

HORUS VIEW & EXPLORE

TECHNICAL DOCUMENTATION

Horus Handbook

Horus View & Explore
info@horus.nu



August 4, 2020

Contents

1	Introduction	5
2	Getting Started with Linking Lab	6
2.1	The Horus Linking Lab Application	6
2.1.1	What is linking lab?	6
2.2	Linking Lab Components	7
2.2.1	What are Components?	7
2.3	Installing Linking Lab	9
2.3.1	Introduction: Installation of Linking Lab	9
2.3.2	The Installation Process on Windows	9
2.3.3	The Installation Process on Linux	14
2.4	License Keys	18
2.4.1	Introduction: License Keys	18
2.4.2	Activating License Keys	18
2.4.3	Renewing License Keys on a Windows System	20
2.4.4	Renewing License Keys on a Linux System	21
2.5	The Linking Lab User Interface	22
2.5.1	Introduction: User Interface	22
2.5.2	Connection Screen	22
2.5.3	Main Menu	25
2.5.4	Graph Builder	28
2.5.5	The Component Browser	32
2.5.6	The Scene Builder	34
2.5.7	The Scene Viewer	39
2.5.8	The License Manager	42
3	Creating a simple Pipeline	43
3.1	Introduction: Creating Pipelines	43
3.2	Basic Functions for Creating Pipelines	43
3.2.1	Basic functions of Linking Lab	43
3.3	Creating a simple Pipeline with Linking Lab	47
3.3.1	Creating your first Pipeline	47
4	Horus Security Suite	52
4.1	What is Security Suite	52
4.1.1	The Security Suite Software	52
4.2	Installing Security Suite	54
4.2.1	Introduction: Installation of Security Suite	54

4.2.2	The Installation Process on Windows	54
4.3	Configuring Security Suite	58
4.3.1	Introduction: Configuring Security Suite	58
4.3.2	Configuration process	58
4.4	The User Interface of Security Suite	63
4.4.1	Introduction: User Interface	63
4.4.2	Menu Window	64
4.5	The Windows of Security Suite	71
4.5.1	Introduction: Security Suite Windows	71
4.5.2	Browser Window	71
4.5.3	Status Window	72
4.5.4	Settings Window	73
4.5.5	Report Window	76
4.5.6	Details Window	80
4.5.7	PTZ Control Window	83
5	CityMapper Manual	86
5.1	Introduction	86
5.2	The Citymapper System	87
5.2.1	What is the Citymapper?	87
5.2.2	Citymapper Kit Contents	87
5.2.3	Included Software	89
5.2.4	System Requirements	90
5.3	Setting up the System	91
5.3.1	Assembling in the System	91
5.4	Initial Setup of the System	93
5.4.1	Configuring the Software	93
5.5	Recording with the Citymapper	105
5.5.1	How to Record	105
5.6	Processing	109
5.6.1	Introduction: Processing	109
5.6.2	Processing: Importing Data	109
5.6.3	Processing: POSPac	111
5.6.4	Processing: Position Fixer	119
5.6.5	Processing: Movie Player	120
6	ASCII PTZ Protocol	123
6.1	Horus View & Explore ASCII PTZ Protocol	123
6.1.1	Introduction: ASCII PTZ Protocol	123
6.1.2	In practice: connectivity	123

6.1.3	Network connectivity	123
6.1.4	Serial connectivity	124
6.2	The protocol parser.	124
6.3	The protocol definition.	125
7	ASCII JSON Sensor Protocol	127
7.1	Horus View & Explore ASCII Sensor Protocol	127
7.1.1	Introduction: ASCII Sensor Protocol	127
7.1.2	In practice: connectivity	127
7.1.3	Network connectivity	127
7.1.4	The protocol parser.	128
7.1.5	The protocol definition.	128
8	Converting GoPro Recordings to the Horus Format	130
8.1	Introduction	130
8.2	Converting GoPro Recordings for use in the Horus Environ- ment	131
8.2.1	Converting footage for Horus software	131
9	Horus Webservices	140
9.1	Horus Webservices: System Architecture	140
9.1.1	System Architecture	140
9.2	Horus Webservices: Installation	142
9.2.1	Introduction: Installation	142
9.2.2	Installing the Media Server	142
10	Appendixes	152
10.1	Glossary and terminology	152
10.1.1	Glossary	152
10.2	Flashing an Image onto an SD Card	154
10.2.1	How to Flash an Image onto an SD Card	154
10.3	HLL & HSS .ini File	156
10.3.1	introduction: HLL & HSS .ini File	156
10.3.2	.Ini File Options	157
10.4	report_settings.json	165
10.4.1	introduction: report_settings.json	165
10.4.2	Configuring: report_settings.json	165
10.5	SSH Connection with Embedded Devices	168
10.5.1	SSH Connection with Raspberry Pi	168
10.6	Citymapper Spec Sheet	170

10.7 Citymapper Hardware Diagram 171

1 Introduction

In front of you is the Horus Handbook, a compilation of all the documentation available for Horus software.

The handbook contains information on the following topics:

- Linking Lab
- Creating a pipeline in Linking Lab
- Security Suite
- ASCII PTZ Protocol
- Converting GoPro recordings to Horus format
- Horus Web Services
- Appendixes
 - Glossary
 - Flashing Horus Software on SD card
 - Editing .ini Files
 - Editing report_.json file
 - SSH connection with embedded devices

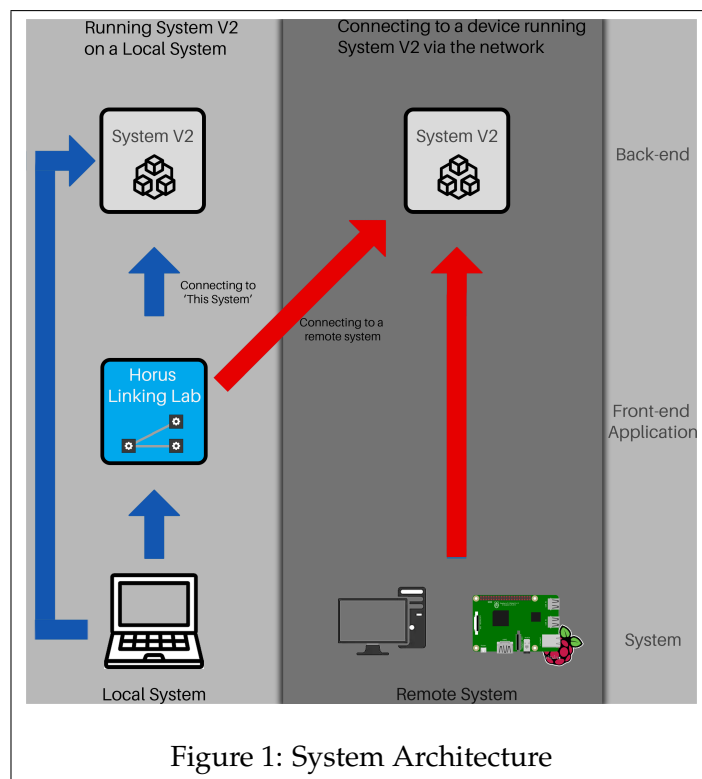
2 Getting Started with Linking Lab

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.



In figure 1 you can see how the Horus software is set up.

The blue arrows show 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 shown 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.

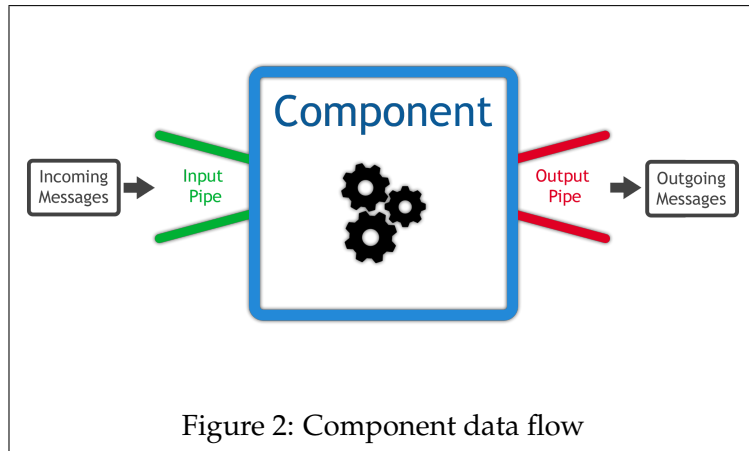
The benefits of this platform are that it allows you to quickly design and deploy your created software solutions, using an accessible visual user interface, that does not require programming or deep technical knowledge, this results in quicker development times, and lower costs.

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.



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.

2.3 Installing Linking Lab

2.3.1 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.

2.3.2 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

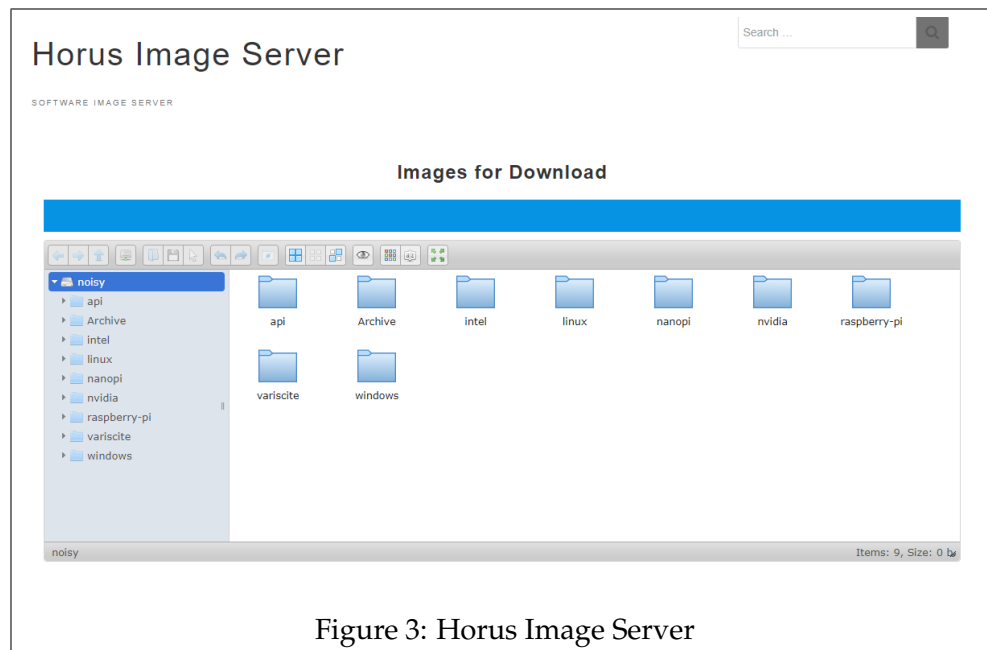


Figure 3: Horus Image Server

A page like the one above should appear.

To find the installer, click on *Windows->x64->LinkingLab*

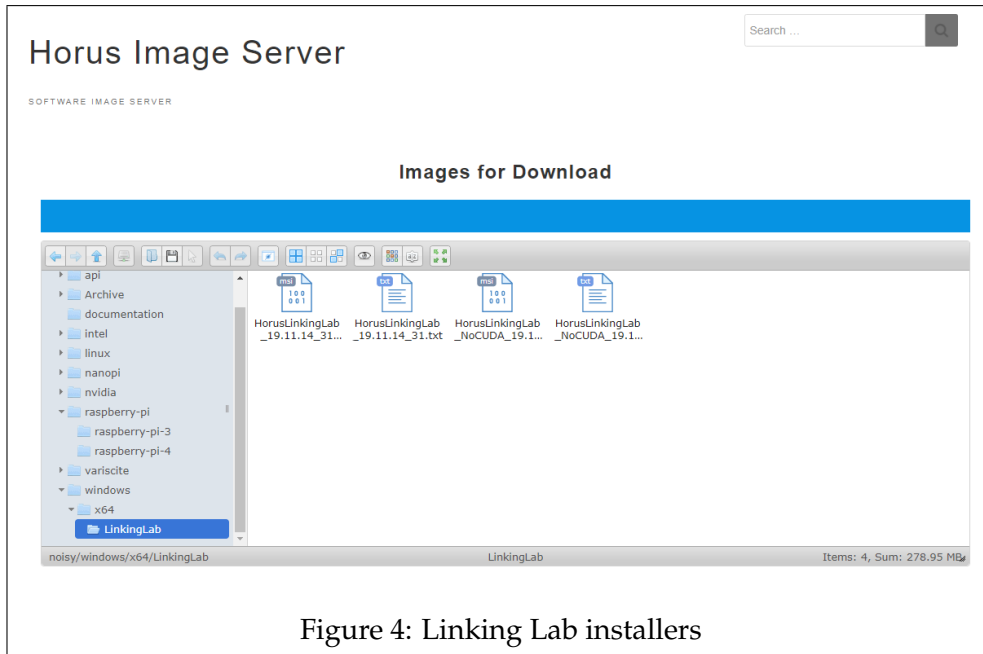


Figure 4: Linking Lab installers

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

Step 2.

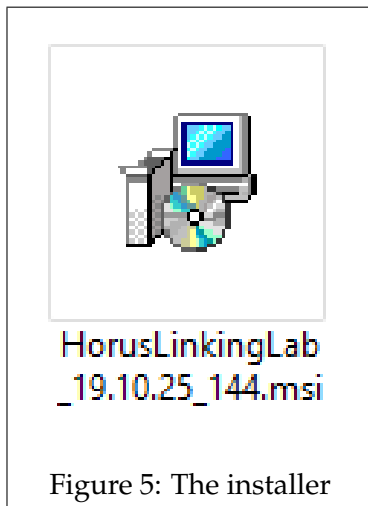


Figure 5: The installer

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

Step 3.

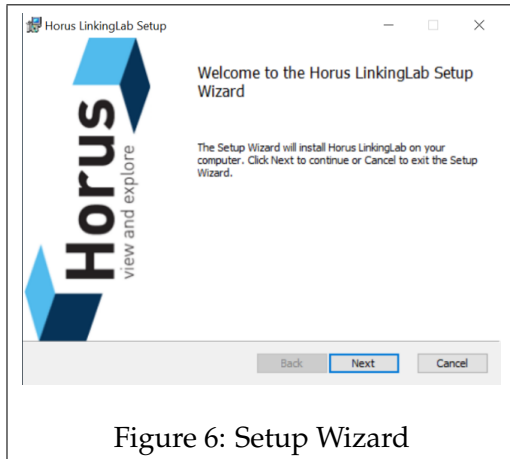


Figure 6: Setup Wizard

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

Step 4.

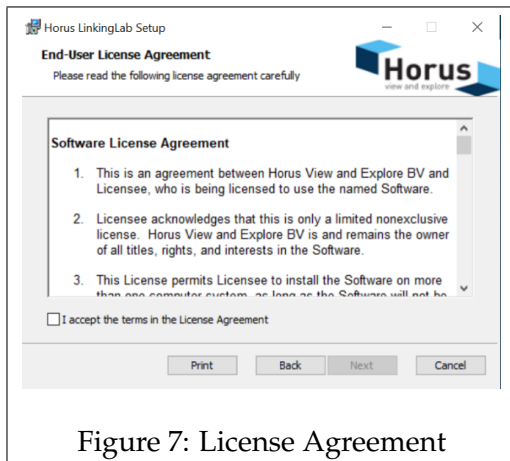


Figure 7: License Agreement

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.

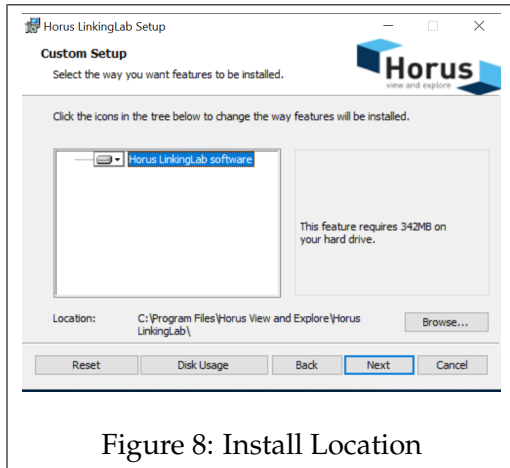


Figure 8: Install Location

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.

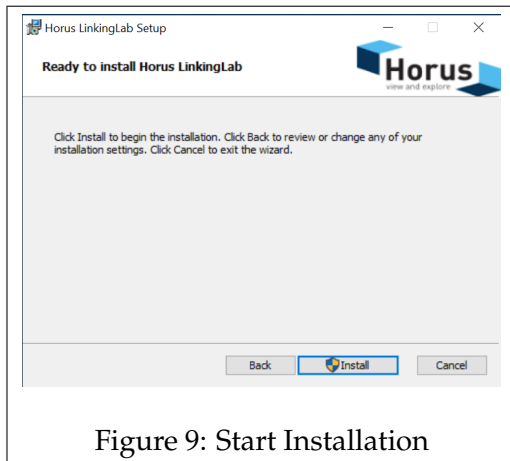


Figure 9: Start Installation

The setup wizard requires final approval before it can install Horus Linking Lab.

Click *Install* to start the installation.

Step 7.

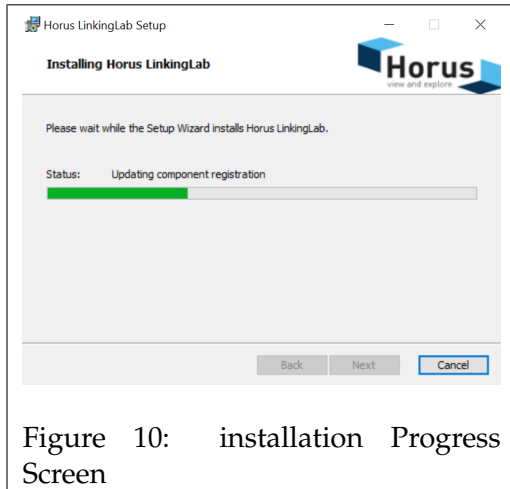


Figure 10: installation Progress Screen

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.

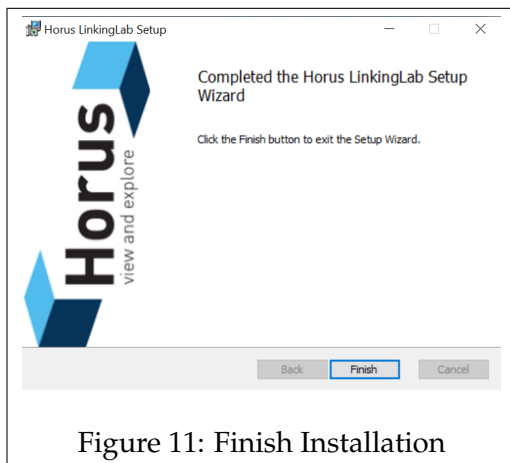


Figure 11: Finish Installation

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

Horus Linking Lab is now successfully installed on your system.

2.3.3 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*

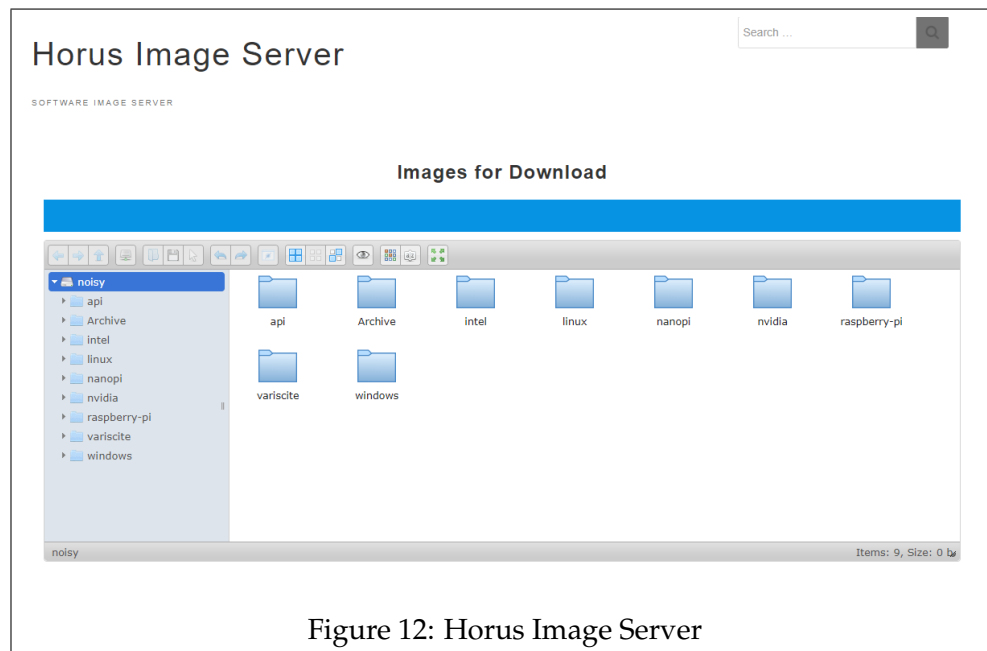


Figure 12: Horus Image Server

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

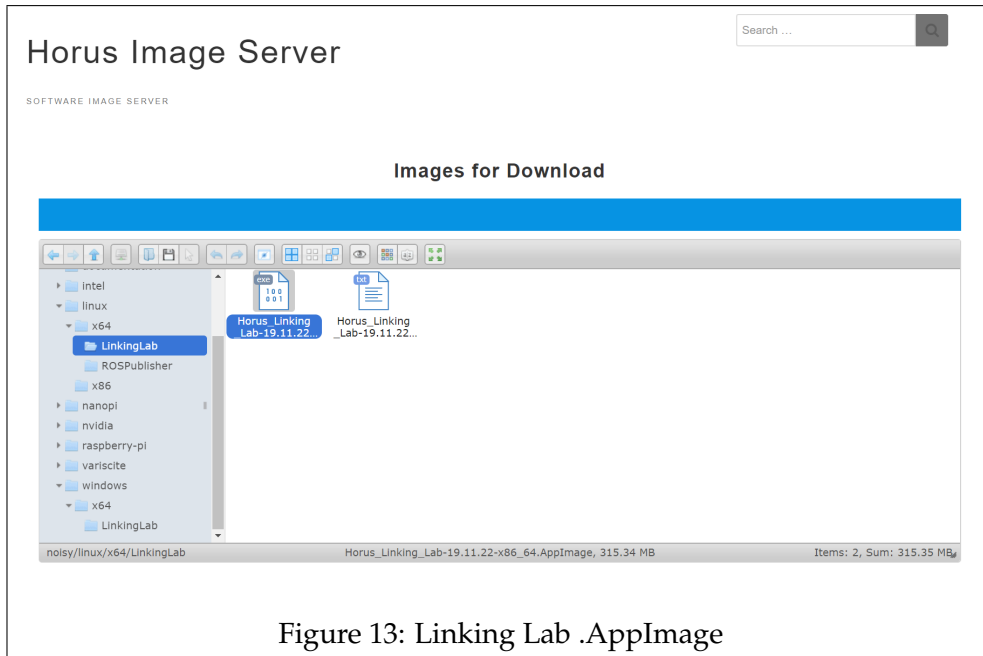


Figure 13: Linking Lab .AppImage

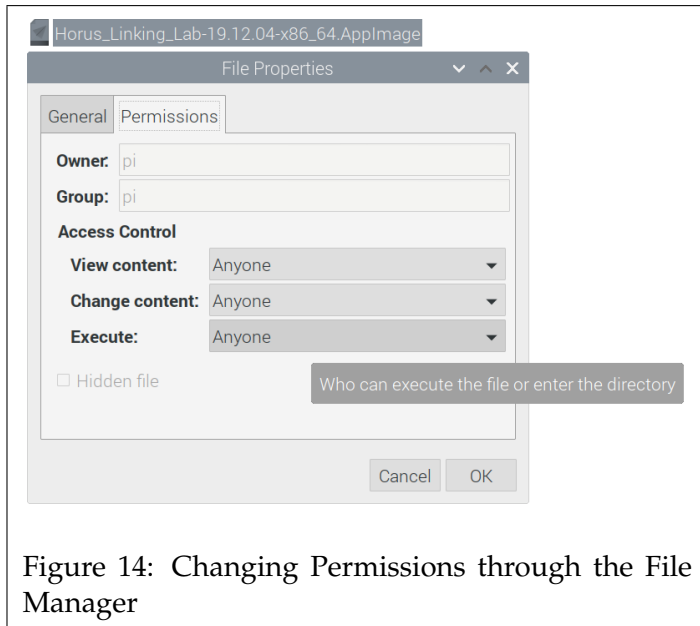
Download the .AppImage and proceed to the next step.

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.



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.

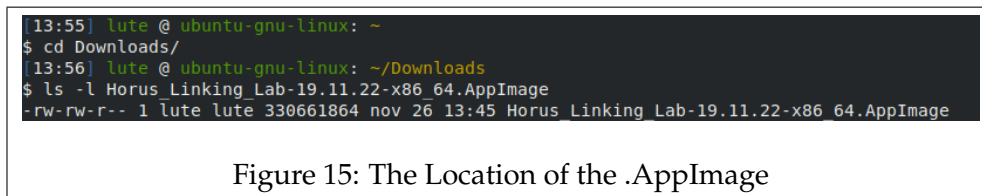


Figure 15: The Location of the .AppImage

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.

```
13:56] lute @ ubuntu-gnu-linux: ~/Downloads
$ chmod +x Horus_Linking_Lab-19.11.22-x86_64.AppImage
```

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.

```
13:56] lute @ ubuntu-gnu-linux: ~/Downloads
$ ls -l Horus_Linking_Lab-19.11.22-x86_64.AppImage
-rwxrwxr-x 1 lute lute 330661864 nov 26 13:45 Horus_Linking_Lab-19.11.22-x86_64.AppImage
13:56] lute @ ubuntu-gnu-linux: ~/Downloads
$
```

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.

2.4 License Keys

2.4.1 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.

2.4.2 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.

