must be quoted.

Messages are defined by a ‘command’ tag with various attributes depending on the message. Typical attributes include: cmd, path, type, value, makeactive and fullpath.

The core message set – as described in the HTML API – must have at a minimum a ‘cmd’ attribute.

```
<command cmd="status" />
```

The extended message set is accessible via the ‘action’ command. The availability of these commands depends on the current system configuration: e.g., adding a Juggler to the system will make additional commands available to the external control user.

Apps that use backslash as an escape character

Where the backslash \ is used as an escape character (e.g. C++, or a showcontroller app) paths to a file must themselves be escaped:
e.g., file="C:\\Images\\foldername".

Example (Windows o/s):

```
<command cmd="actions" type="Import MPCDI" file="C:\Users\[UserName]\Downloads\mpcdi-file-name.mpcdi"/>
```

and in plain text:

```
actions type="Import MPCDI" file="C:\Users\[UserName]\Downloads\mpcdi-file-name.mpcdi"
```

Response format

All valid messages are acknowledged immediately with 'success=true/false'. The XML response is a well-formed XML document which may be parsed by any established third-party library.

Note: the Compere version example below includes a date string and a build hash which can be used for build tracking/reporting.

Request

```
<command cmd="get" path="compereversion" />
```

Response

```
<replies success="true" path="compereversion" numberofreplies="1"
commandDuration="123us">
<compereversion value="1.0.0.1 (20200924 50473f3)"
fullpath="GroupSet/Group/JugglerName/atom/compereversion"/>
</replies>
```

Replies include the number of replies (relative paths especially are likely to elicit multiple responses) and how long the reply took (commandDuration), in microseconds.

System configuration

The complete system configuration can be read using the 'get' command with an additional 'path' (keyword) attribute.

Get current project and default project

```
<command cmd="status" />
```

Get info for all Jugglers in a project

```
<command cmd="get" path="juggler" />
```

Get info for a specific Juggler

```
<command cmd="get" path="[JugglerName]" />
```

Get all versions from all Jugglers in a system

```
<command cmd="get" path="fwversion" />
```

```
<command cmd="get" path="swversion" />
```

```
<command cmd="get" path="compereversion" />
```

Get versions for a specific Juggler

```
<command cmd="get" path="[JugglerName]/atom/fwversion" />
```

```
<command cmd="get" path="[JugglerName]/atom/swversion" />
```

```
<command cmd="get" path="[JugglerName]/atom/compereversion" />
```

Unsolicited notifications: registers

Typically, commands elicit an immediate response, but an external control user can add callbacks via registers to be notified asynchronously when a parameter changes: e.g., when the status of an interface is updated.

Get all CPU temperatures in the system

```
<command cmd="get" path="temperatureCPU" />
```

Receive a notification when any CPU temperature changes

```
<command cmd="register" path="temperatureCPU" />
```

Plain text messages

For convenience, when typing at a console, all XML messages have a shorter, plain text equivalent. This can be constructed by removing the outer command tag and dropping the quotes on the 'cmd' attribute.

Example: get all IP addresses in the system

XML

```
<command cmd="get" path="externalipaddress" />
```

Plain text

```
get path="externalipaddress"
```

Example: MPCDI import

XML

```
<command cmd="actions" type="Import MPCDI" file="C:\\path\\to\\system.mpcdi" />
```

Plain text

```
actions type="Import MPCDI" file="C:\\path\\to\\system.mpcdi"
```

Further examples

Get all temperatures for all Jugglers in a system

```
<command cmd="get" path="temperaturePL" />  
<command cmd="get" path="temperatureCPU" />
```

Get CPU temperature for a specific Juggler

```
<command cmd="get" path="/GroupSet/Group/[JugglerName]/SystemMonitor/temperatureCPU" />
```

Disable red beams system-wide

```
<command cmd="actions" type="Red Beam Off" />
```

Enable blue beams for a specific Juggler

```
<command cmd="actions" type="Blue Beam On" path="/GroupSet/Group/Juggler4540" />
```

Get active pixels and lines for a specific Juggler's video input

```
<command cmd="get" path="/GroupSet/Group/[JugglerName]/DpRx:2/VideoTiming/activepixels" />  
<command cmd="get" path="/GroupSet/Group/[JugglerName]/DpRx:2/VideoTiming/activelines" />
```

Heartbeats

A heartbeat could be implemented by periodically requesting 'status' from each Juggler.

