Insert Tags To Displays Reference

Link live tags to display elements.

This page is a guide on using the "Paste Tags" feature on displays. The feature allows for direct integration of solution tags into displays, streamlining the configuration process and enhancing data visualization. Additionally, we discuss how to copy and paste tags onto symbols, emphasizing that when copying and pasting a tag from a custom type that shares the same name as a symbol, the symbol will be pasted with the tag already configured.

Pasting Tags to Displays
- Create or Select Tags in the Solution
- Copy Selected Tags to Clipboard
- Paste Tags in the Draw area

Paste Tags with Symbols
Using Assets in Displays

Pasting Tags to Displays

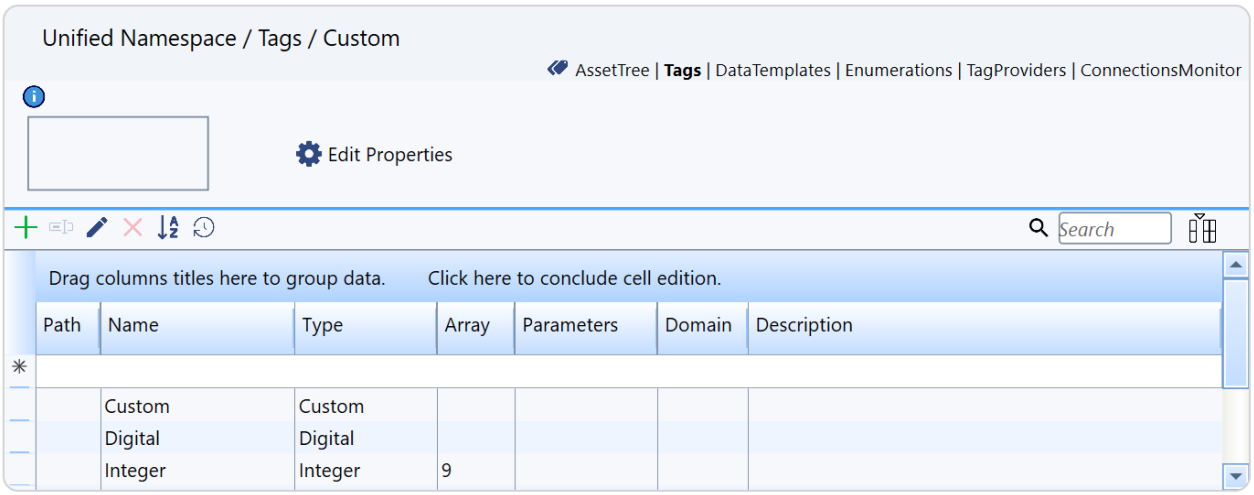
This section covers the process of creating or selecting tags, copying them, and pasting them directly into the display area. It also explains how to automatically format simple tags, arrays, and custom types, as well as how to paste tags onto symbols with matching names, ensuring accurate and seamless tag integration within the display environment. Topics include tag creation, copying and pasting tags, formatting different tag types, and handling symbol integration.

