

## 1 WARNING

Despite the great accuracy of the software, it should not be used in safety relevant areas. Only use this software for its intended purpose.

## 2 Support

For questions and support, please contact the e-mail: [opdra@vischeck.de](mailto:opdra@vischeck.de).

## 3 Quick Description

There are three components for operating the Opdra software:

**Opdra Server:** The evaluation of the camera data runs on this software. The server is installed on the computer that is connected to the cameras.

**Opdra Genius:** Genius is the user interface for the server. Genius is installed on a computer from which the server will be configured.

**Annotation:** Annotation is used to create the ROI's. Annotation is installed on the same computer as Genius. (Autonomous software for image processing)

It is also possible that all three components run on the same computer if the performance is sufficient.

## 4 Quickstart (How to setup a project)

### ▲ This Software will only run on 64-Bit Windows

#### 4.1 Server

1. Start the Opdra\_Server\_Installer.exe and install all the packages. (Follow the installation guide)
2. To successfully run the server.exe, all of the following packages from the installer have to be installed.
  1. [Microsoft Visual C++ Redistributable 2015, 2017 und 2019 x64](#)
  2. [Asp.Net Core Runtime 5.0.6 x64](#)
  3. [Microsoft .Net Framework 4.7.2 x64](#)

3. Open the document path  
"C:\Users\[User]\Documents\VisCheck.OpdraServer\Config\config.xml"  
and edit server ip in the config.xml

▲ **If Genius and Server are running on the same computer, then you can leave the "127.0.0.1" ip address.**



Dieser PC > Dokumente > VisCheck.OpdraServer > Config

Name	Änderungsdatum	Typ	Größe
config.xml	02.07.2021 08:07	XML-Datei	1 KB

```
<?xml version="1.0" encoding="utf-8" ?><SERVER><ip>127.0.0.1</ip><port>54000</port></SERVER>
```

*Config.xml screenshot*

4. The server is now missing files, which can be created and uploaded with genius to the following path:  
"C:\Users\[User]\Documents\VisCheck.OpdraServer\project".  
This will be explained in the next steps.

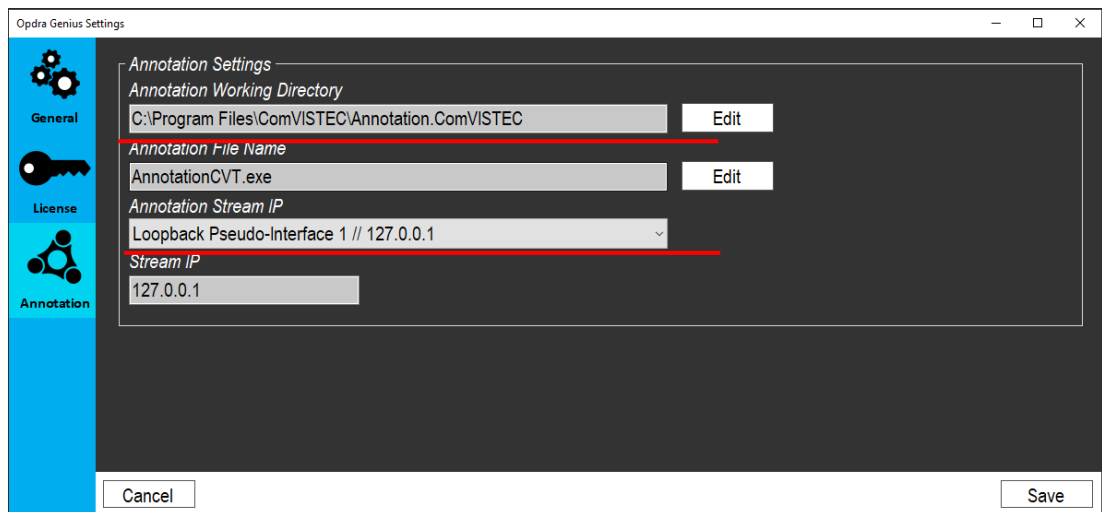
## 4.2 Genius and Annotation

1. Start the "Opdra\_Genius\_Installer.exe" and the "Annotation\_ComVistec\_Installer.exe".  
Install all the packages.
2. To successfully run Genius and Annotation, all of the following packages from the installer have to be installed.
  1. [Microsoft Visual C++ Redistributable 2015, 2017 und 2019 x64](#)
  2. [Microsoft .Net Framework 4.7.2 x64](#)

#### 4.2.1 Settings License, Annotation, Server

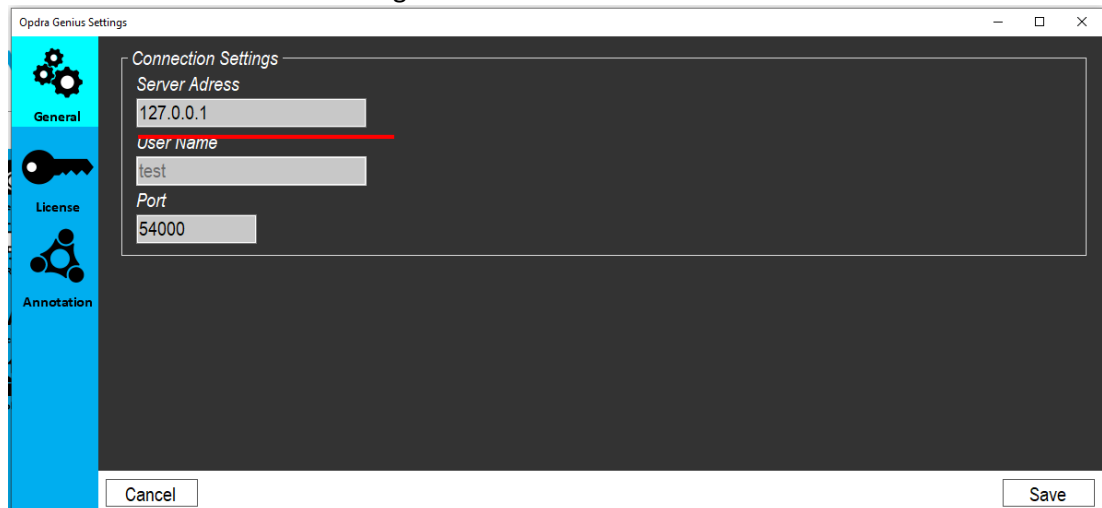
1. Open Genius, go to “**Edit > Settings Genius > Annotation > Annotation**”. In “**Annotation Working Directory**” add the current path of the Annotation folder if it is not already included. (With this path, Genius knows where to open the annotationCVT.exe) (Annotation must be installed).