System Status

Contains status elements:

- Timestamp of Compere
- Task manager
- Current project
- Default project
- Comms mode (client/server/off)
- Peer data

Command

```
<command cmd="status" />
```

Reply

```
<status success="true" time="Thu Jan 21 15:32:21 2021">
  <taskSummary total="0" failed="0" pending="0" complete="0"
    percentComplete="0.000000"/>
  <projectStatus ProjectPath="C:\Users\[UserName]
    \AppData\Local\compere\Projects\myproject.prj" DefaultProject="C:
    \Users\[UserName]\AppData\Local\compere\Projects\myproject.prj"/>
  <networkInfo commsMode="server"/>
  <peers>
    <peer uniqueId="79833e07d0e64ffe9a874c49e4516b5e" auxData="Host Name=BootCamp-0263E
      (Windows);MemUse=85% DiskFree=57.6GiB CPU idle=239 kernel=18 user=34;Curr Tree=My
      Project (8c5202e4b1174e4ea6055df9279ca500);Project Group: MainGroup;machineID=00-
      60-73-06-86-16;10.100.150.20;10.0.10.207;192.168.0.11;172.29.144.1;172.28.80.1;"
      userData=""/>
  </peers>
</status>
```

Juggler Status

Uptime

Command

```
<command cmd="get" path="[JugglerName]/juggleruptime"/>
```

Reply

```
<replies success="true" path="[JugglerName]/juggleruptime" numberOfreplies="1"
  commandDuration="123us">
```

```
<juggleruptime value="0d:01h:52m" fullpath="GroupSet/Group/
  [JugglerName]/juggleruptime"/>
</replies>
```

Bus Status

Get bus status

Command

```
<command cmd="get" path="/GroupSet/Group/[JugglerName]/Bus:1/health"/>
```

Reply

```
<replies success="true" path="/GroupSet/Group/[JugglerName]/Bus:1/health"
  numberOfreplies="1" commandDuration="123us">
  <health value="~RxLane:1-LaneDown~RxLane:1-CtrlBad~RxLane:2-
    LaneDown~RxLane:3-LaneDown~RxLane:3-CtrlBad~RxLane:4-
    LaneDown~RxLane:4-CtrlBad~RxLane:5-LaneDown~RxLane:6-
    LaneDown~RxLane:7-LaneDown~RxLane:8-LaneDown~"
    fullpath="GroupSet/Group/[JugglerName]/Bus:1/health"/>
</replies>
```

Bus Reset

Flash the bus

Command

```
<command cmd="set" path="/GroupSet/Group/[JugglerName]/Bus:1/reset" value="1"/>
```

Reply

```
<set success="true"/>
```

Genlock

Get genlock status

Command

```
<command cmd="get" path="/GroupSet/Group/
  [JugglerName]/VideoChannel:2/outputgenlocked"/>
```

Reply

```
<replies success="true" path="/GroupSet/Group/
  [JugglerName]/VideoChannel:2/outputgenlocked" numberofreplies="1"
  commandDuration="123us">
  <outputgenlocked value="0" fullpath="GroupSet/Group/
    [JugglerName]/VideoChannel:2/outputgenlocked"/>
</replies>
```

Get current genlock ratio and available options**Command**

```
<command cmd="get" path="/GroupSet/Group/[JugglerName]/HdmiTx:3/genlockratio"/>
```