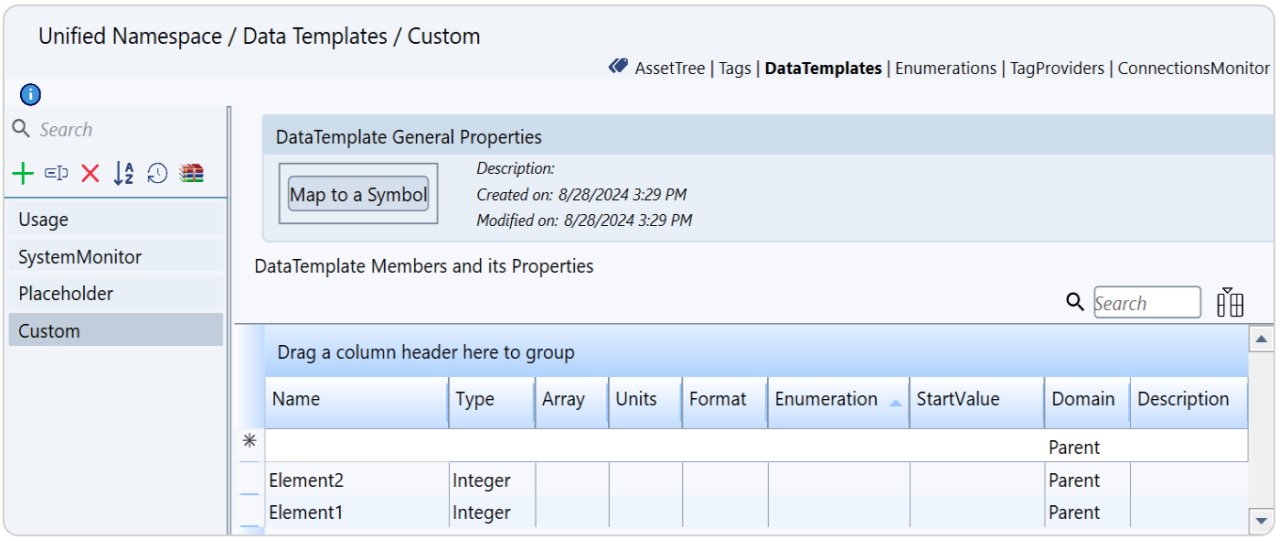
Create or Select Tags in the Solution

The first step to use this feature is to create a Tag in your Solution. This tag can be a simple one, an array or Custom Type.

To better understand the feature, three examples are shown below. The tags created are as follows:

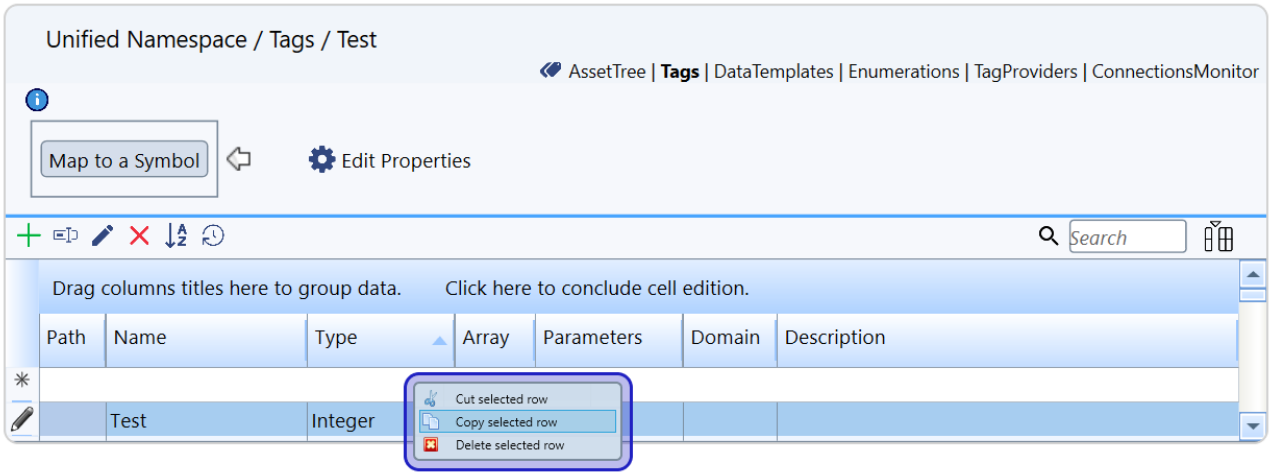
```
1. Tag.Integer -- Integer DataType -- Array of 10 elements.
2. Tag.Digital -- Digital DataType -- Simple Tag.
3. Tag.Custom -- Custom DataType --
Sub-Elements: Element1 (Integer), Element2 (Integer.)
```





