Palletizer 3D WinCC component manual

This component is designed for use with SIMATIC WinCC Unified V18/V19 software. The official documentation from Siemens can be found at the following [link](#)

Table of Contents

Description

A component for displaying your box stacking on a pallet in 3D, allowing you to rotate the model, change the parameters of the box and pallet, create your own configurations, save and load them, save an image in ".png" format, adjust the background color and its transparency, as well as certain camera parameters.

`The component is only used to display your recipes for stacking boxes on a pallet. And a configurator for creating recipes.`

`The component is not currently directly connected to the stacking robots. It cannot send a task to the robot or receive feedback from it on how to stack boxes.`

Limitations

Full functionality is supported in WinCC Unified PC RT (V18, V19 tested). Since panels have limited support for custom components.

Direct launch of the component in other systems is not possible. But for this purpose, saving the assembly in the `.png` format is provided. But for this, you will need to create these assemblies in `WinCC Unified PC RT`.

Usage

Installing

A step by step series of examples that tell you how to get a development env running

1. Create a new WinCC Unified project in TIA Portal and add a Unified PC station or Unified Comfort Panel
2. Close TIA Portal

3. Add the file {04F653D0-8999-4494-BDE7-1FDD159CBE57}.zip to the project folder:
   `YOUR_PROJECT_FOLDER\UserFiles\CustomControls`. if folder `CustomControls` does not exist, then
   create it
4. Start TIA Portal and open a WinCC Unified screen
5. Drag&Drop the "CustomWebControl" from the toolbox on the right under tab "My controls" into to the
   screen
6. Download and start the runtime locally (use simulation)
7. Open your browser and see the CustomWebControl

Folder structure

Here's a folder structure for a "WinCC 3D control" component:

```
{04F653D0-8999-4494-BDE7-1FDD159CBE57}/
|- assets/              # Folder used to store icons, images, etc.
|- control/             # Folder used to store main index.html and other requires
  |- config/            # Starter configurations for quick testing
    |- 1.json           # The first test configuration.
    |- 2.json           # The second test configuration.
    |- 3.json           # The third test configuration.
    |- 4.json           # The fourth test configuration.
    |- 5.json           # The fifth test configuration.
  |- jsm/               # Javascript folder
  |- index.html         # main HTML file for CustomWebControl
  |- main.js            # main JS file for CustomWebControl
  |- styles.css         # Component styles
|- manifest.json/       # Component manifest description.
```

To call the test configuration files for testing, you need to specify the path in the field as `./config/1.json`.

Properties

Name	Type	Description
Path	[String]	Path to the configuration file*
BackgroundColor	[Color]	The color of background
BackgroundOpacity	[number]	Transparency value for background from 0 to 1
DebugWindow	[Bool]	Pop-up window with camera position data
Configuration	[Bool]	Everything you need for the configurator
Admin	[Bool]	Provides access to saving a file, creating a screenshot and loading a configuration in the configurator mode
FixedView	[Bool]	Prevent the camera from rotating around an object
CameraOptions	[object]	Object with camera settings
- View	[String]	Select 1 of 9 object view options. Including "manual"**

Name	Type	Description
- ManualPosition	[String]	If "manual" is selected, enter the coordinates of the camera position in the format - 'x, y, z'.
- Distance	[number]	Distance from the camera to the object. Doesn't work if "manual" is selected.
- Fov	[number]	Camera frustum vertical field of view
Pallete	[object]	Pallet parameters
- Height	[Bool]	Pallet height
- Length	[Bool]	Pallet length
- Width	[Bool]	Pallet width
Carton	[object]	Carton parameters
- Height	[Bool]	Carton height
- Length	[Bool]	Carton length
- Width	[Bool]	Carton width
ConfigJSON	[String]	All the necessary parameters for saving to a file yourself `(READ ONLY)`***

- *Path - the path to the folder should be specified in form `\root folder\"your configuration files folder"\file.json`. Root folder must be defined in IIS Manager as virtual directory or you can use default path ~`C:\Program Files\Siemens\Automation\WinCCUnified\SimaticUA\"your configuration files folder"\file.json`.
- **View - list of available species:
    - front
    - behind
    - above
    - on right
    - isometric 1
    - isometric 2
    - isometric 3
    - isometric 4
    - manual
- ***ConfigJSON - Allows you to independently implement the function of saving parameters to a file. For secure access to files on the server. To access a file on the server, the component only needs to specify the correct file name and folder. The component does not have a mechanism for receiving data in raw form. Therefore, there is no need to independently read the file and pass data to the component.
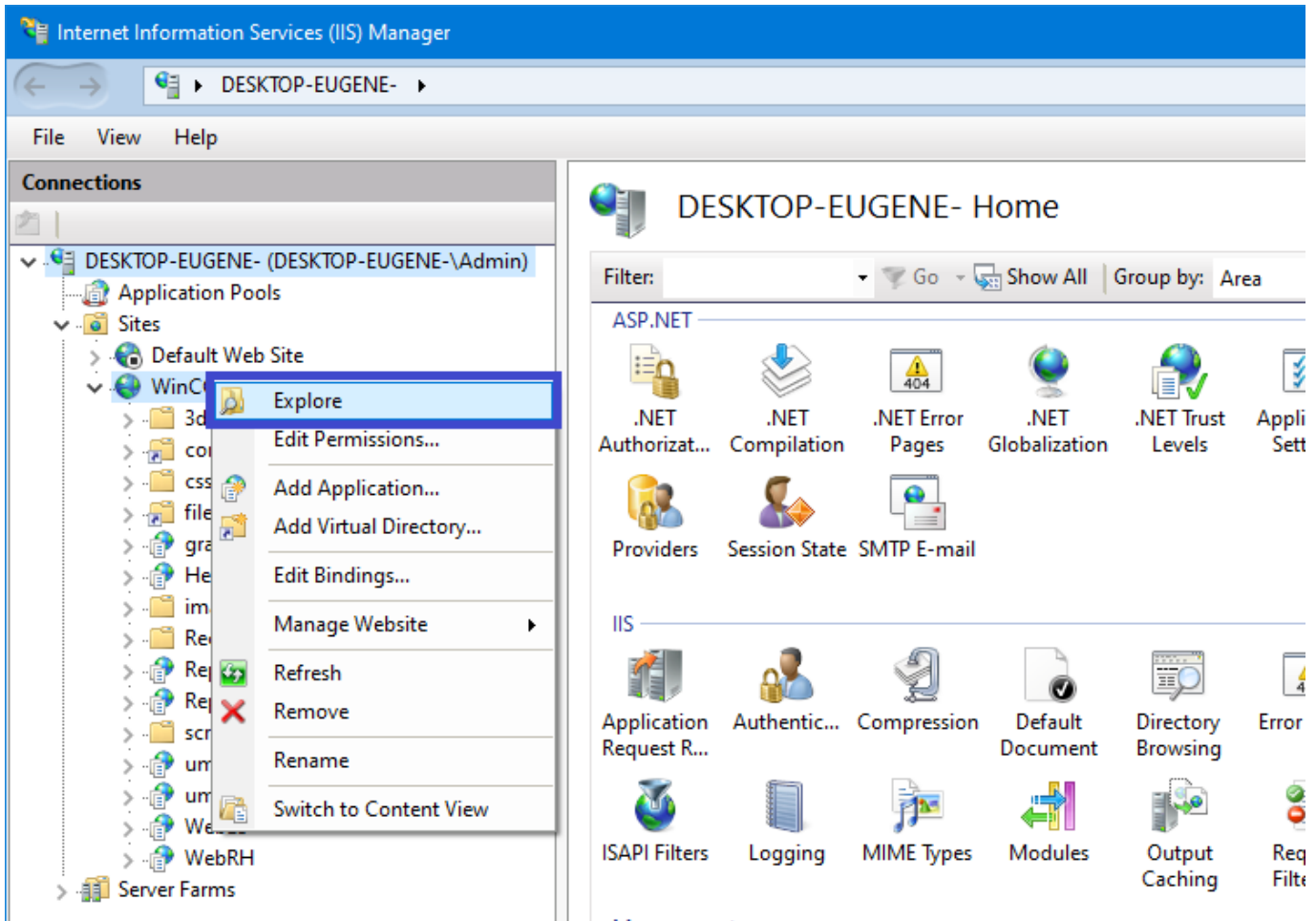
Path