Reply

```
<replies success="true" path="/GroupSet/Group/[JugglerName]/HdmiTx:3/genlockratio"
  numberofreplies="1" commandDuration="234us">
  <genlockratio typeName="MemberArray" value="1:1" enabled="1"
    fullpath="GroupSet/Group/[JugglerName]/HdmiTx:3/genlockratio">
    <memberarrayitem UUID="04e120c500000000040000000000178" value="1:1"
      enabled="1"/>
    <memberarrayitem UUID="04e120c500000000040000000000179" value="1:2"
      enabled="1"/>
    <memberarrayitem UUID="04e120c500000000040000000000180" value="2:1"
      enabled="1"/>
  </genlockratio>
</replies>
```

Get current genlock configuration and available options**Command**

```
<command cmd="get" path="/GroupSet/Group/[JugglerName]/HdmiTx:3/genlockconfig"/>
```

Reply

```
<replies success="true" path="/GroupSet/Group/[JugglerName]/HdmiTx:3/genlockconfig"
  numberofreplies="1" commandDuration="123us">
  <genlockconfig typeName="MemberArray" value="Free Run" enabled="1"
    fullpath="GroupSet/Group/[JugglerName]/HdmiTx:3/genlockconfig">
    <memberarrayitem UUID="04e120c500000000040000000000165" value="Free Run"
      enabled="1"/>
    <memberarrayitem UUID="04e120c500000000040000000000166" value="Lock to Rx:1"
      enabled="1"/>
    <memberarrayitem UUID="04e120c500000000040000000000167" value="Lock to Rx:2"
      enabled="1"/>
  </genlockconfig>
</replies>
```

```

<memberarrayitem UUID="04e120c5000000000400000000000168" value="Lock to Rx:3"
  enabled="1"/>
<memberarrayitem UUID="04e120c5000000000400000000000169" value="Lock to Rx:4"
  enabled="1"/>
<memberarrayitem UUID="04e120c5000000000400000000000170" value="Lock to Tx:1"
  enabled="1"/>
<memberarrayitem UUID="04e120c5000000000400000000000171" value="Lock to Tx:2"
  enabled="1"/>
<memberarrayitem UUID="04e120c5000000000400000000000172" value="Lock to Tx:3"
  enabled="1"/>
<memberarrayitem UUID="04e120c5000000000400000000000173" value="Lock to Tx:4"
  enabled="1"/>
<memberarrayitem UUID="04e120c5000000000400000000000174" value="Lock to
  GPIO:1" enabled="1"/>
<memberarrayitem UUID="04e120c5000000000400000000000175" value="Lock to
  GPIO:2" enabled="1"/>
<memberarrayitem UUID="04e120c5000000000400000000000176" value="Lock to
  GPIO:3" enabled="1"/>
<memberarrayitem UUID="04e120c5000000000400000000000177" value="Lock to
  GPIO:4" enabled="1"/>
</genlockconfig>
</replies>

```

Hotplug Detect

Simulate removal and reinsertion of HDMI output

Command

```

<command cmd="set" path="/GroupSet/Group/[JugglerName]/HdmiTx:3/triggerhpd"
  value="1"/>

```

Reply

```

<set success="true"/>

```

EDID Lists

Get current EDID selection and EDID list

Command

```

<command cmd="get" path="/GroupSet/Group/[JugglerName]/DpRx:2/edidlist"/>

```


Reply

```

<replies success="true" path="/GroupSet/Group/[JugglerName]/DpRx:2/edidlist"
  numberofreplies="1" commandDuration="567us">
  <edidlist typeName="MemberArray" value="HD60-Internal" enabled="1"
    fullpath="GroupSet/Group/[JugglerName]/DpRx:2/edidlist">
    <memberarrayitem UUID="04e120c5000000000400000000000081" value="HD60-
      Internal" enabled="1"/>
    <memberarrayitem UUID="04e120c5000000000400000000000082" value="2xHD120.bin"
      enabled="1"/>
    <memberarrayitem UUID="04e120c5000000000400000000000083" value="2xHD60.bin"
      enabled="1"/>
    <memberarrayitem UUID="04e120c5000000000400000000000084" value="4k30.bin"
      enabled="1"/>
    <memberarrayitem UUID="04e120c5000000000400000000000085" value="4K60.BIN"
      enabled="1"/>
    <memberarrayitem UUID="04e120c5000000000400000000000086" value="720p60.bin"
      enabled="1"/>
    <memberarrayitem UUID="04e120c5000000000400000000000087" value="HD120-
      10.bin" enabled="1"/>
    <memberarrayitem UUID="04e120c5000000000400000000000088" value="HD120.BIN"
      enabled="1"/>
    <memberarrayitem UUID="04e120c5000000000400000000000089" value="HD60.BIN"
      enabled="1"/>
    <memberarrayitem UUID="04e120c5000000000400000000000090" value="UHD60.BIN"
      enabled="1"/>
  </edidlist>
</replies>

