HORUS VIEW & EXPLORE

TECHNICAL DOCUMENTATION

Getting Started with Linking Lab

Horus View & Explore
info@horus.nu

May 15, 2020

Contents

1 Introduction

First of all, we at Horus View & Explore would like to thank you for using Linking Lab and our System V2 framework.

The aim of this guide is to give you a basic introduction to the Linking Lab software, giving you the tools to create and deploy so-called pipelines. This document describes the installation process, user interface, basic functions of the application, and at the end a step by step description of how a pipeline is built.

For more information on Horus Linking Lab, see the Horus Handbook.

2 What is Linking Lab

2.1 The Horus Linking Lab Application

2.1.1 What is linking lab?

Horus Linking Lab is a front-end application that allows users to access systems/devices running the Horus System V2 framework.
Within Linking Lab you view, design, edit, and deploy pipelines.

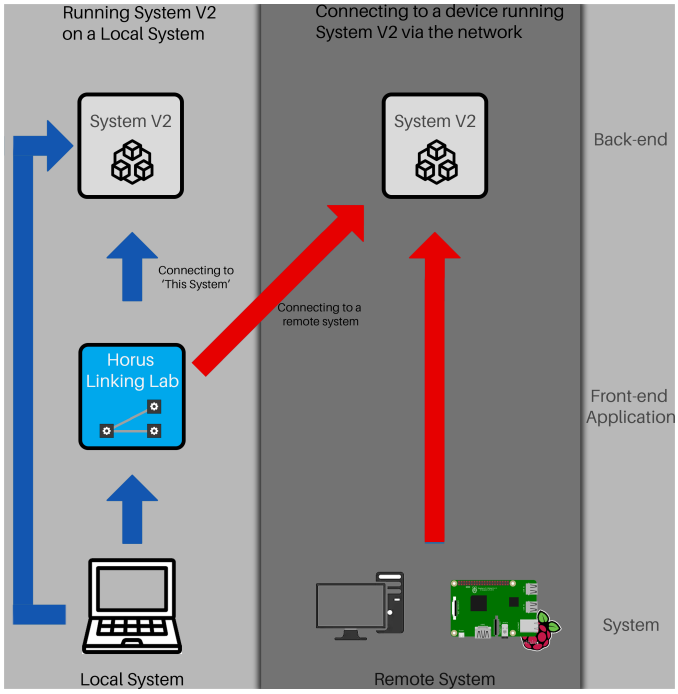


Figure 1: System Architecture

In figure 1 you can see how the Horus software is set up.
The blue arrows shows how the system functions when you create and edit a System V2 instance on a local machine. By connecting to the local system a System V2 instance is created. The user can then configure the framework through Linking Lab.
Due to how the system is designed, users are also able to connect to other systems or devices running System V2 instance, this is show by the red arrows. You open Linking Lab on your own local device, and you can then

connect to a System V2 instance that is running on another PC or embedded device, for example.

System V2 is a modular framework, that primarily consists out of pre-made 'building' blocks referred to as components. By combining these blocks, they can accomplish complex tasks.
These components, besides a set of free ones, are individually licensed and can be bought separately depending on the users requirements.
Some of the major benefits of our framework are that it allows users to easily control and configure embedded devices or remote systems from a local system that is on the same network. Another benefit is that our system can work with almost any hardware device and/or SDK.
Our System V2 framework forms the basis of our verticals, acting as the engine that enables us to quickly create software for mobile mapping, and security purposes.

The benefits of this platform are that it allows you to quickly design and deploy your created software solutions, using an accessible visual user interface, that does not require programming or deep technical knowledge, this results in quicker development times, and lower costs.

2.2 Linking Lab Components

2.2.1 What are Components?

Components are the building blocks from which pipelines are built. Every component has a specific function, and by using multiple in concert, they can accomplish more complex tasks.

Each component has one or more in/output pipes. Data referred to as messages are received through the input pipe, whilst the output pipe, conversely, sends outgoing messages. You can send messages from one component to another by linking the output pipe of a component to the input pipe of another one.

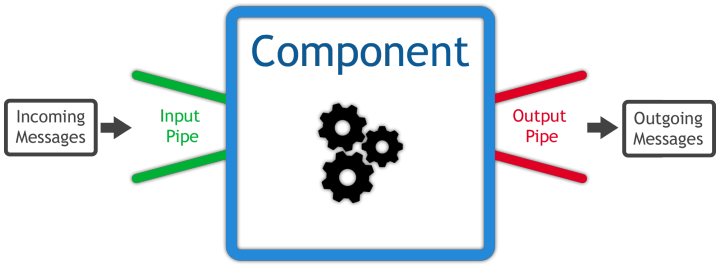


Figure 2: Component data flow

Each component has a set of configurable properties that allows them to be more flexible.

The components can be divided into three different main groups.

The first group are data **grabbers**.
These components can output data from camera's, GPS and IMU devices, and the network, to name a few.
These usually only have an output pipe.

The next group are components that **process** data.
They get data from other sources, then process it, before outputting the processed data. These type of components may convert, en/decode, and en/decrypt data, to name a couple a couple of examples.

The last group of components are data **writers and streamers**.
They get information from other sources and before sending this data to either the network or a hardware device.
These are characterized by the lack of any output pipes.

In the *Component Browser* you can find the full list of installed components and read about their specific functionality.

3 Installing Linking Lab

3.1 Introduction: Installation of Linking Lab

To start working with the Linking Lab software, you will require a working installation of said program. In this chapter you will find the instructions for installing the application on your system.

3.2 The Installation Process on Windows

To install the program on a Windows system, follow the steps as described below.

Step 1.
To start the installation the software, an installer is required.
Open an internet browser and enter the following into the address bar:
embed.horus.nu

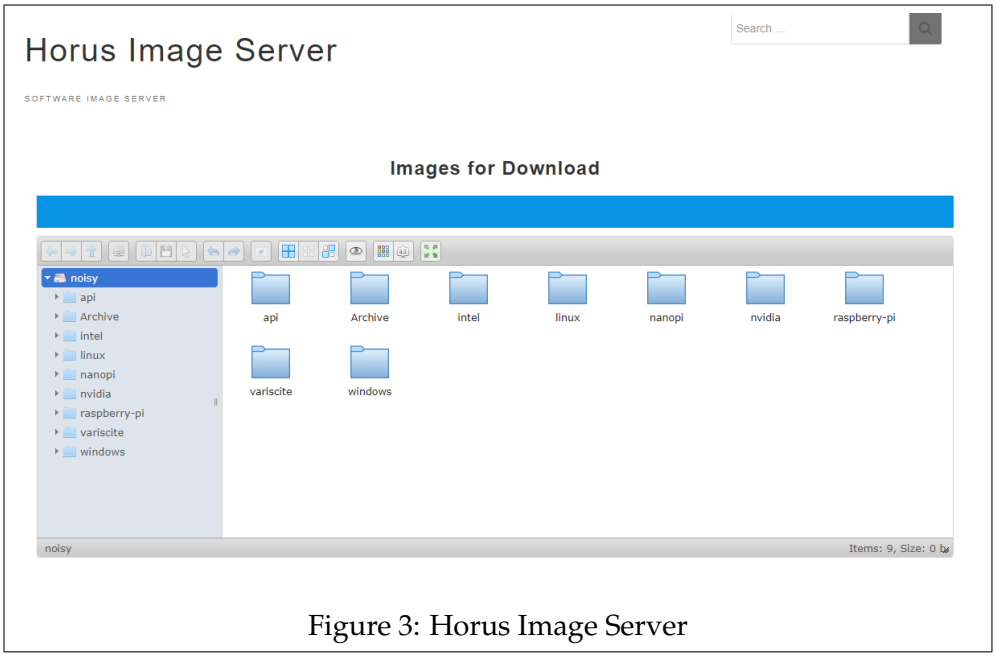


Figure 3: Horus Image Server

A page like the one above should appear.
To find the installer, click on *Windows->x64->LinkingLab*

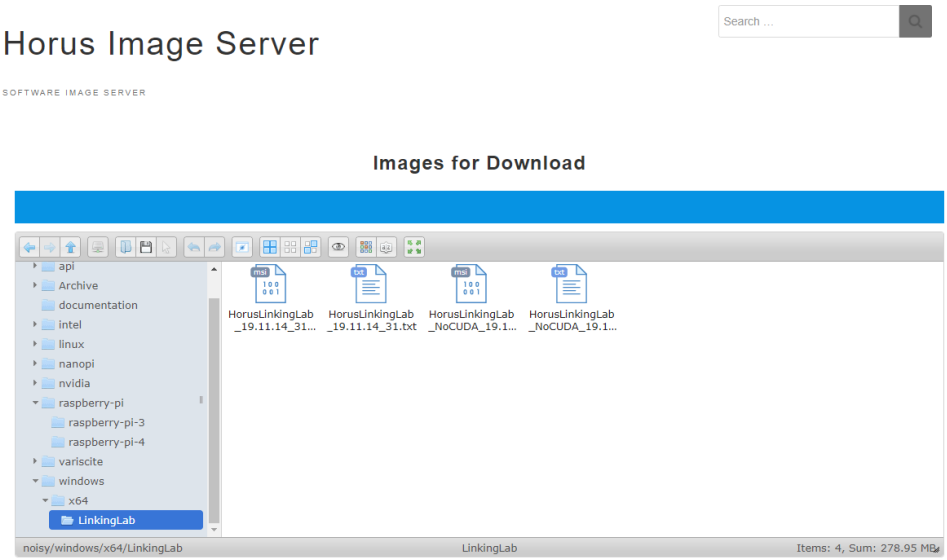


Figure 4: Linking Lab installers

You can then choose the desired installer. Double-click on the icon to start the download.

Step 2.

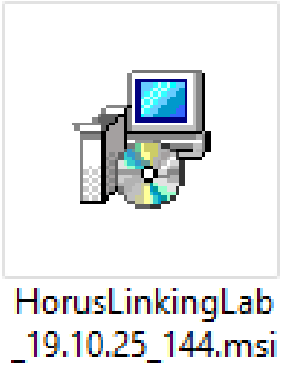


Figure 5: The installer

Once the installer has downloaded, run it to start the setup wizard.

Step 3.

