axle ai

connectr™

automate your media workflow

User Manual

Contents

1. Overview 4
1.1 What is connectr? 4
1.2 What is connectr designer? 4

2. Setup & Examples 5
2.1 Installation 5
2.2 Starting connectr designer 5
2.3 Tutorial Examples 6

3. User Interface 6
3.1 Workflow Tabs 7
3.2 Workflow View 8
3.2.1 Zooming & Panning 8
3.2.2 Unit Creation & Deletion 8
3.2.3 Unit Arrangement 8
3.2.4 Unit Distribution and Alignment 9
3.2.5 Unit Editing 9
3.2.6 Connecting Units 9
3.2.7 Creating Workflows from Unit Selections 9
3.2.8 Dataflow Visualization 9
3.3 Project Panel 9
3.3.1 Tab Overview 9
3.4 Project Tab 10
3.4.1 Workflows List 10
3.5 Variables List 11
3.6 Inspector Tab 11
3.7 Library Tab 11
3.8 Message Log & Memory Monitor 12
3.9 Preferences 14

4. Fundamentals 15
4.1 Projects, Workflows & Units 16
4.1.1 What is an connectr Project? 16
4.1.2 What is an connectr Workflow? 16
4.1.3 What is an connectr Unit? 17
4.1.4 Workflow/Unit Interaction 17
4.2 Connectr Execution Model 18
4.3 Unit Anatomy & Ports 18
4.3.1 Unit Anatomy 18
4.3.2 Unit Ports 19
4.3.3 Unit Types 19
4.4 Unit Attributes 20
4.4.1 Modifying Units via the Inspector 20
4.4.2 Modifying Units Directly 21
4.5 Unit Creation & Deletion 22
4.5.1 Adding Units via the Library 22
4.5.2 Adding Units via the Context Menu 22
4.5.3 Deleting a Unit 23
4.6 Cables 23
4.6.1 Connecting a Cable 23
4.6.2 Removing a Cable 24
4.6.3 Rewiring a Cable 24
4.7 Unit Triggers & Events 24

4.7.1 Unit Triggers *25*
4.7.2 Event Units *25*
4.8 Data Types & Color Coding 26
4.8.1 Other Port Types *27*
4.9 Variables 27
4.9.1 Project Variables *28*
4.9.2 Local Variables *28*
4.9.3 Variable Types *29*
4.10 Published Inputs and Outputs 29
4.10.1 Adding a Published Input/Output *29*
4.10.2 Editing a Published Input Value *30*
4.10.3 Renaming a Published Input/Output *30*
4.10.4 Removing/Rewiring a Published Input/Output Port Connection *30*
4.10.5 Removing a Published Input/Output *30*
4.10.6 Reordering Published Inputs/Outputs *30*
4.11 The Workflow Unit (Subworkflows) 30
4.11.1 Using the Workflow Unit *31*
4.11.2 Editing a Workflow Unit *31*
4.12 Dynamic Ports 32
4.13 Pending Connections 33
4.14 Annotations 34
4.14.1 How to Use *34*
4.15 Exposed Functions 34
4.16 JavaScript Unit 35
4.16.1 Declaring Input and Output Ports *35*
4.16.2 Input and Output Ports Data Types *35*
4.16.3 Default Functions *35*
4.16.4 Using Project Variables *35*

5. Tutorial Project 35
5.1 Aims of this Tutorial 35
5.2 Adding Units 36
5.3 Connecting Units 36
5.4 Modifying Unit Attributes 37
5.5 Running the Project 37

6. Running an connectr Project 38
6.1 How to Run a Project 38
6.2 How to Stop a Project 38

7. Further Reading 38
7.1 Best Practices 38
7.1.1 General Workflow Building *39*
7.1.2 Main Bus Design *39*
7.2 Debugging / Workflow Analysis 39
7.2.1 Workflow Stepping *40*
7.3 Workflow Profiler 40
7.3.1 Introduction *40*
7.3.2 Overview *40*
7.4 Connectr Control Protocol 41
7.5 Menu Commands 41

8. Limitations & Known Issues 42
8.1 Version 1.0 42
8.2 Version 1.0.1 42

1. Overview

1.1 What is connectr?

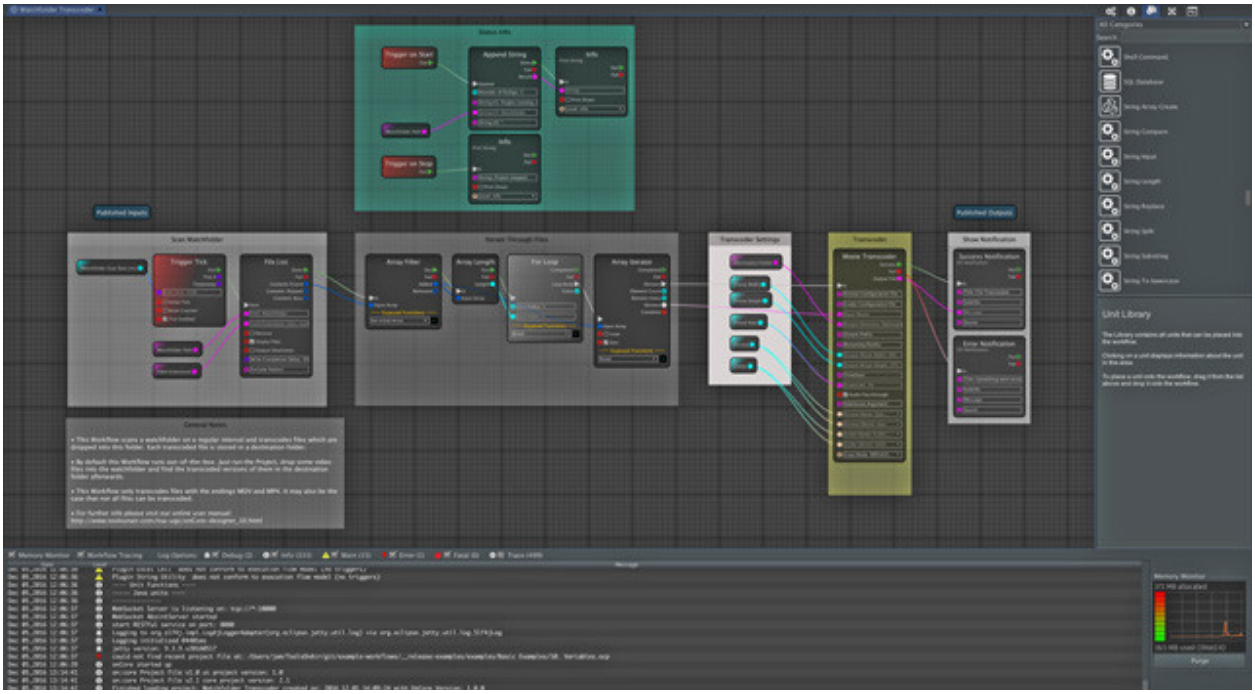
axle ai's **connectr** is a powerful and groundbreaking new **software component architecture** at the heart of the system, designed to deliver the ultimate in multipurpose broadcast and workflow automation. With connectr, axle ai has created an all-new software infrastructure; combined with **mediaCore**, axle ai's Linux-based media engine, broadcast professionals on any Linux- or Mac-based platform can now experience axle ai's intuitive workflows and seamless integration of **ingest**, **playout**, **storage**, and **realtime graphics**.

With connectr you can:

- Build Complex Broadcast Workflows using **connectr designer**
- Integrate and Customize axle ai and Third Party Products

1.2 What is connectr Designer?

connectr Designer gives users a **visual toolkit** to design and create their own custom **Workflows**, **Microservices** and **Applications**. axle ai has designed connectr to enable smooth integration with your existing hardware and software.



Connectr Designer is a visual programming environment where it is not mandatory to work with code, instead you work with graphical elements – so called Units. Units are essentially building blocks and can be interconnected with other Units to exchange data. The node-based paradigm is used in many modern applications as it's an easy-to-learn, yet powerful and versatile approach to accomplish both simple and complex tasks.

2. Setup & Examples

2.1 Installation

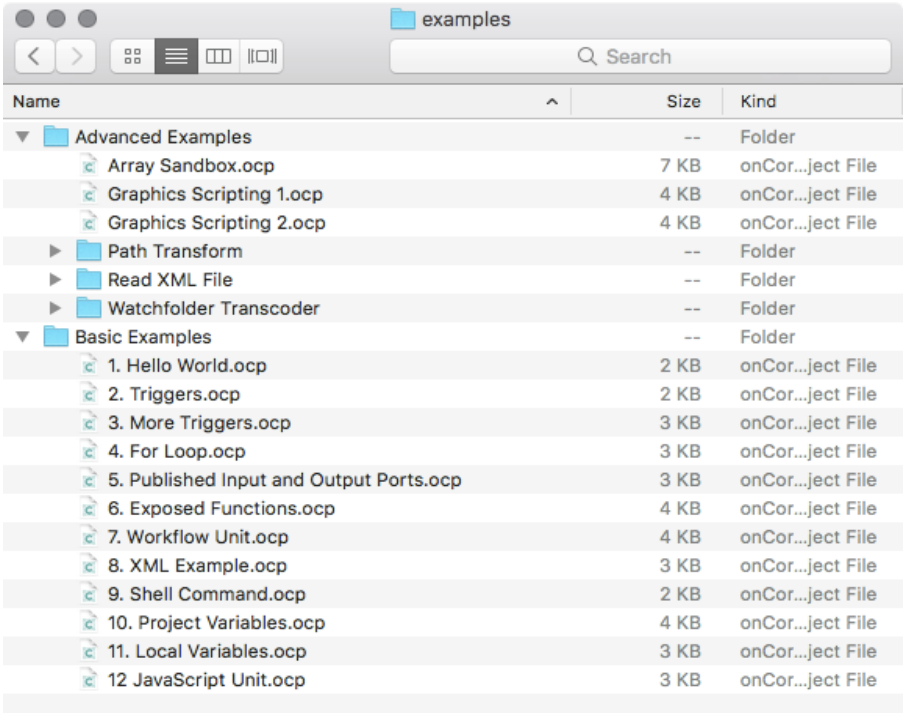
1. Open the **disk image** and drop the **connectr designer** folder onto the **Applications** folder.
2. **Eject** the disk image.