The IP address of the computer from which Genius was started is entered under „**Annotation Stream IP**“. (This is where you will receive the server stream)



*Screenshot of the Annotation Settings*

2. Click in the same window on General. Enter here the IP address from the server, it should be the same as in the server's config.xml file.



*Screenshot of the Connection Settings*

- If you have a license, you can also open the license window and enter the license under “**Licence Key**”. Should the verification be successful, the Activation and Expiration Date will be displayed. This will generate a file called “licence.xml”. This file is necessary for the server, without this file the server won’t run.

Licence		
First Name	Last Name	Company
TestName	TestLaseName	TestCompany
Licence Key	Activation Date	Expiration Date
gCvdEETerhKMzS8c	18.05.2021 11:45:54	17.07.2021 11:45:54

Verify

*Screenshot of the Licence settings*

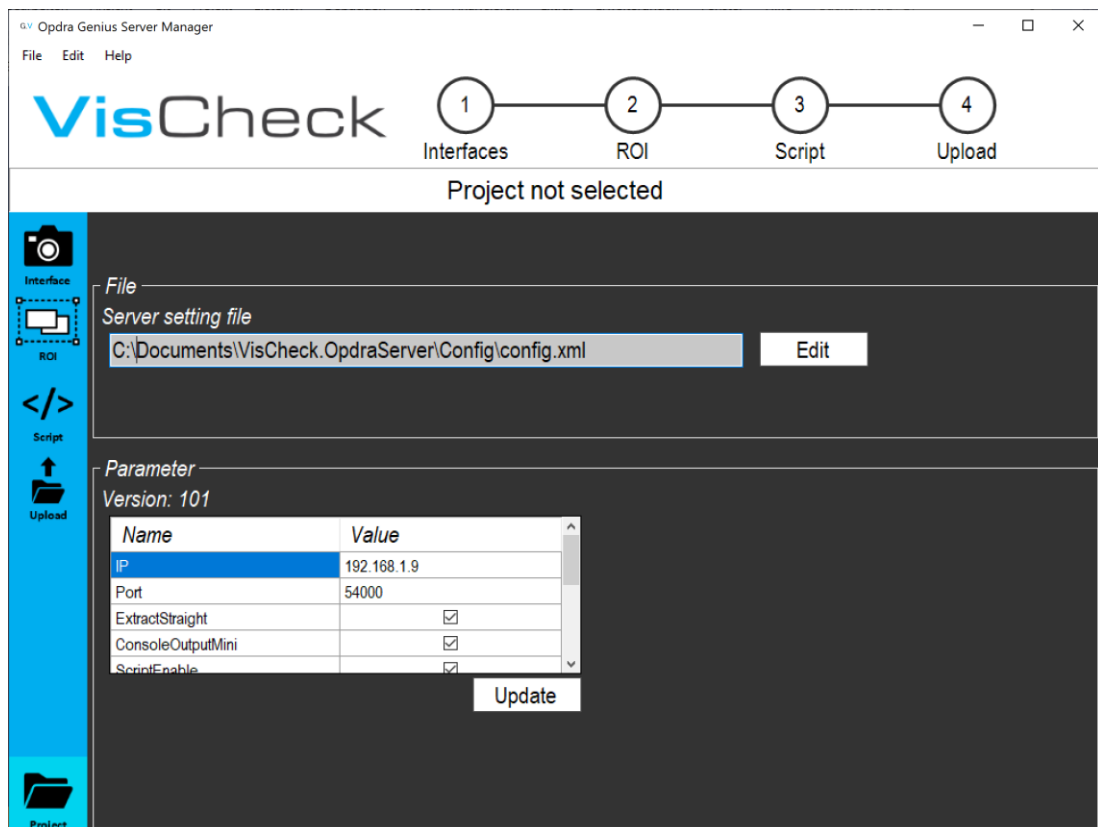
Press “**Save**” to save all the changes. Genius will restart now to implement all the changes.

## 4.2.2 Server settings about Genius

⚠️ This is Optional, you can skip this step!

⚠️ To edit the server settings with Genius, you need the config.xml from the server on the same PC as Genius. Or you have access to the file from this PC.

Under the upper menu go to **Edit > Settings Server**. Add the current path of the conifg.xml from the server. You can choose the config.xml with the “**Edit**” Button. If the correct file was selected, the server settings will now be displayed under “Parameter”. There you can edit the settings. Press “**Update**” to implement the changes.



Screenshot of the server settings

#### 4.2.3 Creating new Project


1. Create a new project by assigning it a name and pressing **“Create Project”**. If a project is available, you can also load it by selecting the project from the dropdown menu and pressing **“Load Project”**.