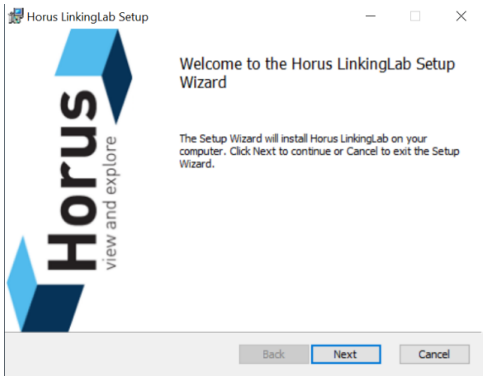


Figure 6: Setup Wizard

Once the Setup Wizard has started up, select *Next* to start the installation procedure.

Step 4.

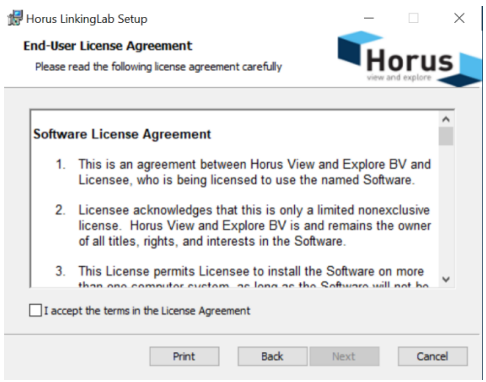


Figure 7: License Agreement

Before continuing ensure that you have read the *Software License Agreement*.
Check the box to accept the terms in the agreement, and press *next* to proceed.

Step 5.

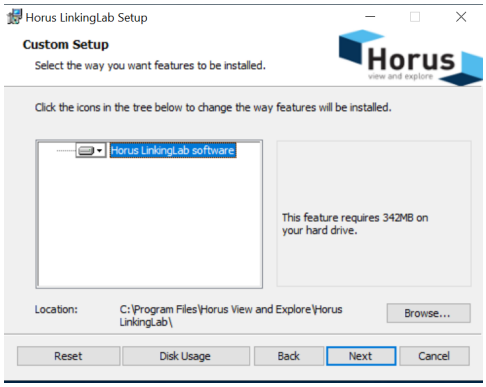


Figure 8: Install Location

You may change the location where Horus Linking Lab will be installed.
Click *Next* when the install path is set to the desired location.

Step 6.

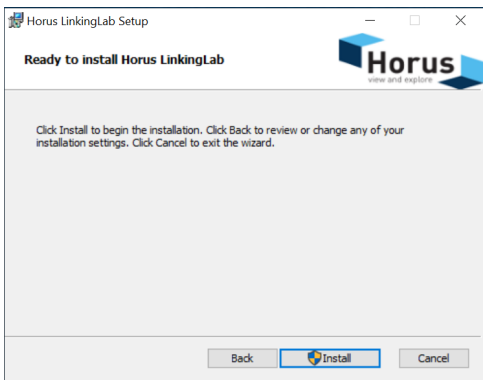


Figure 9: Start Installation

The setup wizard requires final approval before it can install Horus Linking Lab.
Click *Install* to start the installation.

Step 7.

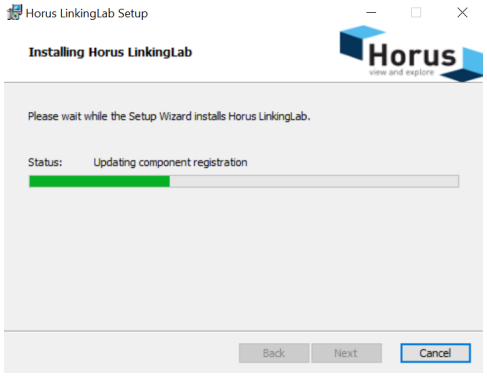


Figure 10: installation Progress Screen

The setup wizard will start installing the program, the progress can be followed by a status bar. Wait for the installation to finish

Step 8.

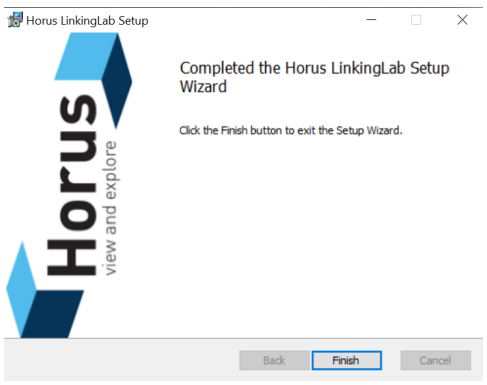


Figure 11: Finish Installation

Once the installation has finished, select *Finish* to close the setup wizard.

Horus Linking Lab is now successfully installed on your system.

3.3 The Installation Process on Linux

This chapter describes how to acquire a working image of Linking Lab on your system.

Step 1.
The first step is to download the *.AppImage* of the Linking Lab application. Open an internet browser and enter the following into the address bar: *embed.horus.nu*

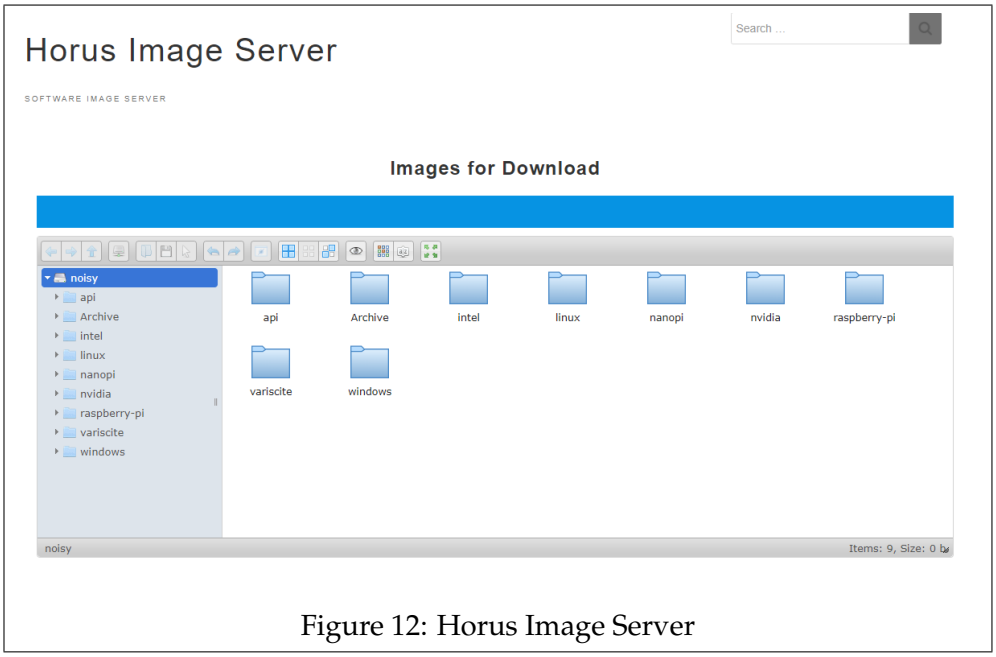


Figure 12: Horus Image Server

A page similar to the one above should appear.
To find the .AppImage, click on *linux->x64->LinkingLab*

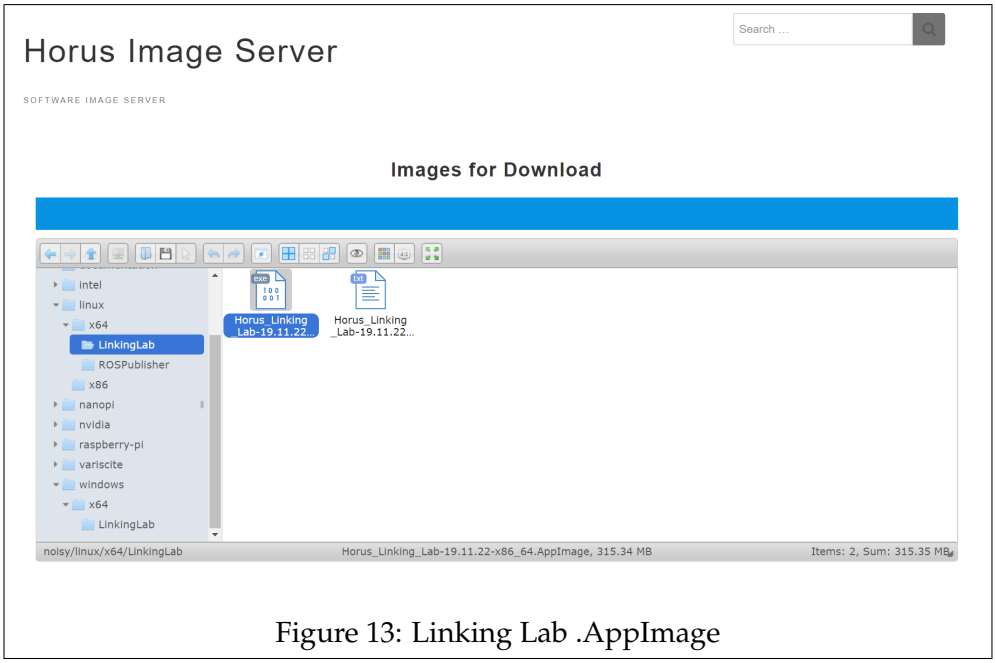


Figure 13: Linking Lab .AppImage

Download the .AppImage and proceed to the next step.

Step 2.
.Appimages do not require an installation, but the permissions might need to be changed.

Changing Permissions through the File Manager
If your OS supports it, find the .AppImage in the File Manager and open the Properties, and then look for options to change the permissions. Find the option that changes the Executable permissions and ensure that, you, the user, is able to run the .AppImage. See Figure 14 for an example.

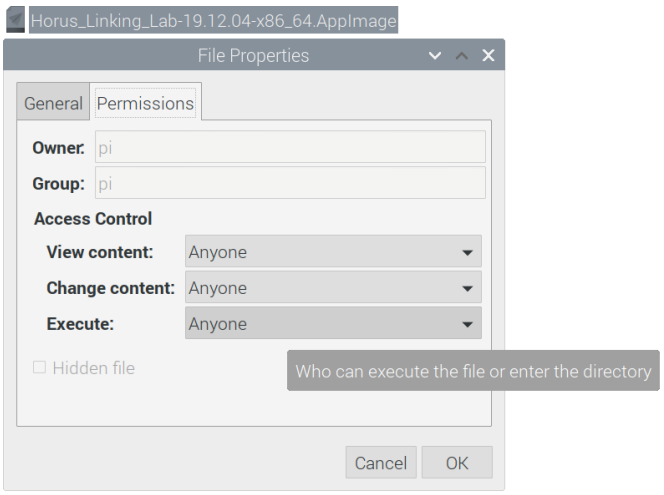


Figure 14: Changing Permissions through the File Manager

If your OS does not support this feature, then you can also change the permissions via the terminal.