2.2 Starting connectr Designer

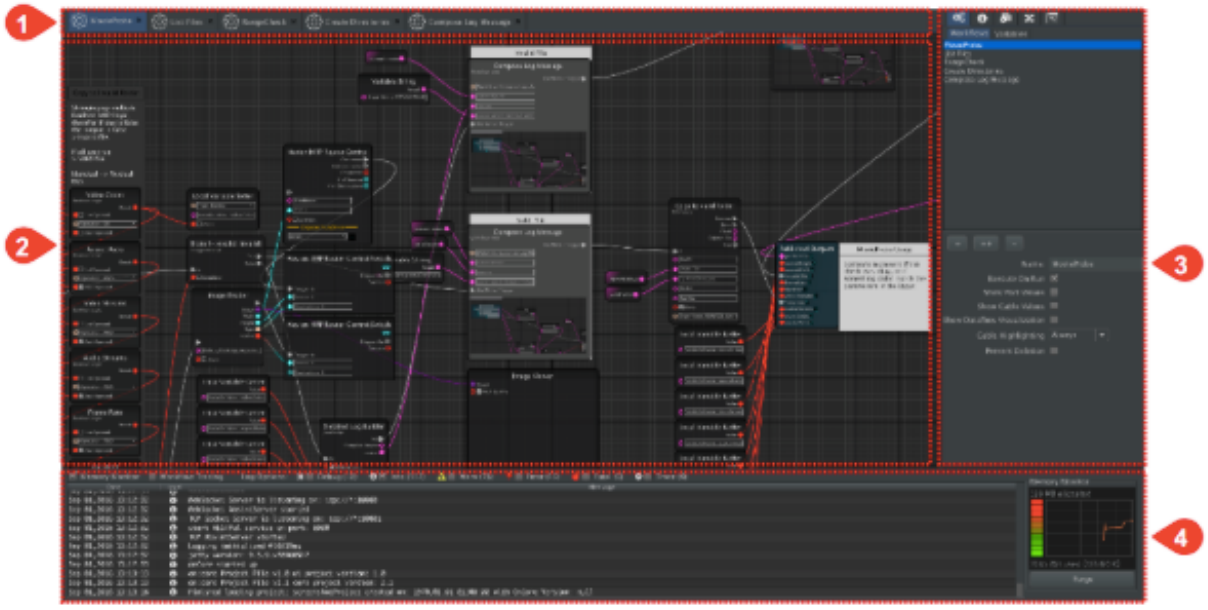
1. Open your **Applications** folder, then open the **connectr Designer** folder.
2. Double click the **connectr Designer Application** to start it.

2.3 Tutorial Examples

We have added several example Projects which can help you getting used to connectr Designer. They are located in the **examples** folder.

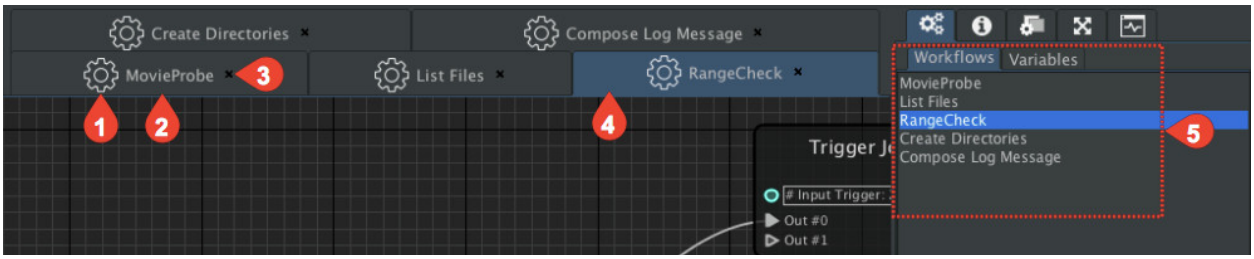


3. User Interface



→ Workflow Tabs Shows all open Workflows as tabs and informs about the state of each Workflow.
→ Workflow View The Workbench: Create the Workflow, connect Units with Cables, create Annotations and more.
→ Project Panel Contains the Workflows & Variables List, Unit Settings, the Unit Library, Distribute & Align tools and the Workflow Profiler.
→ Message Log & Memory Monitor Displays all kind of messages and informs about the memory usage.

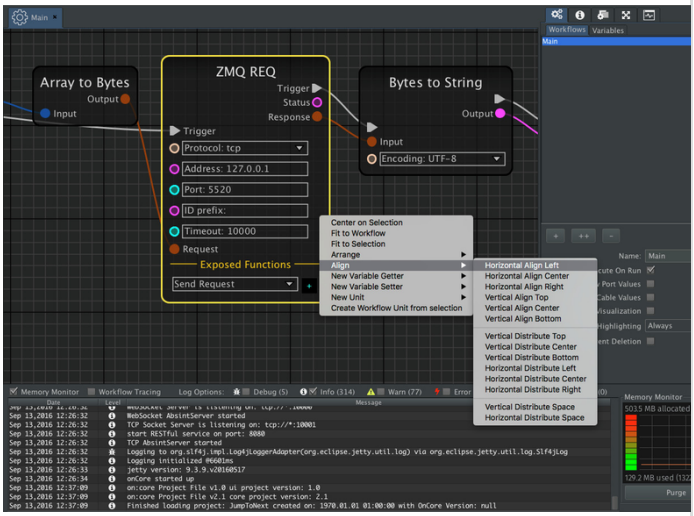
3.1 Workflow Tabs



1	**Workflow State:** The gear-icon informs about the state of a Workflow. **Empty Gear:** Project not running, Workflow not executing. **Filled Gear, not rotating:** Project running, Workflow not executing. **Filled Gear, rotating:** Project running, Workflow executing. Whether a Workflow will be executed when the Project is running is defined in the **→ Project Panel**.
2	**Workflow Name:** Name of the Workflow. It can be changed in the Project Panel.
3	**Close Button:** Closes the respective Workflow tab. A Workflow can be reopened by double clicking the Workflow in the **Workflows List** (→ Project Panel).
4	**Workflow Tab:** Each open Workflow is represented by a tab. The currently active Workflow tab is highlighted in blue.

5	**Workflows List:** This list contains all Workflows of the Project. Double clicking a Workflow shows/opens the Workflow in the Workflow View.

3.2 Workflow View



The **Workflow View** is where Units are placed, arranged, connected and modified. It offers the following functionality:

- Zooming & Panning
- Unit Creation & Deletion
- Unit Arrangement
- Unit Distribution and Alignment
- Unit Editing
- Creating Cables
- Creating Workflows from Unit Selections
- Dataflow Visualization

3.2.1 Zooming & Panning

The Workflow View can be zoomed and panned.

Zooming	Hold the **option key** while using the **scroll wheel** of the mouse. The mouse pointer hast to be in the Workflow View while doing so.
Panning	Hold the right mouse button to pan the Workflow View. The mouse pointer has to be in the Workflow View while doing so.

Apple Magic Mouse / Magic Trackpad users can use **two fingers** to pan around the Workflow View. Zooming requires to hold the **option key** and using two fingers. Pinching is currently not supported.

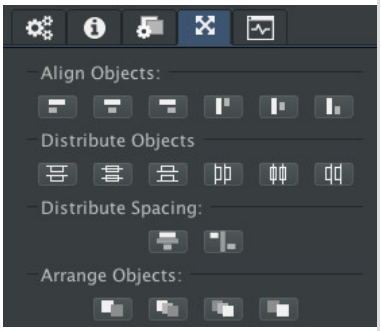
3.2.2 Unit Creation & Deletion

Refer to the chapter → Unit Creation & Deletion.

3.2.3 Unit Arrangement

Units can be moved to any position of the Workflow View via simple drag & drop. You can select multiple units by dragging a Lasso over a group of units or you can use the **shift key** to click-select multiple Units.

3.2.4 Unit Distribution and Alignment



The **Project Panel** contains the **Align & Distribute** tools which can be used on a multi-selection of Units.

3.2.5 Unit Editing

Input-Ports of Units can be edited directly in the Workflow View by clicking/double-clicking a Port. Refer to the chapter → Unit Attributes for further information.

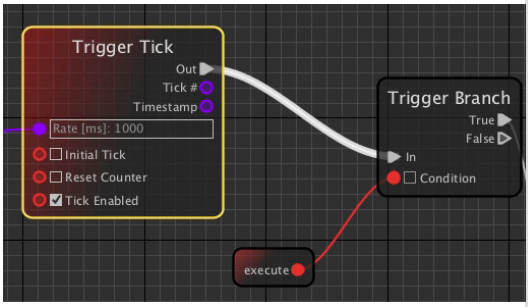
3.2.6 Connecting Units

The Workflow View is used to draw Cables between Units. Refer to the chapter → Cables for further information.

3.2.7 Creating Workflows from Unit Selections

You can create Workflows from any Unit selection. Refer to the chapter → The Workflow Unit for further information.

3.2.8 Dataflow Visualization



The flow of data can be visualized by activating the corresponding checkbox in the → Project Panel. With Dataflow visualization turned on, cables are highlighted whenever data or a trigger is transmitted.

3.3 Project Panel

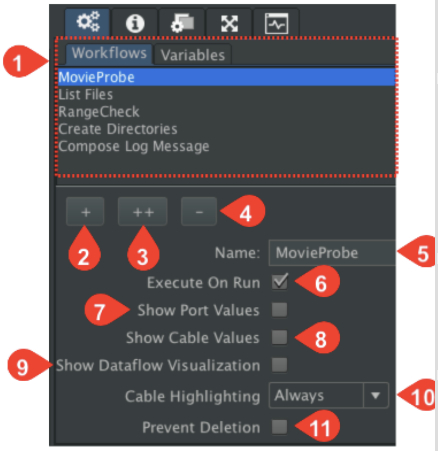
3.3.1 Tab Overview

	1	**Project:** Contains the **Workflows List**, the **Variables List** and **Workflow specific settings**.

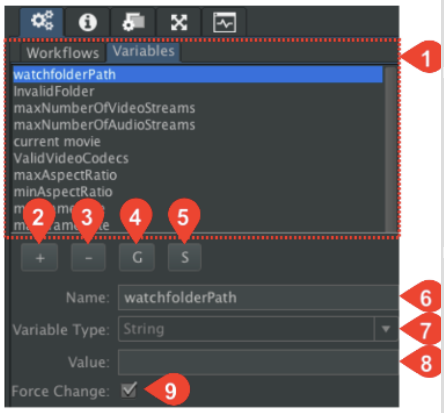
	2	**Inspector:** Shows attributes of the selected Unit.
	3	**Library:** Lists Units and shows Unit documentation.
	4	**Layout:** Align and distribute tools.
	5	**Workflow Profiling:** Analyzes the Workflow in regard to execution time or memory consumption.

