Target Setup for Professional Edition Raspberry Pi

This is a setup guide for the **CGI Studio Professional Edition** on the Raspberry Pi 4 or Raspberry Pi 5 with a Wayland display server. It covers the preparation of the target image and environment, the installation of the CGI Studio Debian package and the creation of a simple solution including a data binding script.

This target setup guide does not cover the optional **SoundHound integration**, which is used by the BloodPressureSample for voice interaction. Please check the Blood Pressure Sample for further information about it.

- Target Setup and First Steps

Target Setup and First Steps

Before you Begin

Before starting with the setup guide, we recommend the following to get familiar with CGI-Studio:

- **Explore CGI-Studio**

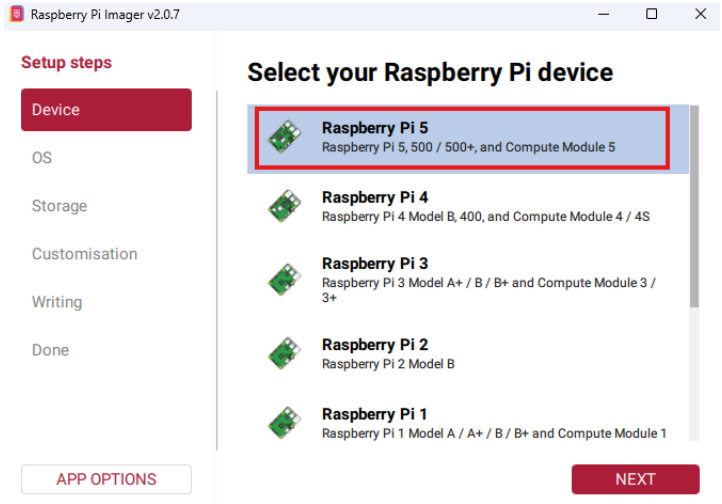
  Read the Quick Start Guide.
- **Learn to create SceneComposer Solutions**
  The documentation provides instructions on how to create simple SceneComposer solutions. To access SceneComposer navigate to:
  <cgi-studio-installation-path>\bin\SceneComposer\SceneComposer.exe
- **Generate Assets for Simulation and Target Environments**
  You can find guidance on generating assets from solutions (binary files containing solution information for the player) for both simulation and target environments in the documentation.

With this knowledge you should be fit to complete this guide.

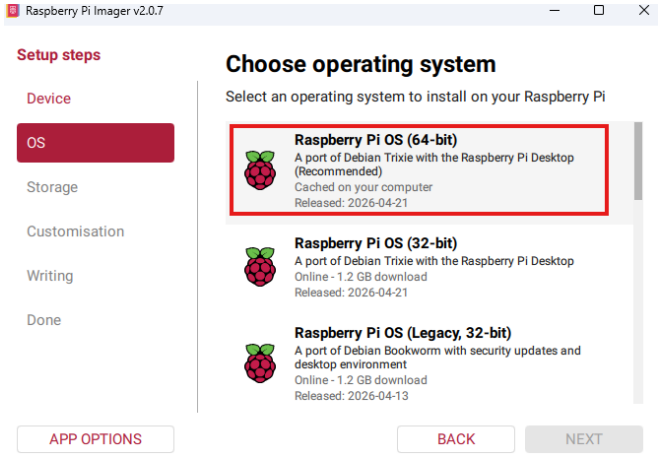
Software Image Setup

The CGI Studio Community Edition supports both Raspberry Pi 4 and Raspberry Pi 5. The following section describes how you can install the Community Edition on your Raspberry Pi device. Additional information about the proper setup and configuration of a Raspberry Pi device can be found in the official Raspberry Pi documentation. It also contains information on how to get started with your Raspberry Pi device. Complete the following steps to prepare your Raspberry Pi for the installation of the CGI Studio Community Edition.