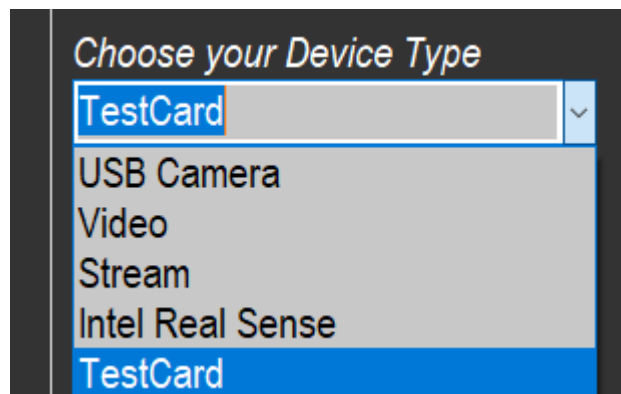


*Screenshot of the starting page in Genius.*

#### 4.2.4 Connecting to the server and choosing the streaming device

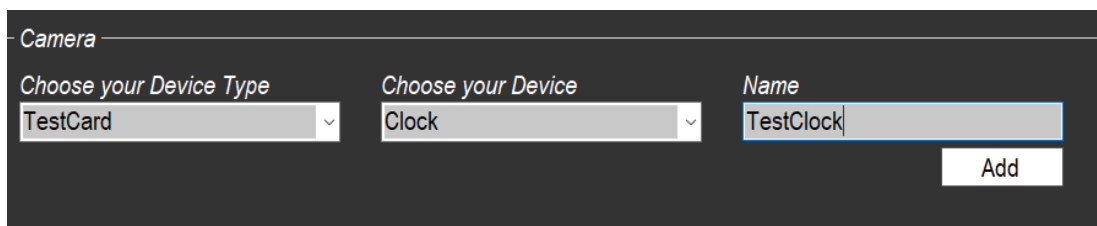


1. Now click on the interface icon . A connection to the server will be established here. (The server must be running. IP and Port number must be correct.) If the connection is successful, you should be able to choose one device type.



*Screenshot of the „Choose your Device Type“ block in „Interface“*

2. For this example, we choose the **“TestCard”** as device type. This simulates a camera stream from the server. Choose the device **“Clock”** and assign it a name. Click on **“Add”** to add this device. Later you can choose a camera as device type.



*Screenshot of the camera setup in the „Interface“*

The settings for the camera stream will now open. Here we set the Width value to **1920** and the Height Value to **1080**. Press **“Update”** to save the changes.

Name	Type	Value
Width	int	1920
Height	int	1080
FPS	int	30

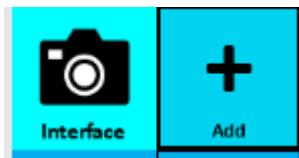
*Screenshot of the Camera settings*

By clicking **“Open”**, Genius will now receive the stream of a clock from the server. Afterwards press **“Close”** again to end the stream.

Name	Value
Background	0
Foreground	255
GMT	2

*Screenshot of an open camera stream*

- For this example, we also need to create an Input / Output.



Press the “Add” icon to create a new Input / Output.

Choose “Timer” as the Input / Output type. Assign a name to it.  
After clicking “Add” this Input / Output will be saved.

This will determine how often the server should read the camera stream.

**▲ An Input / Output must be created, otherwise the server won't read any data.**

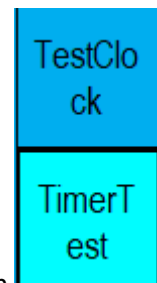
Input / Output

Type: Timer

Name: TimerTest

Add

Screenshot of the Input / Output setup in the „Interface“



- If you wish to adjust the Time you can press the created “TimerTest” Icon. Assign it a value you like (**1000 equals 1 second**). Press “Update” to save the changes.

Input / Output

Name: TimerTest

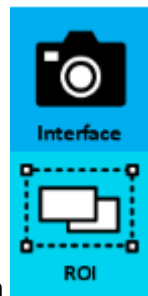
Parameter	
Name	Value
Interval	1000

Update

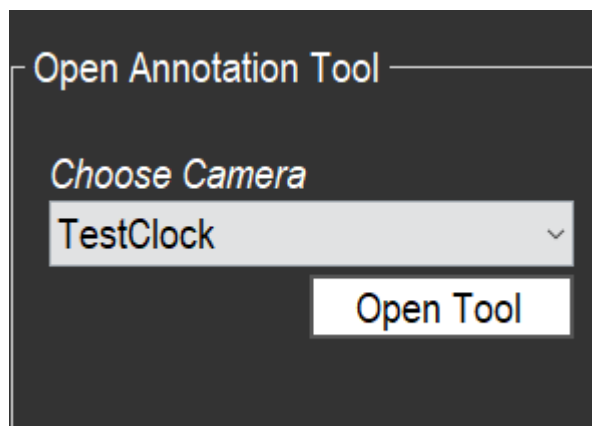
Screenshot of the Timer settings



## 4.2.5 ROI



1. Under the ROI icon you can now choose the created camera input and open the Annotation Tool with the “**Open Tool**” button. At this point we are working with another software. Annotation will start automatically. After we are done with Annotation, we come back to Genius.



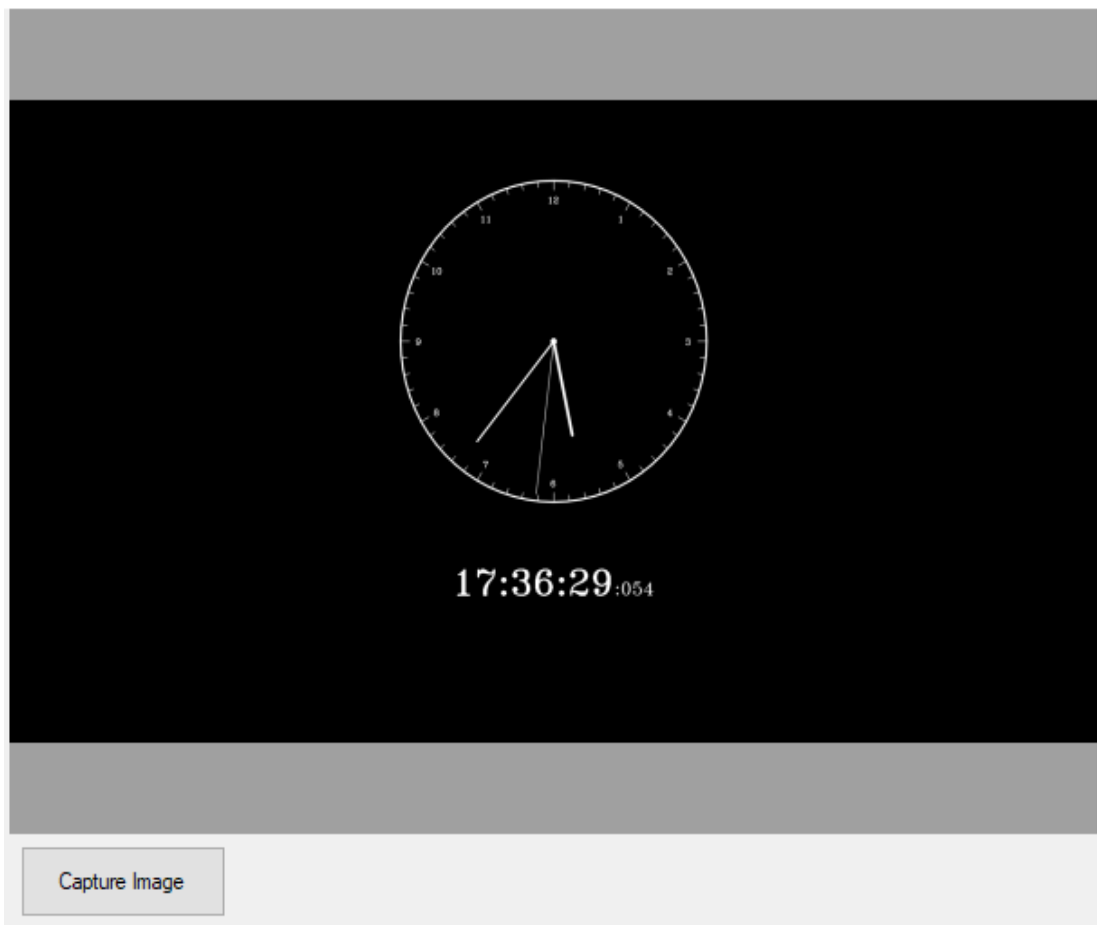
*Screenshot on how to open the Annotation tool*