3.4 Project Tab

3.4.1 Workflows List

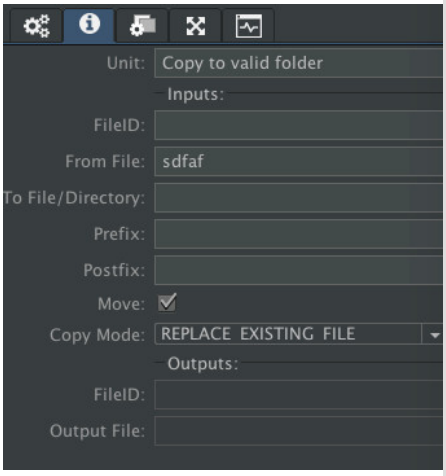
	1	**Workflows List:** Lists all Workflows in the current Project.
	2	**Add Workflow:** Adds a new Workflow to the current Project.
	3	**Duplicate Workflow:** Duplicates the selected Workflow.
	4	**Remove Workflow:** Removes the selected Workflow.
	5	**Workflow Name:** Name of the selected Workflow.
	6	**Execute On Run:** When enabled, the selected Workflow will be executed when the Project is running.
	7	**Show Port Values:** Shows values (which are transmitted via Cables) next to the ports of Units.
	8	**Show Cable Values:** Shows values which are transmitted via Cables.
	9	**Show Dataflow Visualization:** Highlights Cables when triggers and data are transmitted.
	10	**Cable Highlighting:** Select how Cables will be displayed: • **Always:** Shows Cables in full opaqueness all the time. • **Mouse Over:** Shows Cables while the cursor hovers over a Unit where the Cable is connected to. • **Selected:** Shows Cables if one of the Units the Cable is connected to is selected.
	11	**Prevent Deletion:** Set whether a Workflow can be deleted or not.

3.5 Variables List



1	**Variables List:** Lists all Variables of the current Project.
2	**Add Variable:** Adds a new Variable to the current Project.
3	**Remove Variable:** Removes the currently selected Variable from the Project.
4	**Create Getter Unit:** Places a Getter Unit in the Workflow View.
5	**Create Setter Unit:** Places a Setter Unit in the Workflow View.
6	**Name:** Name of the Variable.
7	**Variable Type:** Type of the Variable.
8	**Value:** Value of the Variable.
9	**Force Change:** When enabled the change filter of the selected Variable will be disabled.

3.6 Inspector Tab



The **Inspector** is used to change the name of a Unit and to modify Port Values of Units.

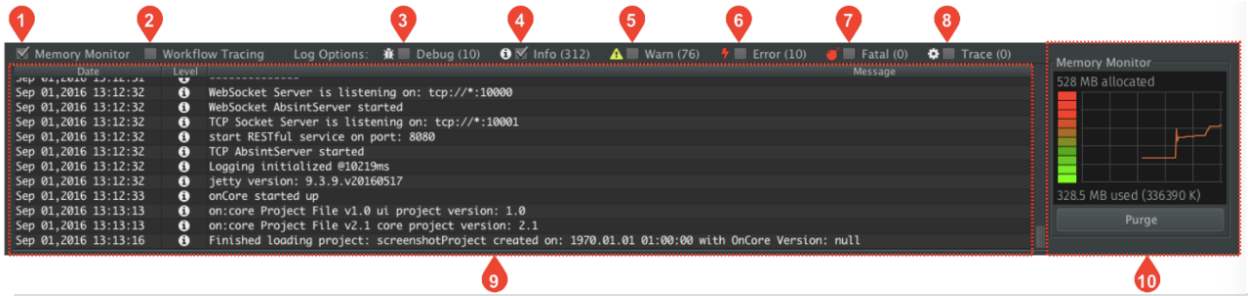
The **Unit Name** is the only one attribute which is available for all Units (excluding **Getters** and **Setters**).

Also see chapter → Unit Attributes.

3.7 Library Tab

1	**Unit Category:** Select from different Unit categories.
2	**Search:** Unit lookup. Entering a search term immediately shows all matching Units. Pressing the **escape key** after entering a search term cancels the search and shows all Units again.
3	**Unit List:** Lists all Units from the selected category/search result.
4	**Unit Documentation:** Shows the Unit documentation.

3.8 Message Log & Memory Monitor

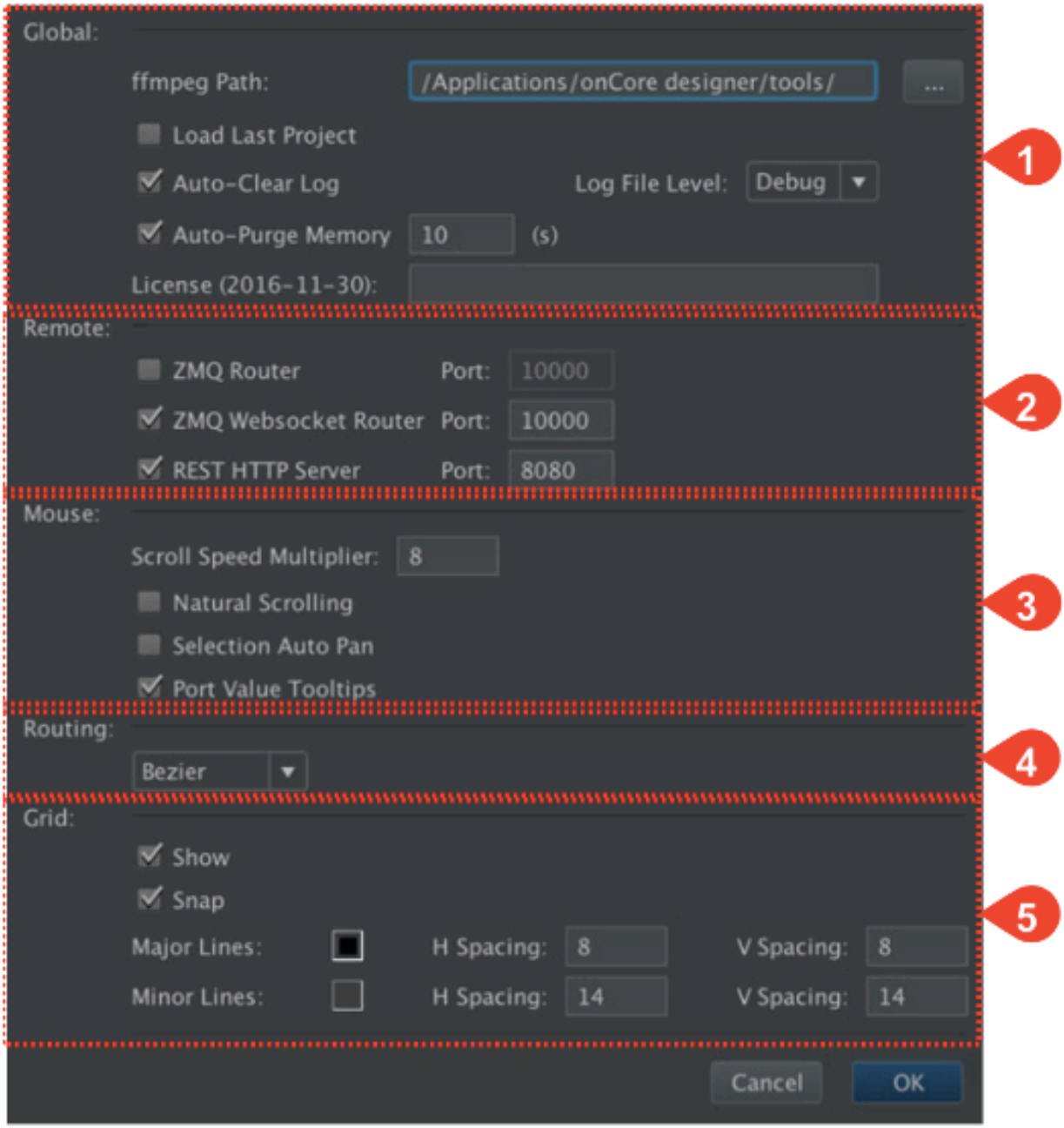


1	**Memory Monitor:** Shows or hides the Memory Monitor.
2	**Workflow Tracing:** When enabled the Workflow(s) are traced which means that each step/action/event/change is displayed in the Message Log. The **Trace Log Option** must to be activated in order to see trace entries.
3	**Log Option – Debug:** When enabled the log will show **debug** messages.
4	**Log Option – Info:** When enabled the log will show **info** messages.
5	**Log Option – Warn:** When enabled the log will show **warning** messages.
6	**Log Option – Error:** When enabled the log will show **error** messages.
7	**Log Option – Fatal:** When enabled the log will show **fatal** messages.
8	**Log Option – Trace:** When enabled the log will show **trace** messages.
9	**Log Entries:** Shows messages accordingly to the log options. There are three columns (**Date**, **Level** and **Message**). Double click an entry to jump to the corresponding Unit.

10	**Memory Monitor:** Displays how much memory is allocated and used for the current Project. The **Purge** button can be used to manually free up memory.

To reveal the log files, right-click the Message Log and select **Reveal log folder in Finder**.

3.9 Preferences



1	**Global** • **ffmpeg Path:** Path to ffmpeg executables. connectr designer is shipped with ffmpeg version 2.7.2. Alter this option if you want to use another version of ffmpeg. • **Load Last Project:** If checked, loads the last Project upon launch of connectr designer. • **Auto Clear Log:** Clears the Message Log each time a Project is set into running state. • **Auto-Purge Memory**: Purges the memory based on the specified interval. • **License:** Shows the license key and when your license expires.
2	**Remote (**see chapter → connectr Control Protocol) • **Enable ZMQ Router:** Starts a ZMQ Router to remote control connectr Designer • **Enable ZMQ Websocket Server:** Starts a ZMQ Websocket Router • **Enable REST HTTP Server:** Enables the REST interface.
3	**Mouse** • **Scroll Speed Multiplier:** Speed settings of the scrolling. • **Natural Scrolling:** Reverses scroll axis. • **Selection Auto Pan:** Pans to the selected Unit automatically. • **Port Value Tooltips:** Enable or disable Tooltips for Unit-Ports
4	**Routing** defines the appearance of Cables • **Bezier:** Curved cable style. • **Orthogonal:** Edgy cable style.
5	**Grid** settings for the Workflow View Background • **Show:** Show or hide the background grid. • **Snap:** Enable or disable snapping of Units to the background grid. • **Major/Minor Lines:** Defines the color and spacing of the background grid.

4.Fundamentals

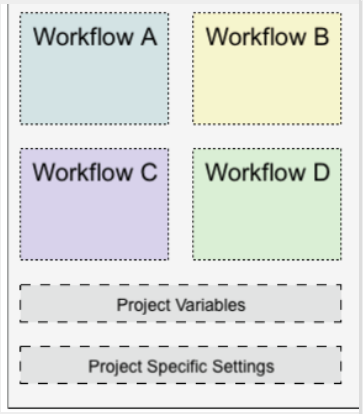
This chapter deals with the **fundamentals** of connectr designer and explains **paradigms**, **functionalities** and **strategies** to understand and master connectr designer. However, if you are already an experienced node-based visual programming environment user, you can skip these chapters and use them as reference.