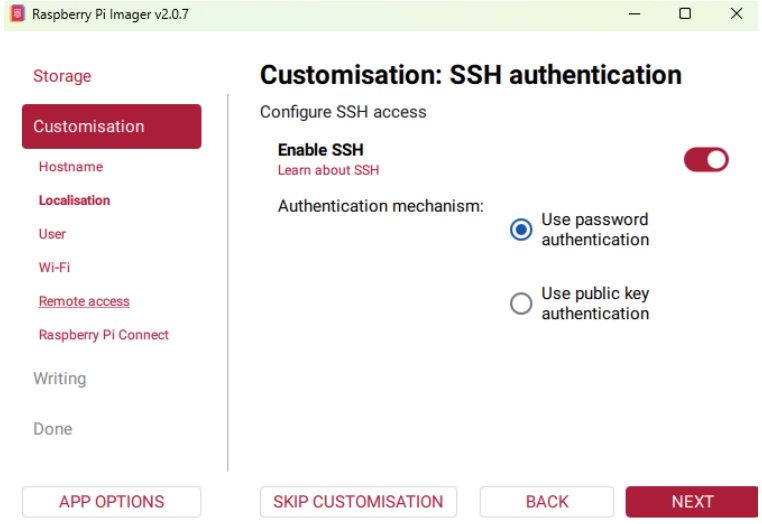
1. Download and install the **Raspberry Pi Imager** from this link: Raspberry Pi software – Raspberry Pi.
2. Insert a compatible SD card into your computer for flashing the image.
3. Open **Raspberry Pi Imager** and in the **Device** section select Raspberry Pi 4 or Raspberry Pi 5, depending on your model. Afterwards click "**Next**".



4. In the menu **Choose operating system** select "**Raspberry Pi OS (64-bit)**". Then click "**Next**".



5. In the storage section, select the inserted SD card, and click "**Next**".
6. In the **Customization** step click "**Next**" till the "**Customisation: SSH authentication**" configuration occurs. Make sure that **SSH** is **enabled**.



7. Click "**Next**" and write the image to the SD card. This may take some time.
8. Finally eject the SD card from your PC.

Raspberry Pi Startup

This section outlines the initial hardware setup, booting, and first-time configuration process for setting up the operating system on a Raspberry Pi 4 or a Raspberry Pi 5.

1. **Insert the SD Card:** Insert the flashed microSD card securely into the dedicated card slot on your Raspberry Pi board.
2. **Connect Hardware:** Connect your Raspberry Pi to an external display (using a micro-HDMI to HDMI cable), plug in your USB touch interface/peripherals or connect to DSI cable, and connect the official USB-C power supply to power it on.
3. **Initial Boot & Setup:** The OS will automatically begin its initial boot diagnostics. Follow the on-screen configuration wizard to set up your region, skip the Wi-Fi and connect LAN, and establish your core username and password.
4. **Boot to Desktop:** Once you complete the wizard, the system will finalize its settings and boot directly into the desktop user interface.
5. Once installed, the OS UI will appear on the display.

Connecting to the Raspberry Pi using SSH (WinSCP)

After booting, connect to the target device via SSH or a serial terminal (e.g., Tera Term). This connection is required to transfer files and execute commands. Additional information on how to connect can be found here.

At Candera we use WinSCP to transfer additional files to the device.

Run Blood Pressure Sample with Target Player

The Blood Pressure Device sample is a Scene Composer sample that demonstrates a blood pressure measurement device with exemplary scenes and animations. General information on this sample can be found in the Blood Pressure Solution documentation.

Execute the following command on the Raspberry Pi device to run the sample using our target player.

```
Player /opt/candera/Samples/BloodPressureDevice/BloodPressureDevice.bin
```

The asset will be visible on the target device.



Run Blood Pressure Sample with Player Connector Library

The Blood Pressure Sample sample is extended with a data binding programmed in Python. The Python part of the sample relies on the Python-Based Player Data Binding Library to interact with the data bindings of the scene. Once the hardware is set up follow these steps to configure the Python environment and execute the sample.

The Python code for the blood pressure device sample can be found under: *<cgi-studio-installation-path>\bin\CommunityEdition\Python\Samples\BloodPressureDevice* on the host computer and in */opt/candera/Python/Samples/BloodPressureDevice* on the **target**, after successful installation of the **debian** package.

Follow the steps below to run the Blood Pressure Sample with player connector library.