Changing Permissions with the terminal
Open the terminal and navigate to the downloads location.

```
[13:55] lute @ ubuntu-gnu-linux: ~
$ cd Downloads/
[13:56] lute @ ubuntu-gnu-linux: ~/Downloads
$ ls -l Horus_Linking_Lab-19.11.22-x86_64.AppImage
-rw-rw-r-- 1 lute lute 330661864 nov 26 13:45 Horus_Linking_Lab-19.11.22-x86_64.AppImage
```

Figure 15: The Location of the .AppImage

Step 3.
Once you are in the downloads directory, locate the .AppImage, and check the permissions.
The permissions in figure 15 indicate that we only have read and write permissions.
In this case we require an *X*; for executable permissions to run the program.

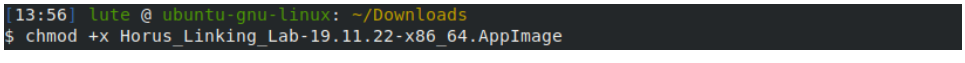


Figure 16: Changing Permissions of the .AppImage

enter the following: *chmod +x (the .appImage file name)*
This will give the user permission to run the .AppImage.

Step 4.
Check that you now have executable permissions, and then run the program.

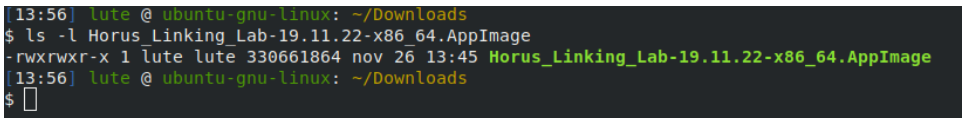


Figure 17: .Appimage with the Altered Permissions

After following these steps, you should now have a working version of Horus Linking Lab on your system.

4 License Keys

4.1 Introduction: License Keys

Linking Lab comes with a set of free components, but if you wish to use other components, a license is required.

Licenses can be purchased for individual components, thus resulting in only having to acquire the components that you require.
After purchasing one or more components, you will receive a license key which will activate all your owned components.

When activating a license key through Linking Lab it will be linked to the system that you are connected to.
This denotes that, when you connect to a remote system and enter a license key, that it will be activated on that remote system and not your local system.
If you connect to a remote system that has licenses for components that your local system does not have, then you can still use them.

4.2 Activating License Keys

To gain access to components for which you have a license, you will first have to enter your license key.
A window similar to Figure 18 will appear in the main menu, after connecting to a system that does not have a valid license.

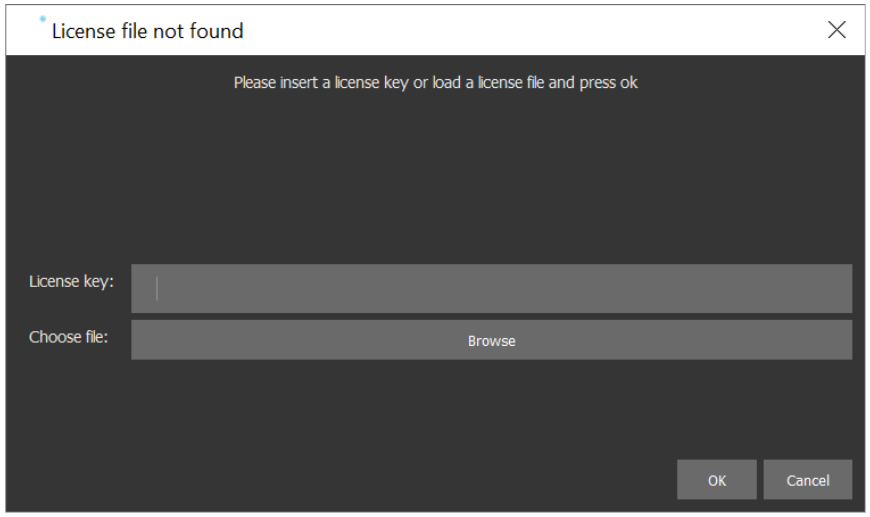


Figure 18: License key activation window

Enter your license key in the top input field and press *OK* to proceed.*

The 'Choose file' input field is for entering a JSON file which contains the licenses for the components. Useful when there is no internet access, or when access to the license server is unavailable.

**Note: When entering a 'License key' an internet connection is required.*

4.3 Renewing License Keys on a Windows System

When a license expires or when you gain access to new components, you might need to renew your license key.
To renew or update the license key, the old license file needs to be removed first.

The file that should be deleted can be found in the following path:
*. . . \AppData\Roaming\Horus View and Explore**

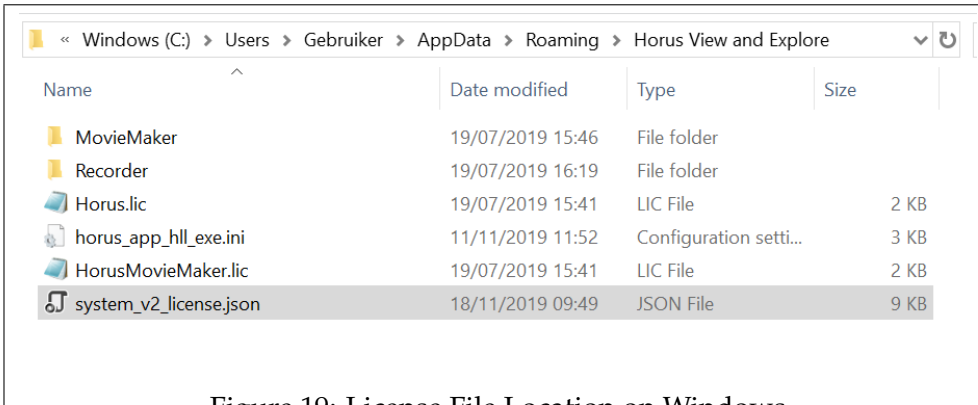


Figure 19: License File Location on Windows

Then delete the: *system_v2_license.json*.
Once the old license has been deleted, restart Linking Lab and re-enter your license.

**Note: When renewing a license key from a remote system, you will have to delete the license file from that remote system.*

4.4 Renewing License Keys on a Linux System

When a license expires or when you gain access to new components, you might need to renew your license key.
To renew or update the license key, the old license file needs to be removed first.

The file that should be deleted can be found in the following directory: *~/.config/horus**

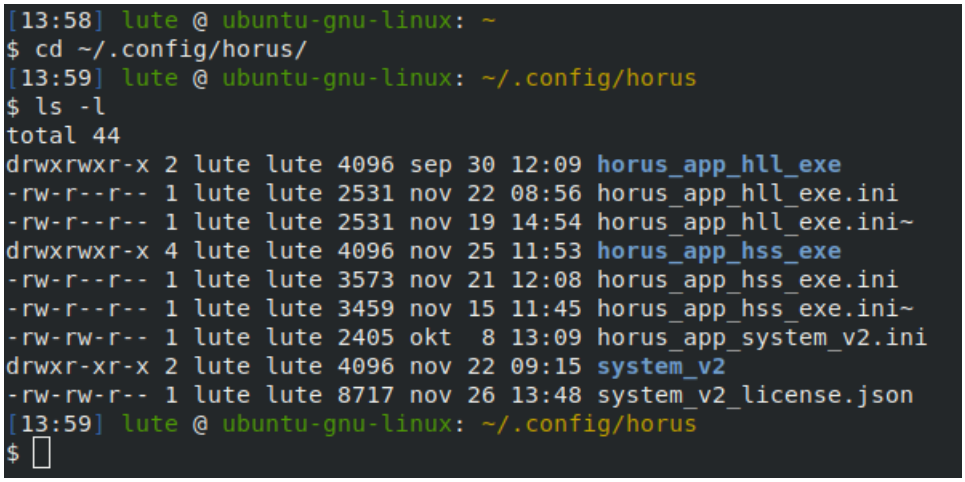


Figure 20: License File Location on Linux

Then delete the following file: *system_v2_license.json*
Once the old license has been deleted, restart Linking Lab and re-enter your license.

**Note: When renewing a license key from a remote system, you will have to delete the license file from that remote system.*

5 The Linking Lab User Interface

5.1 Introduction: User Interface

Chapter 5 describes the user interface, the different windows of Linking Lab, and their basic functions will be explained.

5.2 Connection Screen

After starting up Linking Lab you will be greeted by the connection screen. In this window, you can choose the system that you wish to connect to.

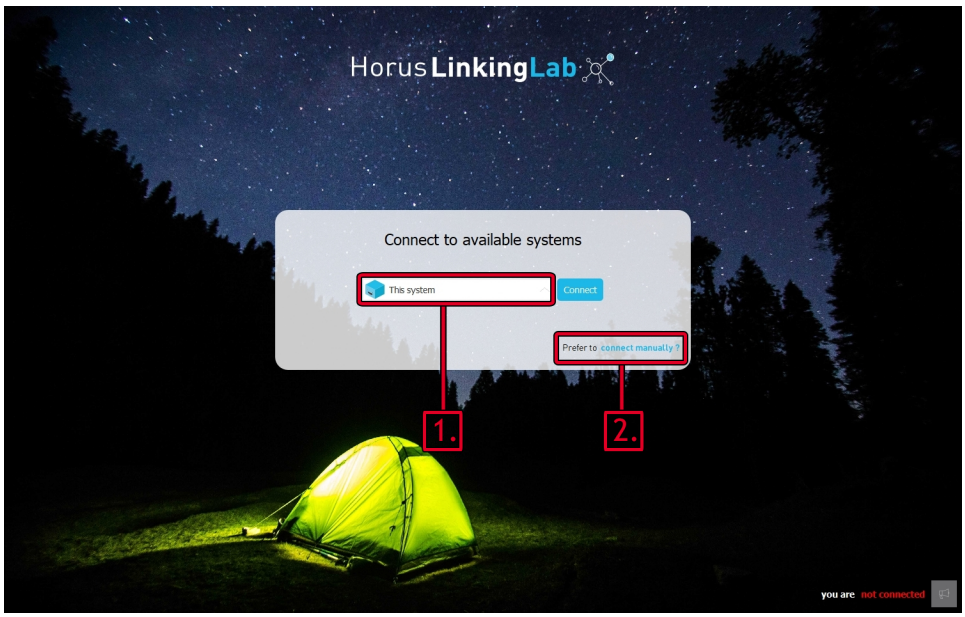


Figure 21: The connection Screen

Looking at figure 21, the box marked with '1.' displays the currently selected system to connect to, to change it, click on it and a drop down menu will appear.