```

Resolution

Get active lines for an input**Command**

```

<command cmd="get" path="/GroupSet/Group/
  [JugglerName]/DpRx:2/VideoTiming/activelines"/>

```

Reply

```

<replies success="true" path="/GroupSet/Group/
  [JugglerName]/DpRx:2/VideoTiming/activelines" numberofreplies="1"
  commandDuration="123us">
  <activelines value="2160" fullpath="GroupSet/Group/

```

```
[JugglerName]/DpRx:2/VideoTiming/activelines"/>
</replies>
```

Get all input info

Command

```
<command cmd="get" path="/GroupSet/Group/[JugglerName]/DpRx:2/VideoTiming"/>
```

Reply

```
<replies success="true" path="/GroupSet/Group/[JugglerName]/DpRx:2/VideoTiming"
  numberofreplies="1" commandDuration="789us">
  <videotiming UUID="04e120c5000000000000000000000010" fullpath="GroupSet/Group/
    [JugglerName]/DpRx:2/VideoTiming" value="">
    <name value="VideoTiming"/>
    <enabled value="1"/>
    <state value="0"/>
    <activepixels value="4096"/>
    <activelines value="2160"/>
    <totalpixels value="4256"/>
    <totallines value="2222"/>
    <hsyncwidth value="32"/>
    <vsyncwidth value="10"/>
    <hactivestart value="112"/>
    <vactivestart value="59"/>
    <vrate value="59.999523"/>
    <pixelclockmhz value="567.409999999999996816769"/>
    <bpp value="8"/>
    <colourspace value="RGB-4:4:4"/>
    <interlaced value="0"/>
    <stereo3d value="0"/>
    <direction value="3"/>
    <icon value=""/>
    <dynamicicon value="0"/>
    <bitsperpixel value="8"/>
  </videotiming>
</replies>
```

MPCDI

Get import progress

Command

```
<command cmd="status"/>
```

Reply

```
<status success="true">  
  <taskSummary total="7" failed="0" pending="1" complete="6"  
    percentComplete="85.714286" MPCDI-Processing="85.714286%"/>  
  <projectStatus ProjectPath="C:\Users\[UserName]  
    \AppData\Local\compere\Projects\primary.prj" DefaultProject="C:  
    \Users\[UserName]\AppData\Local\compere\Projects\myproject.prj"/>  
  <networkInfo commsMode="server" peerData="64-4b-f0-12-75-fd "/>  
</status>
```

Project Controls

Note: Saving and Loading is to the server and the path must use the correct slashes for the server OS.

Open a project

Command

```
<command cmd="open" file="C:\Users\[UserName]  
  \AppData\Local\compere\Projects\primary.prj"/> (Windows)  
<command cmd="open" file="/share/compere/bin/Projects/primary.prj"/> (Linux)
```

Reply

```
<open success="true" commandDuration="123us"/>
```

Save a project

Command

```
<command cmd="save" file="C:\Users\[UserName]  
  \AppData\Local\compere\Projects\secondary.prj"/> (Windows)  
<command cmd="save" file="/share/compere/bin/Projects/secondary.prj"/> (Linux)
```

Reply

```
<save success="true" commandDuration="123us"/>
```

Create a new project

Command

```
<command cmd="new" projectname="My project"/>
```

Reply

```
<new success="true" commandDuration="123us"/>
```

Presets and Clones

Use a Preset or Clone resource

(Recalling a Preset imports values to matching UUIDs. Loading a Clone imports saved copies of objects with new UUIDs.)