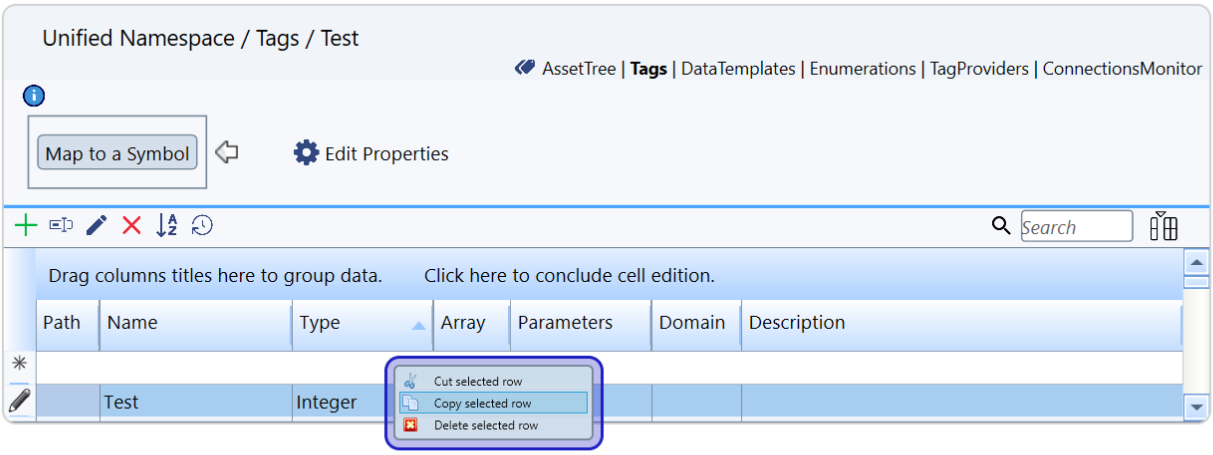
Copy Selected Tags to Clipboard

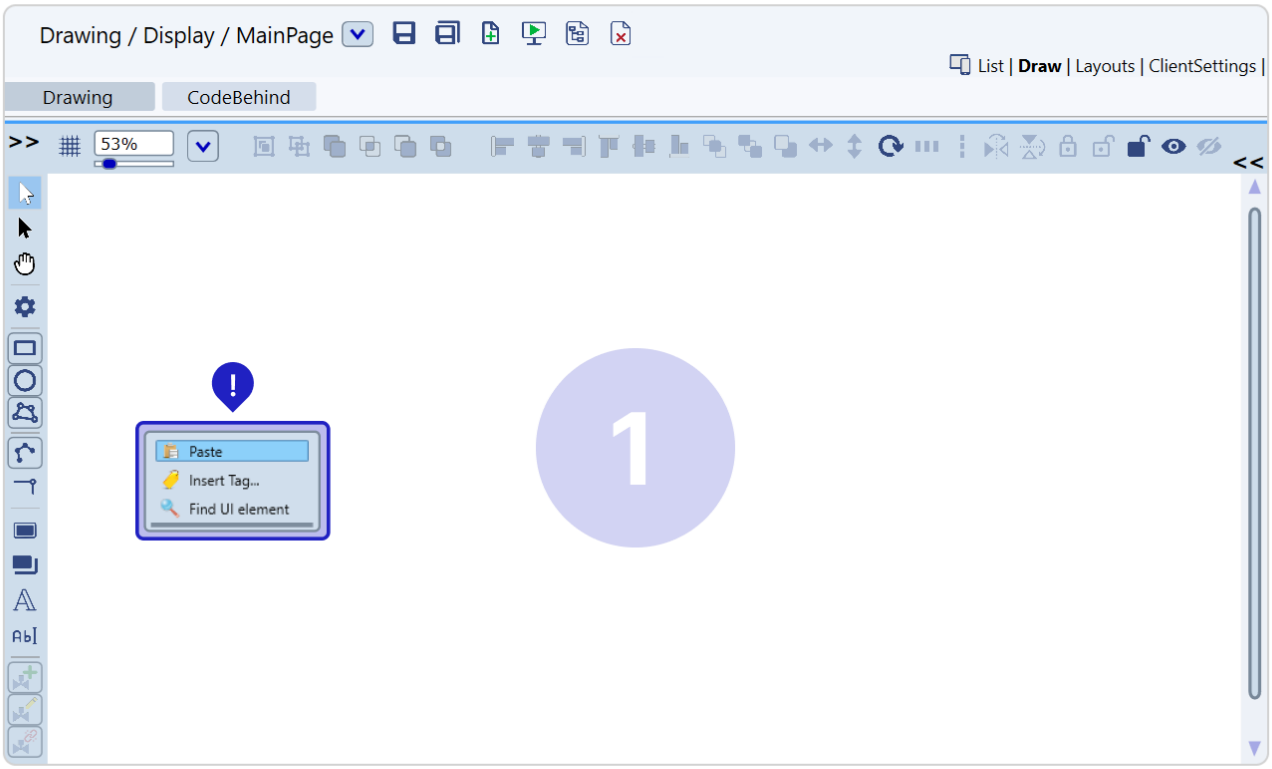
Now that our tags were created, we can start adding them to the display. Right-click the **Test** Tag, and select **Copy Selected Row** option.