- Projects, Workflows & Units
- Connectr Execution Model
- Unit Anatomy & Ports
- Unit Attributes
- Unit Creation & Deletion
- Cables
- Unit Triggers & Events
- Data Types & Color Coding

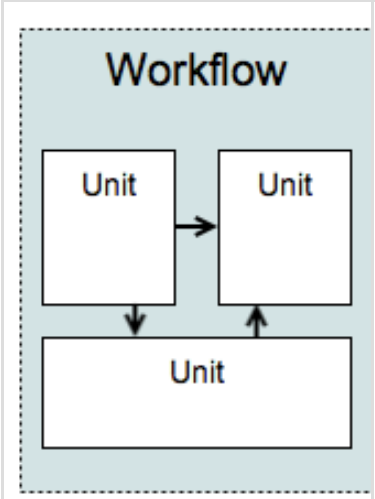
- Variables
- Published Inputs and Outputs
- The Workflow Unit (Subworkflows)
- Dynamic Ports
- Pending Connections
- Annotations
- Exposed Functions
- JavaScript Unit

4.1 Projects, Workflows & Units

4.1.1 What is an connectr Project?

	An **connectr Project** is the environment which contains: • **Workflows** • **Project Variables** • **Project Specific Settings.** connectr stores Projects as **OCP** files.
	**Specifications of an connectr Project** • connectr designer is Project based. • Projects contain at least one **Workflow**. • Projects may contain several **Workflows**, **Project Variables** and **Project Specific Settings**. • Variables are accessible by all Workflows of a Project, but there are also Workflow specific → Variables.

4.1.2 What is an connectr Workflow?



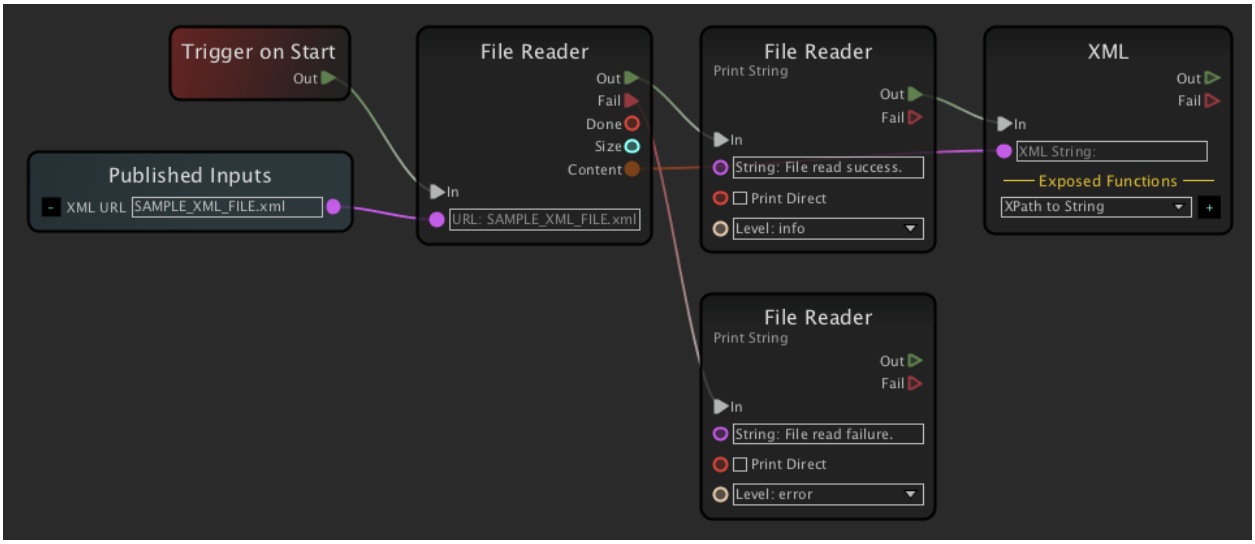
Specifications of an connectr Workflow

- A **Workflow** may perform a simple or complex task – it is up to you, how much complexity you add to a Workflow and which strategy you use to solve a specific issue or requirement.
- Inside of a Workflow **Units** are the 'building blocks'.
- Similar to a Workflow, a Unit performs an even more specific task.
- Each Workflow has a name. Make sure to name your Workflows properly, so you know what it is doing.

An **connectr Workflow** performs a specific task and is part of a Project.

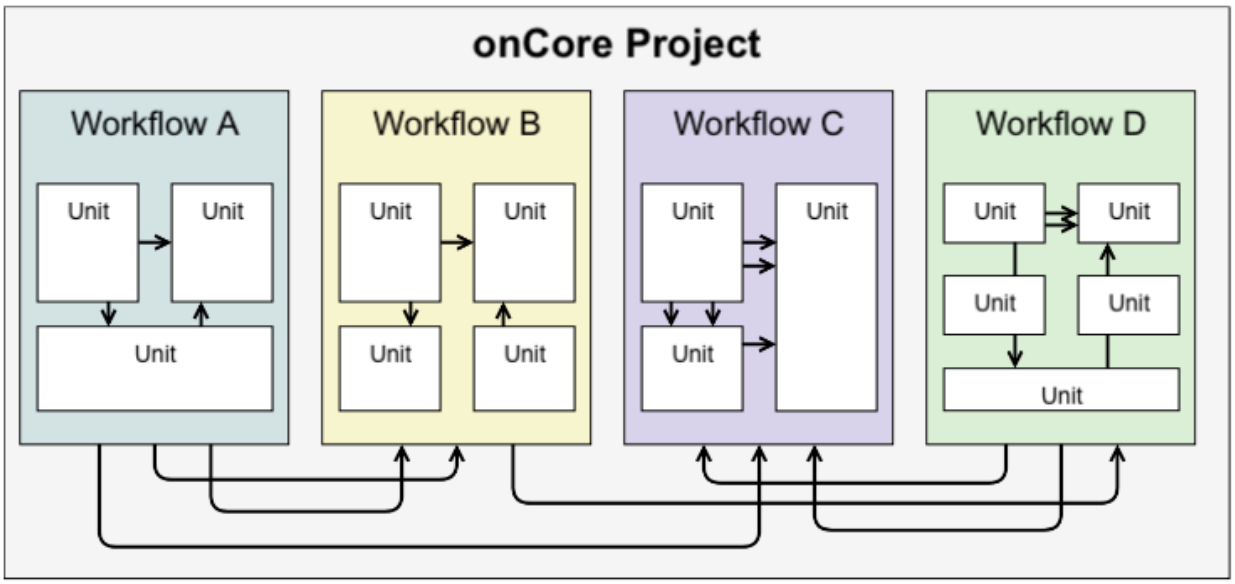
4.1.3 What is an connectr Unit?

An **connectr Unit** can be understood as a **graphical representation** of a **function** (as in traditional programming). It has one or multiple attributes and is usually interconnected with other Units. Think of it as a function which transforms a set of inputs into a set of outputs. Units are described in detail in the chapter → Unit Anatomy & Ports.



4.1.4 Workflow/Unit Interaction

This is a simplified schematic of an connectr Project. Workflows can also be interconnected end exchange data. This is described in detail in the chapter → The Workflow Unit.



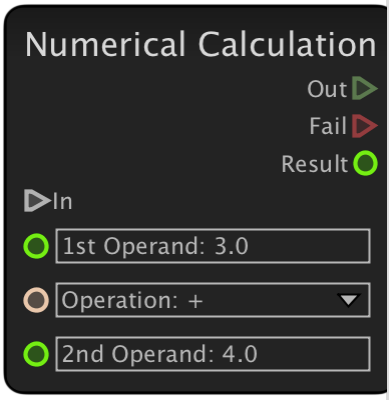
4.2 Connectr Execution Model

Connectr is based on a **sequential flow-of-control-model**, where execution order can be guaranteed and where data is delivered to Units in a well-defined and controlled manner. Instead of pushing data around and initiating execution, Units execute in an exactly defined sequence, **receiving** and **transmitting** data from Units and ports whenever required.

Workflows and therefore Units **do not** have a regular global execution interval. **Workflow and Unit execution must be initiated by Events and Triggers.**

4.3 Unit Anatomy & Ports

4.3.1 Unit Anatomy

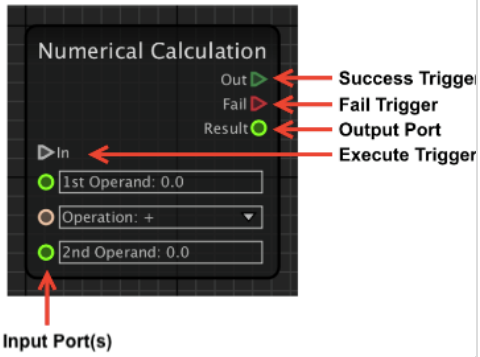


Exemplary connectr Unit

- This exemplary Unit adds two numbers and provides the result.
- Units have **attributes** which can be modified by the user – or other Units.
- Units can **exchange data** via **Cables**. Think of them as small 'tunnels' where data is able to travel between the Units.
- Cables are usually connected/disconnected by the user, but there are also situations where Cables between Units are automatically re-created.
- On the left and right side of each Unit there are small **circles** and **triangles**. These are the **connection points** of Cables and they are called **Trigger Ports**, **Input Ports** and **Output Ports**.

An **connectr Unit** can be seen as a **graphical representation** of a **function** (as in traditional programming). It has one or multiple attributes.

4.3.2 Unit Ports



There are three types of **Ports**:

1. **Trigger Port:** Trigger the execution of a Unit (= **Execute Trigger**) or trigger the next Unit which is connected to one of the outgoing Triggers (= **Success/Fail Trigger**).
2. See chapter → Unit Triggers & Events.
3. **Input Port:** These Ports accept data from the User or other Units. They can also represent the attributes of a Unit. Not all Input Ports can be modified by the User, some of them only accept data from other Units as their input, i.e. a Cable has to be connected to them (for example Structure-based Units). Once a Port is connected with a Cable, it can not be altered by the user anymore.
4. See chapter → Unit Attributes.
5. **Output Port:** These Ports output data and may be connected to other Units.