Let's take a closer look at how to specify the correct path to our files. You must add `root folder` to `IIS Manager`. To do this, open `Control panel`
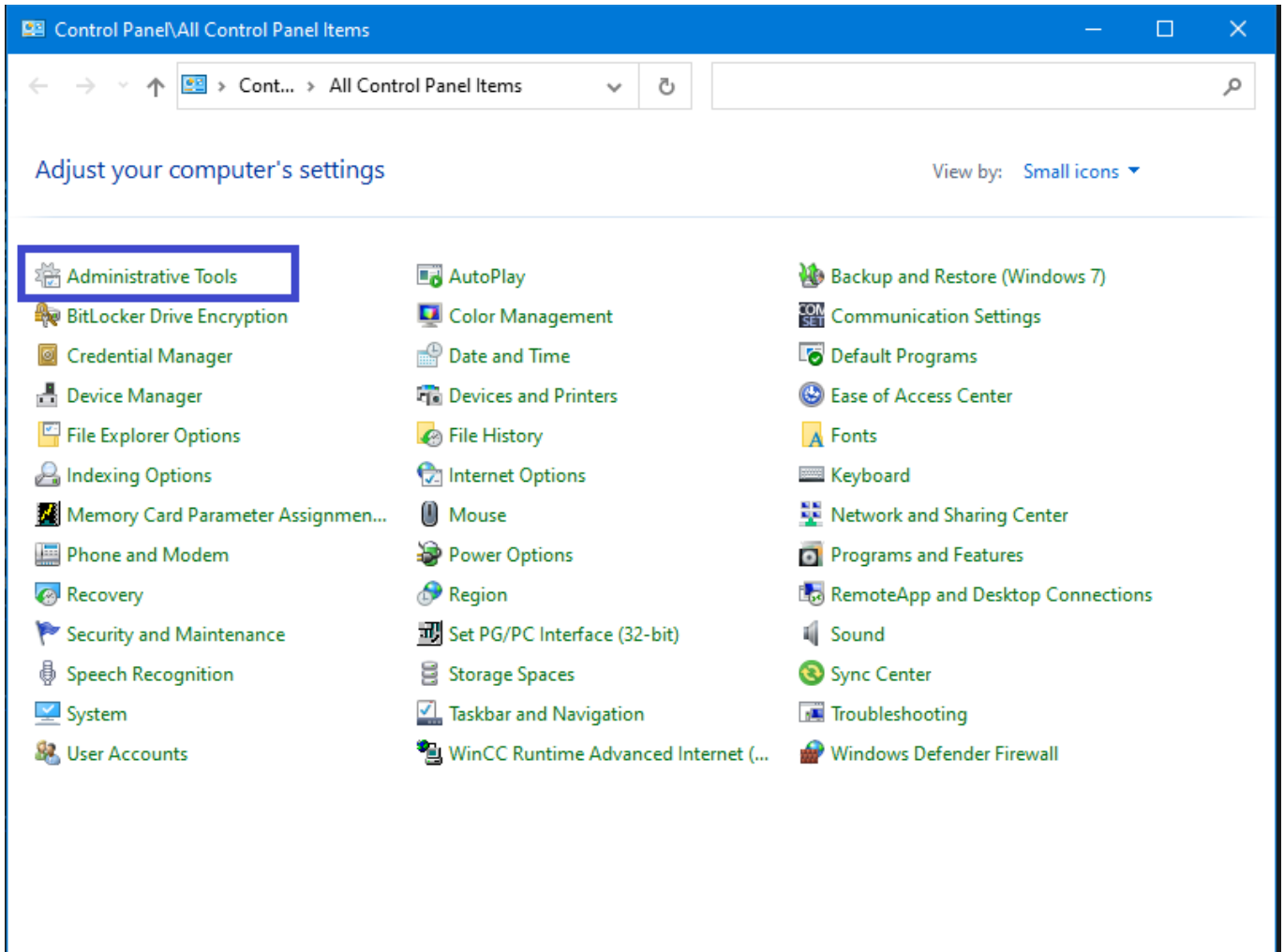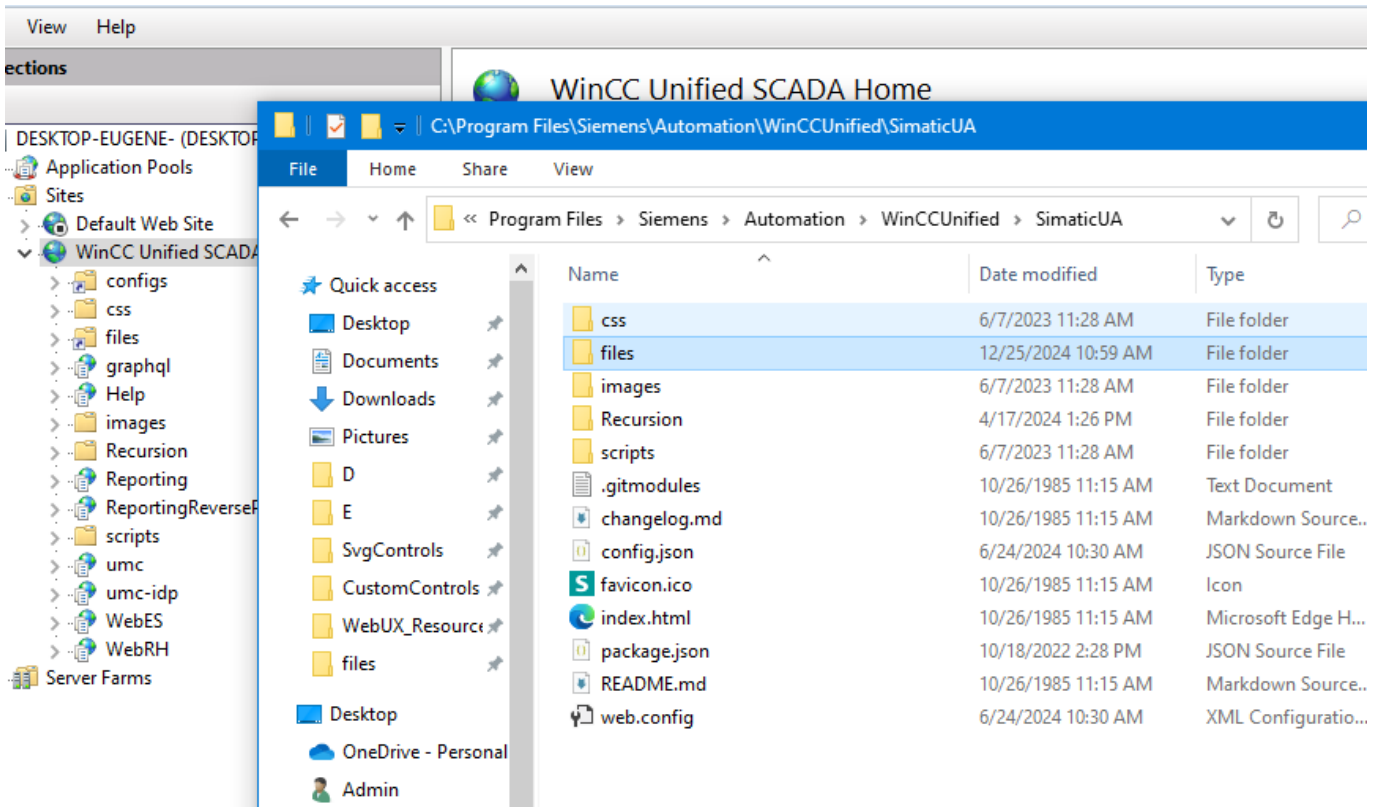
find `Administrative tools`, then select `Internet Information Services (IIS) Manager`.
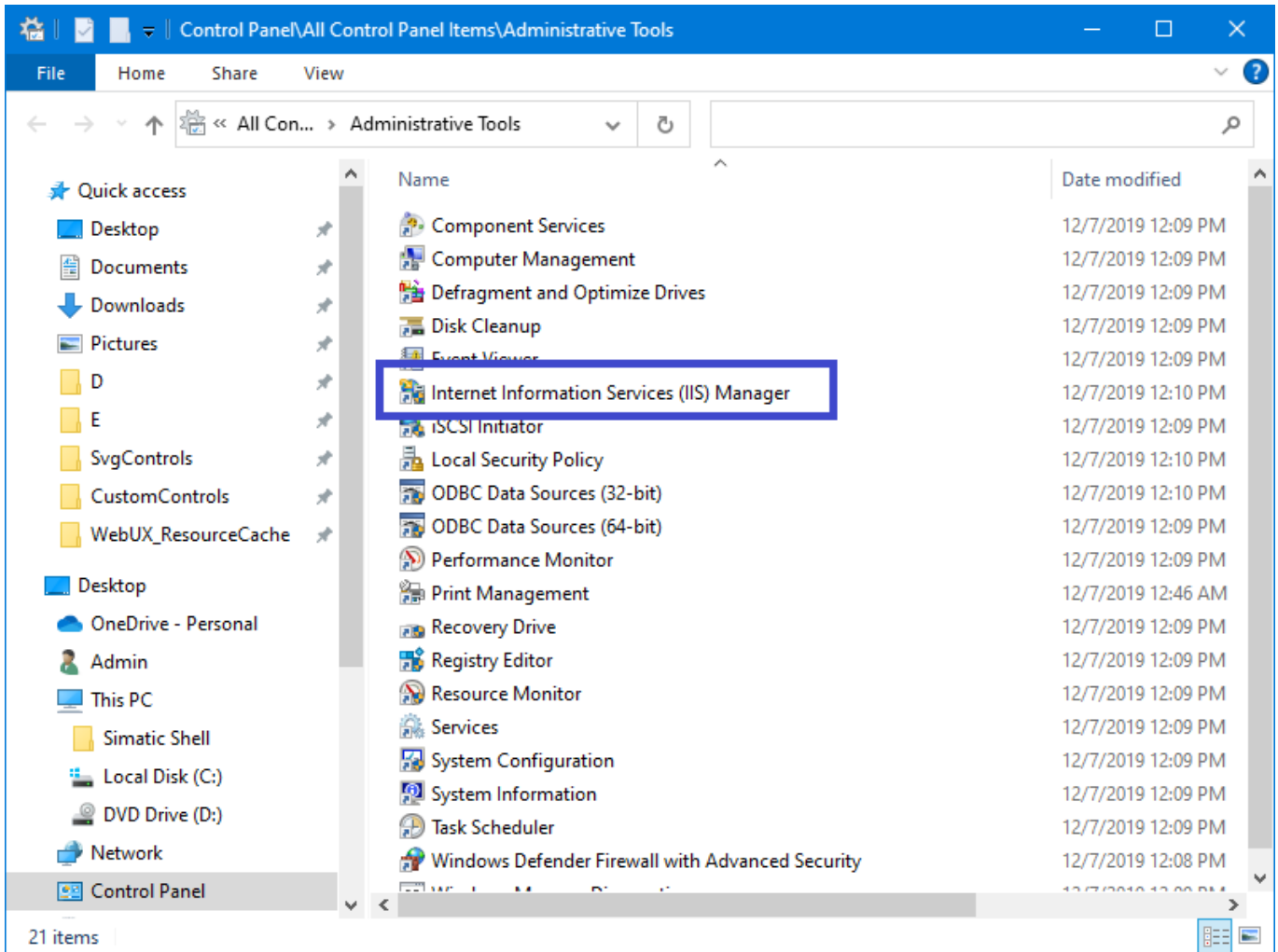
Then follow the drop down tabs to this path `Sites/WinCC Unified SCADA` and here we have two options. `The first one` is right click on this site, and select `Explore`.
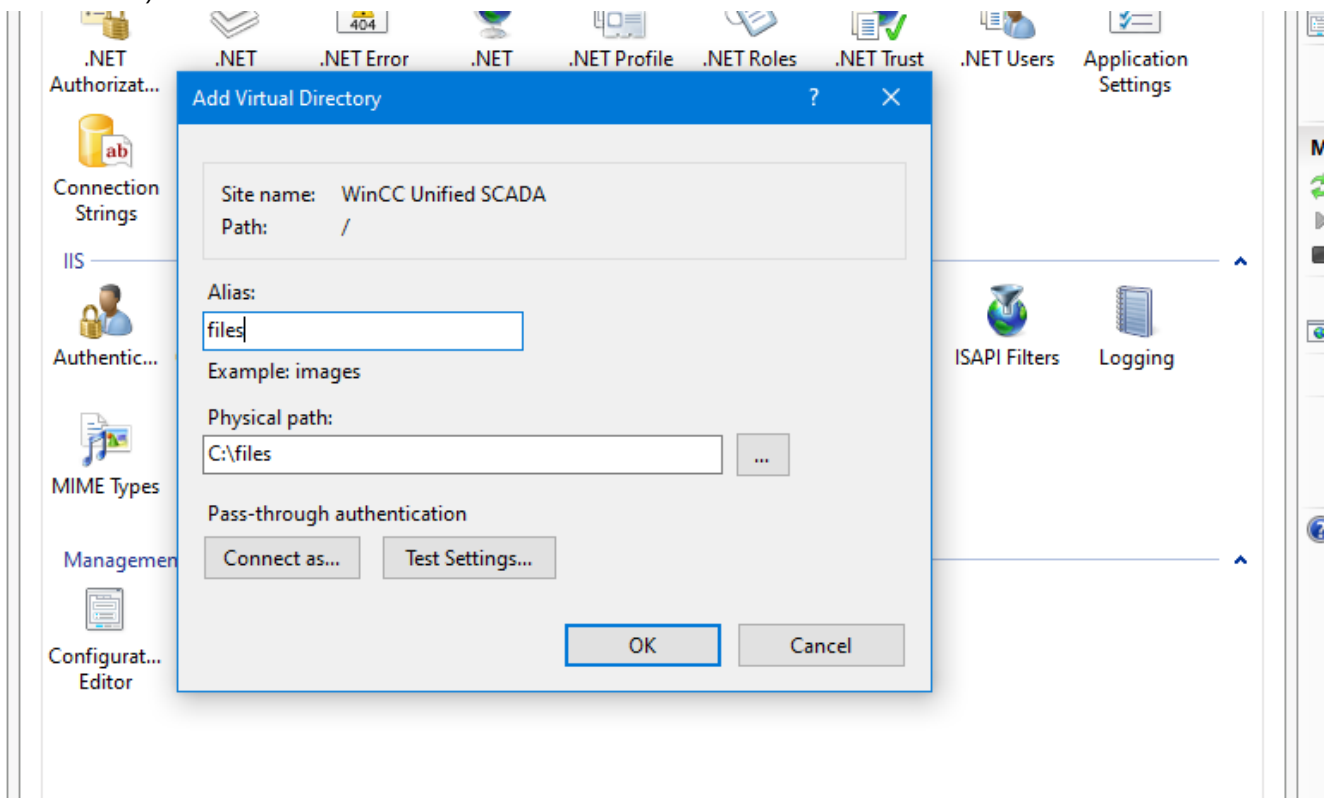
In the opened folder, add a new folder named `files` and place our configuration files inside it.