Command (Presets)

```
<command cmd="recallpreset" file="C:\[Compere Root Path]\Presets\[file].pre"/>
```

or (implied file location)

```
<command cmd="recallpreset" file="[file].pre"/>
```

Reply

```
<recallpreset success="true" commandDuration="504us"/>
```

Command (Clones)

```
<command cmd="loadclone" file="C:\[Compere Root Path]\Clones\[file].clo"/>
```

or (implied file location)

```
<command cmd="loadclone" file="[file].clo"/>
```

Reply

```
<loadclone success="true" commandDuration="652us"/>
```

Shutdown or Reboot Jugglers

Shut an individual Juggler down

There is currently no wake on LAN facility in Juggler.

Command:

```
<command cmd="set" path="/GroupSet/Group/[JugglerName]/shutdown" value="1"/>
```

Reply

```
<set success="true"/>
```

Reboot an individual Juggler

Command:

```
<command cmd="set" path="/GroupSet/Group/[JugglerName]/reboot" value="1"/>
```

Reply

```
<set success="true"/>
```

Logs and Diagnostics

Compere generates a log file (CompereLog.txt) which can be very useful for system debugging and troubleshooting. On a Windows machine this can be found by default in: C:\Users\[USERNAME]\AppData\Local\Compere\logs.

➤ Default file locations can be edited. See File Locations in [Preferences Panel](#) ³⁷.

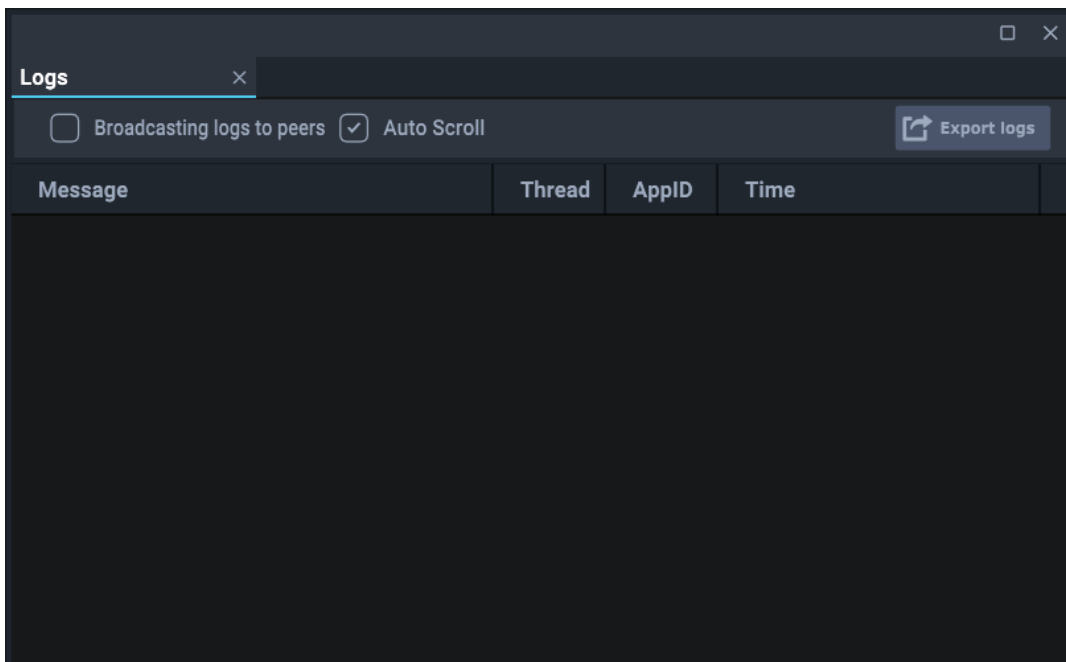
These files inflate, so when they get to a certain size they are archived with a sequential file suffix: CompereLog_nn.txt.