4.3.3 Unit Types

Units can be categorised in **three types.**

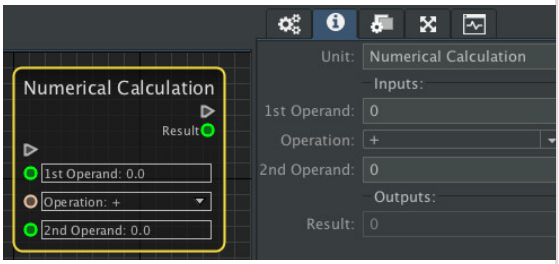
Unit Type	Description	Example
Provider Unit	They only have one or multiple **Output Ports** and "provide" something.	Variable Getter, System Timestamp

Processing Unit	They have one or multiple **Input and Output Ports** and process or modify their input.	String Replace, String to Time
Consumer Unit	They only have one or multiple **Input Ports** and "consume" data.	Print String, Variable Setter

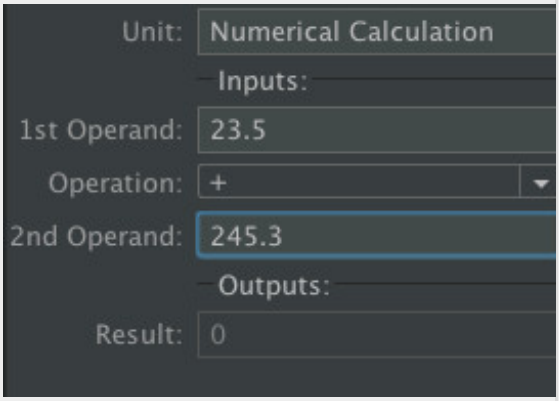
4.4 Unit Attributes

The **attributes** of a Unit are represented by **Ports** (as described in chapter → Unit Anatomy & Ports). This chapter describes how to modify the attributes of a Unit.

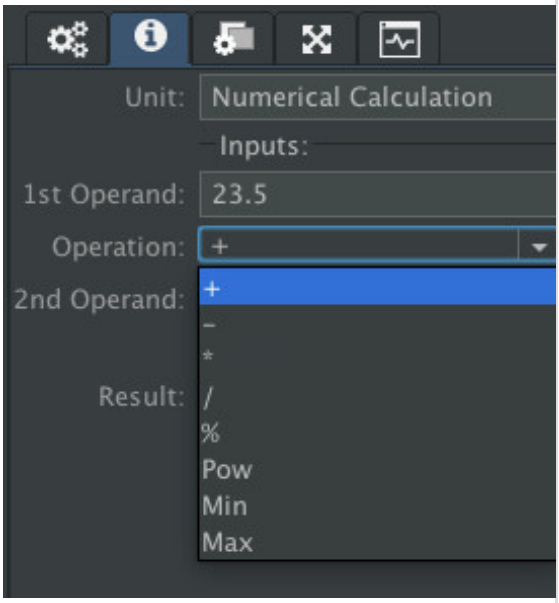
4.4.1 Modifying Units via the Inspector



1. In the Workflow View **select the Unit** you want to modify. Selected Units are marked with a **yellow** border.
2. Go to the **Project Panel** (shortcut: command + shift + P).
3. Click the **Inspector Tab.**
4. The **Unit's attributes** will show up.

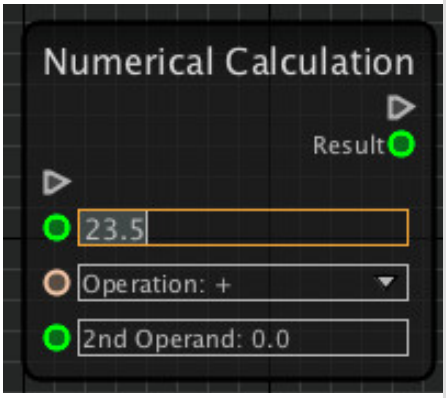


1. Below the **Inputs** divider you can modify each Output Port by clicking the respective **text field**, **checkbox** or **drop-down menu**.
2. Enter your value.
3. Press the **enter key** to accept the value.
4.
   - Pressing the **escape key** after entering a value **rejects the change** and **restores the previous value**.
   - **Outputs** cannot be modified directly, but in the Inspector they are listed to inform you about their current value (where possible).
   - Input Ports with an attached Cable cannot be modified in the Inspector.
   - Invalid values (for example text in a number Input Port) will be rejected.



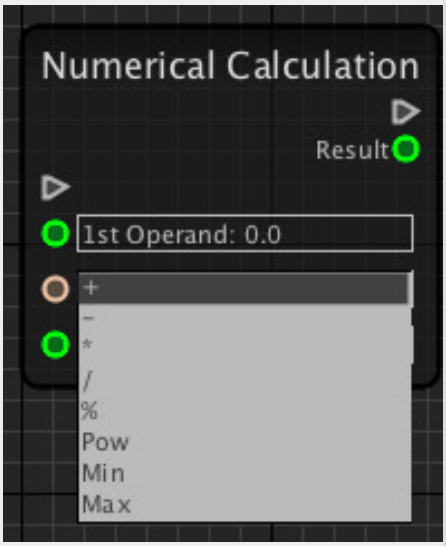
There are different attribute types, as described in the chapter → Data Types & Color Coding. In this figure you can see a number value and a drop-down menu.

4.4.2 Modifying Units Directly



1.In the Workflow View **choose the Unit** you want to modify. **Click** (or double click) the attribute you want to modify.
2.Enter your value.
3.Press the **enter key** to accept the value.
4.

- Pressing the **escape key** after entering a value **rejects the change** and **restores the previous value**.
- Connected Input Ports cannot be modified directly.
- Invalid values (for example text in a number Input Port) will be rejected.

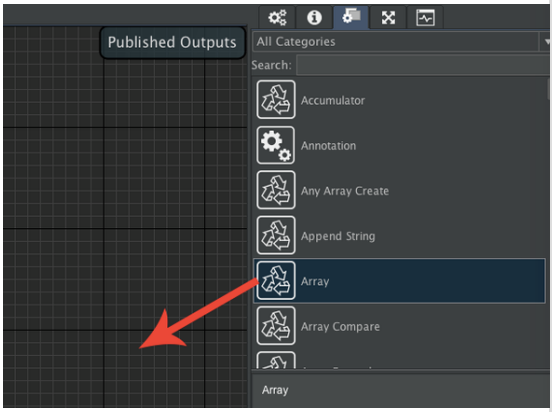


There are different attribute types, as described in the chapter → Data Types & Color Coding. In this figure you can see a number value and a drop-down menu.

There are attributes which **cannot** be modified via the Inspector or directly on the Unit. These are – for example – **Arrays**, **Structures**, **Trigger Ports**, and **Images**.

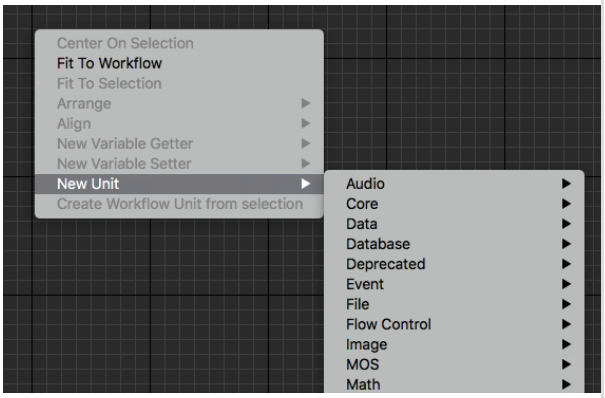
4.5 Unit Creation & Deletion

4.5.1 Adding Units via the Library



1. Go to the → **Project Panel**.
2. Click the **Library** tab.
3. Select the desired Unit.
4. **Drag the Unit** into the Workflow View **or Double click** on it to add it to the Workflow View – or press the **enter** key when the unit is selected.

4.5.2 Adding Units via the Context Menu



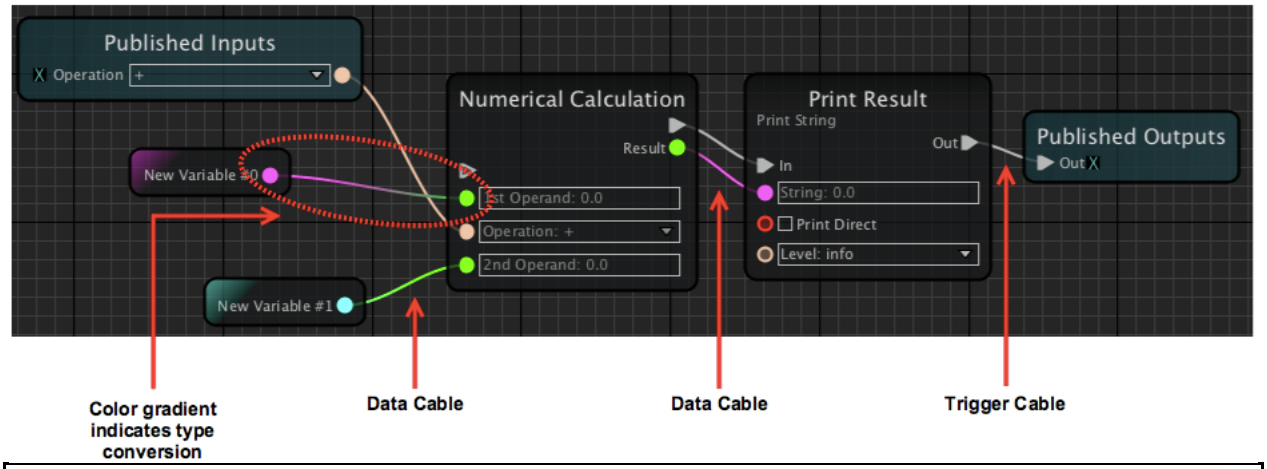
1.**Right click** somewhere in the → **Workflow View**.
2.Select the **New Unit** menu item.
3.Select the desired Unit you want to add.

4.5.3 Deleting a Unit

1. Select the Unit(s) you want to delete in the Workflow View.
2. Press the **backspace key**.

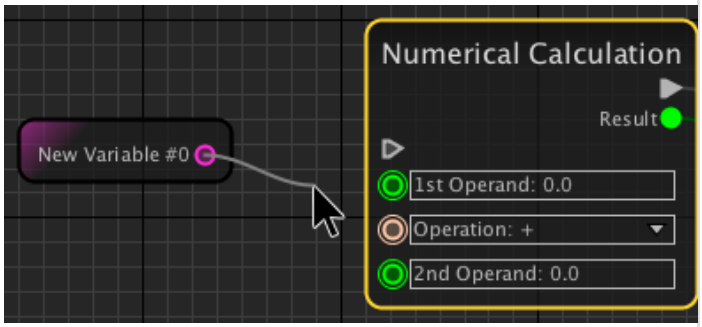
4.6 Cables

Cables **transmit data** between Units and are also used to connect Trigger Ports. **Ports** are the connecting points of Cables.



Although connectr designer does **type conversion** automatically, not all types are compatible with each other. For example you can't connect a **String Port** to an **Image Port** (or vice versa). Connectr designer **refuses** impossible connections, so you will not be able to connect Ports which are not compatible.

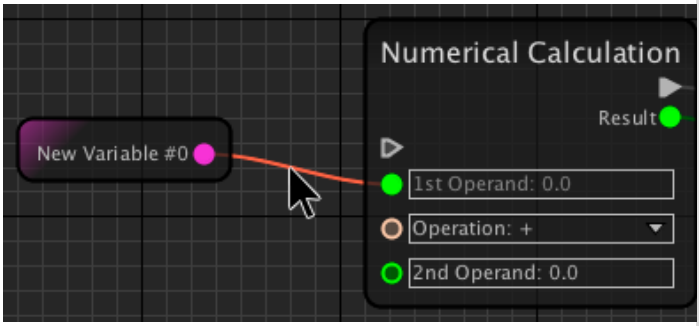
4.6.1 Connecting a Cable



1. In the Workflow View, click on the **source Port**.
2. Draw the Cable to the **destination Port**.
3. Click the **destination Port** to connect the Cable.