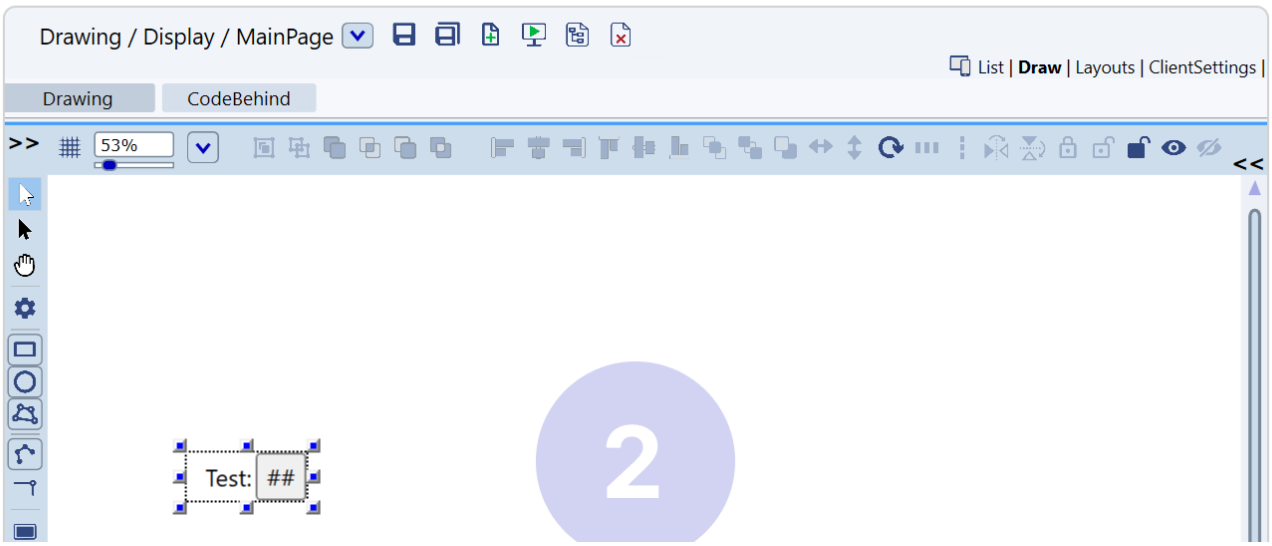


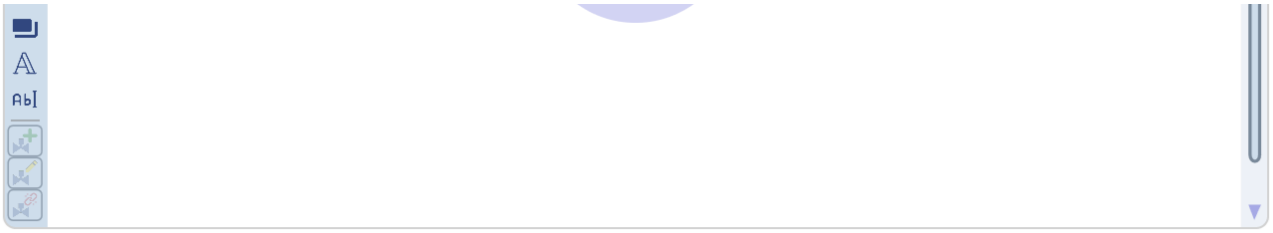
Paste Tags in the Draw area

Go to the **Draw** Environment, at the desired Page. Right-click the display and select **Paste**. You should see the Tag being inserted into the display, already formatted with a TextOutput, for label, and a TextBox for displaying value.