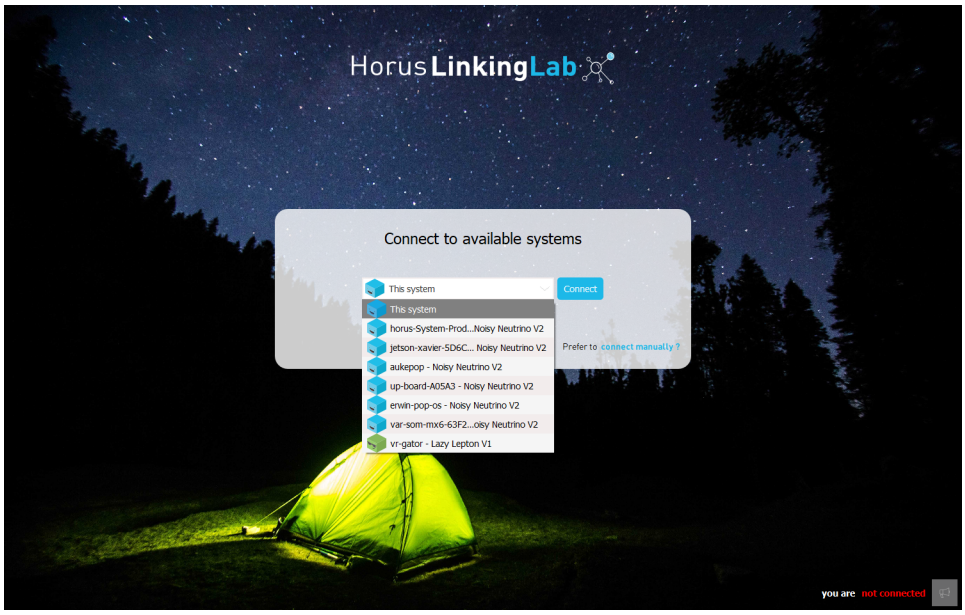


Figure 22: Drop down menu with detected Systems

This menu will display all the detected systems that are running System V1, or V2, click on the desired system, and then press 'Connect' to continue.

Connecting Manually
You have the option of connecting directly to a system, this can be used when the desired system does not get detected, or when you wish to directly connect to a system.
To connect manually click on: 'Prefer to connect manually ?' as seen on figure 21 marked by the '2.'.

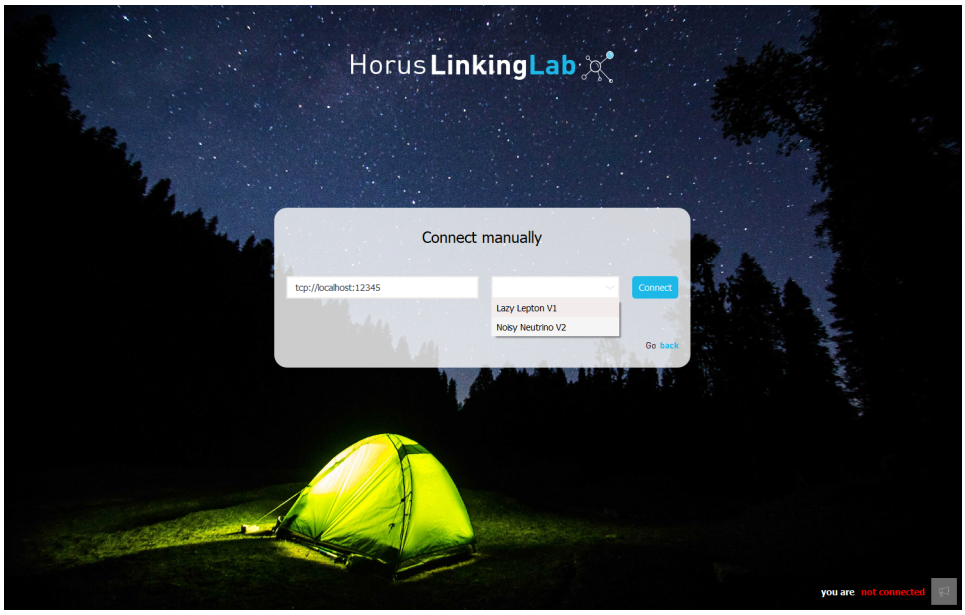


Figure 23: Connect Manually

Enter the desired hostname or IP address in the left input field, and to the right of that, select between the System V1 and V2 frameworks, depending on what that system is running.
Press 'Connect' to continue.

5.3 Main Menu

Once you have connected, you will be directed to the main menu, from here you can access all of the main features of Linking Lab.

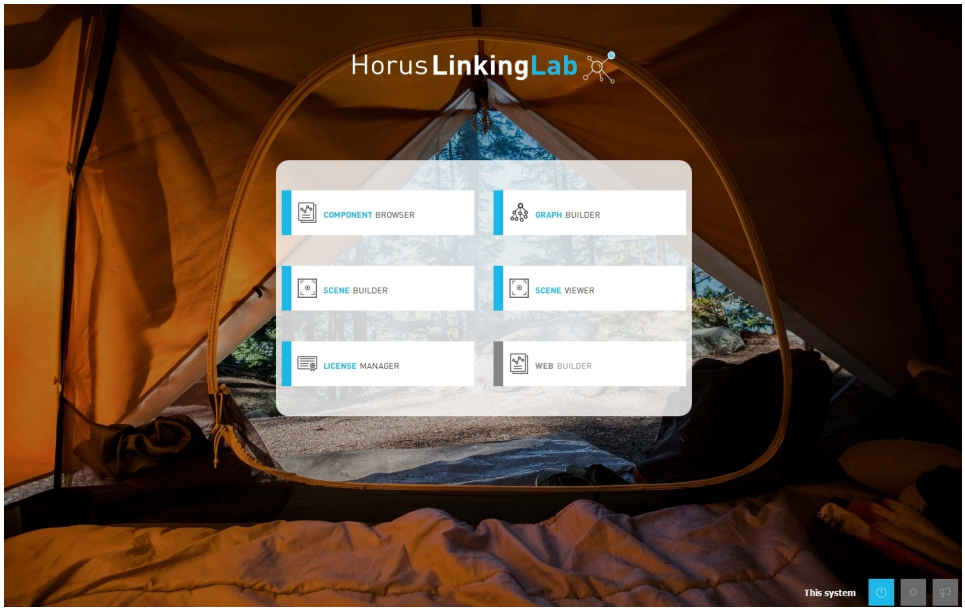


Figure 24: The Main Menu Screen

Graph Builder
The Graph Builder is where you can create a pipeline by adding and linking components together.

Component Browser
The Component Browser contains a list of all the installed components, including descriptions of their functionality and use. components may also be added manually through the Component Browser.

Scene Builder
The Scene Builder is where you can configure your 'projection_setup.json' file for use with the Horus Security Suite and the Gstreamer plugin. This tool enables you to assign the video outputs from the pipeline to various 'scenes' which can then be viewed in the Security Suite.

Scene Viewer
The Scene Viewer allows you to watch the video streams from an active pipeline.

Web Builder
Through the Web Builder you can configure a web page that allows you to control, and/or monitor the running pipeline.

License Manager
The License Manager has a list of the components and their respective licenses and expiration dates.

On the bottom right you will find a row of three buttons.

Figure 25: Disconnect Button

Disconnect Button

After pressing this button, all opened Linking Lab windows will close, and you will be disconnected, and send back to the connection screen.

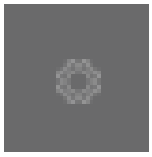
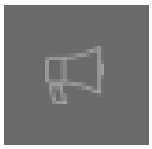


Figure 26: System Configuration Button

System Configuration Button

Opens the System Configuration window. Allows you to configure various settings.



Notification menu button

Opens the Notification window, keeps a log of debug, and other system related information.

Figure 27: Notification Button

The disconnect and notification menu buttons will appear on most windows in Linking Lab.
The System Configuration Window can only be accessed through the main menu.

5.4 Graph Builder

After selecting 'Graph Builder' in the main menu, The graph builder window will open. The Graph Builder is used to create and/or edit pipelines, and to then deploy them.

The main graph area is where you place and link components together to create pipelines, the rest of the interface can be divided into three sections as shown in figure 28.

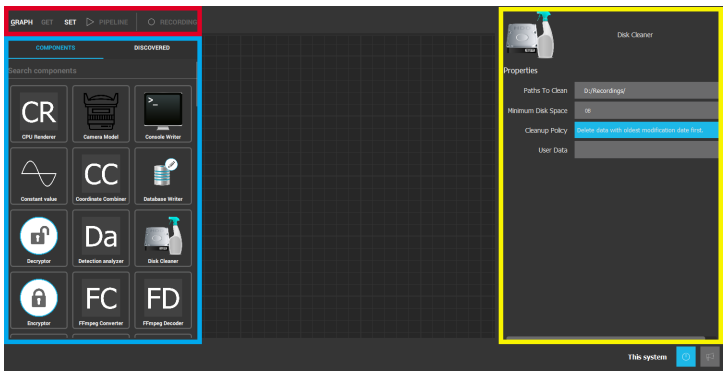


Figure 28: The Graph Builder Sections

The *blue* section contains a list of all the available components.
At the top of the section you have two tabs; one for showing the components, and the other for showing Onvif Devices that the system has discovered.
Below the two tabs, you can find a search bar; enter (part of) the name of the component to filter the list.