- **Move to the directory:** Open the terminal on your Raspberry Pi and move to the sample directory.

  ```
  cd /opt/candera/Python/Samples/BloodPressureDevice
  ```

- **Configure and activate the Python Virtual Environment:**
  Creating a virtual environment ensures that the dependencies for the Candera sample do not conflict with other system libraries.

  ```
  python -m venv .
  source bin/activate
  ```

- **Install Dependencies:**
  Install the required Python packages defined in the project's requirements file.

```
pip install -r ./requirementsWithoutSoundhound.txt
```

- **Run the Sample:**
  Execute the Blood Pressure Device python script using the following flags.
  - **-wsh** : Disable SoundHound integration
  - **-lbp** : Library binary path
  - **-abp** : Asset binary path

```
python BloodPressureDevice.py -lbp /opt/candera/lib/libPlayerConnectorLibrary.so
\
-abp /opt/candera/Python/Samples/BloodPressureDevice/BloodPressureDevice.bin -
wsh
```

Verification: Upon execution, the Blood Pressure Device UI should initialize and appear on your connected Raspberry Pi display.

NOTE: If the display is portrait mode then it needs to be rotated to landscape mode.
Command - `wlr-randr --output DSI-2 --transform 90`

Empty Solution with Slider and Gauge Control

This chapter shows how you can create your own solution with Python data binding through a simple use case, and how to run it on host and target. It is using a Gauge Control for HMI data output for the user and a Slider Control for HMI data input via user.

Before proceeding with the example, it is recommended to explore CGI-Studio by reading the Quick Start Guide, both of which are available in the CGI-Studio Documentation.

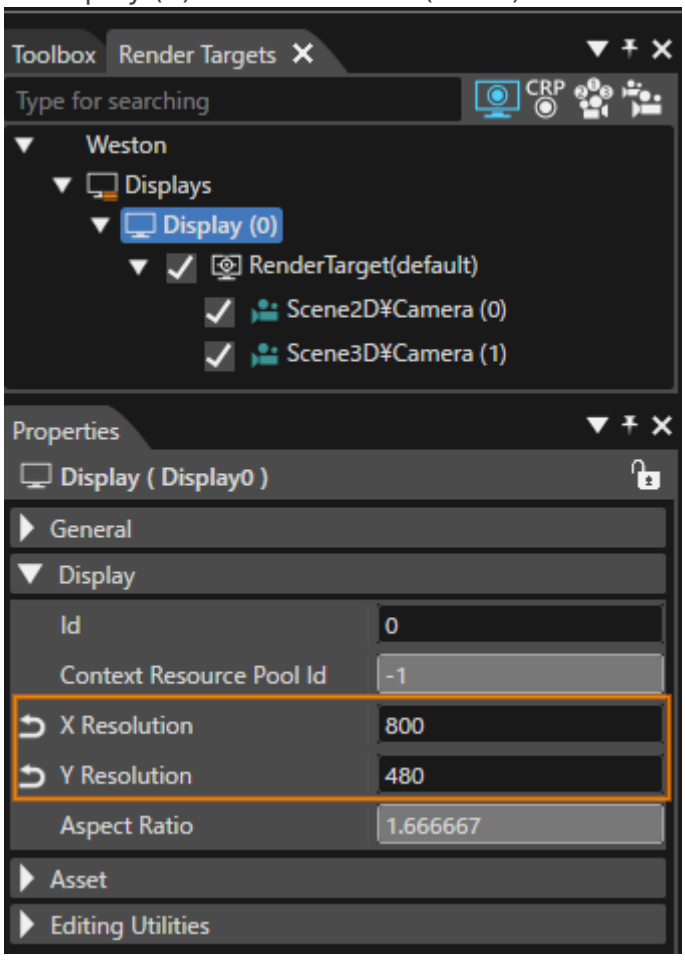
The documentation also contains information on how to create simple SceneComposer solutions.