On Linux, this will be in a directory one up from the executable directory and then in a folder called logs.

Juggler's internal Linux control CPU also runs a reduced version of Compere, running from /share/compere/bin. Its logs can be found in /share/compere/logs.

On Mac these should be found in ~/Library/Compere/logs.

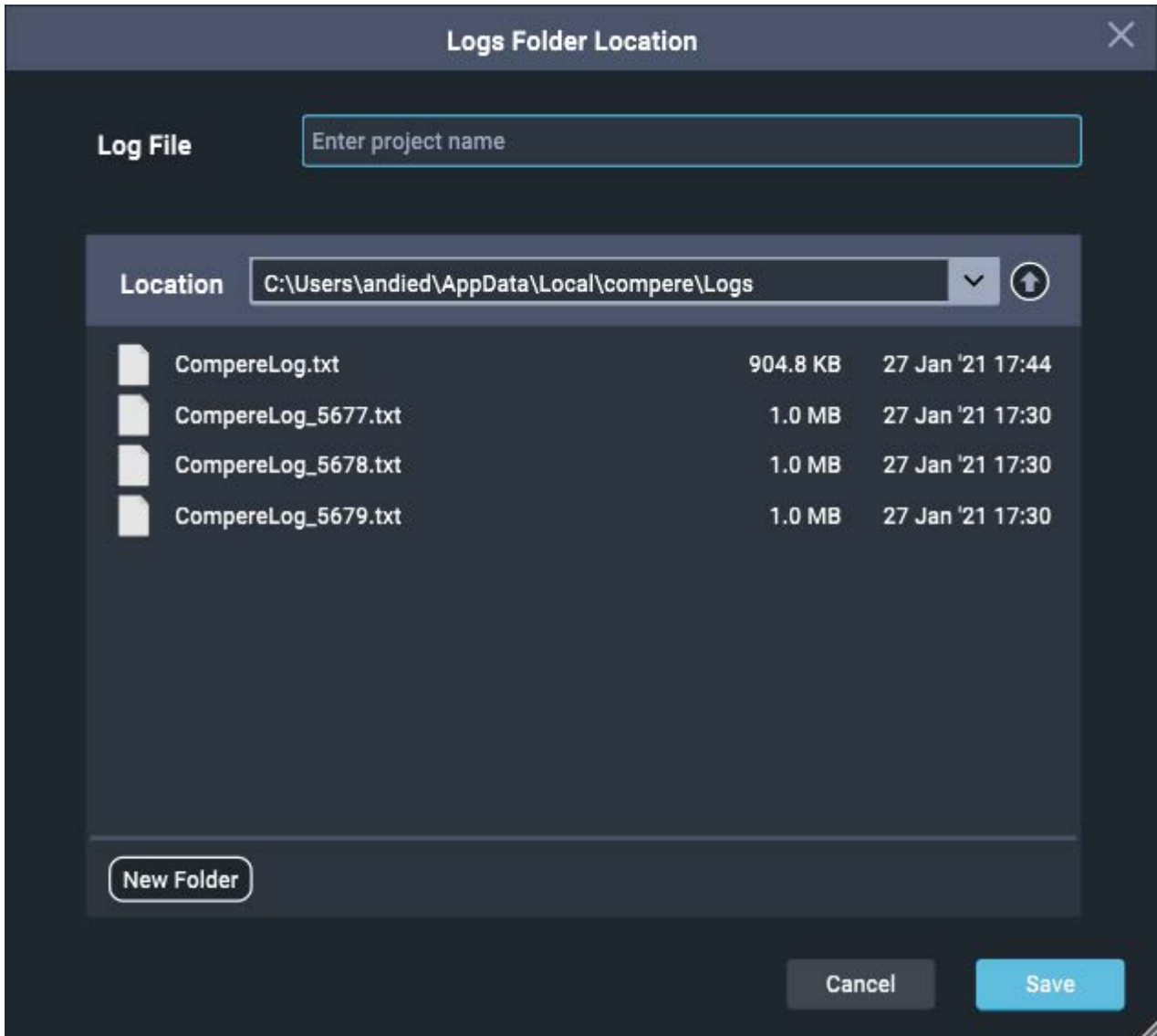
Logs can be viewed dynamically, from the Panels > Logs menu:



- Choose whether to broadcast the log to Compere peers in the Group.
- Check Autoscroll to keep latest logs in view.
- Export logs to a LogData.txt file (location: C:\Users\[UserName]\AppData\Local\compere\logs).

Gather Project Data

Lists logs per Juggler in the Project Group. Click 'Open' to create a zip file in this location of all logs and the local Compere Preference.pref file.



Reporting diagnostic information

Send the following information to us to help us understand what you are seeing:

- Save server Project File locally (XML)
- Screen shot of Nodes View
- Current log file (no number suffix)

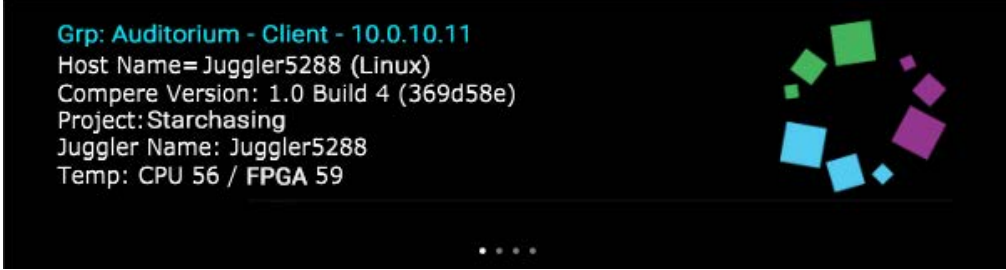
For technical support, please contact support@7thsense.one.

Command Line Arguments

Occasionally it can be useful to run Compere with a command line switch, to force it to run in a particular way.

For example, different Windows shortcuts can be created to do this, by adding the switches into the shortcut target:

C:\7thSense\compere\compere.exe **networkserver**

Switch	Function
networkserver	opens Compere as a network server on startup
networkclient	opens Compere as a network client on startup
mingui	opens Compere in a minimal form, as it does on a Juggler front panel: 
mingui systemtray	runs Compere in the Windows System Tray, with the option to Quit (see Compere MinGUI) ⁴⁷

Troubleshooting a Juggler System

Bus faults

Faults in the bus will be indicated by the Rx bus pins in a Juggler in the Nodes panel being red when they should be green (not set to Off, but cabled). The majority of faults in the bus system will be caused by connector contamination. See [Connect a Juggler Optical Data Bus](#) for cable care and cleaning.

Media connection faults

When disconnected or valid media is not present, Juggler input and output pins in the Nodes panel will show red. This is an important indicator of a poor connector or bad cable.

NDI continuity

NDI preview in Jugglers in the Nodes panel should be showing the continuity of media flow. When a feed is unavailable but NDI preview is enabled, these previews will show a grey NDI icon only:



Network faults

In the Nodes panel, disconnection of a Juggler is indicated by the node going red. Loss of communications by a control PC will be indicated by the Compere UI going red.

Check in *Configuration > Network Configuration* that the Project Group is correct and that the affected instances of Compere (control PC or Jugglers) are in a Group with a server. Reconnecting a Juggler as server will replace a lost server.

Document Information

Date	Document edition	Software version	Revision Details	Author/Editor
December 2020	1	Compere 1.0 build 1 beta	New release	Andie Davidson
February 2021	2	Compere 1.0 build-2 beta	Compere version updates	Andie Davidson
April 2021	3	Compere 1.0-2 beta	Updates to clones and presets; external control	Andie Davidson
September 2021	4	Compere 1.0 build 4 beta	Revised UI, output canvas busing, extensions to external control	Andie Davidson
December 2021	5	Compere 1.0 build 5 beta	Logic / processing nodes added, Front panel display changes, Juggler node revisions, general UI improvements. Republished as Compere guide, companion to M611 Operating Juggler 1 Pixel Processor	Andie Davidson

The Adopted Trademarks HDMI, HDMI High-Definition Multimedia Interface, and the HDMI Logo are trademarks or registered trademarks of HDMI Licensing Administrator, Inc. in the United States and other countries.