Cables can also be drawn "reverse style", i.e. from destination to source.

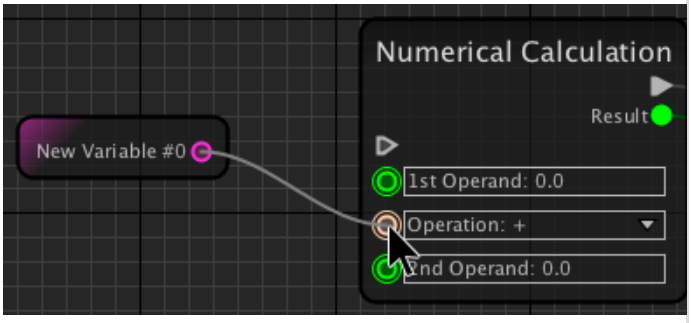
4.6.2 Removing a Cable



1. In the Workflow View, **select the Cable** you want to disconnect.
2. It will be **highlighted**.
3. Press the **backspace key**.

Alternatively you can grab the Cable on the destination port and just release it on some empty area of the Workflow View.

4.6.3 Rewiring a Cable



1. In the Workflow View, **click on the Port** to which the Cable you want to rewire is connected.
2. Connect it to **another Port** the same way as described above.

4.7 Unit Triggers & Events

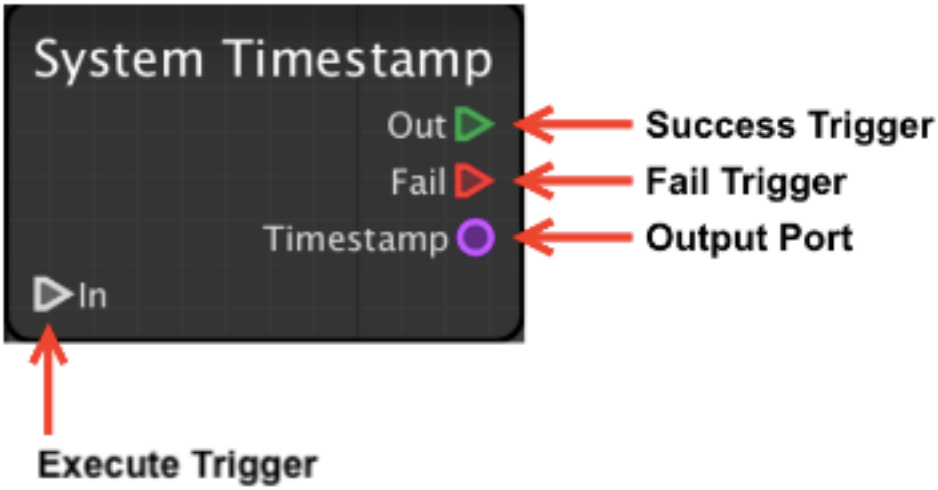
As already mentioned in the chapter → connectr Execution Model the **execution model** of connectr is based on a **sequential flow-of-control-model**.

Triggers and **Events** must be used to **initiate the execution of Units**. Therefore a **Unit** where the **Execute Trigger** is not wired will not do anything.

4.7.1 Unit Triggers

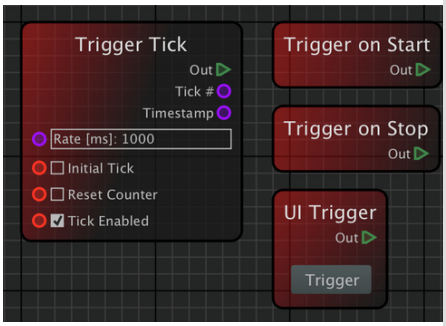
Triggers are represented by grey, green and red triangles. They cannot be wired to anything else than their own kind.

Trigger Type	Trigger Representation	Trigger Description
Execute Trigger		This trigger gets fired by the **execution flow** and the Unit will be called to execute.
Success Trigger		Upon completion, this trigger will be automatically fired and the next Unit, which has its **Execute Trigger** wired to this **Complete Trigger**, will be executed next.
Fail Trigger		If something goes wrong while a Unit is being executed, the **Fail Trigger** will be fired instead of the Success Trigger.



Units usually have two output Triggers (**Success** and **Fail**), which can be used to **branch** to individual flows. These triggers precisely define the execution flow of the Workflow. As a consequence to the strictly controlled execution flow, it is **not** possible to wire a Trigger to multiple other Triggers. If this is required please use the appropriate Units, such as **Multi Trigger**, **Branch** and **Trigger Join**.

4.7.2 Event Units



Event Units are used to trigger execution of an entire Workflow, or just a subset of a Workflow, or a Subworkflow. This is a very powerful concept, allowing Workflows to be triggered from any Workflow condition, but also – for example – from network events. The simplest example is the **Trigger on Start** event.

Event Type	Event Description
Trigger on Start	Sends a Trigger when the Workflow is **started**.
Trigger on Stop	Sends a Trigger when the Workflow is **stopped**.
UI Trigger	Sends a Trigger when the **Trigger Button** has been clicked.
Trigger Tick	Sends a trigger based on a **time interval**. If the old-style method of **regular execution** is required, this Unit can be used.

The **UI Trigger Unit** is very helpful to quickly test a Workflow or a certain part of a Workflow.

4.8 Data Types & Color Coding

Connectr designer uses **color coding** for **data types**. Cables have the same color as the **Output Port** they originate from. If a cable has a color gradient, type conversion happens, for example turquoise to pink means **Integer** to **String**.

Not all types can be converted to another type. Sometimes you need to use one or several Units to convert one type to another.

Port Appearance	Port Type	Description	Example
	Boolean	Binary variable.	false, true
	Integer	32-bit number without a fractional component.	11, 23, -7
	Long	64-bit number without a fractional component.	11, 23, -7
	Index	Positive number without a fractional component. Used for drop-down boxes.	1, 2, 55, 93
	Timecode	Timecode in HH:MM:SS:FF format.	11:45:12:23

	Any	Any type.	–
	Structure	Key-Value pairs.	–
	Bytes	8-bit Values.	111, 110, 67, 111, 114, 101
	Double	Number with a fractional component.	124.22, -45.95, 256
	String	One or multiple characters.	Hello World, 23, $48
	Array	Array of data.	–
	Color	Color in RGBA format.	#FF33ADFF
	Image	Image.	–

String Ports can be connected to **Byte** Ports.

4.8.1 Other Port Types

Port Appearance	Port Type	Description
	Execute Trigger	This trigger gets fired by the **execution flow** and the Unit will be called to execute.
	Success Trigger	Upon completion, this trigger will be automatically fired and the next Unit, which has its **Execute Trigger** wired to this **Complete Trigger**, will be executed next.
	Fail Trigger	If something goes wrong while a Unit is being executed, the **Fail Trigger** will be fired instead of the Success Trigger.
	Password	String Port with obscured password. Can only be connected to Password Ports. There are Units to convert a Password Port to a String Port (and vice versa).

It is **not** possible to wire a Trigger to multiple other Triggers. If this is required please use the appropriate Units, such as **Multi Trigger**, **Branch** and **Trigger Join**.

4.9 Variables

Variables are used to store and retrieve data which may be used globally and/or on multiple positions in a Workflow at the same time. There are two types of Variables:

Variable Type	Description
Project Variable	This type of Variable can be accessed globally by all Workflows of your Project. Project Variables are defined and modified using the **Variables tab** in the → **Project Panel**.
Local Variable	This type of Variable can only be accessed locally inside of a Workflow. Other Workflows cannot access them. They are especially useful if you need to transmit data over long distances and they help avoid drawing Cables which may clutter up the structure of your Workflow.

Variables can contribute much to a well structured Workflow and reduce the usage of Cables. At the same time, overusing them can obfuscate a Workflow. **Properly annotate your Workflows when making extensive use of Variables!**

4.9.1 Project Variables

	• Project Variables are defined in the **Variables tab** in the → **Project Panel** and require at least a **name** and a **data type**. • The content (or value) of a Variable is initially empty, 0 or false and can either be defined in the Variables tab or via a **Variable Setter** Unit. • Variable Units are color coded regarding the Unit background, Output/Input Port, and the connections from and to the Port.
	Project Variable Getter • This Unit is used to **get** the content of a Project Variable, that means its **Output Port** provides the content you set for that specific Variable in the Variables tab. • When changing the name of the Variable in the Variables tab, the Getter's name is also updated automatically.
	Project Variable Setter • This Unit is used to **set** the content of a Project Variable, that means the content of the Variable defined in the Variables tab will be changed accordingly when data is transmitted to the **Input Port** of this Unit.

Project Variables can be used **globally** in certain Units (JavaScript, Shell Command). To get the content of a Variable, enclose its name in '$', for example "**$myVariable$**".

4.9.2 Local Variables

	• Local Variables are created with Units which you can find in the **Library** (see → Project Panel).
	Local Variable Setter • This Unit is used to define and set a Local Variable. • You need to set a **Type** and a unique **Variable Name**. • After defining the Variable Name, a new **Value Port** will appear which can be used to set the Variable content.
	Local Variable Getter • This Unit is used to retrieve a local Variable. • After defining the Variable Name, a new **Value Port** will appear which can be used to retrieve the Variable content.

When there are **multiple instances** of a Local Variable Setter with the same name, changing the type on one of them will change the type of all other Setters (with the same name).

4.9.3 Variable Types

Refer to the chapter → Data Types & Color Coding.

4.10 Published Inputs and Outputs

The **Published Inputs** and **Published Outputs Units** are different from other Units. They can be seen as the "**interface**" which can be accessed by other Workflows (or Applications). Each Workflow has exactly one **Published Inputs** and one **Published Outputs** Unit. They are initially empty.

4.10.1 Adding a Published Input/Output

The steps are the **same** for Published **Inputs** and Published **Outputs**.

	1.Go to the **Workflow View**. 2.Look for the **Published Inputs** Unit.
	1.On the desired Unit, **draw a cable** (as explained in chapter → Cables) from the **Input Port** you want to add as a Published Input. In the screenshot on the left this is the **String** Input Port. 2.Hover the Cable over the **Published Inputs** Unit and release it. 3.A new Published Input will be created on the Published Input Unit.
	Result • The **Published Input** has been created for the **String** Input Port. • Note that the Published Input has the same **Name** and **Data Type** as the originating Input Port.