- Start Scene Composer in *<cgi-studio-installation-path>\bin\SceneComposer\Scenecomposer.exe*

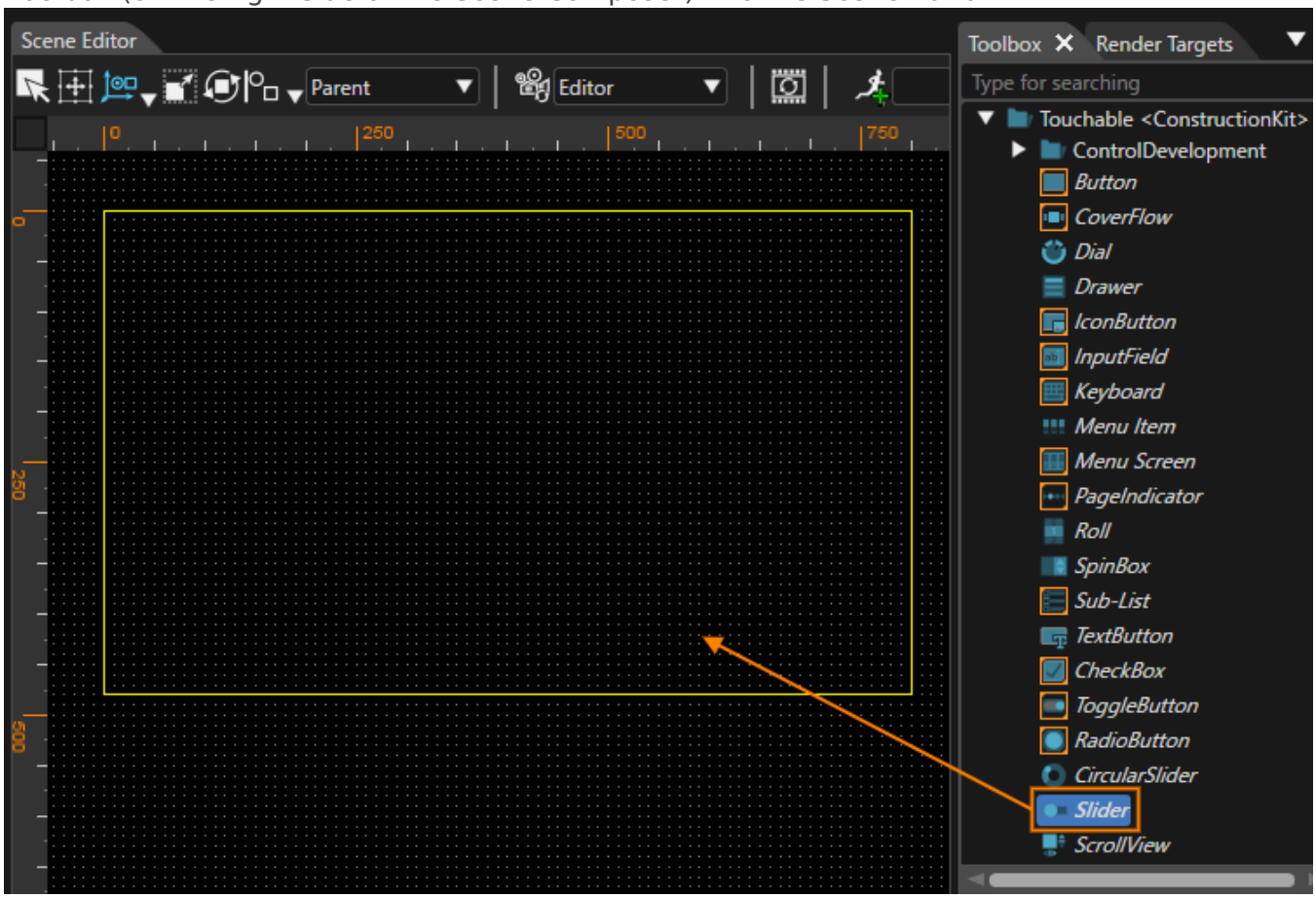
- Open Empty Solution by either double clicking the empty solution thumbnail or through the File > New Solution dialog.