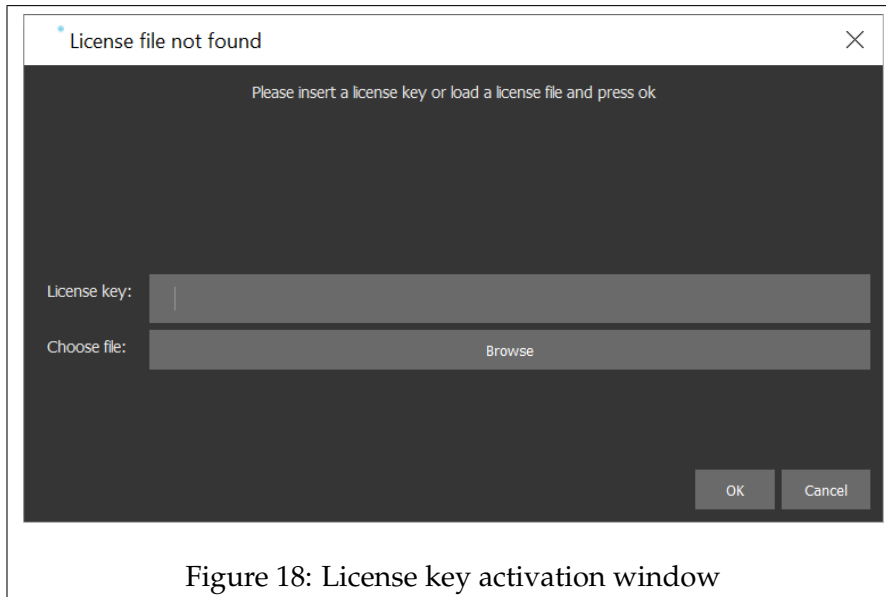


Figure 18: License key activation window

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.*

2.4.3 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**

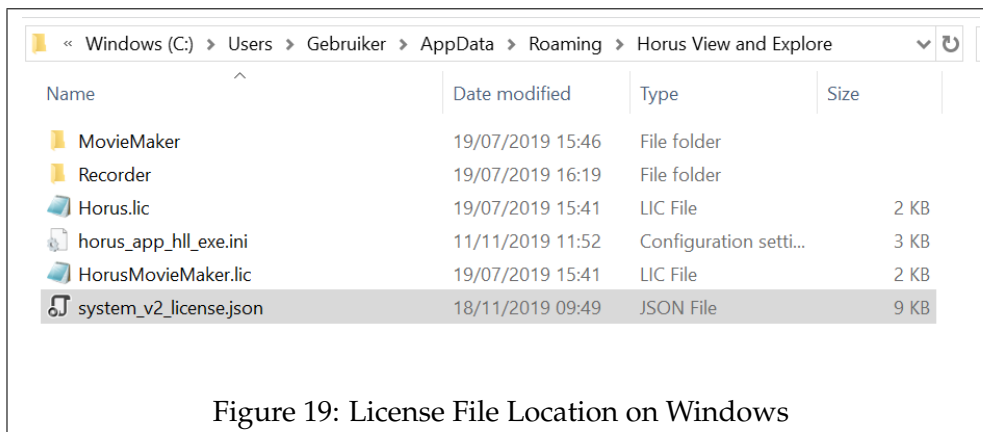


Figure 19: License File Location on Windows

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.*

2.4.4 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*`

```
[13:58] lute @ ubuntu-gnu-linux: ~
$ cd ~/.config/horus/
[13:59] lute @ ubuntu-gnu-linux: ~/.config/horus
$ ls -l
total 44
drwxrwxr-x 2 lute lute 4096 sep 30 12:09 horus_app_hll_exe
-rw-r--r-- 1 lute lute 2531 nov 22 08:56 horus_app_hll_exe.ini
-rw-r--r-- 1 lute lute 2531 nov 19 14:54 horus_app_hll_exe.ini~
drwxrwxr-x 4 lute lute 4096 nov 25 11:53 horus_app_hss_exe
-rw-r--r-- 1 lute lute 3573 nov 21 12:08 horus_app_hss_exe.ini
-rw-r--r-- 1 lute lute 3459 nov 15 11:45 horus_app_hss_exe.ini~
-rw-rw-r-- 1 lute lute 2405 okt  8 13:09 horus_app_system_v2.ini
drwxr-xr-x 2 lute lute 4096 nov 22 09:15 system_v2
-rw-rw-r-- 1 lute lute 8717 nov 26 13:48 system_v2_license.json
[13:59] lute @ ubuntu-gnu-linux: ~/.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.*

2.5 The Linking Lab User Interface

2.5.1 Introduction: User Interface

Chapter 4.5 describes the user interface, the different windows of Linking Lab, and their basic functions will be explained.

2.5.2 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.

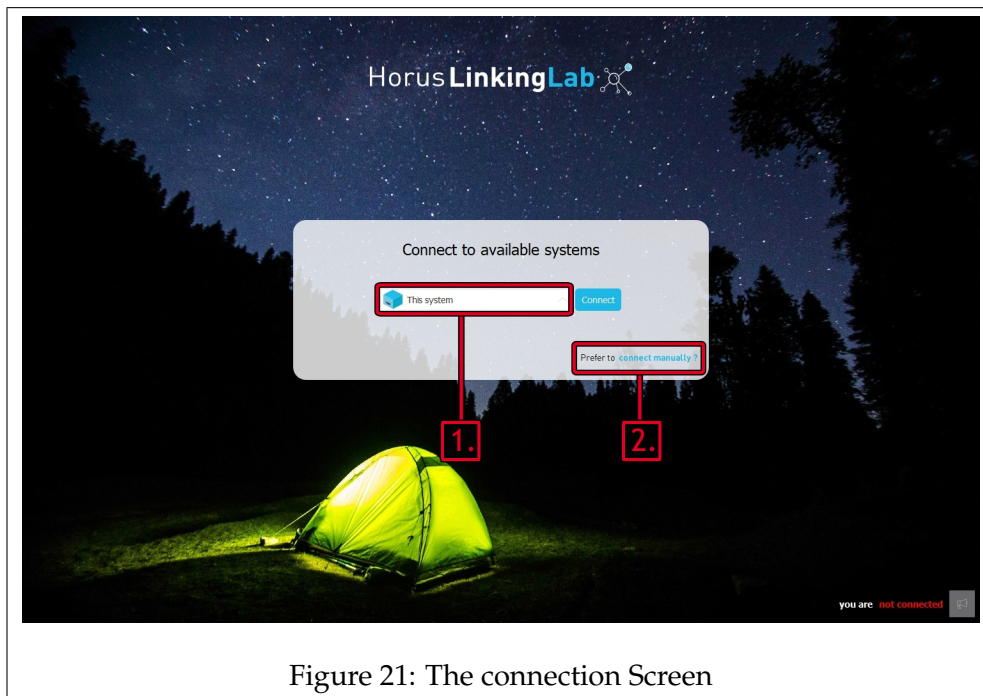


Figure 21: The connection Screen

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.

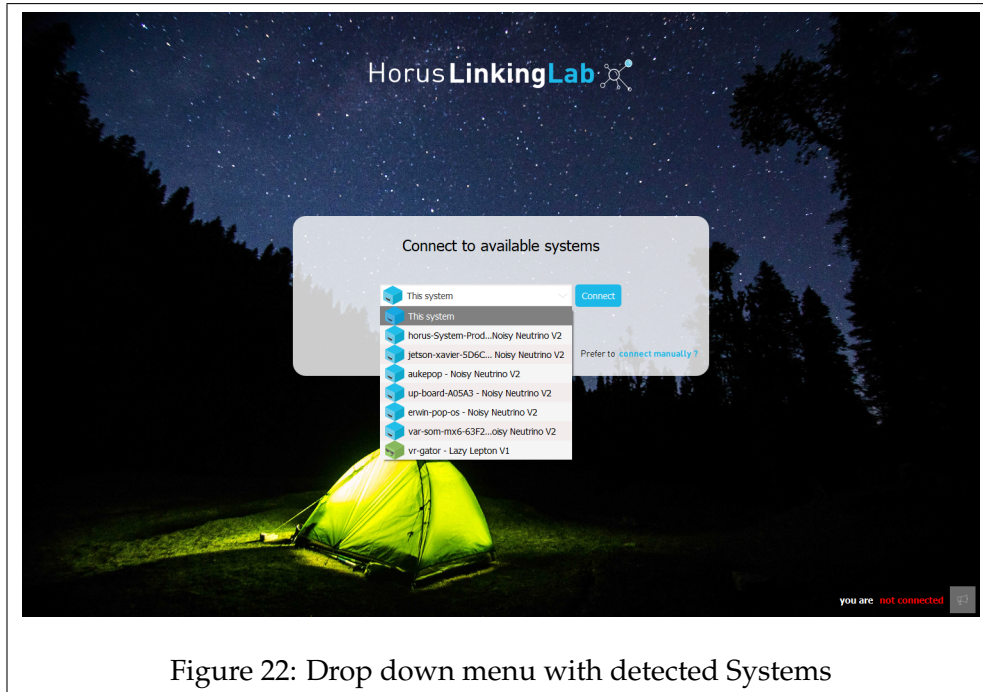


Figure 22: Drop down menu with detected Systems

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'.

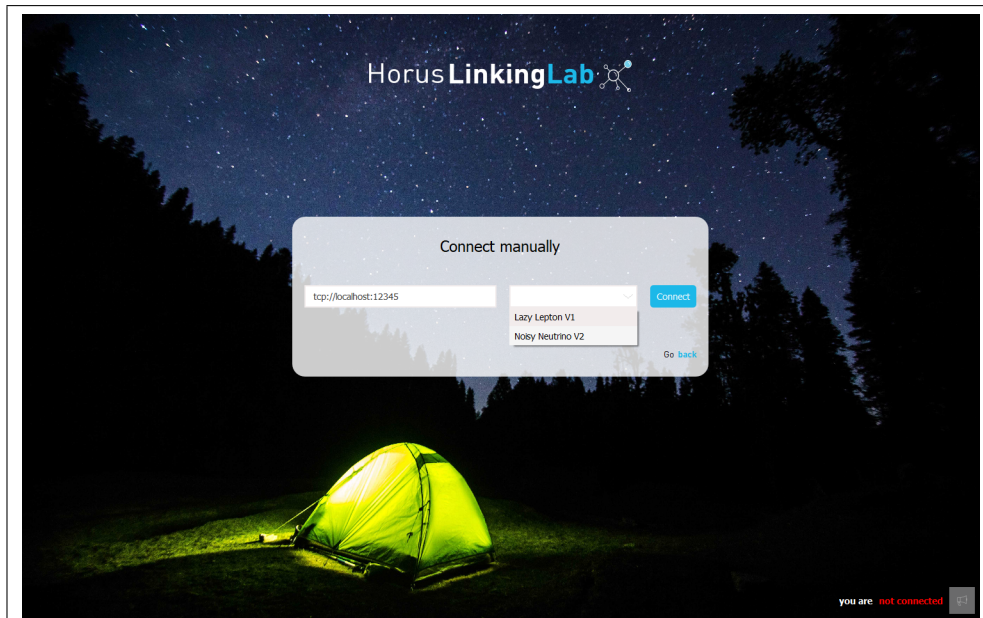
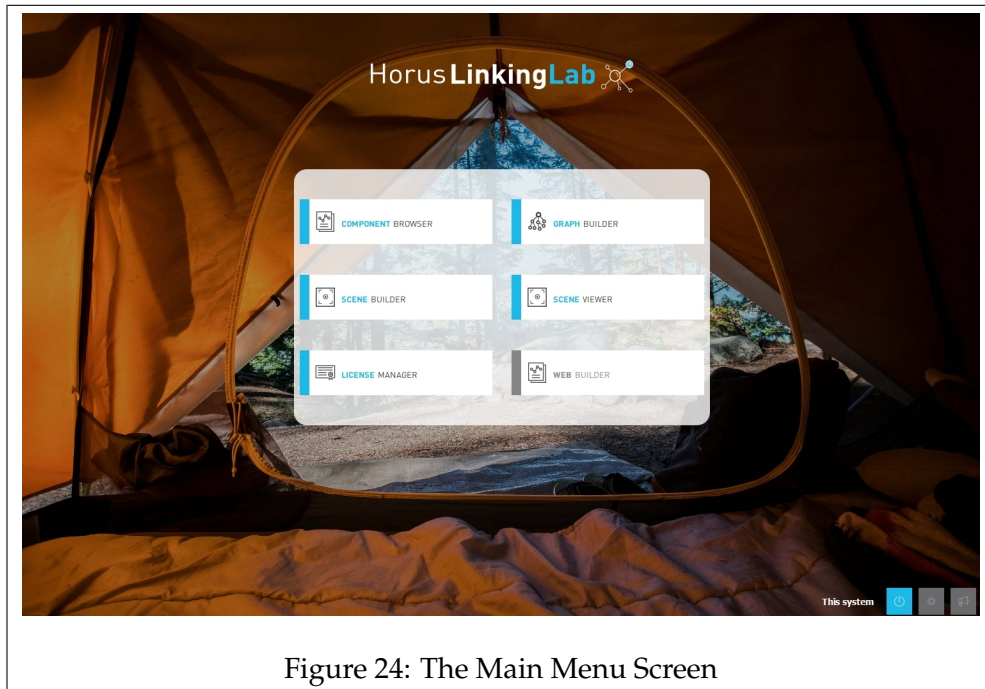


Figure 23: Connect Manually

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.

2.5.3 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.



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.

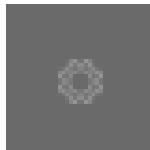
On the bottom right you will find a row of three buttons.



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.

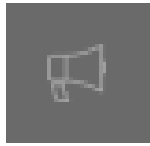
Figure 25:
Disconnect
Button



System Configuration Button

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

Figure 26:
System
Config-
uration
Button

**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.

2.5.4 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.

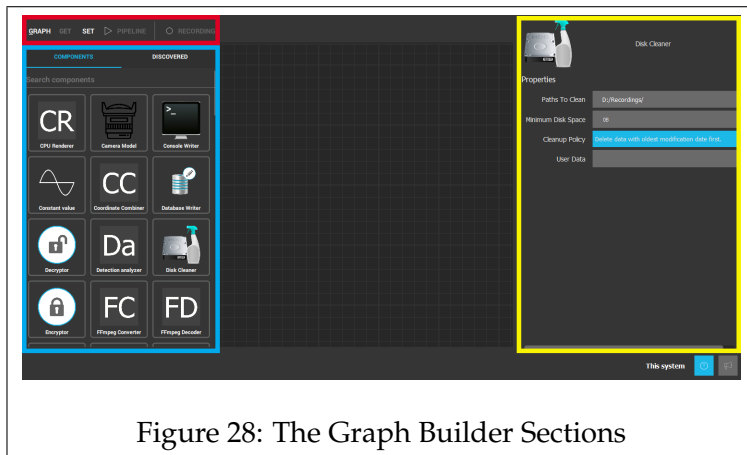


Figure 28: The Graph Builder Sections

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.

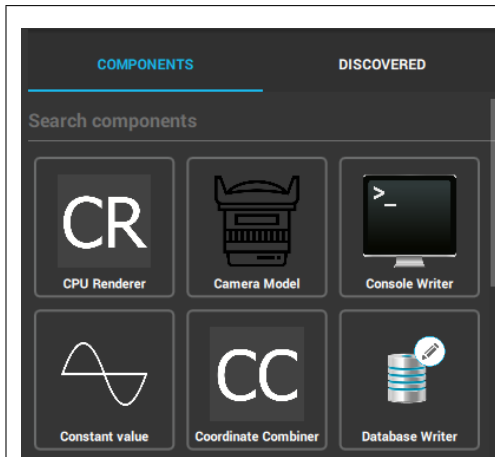


Figure 29: Component List

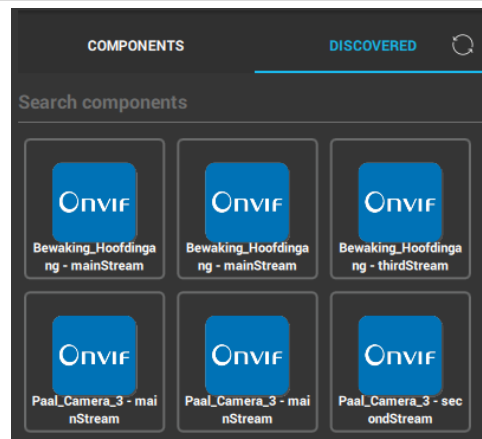


Figure 30: List of Discovered Onvif Devices

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.

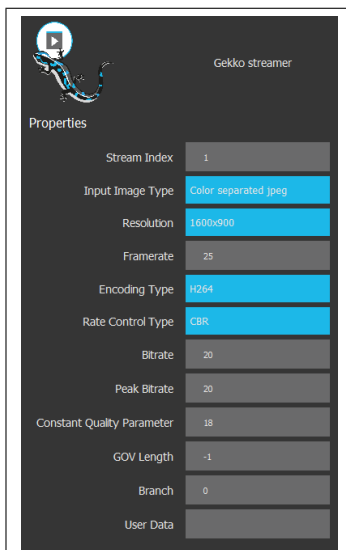


Figure 31: Component properties

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



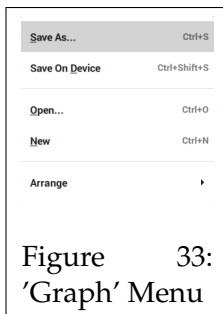
The 'GET' button is used to retrieve the currently set pipeline. Useful if you made changes, but you want to revert to the last set pipeline.

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.

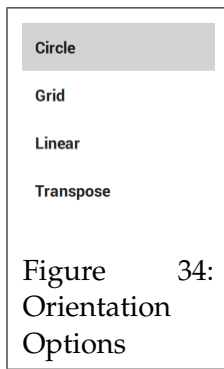
If you press the 'GRAPH' button, a menu will appear. This menu contains several options for saving/loading pipelines, and setting the orientation.



- **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.



- **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.

2.5.5 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.

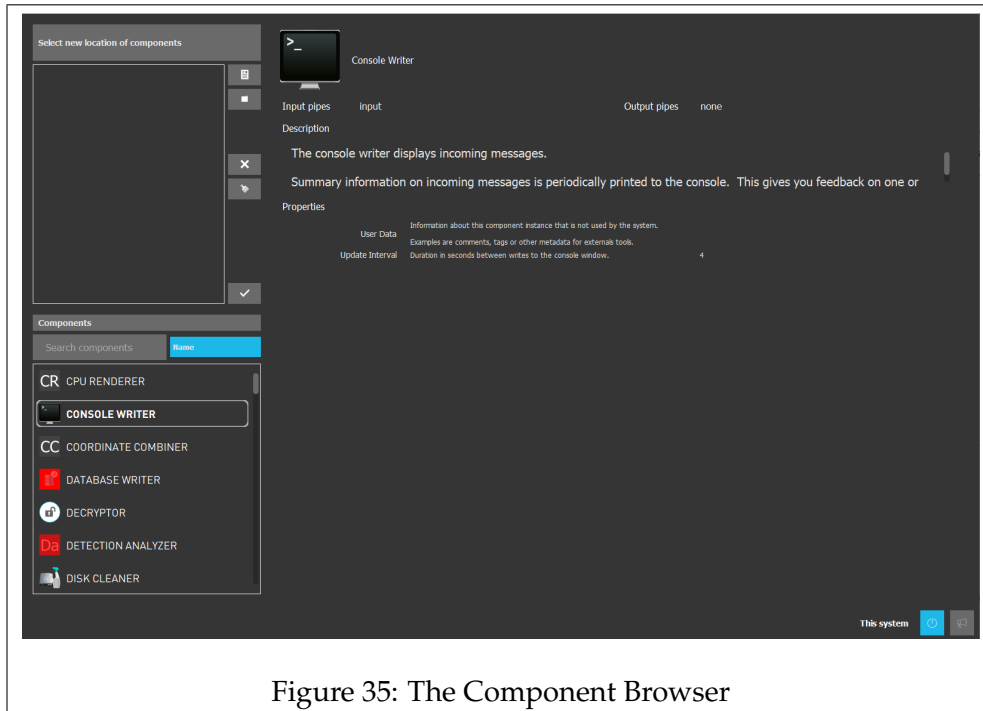


Figure 35: The Component Browser

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.

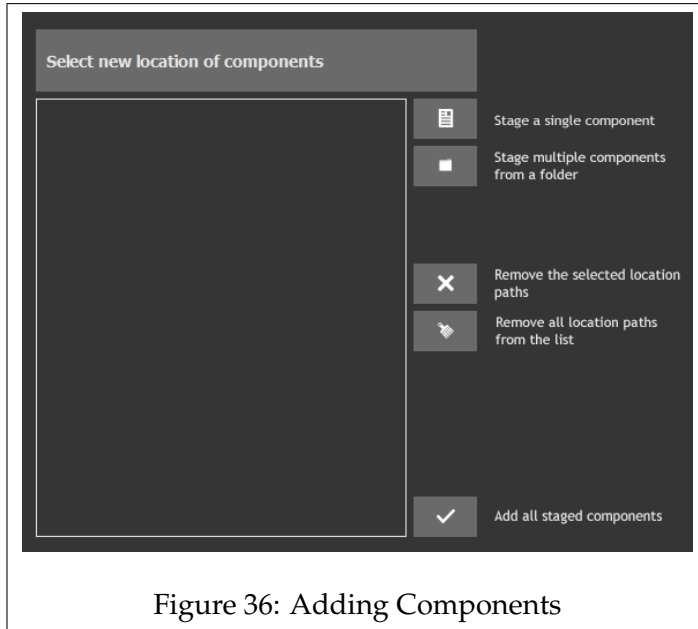


Figure 36: Adding Components

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.

2.5.6 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.

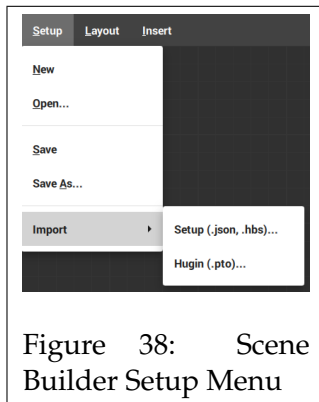


Figure 38: Scene Builder Setup Menu

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.

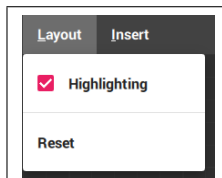


Figure 39:
Scene Builder
Layout Menu

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.

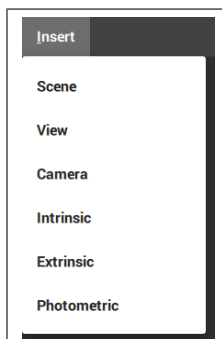


Figure 40:
Scene Builder
Insert Menu

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.

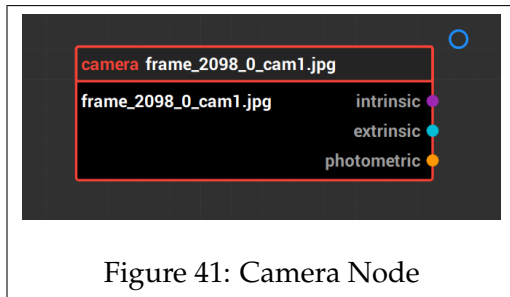


Figure 41: Camera Node

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.

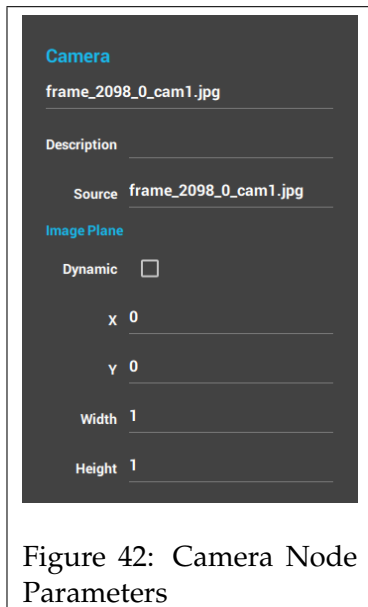


Figure 42: Camera Node Parameters

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.

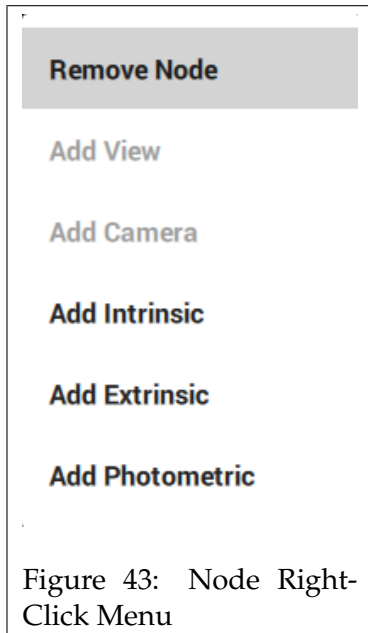


Figure 43: Node Right-Click Menu

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.

2.5.7 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.

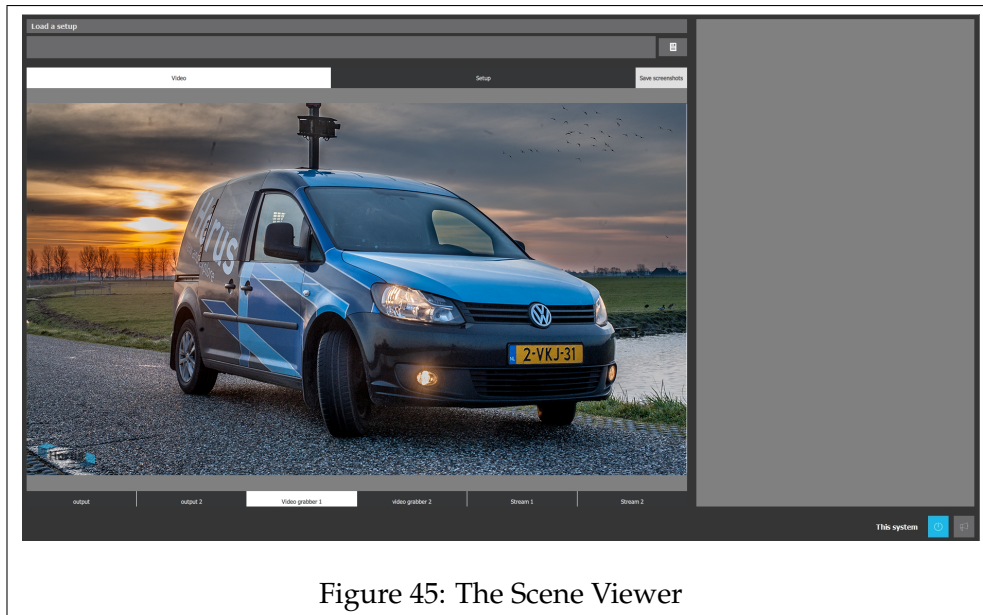


Figure 45: The Scene Viewer

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.

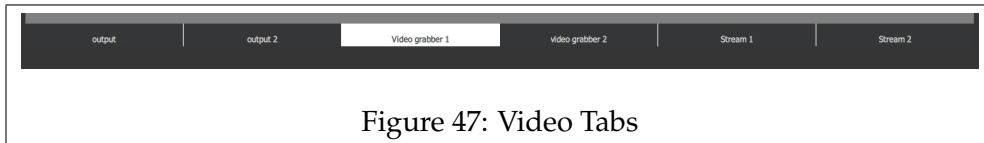


Figure 47: Video Tabs

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'.

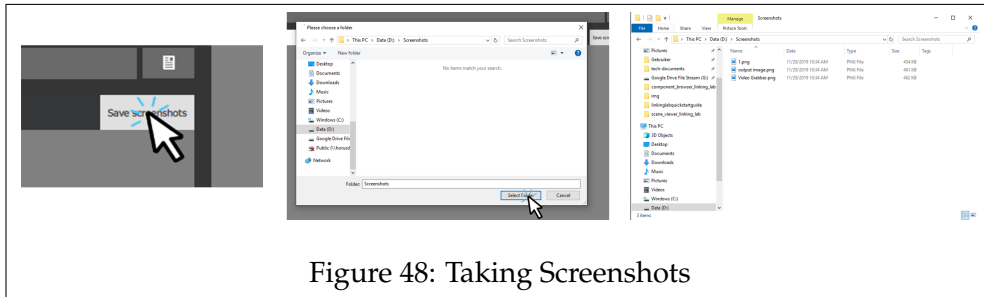


Figure 48: Taking Screenshots

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.

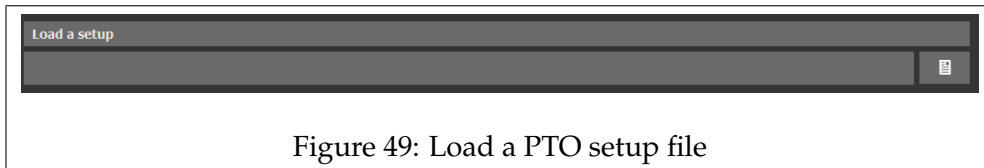


Figure 49: Load a PTO setup file

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.

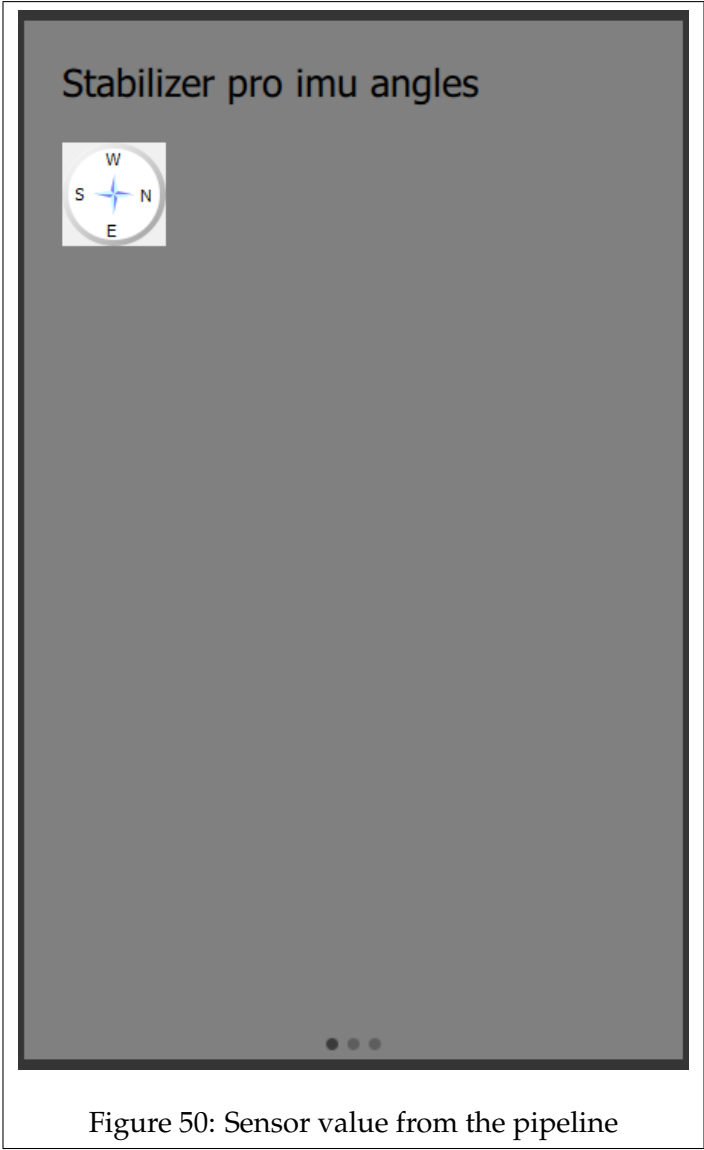


Figure 50: Sensor value from the pipeline

2.5.8 The License Manager

The License Manager summarizes the licenses for each component.

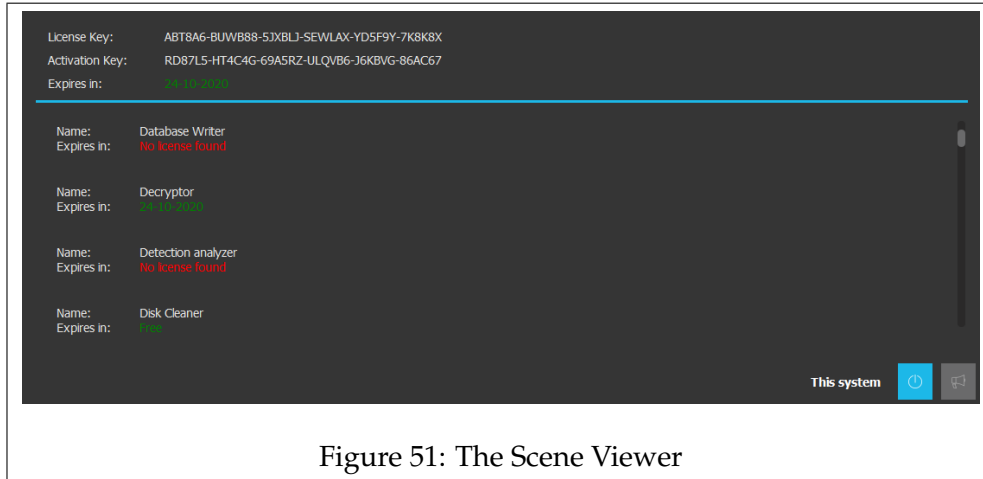


Figure 51: The Scene Viewer

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.

3 Creating a simple Pipeline

3.1 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.

3.2 Basic Functions for Creating Pipelines

3.2.1 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.

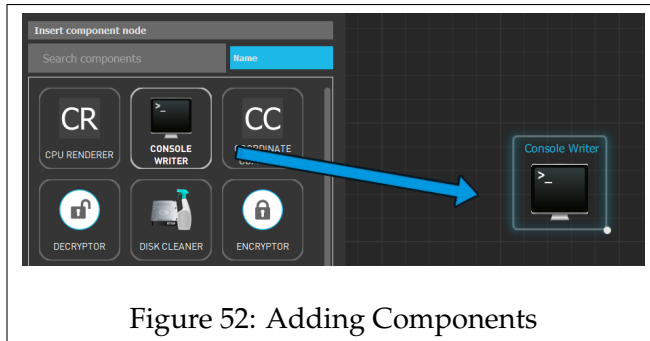


Figure 52: Adding Components

General Component controls

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

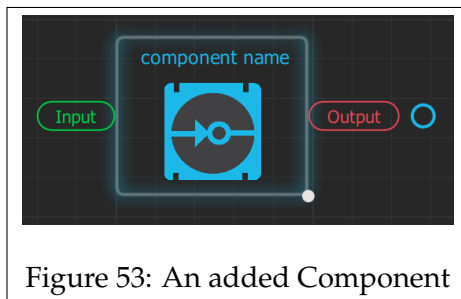


Figure 53: An added Component

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.

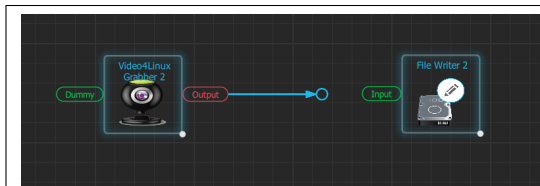


Figure 54: Linking components 1

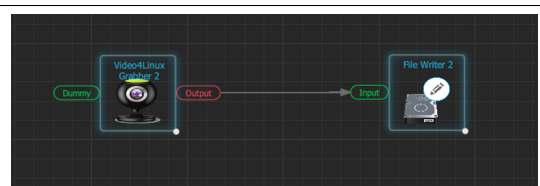


Figure 55: Linking Components 2

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.

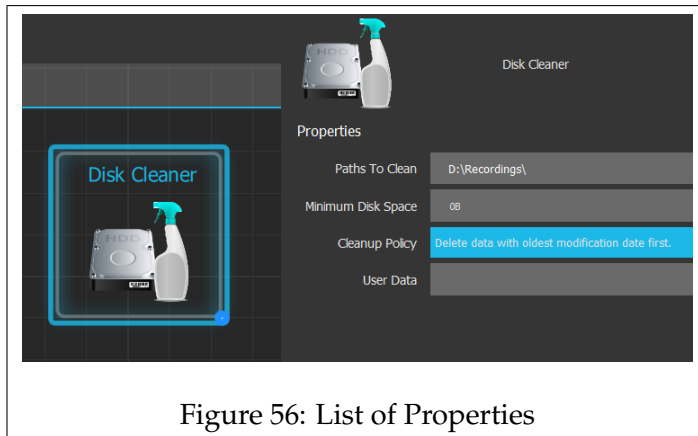


Figure 56: List of Properties

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.

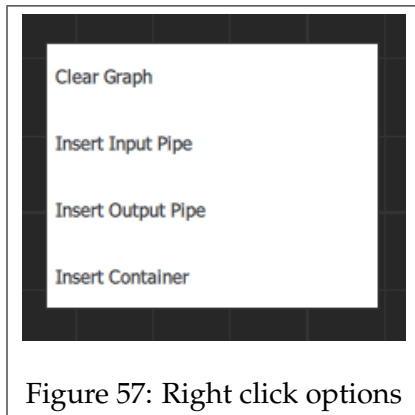
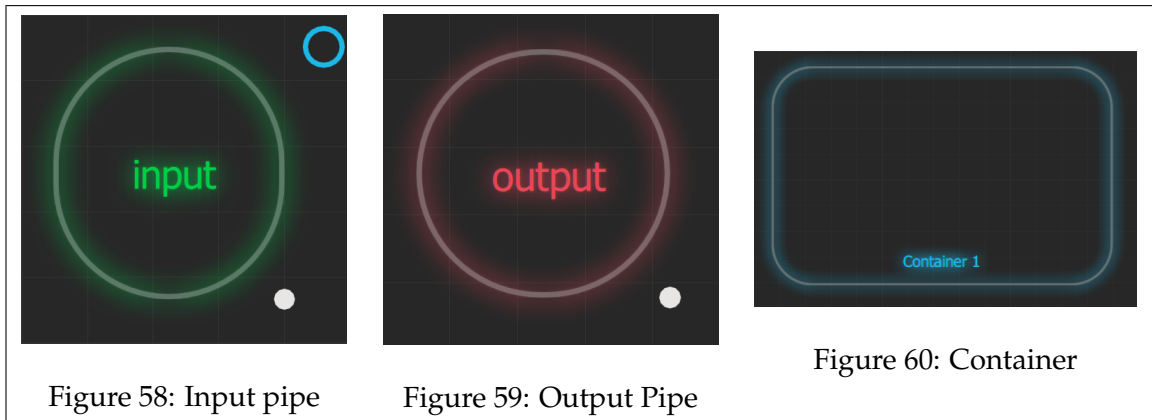


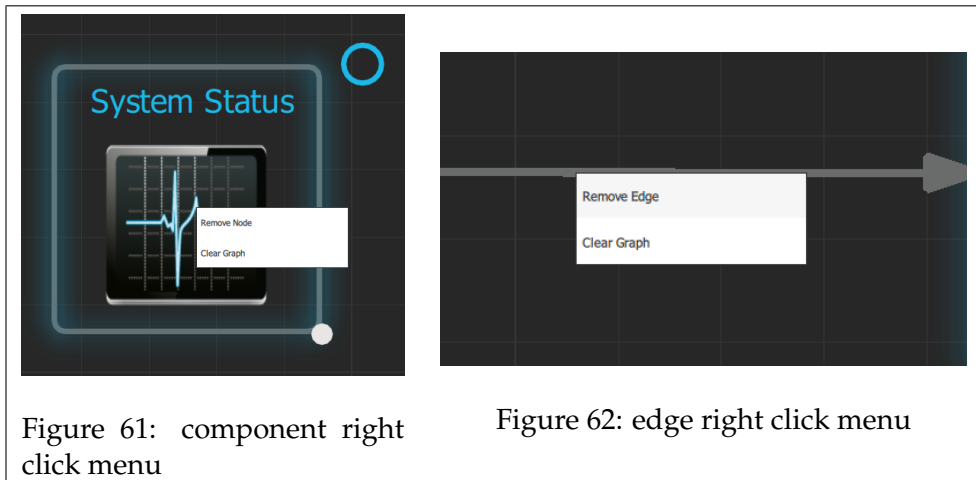
Figure 57: Right click options

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.



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.



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.

3.3 Creating a simple Pipeline with Linking Lab

3.3.1 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.