4.10.2 Editing a Published Input Value

1. Go to the **Workflow View**.
2. Look for the **Published Inputs** Unit and select it.
3. Open the **Project Panel** and head to the **Inspector** tab.
4. You should now see all Published Inputs of the selected Published Inputs Unit.
5. Change the desired values (as described in chapter → Unit Attributes).

Changing the **type** of a Published Input is **not** possible. You have to remove the respective Input Port first and then add a new one.

4.10.3 Renaming a Published Input/Output

	1.Go to the **Workflow View**. 2.Look for the **Published Inputs** Unit. 3.**Double-click** the Name of the desired Input Port. 4.The **Name** will be highlighted and can then be changed. 5.Press the **enter key** when you are done.

4.10.4 Removing/Rewiring a Published Input/Output Port Connection

	1.Go to the **Workflow View**. 2.Look for the **Published Inputs** Unit. 3.**Single-click** the **Cable** which originates from the desired Published Input. 4.Press the **backspace key** to remove the connection. 5.The Published Input stays intact and can be rewired to any other Unit.

4.10.5 Removing a Published Input/Output

1. Go to the **Workflow View**.
2. Look for the **Published Inputs** Unit.
3. Click the
4. of the Published Input you want to remove.

4.10.6 Reordering Published Inputs/Outputs

1. Go to the **Workflow View**.
2. Look for the **Published Inputs** Unit.
3. Hold the **ctrl** key.
4. **Click** the name of the Published Input and **drag** it to a new position.

4.11 The Workflow Unit (Subworkflows)

	It is possible to 'call' another Workflow with the **Workflow Unit.** This is especially useful to structure your main Workflow and to reduce the (visual) complexity of it. A Workflow which is called via a Workflow Unit can be used like a standard Unit by defining **Input** and **Output Ports** (as described in chapter → Published Inputs and Outputs).

4.11.1 Using the Workflow Unit

There are two ways of using the Workflow Unit.

	Using the Context Menu 1. In the Workflow View, select the **Units** you want to put into a separate Workflow. 2. Right-click one of the selected Units and select **Create Workflow Unit from selection** from the context menu. 3. You will be asked to provide a **name** for the new Workflow. 4. The Units are moved into a separate Workflow and a Workflow Unit is automatically created and placed into the current Workflow. If there were any connections to the respective Units, the **Input and Output Ports** are automatically created and connected.
	Using the Workflow Unit 1. Add a **new** Workflow to your Project. 2. Add the desired Units to this new Workflow. 3. Publish Inputs/Outputs (as described in chapter → Published Inputs and Outputs). 4. Select the Workflow which will call the previously created Workflow. 5. Add a **Workflow Unit** to this Workflow. 6. Select the desired Workflow in the "**Workflow" drop-down menu**. 7. Once selected, the Workflow Unit will change its appearance and shows the Inputs/Outputs as well as a preview of the selected Workflow.

4.11.2 Editing a Workflow Unit

Editing a Workflow directly

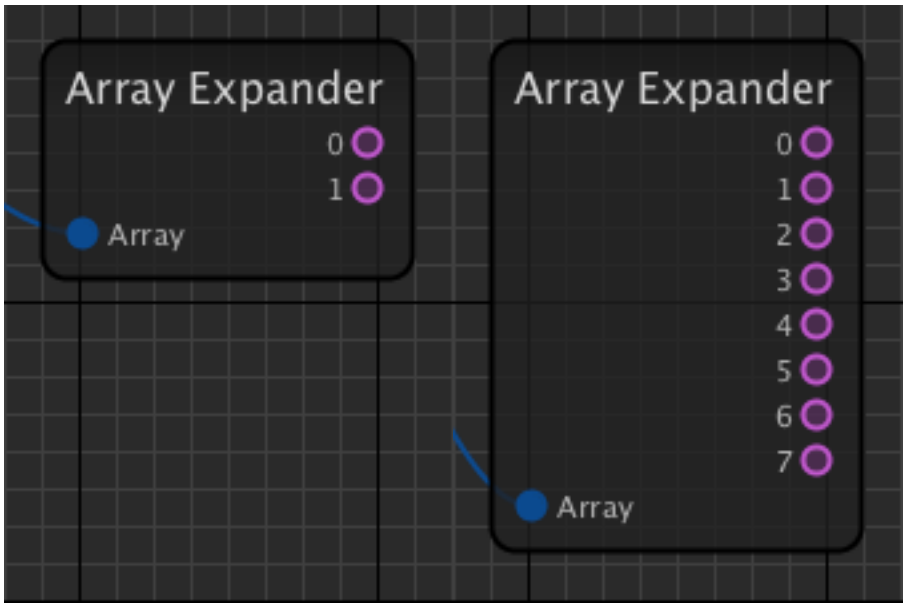
1. Click the **tab** of the the **Workflow** you want to edit. If the respective tab is not visible, open the Workflow first by using the **Workflows List** in the **Project Panel**.
2. Make your edits.
3. Head back to the **previous Workflow** by clicking its **tab**. The Preview will have changed accordingly to your changes and Input/Output Ports are also updated (if they changed).

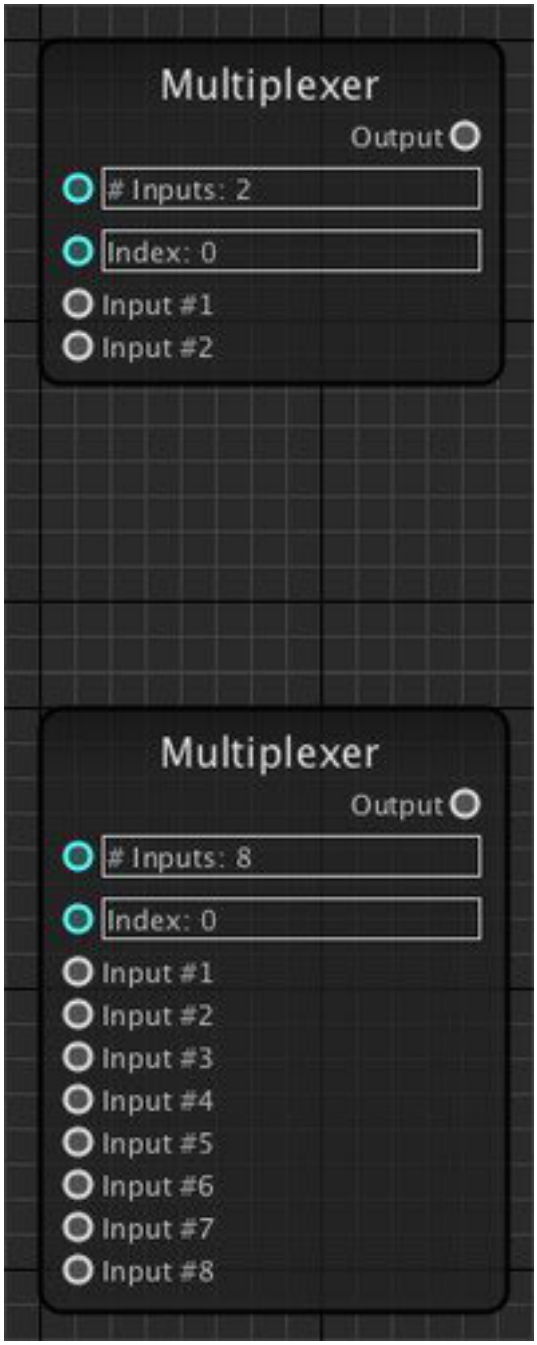
Editing a Workflow via the Workflow Unit Preview

1. **Double-click** the **Workflow Preview** on the **Workflow Unit.**
2. Make your edits.
3. Head back to the **previous Workflow** by clicking its **tab**. The Preview will have changed accordingly to your changes and Input/Output Ports are also updated (if they changed).

4.12 Dynamic Ports

Units may have dynamic ports which means that the **number of Input/Output Ports** may change **dynamically**. This can either be specified by the user itself (by defining the number of Inputs/Output Ports) or based on the type of data which arrives at the Unit.

Type	Example
Based on data	

User Defined (may also be based on data)	

Please note that a drastic increase of dynamic Ports may result in slow Workflow execution.

4.13 Pending Connections

When creating cables in connectr designer, those connections are remembered when Units in the execution flow are being deleted. That means when undoing a Unit deletion, the cables will

be restored – where possible. There are also pending connections. They can occur on Units which dynamically change their Ports (→ Dynamic Ports). When a source or destination Port is not available for a certain amount of time a pending connection will be dropped, which means it will not be recovered and has to be connected again. The Message Log informs you about a dropped connection.

Dropped connections cannot be undone.

4.14 Annotations

Annotations can be used to document your Workflows and to add important information or notes.

Annotations are a requirement for reusable, maintainable Projects and we strongly recommend using them.

4.14.1 How to Use

Annotations can be added via the Unit Library. Use the Inspector to customize them.

4.15 Exposed Functions

Exposed Functions are used to **execute certain functions of a Unit**. Instead of a Unit which provides dozens of Input and Output Ports, a Unit may provide several Exposed Functions. This approach enables a much more structured Workflow View and avoids unnecessary visual noise. It also allows you to reuse functions of a Unit at multiple points of your Workflows.

	Exposed Function Parent Unit The **drop-down menu** is used to **select the Exposed Function** you want to create. Use the + button to create the desired Exposed Function.
	Exposed Function Child Unit Exposed Function Child Units have a sharp edge and the border is turquoise. They are used in the same way as common Units.

All Exposed Units will be deleted when the parent Unit gets deleted.

4.16 JavaScript Unit

	The JavaScript Unit can be used to write and execute JavaScript. This Unit uses the **connectr Code Editor** which can be accessed via the **Inspector**.

4.16.1 Declaring Input and Output Ports

Declare Input Ports with **i_MyVar** and Output Ports with **o_MyVar.**

4.16.2 Input and Output Ports Data Types

There are five types of Input/Output ports:

- String
- Integer
- Double
- Long
- Boolean

Examples:

- **ib_myBooleanInput** would be an Input Port with Boolean data type.
- **os_myStringOutput** would be an Output Port with String data type.

4.16.3 Default Functions

By default there are three functions:

- **Startup**: is executed once when the JavaScript Unit is triggered after the Project is set into running state.
- **Shutdown:** is executed when Workflow stops executing (e.g. when the Project stops).
- **Execute:** is executed when the JavaScript Unit is triggered.

4.16.4 Using Project Variables

Project Variables can be used globally in certain Units (JavaScript, Shell Command). To use a Project Variable, enclose the Variable name in '$', for example “$myVariable$".

5. Tutorial Project

Several Example Projects are shipped with connectr designer.

5.1 Aims of this Tutorial

1. Adding **Units** to an empty Workflow.
2. Adding a **Trigger Unit**.

3. Connecting **Cables** between Units.
4. Setting **Unit attributes**.
5. **Running the Project**.
6. Using the **Message Log** to see Unit output.