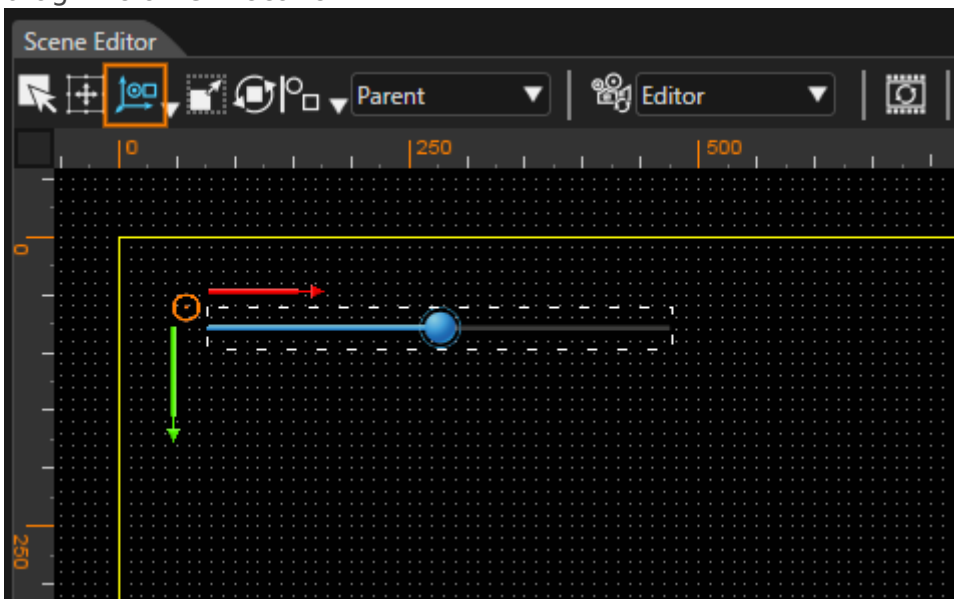
- In the Render Targets panel (on the right side of the SceneComposer), set the resolution of Display (0) to X Resolution (width) of 800 and Y Resolution (height) of 480.



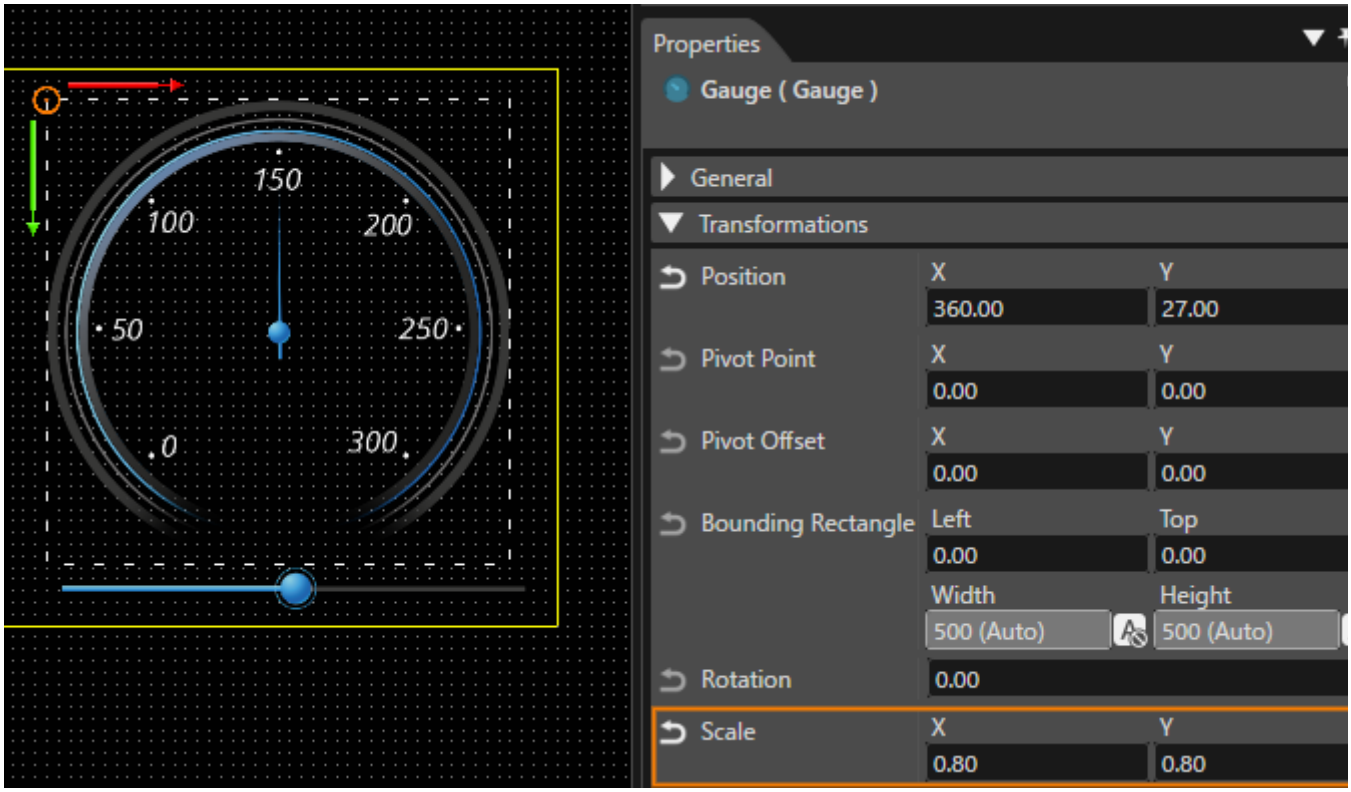
- Drag and drop a Slider Control and a Gauge Control from the Control section of the Toolbox (on the right side of the Scene Composer) into the Scene Editor.