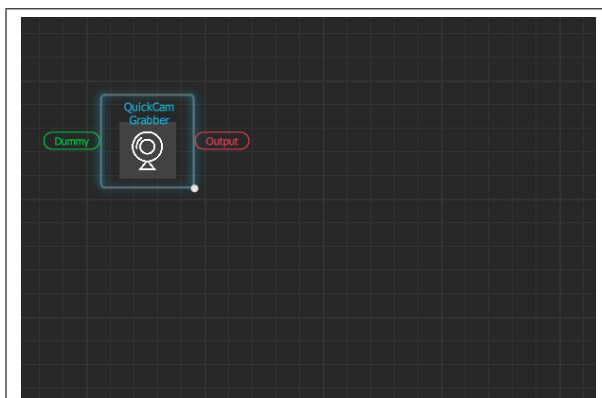


Figure 63: The Quickcam Grabber

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.

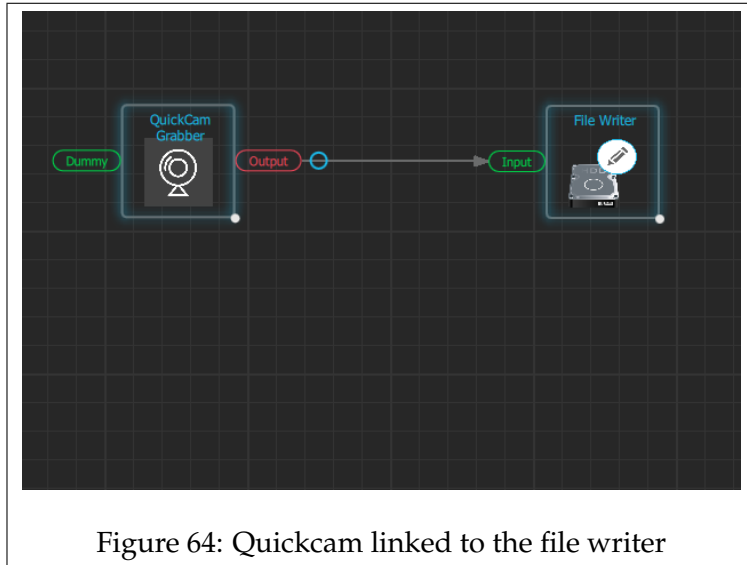


Figure 64: Quickcam linked to the file writer

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.

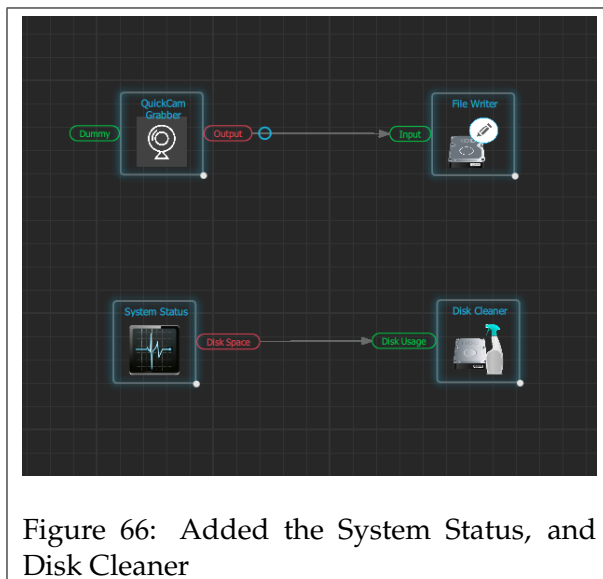


Figure 66: Added the System Status, and Disk Cleaner

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

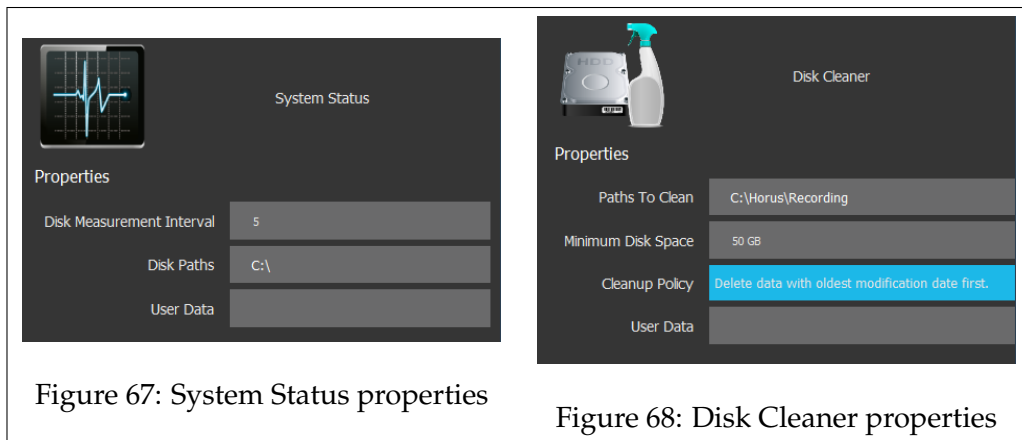


Figure 67: System Status properties

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.

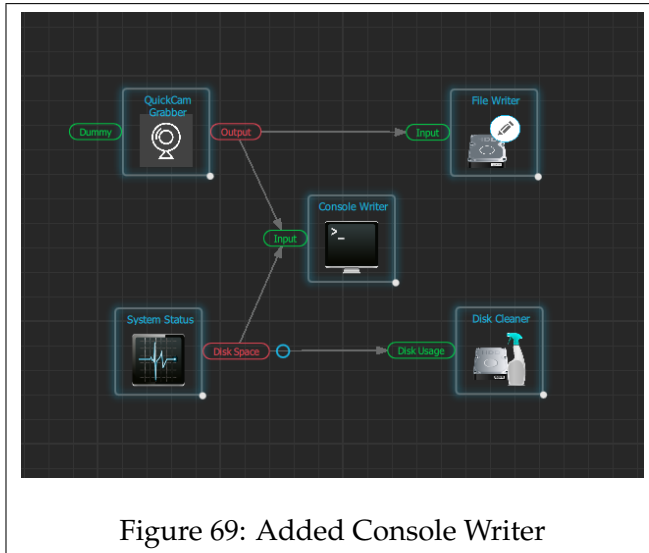


Figure 69: Added Console Writer

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.

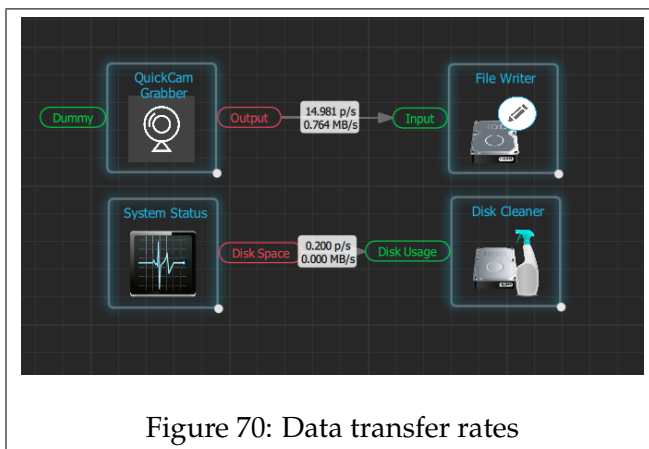


Figure 70: Data transfer rates

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

4 Horus Security Suite

4.1 What is Security Suite

4.1.1 The Security Suite Software

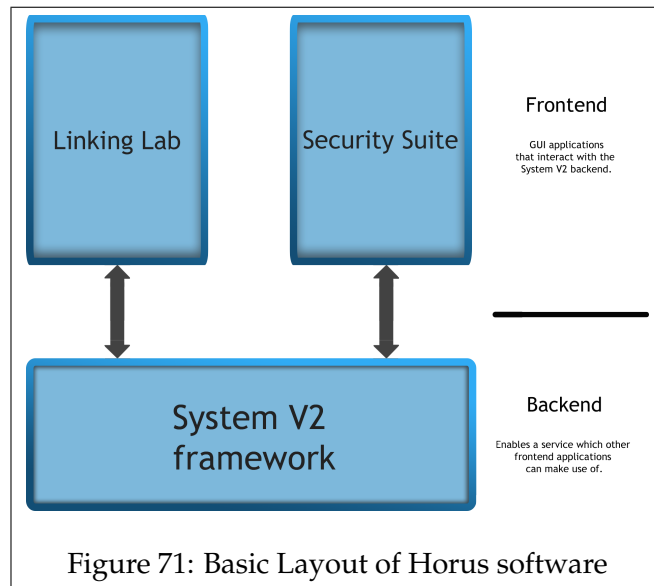
The Horus Security Suite enables persistent wide area surveillance, in either realtime or through recordings.

Build on our modular System V2 framework, any type of camera can be easily integrated into the Security Suite, from specialized cameras, to off-the-shelf consumer cameras.

from large scale home security scenarios, to small businesses, our software has the flexibility to create a system tailored to your needs.

When starting up the Security Suite, a System V2 backend will be created. The System V2 backend runs a so-called pipeline, which can be configured with the Linking Lab software. This pipeline determines the functionality of the backend; data from cameras and other devices is grabbed, this data is then either recorded or sent to an output pipe.

The Security Suite accesses the outputted data from the backend, and displays it in the application.



The Security Suite thus requires a configured pipeline run by the System V2 framework to display received data. This pipeline in turn can be

configured through the Linking Lab software.

4.2 Installing Security Suite

4.2.1 Introduction: Installation of Security Suite

To start working with the Security Suite 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.

4.2.2 The Installation Process on Windows

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

Step 1.

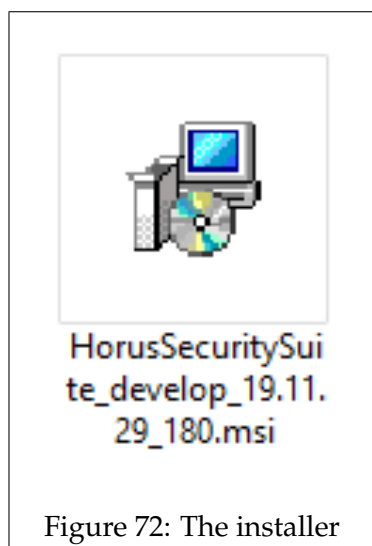


Figure 72: The installer

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

Step 2.

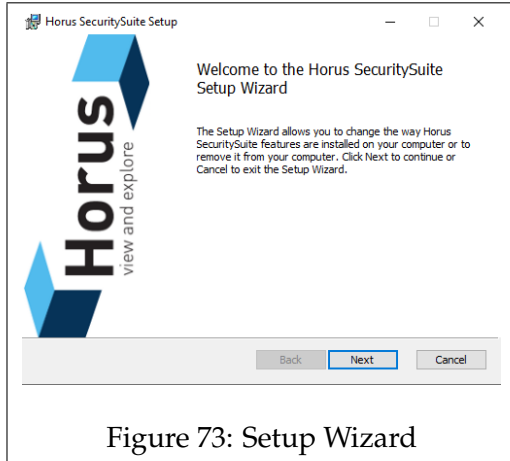


Figure 73: Setup Wizard

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

Step 3.

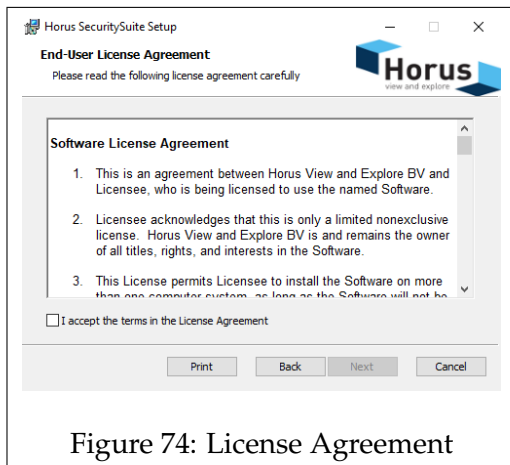
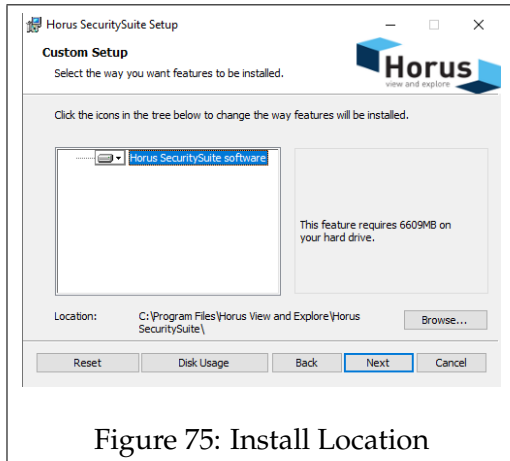


Figure 74: License Agreement

Before continuing ensure that you have read the *Software License Agreement*.

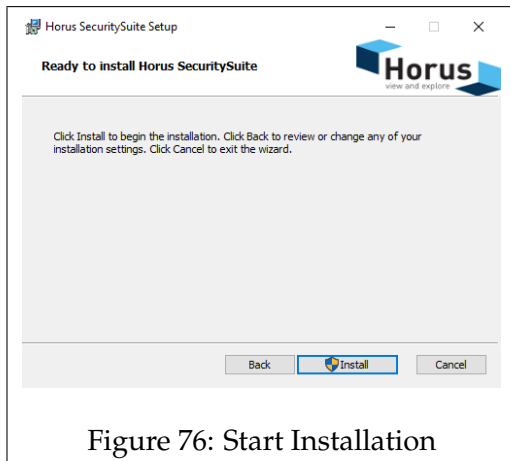
Check the box to accept the terms in the agreement, and press *next* to proceed.

Step 4.



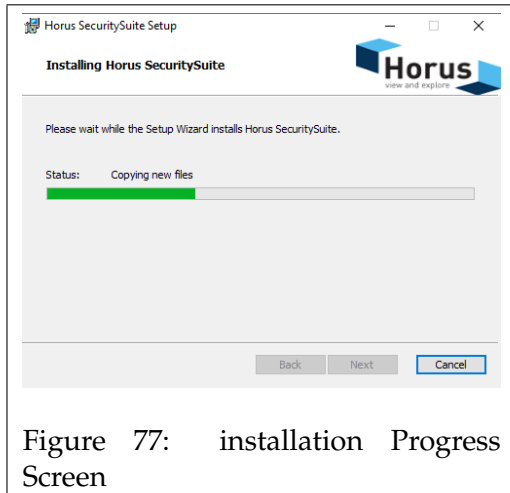
You may change the location where Horus Security Suite will be installed. Click *Next* when the install path is set to the desired location.

Step 5.



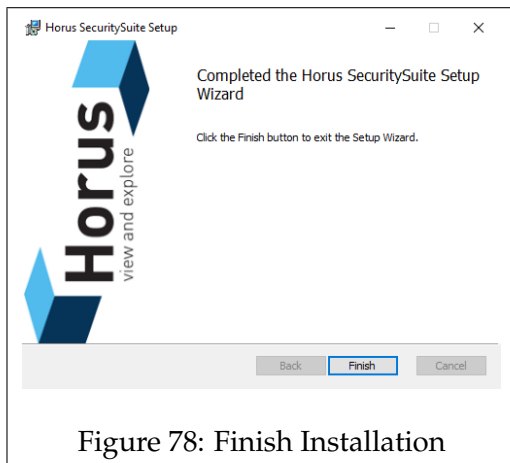
The setup wizard requires final approval before it can install Horus Security Suite. Click *Install* to start the installation.

Step 6.



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

Step 7.



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

Horus Security Suite is now successfully installed on your system.

4.3 Configuring Security Suite

4.3.1 Introduction: Configuring Security Suite

Before you can start working with the Security Suite, you will need to set up the software.

The configuration process involves configuring the backend, this entails creating a pipeline in Horus Linking Lab and then enabling it through the .ini files.

4.3.2 Configuration process

To import video, and other data into Security Suite, a running System V2 backend is required.

To set up the backend, a pipeline is needed. If you do not have a configured pipeline, you will need to create one with Linking Lab.

If you have already have a working pipeline, you may skip the next paragraph.

Creating a Pipeline with Linking Lab

To create a pipeline, start up Linking Lab and open the Graph Builder. for an initial setup the pipeline should be able to output a video stream to the Security Suite.

The minimum amount of components that are required for this are a video grabber, and an output pipe. To enable recording and playback, add a File Writer.

Once the components are in place, link the in/output pipes, and set the properties.

If you have multiple cameras, you can add them by simply adding more video grabbers and linking them to a separate output pipe, and/or File Writer.

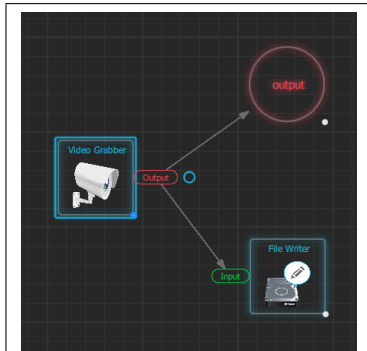


Figure 79: Video Grabber outputting to an output pipe, and File Writer

The setup as shown in figure 79 will output live video, through the 'Output Pipe', to the Security Suite, and can record video, via the 'File Writer'.

More advanced functionality such as positional tracking and PTZ commands will be described in their respective chapters.

You can test the pipeline by running it, if it successfully starts, check the video streams through the Scene Viewer.

Once the pipeline is finished, name and save it to the desired folder.

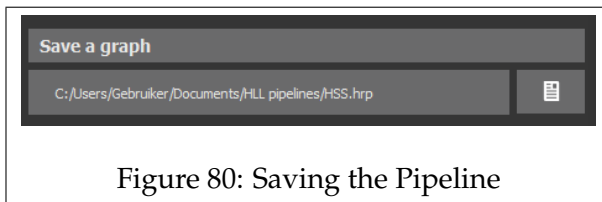


Figure 80: Saving the Pipeline

You can always come back later and update your pipeline.

For more information regarding the Linking Lab software see the *Horus Handbook*, or the *getting started with Linking Lab* document.

Configuring the .ini File

Now that you have a working pipeline, the system needs to be told to load and run it on startup. This can be done through the .ini file.

For Windows the .ini file can be found in the following directory:

C:\Users\Username\AppData\Roaming\Horus View and Explore

When using Linux the .ini file can be found in the following directory:

~/config/horus

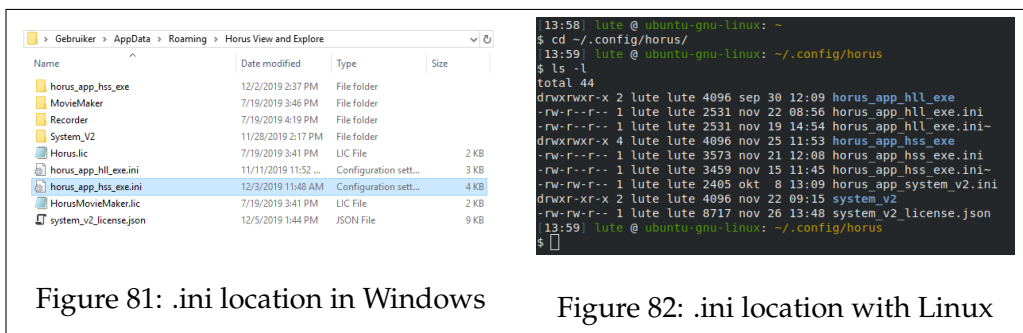


Figure 81: .ini location in Windows

Figure 82: .ini location with Linux

Open the *horus_app_hss_exe.ini* with a text editor. This file contains various configuration options for Security Suite, and the backend.

Most of the options will be commented out with an #. To edit a setting, remove the # and amend the option. Below each option is a short description of the options functionality.

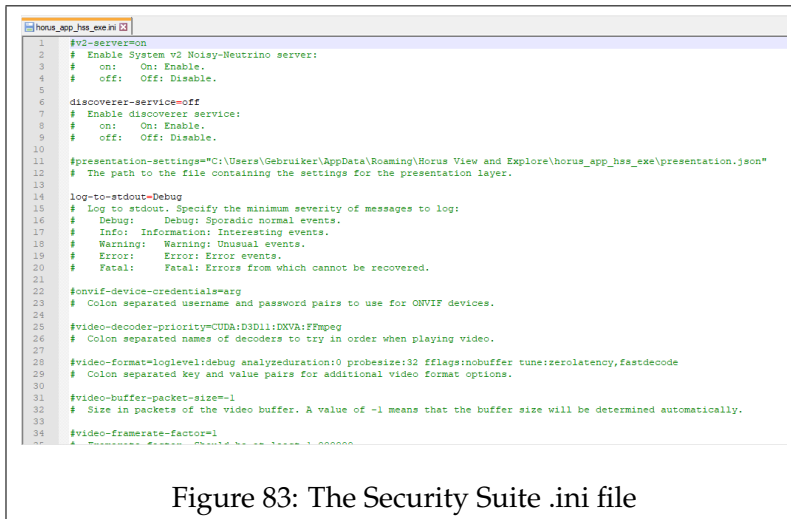


Figure 83: The Security Suite .ini file

We recommend that you configure the following three options as described below.

- **v2-cmp-search-path**
Replace 'arg' with the path of the folder containing the plugins/components between citation marks.
- **v2-pipeline**
Replace 'arg' with the file location of the saved pipeline between citation marks.
- **v2-start-pipeline**
This settings is set to true by default, you only have to remove the # of this line.