## 4.2.6 Annotation

### ⚠ Software switch (Genius -> Annotation).

1. In the upper menu go to: “Tools > Get Image from Devices”.

Double click the stream Text **Stream (UDP (H264))** to start the clock stream.

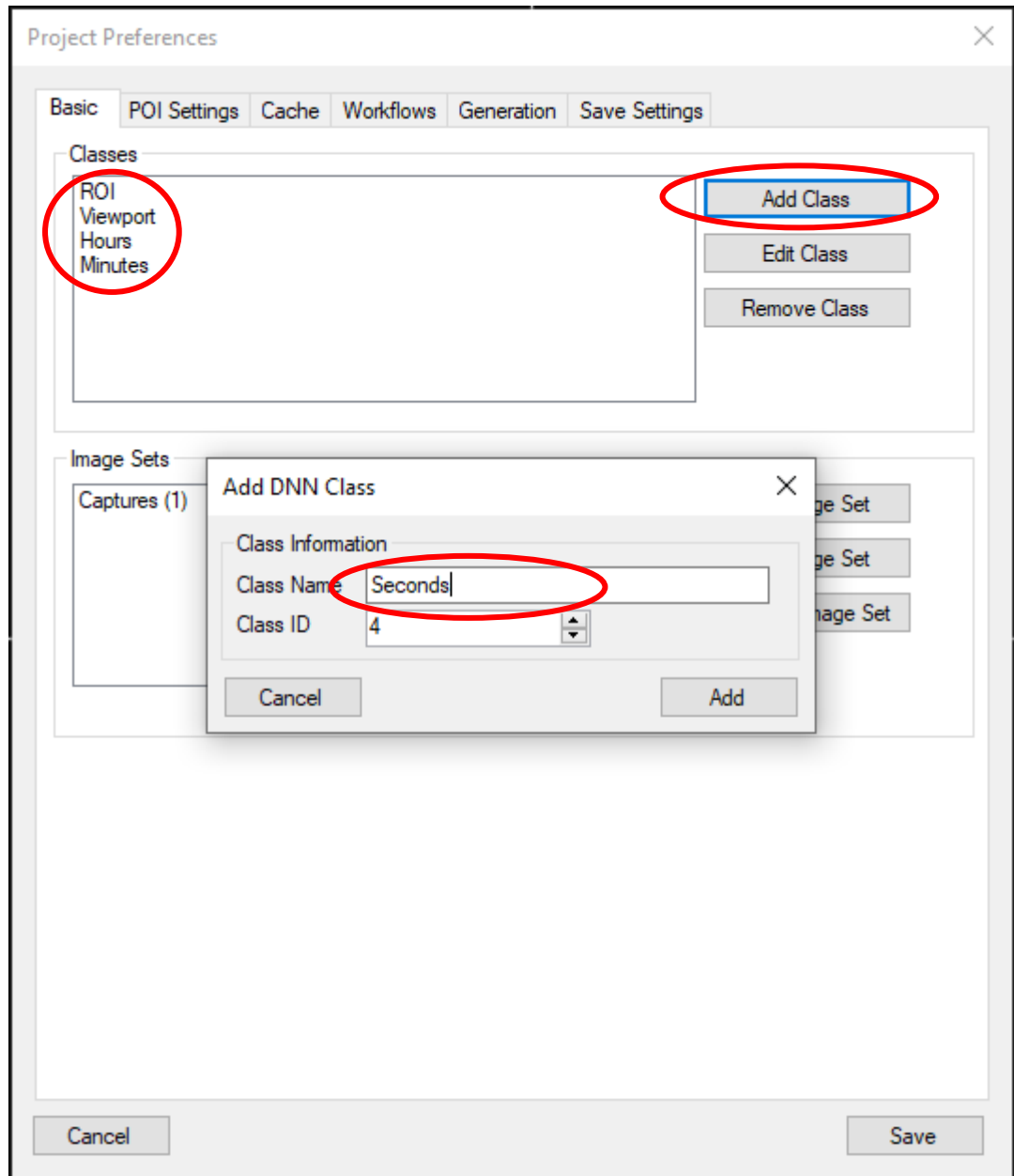


*Screenshot of the Clock stream from the server*

You can now click “**Capture Image**” button to take a picture.