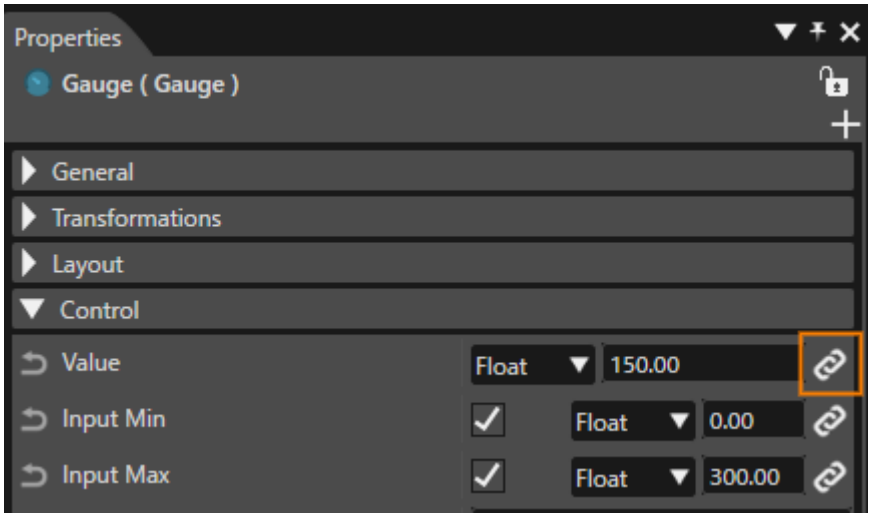
- You can move the controls in the Scene Editor by selecting the translation icon (the red square in the screenshot below) in the top left corner of the Scene Editor and then selecting the control which you want to move. Just select the item you want to move and drag it to a new location.