```
v2-cmp-search-path="C:\Program Files\Horus View and Explore\Horus SecuritySuite\horus_plugins"  
# Locations of the plugin components.  
  
v2-pipeline="C:\Users\Gebruiker\Documents\HLL pipelines\HSS.hrp"  
# A V2 hrp file containing a pipeline to load at start-up.  
  
v2-start-pipeline=true  
# Start pipeline at start-up.
```

Figure 84: Change these options

By changing these settings, upon starting Security Suite, the application will start running a System V2 backend and load in the required components and pipeline.

Optional Options

The following settings are not required to be set, but can be useful.

- **Discoverer-service**
This setting determines if Linking Lab will try to detect other System V2 instances. Setting it to 'Off' will result in only your local system showing up in the 'Connect Screen' in Linking Lab.
- **log-to-stdout**
This setting changes the messages that are printed to the terminal window. specifies the minimum severity of event that are shown.
- **v2-log-events**

This setting changes the messages that are printed to notification window in Linking Lab, and the Status Window in Security Suite. specify the minimum severity of events that are to be shown.

- **v2-ser-net-url**

This option sets the address and port for the System V2 backend. It is recommended to change the port number, so that it differs from the port set in the *horus_app_hll_exe.ini*. This will avoid any possible conflicts, and makes it easier to differentiate between a backend run by Linking Lab or one from Security Suite.

The Scene Projection Setup

The Security Suite displays the video in what we call 'Scenes', these scenes use the video outputted by the video grabbers from the pipeline. The scenes can be configured to warp the image, create a virtual camera, or to add PTZ functionality to the video.

The scenes are configured in a file called the: 'projection_setup.json'. This file can be found in the following directory:

- **Windows**

C:\Users\Gebruiker\AppData\Roaming\Horus View and Explore\System_V2

- **Linux**

~/ .config/horus/system_v2

Without a projection_setup.json the Security Suite will not display any video.

**note: More details on how to configure this file will be added later*

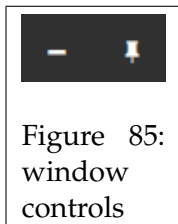
By following the instructions from this chapter, your Security Suite should now have an initial setup with which you can start.

4.4 The User Interface of Security Suite

4.4.1 Introduction: User Interface

The graphical user Interface of Security Suite is made up out of different windows, These windows can display video footage, positional tracking data, or various system related controls.

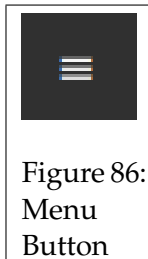
The user can configure their user interface to their personal preferences by adding/removing, moving and pinning windows to the screen.



At the top right of most windows are two icons, as shown in figure 85.

The '-' on the left is used to minimize or close the window.

The right pin icon, pins the window to the screen. You can choose where to pin the window by holding and dragging the windows to the desired location, a highlight will show where the window will be pinned.



Some of the windows may have a menu button at the top left, seen in figure 86.

When clicking on it, a menu will appear. This menu will contain options related to that particular window.

You can resize windows by clicking and dragging the borders to the desired size.

4.4.2 Menu Window

When you start up Security Suite for the first time, the only window that will be active is the *Menu* window



Figure 87: The Main Menu Screen

This window acts as a hub to (de)activate other windows, and contains the controls for playing recorded footage. This window is always active and cannot be minimized.

Playback Video Section

The right half of the window is taken up by the section used for playing back recorded video footage.

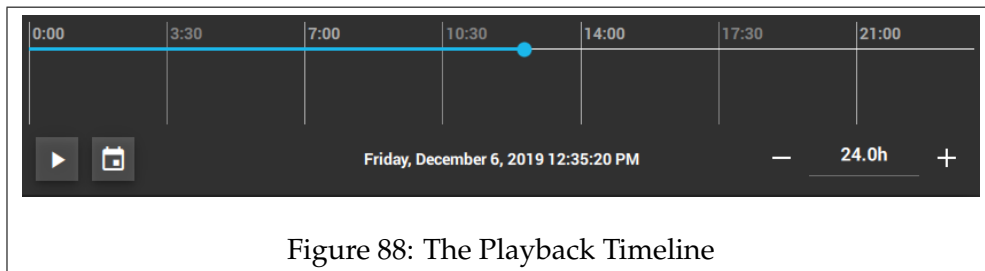


Figure 88: The Playback Timeline

It contains a timeline that the user can scrub through to set the desired time. Below the timeline, the date and time are displayed; this corresponds to the time of the timeline. On the bottom right, you can in/decrease the timescale of the timeline.

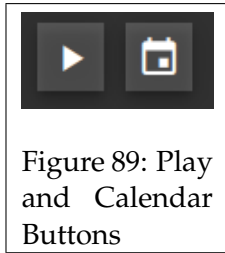


Figure 89: Play and Calendar Buttons

At the bottom left you have two buttons. The left button toggles between playing and pausing. The right button opens a calendar, this is used to access video from a different day.



Figure 90: The Calendar

System V2 information

The bottom right of the window shows information related to the backend.

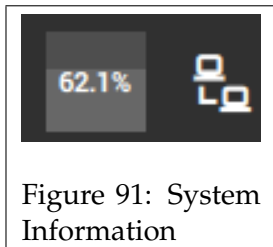


Figure 91: System Information

The icon on the right shows the connection status.

the one on the left, shows information on the storage disks, the fraction of the disk that is in use will be displayed in percentages.

Quick Access Windows Section

The left of the Menu window is primarily used for quick access to different windows and functions.

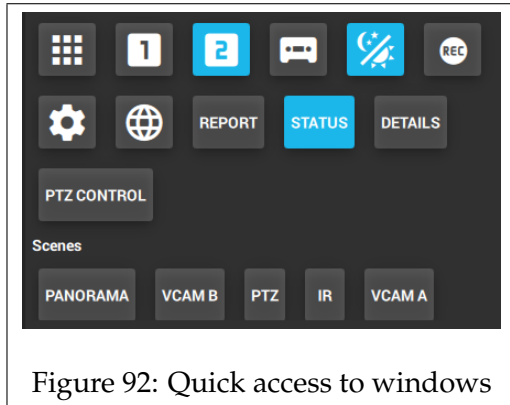


Figure 92: Quick access to windows

The top row consist of various buttons for functions such as: changing layouts, switching between playback and live mode, start/stop recording, and the 'Windows Switcher' menu.

The rest of the buttons are used to show /hide windows. You can add windows to this section via the 'Windows Switcher' menu.

The bottom row contains 'Quick access' buttons for the different scenes.

Windows Switcher

You can open the Windows Switcher menu by clicking on the button on the top left of the Menu window, alternatively, you can open it with the following hotkey: *Ctrl + `*.

This menu allows you to access any of the available windows, and add them to the quick access section of the Menu window.

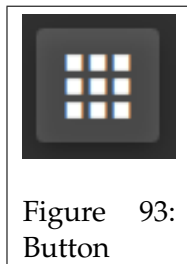


Figure 93:
Button

After clicking on the button, the Windows Switcher menu will appear as seen in figure 94.

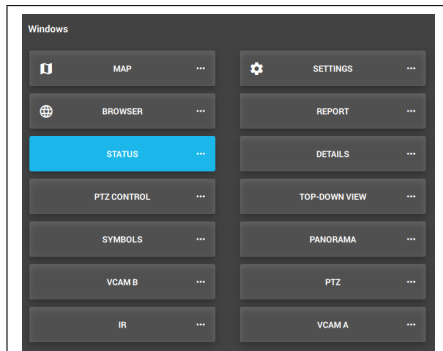


Figure 94: Windows Switcher menu

All the available windows will be displayed.
Windows that are highlighted in blue, are currently active on the screen.

By clicking on one of the windows, you can show or minimize that window.

The three dots on the right of each window, lead to a dropdown menu. This menu contains at least two options. The 'Quick access' option when checked adds that window to the 'Quick Access Windows Section'.

'Minimize' can only be used on active windows. Clicking on it will hide the window.

When a window is pinned, the option to 'Undock' it gets added to the menu. Clicking on it will unpin the window.

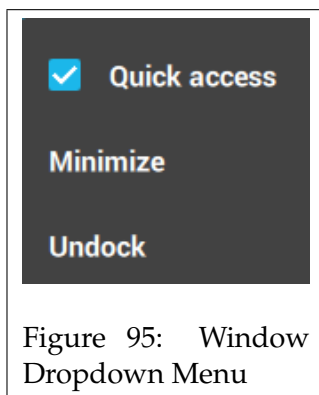


Figure 95: Window Dropdown Menu

Menu

clicking on the menu button at the top left of the window will open a menu as shown in figure 96

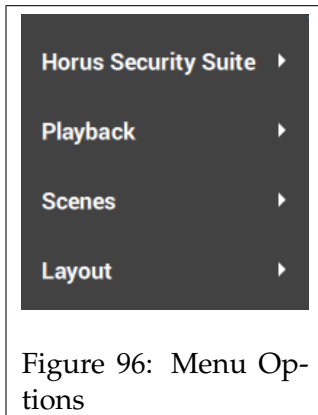


Figure 96: Menu Options

Clicking on 'Horus Security Suite' will open a submenu with the following options:

- **'Stop pipeline'** is used to manually stop and start the System V2 backend.
- **'Night mode'** toggles between dark and light themes.
- **'Windows Switcher'** Opens the 'Windows Switcher' window.
- **'Quit'** will close the Security Suite software

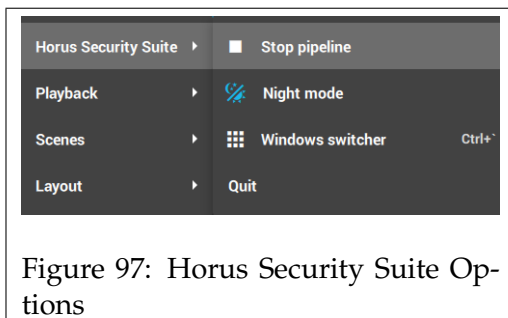


Figure 97: Horus Security Suite Options

The 'Playback' tab contains three options to control the playback of recorded video.

- **'Playback'**, when enabled the application will go into Playback mode, if you wish to view live video, uncheck this option.
Hotkey: *Ctrl + Shift + P*.
- **'play'** toggles between the playing and pausing of recorded footage.
Hotkey: *Ctrl + P*.
- **'Start recording'** start or stop recording video.
Hotkey: *Ctrl + R*.

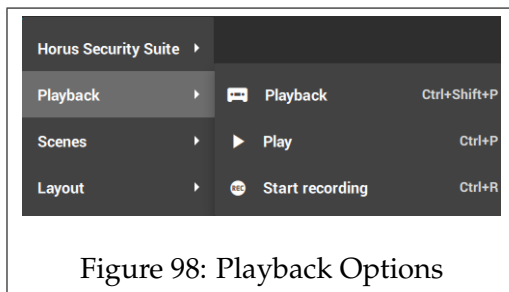


Figure 98: Playback Options

The next tab, 'Scenes', has two functions. These functions change the active scene of the 'Details' window.

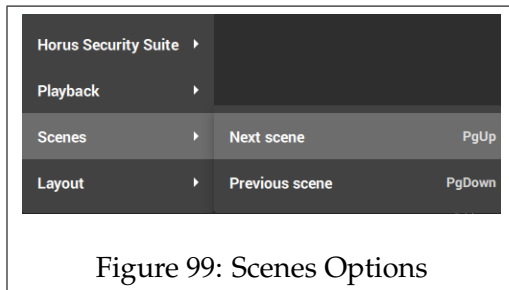
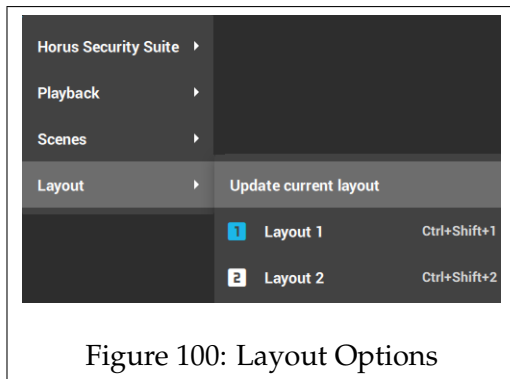


Figure 99: Scenes Options

The last tab, 'Layout', contains options for the layout of the application. The following settings are used to switch between two layout presets, and to save the current placement of windows to one of the two layout presets.

- **'Update current layout'**, saves the current arrangement of windows to the active layout preset.
- **'Layout 1'**, sets Layout 1 as the active layout preset.
Hotkey: *Ctrl + Shift + 1*.

- **'Layout 2'**, sets Layout 2 as the active layout preset.
Hotkey: *Ctrl + Shift + 2*.



4.5 The Windows of Security Suite

4.5.1 Introduction: Security Suite Windows

The User Interface of Security Suite is made up out of separate windows that the user can arrange to their preferences.

In this chapter, the various windows will be described, what they are used for, how they work, and for some windows, how they are configured.

4.5.2 Browser Window

The Browser window adds an internet browser to the screen.



Figure 101: Browser Window

Useful when you wish to look up something on the internet, without having to close the software.

4.5.3 Status Window

The status window logs events in a window. Useful for debugging, or monitoring the system.

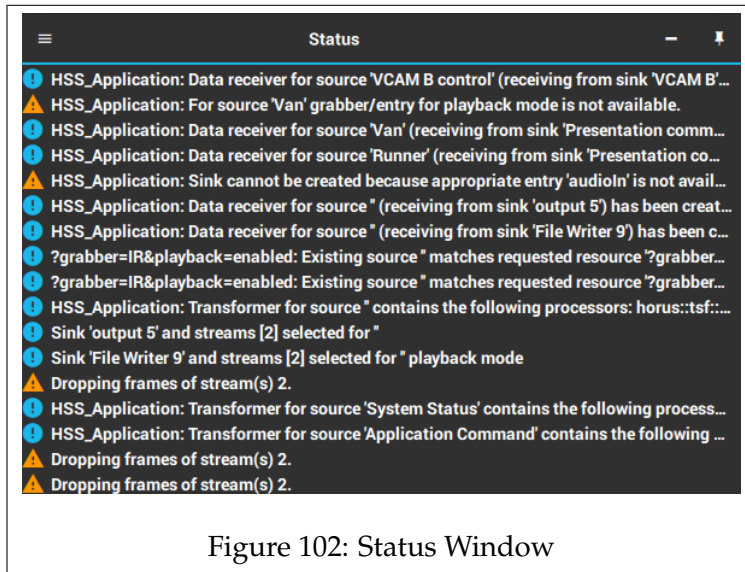


Figure 102: Status Window

The *'v2-log-events'* option in the `hss.ini` file determines the type of event messages that are shown on the Status window.

To clear the window of notifications, click on the menu icon on the top left, and press *'Clear Notifications'*.

4.5.4 Settings Window

The Settings window contains various options for the application. Below, you will find a short description for each setting.

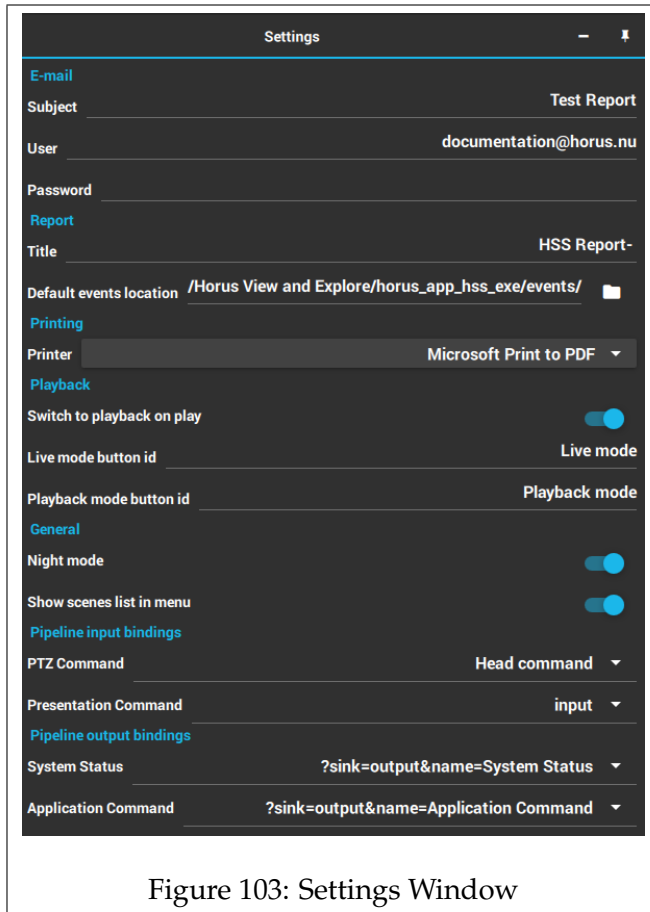


Figure 103: Settings Window

E-mail

- **Subject**
The subject name that is given to the sent E-mail.
- **User**
The username of the SMTP server.
- **Password**
The password of the SMTP server.

Report

- **Title**
Select the printer used to print out reports.
- **Default events location**
The default directory to store events in.

Printer

- **Printer**
Select the printer used to print out reports.

Playback

- **Switch to playback on play**
If enabled, The software will automatically switch to playback mode when the play button or key is pressed.
When disabled, you will need to manually switch to playback mode to play recorded video.
- **Live mode button id**
The button id used when switching to 'Live mode'
- **Playback mode button id**
The button id used when switching to 'Playback mode'

General

- **Night mode**
This setting toggles between a dark and light theme.
- **Show scenes list in menu**
Toggles the quick access buttons for the scenes in the Menu window.

Pipeline input bindings

- **PTZ Command Sink**
Select the input pipe that receives PTZ commands from the 'PTZ Control' window.

- **Presentation Command Sink**

The Presentation Command Sink receives commands for the presentation layer. Select the input pipe that is linked to the output pipe acting as the Presentation Command Sink.



Figure 104: Presentation Command Sink in the Pipeline

Pipeline output bindings

- **System Status Source**

This setting is used to get the disk usage information for the 'Menu' window. The pipeline should contain a 'System Status' component that is linked to an output pipe.

Click on the triangle and select the output pipe that is linked to the 'System Status' component.

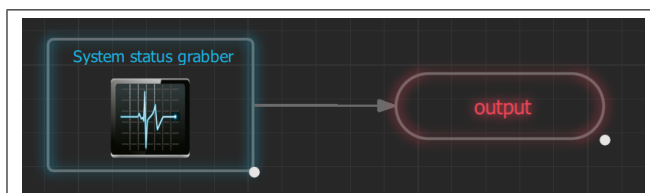


Figure 105: System Status Source in the Pipeline

4.5.5 Report Window

The Report window is used to manage reported events.
The Report Window:

- Files captured images by date and time through events.
- Enables the reviewing of taken photos and their location (if applicable), along with adding comments for each image.
- Allows printing of reports, or sending them by E-mail.

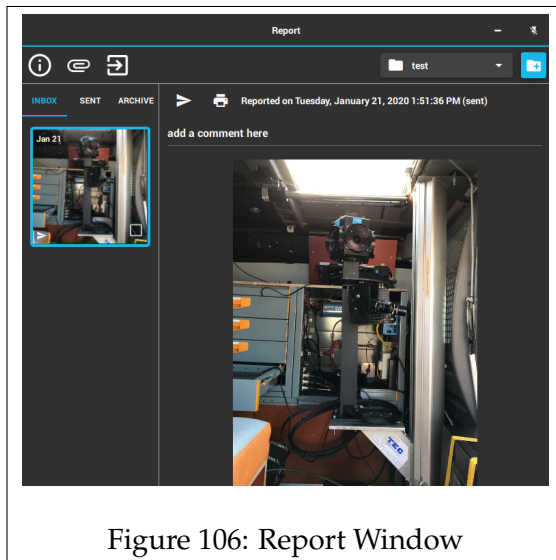


Figure 106: Report Window

Creating Events

The saved images are stored in so-called events. Multiple events can be created to group relevant screenshots together.

The blue icon at the top right is used to create a new event. Enter information pertaining to the event, and press 'Save' to create the event.

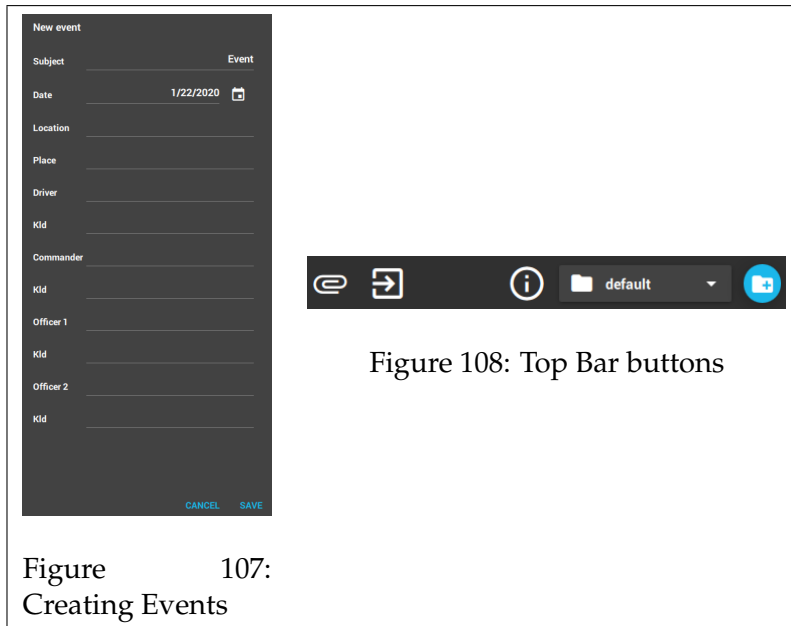


Figure 108: Top Bar buttons

To the left of the 'create event' button there is a dropdown menu, this menu is used to switch between events.

The information icon on the top bar displays information of the current event.

You can open the file directory of the current event by pressing on the 'Open Event Folder' button on the top left.

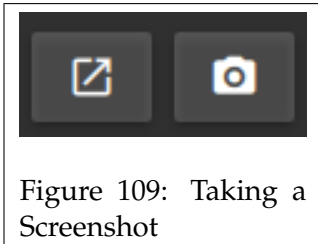
Adding Images

Images can be added via two methods.

- **Taking screenshots inside the Security Suite.**

You can take screenshots in the Security Suite through the Details window.

Click the camera button (figure 109) on the bottom right to take a screenshot of the active scene.



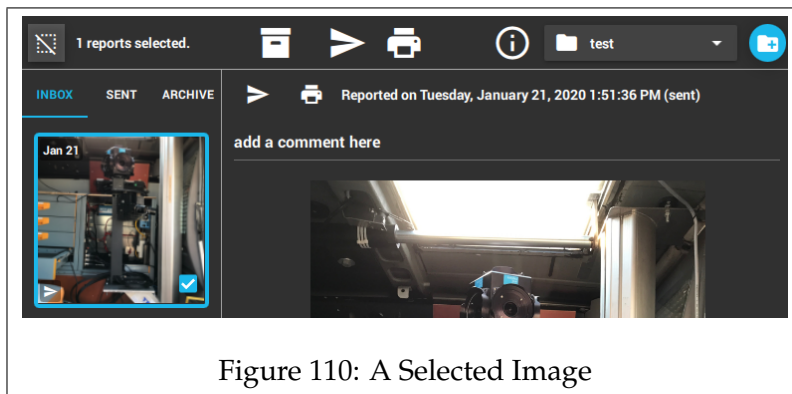
- **Importing photos from disk.**

Images can be imported by pressing on the paperclip icon on the top left, as seen in figure 108. This will open a file explorer window, select the desired image, and press 'Open' to add it to the Report window.

Managing Images

Once you have added an image, it will be shown in the column on the left. By clicking on an image it will be shown on the main portion of the window.

From here you can see the date and time of when the image was taken, and also the position (if there is positional data available). You can also add notes in the input field for each image.



Once you have selected one or multiple images, you can do various actions such as printing or sending a report. You can also archive the selected images, do this by pressing on the 'Archive Reports' button located on the top bar.

Printing a report

You can print out a report by selecting the images that you wish to include, and then pressing on the 'Print' button located on the top bar. The report will be printed on the device specified in the 'Settings' window.

Sending a report via E-mail

You can print out a report by selecting the images that you wish to include, and then pressing on the 'send' button located on the top bar. To send a report via E-mail an E-mail address and password are required. these can be set in the 'Settings' window.

To set the recipient, and additional settings regarding sending reports by E-mail, see the 'report_settings.json'.

This file can be found in the following location:

- **Windows**

- C:\Users\\AppData\Roaming\Horus View and Explore\horus_app_hss_exe\report_settings.json

- **Linux**

- ~/ .config/horus/horus_app_hss_exe\reports_settings.json

For more information on how to configure the 'report_settings.json' see Appendix 10.4.

4.5.6 Details Window

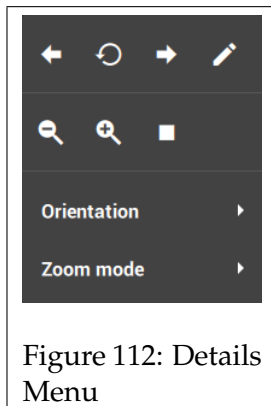
The 'Details window' is where you can view all the different scenes.



At the bottom of the window you have a tab for each scene, by clicking on a tab, you can view that particular scene.

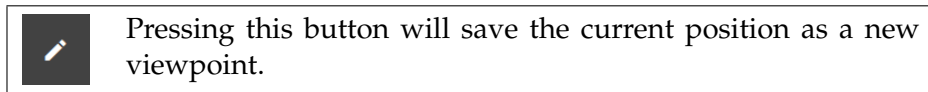
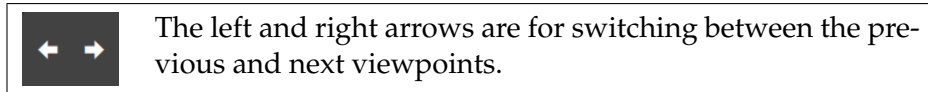
The active tab will have an icon on the right, clicking on this icon will open a 'clone' window, with the video of that tab.

On the bottom right of the window is a button which will take screenshots of the active scene.

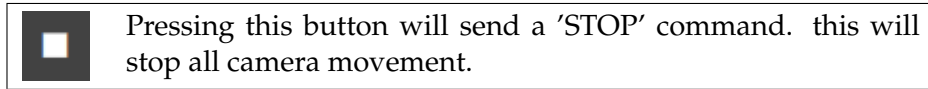


When you click on the menu icon at the top left in the 'Details' tab or any other video window, it will open a menu seen in figure 112. This menu includes buttons to interact with the viewport.

The buttons on the top row are for controlling the viewpoints.



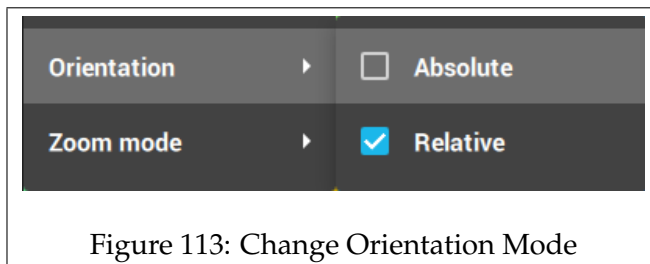
The following buttons are from the second row.



open the Orientation submenu to switch between 'Absolute' and 'Relative' mode.

In 'Absolute' mode the viewpoint will move to the location where you click on the video

In 'Relative' mode you click and drag, and the video will follow the direction that you are moving towards.



The 'Zoom mode' can be switched between 'Stepped' and 'Speed'. This will change the logic used to zoom in and out with the scroll wheel.

- '*Stepped*' will send an absolute value to zoom in or out.
- '*Speed*' sends a relative value to zoom in or out. This value will be sent for the duration of the scrolling action, upon stopping with scrolling a 'Stop' command will be sent.

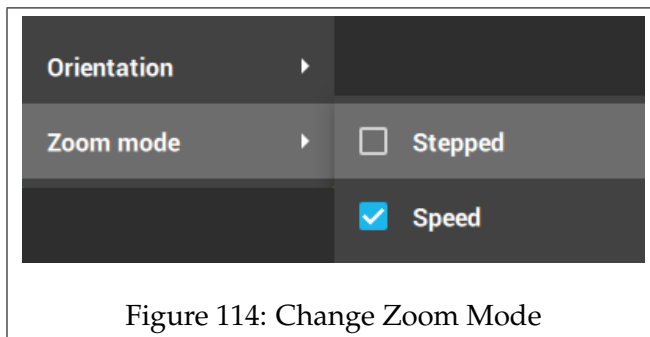


Figure 114: Change Zoom Mode

4.5.7 PTZ Control Window

The 'PTZ Control' window is used to control PTZ devices or scenes by sending virtual PTZ commands.

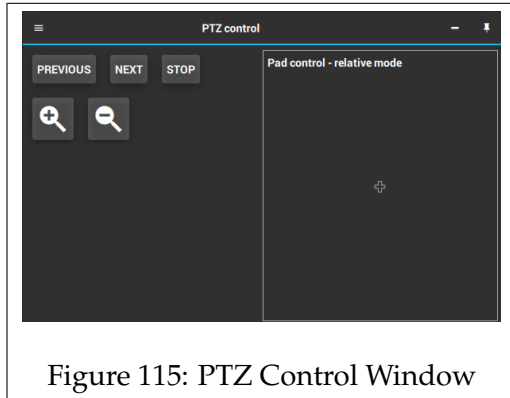


Figure 115: PTZ Control Window

The pad on the right of the window can be dragged around. This will send PTZ commands to the active scene.

The 'Previous' and 'Next' buttons will switch the scene that gets controlled by the PTZ Control window. Pressing the 'Stop' buttons will sent a stop command; this will stop all ptz movement.

The zoom buttons are used to zoom in and out.

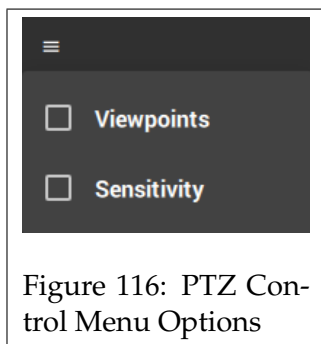


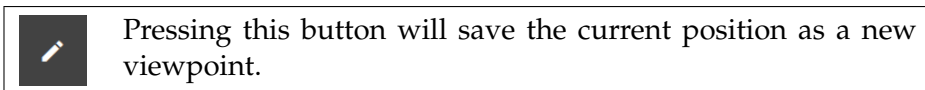
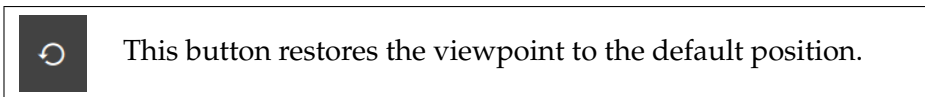
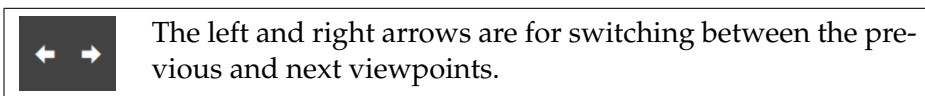
Figure 116: PTZ Control Menu Options

Opening the menu at the top left of the window, will display two options. Checking the box will add additional buttons to the window.



Figure 117: Viewpoints Buttons

Checking the Viewpoints option will add four buttons to the window, which are used to change, and create new viewpoints.



An identical set of buttons can be found in the menu at the top left of most video windows, such as the 'Details' window.

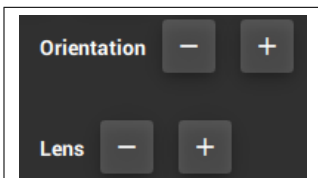


Figure 118: Sensitivity Buttons

The Sensitivity option adds settings to the window that allows you to adjust the sensitivity of the panning and zooming of the viewpoint.

Orientation, This adjusts the speed at which the viewpoint pans around.

Lens, Changes the amount that is zoomed in/out per button press.

These settings also affect the rate at which you can pan and zoom in a video window if the 'Relative', and 'Stepped' modes are selected. (see chapter 4.5.6)

5 CityMapper Manual

5.1 Introduction

Thank you for using our Horus Citymapper system. Equipped with a Ladybug 5 camera, and an Applanix GNSS/INS system, which, when used alongside our software suite, provides users with a powerful all-in-one mobile mapping system.

This document will describe the process of assembly and installation, the initial configuration, recording, post-processing, and viewing the recorded data.

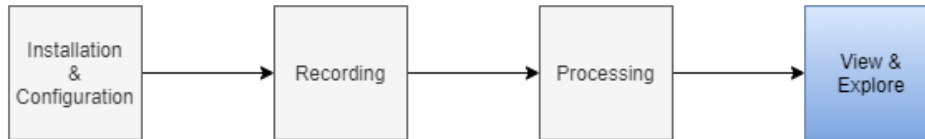


Figure 119: Citymapper manual process

The goal of this manual is to provide you with the information required to use the Citymapper system.

If you run into issues or require further assistance, contact Horus for support.

For more information on the software applications used, see their specific manuals.

5.2 The Citymapper System

5.2.1 What is the Citymapper?

The Citymapper is a turn key solution from Horus for mapping your surroundings with 360 degrees streetlevel imagery.

The system can be employed on different platforms such as: cars, boats, trains.

The system uses a FLIR Ladybug 5 camera, and an Applanix GNSS/INS system, which is mounted on an all-in-one system, additional sensors can easily be mounted and integrated into the system.

The system runs on our proven Horus Geosuite, which is easy to learn and use, and requires minimal setup time.



See Appendix 10.6 for the spec sheet of the Horus Citymapper.

5.2.2 Citymapper Kit Contents

The Citymapper system contains the following components:

- 1x FLIR Ladybug 5
- 1x GNSS/INS system from Applanix
- 1 or 2 Trimble LV59 antennas
- 1x Triggerbox
- 1x Roof mount with required hardware

- 1x USB 3 type A to micro USB 3.0 cable
- 1x USB type A to B cable
- 1x XLR to IO cable
- 1x XLR to Serial cable
- 1x IO to 3 Serial connectors cable
- 1x USB to Serial cable
- 1x Serial to Serial loopback cable
- COAX to COAX cable (1 for each antenna)
- 1x Ethernet cable
- 1x power cable (Triggerbox)
- 1x power cable (Applanix)

- Computer system

Included components may differ.



Figure 121: Citymapper Case

By default the Applanix can be used to measure the distance and determine when to take an image. If you wish to more accurately determine when to trigger the system, then an OBD Reader can be used.

OBD Reader

- 1x OBD Trigger unit
- XLR to XLR cable

The system can be integrated with extra sensors, such as high resolution cameras, and LIDAR. Adding an HR camera to the system will add the following components:

HR Camera

- HR camera
- USB 3 type A to micro USB 3.0 cable
- XLR to IO cable

5.2.3 Included Software

The Citymapper comes included with Horus software, allowing you to record and view data.

This includes:

- Horus Data Grabber *
- Horus Recorder Builder *
- Horus Movie Recorder *
- Horus Movie Player * #
- Horus Position Fixer #

Due to using off-the-shelf components from FLIR and Applanix, their software is also required.

These are:

- LV POSView *
- POSpac MMS #
- FLIR LadybugCapPro *
- FLIR Flycapture SDK (For FLIR Grasshoper HR cameras) *

We also recommend installing the following applications on your system.

- FileZilla (FTP client) *
- Realterm: Serial Terminal *
- OSMfiledownloader *

In cases where you have a dedicated recording and post-processing PC, we recommend installing the applications with an '*' to the recording system, and the software with a '#' to be installed on the processing system.

5.2.4 System Requirements

Power Requirements

The Citymapper system requires **19 V DC** power.

If your vehicle does not output 19 V, then a converter is required.

In cases where a 12 V car battery is your power source, a 12-19 V converter with at least 12 amps is needed.

Horus recommends a 12-220 V Victron converter, but it is important to have clean power delivery for a stable system.

PC System Requirements

The minimum requirements for the PC system are:

- I7 or equivalent CPU
- GTX 10X0 GPU
- 8 GB RAM
- Storage Drive with enough available space for recording
- 3+ USB 3.0 ports

5.3 Setting up the System

5.3.1 Assembling in the System

When you get the Horus Citymapper it requires some assembly. In appendix 10.7 you can see how the components are connected.

Applanix



- Plug the power cable into the 'PWR' port.
- If you have a wheel speed sensor, plug the cable into the 'DMI' port.
- the IO to 3x serial connectors cable can be plugged into the 'IO1' port.
- The antennas are plugged into the 'A1' & 'A2' port of the Applanix with the COAX to COAX cables. The antenna plugged into the A1 port will be the primary antenna.
- Plug the Ethernet cable into the 'LAN' port.

Note: The number and location of the ports might differ between the different models.

Triggerbox

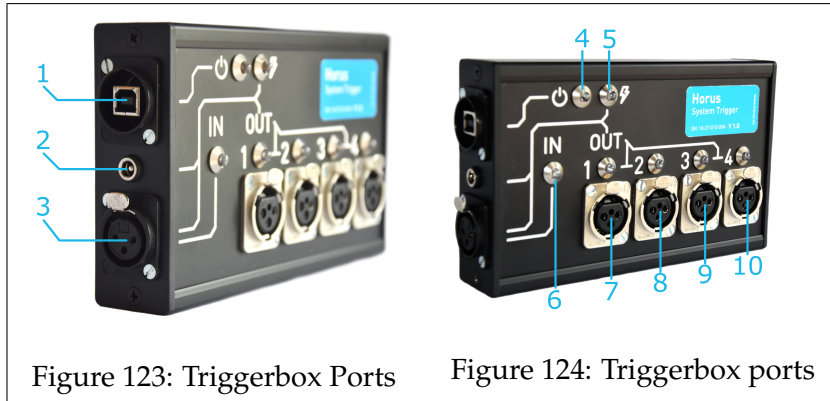


Figure 123: Triggerbox Ports

Figure 124: Triggerbox ports

1. Plug the USB type A to B cable into this port. This is used for the connection to the PC.
2. Optional power throughput, 5-24 Volt DC. This will provide additional power to the hardware that are plugged into 'TRG Out' ports.
3. 'TRG In', The Triggerbox receives trigger pulses via this port from either the Applanix or an OBD Reader Unit.
4. Power LED
5. Power throughput LED, Turns on when power is received through port 2 or 3.
6. trigger input LED, it will blink when a pulse is received through port 3.
7. Port 7 through 10 are all 'TRG Out' ports.

Once the system is assembled, turn on the Computer and Applanix, and check that all components are powered. With the system up and running, continue to the next chapter in which the initial configuration process is described.

5.4 Initial Setup of the System

5.4.1 Configuring the Software

With the system assembled and the software installed, you may continue with the initial configuration.

Step 1

Ensure that the system has power, and start the recording computer, and the Applanix.

Step 2

Start LV POSView.

This application is used to monitor and log the incoming data from the Applanix.

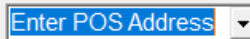
A screenshot of a software interface showing a text input field with the placeholder text "Enter POS Address" and a small downward arrow on the right side.

Figure 125: POS Address Field

Wait for the IP address to show up in the 'Enter POS Address' input field, or select it manually.

The default IP address of the Applanix is:

192.168.53.100

Once the Applanix has been found, connect to it. Check that the status at the bottom left of the window states 'Connected'.

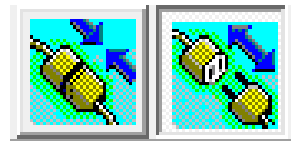


Figure 126: Connect Button

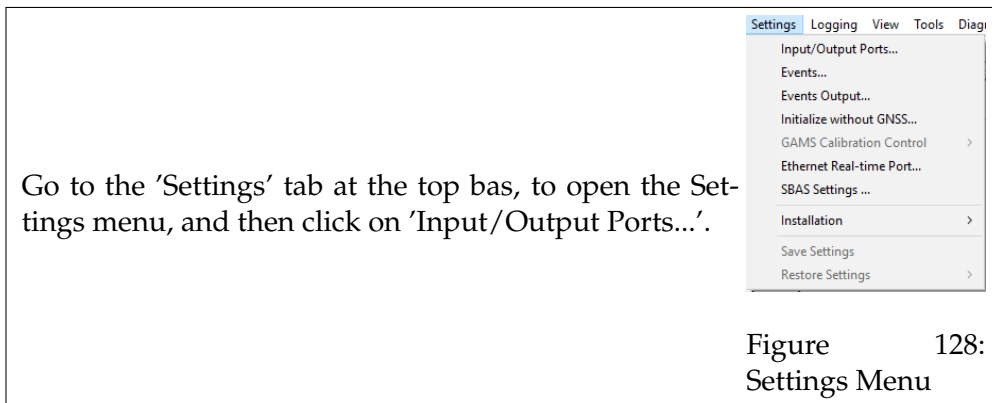