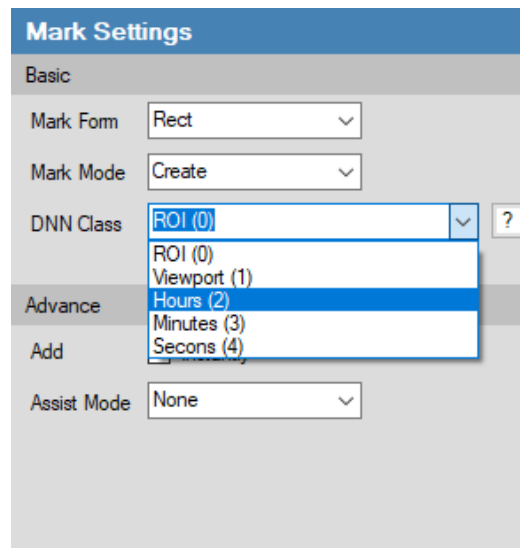
Press “**Save**” to save the picture.

2. In the upper menu bar click on **“Edit > Project Settings”**.  
Here we can edit or add a new class. For this example we add three new classes (**Hours, Minutes, Seconds**).
3. To add a new class, click on **“Add Class”** and choose a name.  
We add the class **“Hours, Minutes, Seconds”**. Make sure it is written correctly.



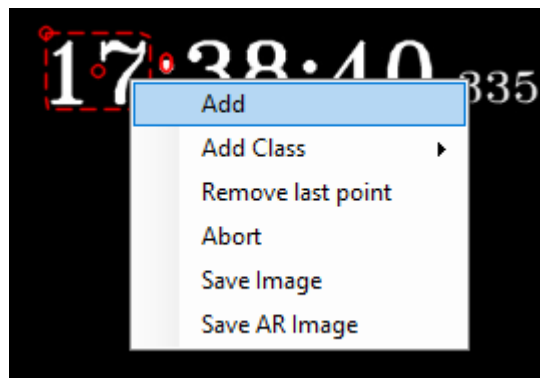
*Screenshot of the class configuration under Project Preferences*

- On the left side, under Mark Settings you can now choose one of your created classes.
- Pick the Class **Hours**.



*Screenshot of the Mark Settings in Annotation*

- Now you can drag the mark over the image and by pressing “**right mouse button > Add**”, the ROI will be created.



*Screenshot on how to Add a ROI*


**⚠** When creating the ROI, the height is very important. No other Objects should me in the ROI.

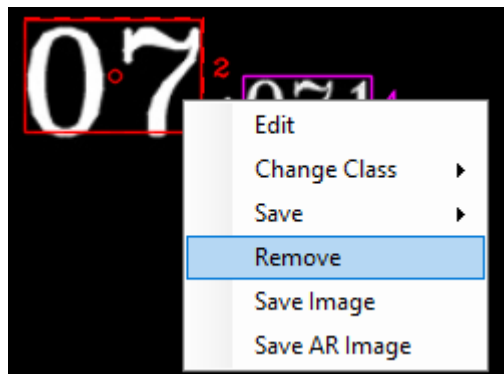


NOT LIKE THAT



LIKE THAT

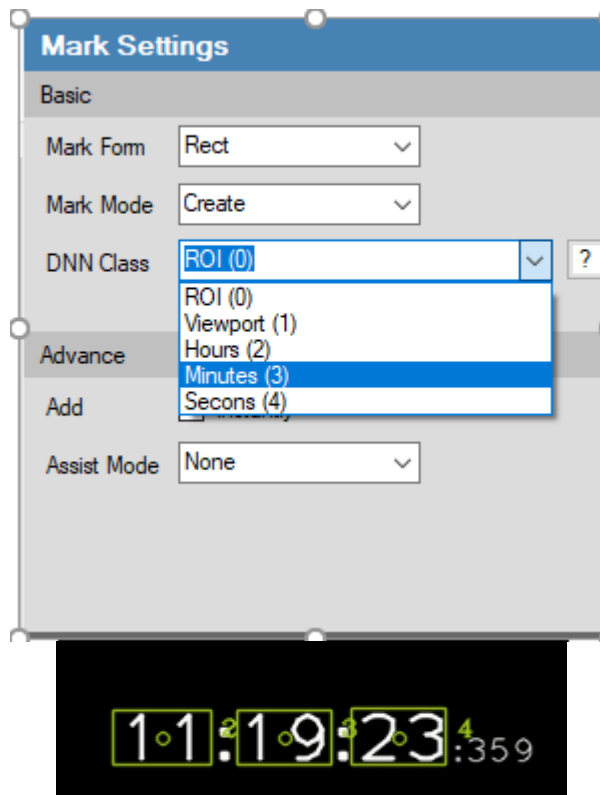
7. If you misplaced an ROI, you can remove it. Click on the “**Edit Mark**” icon  in the upper menu, then left-click the mark you want to remove, right-click it and press remove.



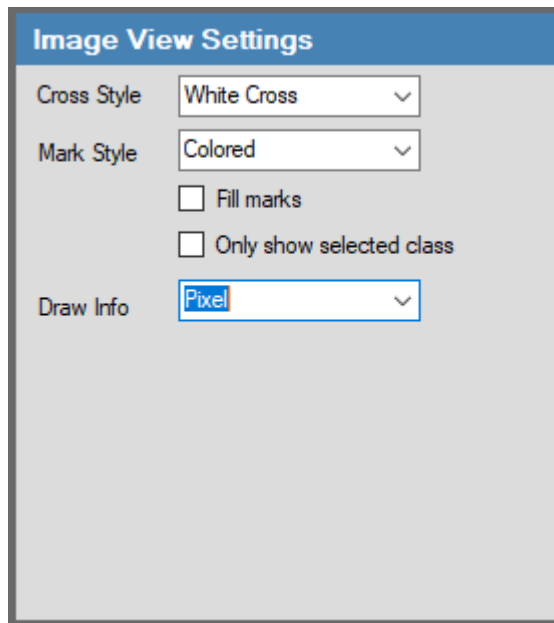
*Screenshot on how to remove an ROI*

Click on the  icon to add new ROI's.

8. We now change the class under “Mark Settings” to “**minutes**”, “**seconds**” and mark them accordingly.



9. Select on the right side under “Image View Settings” the “Draw Info” and set it to “Pixel”.



*Screenshot on how to select the Draw Info*

This will allow us to see the resolution of our ROI's.

10. Now we choose the “**Viewport**” class and mark it around the clock and time like shown in the image below.

This will be later used as viewport.