A

add, select, edit, delete, remove objects 20

B

black level 77
 bus capability 102
 bus connections 102

C

canvas 13
 canvas arrangement 70
 canvas assembly panel 60
 canvas object 60
 Canvas Outputs 63
 canvas panel shortcuts 60
 canvas preset 70
 Canvas Windows 67
 canvas wizard 116
 clones 29, 129
 colour meanings 86
 colours in nodes 79
 command line switches 152
 Compere network ports (list) 131
 Compere on Jugglers
 upgrading Compere 11
 connecting Juggler inputs, outputs 119
 connection timeout 132
 context menus 45
 control PC 13

D

debug menu 33
 diagnostics 150
 display configuration 60
 docking panels 27
 document information 154

E

EDID spoofing 121
 External Control 13
 bus reset 142
 bus status 142
 EDID lists 144
 genlock 142
 hotplug detect 144
 juggler status 141

message formats 137
 MPCDI 146
 open project 147
 Presets and Clones 148
 project controls 147
 reboot a Juggler 149
 resolution 145
 shutdown a Juggler 149
 system status 141
 wildcards 136

External Control (ASCII / XML)
 command groups 132
 protocol 132
 static and dynamic 132

F

failover systems 96
 file distribution 33

G

gather project data 150

I

install a control PC 15
 IntelligentSource, properties 96
 interaction mode (canvas panel) 60

J

Juggler
 reset 11
 upgrading Juggler firmware 11
 Juggler / Compere systems 6
 bus bridging 106
 Juggler connections in Compere 9
 Juggler connector properties 99
 Juggler network ports (list) 131
 Juggler node 79, 86
 Juggler properties
 Atom (internal porocessor) 84
 defaults 85
 GPIO 85
 System Monitor 86
 video channels 92
 Juggler Properties: Bus 105
 Juggler unit properties 81
 Juggler Video interface timing 102
 Juggler view 20

K

keyboard shortcuts 18

L

layers, windows 67

layouts 27

log files 150

logic/processing nodes 123

M

main menu 29

minGUI 47

MPCDI 77

N

NDI monitoring 13, 119

NDI source name per channel 99

network configuration 49

network discovery 49

nodes 13

Nodes assembly panel 86

O

output canvas busing 110

overview 13

P

panels 27, 29

panels, arrangement 20

PC as server only 47

peer data panel 33

positioning elements 27

presets 29, 70, 129

project groups 7, 13, 49, 56

project panel 20, 43

projects 7

projects, project files 42

properties panel 20

R

recall preset 70

resources 20

S

save layout 20

server-client roles 49

settings 37

3D panel 40

Canvas panel 40

file locations 37

Juggler 40

mini map 41

Nodes panel 41

preferences panel 37

shortcuts 27

status bar 35

system config table 29

T

task progress 36

task tracker 36

technical support 150

terms and definitions 16

toolbar, canvas panel 60

toolbar: Nodes panel 86

troubleshooting

cable/connector faults 153

NDI continuity monitoring 153

U

update Compere on Jugglers 15

V

version 15, 29

Video Channel, Video Measure 97

video channels 86

W

warp and blend 77

wizard 116

workflow, output canvas bus 110

E: info@7thsense.one
W: 7thsense.one

7thSense Design Ltd
2 The Courtyard, Shoreham Road
Upper Beeding
Steyning
West Sussex
BN44 3TN
UK

T: +44 (0) 1903 812299

7thSense LLC
4207 Vineland Rd
Suite M1
Orlando, FL 32811
USA

T: +1 407 505 5200