Once Applanix's address is entered in the application, the live data from the IMU will be shown.

Figure 127: LV POSview

Step 3

The next step is to set the 'Input/Output Ports...'



Go to the 'Settings' tab at the top bar, to open the Settings menu, and then click on 'Input/Output Ports...'.

Figure 128: Settings Menu

This will then open a window as seen in figure

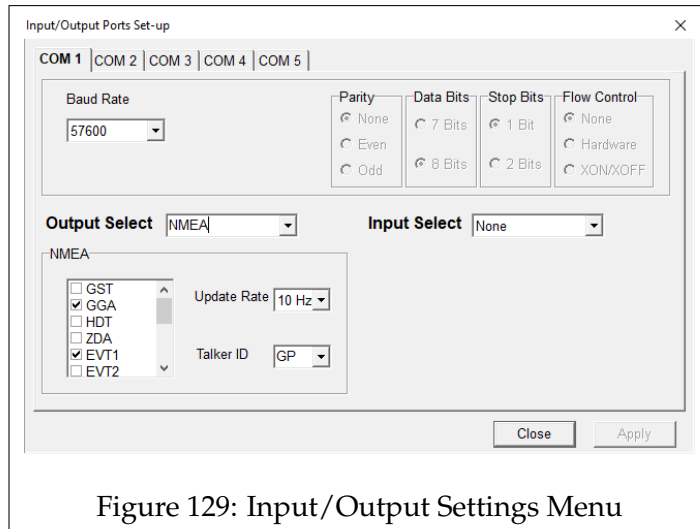


Figure 129: Input/Output Settings Menu

At the top of the window, there are 5 comports tabs, these correspond to the comports of your PC. You want to configure the settings for the used comport devices.

- **Baud Rate**, The Baud Rate defines the rate at which information is transferred. Set the rate to the same as set in the 'Baud Rate' property of the Recorder Builder pipeline. If the system drops frames, increase the Baud Rate.
- **Output Select**, Set the output to NMEA.
- Check the boxes for: 'GGA', 'EVT1', and 'PASHR'.
- **Update Rate**, Set the update rate to be about double of the frames per second, this will ensure that there is always data available when an image is made.
- **Talker ID**, Change this to 'GP'.

The Input/Output Settings are now set.

Step 4

Next, click on the Settings tab, and open the 'Events' window.

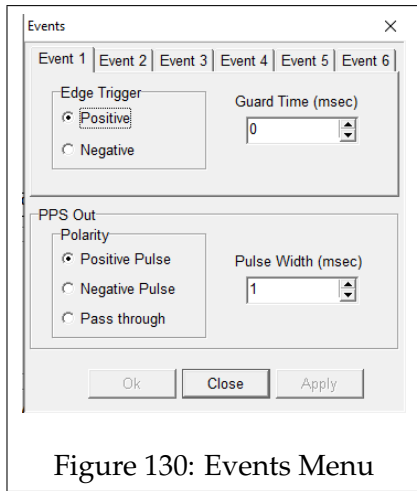


Figure 130: Events Menu

In this window set the 'Edge Trigger' option to either 'Positive', or 'Negative'. It does not matter which one you choose, as long as the trigger polarity of the camera is the same.

Open LadybugCapPro and check the 'Trigger Polarity', and ensure that both are set the same.

Step 5

Whilst you're in the Ladybug software, enable the 'Trigger Control' setting.

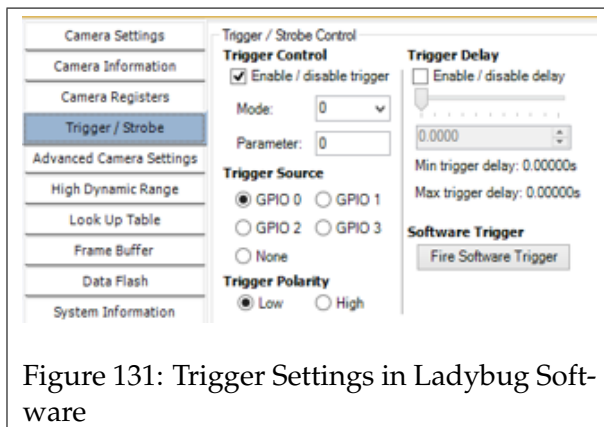


Figure 131: Trigger Settings in Ladybug Software

Step 6

Next, open the 'Events Output' menu.

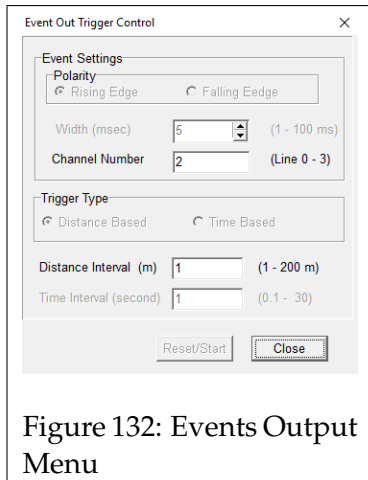


Figure 132: Events Output Menu

- **Channel Number**, Set this to 0.
- **Distance Interval**, Set the interval between each image. a value of 1 equals an interval of 0.5 meter.

Note that these properties do not get saved, so during each startup these two will need to be set again.

Step 7

This step involves setting the lever arms of the IMU.

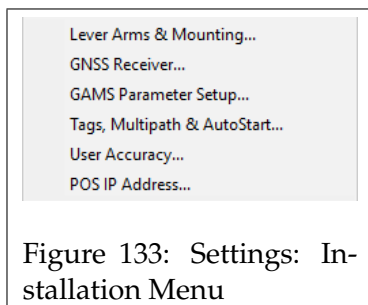


Figure 133: Settings: Installation Menu

Go back to the Settings menu, and then click on 'Installation', additional options will appear, click on Lever Arms & Mounting.

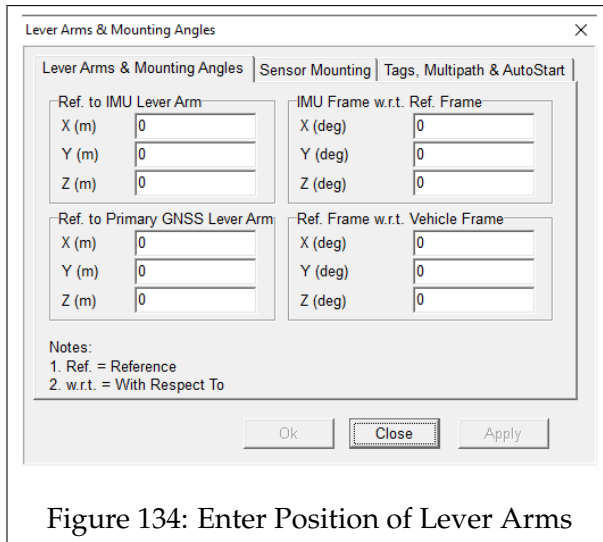


Figure 134: Enter Position of Lever Arms

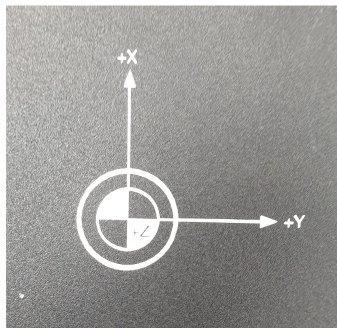


Figure 135: XYZ axis

Measure the distances, and rotation of the IMU to the reference point, and the primary antenna to the reference point. The Ladybug camera is generally the reference point.

- **X** Longitudinal axis, forward is positive.
- **Y** Lateral axis, right is positive.
- **Z** Vertical axis, down is positive.

- **Ref. to IMU Lever arm**

The location of the IMU from the reference point.

- **Ref. to Primary GNSS Lever Arm**

The location of the primary antenna from the reference point.

- **IMU Frame w.r.t. Ref Frame**

The rotation of the IMU in relation to the reference point.

- **Ref. Frame w.r.t. Vehicle Frame** The rotation of the reference in relation to the vehicle.

The Lever arms values can be adjusted at a later point, so they do not have to be exact. After entering the values, click on 'Apply'.

Step 8

In the Installation menu click on 'GAMS Parameter Setup'.

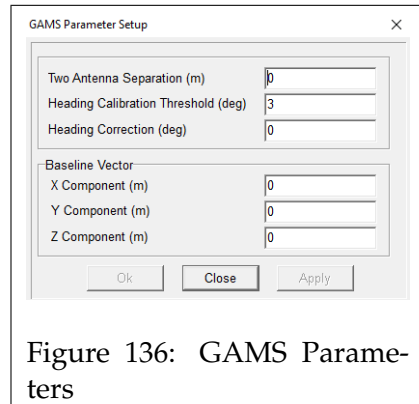


Figure 136: GAMS Parameters

In this window you can enter the location of the secondary antenna from the primary one. The second antenna is used to more accurately calculate heading.

Step 9

Go back to the Installation menu and open the 'User Parameter Accuracy' window.

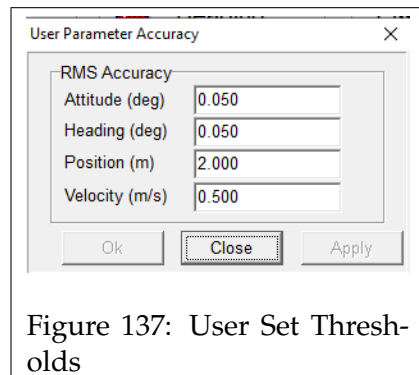


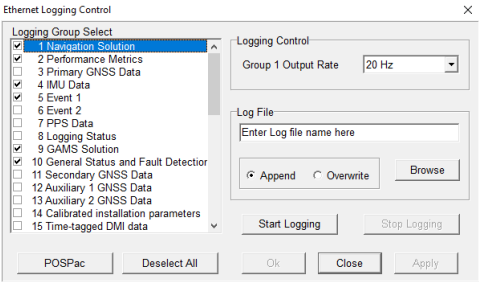
Figure 137: User Set Thresholds

In this window you can specify the threshold for when the 4 lights on the main panel turn green. The lower the value, the more accurate the solution needs to be, in order to turn green.

Step 10

At the top, click on 'Logging', two options should appear; one for logging via Ethernet, and the other for logging towards removable media. Choose your desired logging method.

Ethernet logging



- Set 'Group 1 Output Rate' to 20 Hz.
- Enter the 'Log File' name and save location.
- Check the following logging groups:
 - 1 Navigation Solution
 - 2 Performance Metrics
 - 4 IMU Data
 - 5 Event 1
 - 9 GAMS Solution
 - 10 General Status and Fault Detection

Figure 138: Ethernet Logging Window

Removable Logging Media

- Ensure that the removable media has enough free space.
- Set 'Group 1 Output Rate' to 20 Hz.
- Enter the 'Log File' name.
- Check the following logging groups:
 - 1 Navigation Solution
 - 2 Performance Metrics
 - 4 IMU Data
 - 5 Event 1
 - 9 GAMS Solution
 - 10 General Status and Fault Detection

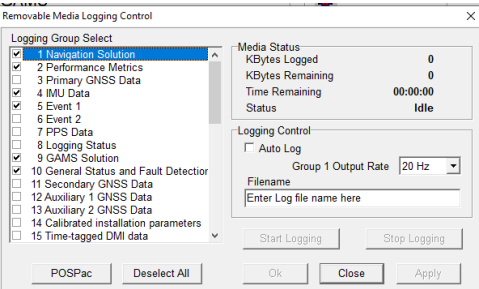


Figure 139: Removable Logging Media Window

Press 'Apply' and close the window

Step 11

Once you configured the settings, press 'Save Settings' in the Installation menu to save your adjustments.

Step 12

Start the Data Grabber and then the Recorder Builder.

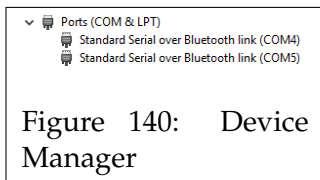
Step 13

Click on 'Connect' at the top left, and load the pipeline.
The pipeline should then be displayed on the graph.

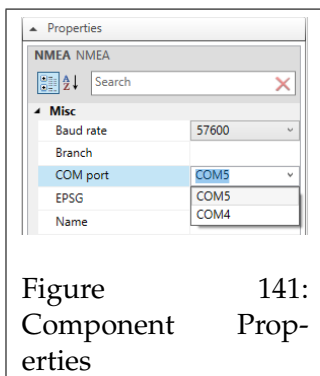
Step 14

The first step is to tell the system from which Comports it can grab the data for the 'Trigger' and 'NMEA' components.

Open Device Manager and expand the 'Ports' section. Find the entry for the Triggerbox and the Timerbox/NMEA and note down the Comport numbers.



Go back to the Recorder Builder, click on either the 'Trigger' or 'NMEA' component and on the left, scroll all the way down till you find the property window, then set the correct Comport number. Do this for both components.



Once the Comports are set, use the same USB ports. If you plug the Applanix or Trigger into another USB port, the Comport will change.

Also check the properties of the camera grabbers, and that they are in 'Triggered' mode

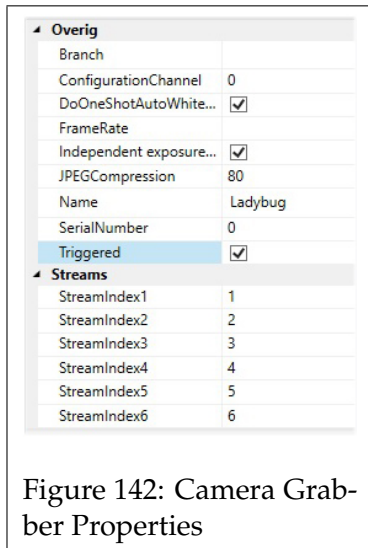


Figure 142: Camera Grabber Properties

Step 15

save the pipeline.

Step 16

Initialize the pipeline and then start it.

In the Data Grabber window, check that the system is running correctly.

Step 17

Start the Movie Recorder.

We want to add tabs such as the 'Arduino' to the interface, so they can be accessed.

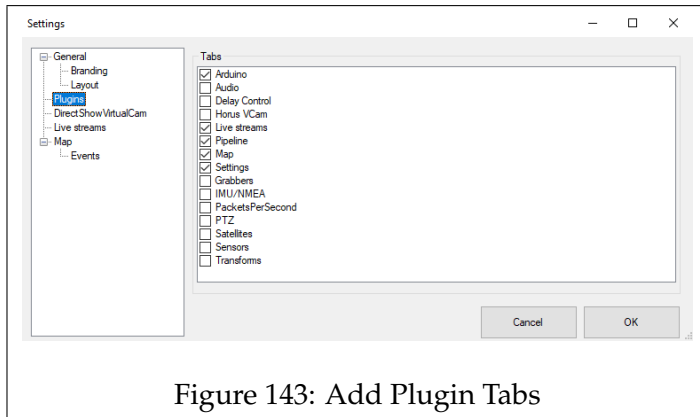


Figure 143: Add Plugin Tabs

We recommend to at least, enable the modules that are checked in figure 143.

Step 18

Open the 'Arduino' tab and set it to 'Free' mode, and set the 'Frequency' as desired. go to the 'Pipeline' tab and check that the data that is being transferred from the cameras/sensors is similar to the frequency set in the Arduino tab.

If a sensor does not transmit data, take the USB connector out, plug it back in, and restart the Pipeline. The device should now sent data.

Step 19

Set the Arduino to 'External' mode and set the 'Threshold'.

- If the Applanix is the trigger, set the threshold value to 1.
- If a CANbus Reader is the Trigger, the value will determine the interval. 1 = 0.5 meters.

Step 20

You may now check the system for proper functioning. Ensure that you have the Pipeline, Map and Livestream tab showing in the Movie Recorder, and also LV POSview on your screen.

Start driving around and monitor the system.

- In LV POSview the accuracy values should lower, and the lights should turn green.

- Check in the Movie Recorder that you're getting live video, and position updates in the Livestream, and Map tab respectively.
- Check the triggerbox for proper functioning, check in the Pipeline tab that you are getting data according to the set interval. The number of frames shown in the pipeline should correspond to the Trigger Count in LV POSview.

System is Ready

If the system functioned properly during step 19, then your system should now be ready for recording.

In the next chapter, you will learn, how to record from start to finish with the Citymapper system.

5.5 Recording with the Citymapper

5.5.1 How to Record

Now that the system is ready to go, you can start recording. This chapter will describe step by step the process of recording data with the Citymapper system.

Step 1. Starting and checking the System

Power on the system, start the computer and the Applanix.

Start the Data Grabber, the Recorder Builder, and then the Movie Recorder, also start LV POSview.

In the Recorder Builder, connect, then load the pipeline, and then start running it.

Starting the pipeline can also be accomplished via the included .BAT file.

In LV POSview, connect to the Applanix, and see if you're getting data from it.

In the Movie Recorder check that the pipeline is functioning correctly. Once you're done checking, set the 'mode' to 'External'.

Step 2. Final Configuration

Each time you start the system, or recording, there are a few settings that you want to adjust.

Movie Recorder

- Set the 'Threshold' interval to the desired distance if you're using an OBD Reader Unit.
- In the 'Settings' tab, define the Recording name, and path of the recording.

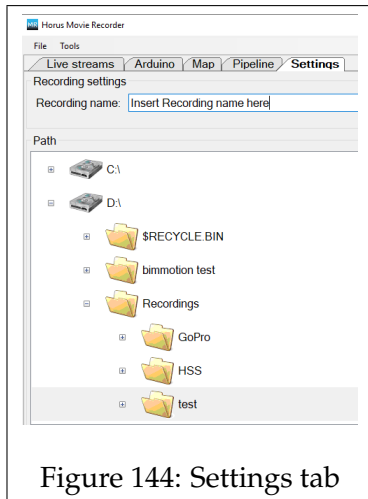


Figure 144: Settings tab

LV POSview

- In the 'Events Output' window, set the 'Channel Number' to 0.
- If the Applanix is used as the trigger, set the Distance Interval here.

Step 3. initializing the IMU

The IMU requires alignment, this can simply be done by driving in a straight line at a reasonable pace, you should see the 'accuracy' values drop once you are driving. Keep driving till you have 4 green lights, indicating that the alignment is good.

For this reason we recommend starting LS POSview prior to recording/logging, to give it time to initialize.

Step 4. Starting the Log

Drive to a clear, open area near the location where you wish to record. Go to 'Logging' and select your method of logging. In the window that opens, press 'Start Logging'.

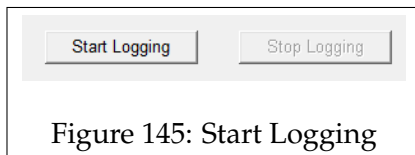


Figure 145: Start Logging

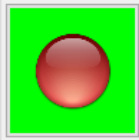
Once the log starts, stand still for another minute before you start moving, this will ensure that the Applanix has a good GPS fix to start with.

If you're making multiple recordings, we recommend letting the log run for both recordings, otherwise you will have to find the correct log file for a specific recording, instead of being able to use a single log file for all the recordings made that day.

Whilst the log is running, refrain from stopping in a secluded area, instead park in a clear open area.

Step 5. Recording

Drive to the location where you wish to start the recording.



Press the record button in the Movie Recorder to start recording data.

Figure 146:
Record
Button

Whilst recording, monitor the accuracy and the trigger count in LV POSview. Also regularly check the pipeline and map to ensure that you are still getting data.

If you're recording in a city or otherwise secluded area, we advise you to move towards a clear area in between the recordings, this will ensure that each recording starts with a solid GPS fix.

We advise against recording for an hour or longer, instead stop, and restart the recording.



Once you are done recording, simply stop it by hitting the stop button.

Figure 147:
Pause and
Stop Button

Step 6. Stopping the Log

Once you have finished recording for the day, move to an open, and clear area and come to a standstill. Let the system log for another minute before

stopping the log, the log can be stopped by clicking on the 'Stop Logging' button as seen in figure 145.

Exporting the Recorded Data for post-processing

After recording the data, the recordings are stored on the device selected in the Movie Recorder, and the log file can be found on the PC, or on the USB stick of the Applanix depending on your logging method used in LV POSview.

The recorded data requires some post-processing before it is usable, this process is described in the following chapter. This can either be done on a dedicated processing PC, or on the Citymapper laptop, if the required software is installed. In either case, the computer should have access to both the Recording, and the Log data.

5.6 Processing

5.6.1 Introduction: Processing

After recording, you have the imagery from the cameras, and the log file from the Applanix.

The data from the Applanix requires some post-processing to improve the accuracy, and needs to be synchronized with the images.

The processing consists of the following main steps:

- Post-processing the GPS data in POSPac MMS
- re-attaching the processed data to the recorded imagery with the Position Fixer
- Configuring the video streams in the Movie Player

5.6.2 Processing: Importing Data

Before the post-processing process can start, the recorded data from the Applanix and the Movie Recorder needs to be moved to the system used for the post-processing.

Images from the Movie Recorder

The recorded images can be found in the folder that was selected in the Movie Recorder.

The recording folder are also given a date and sequence number after the given name.

Log file from the Applanix

Depending on what logging mode you used the log file can be found in the following two places:

- **Ethernet Logging...**
The Log file can be found in the specified directory.
- **Removable Logging Media...**
The log file can be found on the USB stick inserted into the Applanix.
Note: the Applanix USB stick must only be formatted by a Linux system.

- **Back up**

The Applanix also writes the log to its internal memory as a backup. This log can be accessed by connecting to it via an FTP client such as 'FileZilla'

Folder Structure

We recommend placing all data from same recording into a single directory.

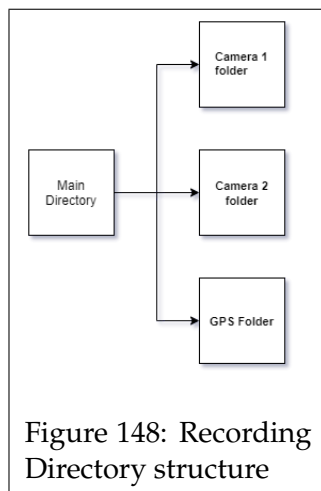


Figure 148: Recording Directory structure

- If you only have a single recording directory containing the data from all the cameras, then create a single 'recording' folder and copy the recording directory to it.
If you have multiple recording directories for multiple cameras, then create folders for each camera. (Name the folders appropriately; Ladybug as '360', and LIDAR as 'LIDAR' for example.)
- Move the log file from the Applanix into the 'GPS' folder.

Now that your files are in the right place, you can start post-processing.

5.6.3 Processing: POSPac

The first step is to process the raw data from the Applanix in POSPac MMS.

Step 1

Start POSPac MMS.

Step 2

Click on 'New Default Project' to start a new project.

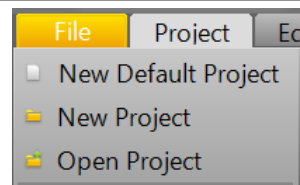
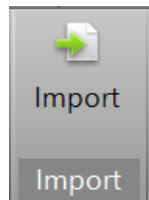


Figure 149: Start New Default Project

Step 3

Import the log file from the Applanix.



Click on the 'Import' to open the import menu.

Figure 150:
Import
Data

Enter the path to the directory containing the GPS data.

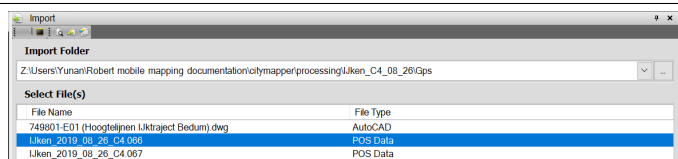


Figure 151: Import Menu

Select the log file, click on Import, and wait for the data to be imported. Once the data has been imported, the trajectory of the data will be shown on the graph.

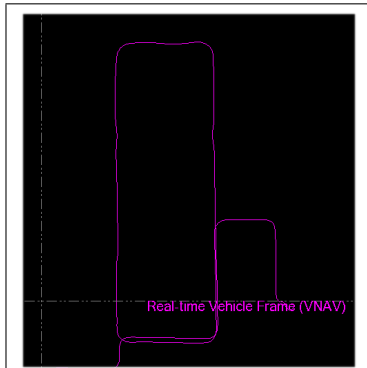
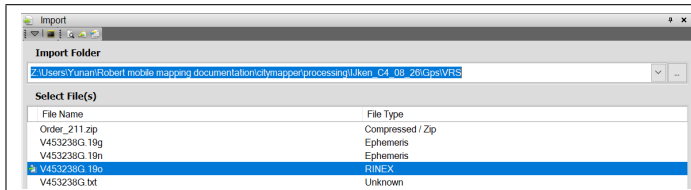


Figure 152: GPS Trajectory on the Graph

Step 4

The next step is to add a base station, it will be used to reference and compare the recorded data to. Using a base station will thus increase the accuracy of the processed data. It is not required to use a base station, but the accuracy of the data will be degraded.

Get VRS or CORS data from providers such as Trimble, ensure the data starts prior to the the start time of your recording and ends after you stopped your recording. Extract the downloaded data to a directory. (We recommend creating a folder inside the directory with the GPS data.)



Enter the path to the directory containing the base station data.

Figure 153: Import Base Station Data

Select the 'RINEX' file and import it. Once the data has been imported, you can find it on the graph.

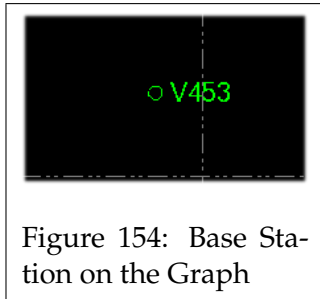


Figure 154: Base Station on the Graph

Step 5

Now that the base station has been added to the graph, we need to tell the system how to interpret the data from the base station data.

Right click on the base station to open a menu.

Select the 'Properties' at the bottom to get the base station its properties on the right side of the screen.

To continue with this step, click on 'Coordinate Manager'.

Figure 155: Import Menu

In the Coordinate Manager, select the 'Frame' tab. Set the 'Frame' property to the coordinate system/format that is used in the area of the recording/base station.

Figure 156: Coordinate Manager

Step 6

Right click on the base station icon and in the menu from figure 155 click on 'Set Base Station'.

Once the process has finished, a window will appear showing you the quality of the data.

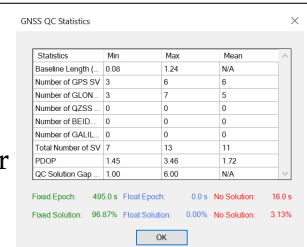


Figure 157: GNSS QC Statistics



The base station icon should now have a flag.

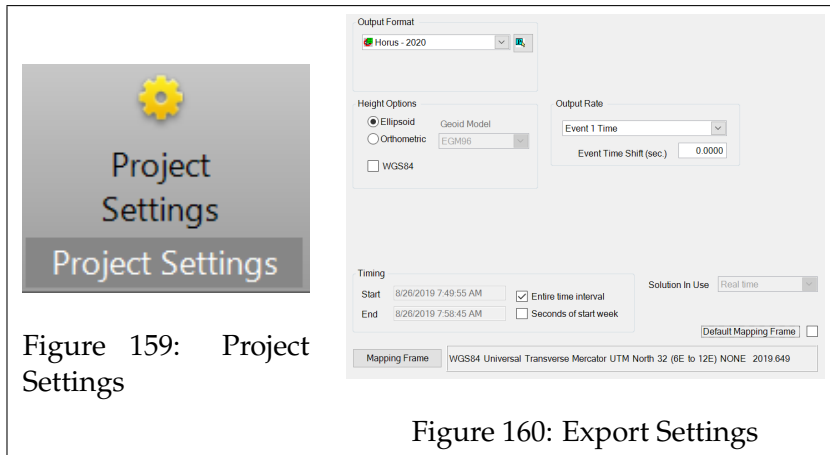
Figure 158: Base station icon after conversion

Step 7

The next step is to set the 'Project Settings'.

Click on 'Project Settings' to open the settings menu, in this menu you will configure the export settings, and set the 'Lever arms and Mounting angles' values.

First, we'll set the 'Export settings', so go to 'Exports' -> 'Settings'. This will bring you to the menu as seen in figure 160.

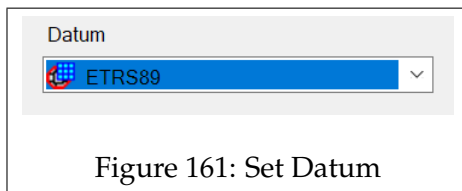


In the menu select the following settings:

1. Set the output format to our 'Horus Format'.
If the format is not in the list, then you will need to add it.
This can be done by placing the '<filename>.xsd' file in the following directory:

C:\ProgramData\Applanix\User Format Profiles

2. Set the 'Output Rate' to 'Event 1 Time'.
3. Uncheck the 'Default Mapping frame' box', and then click on the 'Mapping Frame Button'.



In the window that opens, set the 'Datum' to the coordinate system of the area you recorded in.

The next step is to enter the *lever arm* values.
These values can be entered by going to: 'GNSS-Inertial Processor' -> 'Lever Arms and Mounting Angles'.

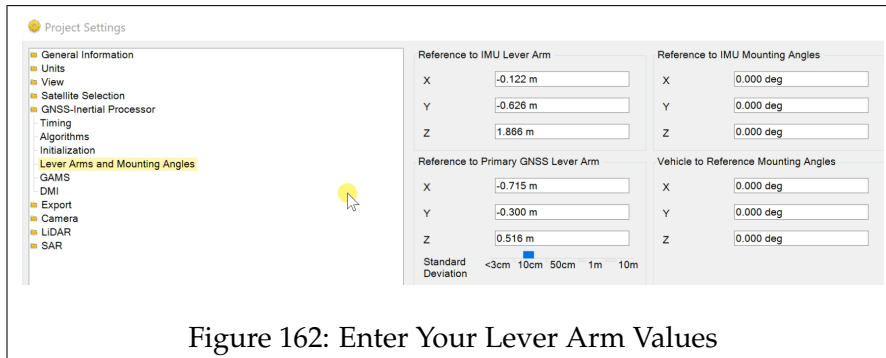


Figure 162: Enter Your Lever Arm Values

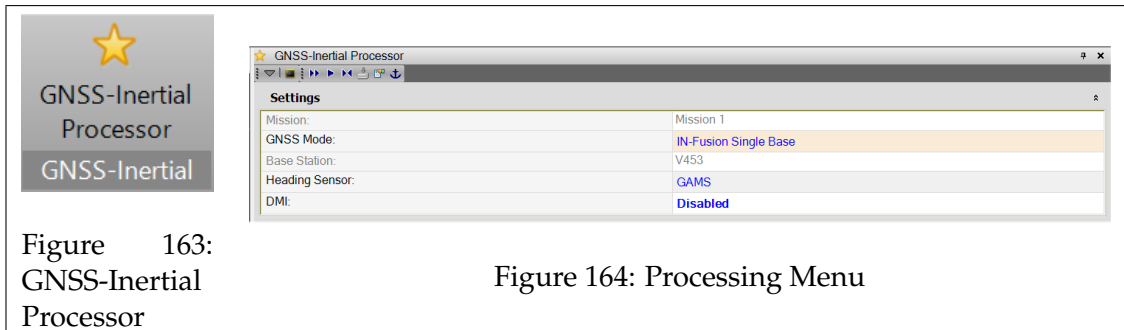
- Reference to IMU Lever Arm**
 The position of the IMU in relation to the reference. (If the IMU is the reference; enter 0,0,0)
- Reference to Primary GNSS Lever Arm**
 The position of the primary GNSS antenna in relation to the reference.
- Reference to IMU Mounting Angles**
 The rotation of the IMU in relation to the reference.
- Vehicle to Reference Mounting Angles**
 The rotation of the reference in relation to the vehicle.

If you have a second GNSS antenna, then you can also enter the GAMS values in the 'GAMS' menu. by comparing the position of the first and second antennas, the heading can be calculated.

The values from for the lever arms and GAMS, should be the same as the ones set in LV POSview. These can be adjusted later on.

Step 8

The system is now almost ready to start processing. Click on the 'GNSS-Inertial Processor' button to open the processing menu.



The processing menu contains a number of settings, ensure that these are set correctly.

GNSS-Inertial Processor Settings

- **GNSS Mode:** Sets the processing mode.
If using a single base station use: 'IN-Fusion Single Base'.
When using no base stations use: 'IN-Fusion Autonomous'.
- **Heading Sensor:** Set to 'GAMS' if you have a second antenna.
- **DMI:** *Distance Measurement Indicator*. Enable if you have a wheel speed sensor.

Once the settings have been configured, press the desired processing button at the top of the menu.

The software will then start processing the data, wait till the process has finished. Once finished a green line should be drawn onto the graph, showing the difference between the raw and processed data.

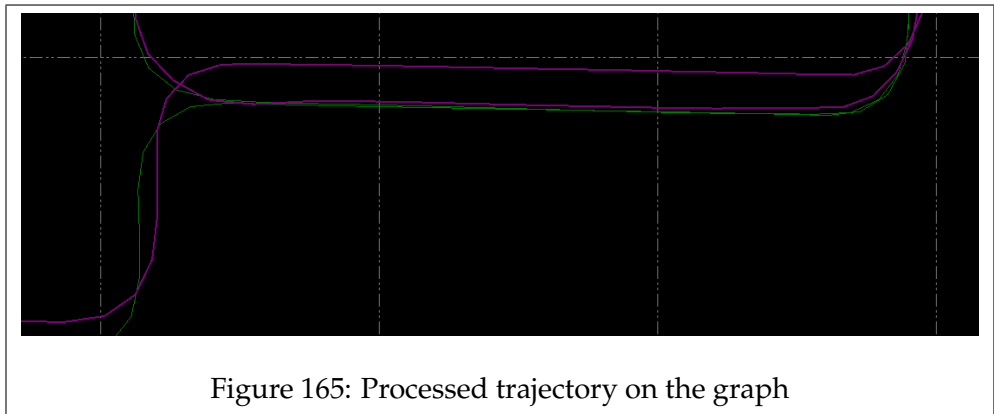


Figure 165: Processed trajectory on the graph

Step 9

The final step is to export the processed data.

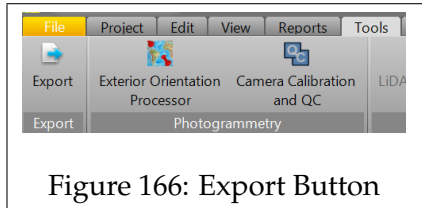


Figure 166: Export Button

Open the Export menu which can be found under 'Tools'.

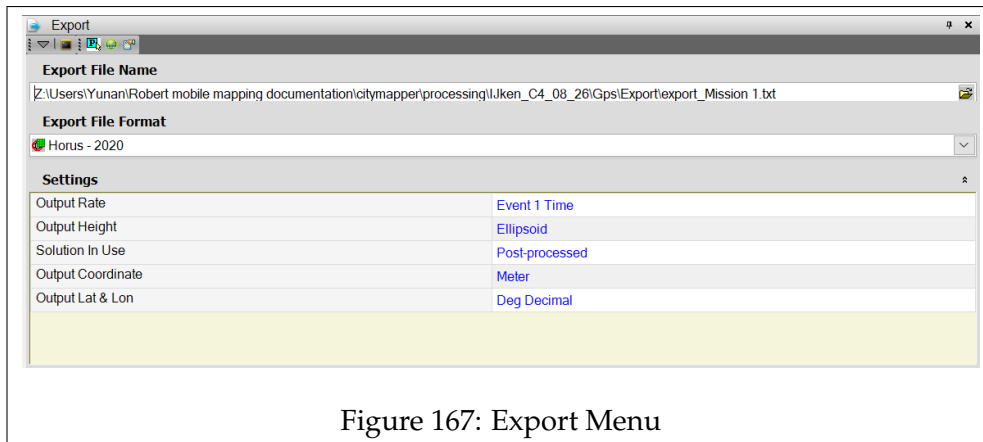


Figure 167: Export Menu

In the Export Menu you can define the directory where the data is exported to.

Ensure that the 'Export File Format' is set to the Horus format, and that the 'Output Rate' is set to 'Event 1 Time'.

Press 'Export' to start exporting the data.

The exported data will be stored in a '.txt' file. The first line of this file shows the coordinate system that is used, check that this is the correct format.

The positional data from the Applanix has now been processed, the next step is to link it to the imagery.

5.6.4 Processing: Position Fixer

The next step in the process is to attach the positional data from the Applanix to the recordings

- Start the Horus Position Fixer software.

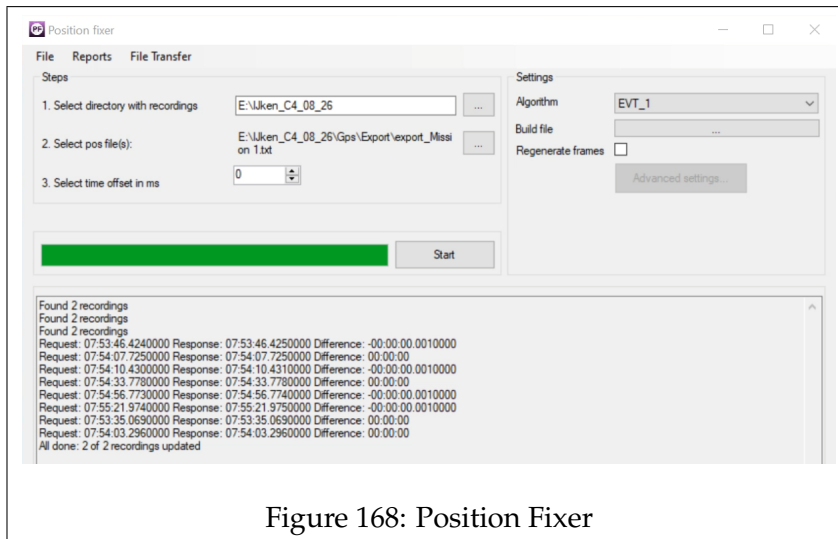


Figure 168: Position Fixer

- Add Paths

Select directory with recordings: Enter the path to the directory containing the recordings.

Select POS file(s): Select the pos file that you exported from POSpac MMS.

Regenerating Frames

In some cases, you may want to update your frames file, you can check the 'Regenerate frames' box to renew the frames file.

Alternatively you can also delete the 'frames.xml' file from the recording directory, upon starting the Movie Player a new one will be generated.

When you have added the paths to the recording and pos file, press 'Start', and wait for the process to finish. Once finished, the processed data from the Applanix is attached to the recordings.

5.6.5 Processing: Movie Player

The recorded data is now ready to be viewed in the Horus Movie Player.

Loading in Recordings

You can load recordings in two ways.

1. Click on one of the '.idx' files in your recording directory, this will start the Movie Player, and load the recordings from that folder.
2. Start the Movie Player and load any '.idx' file from the recording directory, by drag and dropping it on the Movie Player, or selecting it through the 'Open file...' option.

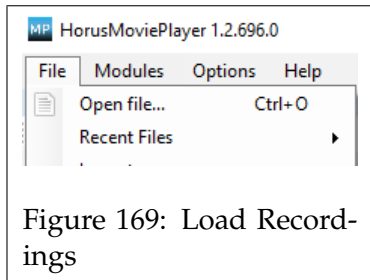


Figure 169: Load Recordings

It does not matter which '.idx' you load, the Movie Player will load in all the data from that folder, provided they are called for in the 'setup.hsf'.

Positioning the video streams

Now that we can view the images from all cameras, we may want to adjust the position, color, etc, of the video streams.

To position each individual video stream, we need the 'Immersive View Builder' module, which can be added to the GUI by going to the Modules tab as seen in figure 170.

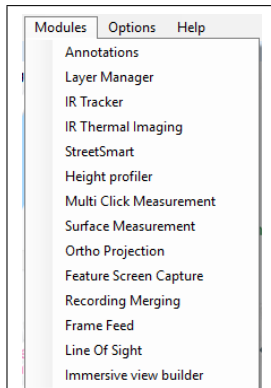


Figure 170: Movie Player Modules Menu

The 'Immersive View Builder' module should be added on the left side of the screen.

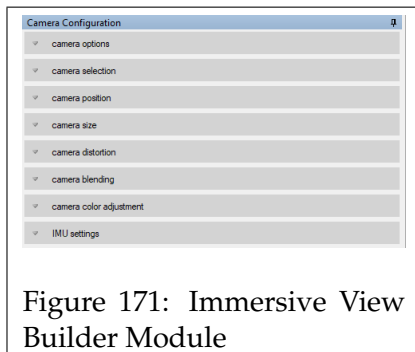


Figure 171: Immersive View Builder Module

Under the '*camera selection*' tab you can choose which video stream you want to edit.

In the '*Camera position/size*' tabs you will find controls to move and resize the image of the selected stream.

To crop and blend the image, use the settings in the '*camera blending*' tab.

To correct the colors from the selected stream, use the '*camera color adjustment*' controls.

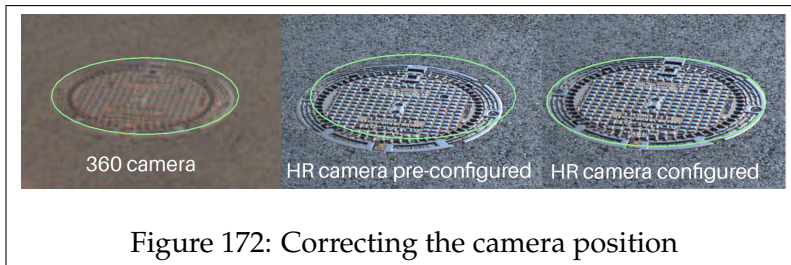


Figure 172: Correcting the camera position

Move the video streams in such a way that they are correctly overlapped, you can make the overlapping view transparent, by adjusting the opacity under the camera blending tab.

The exact position might change depending on the distance of the objects that you are viewing, so set it to the average distance that you wish to view.

Click on the gear icon at the top right to open the 'Settings' menu. Set the camera height, by adjusting the 'cam h' value to the height of the camera's center in meters.

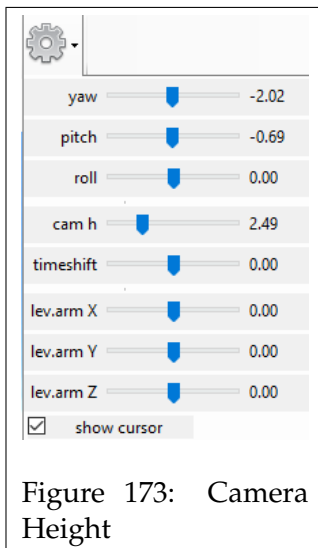


Figure 173: Camera Height

6 ASCII PTZ Protocol

6.1 Horus View & Explore ASCII PTZ Protocol

6.1.1 Introduction: ASCII PTZ Protocol

The ASCII protocol was designed to help software engineers in testing PTZ controllable domes.

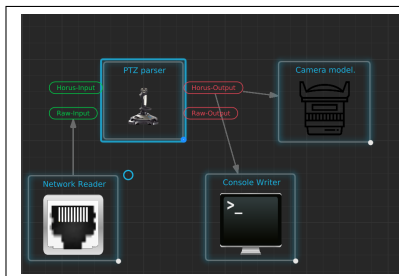
Often when working on site or at home, one needed to generate joystick or button commands in order to engage a certain behavior. In most cases no physical device was available, and since few people can write binary protocols from the top of their heads a human readable format was needed. Thus we created a small and easy to remember protocol allowing simple commands such as:

```
LEFT 0.1
OFFSET ABSOLUTE_OFFSET UNITS RADIANS
YAW 10
UNITS NORMALIZED
UP -1
OFFSET ABSOLUTE_OFFSET UNITS RADIANS PITCH 1
```

6.1.2 In practice: connectivity

There are several ways to insert and extract PTZ commands within the Horus environment. The two that we will focus on are serial and network connectivity.

6.1.3 Network connectivity



Connecting a **Network Reader** { Protocol: TCP, Behavior: Read } to a **Ptz parser** {incoming protocol: ASCII} allows for ASCII/PTZ over Network communication.

Using simple commandline utilities such as netcat allows messages to be send to the Network Reader.

```
$>> nc -4 localhost 5567
OFFSET ABSOLUTE_OFFSET UNITS RADIANS
YAW 10
PITCH 3
```

6.1.4 Serial connectivity

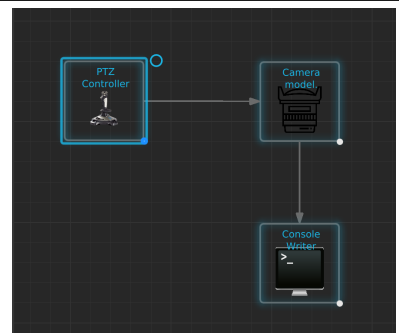
In order to use serial communication as a testing input, virtual serial port pairs needs to be configured.

On windows platforms, the <http://com0com.sourceforge.net> can be used.

For Linux users we recommend the *socat* utility (make sure you have the correct rights or add the user to the "dialout" group).

```
$>> socat PTY,link=/dev/ttyS98 PTY,link=/dev/ttyS99
```

Setting up a **PTZ controller** { incoming protocol: ASCII, outgoing protocol: ASCII } allows for ASCII/PTZ over serial communication.



6.2 The protocol parser.

As we have seen in the connectivity section, the ASCII protocol parser is embedded in several different components. The ASCII protocol parser is implemented as a state-machine, meaning that the following statement has no direct effect:

```
OFFSET ABSOLUTE_OFFSET UNITS RADIANS
OFFSET ABSOLUTE_OFFSET UNITS RADIANS
```

It does make sense however to start your session by providing:

```
OFFSET ABSOLUTE_OFFSET UNITS RADIANS
```

after which all orientation based commands will be of the later configuration. After which commands such as:

```
YAW 180
PITCH 10
ZOOM 45
```

can be issued. The latter statements will be interpreted as:

```
OFFSET ABSOLUTE_OFFSET UNITS RADIANS YAW 0.2
OFFSET ABSOLUTE_OFFSET UNITS RADIANS PITCH 0.8
OFFSET ABSOLUTE_OFFSET UNITS RADIANS ZOOM 0.25
```

6.3 The protocol definition.

Setting the camera offset.		
OFFSET	RELATIVE_OFFSET	ABSOLUTE_OFFSET

Setting the units.			
UNITS	RADIANS	DEGREES	NORMALIZED

Orientation commands, example LEFT 0.1			
LEFT	RIGHT	UP	DOWN
YAW	PITCH	ROLL	ZOOM_TELE
ZOOM_WIDE	ZOOM	FOCUS_FAR	FOCUS_NEAR
FOCUS			

The stop command
STOP

Best practice combinations		
TOKEN	OFFSET	UNITS
LEFT	RELATIVE_OFFSET	RADIANS
RIGHT	RELATIVE_OFFSET	RADIANS
UP	RELATIVE_OFFSET	RADIANS
DOWN	RELATIVE_OFFSET	RADIANS
YAW	ABSOLUTE_OFFSET	RADIANS
PITCH	ABSOLUTE_OFFSET	RADIANS
ROLL	ABSOLUTE_OFFSET	RADIANS
ZOOM_TELE	RELATIVE_OFFSET	RADIANS
ZOOM_WIDE	RELATIVE_OFFSET	RADIANS
ZOOM	ABSOLUTE_OFFSET	RADIANS
FOCUS_FAR	RELATIVE_OFFSET	RADIANS
FOCUS_NEAR	RELATIVE_OFFSET	RADIANS
FOCUS	ABSOLUTE_OFFSET	RADIANS

7 ASCII JSON Sensor Protocol

7.1 Horus View & Explore ASCII Sensor Protocol

7.1.1 Introduction: ASCII Sensor Protocol

The Sensor protocol is an easy protocol to understand, remember and implement. Its strength is therfor shown during implementation and as commandline interface. Since it is implemented as an state machine only changes in value, name and structure need to be addressed.

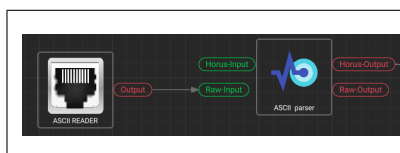
The following listing shows a series of easy to understand commands

```
NAME temprature VALUE 0.1
VALUE 0.2
VALUE 0.3
STRUCTURE VECTOR
DIM_START 3 DIM_STOP
VALUES 1 2 3
STRUCTURE MATRIX
DIM_START 2 3 DIM_STOP
VALUES 1 2 3 4 5 6
```

7.1.2 In practice: connectivity

There are several ways to insert sensor messages within the Horus environment. The one that we will focus on is network connectivity.

7.1.3 Network connectivity



Connecting a **Network Reader** { Protocol: TCP, Behavior: Read } to a **Sensor parser** {incomming protocol: ASCII} allows for ASCII-I/Sensor over Network communication.

Using simple commandline utilities such as netcat allows messages to be send to the Network Reader.