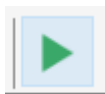
A viewport can determine if the screen changes and adjust its ROI reading accordingly.

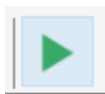
It can also straighten a crooked image.

**⚠** We will need the resolution from the “**Viewport**” later on “636x772”. You can view this again by opening the Annotation tool.



*Screenshot of a picture with marked ROI's*




11. Click the  button in the upper menu of Annotation to generate the project data. After the generation is done, Annotation will close, you can open Genius again.

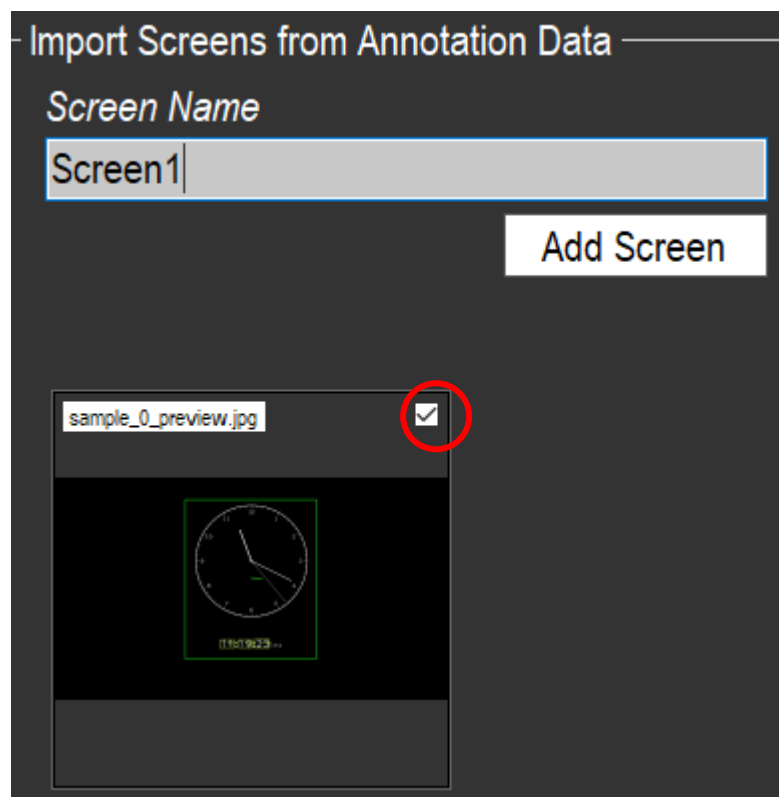
#### 4.2.7 Create Screens

##### ▲ Software switch (Annotation -> Genius).

1. Back in genius under the “ROI icon > Import Screens from Annotation Data” a name can now be assigned to the screen.

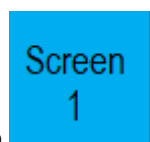
A screen must be selected below.

You can see what screen is selected by the little “checkbox” icon . This Screen can now be added with the button “Add Screen”.



Screenshot of the „Import Screens form Annotation Data“ section in „ROI“

2. Now click the newly created screen Icon





Click on the “Add Viewport” button besides “Viewport” to generate a viewport.

ROI ID	Name	Viewport	Add Viewport
41658	Hours	None	Add Viewport
73320	Minutes	None	Add Viewport
38034	Seconds	None	Add Viewport
27978	Viewport	None	Add Viewport

Screenshot of the Regions of Interest section under ROI

**⚠ The width and height of the viewport have to be very accurate! By writing to large or to little values the server wont be able to read any data from the screen.**

In the newly created viewport assign the width and hight from the ViewportClock resolution we took. You can look up the values again by opening the Annotation tool in the taskbar.



ROI ID	Name	Width	Height	Remove
54714	Viewport	636	772	Remove

Screenshot of the created viewport in ROI

Now you can adjust the ROI’s and assign them to a viewport by clicking the selection bar under “Viewport”. Assign all the ROI’s to Viewport.

ROI ID	Name	Viewport	Add Viewport
41658	Hours	Viewport	Add Viewport
73320	Minutes	Viewport	Add Viewport
38034	Seconds	Viewport	Add Viewport

Screenshot of the Region of Interest section under ROI



Screenshot of the screen update button

Press **“Update”** to save all the changes.

#### 4.2.8 Create Script



1. Now you can create the Script.  
We have four Script parts **“Init”**, **“Trigger”**, **“Destroy”** and **“Event”**  
Init, Destroy and Event are ready to use for this example, you don't need to edit them.

We only need to adjust the **Trigger** Script. Paste in the **Trigger** window with the keyboard shortcut **“Strg+V”** the following code:

**⚠ The names must be the same as in this example.**

---

```
function opdra_trigger(IOName, Message)
    roiHoursID = Opdra:roiNameToId("Hours")
    roiMinutesID = Opdra:roiNameToId("Minutes")
    roiSeconsID = Opdra:roiNameToId("Seconds")

    Opdra:deviceCapture()
    Opdra:debugShowCaptured(800, true)

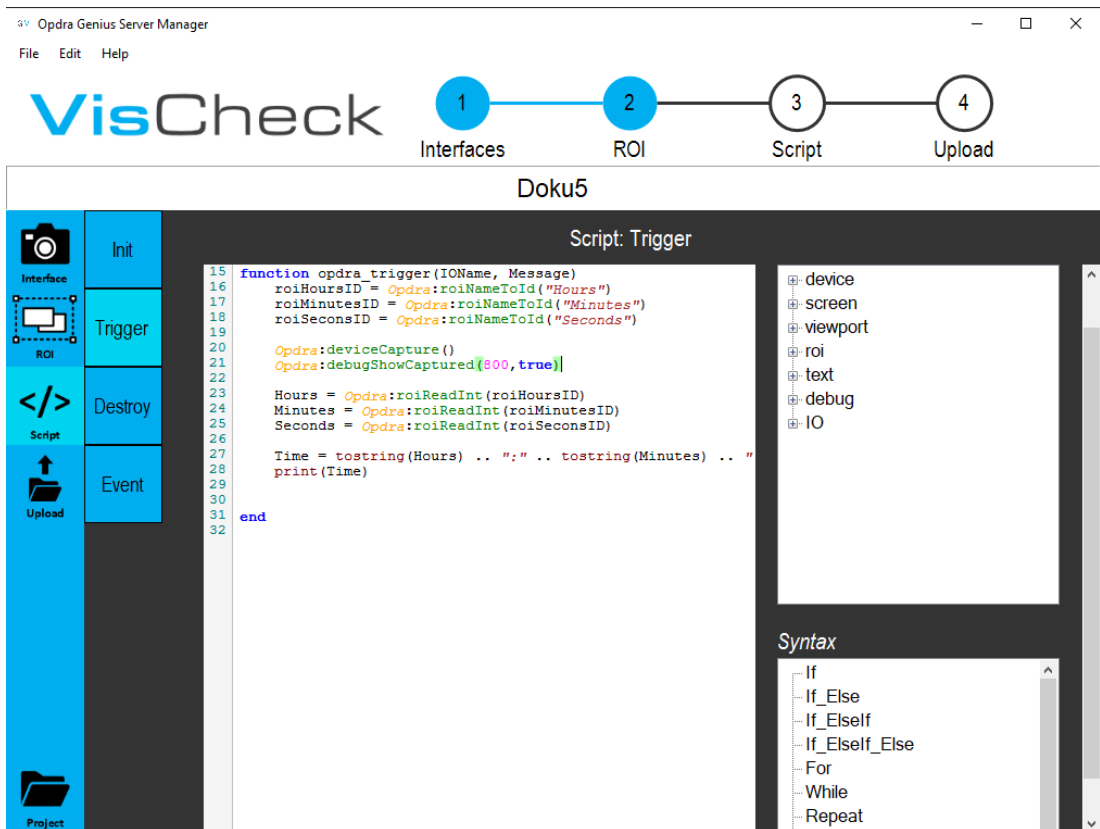
    Hours = Opdra:roiReadInt(roiHoursID)
    Minutes = Opdra:roiReadInt(roiMinutesID)
    Seconds = Opdra:roiReadInt(roiSeconsID)

    Time = toString(Hours) .. ":" .. toString(Minutes) .. ":" .. toString(Seconds)
    print(Time)

end
```

---

The script should look like this:

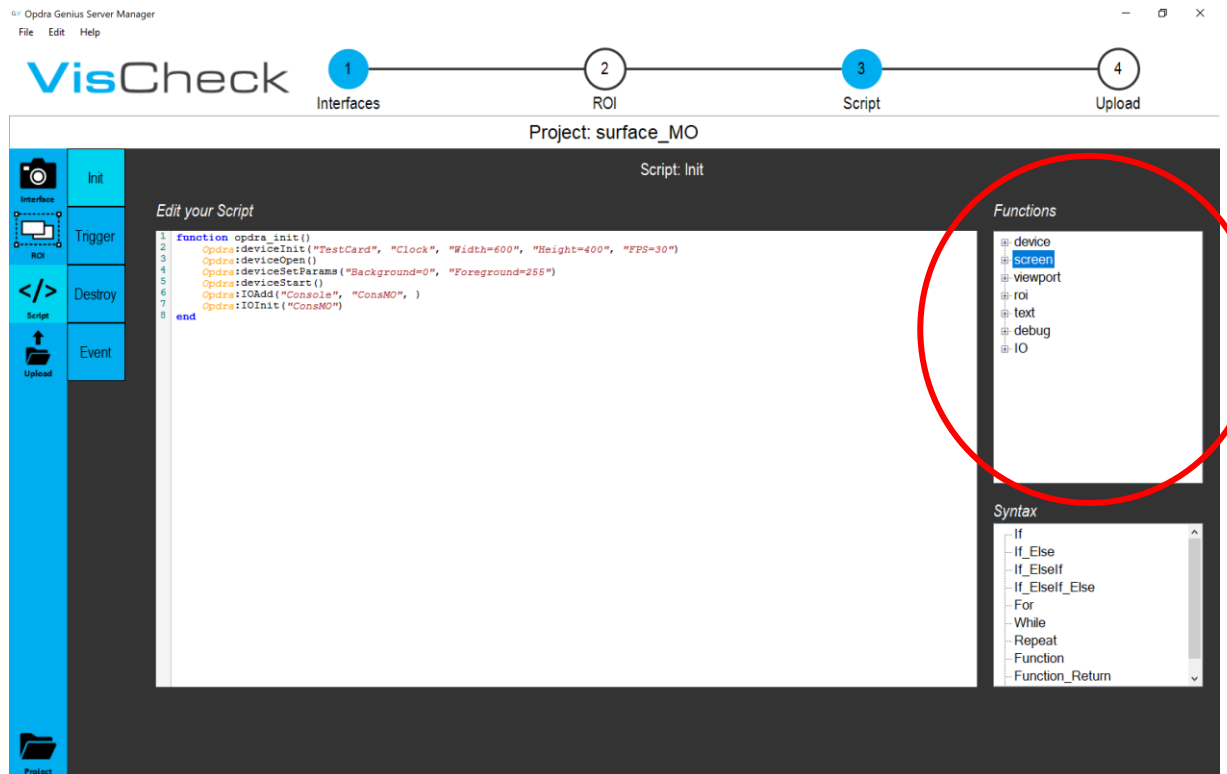


Screenshot of the pasted Trigger script.

⚠️ This is Optional, you can skip this step!

### Adding your own functions

If Server is running and Genius is connected you will get the script functions from the server.