5.2 Adding Units

Visual Instructions	Instructions	Notes
	• In the **Project Panel**, click the **Library** tab.	You can access the Library quickly by using the **option + space** shortcut.
	• Use the **search field** to find Units. • Look for "print". • **Double-click** on the **Print String** Unit to add it to the **Workflow View**.	After entering a search term you can directly jump to the list of found Units by using the **tabulator** key. Select the desired Unit and add it to the **Workflow View** by pressing the **enter** key.
	• The **Print String** Unit is now visible in the **Workflow View**.	
	• Add a **Trigger on Start** Unit.	After entering a search term you can directly jump to the list of found Units by using the **tabulator** key. Select the desired Unit and add it to the **Workflow View** by pressing the **enter** key.
	• Both Units are now visible in the Workflow View.	You can also use the **Align & Distribute** functions to reorder Units in the Workflow View.
	• Reposition the Units by dragging them. • Move the **Trigger on Start** Unit to the left and the **Print String Unit** to the right.	

5.3 Connecting Units

Visual Instructions	Instructions	Notes

	• Click on the **Out** Port of the **Trigger on Start** Unit. • Connect the **Cable** to the **In** Port of the **Print String** Unit by clicking on the **In** port. •	By connecting the **Trigger Ports** you make sure that the **Print String** Unit gets **executed** after starting (= running) the Project. For detailed information about **Cables**, please refer to the chapter → Cables. For detailed information about **Unit Triggers**, please refer to the chapter → Unit Triggers & Events.

5.4 Modifying Unit Attributes

Visual Instructions	Instructions	Notes
	• Double-click the "**String:**" Port on the **Print String** Unit.	Instead of modifying the Unit directly in the Workflow View you can also use the **Inspector** in the **Project Panel**. For more info refer to the chapter → Unit Attributes.
	• Enter "**Hello World**" and press the **enter key**.	You have now **modified** one of the **attributes** of the **Print String** Unit. The text you entered will be printed to the Message Log. For detailed information about **Unit Attributes**, please refer to the chapter → Unit Attributes.

5.5 Running the Project

Visual Instructions	Instructions	Notes
	• Open the **Project** menu. • Select **Run Project.**	The shortcut to run Projects is **command + R**. For detailed information about running **connectr Projects**, please refer to the chapter → Running an connectr Project.

	Project Running Indicator: This indicator is visible as soon as you run a Project.

	Message Log Entry: The Message Log shows the value which you have specified on the **Print String** Unit.

6. Running a connectr Project

Running an connectr Project means that all Workflows where the option **Execute On Run** is checked will be executed. Once a Project is running, events are triggered and data is transmitted between Units.

Workflows will only be executed when their **Execute On Run** option is set. This can be done in the **Project Panel**.

6.1 How to Run a Project

	1.Open the **Project** menu. 2.Select the **Run Project** command. **Command + R** can be used to run/stop a Project.
	At the top center of the **Workflow View** you can see if a Project is running. In addition a cog icon inside the **Workflow tabs** inform about the state of a workflow. Spinning cog means executing.

6.2 How to Stop a Project

1. Verify if the Project is running (by looking at the top center of the Workflow View).
2. Open the **Project** menu.
3. Select the **Run Project** command.

Command + **R** can be used to run/stop a Project.

7. Further Reading

- Best Practices
- Debugging / Workflow Analysis
- Workflow Profiler
- Connectr Control Protocol
- Menu Commands

7.1 Best Practices

Connectr designer is a Visual Programming Environment which makes Workflow, Microservice and Application development possible to almost everybody. As in any programming environment, it is also possible to create Workflows which are error prone, contain infinite loops or other problematic situations. If a Workflow should hang or crash, double check your implementation. If problems still persist don't hesitate to → contact us.

7.1.1 General Workflow Building

1. When building your Workflow, make sure you put more space between Units than what seems necessary at first. It is almost always the case that you will have to revisit sections and add systems later.
2. If you know that you will need a Variable in more than one place, it is advisable to create a Variable for it.
    - If the Variable is only used within one workflow, use **Workflow Variables** (not Project Variables)
3. Use Workflow Variables to cover large distances between units, to avoid unnecessary cable overlapping.
4. If you find yourself duplicating a system for multiple parts of your Project, encapsulate its functionality in a Subworkflow (using the Workflow Unit).
    - When designing a Workflow Unit it is important that you expose both in- and out-triggers, since Workflow units will only pass Variables to the Subworkflow on a trigger tick.
    - When you are done with your main Workflow, expose the Triggers, and any Variables that you deem necessary. This will allow someone else to reuse your workflow.
    - Provide necessary information about published Input Ports via an Annotations.

7.1.2 Main Bus Design

Having a well structured Workflow is crucial for larger Projects, as it severely decreases the overhead which is required to fix problems, implement changes and introduce other developers to the Workflow design. There are three "bus design" practices:

	Indistinct Units Layer Units of the same type vertically. This decreases the width of the Workflow and still allows for good structure.
	Distinct Units Layer Units (or Unit groups) horizontally. Adding Annotations is a good way to keep it structured.
	Branching Branching can be done by using the Workflow Unit, i.e. a group of Units is put together as a Subworkflow. Exposed Inputs and Outputs are used to send and get data to and from the Subworkflow.

7.2 Debugging / Workflow Analysis

Sometimes it is required to execute a Workflow step-by-step to understand and fix various issues. For this purpose connectr designer provides stepping, dataflow visualization and port/cable value visualization.

Functionality	Description
Stepping	A Workflow can be switched to **Step mode**. When activated, each action/event/transmission needs to be "stepped" manually.
Dataflow Visualization	When enabled, cables are being highlighted when data/triggers are transmitted.
Port/Cable Visualization	When enabled, ports and/or cables show which value they are currently receiving/sending/transmitting.

7.2.1 Workflow Stepping

Workflow Stepping means that each step (or execution) has to be **initiated manually** by issuing a **step command**. This is useful in situations where it may be required to narrow down issues or when a Workflow is too complex to follow when being executed at normal speed.

Using Workflow Stepping

1. To activate Workflow Stepping, go to the menu **Project → Enable Stepping** (Shortcut: command + B).
2. Run the Project by using the menu command **Project → Run Project** (Shortcut: command + R).
3. Issue the Step command by using the menu command **Project → Step** (Shortcut: command + shift + Y).

It may be helpful to activate the showing of **Port/Cable values** and also the **Dataflow Visualization** (see → Project Panel)

7.3 Workflow Profiler

7.3.1 Introduction

The Workflow Profiler is a tool to visualize Workflow and Unit performance. Both execution speed and memory consumption are profiled and visualized graphically.

Key design features of the profiler are:

- Minimal memory footprint (it uses <1k per profile entry).
- Minimal performance impact (less than 2% of added CPU usage when activated).
- Since sampling performance was always there, there is no additional runtime overhead for that, just memory profiling needed to be added.
- Only uses CPU time when the panel is active.

7.3.2 Overview

	1	**Settings:** Toggles between timing and memory profiling.
	2	**Workflow Reference Time:** Enter a value in milliseconds which is used to draw the Workflow timing bar (default = 20 ms, meaning 50 executions per second). • Increasing the value: The workflow timing profile bar gets recalculated and will be smaller. • Decreasing the value: The workflow timing profile bar gets recalculated and will be larger.

	3	**Workflow:** Shows the workflow name and the total execution time or memory consumption, depending on the profile type setting. The Workflow memory performance bar is calculated relative to the total memory assigned to the JVM. It should always be pretty small.
	4	**Units:** For each Unit in the active Workflow there's an entry describing the Unit's performance, either execution time or memory consumption. Variables and Published Input/Output Units are excluded from that list. The list is selectable and supports multiple and disjoint selections. If one or more Units are selected in the list, the corresponding Units are selected in the Workflow view, and vice versa. The Unit performance bars are relative to the Workflow performance. A Unit profile bar going fully to the right to red does not necessarily mean a bad thing – it just means that it consumes most of the workflow execution time or memory used.

7.4 Connectr Control Protocol

Connectr designer Projects can be remote controlled. For further information please → contact us directly.

7.5 Menu Commands

Some menu commands are not listed here. Common commands (like cut, copy paste) are not explained here.

- **Connectr designer**
  - **About Connectr designer:** shows version info and credits.
  - **Preferences:** shows the global Connectr → Preferences.
- **File**
  - **New Project:** closes the current Project and creates a new one.
  - **Open Project:** opens a previously saved Project.
  - **Open Recent:** shows recently opened Projects.
  - **Save Project:** saves the current Project.
  - **Save Project As:** saves the current Project as a new file at a user defined location.
- **Edit**
  - **Select Inverse:** inverts the current selection (i.e. selects all Units in the Workflow View which are currently not selected.)
  - **Select Upstream Units:** selects all Units which are connected "before" the currently selected Unit.
  - **Select Downstream Units:** selects all Units which are connected "after" the currently selected Unit.
  - **Duplicate Selection:** duplicates the selection. Please note that a large selection of Units may take its time to be duplicated.
  - **Create Workflow Unit from selection:** creates a separate Workflow from the current selection of Units. See chapter → The Workflow Unit.

- **View**
  - **Show Logs:** shows/hides the → Message Log.
  - **Show Project Panel:** shows/hides the → Project Panel.
  - **Show Next/Previous Workflow:** switches between Workflow tabs.
  - **Hide Non-Wired Ports:** hides Ports on Units which are not wired.
  - **Fit To Workflow:** adjusts the zoom level of the Workflow View to fit the whole Workflow.
  - **Fit To Selection:** adjusts the zoom level of the Workflow View to fit the current selection.
  - **Center On Selection:** centers the Workflow View on the current selection.

- **Project**
  - **Run Project:** runs the Project.
  - **Enable Stepping:** enables/disables → Workflow Stepping.
  - **Step:** when Workflow Stepping is activated, use this command to step the Workflow execution.
  - **New Workflow:** adds a new Workflow to the current Project.
  - **Duplicate Workflow:** duplicates the currently active Workflow.
  - **Delete Workflow:** deletes the currently active Workflow.
  - **New Variable:** adds a new Variable to the Project.
  - **Edit Frontend Code:** shows the frontend code editor.
  - **Run Frontend Code:** runs the frontend code.

- **Search**
  - **View Online User Manual:** opens the connectr User Manual in your web browser.

8. Limitations & Known Issues

8.1 Version 1.0

1. The **Array Iterator** "Element" Trigger cannot be used as a Published Output in a Subworkflow.
2. The **OS Notification** Unit's notifications are reported as "Script Editor" notifications in the **Notification Center** of OS X/macOS.

8.2 Version 1.0.1

1. The **Trigger Tick** Unit's **Out Port** cannot be published from within a Workflow.
2. The **Auto-Clear Log** option in the Connectr designer preferences does currently not clear the **Message Log** when opening another Project. It does clear when running a Project, though.