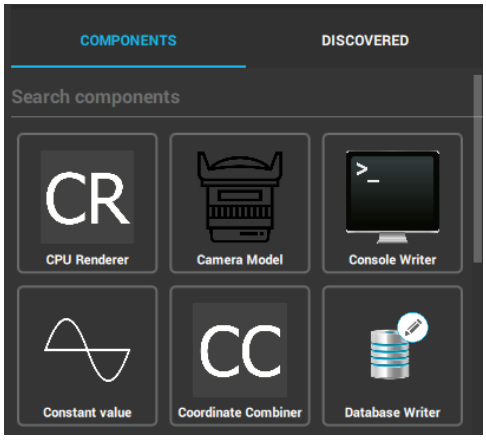


Figure 29: Component List

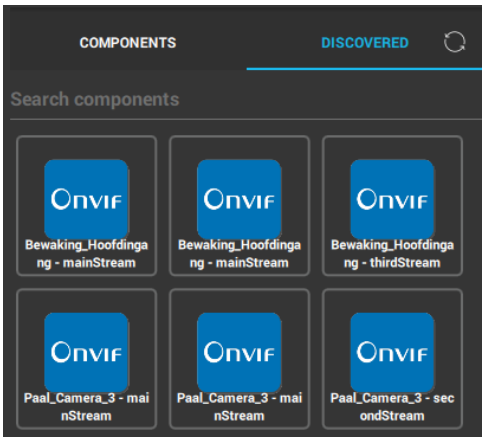


Figure 30: List of Discovered Onvif Devices

Clicking on a component once it has been placed on the graph will open a panel on the right side of the window. This is the *yellow* section and it contains the the properties of the selected component. These properties can be configured based on user/system requirements.

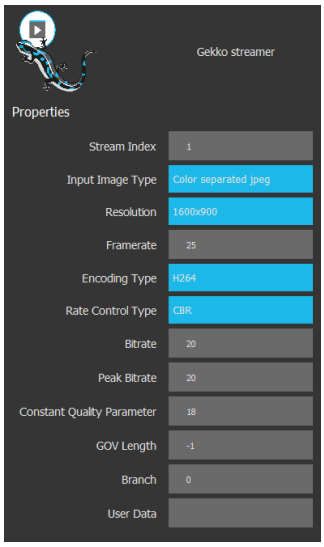


Figure 31: Component properties

The *red* section marks the top toolbar which has controls for starting the pipeline, recording, and saving/loading.

Figure 32: Graph Builder Toolbar

The '**GET**' button is used to retrieve the currently set pipeline
Usefull if you made changes, but you want to revert to the last set pipeline.

The '**SET**' button is used to set the pipeline that is currently showing on the graph as the active pipeline.

The '**PIPELINE**' button is used to start and stop running the currently set pipeline. If you are running a pipeline for the first time, or after making changes, press the 'Set' button prior to starting.

On the right, you have a button for '**RECORDING**'.
If you have a pipeline that includes components that record data, such as the File Writer, then this button can be used to start/stop the recording.

If you press the '**GRAPH**' button, a menu will appear. This menu contains several options for saving/loading pipelines, and setting the orientation.

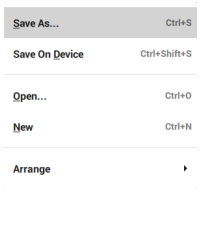


Figure 33: 'Graph' Menu

- **Save As**, Save the pipeline to a specified location as an '.hrp' file.
- **Save On Device**, This will save the pipeline directly to the connected device.

Used for saving pipelines to embedded devices. (Requires the 'v2-persistent-pipeline' setting in the '.ini' file to be enabled.)

- **Open**, Load a saved pipeline (.hrp).
- **New**, Start creating a new pipeline. This will clear graph of any components.
- **Arrange**, Change the orientation of the components.

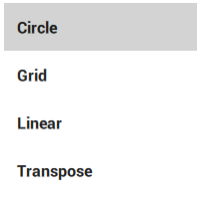


Figure 34: Orientation Options

- **Circle**, Arranges the components in a circle.
- **Grid**, Orders the components in a grid.
- **Linear**, The components will be arranged by their depth in the tree.
- **Transpose**, Toggles between arranging in a horizontal or vertical layout.

5.5 The Component Browser

The Component Browser can be accessed through the main menu by clicking 'Component Browser'.
This window contains information on how certain components or their properties function.
Besides showing information about the components, you may also manually load in additional components.

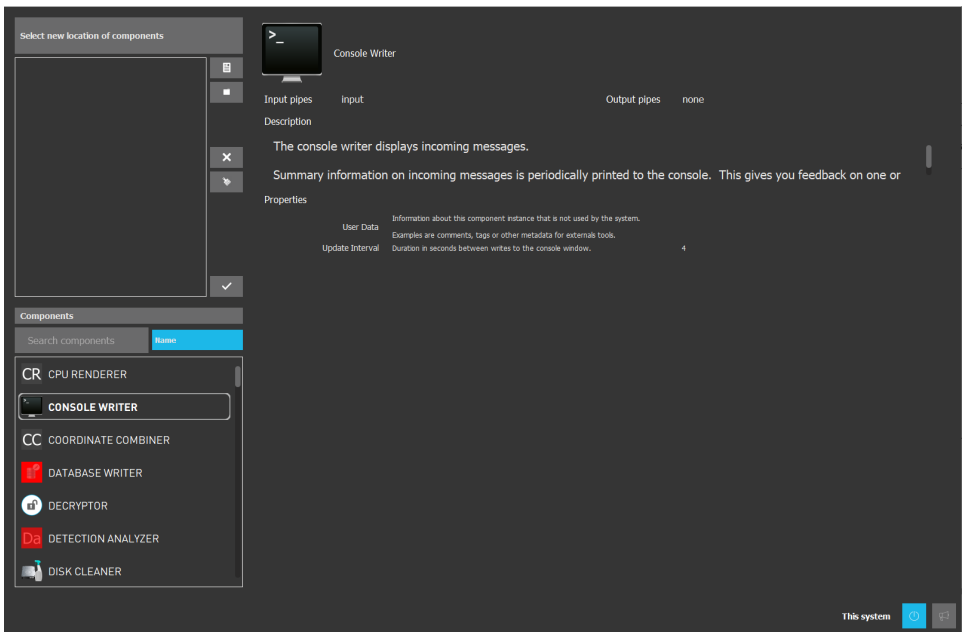


Figure 35: The Component Browser

The bottom left of the window contains a list of the installed components. Click on one to display information about that particular components. Information about the components itself, along with the properties, and in/output pipes, are displayed.

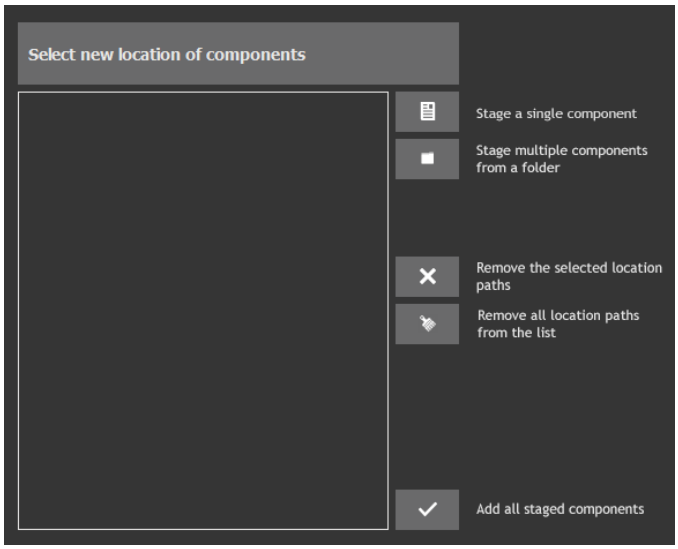


Figure 36: Adding Components

The top left section, as shown shown in figure 36 enables you to manually load in additional components.

Figure 37: An unlicensed component

Components that have a red icon, similar to the one from figure 37, are ones that are installed, but without a valid license.
When referencing the License Manager, that component should state: 'No license found'.
These components will not show up in the component list of the Graph Builder.

5.6 The Scene Builder

The Scene Builder is a tool used to configure scene setup files, these are used with the Gstreamer plugin, or our Horus Security Suite.
With the Scene Builder you can configure how the video from the pipeline is shown on the screen, this allows you to stitch, straighten, and correct images to name a few examples.

The tool supports: *'.json'*, *'.hbs'* files, and is capable of importing *'.pto'* files created by 3rd party software such as Hughin.

Menu

Through the top left of the window you can access 3 menus.
The first menu, 'Setup', contains options for opening and saving scene setup files.

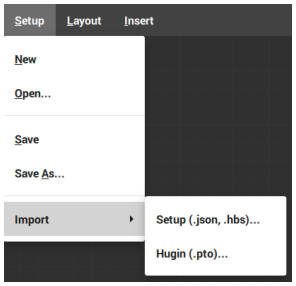


Figure 38: Scene Builder Setup Menu

Setup

- **New**
  Clears the graph.
- **Open**
  Open scene setup file (*.json*, *.hbs*)
- **Save**
  Saves the current graph.
- **Save As**
  Save the current graph to a specified name and location.

- **Import**
  Import a *.json*, *.hbs*, or *.pto* file and add it to the existing graph. Importing scene setup files will not clear the graph.

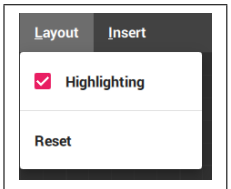


Figure 39: Scene Builder Layout Menu

The next menu contains settings for the 'Layout' of the graph.

Layout

- **Highlighting**
  If enabled, upon selecting a node, it will be highlighted along with the nodes that are connected to it.
- **Reset**
  This reorganizes the nodes.

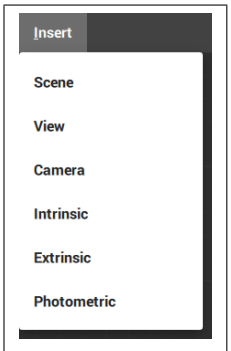


Figure 40: Scene Builder Insert Menu

The third menu is for inserting nodes onto the graph.
Click on the desired node, and it will be added to the graph.

Nodes

There are six types of nodes.

- **Scene**
  The scene consist of one or multiple 'View' nodes, and is the end-product of the scene setup.
- **View**
  The View node takes the image from a single camera.
  Extrinsic, and Intrinsic parameters can be added to the View node.
- **Camera**
  Grabs the video stream from the pipeline.
  Extrinsic, Intrinsic, and Photometric nodes can be linked to the camera node.
- **Intrinsic**
  This node allows the configuring of the intrinsic parameters of the camera. This allows you to change how the coordinates are transformed between 3D and 2D.
- **Extrinsic**
  The extrinsic parameters determine the position and rotation of the camera/view in the world.
- **Photometric**
  The parameters from the Photometric Node can adjust the light and color of the video image.