```
$>> nc -4 localhost 5567
NAME Pressure VALUE 3
VALUE 3.5
VALUE 3
```

7.1.4 The protocol parser.

As we have seen in the connectivity section, the ASCII protocol parser is easily embedded within a system v2 system. The ASCII protocol parser is implemented as a state-machine, meaning that the following statements will not generate a sensor message:

```
STRUCTURE VECTOR
DIM_START 3 DIM_STOP
STRUCTURE VECTOR
DIM_START 3 DIM_STOP
```

nor do the last two commands change the state of the parser. It does make sense however to start your session by providing:

```
NAME mySensor STRUCTURE VECTOR DIM_START 3 DIM_STOP DATATYPE UINT
```

after which all generated sensor message will be vectors of unsigned integers with length 3. After which commands such as:

```
VALUE 1 2 3
VALUES 3 4 5
VALUES 0 1 2
```

can be issued. Resulting in the following vectors: $[1 \ 2 \ 3]$, $[3 \ 4 \ 5]$, $[0 \ 1 \ 2]$.

7.1.5 The protocol definition.

TOKEN	ARGUMENT(S)
NAME	literal with no spaces
UNITS	DEGREES, RADIANS, NORMALIZED, UNKNOWN_UNITS
STRUCTURE	SCALER, VECTOR, MATRIX, COMPLEX
DATATYPE	STRING, INT, DOUBLE, FLOAT, LONG, UINT, ULONG
VALUE	list of values, that corresponds to the dimensions.
VALUES	list of values, that corresponds to the dimensions.
DIM_START	list of values, where each entry corresponds to one dimension.
DIM_STOP	
RESET	reset parser in default state.

DEFAULT STATE	
TOKEN	ARGUMENT(S)
NAME	unknown
UNITS	UNKNOWN_UNITS
STRUCTURE	SCALER
DATATYPE	DOUBLE

8 Converting GoPro Recordings to the Horus Format

8.1 Introduction

Interested in Mobile Mapping, but not quite ready to invest in expensive hardware?

Horus has a low-cost solution, using consumer grade 360 cameras such as the GoPro Fusion/Max, or the Garmin Virb. These cameras can still provide detailed 360 degrees recordings and positional data, and are perfect for getting started with mobile mapping, at a much lower price.

The recordings from the GoPro or Garmin are not natively supported by Horus, but through the use of a plugin, these recordings can be converted to become usable in the Horus environment.

This document will describe the process of converting these recordings, for use with Horus software.

8.2 Converting GoPro Recordings for use in the Horus Environment

8.2.1 Converting footage for Horus software

The Horus software does not support the recordings from the GoPro natively, so it is necessary to convert them. This is a 2-step process; the footage from the GoPro needs to first be exported through GoPro software, before being converted into the Horus Format via the 'Horus Movie Maker'.

Recording with a GoPro

To ensure that the recordings are usable for mobile mapping ensure that you have the following settings selected:

1. Recording Mode

- **GoPro Max**

Check that you are in '360 Video' mode indicated by the icon indicated by the blue circle in figure 174. This mode is required to capture 360 degree video, the normal video or 'Hero' mode will only record from a single camera.

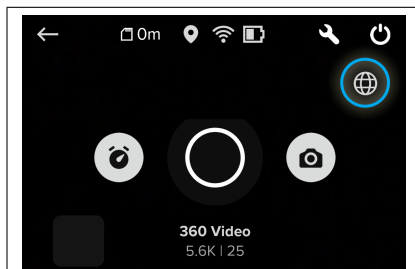


Figure 174: GoPro Viewfinder

- **GoPro Fusion**

The Fusion doesn't have different modes for video recording, so simply select the 'Video' mode.

2. GPS Data

To record GPS data during your recordings, ensure that the GPS setting is set to on.

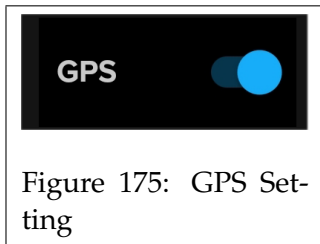


Figure 175: GPS Setting

Exporting Recordings

The recordings from the GoPro are saved in the '360' file format. For our purposes we need an 'MP4' file. Fortunately, GoPro offers software to export the '360' recordings as an 'MP4'. The software that is used differs for either camera. Below, the method for exporting video from either camera is described.

- **GoPro Max**

To export recordings from the GoPro Max, the 'GoPro Max Exporter' software is required.

Install the software, and run it, A window similar to figure 176 should then open.

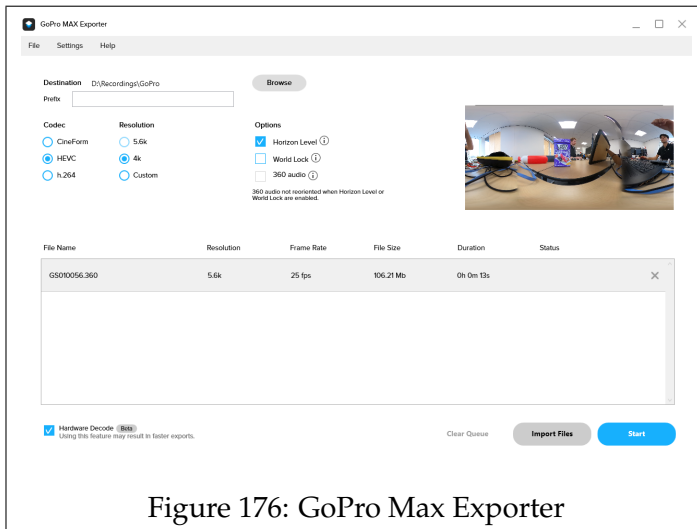


Figure 176: GoPro Max Exporter

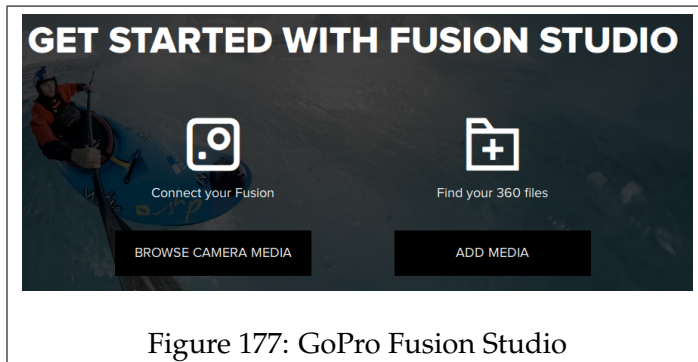
1. Add Recordings by dragging the .360 file from your GoPro's SD card into the box at the bottom of the window.
2. Specify the directory where the exported recordings will be saved to.
3. Select either 'HEVC' or 'h.264', set the resolution to 4K, and check the 'Horizon Level' option.
Ensure that the 'World Lock' option is disabled.
4. Press the 'Start' button to export the recording.

You can find the exported recordings as an MP4 file in the specified directory.

- **GoPro Fusion**

To export recordings from the GoPro Fusion, the 'GoPro Fusion Studio' software is required.

Install the software, run it, and a window similar to figure 177 should open.



If you have the Fusion connected to your PC via USB, then click on 'Browse Camera Media'.

If you moved the recordings from both SD cards to a local directory on your PC, then click on 'Add Media'.

Once the Recordings have been loaded, you will be directed to the window seen in figure 178.

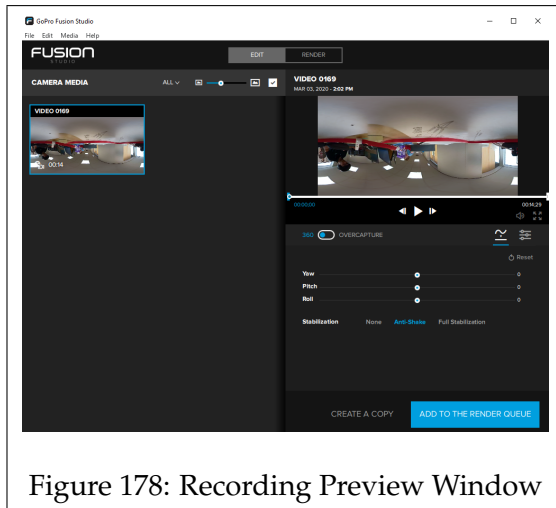


Figure 178: Recording Preview Window

1. Click on the recording that you wish to export.
2. Set the 'Stabilization' to 'Anti-Shake', and click on 'Add To The Render Queue'.

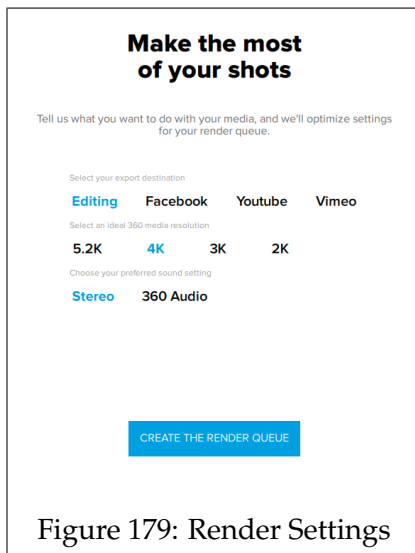


Figure 179: Render Settings

3. A window similar to figure 179 will open. Set the export destination as 'Editing', the output resolution to '4K', and then press 'Create the Render Queue'.
4. Click on 'Render All' to start exporting the recordings.

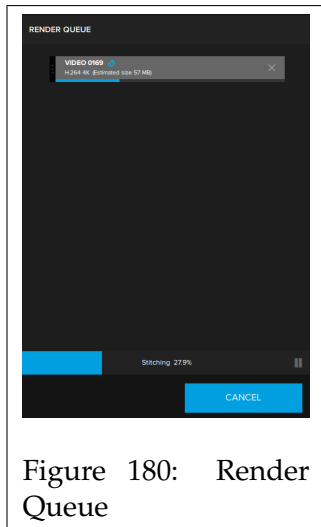


Figure 180: Render Queue

Once the render is complete you can find the recordings in the 'Export Location' as an MP4 file.

Default Export Location

C:\Users\<<Username>\Pictures\GoPro Fusion\Rendered

Checking Recording Telemetry

You can check if the recording contains GPS data, or see if the data is usable by using the GoPro Telemetry Checker.

Converting to Horus Format with the Movie Maker

Now that you have exported the recordings as an MP4, the next step is to convert these into the Horus format.

To convert the recording, you will require the 'Horus Movie Maker'.

Below, you will find step-by-step instructions on how to convert the recordings.

1. Start up the Movie Maker software.
2. Click on 'File' and then 'Import Data'.

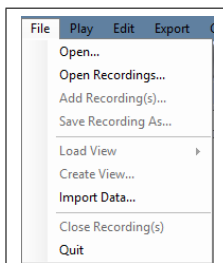


Figure 181:
Importing
Data in Movie
Maker

3. Select 'Video with Metadata' from the list, and press 'OK'.

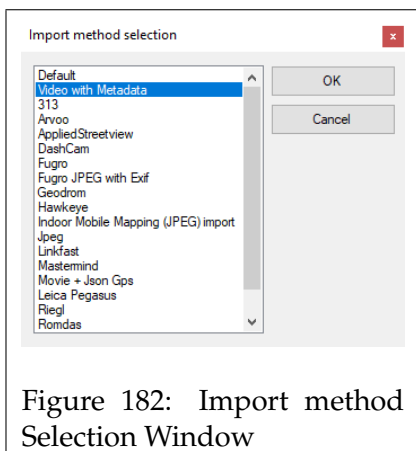


Figure 182: Import method
Selection Window

4. The 'Video with Metadata Settings' window will open.
 - **Type**, Select 'GoPro' or 'Garmin' depending on your camera.
 - **Video**, enter the file path, or browse and select the exported MP4 video that you wish to convert.
 - **Metadata**, enter the file location of the original '360' file.
 - **Interval**, the time between each frame. An interval of 0.1 would result in 10 frames per second. An interval of 0 will export all frames of the recording.
 - **equirectangular**, Enable this setting if you have a 360 degrees recording.

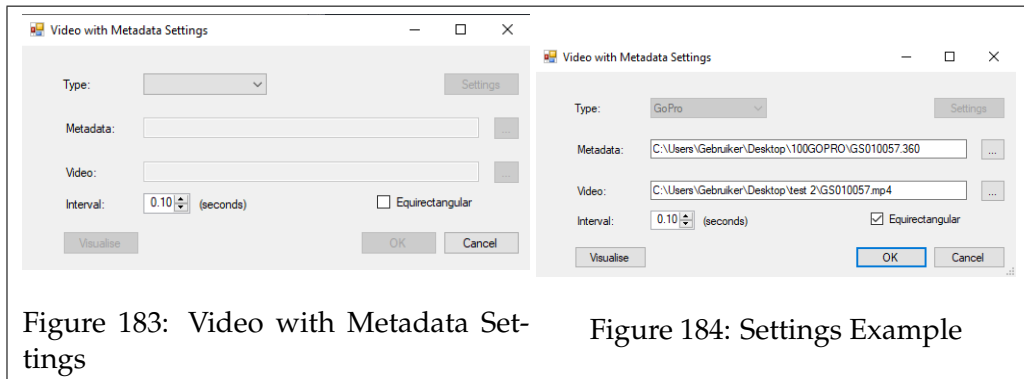


Figure 183: Video with Metadata Settings

Figure 184: Settings Example

- Fill in the requested settings and press 'OK' to continue.

You can find the recording in the Horus format in the same folder as the original 'MP4' file.

Initial Setup for GoPro Recordings in the Movie Player

The recording has now been converted into an '.idx' file, and can now be opened in the Movie Player.

Before you can start geotagging, you need to configure the camera setup. This involves setting the height, roll, pitch, and yaw. Please use the following steps to make this setup file.

1. Turn the button 'Show alignment axis' to on. This makes it easier to visualise your camera configuration.

2. Open the 'Settings' menu by clicking on the gear icon at the top right.

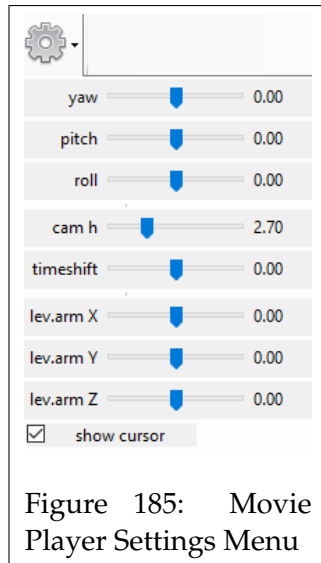


Figure 185: Movie Player Settings Menu

3. Set the camera height (cam h) in meters.
4. Adjust the roll, pitch, and yaw parameters. The redline in the screen should match the horizon.
5. Measure an object on flat ground with a known size to check if the settings are correct, adjust the parameters in the Settings sliders till they are correct.
6. Use the layermanager to draw line geometry (e.g. a curbstone with a known length) and turn on the label of this line, that shows 'Length'.
7. Check on the map if the field of view is pointing at the right direction. If not, adjust the yaw settings (maybe 180 degrees)
8. Once all the settings seem correct, go to: menu -> File -> Save View (*.hsf), and name it 'setup.hsf'. Store/copy this file in each recording folder with the same camera setup.

After configuring the camera setup you should be ready to start geotagging the recording.

Tutorial Videos

Prefer watching a video?

Converting the GoPro Recordings

Geotagging GoPro Recordings

9 Horus Webservices

9.1 Horus Webservices: System Architecture

9.1.1 System Architecture

The system architecture of the Horus Webservices can be divided into three main sections as seen in figure 186

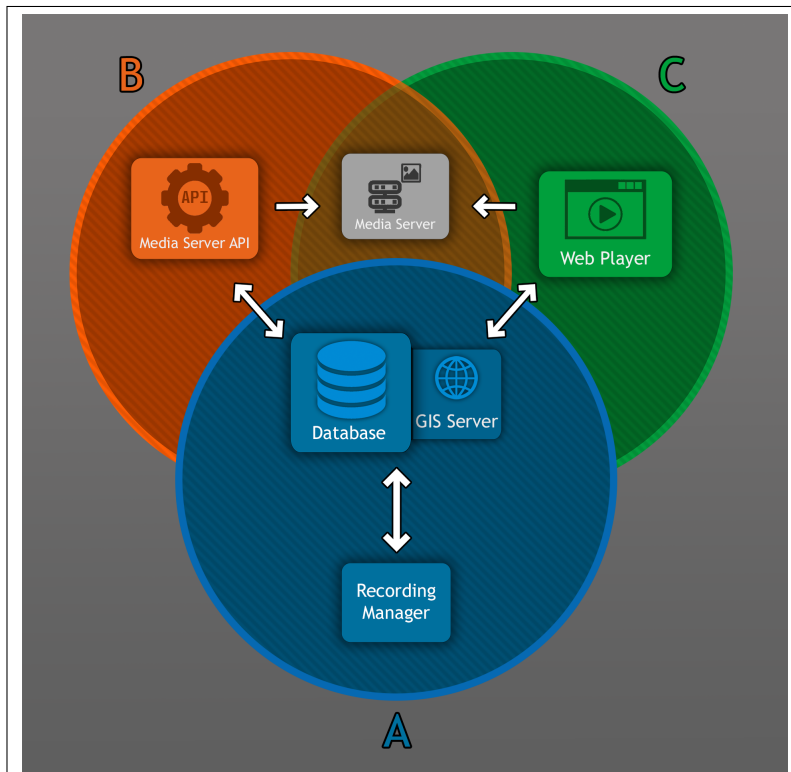


Figure 186: Horus Webservices System Architecture

A - The Recording Database

Section A is comprised of the Recording Manager (RM) and the Database (DB).

The Database contains information on the available recordings, this includes metadata such as: the length of the video, and the time and location of recording.

The Recording Manager acts as a link between the Database and the

physical recordings. It ensures that the data on the DB is up to date with that of the physical recordings.

For the Horus Webservices to function correctly, a running section A is always required.

B - Media Server API

Section B contains the Media Server API (**MS API**), and the Media Server (**MS**).

If you wish to render images from the recordings that are stored in the Database, then you would use the Media Server API.

The API acts as a facilitator between the Media Server and section A. The API contains a set of arguments which can be used to render images along specific parameters.

Some of these arguments are: The location, the field of view, the rendering mode, and the min-max angle of the defined location to the recording frame that is allowed to be rendered.

C - Web Player

Section C contains the Web Player (**WP**) and the Media Server.

This section is used when you want to playback a video that is available in the Database through the Web Player.

How does the Horus Webservices work?

The primary function of the Horus Webservices is to allow you to request images from recordings and have them rendered by a set of parameters.

You can request images from the Recording Database through either the Media Server API, or the Web Player.

The user can request images from recordings stored in the Recording Database through either the API or the WP. These requests contain various parameters to determine which recording/frame to use, and how to render it.

The Recording Database then replies with the requested data, and the API or WP then forwards this to the Media Server. The Media Server then renders the image based on the specified parameters.

9.2 Horus Webservices: Installation

9.2.1 Introduction: Installation

This chapter describes the process of installing the Recording Database (A) on your system.

For the installation of the Media Server you require the following :

- Horus Web Player installer
- PostgreSQL 12 installer
- PgAdmin 4 installer
- Horus License Key for Media Server
- Firewall Exception Script (*add_firewall_exception.vbs*)

During the installation of PostgreSQL and the Horus Web Player, only the settings that require changes are described, otherwise use the default settings of the installer.

9.2.2 Installing the Media Server

Step 1

Begin the installation process by opening the PostgreSQL installer.

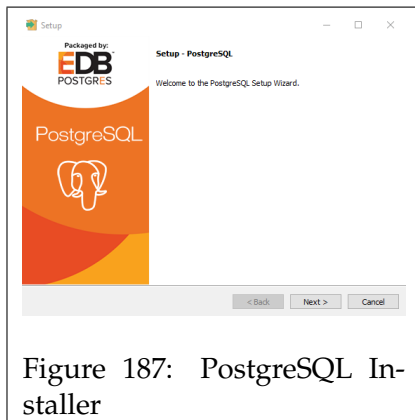


Figure 187: PostgreSQL Installer

Follow the instructions, till you come to the screen as seen in figure 188. Enter a password, this will determine the password of the database.



Then continue the steps, until it starts installing PostgreSQL.



Once the installation process finishes, you will be forwarded to the screen seen in figure 190.



Figure 190: PostgreSQL Complete Installation

Before continuing, check the option to install Stack Builder and then continue the installation with the "Stack Builder" wizard. The Stack Builder utility is used to install additional modules to PostgreSQL.

Step 2

Once the PostgreSQL installer closes, the Stack Builder setup will open.

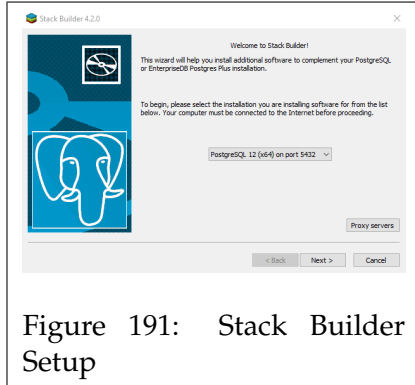
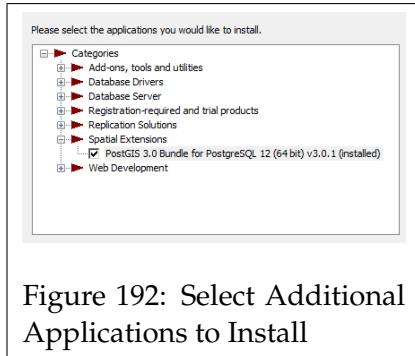


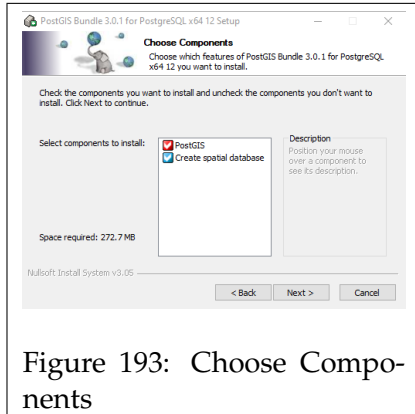
Figure 191: Stack Builder Setup

In the first window, select the PostgreSQL server, and press *next* to continue.



In the page that follows, enable "PostGIS 3.0 Bundle for PostgreSQL" found under "Spatial Extensions", and press *next* to continue.

Continue till you get to the window from figure 193. Check the box for the 'Spatial Database' and press *next* to continue.



After going through the next couple of windows, a window will open asking for your 'Database Connection Information'. Enter the password that you entered in the first step.

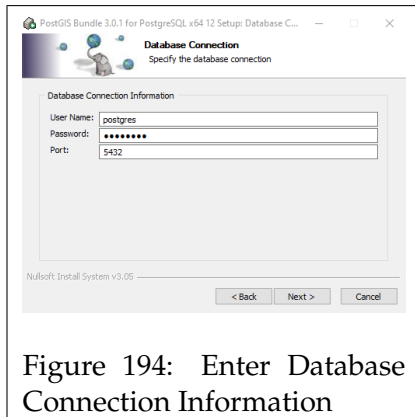


Figure 194: Enter Database Connection Information

Continue and the installation should start. During the installation, you may have a number of pop-up windows, press *Yes* to allow them.

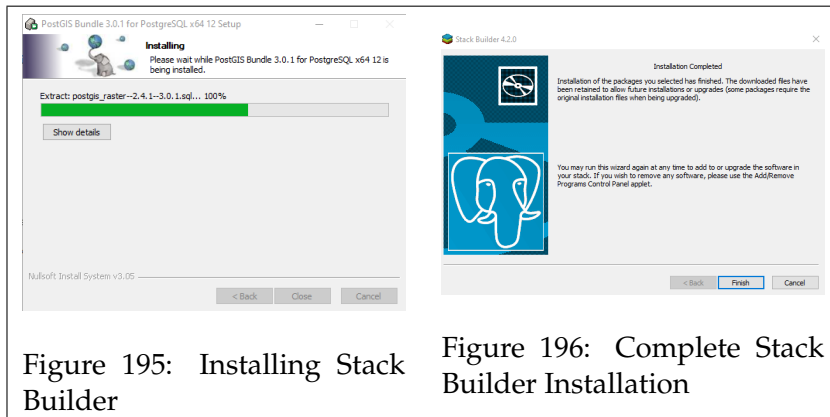


Figure 195: Installing Stack Builder

Figure 196: Complete Stack Builder Installation

Once the installation has finished, you will have installed the Database software on your PC.

Step 3

The next step is to install the Horus Web Player software.

Start the installer for the Horus Web Player.



Figure 197: Horus Web Player Installer

Click on next till the installation starts.



Figure 198: Horus Web Player Installation

During the installation, you will be prompted to enter a license key. If you do not have a license key yet, contact support@horus.nu, and send us your 'Machine key'.

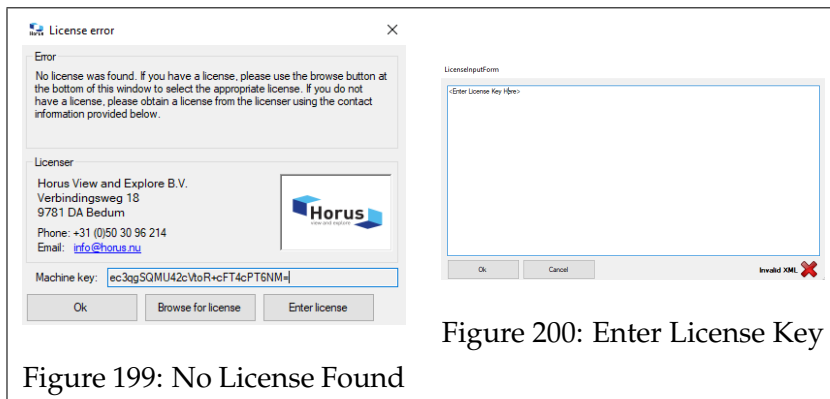


Figure 199: No License Found

Figure 200: Enter License Key

Browse for license, in the explorer window that opens, select the license file (.lic) to add the license.

Enter license, paste the license code into the input field, and click on 'Ok' to add the license key.

The Horus Web Player software is now installed on your PC.

Step 4

The following step is to check and edit a configuration file of the Horus software.

Navigate to the directory down below, and open the *config.json* file in Notepad.

```
C:\ProgramData\Horus View and Explore\HorusWebPlayer\config.json
```

In the config file edit the following lines:

- **Password**
Under the "Database" section, the password should be identical to the one entered during the PostgreSQL installation from step 1.
- **Host**
Ensure that this is: "*localhost*"
- **Recordings**
This determines the directory that the database uses to look for recordings
Default is: "*D:\Recordings*"

Once the changes are made, save and close the file.

Step 5

For the Media Server to function correctly, an exception to the firewall is required.

To add the exception open the included file.

```
"add_firewall_exception.vbs"
```

This file will automatically create the exception, and notify if it was successful.



Figure 201: Firewall Exception Added

Step 6

the next step is to initialize the Recordings Manager.

Open a command prompt as admin.

Navigate to the Recordings Manager folder in the installation directory of the Horus Web Player. The default location is:

```
C:\Program Files\Horus View and Explore\HorusWebPlayer\RecordingsManager
```

When you are in the correct directory enter the following command:

```
Horus.WebPlayer.RecordingsManager.exe --config --migrate
```

This will load the data from the config.json to the Recordings Manager.