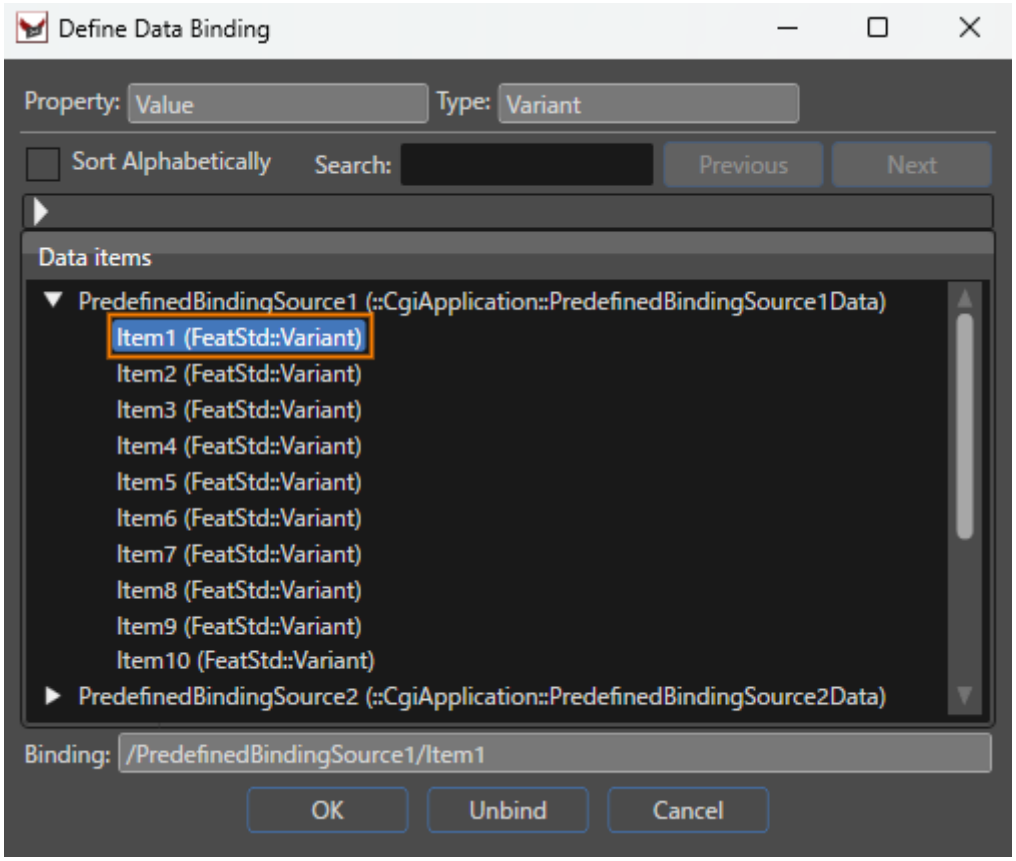
- Select the Gauge Control in either the Scene Editor or the Scene Tree, navigate to the properties, and adjust the scale to ensure it fits comfortably within the display.



- To set up a data binding for the Gauge value, first select the Gauge either in the Scene Editor or the Scene Tree. In the properties panel locate the Value property under the Control section and click the chain icon next to it to establish the binding.

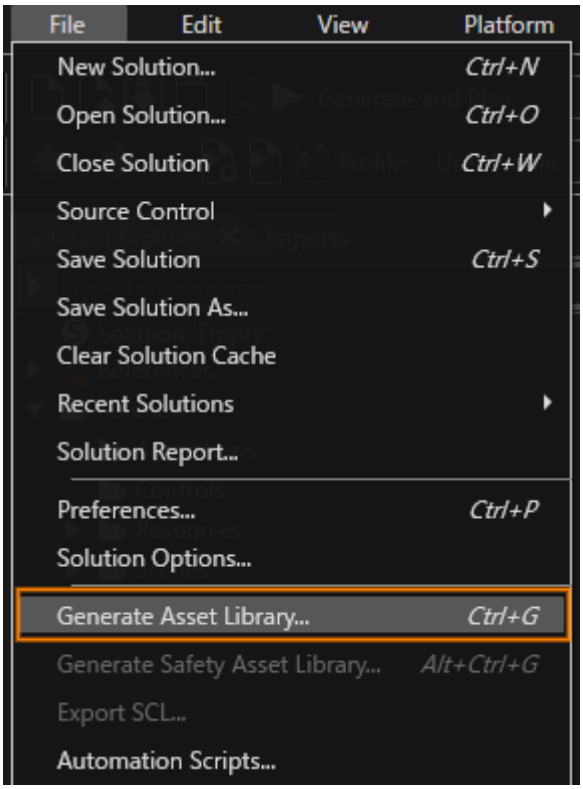


A "Define Data Binding" window will appear. In this window, select Item1 (under PredefinedBindingSource1). Finally, click OK at the bottom to confirm your selection.