The `second` option is right click on this site, and select `Add Virtual Directory....`
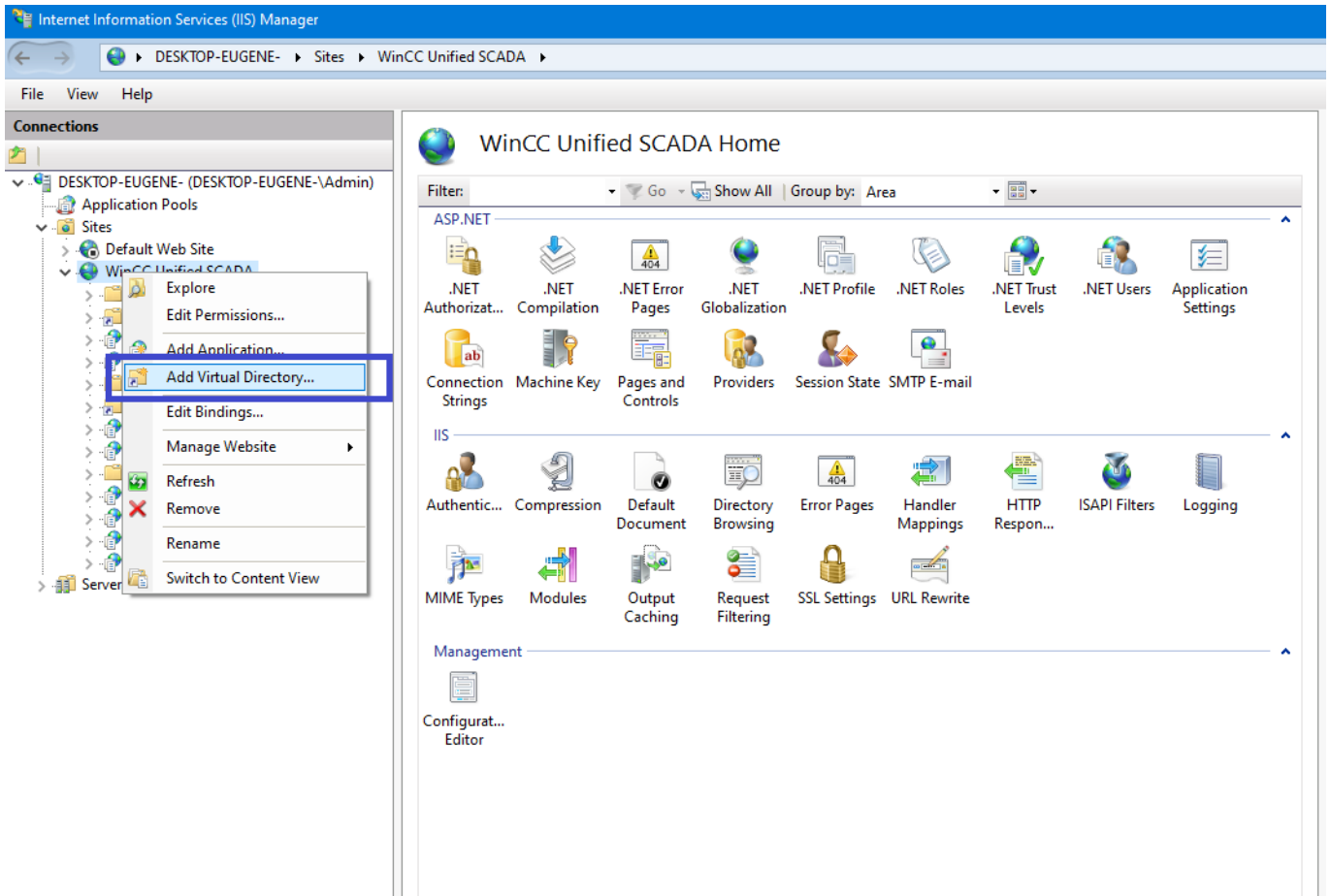
Select the physical path to the folder and give it a name (preferably leave the folder name so there is no confusion later).