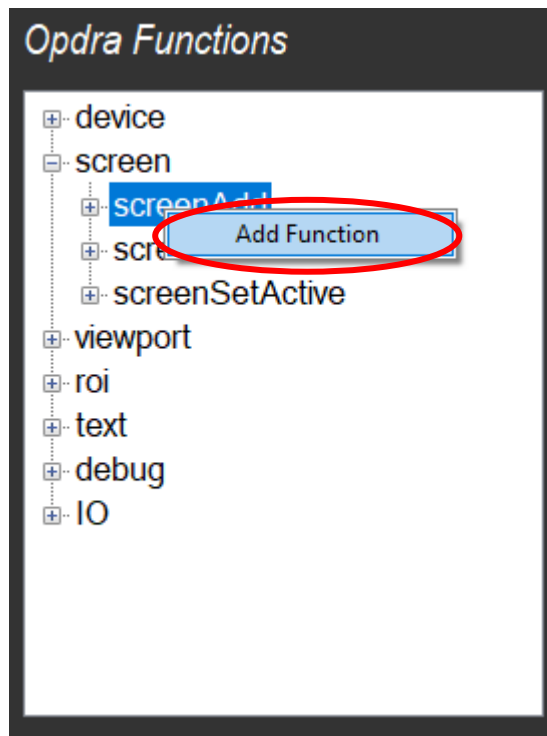


Screenshot of the Functions menu

Place the cursor where you wish to add your code.

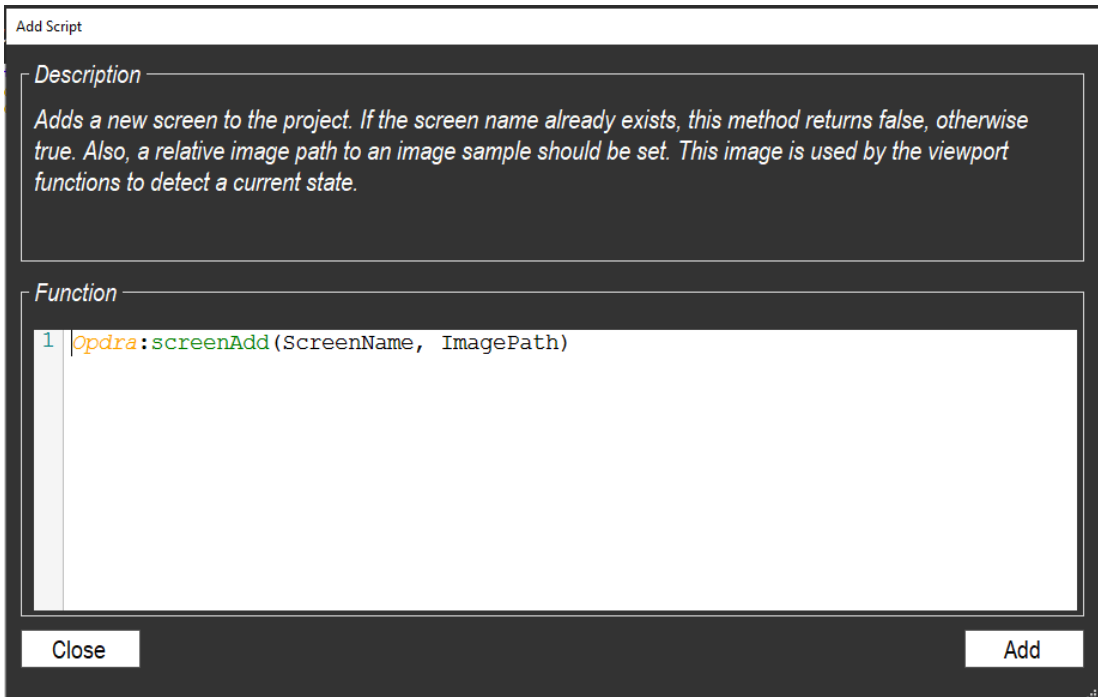
```
1 function opdra_init()  
2   Opdra:deviceOpen()  
3   Opdra:deviceStart()  
4  
5   |  
6  
7  
8 end
```

Choose the function you wish to add under "Opdra Functions". Press left click and then "Add Function". This will open a separate editor window with example code and description.



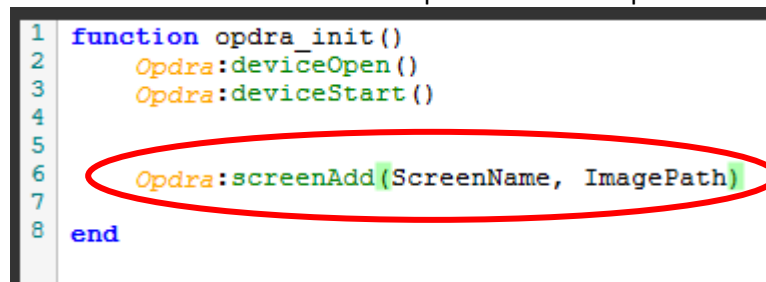
Screenshot of the Opdra Functions submenu

Here you can adjust how you want the function to look like.  
Press **“Add”** to add this function.



*Screenshot of the Add Script window*

The Function will now be pasted to the script.

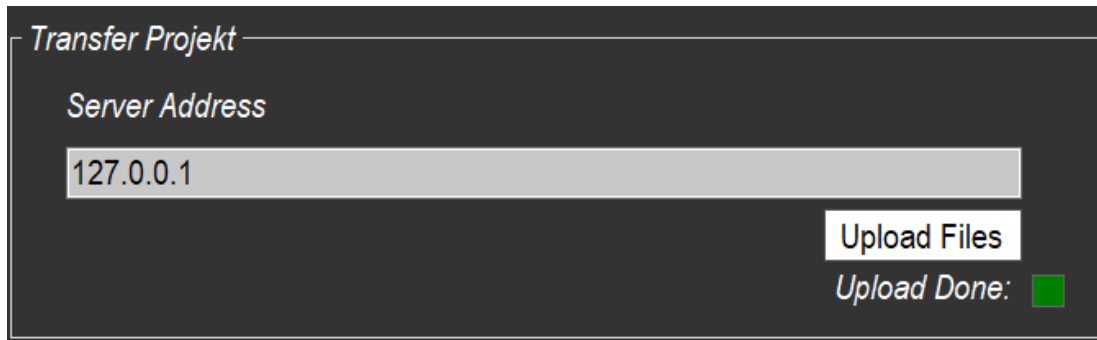


*Screenshot of the pasted function*

## 4.3 Upload



1. Under the icon “**Upload**” you can now transfer all the data to the server.



*Screenshot of the Upload window*

2. The Server will now print out the current time.

We finished the Quick start part, now it is up to you, to test the system.

Have fun. 😊