Once a node has been added to the graph, it will look similar to figure 41.

At the top of the node, you can see the given name, and type of the node. In the middle you can find some of the parameters for quick reference, and on the right side of the node, you have the different nodes that this node can connect to.

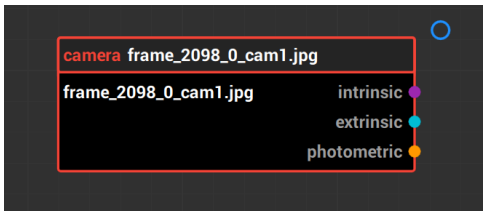


Figure 41: Camera Node

Similarly to the Graph Builder's components, you can resize the node by selecting the node, and dragging the small dot on the bottom right of the node.
The same goes for the circle at the top right, drag it to another node, and they will be connected.

To configure the node's parameters double-click on the node, to open the parameter window as seen in figure 42.

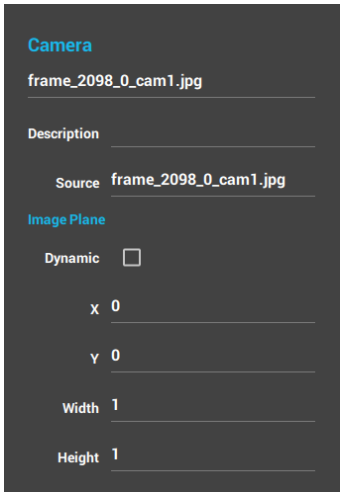


Figure 42: Camera Node Parameters

The parameters shown depends on the type of node.

Right-Click Options

Similar to the Graph Builder, the Scene Builder also has menus that appear when you right-click.

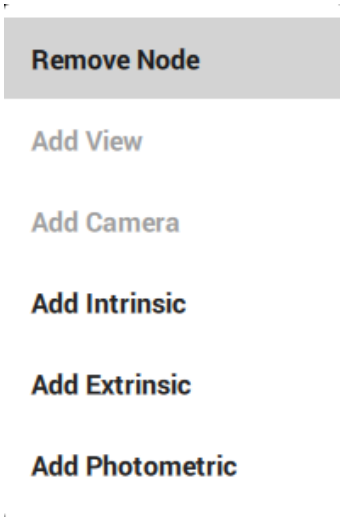


Figure 43: Node Right-Click Menu

When you right-click on a node, a menu similar to figure 43 appears. Selecting 'Remove Node' will delete the node from the graph. The rest of the options are used to quickly add nodes to the graph. Only the nodes that the node can connect to will be highlighted.

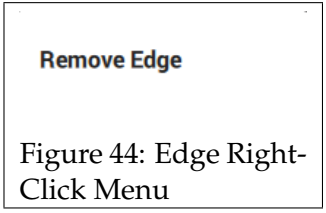


Figure 44: Edge Right-Click Menu

Once you link nodes together, a line known as an edge will appear between the two.
They can be unlinked by holding the mouse and right clicking, the menu that appears contains the option to remove the edge.

5.7 The Scene Viewer

The Scene Viewer is where you can observe video streams, and sensor data.
When you open the Scene Viewer, confirm that your pipeline is running, otherwise the viewer will not display any output data.

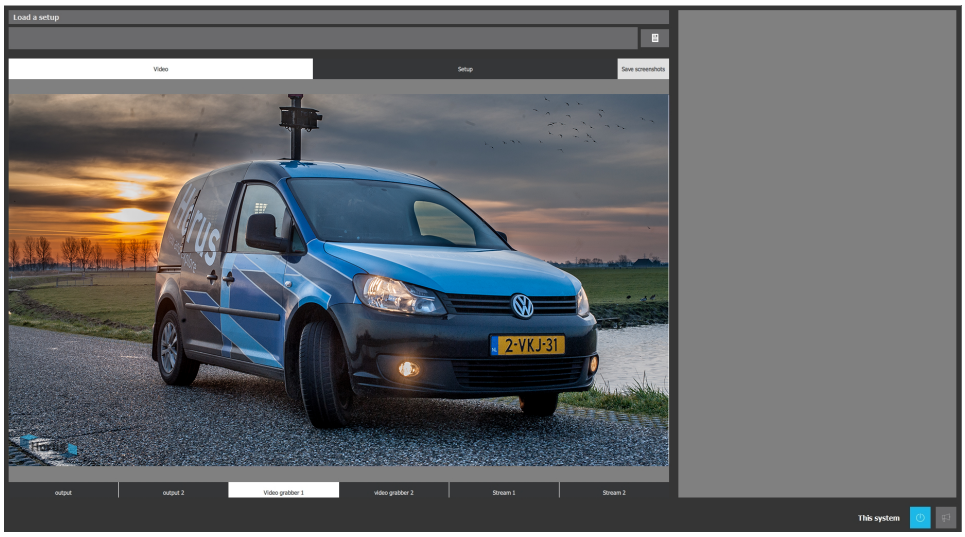


Figure 45: The Scene Viewer

The middle of the screen displays the video stream of the selected video tab.

Above the display area you have the two main tabs, '*Video*', '*Setup*'.
Selecting 'Video' will show the output video streams of the running pipeline, whilst the 'Setup' tab, shows the scenes from the setup.json file which is used with the Horus Security Suite.
When you have the 'Video' main tab selected, you will get a third option: '*Save screenshot*'

Figure 46: Main Tabs

At the bottom of the screen, you have the video tabs.
The number of tabs shown, changes, depending on which main tab is selected. The amount of video tabs is based on the number of output streams or scenes, from the pipeline or setup file respectively.

Figure 47: Video Tabs

When you click on '*Save screenshot*', a window will open, prompting you to select a folder.
Currently, the screenshots will be saved to the path shown at the top of the window, and not the selected folder.
A screenshot from every video tab will be taken, the moment you click on 'Select Folder'.

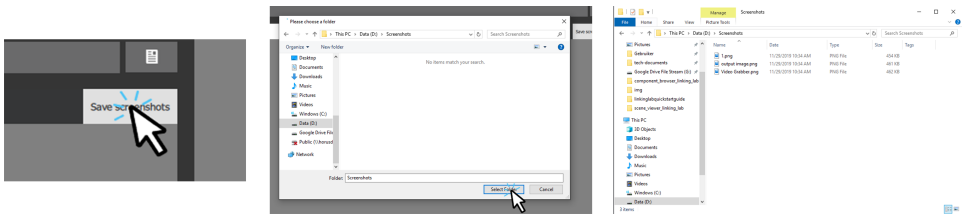


Figure 48: Taking Screenshots

The top section of the window is for loading a PTO setup file.
PTO files can be generated by third party programs such as Hugin, and contain scene configuration information.

Figure 49: Load a PTO setup file

The section on the right shows sensor data from the pipeline.
If there are multiple sensor values, you can check each value by dragging the section left and right. In most cases the information will be displayed visually in a graph or widget, see figure 50 for an example.

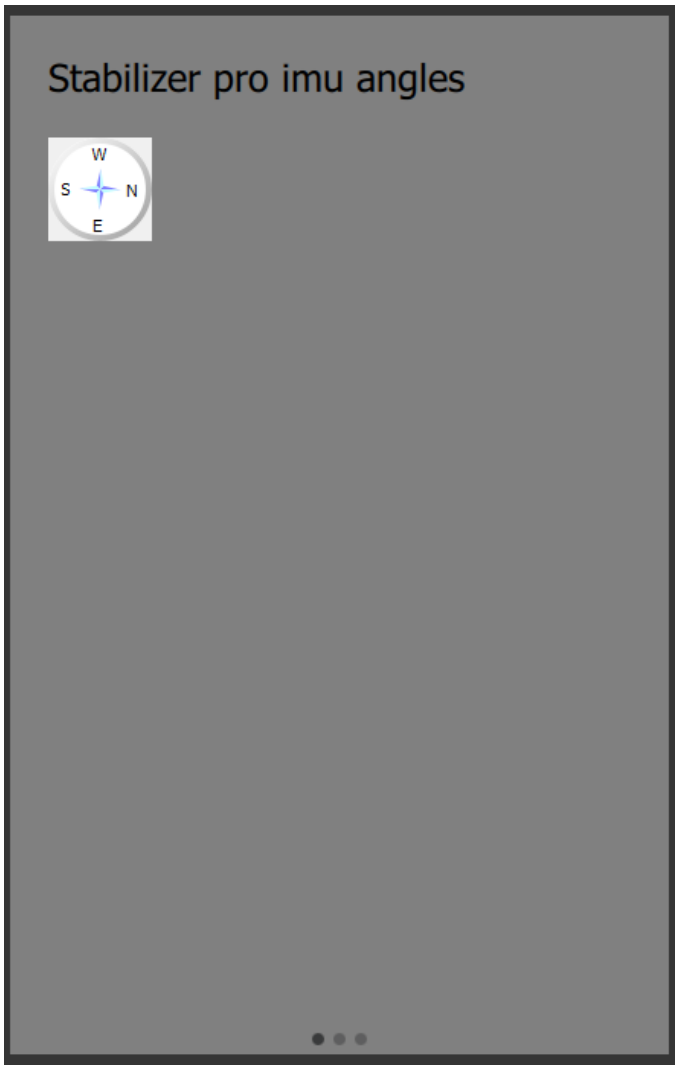


Figure 50: Sensor value from the pipeline

5.8 The License Manager

The License Manager summarizes the licenses for each component.

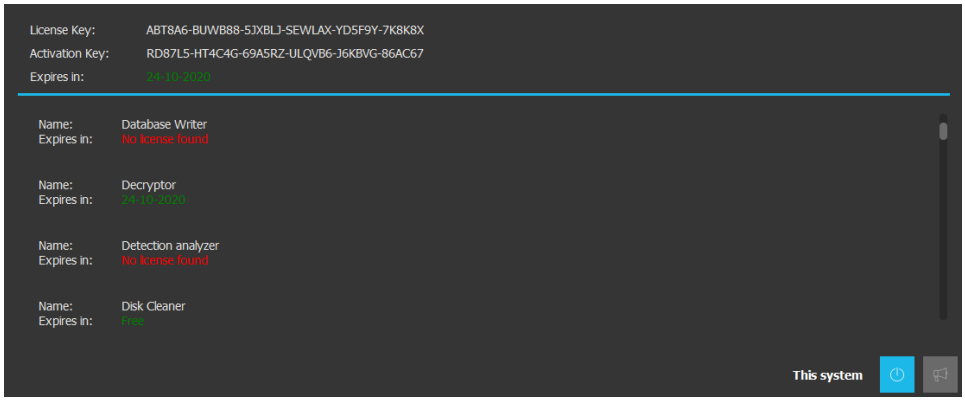


Figure 51: The Scene Viewer

By scrolling through the list you can check the licenses for each components, and in the case of a valid license, the expiration date.