This completes the configuration of accessibility of files from the shared folder for the component. After that the path to these files
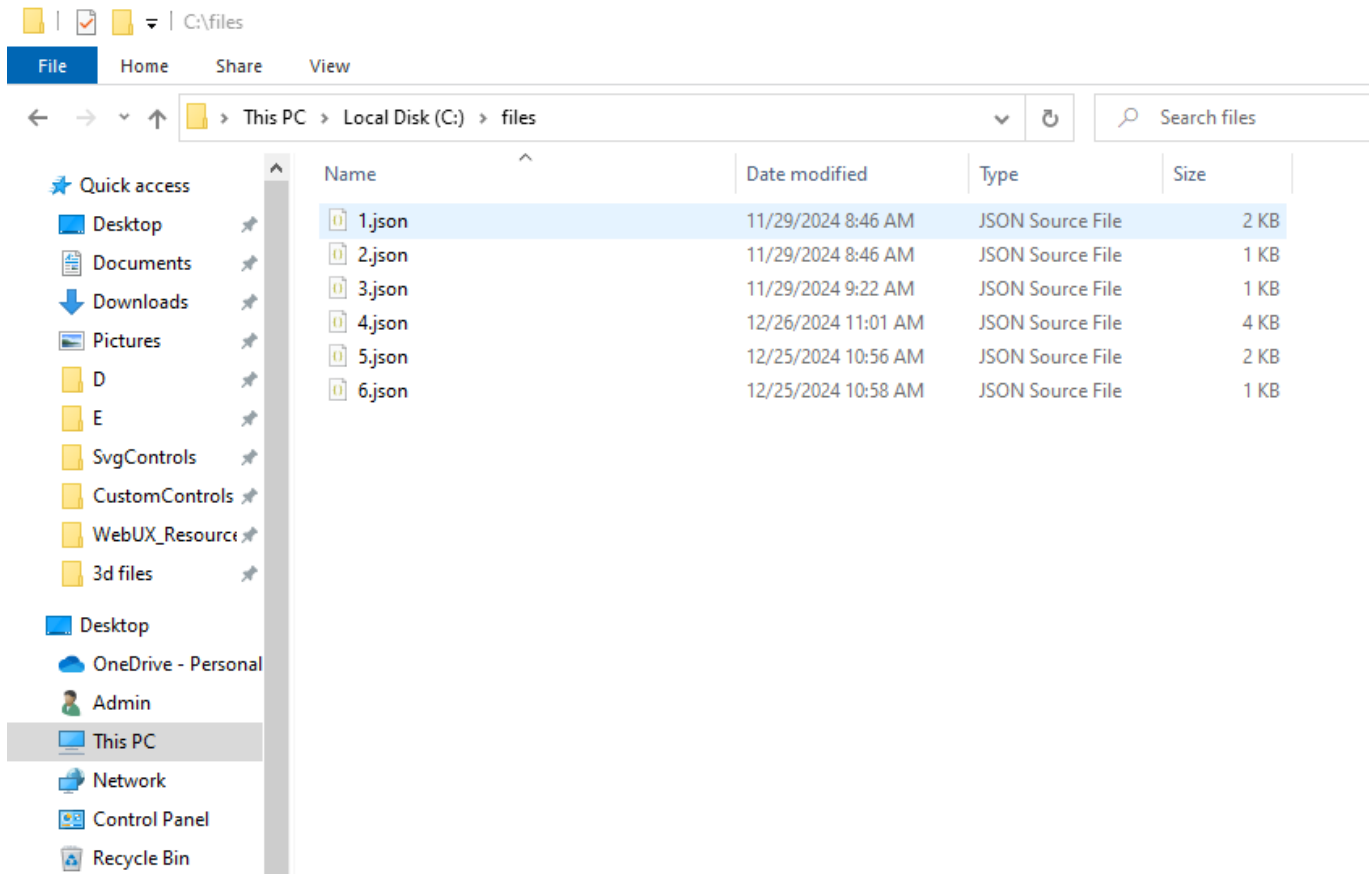
in WinCC Unified in both cases will look like this `/files/1.json`!
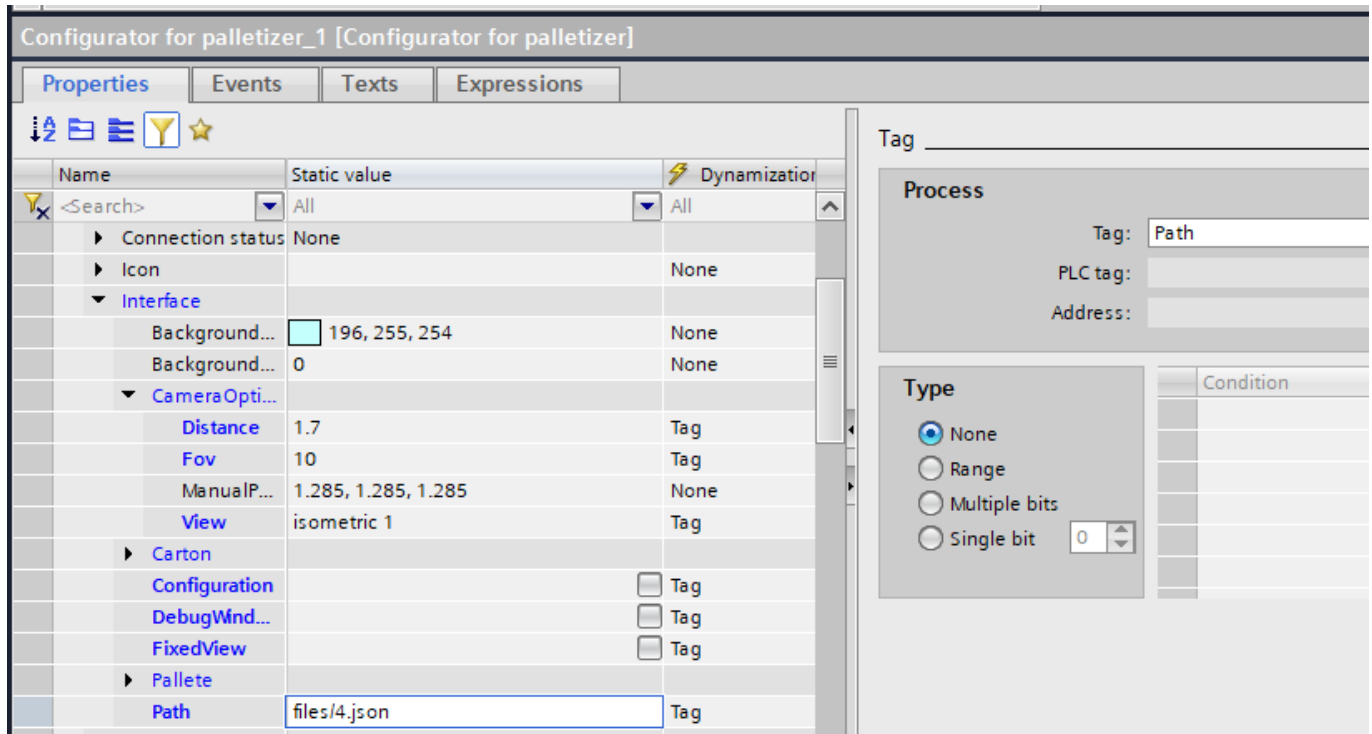
Configuration