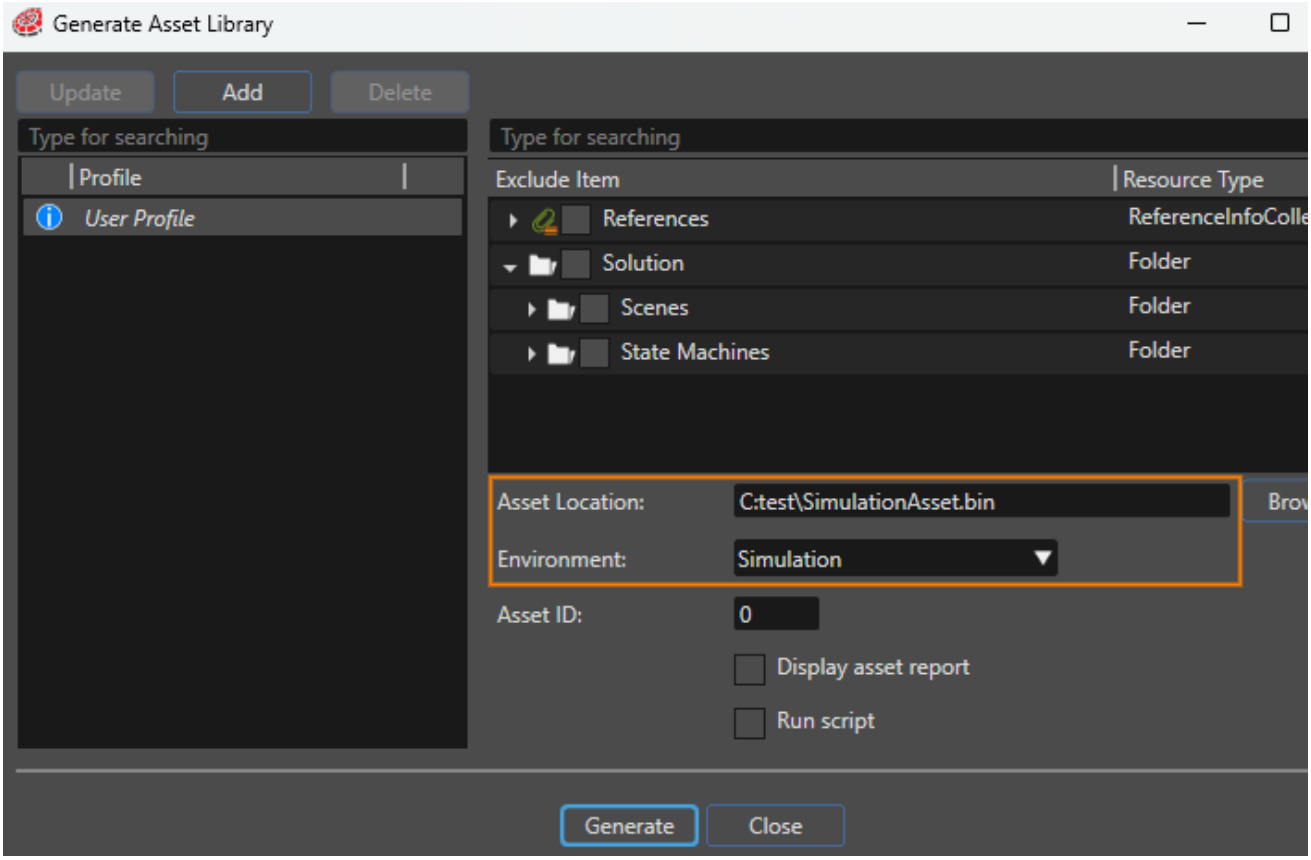


- To set a data binding for the Slider property, select the Slider control in either the Scene Editor or the Scene Tree. Then, click the chain icon next to the Value property in the Control section of the Properties panel. Next, choose Item2 (under PredefinedBindingSource1) from the available options.

- Next, generate an asset library for the target environment to run the solution in the player.
  - Save the solution, then go to File > Generate Asset Library



- A popup window will appear, allowing you to generate the asset library. Choose your desired asset location, set the environment to “Target”, then click “Generate” to create each asset library.



- Next, create a Python script. You can use the provided Sample.py file located in *<cgi-installation-*

path>\bin\CommunityEdition\Pythony\Samples\SliderAndGauge\Sample.py in your local CGI-Studio installation folder and in */opt/cgistudio/Python/Samples/SliderAndGauge/Sample.py* on the target.

- Run the asset on the **target device**:

  Ensure the device is set up correctly according to the chapter Software Image Setup. Use WinSCP or a similar tool to copy the generated target asset from the host computer to the target device.

  Run it analogous to the steps from chapter Run Blood Pressure Sample with Player Connector Library.