```
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd\

C:\>cd Program Files\Horus View and Explore\HorusWebPlayer\RecordingsManager

C:\Program Files\Horus View and Explore\HorusWebPlayer\RecordingsManager>Horus.WebPlayer.RecordingsManager.exe --config --migrate
C:\ProgramData\Horus View and Explore\HorusWebPlayer\Data\Logs\RecordingsManager.log
Starting Horus.WebPlayer.RecordingsManager version 1.5.23
Changed geometry type from 'LINESTRING' to 'GEOMETRY'
Any key to quit
5
C:\Program Files\Horus View and Explore\HorusWebPlayer\RecordingsManager>
```

Figure 202: Loading the Config.Json to the Recordings Manager

Then the Recordings Manager can be started with the following command:

```
Horus.WebPlayer.RecordingsManager.exe
```

```

C:\Program Files\Horus View and Explore\HorusWebPlayer\RecordingsManager\Horus.WebPlayer.RecordingsManager.exe
C:\ProgramData\Horus View and Explore\HorusWebPlayer\Data\Logs\RecordingsManager.Log
Starting Horus.WebPlayer.RecordingsManager version 1.5.23
Any key to quit
RecordingsManager Start: 11:24:28
Inserting D:\Recordings\GoPro\test 2\Recording2020-03-02_09-11-28 of format ProtobufRecording
Importing setup file
Bounding box for 1: 'BOX(6.794413 52.9274494,6.794413 52.9274494)'
Inserting D:\Recordings\recording_16_06_20\360\Recording_16-07-2015_13-38-39 of format ProtobufRecording
Importing setup file
Bounding box for 2: 'BOX(6.601468337 53.299891,6.606536999 53.300570955)'
Inserting D:\Recordings\recording_16_06_20\Hr\Recording_16-07-2015_13-38-39 of format ProtobufRecording
Importing setup file
Bounding box for 3: 'BOX(6.601468337 53.299891,6.606536999 53.300570955)'
Skipping GoPro\name_05-06-2020_15-13-01 because this branch contains no recordings
Skipping GoPro\test 2\name_05-06-2020_16-15-30 because this branch contains no recordings
Skipping HSS\ because this branch contains no recordings
Skipping HSS\2020_01_22_21_00_38\ because this branch contains no recordings
Skipping HSS\2020_01_22_21_00_49\ because this branch contains no recordings
Skipping HSS\2020_01_22_21_00_59\ because this branch contains no recordings
Skipping HSS\2020_01_22_21_01_09\ because this branch contains no recordings
Skipping HSS\2020_01_22_21_01_20\ because this branch contains no recordings
Skipping HSS\2020_01_22_21_01_30\ because this branch contains no recordings
Skipping HSS\2020_01_22_21_01_40\ because this branch contains no recordings
Skipping HSS\2020_01_22_21_01_50\ because this branch contains no recordings
Skipping HSS\2020_01_22_21_02_00\ because this branch contains no recordings
Skipping recording_16_06_20\Gps\ because this branch contains no recordings
Skipping recording_16_06_20\Gps\Export\ because this branch contains no recordings
Skipping recording_16_06_20\Gps\Gps_1jken\ because this branch contains no recordings
Skipping test\ because this branch contains no recordings
Skipping test\2020_03_10_14_38_05\ because this branch contains no recordings
Skipping test\2020_03_10_14_38_15\ because this branch contains no recordings
Skipping test\2020_03_10_14_40_51\ because this branch contains no recordings
Skipping test\2020_03_10_14_40_51\Insert Recording name here_05-06-2020_15-12-52\ because this branch contains no recordings
RecordingsManager Stop: 11:24:29

```

Figure 203: Initializing the Recordings Manager

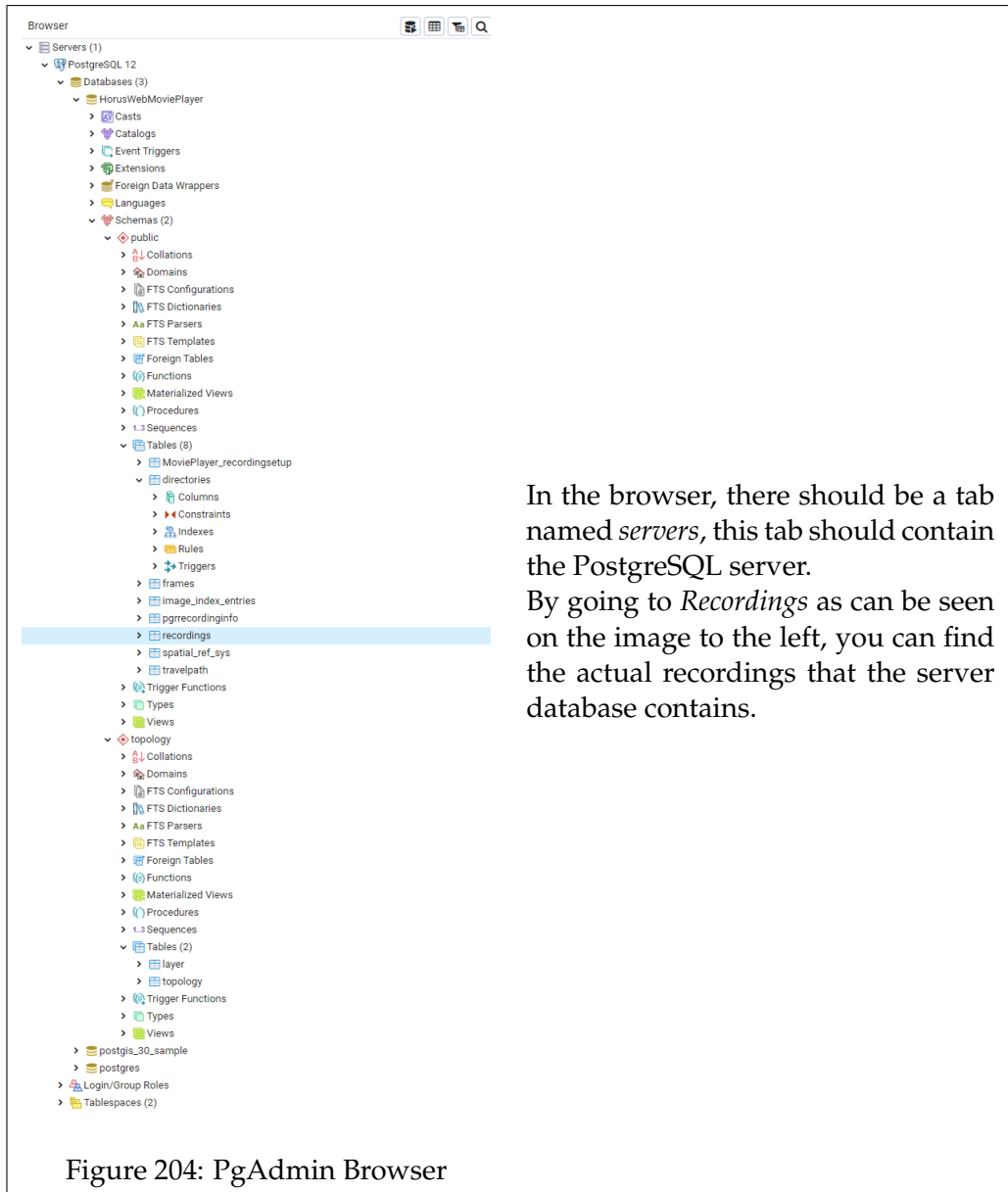
The Recordings Manager will start, and search through the recordings directory for any recordings. The database will only find recordings that have a *setup.hsf*.

The Media Server is now running and ready for use.

Step 6

This final step is to check if the Media Server is functioning correctly.

Install *PgAdmin*, and start the application. When prompted, the database password from step 1.



If you can find the server and it contains the videos from the recordings folder, then your Media Server is functioning correctly.

10 Appendixes

10.1 Glossary and terminology

10.1.1 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.

10.2 Flashing an Image onto an SD Card

10.2.1 How to Flash an Image onto an SD Card

In cases where you wish to update to a newer build of the software, or you have a new SD card for your Raspberry Pi, you might need to flash an image onto an SD card.

This image contains the Horus software and allows the Raspberry Pi to run an instance of our System V2 framework.

This chapter will describe how to install an image onto an SD card.

Flashing an Image

1. Download the image from the 'Horus Image Server' (<http://embed.horus.nu/>). The image can be found in the 'raspberry-pi' folder.
2. format your SD card.

3. Download and install balenaEtcher.* (<https://www.balena.io/etcher/>)
4. Start BalenaEtcher, then select the Horus image file, and the SD card drive that you wish to install the image to.
5. click on 'Burn' and wait for the process to complete.
6. Once finished, your SD card should be ready to use.

Partition size

After flashing your SD card, the partition size of the card might be reduced to the size of the image. This could result in gigabytes of disk space going unused. If you require additional storage space, you might want to increase the partition size on the SD card.

We recommend the 'GParted' tool to resize the partition.

(<https://gparted.org/>)

For more documentation regarding the installation of images, reference the Raspberry Pi documentation

(<https://www.raspberrypi.org/documentation/installation/installing-images>)

**Note: Linux users might require 'Zenity' for BalenaEtcher to function.*

10.3 HLL & HSS .ini File

10.3.1 introduction: HLL & HSS .ini File

The .ini files for both Linking Lab and Security Suite contain various settings for configuring the application.

There are two .ini files, one for Linking Lab and the other for the Security Suite.

The .ini file for Linking Lab is: *horus_app_hll_exe.ini*.

The .ini file for Security Suite is: *horus_app_hss_exe.ini*.

Each file configures their respective application, but the files themselves, should otherwise be identical, containing the same options.

.ini File Location with Windows

For Windows users the .ini file can be found in the following directory:
C:\Users\Username\AppData\Roaming\Horus View and Explore

.ini File Location for Linux

When using Linux the .ini file can be found in the following directory:
~/config/horus

To edit the .ini file open it with any text editor.

```

#v2-server=on
# Enable System v2 Noisy-Neutrino server:
# on: On: Enable.
# off: Off: Disable.

#discoverer-service=on
# Enable discoverer service:
# on: On: Enable.
# off: Off: Disable.

#v1-show-upstream=true
# Display upstream messages.

#v2-ser-net-url=http://112345
# The address for the control-message service [zmq-repl].

#v2-show-upstream=true
# Display upstream messages.

v2-discovery=on
# Enable discoverability:
# on: On: Enable.
# off: Off: Disable.

v2-log-events=Debug
# Minimum severity of log messages to send as events:
# Debug: Debug: Sporadic normal events.
# Info: Information: Interesting events.
# Warning: Warning: Unusual events.
# Error: Error: Error events.
# Fatal: Fatal: Errors from which cannot be recovered.

v2-cmp-search-path="C:\Program Files\Horus View and Explore\Horus LinkingLab\horus_plugins"
# Location of the plugin components.

#v2-pipeline=arg
# A V2 hrp file containing a pipeline to load at start-up.

#v2-start-pipeline=true
# Start pipeline at start-up.

#v2-persistent-pipeline=true
# Allow user to save pipeline on embedded device.

#v2-connection-statistics=true
# Compute connection statistics of a running pipeline.

#v2-lic-key=arg
# The license key of the product, to be activated.

```

Figure 205: Partial List of the .ini File Options

A list similar to figure 205 will appear.

Below, you can read more on the functionality of each option, along with a description on how to edit these options.

10.3.2 .Ini File Options

The list is comprised out of various configuration options

Each of these options has a short description of its functionality, followed by any possible settings.

```

#this-is-a-setting=true
# The description of the setting is placed on this line
# True: Option A: possible settings are placed here
# False: Option B: possible settings are placed here

```

Figure 206: Option Layout

Editing options

All the options are commented out with an # character by default. If you wish to edit an option, you can do this by removing the #, and then changing the setting. You can modify the option by editing the setting behind the =.

Options that are commented out, revert to a default setting. The default setting is generally what the original setting of the option was, prior to editing it.

This means that if an option is commented out, it can still be active.

You can also edit the .ini files through the System Configuration menu of Linking Lab, as seen in figure 207

When Linking Lab is running without the Security Suite, you can edit the .ini file for Linking Lab.

To edit the .ini file from Security Suite, Start the Security Suite, then open Linking Lab, and connect to the backend created by Security Suite.

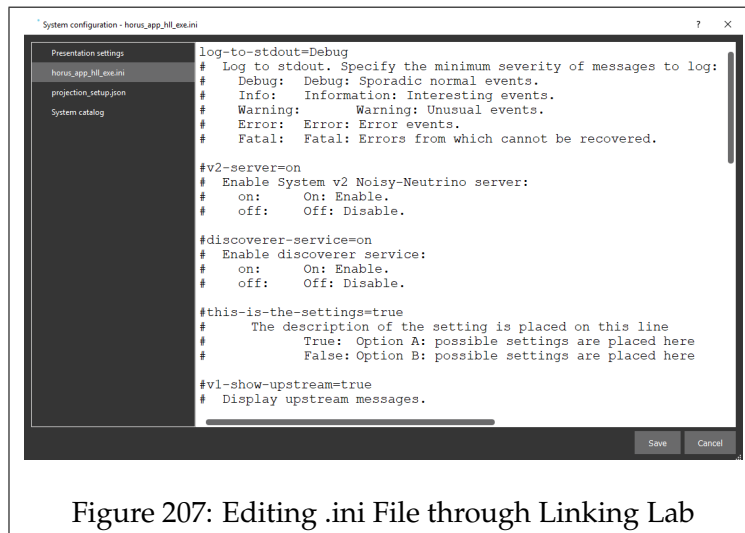


Figure 207: Editing .ini File through Linking Lab

.ini Options in the Terminal

You can also view the options from the .ini file through the terminal. In the terminal, navigate to the installation folder of either Linking Lab or Security Suite, depending on which application you would like to edit. Then type the executable of the program, followed by '*-help*'. This should then print a list containing all the options found in the .ini file.

```
PS C:\Program Files\Horus View and Explore\Horus SecuritySuite> .\horus_app_hss_exe.exe --help
Horus View and Explore - Horus Security Suite:
--help                Display help.
--sdk-version         Display the version of the Horus SDK.
--config arg (=C:/Users/Gebbruiker/AppData/Roaming/Horus View and Explore/horus_app_hss_exe.ini)
                    The configuration file.
--hardware-id        Display the hardware ID of the system.
--keygen              Generate an 2048 bit public & private key.
--min-thread-pool-size arg (=30)
                    Minimum number of threads to use.
--log-to-stdout arg
                    Log to stdout. Specify the minimum severity of messages to log:
                    Debug: Debug; Sporadic normal events.
                    Info: Information; Interesting events.
                    Warning: Warning; Unusual events.
                    Error: Errors; Error events.
                    Fatal: Fatal; Errors from which cannot be recovered.

Services:
--v2-server arg (=on)
                    Enable System v2 Noisy-Neutrino server:
                    on: On; Enable.
                    off: Off; Disable.
--discoverer-service arg (=on)
                    Enable discoverer service:
                    on: On; Enable.
                    off: Off; Disable.
```

Figure 208: Options in the Terminal

Starting the Software through the terminal with a Temporary Option

You can start up the application via the terminal by typing in the executable.

You may also add an option with a desired setting behind the executable. The application should then start up with the specified option.

```
Example: .\horus_app_hll_exe.exe -v2-discovery =off
```

The modified option is only temporary and will not be saved, it will thus only be used during that instance of the software. To make permanent changes, edit the options through Linking Lab or an text editor.

Updating .ini Files

Currently, the .ini file does not automatically update itself. New options could be added with newer builds, the .ini file could thus become outdated as a result. The software will run fine with an outdated .ini file, but you won't be able to configure the missing options.

The '-help' list in the terminal however, will always show the full list with all the .ini options.

To update the .ini file, there are two methods.

When a new option gets added, you could copy over the new one from the terminal to the .ini file.

The alternative method is to delete or move the old .ini file. On startup a new .ini file will be generated with the latest options.

Option Descriptions

Below you will find a list with the options from the .ini file, followed by a description for each option.

v2-server=on

Specify whether or not a System V2 server should be created when starting the application. If disabled, your own local system will not show up in the Connect Screen of Linking Lab. Useful when you only work on remote systems such as embedded systems, for example.

On: Enables the creation of a System V2 server.

Off: Disables the creation of a System V2 server.

discoverer-service=on

Enables or disables the discovery of other System V2 servers.

If turned off the connect screen will only show your own System V2 instance.

On: Enables discovery of other systems running System V2.

Off: Disables the discovery of other systems running System V2.

presentation-settings=

"C:\Users\Gebruiker\AppData\Roaming\Horus View and Explore\horus_app_hss_exe\presentation.json"

Points to the location of the 'presentation.json' file. This file contains configuration settings for the presentation layer.

Add the path between citations. The presentation.json file can be found in the appdata folder for Windows, and the config directory for Linux.

log-to-stdout=debug

On startup a terminal window opens along with the application. This option determines the type of information that gets written to the terminal window.

The specified setting determines the minimum severity of events that are written on the terminal window. From the possible settings, 'Debug' is the least severe, whilst 'Fatal' is the most severe.

Debug: Logs all types of events.

Info: Prints interesting and informative events.

Warning: Only shows warnings and unusual events.

Error: Only show errors.

Fatal: only show fatal errors which require a restart.

onvif-device-credentials=arg

Replace 'arg' with the username, and password to use for ONVIF devices. Separate the username and password with a colon (*username:password*). Used for the Video Proxy component.

video-decoder-priority=CUDA:D3D11:DXVA:FFmpeg

This determines the order of video decoders that are tried. Replace 'arg' with the video decoders to be tried in sequence. separate the video decoders with a colon (*Cuda:D3D11:etc*).

**video-format=loglevel:debug analyzeduration:0 probesize:32
fflags:nobuffer tune:zerolatency,fastdecode**

Additional video format options can be configured with this option. separate the different format options with a colon.

video-buffer-packet-size=-1

The amount of packets that the video buffer may contain. A value of -1 will automatically set the buffer size.

video-framerate-factor=1

The framerate factor acts as a multiplier for the playback speed, and should be at least 1.000000. For example, a factor of 1.25 would play the video 25% faster.

v1-show-upstream=true

Display upstream messages from a System V1 system. Enabling this option will print all control messages going from the server to the client, on the terminal window.
true: Show upstream messages.
false: Do not show upstream messages.

v2-ser-net-url=tcp://*:12345

You can define the address to be used here. When using both Linking Lab and Security Suite, you might want to use different ports for either application.

v2-show-upstream=true

Display upstream messages from a System V2 system. Enabling this option will print all control messages going from the server to the client, on the terminal window.
true: Show upstream messages.
false: Do not show upstream messages.

v2-discovery=on

This setting determines if other System V2 clients can detect your System V2 server.

On: Other System V2 clients are able to discover your System V2 backend.

Off: Other System V2 clients will be unable to detect your System V2 backend.

v2-log-events=debug

Changes the type of messages that get printed to either the notification window in Linking Lab, or the Status window in Security Suite.

The specified setting determines the minimum severity of events that are written to the application. From the possible settings, 'Debug' is the least severe, whilst 'Fatal' is the most severe.

Debug: Logs all types of events.

Info: Prints interesting and informative events.

Warning: Only shows warnings and unusual events.

Error: Only show errors.

Fatal: only show fatal errors which require a restart.

v2-cmp-search-path=arg

Enabling this option will result in the components being automatically loaded on startup.

Replace 'arg' with the path that points to the location of the plugins/component folder, this path should be between citation marks. The plugin folder can be found in the installation directory.

v2-pipeline=arg

Enabling this option will result in a saved pipeline being loaded on startup.

Replace 'arg' with the file path that points to the location of an .hrp file, this path should be between citation marks.

v2-start-pipeline=true

If enabled, the pipeline will start running on startup. Requires a pipeline to be loaded.

Option is set to false when commented out.

true: Starts running a System V2 pipeline on startup.

false: The system will not run the pipeline on startup.

v2-persistent-pipeline=true

This option allows you to save the current pipeline to a remote device; it enables the 'Save to Device' button from the Graph Builder.

Option is set to false when commented out.

true: Enables saving to remote systems.

false: Disables the ability to save to a remote system.

v2-compute-statistics=on

If on, the system V2 backend will compute connection statistics from a running pipeline.

This setting computes the statistics, to display them, use the 'v2-request-statistics' option.

On: System V2 will compute connection statistics.

Off: Connection Statistics will not be computed.

v2-compute-statistics-interval=00:00:05

The time interval between which to compute connection statistics.

Requires the 'v2-compute-statistics' to be enabled.

The interval is set in the following format (*Hours:Minutes:Seconds*).

v2-lic-key=arg

Activate a license key through the terminal.

Replace 'arg' with the license key between citation marks.

Currently, it only works when there is not an active license key already entered.

v2-lic-save-file="C:\Users\Gebruiker\AppData\Roaming\Horus View and Explore\system_v2_license.json"

Specifies the location where the 'license.json' file is saved to.

Place the path pointing to the 'system_v2_license.json' between citations

v2-lic-load-file=arg

The location to load the 'system_v2_license.json' from.

By default it checks the working directory, which is also the default save location.

To set a different location, replace 'arg' with the license key path between citations.

v2-data-content="C:/Users/Gebruiker/AppData/Roaming/Horus View and Explore/system_v2"

The path to the data content directory.

v2-com-service=arg

Provide the comport/serial port for the message service.(/dev/ttyACM0 | COM5)

v2-request-statistics=on

If on, your client will request connection statistics from the connected System V2 backend.

This setting requests the statistics from the server, ensure that the 'v2-compute-statistics' is set to enabled server-side.

On: The client will request connection statistics from the System V2 server.

Off: Connection Statistics will not be requested.

v2-request-statistics-interval=00:00:05

The time interval between which connection statistics are requested.

Requires the 'v2-request-statistics' to be enabled.

The interval is set in the following format (*Hours:Minutes:Seconds*).

extensions-path="D:/T/b/build/extensions/"

The path of the extensions libraries that are to be loaded.

These extensions libraries are required for certain features to function.

The 'extensions' folder can usually be found in the installation directory.

print-extensions=true

Print the loaded extensions and their capabilities to the terminal window.

true: Print extensions.

false: No extensions are printed.

print-extensions-settings=true

Print the settings from the detected 'Settings.json' files of the loaded extension libraries to the terminal window. *true*: Print extensions settings.

false: Do not print the extension settings.

discoverer-url=tcp://*:5556

The address used for the discoverer service.

10.4 report_settings.json

10.4.1 introduction: report_settings.json

The 'report_settings.json' contains settings for sending reports via E-mail, and the information that can be added to events.

A number of these settings are adjusted through the 'Settings' window, but a number of options will have to be set by manually adjusting the options. To send E-mails for example, you will need to configure some of these settings, this includes setting the recipient address, and entering the details for the SMTP server.

The file can be found in the following location:

- **Windows**
C:\Users\\AppData\Roaming\Horus View and Explore\horus_app_hss_exe\report_settings.json
- **Linux**
~/config/horus/horus_app_hss_exe\reports_settings.json

10.4.2 Configuring: report_settings.json

Upon opening the file, it will contain a list with settings. Below, the each of the settings will be described.

```
1  @{\n2    "printer": "Microsoft Print to PDF",\n3    "eventProperties": {\n66     "currentEventPath": "file:///C:/Users/Gebruiker/AppData/Roaming/Horus View and Explore/horus_app_hss_exe/events/test/",\n67     "eventsPath": "file:///C:/Users/Gebruiker/AppData/Roaming/Horus View and Explore/horus_app_hss_exe/events/",\n68     "reportTitle": "HSS Report- ",\n69     "smtp_url": "smtp://smtp.gmail.com:587",\n70     "smtp_user": "documentation@horus.nu",\n71     "smtp_password": "",\n72     "smtp_from_address": "documentation@horus.nu",\n73     "smtp_from_name": "Documentation",\n74     "smtp_to_address": "yunanhaaima@gmail.com",\n75     "smtp_cc": "",\n76     "smtp_subject": "Test Report",\n77     "smtp_inline_text": "This is a report",\n78     "smtp_ca_cert": "file:///C:/Users/Gebruiker/Downloads/ca-certificates.crt",\n79     "smtp_verbose": false\n80   }\n}
```

Figure 209: report_settings.json Options

Report Settings Options

Printer

Defines which printer is used to print reports.
This setting can be adjusted through the Settings Window.

eventProperties

The 'eventProperties' contains a list of the entry fields that are shown when creating a new event.

currentEventPath

This property points to the path of the event that is active.

eventsPath

This setting defines the default folder where events are stored. The 'Default events location' setting in the 'Settings' window changes the set path.

reportTitle

The title given to the report.

qsmtp_url

The url of the SMTP server.
(example: "smtp://smtp.gmail.com:587")

qsmtp_user

The username for the smtp server.

qsmtp_password

The password for the smtp server.

qsmtp_from_adress

The E-mail adress used as the sender of the report.

qsmtp_from_name

The name that is displayed as the sender of the report.

qsmtp_to_adress

The recipients E-mail adress.

qsmtp_ssl

Determines if SSL should be used for the SMTP server.
Enter a value between 0 and 4, these values correspond to the following:

- 0 = NONE, do not attempt to use SSL
- 1 = TRY, try using SSL, proceed anyway otherwise.
- 2 = CONTROL, SSL for the control connection or fail
- 3 = ALL, SSL for all communication or fail
- 4 = LAST, not an option, never use

qsmtp_subject

The subject title that is given to the sent E-mail.

qsmtp_inline_text

Adds text to the sent E-mail.

qsmtp_ca_cert

The path pointing to the certificate for the smtp server.

For more information regarding the certificate:

https://curl.haxx.se/libcurl/c/CURLOPT_CAINFO.html

qsmtp_verbose

Enabling this setting will print information regarding the SMTP connection to the terminal window. useful when you have connection issues, and you wish to debug the system.

10.5 SSH Connection with Embedded Devices

10.5.1 SSH Connection with Raspberry Pi

Connecting with SSH

To connect to an embedded system running the System V2 software, open a terminal window.

Enter the following: `ssh root@<id of the RPi>`

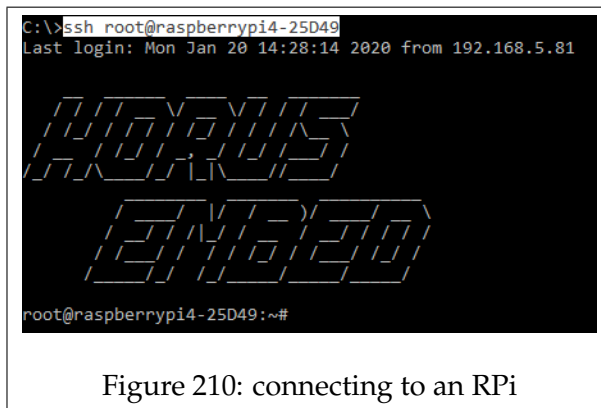


Figure 210: connecting to an RPi

If the 'Horus Embed' logo is printed on the window, then you have a successful connection.

Video4Linux compatible combinations

The Video4Linux Grabber might require a specific framerate for a certain resolution.

To check what these combinations are, you can request a list with all of the possible combinations through the terminal, by entering the following: `v4l2-ctl -list-formats-ext`

```
root@raspberrypi4-25D49:~# v4l2-ctl --list-formats-ext
ioctl: VIDIOC_ENUM_FMT
    Type: Video Capture

    [0]: 'YUVV' (YUVV 4:2:2)
        Size: Discrete 640x480
            Interval: Discrete 0.030s (33.000 fps)
            Interval: Discrete 0.067s (15.000 fps)
        Size: Discrete 1280x720
            Interval: Discrete 0.100s (10.000 fps)
        Size: Discrete 640x360
            Interval: Discrete 0.030s (33.000 fps)
            Interval: Discrete 0.067s (15.000 fps)
        Size: Discrete 544x288
            Interval: Discrete 0.030s (33.000 fps)
            Interval: Discrete 0.067s (15.000 fps)
```

Figure 211: Video4Linux Property Combinations

Linux Commands

Below is a list of commonly used commands for controlling directories.

- **cd**
Changes the current directory, to a specified directory. Enter 'cd' followed by the directory name to switch to. By entering 'cd' without specifying a directory, you will be returned to the root directory.
- **ls (-l)**
This command is used to print a list of the contents of the current directory.
- **mkdir**
New directories can be created with this command. Enter 'mkdir' followed by the name of the created directory.
- **pwd**
Entering this command will display the present working directory.
- **rm**
Enter 'rm' followed by a specified file to delete it.
- **rmdir**
Enter 'rmdir' followed by a specified directory, to delete it.

Use the URL below for a more complete list of Linux commands
<https://www.raspberrypi.org/documentation/linux/usage/commands.md>

10.6 Citymapper Spec Sheet

- ▶ Configuration & Training on site included
- ▶ First year of support Horus Software included
- ▶ High resolution camera inserts can be added

Learn more: www.horus.nu

Horus Citymapper



CAMERA	
Spherical camera	FLIR Ladybug 5+
Chip	32 MP CCD
Resolution	8000 x 4000 pixels

RECORDING LAPTOP	
Operating System	Windows 10 (64 bit)
Storage	2 TB internal SSD

ELECTRONICAL DATA	
Power supply input voltage	19 V DC

Software included	
CAPTURE	
	The Horus Movie Recorder <i>Easy and stable way of capturing hundreds of kilometers a day</i>
VIEW	
	The Horus Movie Player <i>View your data with the free Horus Movie Player</i>
EXPLORE	
Modules	
	<i>Layermanager: work with CAD/GIS</i>
	<i>Orthotool: export orthophotos (GeoTIFF)</i>
	<i>Position Fixer: improve imagery with post processed GPS</i>

PHYSICAL CHARACTERISTICS	
Dimensions Set	1655 mm x 208 mm x 558 mm (L xB xH)
Weight Set	22 kg
Transportcase	800 mm x 500 mm x 300 mm (L xB xH)
Total weight incl.case	34 kg

POSITIONING SYSTEM		
POS Type	Professional (e.g Rural) LV125	Advanced (e.g City) LV220
	Accuracy	
X, Y Position (m)	0,050	0,020
Z Position (m)	0,080	0,050
Roll and Pitch (°)	0,025	0,020
True Heading (°)	0,060	0,025

<https://www.applanix.com/products/poslv.htm>



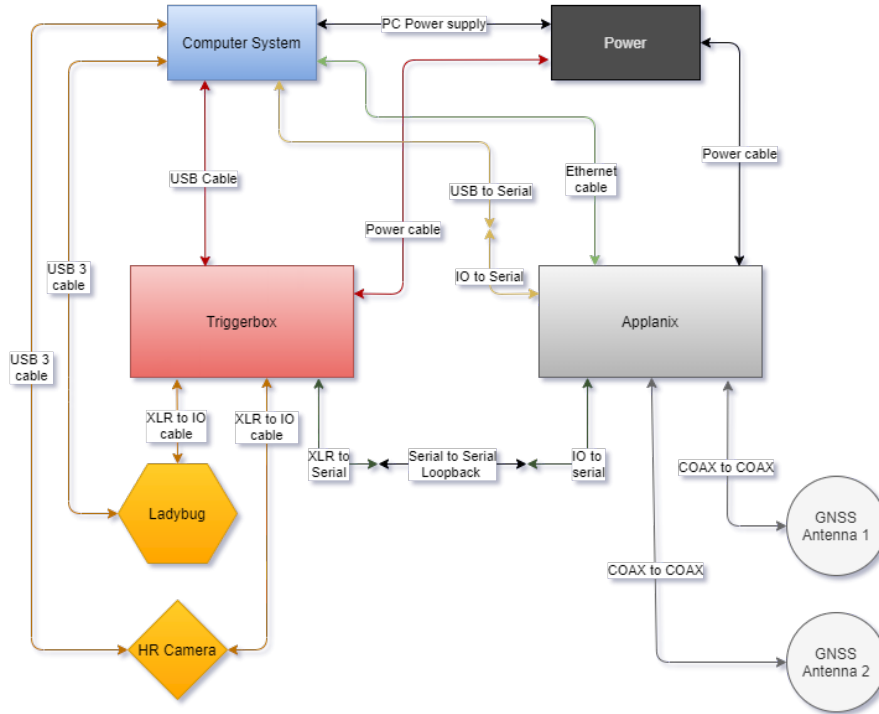
Office: Verbindingsweg 8, 9781 DA Bedum, The Netherlands

Phone: +31 50 309 62 14

Email: sales@horus.nu

10.7 Citymapper Hardware Diagram

The diagram below shows a system where the Applanix is the trigger. The cable over which the pulses are sent is connected to the 'Trigger In' port of the Triggerbox and the cable has a loopback section in between.



The next graph shows a system with an OBD reader unit as the trigger. The OBD unit is connected to the 'Trigger In' port, whilst the Applanix is now connected via the 'Trigger Out' port and is missing the loopback section.