The component has a configurator that will help you create your own layout of boxes on a pallet. After setting the parameters of the pallet and the box, you can start positioning each box.

Control interface

The following interface has been created for mouse control.

- Left mouse button - add an object to the pallet.
- Right mouse button - remove the selected object from the pallet (highlighted in green).
- Holding the middle mouse button - rotates the camera.
- Mouse roller - zoom in.

Name	Description
left	Rotate the box to the left (also possible by pressing the left arrow)

Name	Description
start	Return the box to the initial rotation parameters
right	Rotate the box to the right (also possible by pressing the right arrow)
chooseFile	Loading a configuration file from the user's system
saveConf	Saving a configuration file to the user's system
capture	Creating an image of your configuration in `.png` format
+	Adding a row of boxes (also possible by pressing the up arrow)
number	Number of rows with boxes (read only)
-	Removing a row of boxes (also possible by pressing the down arrow)

Creating the first recipe

1. The provided project already includes a fully prepared configurator, which you can modify as you wish. You can also remove linked tags from certain fields and specify parameters if they are not subject to change (e.g., box or pallet dimensions, fixed camera, FOV).

2. Start by setting the parameters for the box and the pallet. You can define them in any unit system, with the only requirement being that all dimensions must use the same system.

   In case these parameters are changed, the boxes already placed on the pallet will be removed. Please be mindful of this.

After setting the parameters, we should see the following:

3. Next, I start placing the first row of boxes. To place a box, I position it in the desired location and click the left mouse button; to delete it, I use the right mouse button. To rotate the box, I press the "left" or "right" button, and to rotate the model, I hold down the middle mouse button.
And just like that, we've formed the first row!