At the top of the window you have the: *'License Key'*, *'Activation Key'*, and the *expiration date*.
The License Key refers to the key that you entered during the license activation. The license server checked the License Key and sent back the Activation Key. The Activation Key contains the license information for the components themselves.

The expiration date at the top, should correlate with the longest expiration date of a component. Once this date has passed the license will be deactivated.

6 Creating Pipelines

6.1 Introduction: Creating Pipelines

In the upcoming chapter, you will learn how to build pipelines.

The basic actions for creating a pipeline will be described, after which you will create a simple pipeline by following a step by step guide.

6.2 Basic Functions for Creating Pipelines

6.2.1 Basic functions of Linking Lab

In the next topic, the basic functions will be explained, this includes adding, linking,and setting the properties of components.

Adding Components
Components can be added simply by clicking on the desired component and dragging it onto the graph area.

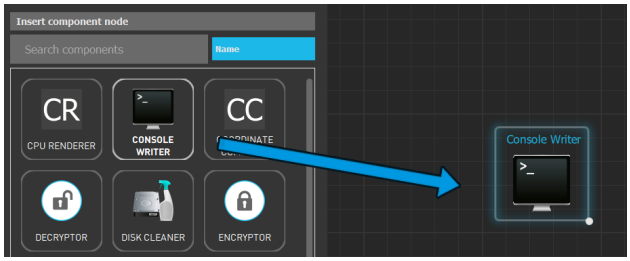


Figure 52: Adding Components

General Component controls
In figure 53 you can see a component once it has been added to the graph.

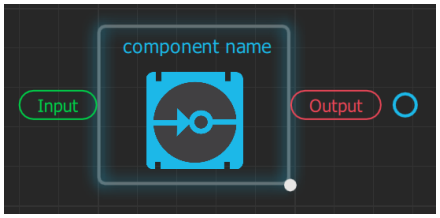


Figure 53: An added Component

The green and red cells on the sides of the component are the in- and output pipes respectively. Some components may have multiple input or output pipes.

You can rename the component by clicking on the name.
The white circle on the bottom right is used to resize the component.
By clicking on an output pipe, a blue circle will appear, this is used to link it to another component.

Linking Components
To link components together, click on the output pipe and drag the circle to the input pipe of the receiving component.

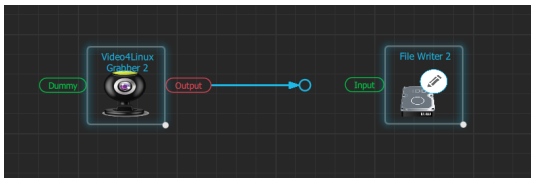


Figure 54: Linking components 1

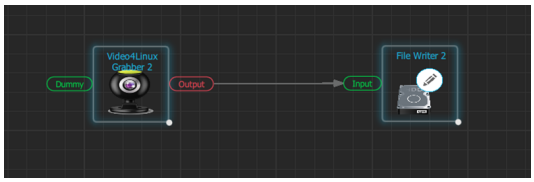


Figure 55: Linking Components 2

The line that appears between the two components is referred to as an edge, and shows the direction of the messages.

If a component does not have an output pipe you will be unable to drag an edge to another component, likewise if the component does not have an input pipe, you will be unable to link a component to it.

Setting the Properties
By clicking on a component, a list with the properties will open. These properties can be configured by the user.
It is good practice to go through the properties of each component and ensure that they are set correctly.
If you have questions about certain properties, either check the Component Browser or hover over the property to get a tool tip.

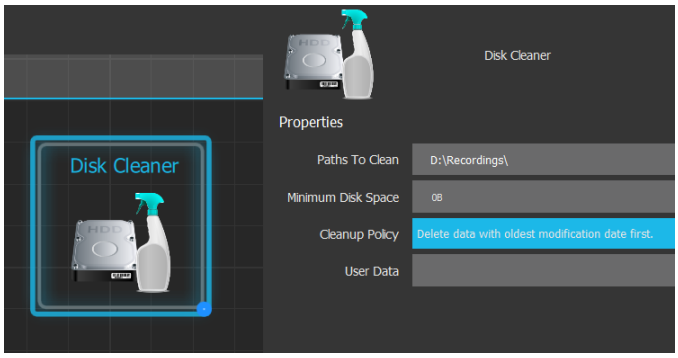


Figure 56: List of Properties

Right-click options
When you have the mouse cursor over the graph and press the *right mouse button*, you will be presented with a small menu of options.

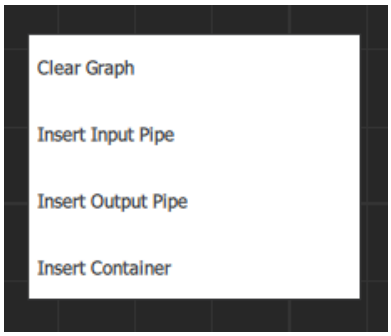


Figure 57: Right click options

Clear Graph, removes all elements from the graph.
Insert Input Pipe, add an input pipe to the graph.
Insert Output Pipe, add an output pipe to the graph.
Insert Container, Add a container to the graph.

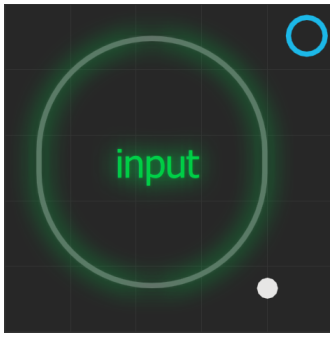


Figure 58: Input pipe

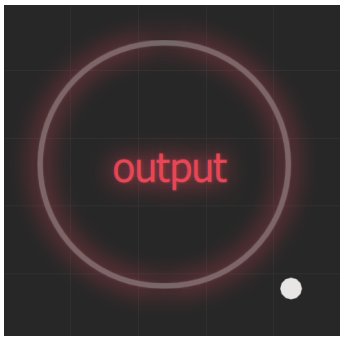


Figure 59: Output Pipe

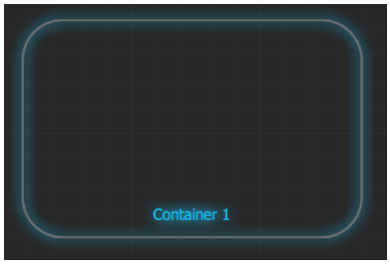


Figure 60: Container

The in/output pipes are used primarily for sending and receiving messages to and from external applications.
Containers are used to group components together; place components together with a similar context.

Right clicking on a component or an edge, will open a menu with the option of removing that component or edge, respectively.

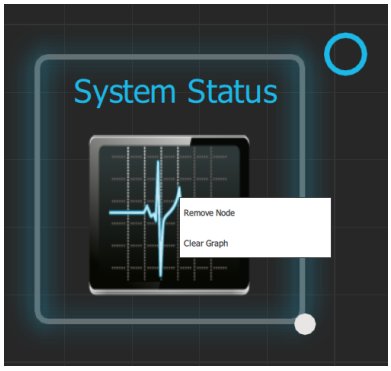


Figure 61: component right click menu

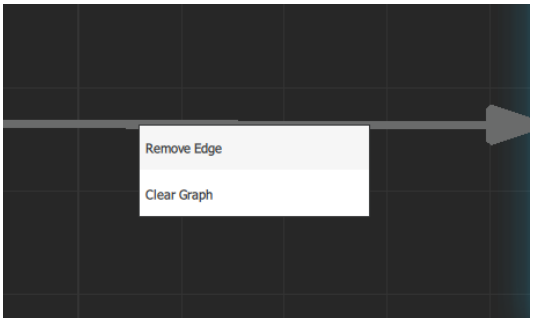


Figure 62: edge right click menu

Remove Node, Removes the selected component.*
Remove Edge, Removes the selected edge.

**Note: you can delete a component with the 'delete' key of the keyboard.*

6.3 Creating a simple Pipeline with Linking Lab

6.3.1 Creating your first Pipeline

In the coming chapter, you will learn how to build a simple pipeline.
You may follow these steps to create a functional pipeline yourself.

It is usually a good practice to have a description of what you wish to achieve with the pipeline, before building it.
For the pipeline that we are going to build it could look like the following:

"I want to write the image data from a camera (or any data grabber) to a storage disk. I want to limit the amount of data that can be recorded to a certain amount, if more data gets recorded I want the oldest data to be deleted."

Now that we know what we want to create, we can start building our pipeline.

Step 1
We can start by adding a camera grabber to the graph.
For this guide we will use the 'Quickcam Grabber' as it does not require any properties to be set.
If you do not have this component you may substitute it for any other data grabber.

Drag the 'Quickcam Grabber' and place it on the graph.

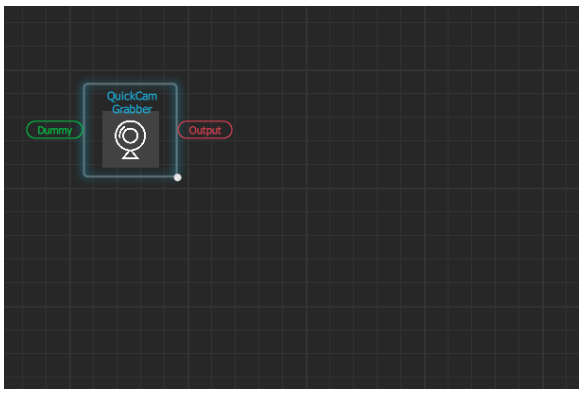


Figure 63: The Quickcam Grabber

Step 2
Now that we have our data source, we need a component that can write the image data to a disk.
For this we can use the 'File Writer', add it to the graph and link the Quickcam Grabber to the File Writer.