4. Now, by pressing the "+" and "-" buttons, we can change the number of rows on the pallet. In my case, it will be 7.
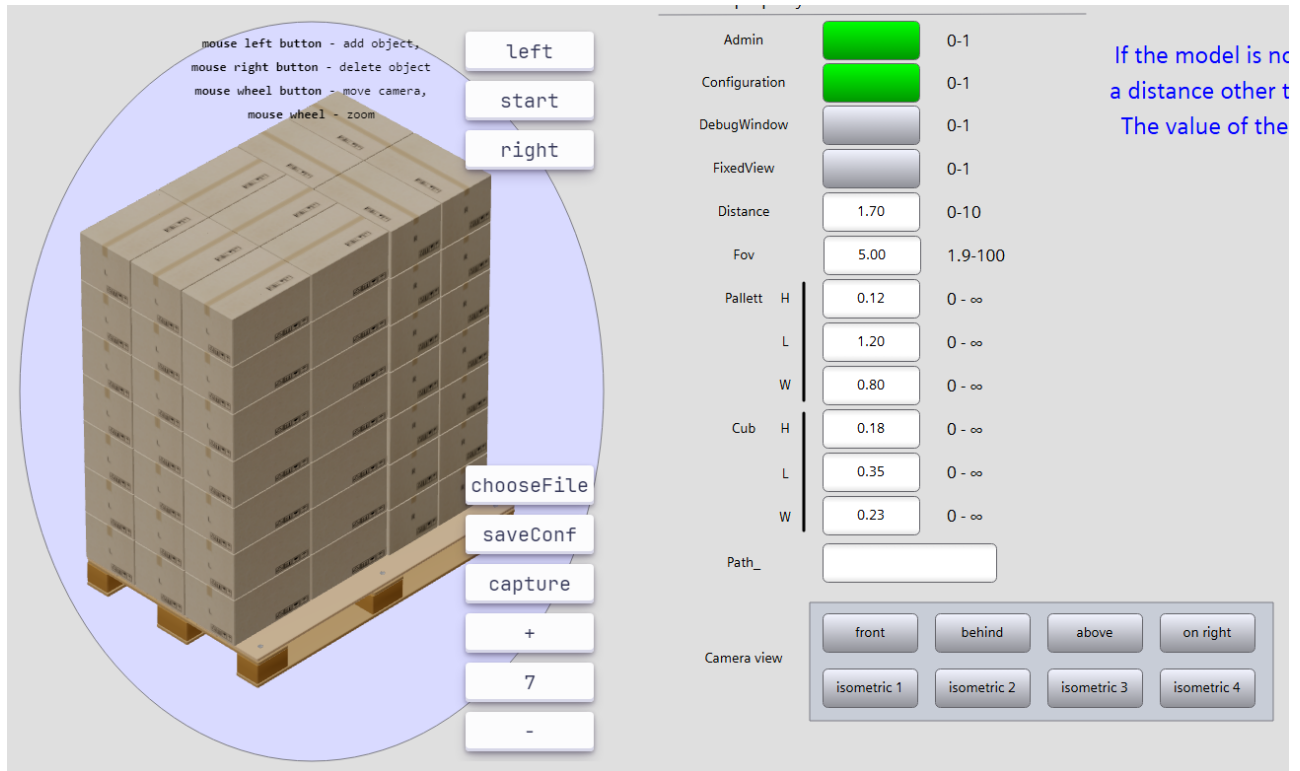
5. Now you can choose a view that suits you and save the recipe. If such a view is not available, you can set it manually.
   To do this, switch to "manual" mode, adjust the camera to your preference, and then save the recipe. In this case, the camera's position parameters will not only be saved in the recipe but also applied when the recipe is loaded.
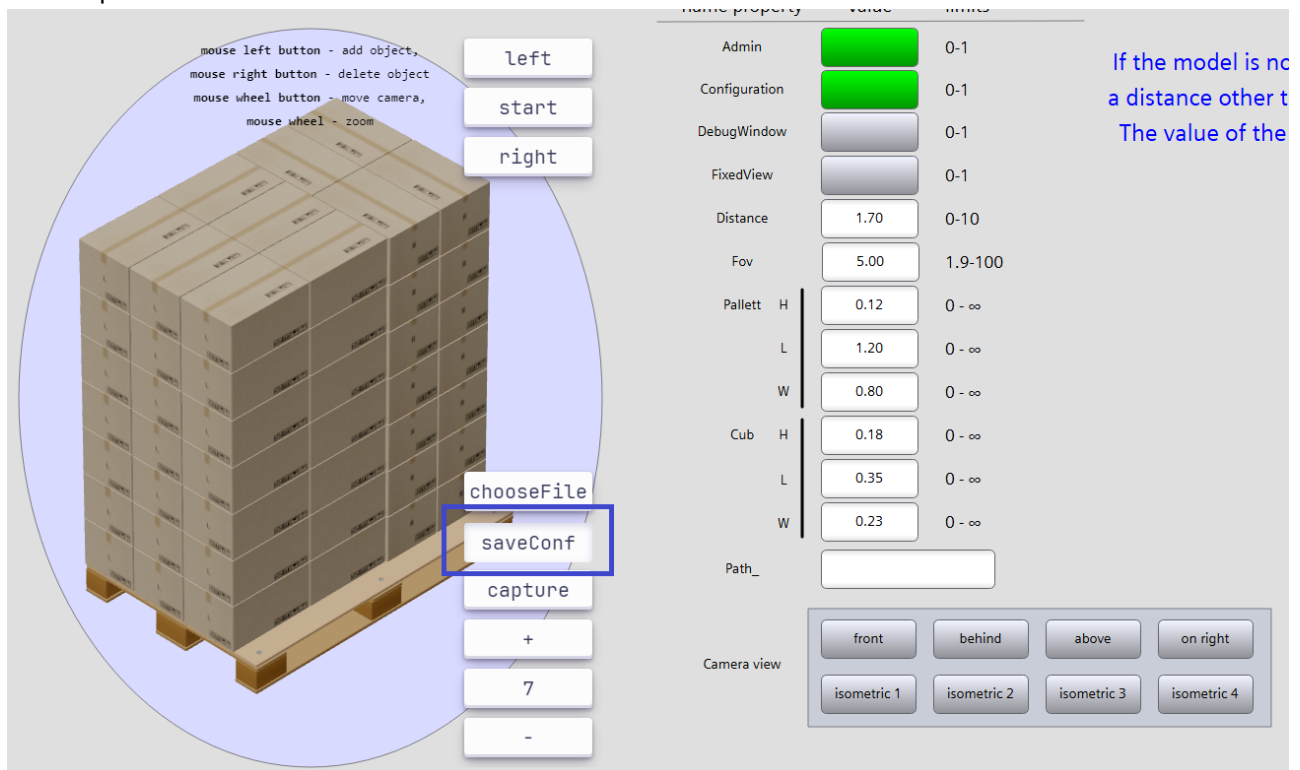
6. Alternatively, if your device cannot use the component directly, you can save an image by clicking on "capture".
   You also have the option to implement saving to a file yourself. For this, the component has a parameter called "Properties.ConfigJSON", which contains the current recipe parameters. You can retrieve these parameters and save them to a file on your server.

7. A bit about how the camera works.

   When selecting predefined display parameters (such as "Isometry 1"), the object is measured by its height, width, and length. Using trigonometric calculations involving `Properties.CameraOptions.Fov**`, the result is multiplied by `Properties.CameraOptions.Distance` (a relative value representing how far the object is from the camera—the larger the value, the farther the object).

   After these calculations, the object is centered within the component. This behavior is particularly noticeable when you start adding a large number of rows. Regardless of the number of rows, the pallet will always fit within the window.

   FOV - Camera frustum vertical field of view. The smaller the value, the less perspective distortion of the object, but performance may decrease. The standard value is approximately **55-65**.

   However, it is not recommended to set the value below **1.9** or above **80**, as this can lead to various artifacts.
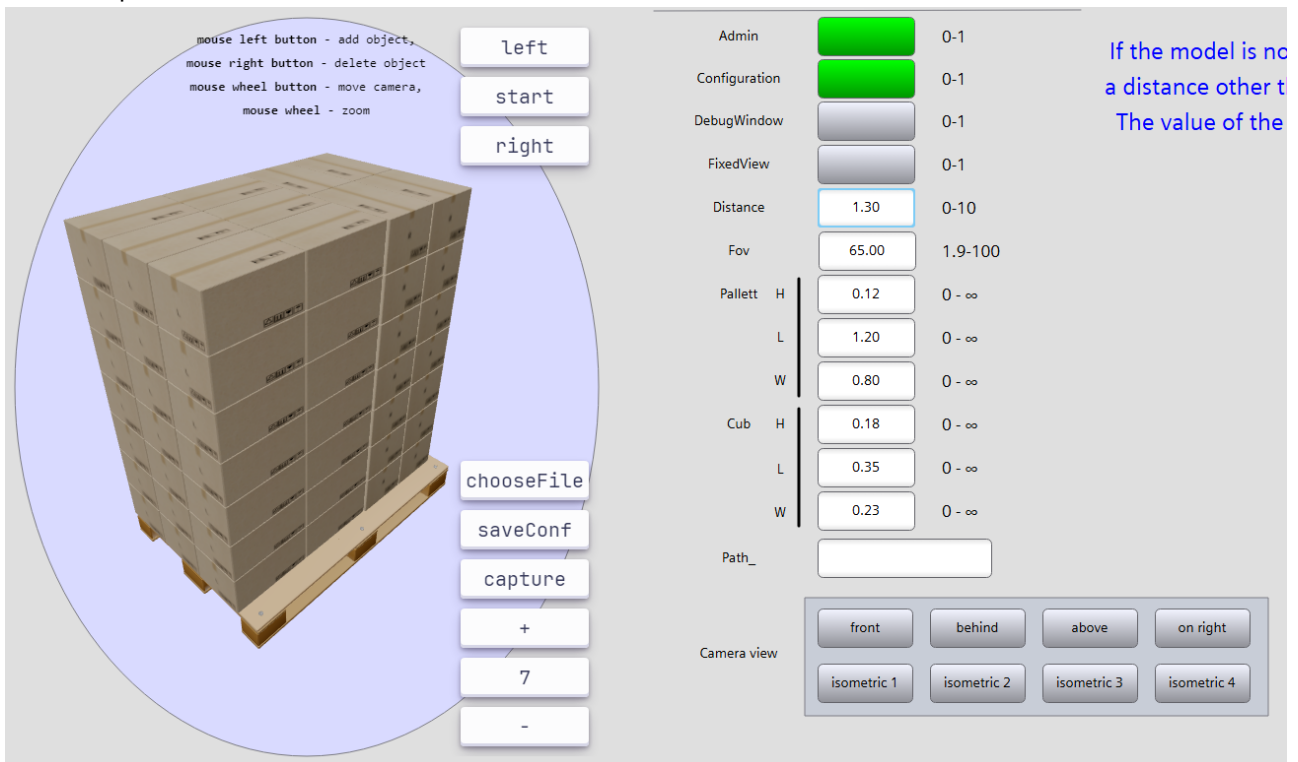
   For example, here **FOV** is set to **5**, **Distance** is set to **1.7**.
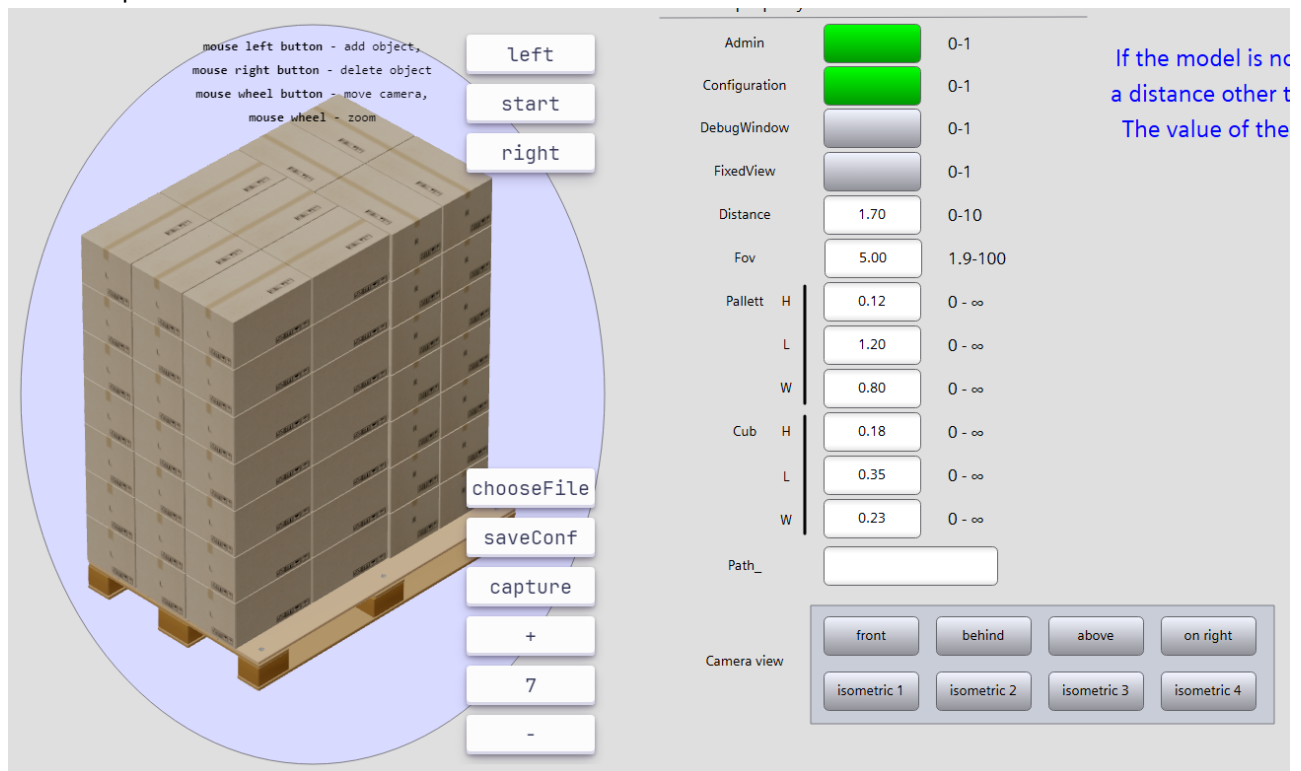
For example, here **FOV** is set to **65**, **Distance** is set to **1.7**.
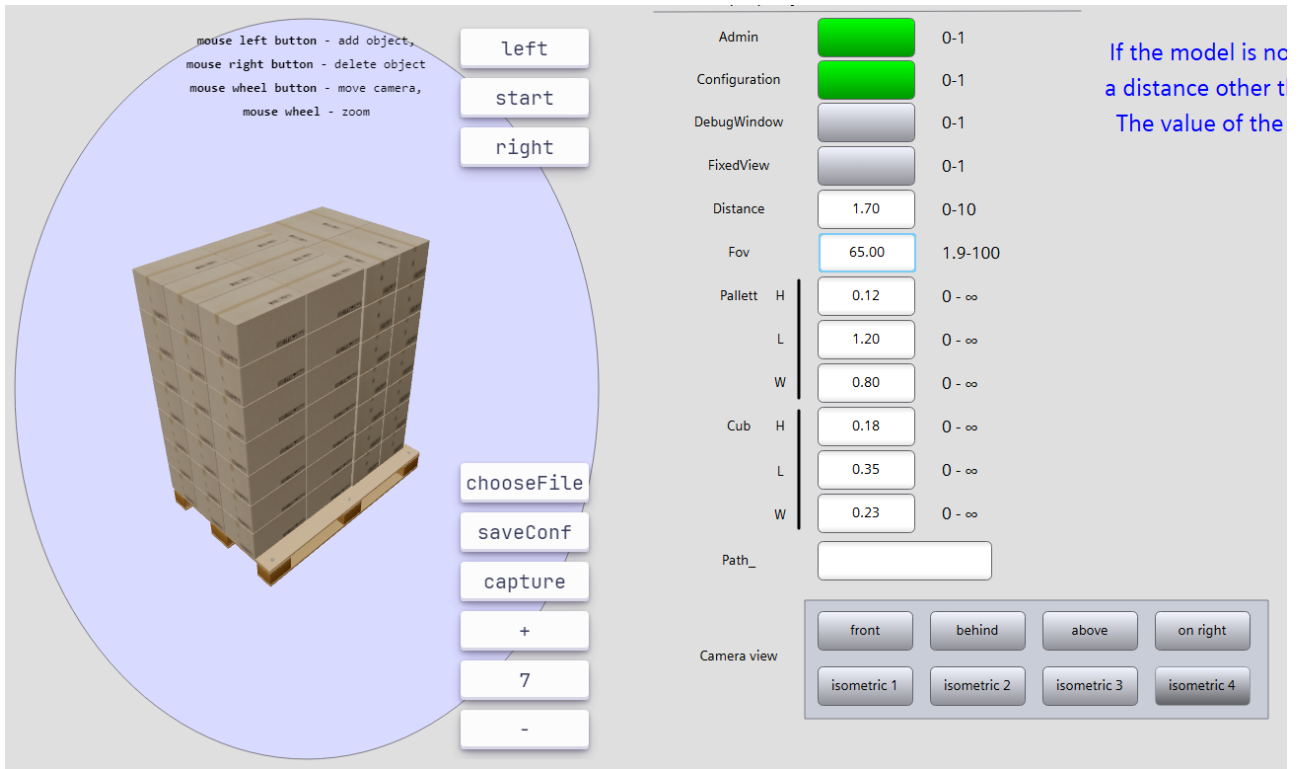
For example, here **FOV** is set to **65**, **Distance** is set to **1.3**.

manual - In this case, after the calculations, the object will simply be centered within the component, but the distance to the camera will not change. The camera position specified in its parameters will be used. You will notice that when adding a large number of rows, the entire object won't move further away; instead, it will just be centered.

Experiment with different values to find the one that suits your needs best.