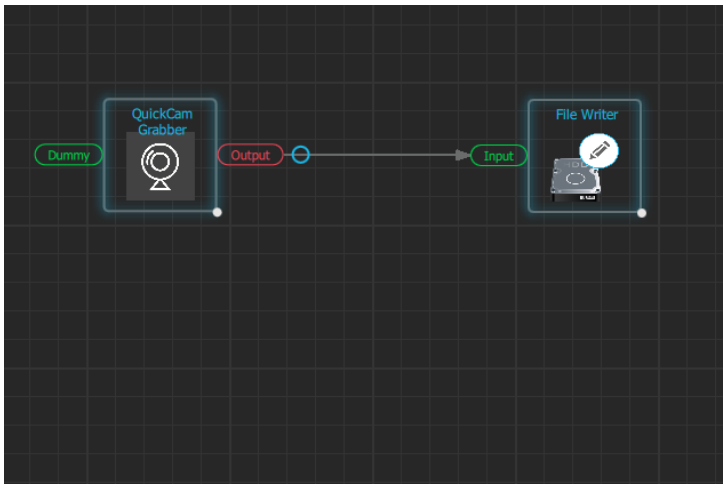


Figure 64: Quickcam linked to the file writer

Step3
Our graph should now show a data grabber that sends messages to the file writer.
To tell the File Writer what to do with the incoming messages, we need to set the properties.

See figure 65 for an example of how the properties can be set.

If you wish to see more information about these properties, check the Component Browser, or the tooltips.

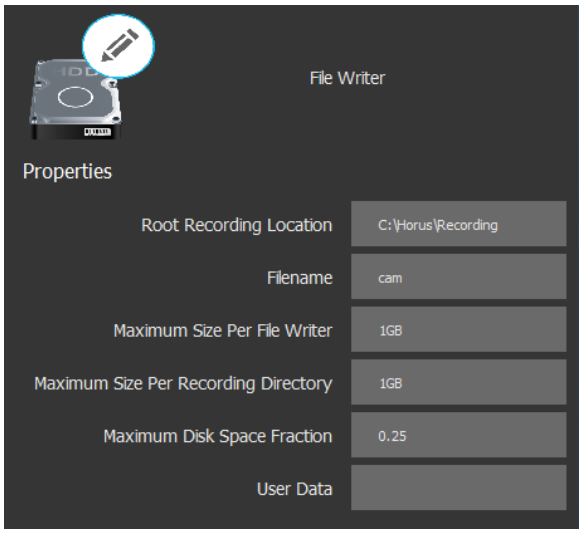


Figure 65: The File Writer Properties

Step 4

By this step we have a system that writes image data to a storage disk, but we want to limit the amount of data that is written, For this we will use the 'System Status', and 'Disk Cleaner' component. Add these components to the graph, and link them as shown in figure 66.

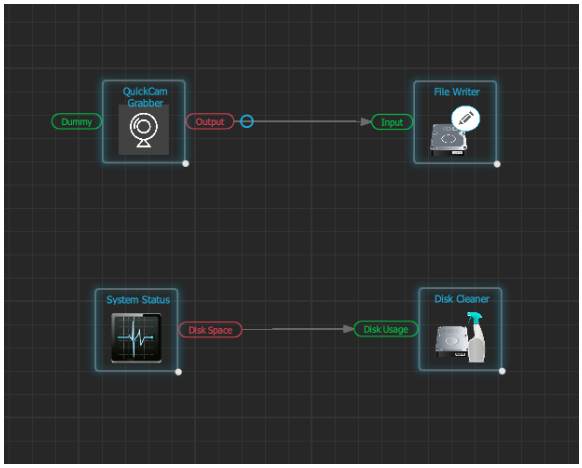


Figure 66: Added the System Status, and Disk Cleaner

The Disk Cleaner removes data from a storage disk when a threshold for the minimum amount of disk space remaining is reached.
The Disk Cleaner, however, cannot read the disk space of the system itself, to get this information it requires the System Status component.
By linking them together this component can read and then forward information about the storage disk to the Disk Cleaner.

Step 5
Set the properties for the system status and disk cleaner. See figure 67 and figure 68 for an example on how to set the properties

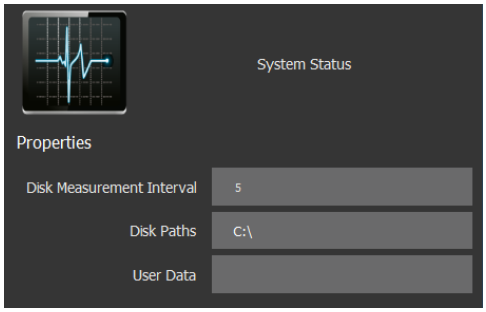


Figure 67: System Status properties

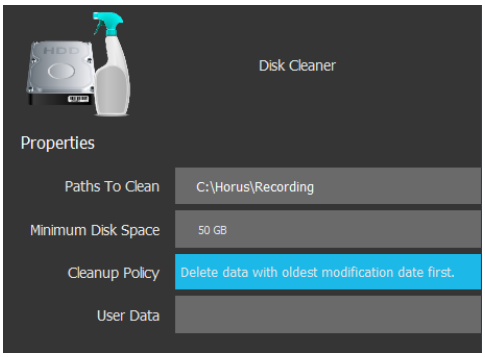


Figure 68: Disk Cleaner properties

Step 6
We now have a working pipeline that writes image data to a storage disk and deletes data when a limit is reached.
Whilst the pipeline is functional at this point, an option would be to add a 'Console Writer', this allows us to monitor information about the camera and disk space.
To implement it, add a console writer and link the camera grabber, and system status component to it.

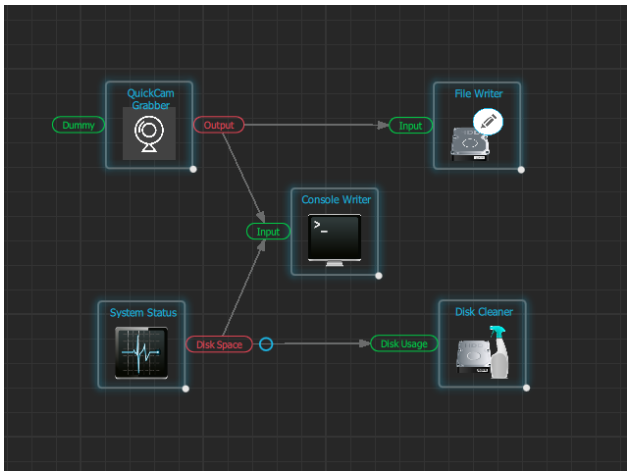


Figure 69: Added Console Writer

Step 7
The pipeline is now complete!
Set the pipeline, and then press start.

Check the notification window for any possible errors.
You should be able to see a box with the packets and MB per second on the edges, this can help you determine if a component is sending data.

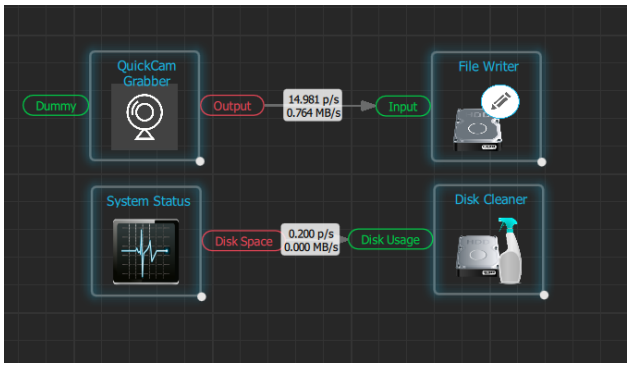


Figure 70: Data transfer rates

Now that you know how to create a pipeline, go and experiment and try building other pipelines.

7 Appendixes

7.1 Glossary and terminology

7.1.1 Glossary

In this appendix you will find a list of the terminology used in the Horus software.

Horus View & Explore
Horus is a Dutch company based in the village of Bedum, and has around 30 employees. Horus focuses on video and sensor solutions for primarily, mobile mapping, security, surveillance, and virtual reality.
Linking Lab
Linking Lab is used to create, edit, and deploy pipelines through a graphical user interface. Linking Lab is able to connect to remote systems, thus allowing users to make changes to pipelines of other devices such as embedded systems.
Security Suite
Security suite is a tool that allows users to monitor, playback, record, and analyze video footage. This application works hand in hand with the System V2 framework and the Linking Lab software to setup and run any camera setup.
System V2
System V2 is a framework which acts as a backend for other Horus software. This framework contains numerous pre-programmed components.
These components can be combined with Linking Lab to create pipelines. The Security Suite uses a System V2 backend to get video footage and other data into the application.

Horus Security Suite terms

Base_type_safe message
The Base_type_safe is a data format that commonly for the messages used within the System V2 framework. These messages can contain various types of data; video, audio, positional, sensor data, etc.
Components
Components are the pre-made building blocks that are part of the System V2 framework.

each component has a specific function, some may grab data, others transform data, whilst some may write or stream data. Each component has a number of properties that can be configured by the user.
For a list of the components along with descriptions, see the Component Browser.

Containers
Containers are used to group components together that have similar a context.

Edge
The edges are the connecting lines between components over which data known as messages are sent. The edges also show the direction of the data flow and the number of packets, and MB per second.

In/Output Pipes
A component receives messages through the Input Pipe and sends messages through the Output Pipe.
Users can also add separate In/Output via the right-click menu, this allows the system to send or receive messages, to and from external programs.

Messages
Messages refer to the data that is transferred between components.

Pipeline
Pipelines consist of multiple components and refers to the combination of linked components displayed on the graph of the Graph Builder.

Component Browser
The Component Browser contains a list of the installed components along with descriptions of the component and its properties. You can also load additional components manually.

Graph Builder
The Graph Builder is where you build, edit, and save pipelines. You may also start the pipelines through the Graph Builder.

License Manager
The License Manager lists the license status for each component.

Scene Viewer
In the Scene Viewer, you can see output data in the form of video streams, and sensor data. You may also preview the scenes for the Horus Security Suite.

Notification Window
The Notification Window logs debug, and other system related events.

System Configuration Menu
In this window, you can access and edit various configuration files.

Horus Security Suite terms

Scenes
Scenes can be configured, they allow you to create different viewports and alter the incoming footage to create panoramic or virtual cameras.
Windows
These are the separate windows that form the foundation of the Security Suite's user interface. These windows may display video footage, positional tracking, or they might contain functions for controlling the application, or other devices.
layout
The layouts refer to the arrangement of different windows. Security Suite can save these arrangements in up to 2 Layout presets. The user can switch between these presets at will.
Events
Events can be created in the 'Report' window. Events can be created to group the taken images.
Reports
Reports can be send or printed out via the 'Report' window. This report includes data from the images that were selected.