8. To use predefined recipes, we need to specify the path to the file in **Properties.Path** (see the example above in Path). The **Path** parameter can be changed at any time, which will result in a change of the configuration.
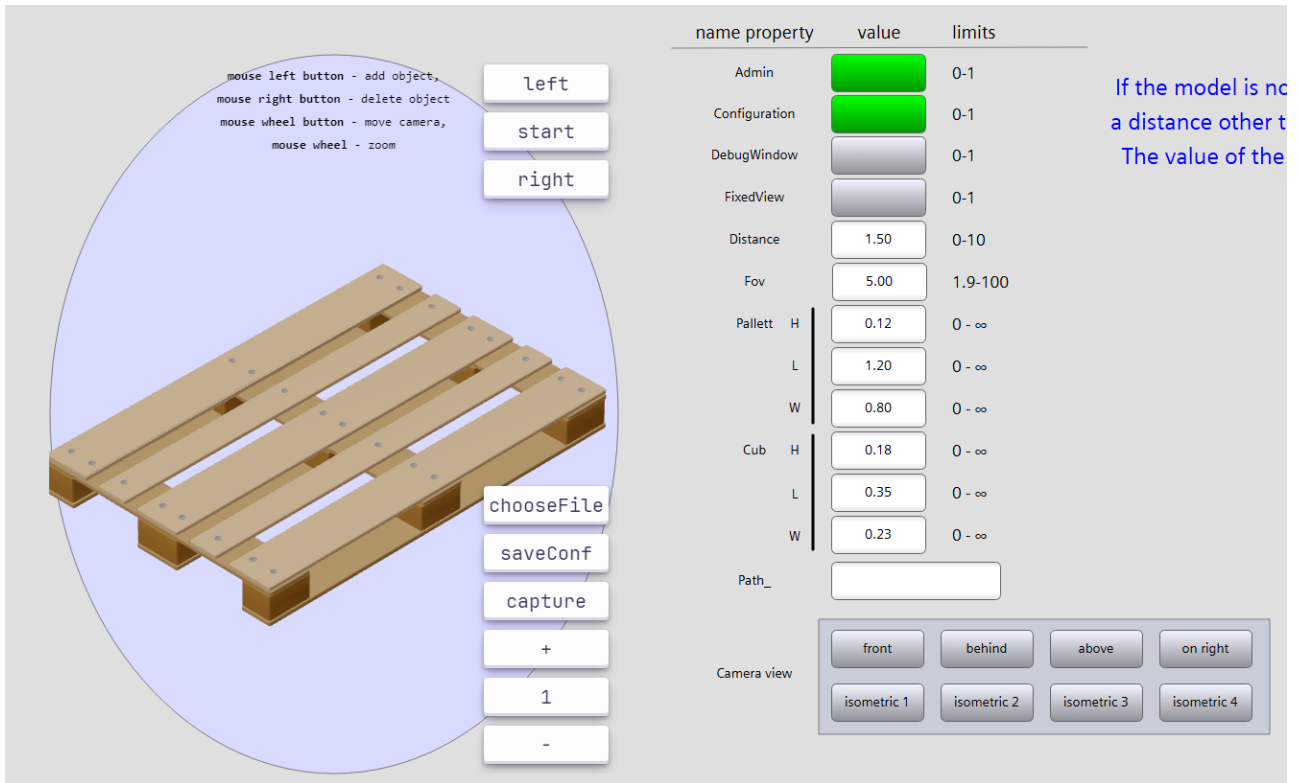
To call the test configuration files for testing, you need to specify the path in the field as ./config/1.json.
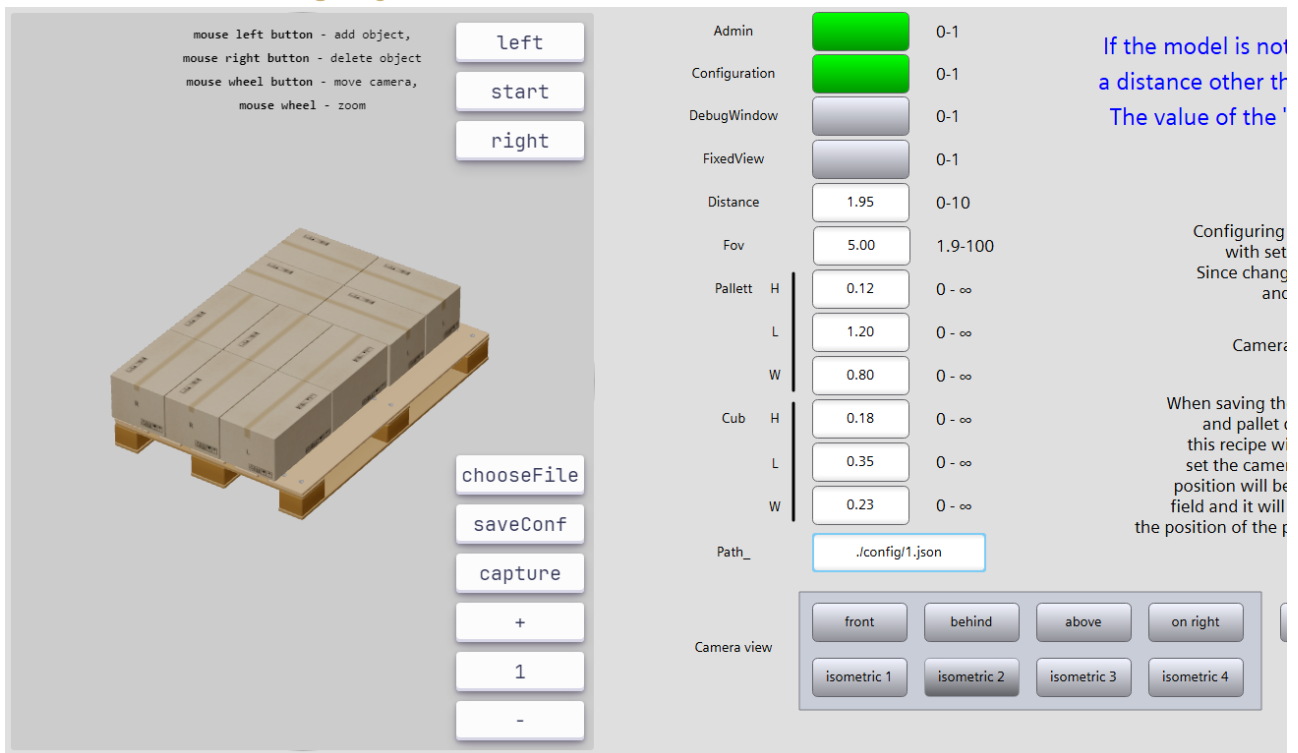
Troubleshooting

- To track some problems, press F12 in the browser, go to the "console" tab and refresh the page.

- If you see an error related to file access, make sure that you have added the folder to your server in `"IIS Manager"`.

Changelog

Version [1.4] - 2024-12-27

Added

- The `Admin` has been added.
- The `ConfigJSON` has been added.

Fixed

- Improving performance.

Version [1.3] - 2024-12-26

Added

- The `configurator` has been added.
- The ability to change pallet and carton parameters has been added.
- The ability to load `configuration files` has been added.

Changed

- The image rendering functions have been reworked for improved optimization of the component's performance, not only in the browser but also in the TIA Portal screen.

Fixed

- Issues with loading configuration files have been fixed.

Version [1.2] - 2024-12-04

Added

- Added ability to select `view` from the list.
- Added possibility to set the camera to the desired position `manually`.
- Added ability to open a `DebugWindow` to track camera coordinates.
- Added ability to change background transparency.

Changed

- Changed the principle of calculating the camera position for correct display of 3D objects of any size.

Fixed

- Performance improvement and optimization.

Version [1.1] - 2024-11-15

Added

- Added possibility to display an object in isometry.

- Added ability to rotate an object on the scene.

Changed

- Changed the principle of object lighting.

Fixed

- Fixed a bug that could result in multiple objects on the scene.

References

- SVGHMI.pro
- Component page