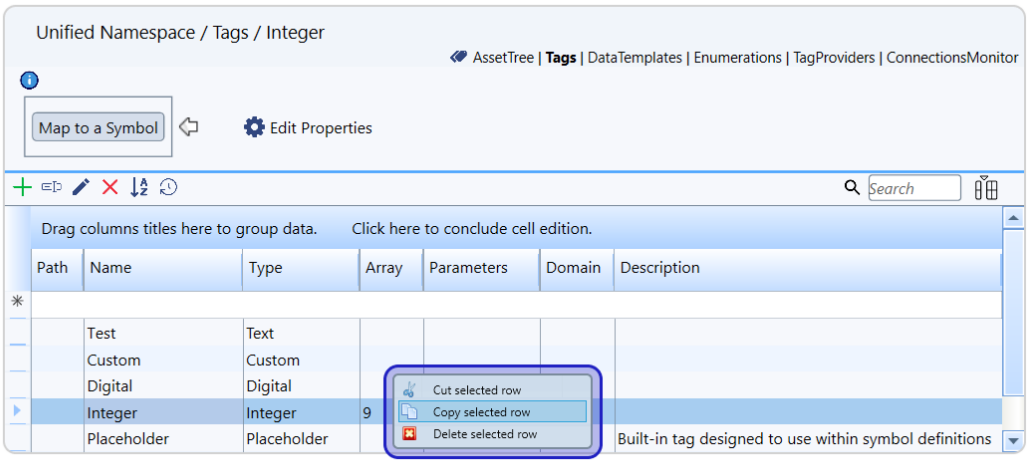

Unified Namespace / Tags / Test
AssetTree | Tags | DataTemplates | Enumerations | TagProviders | ConnectionsMonitor
Map to a Symbol
Edit Properties
Search
Drag columns titles here to group data.
Click here to conclude cell edition.
Path
Name
Type
Array
Parameters
Domain
Description
Test
Integer
Cut selected row
Copy selected row
Delete selected row

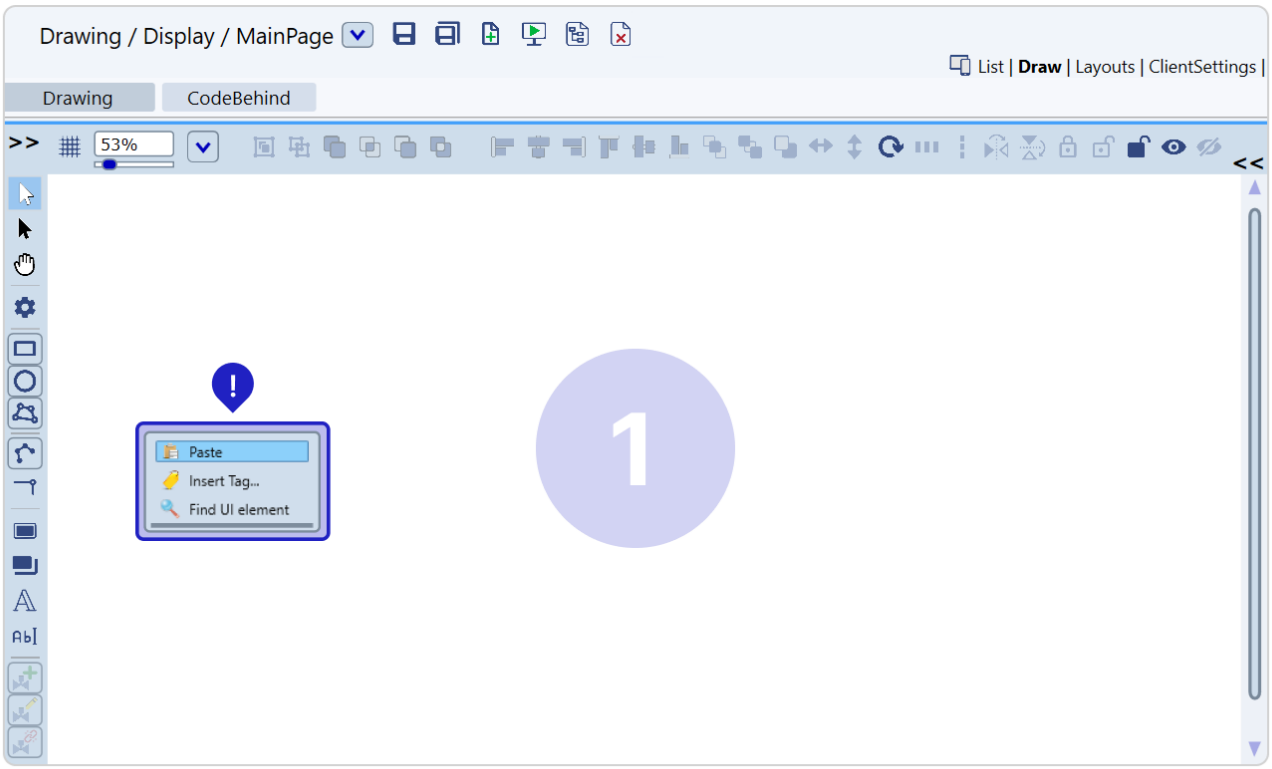

Drawing / Display / MainPage
List | Draw | Layouts | ClientSettings |
Drawing
CodeBehind
53%
Paste
Insert Tag...
Find UI element
1

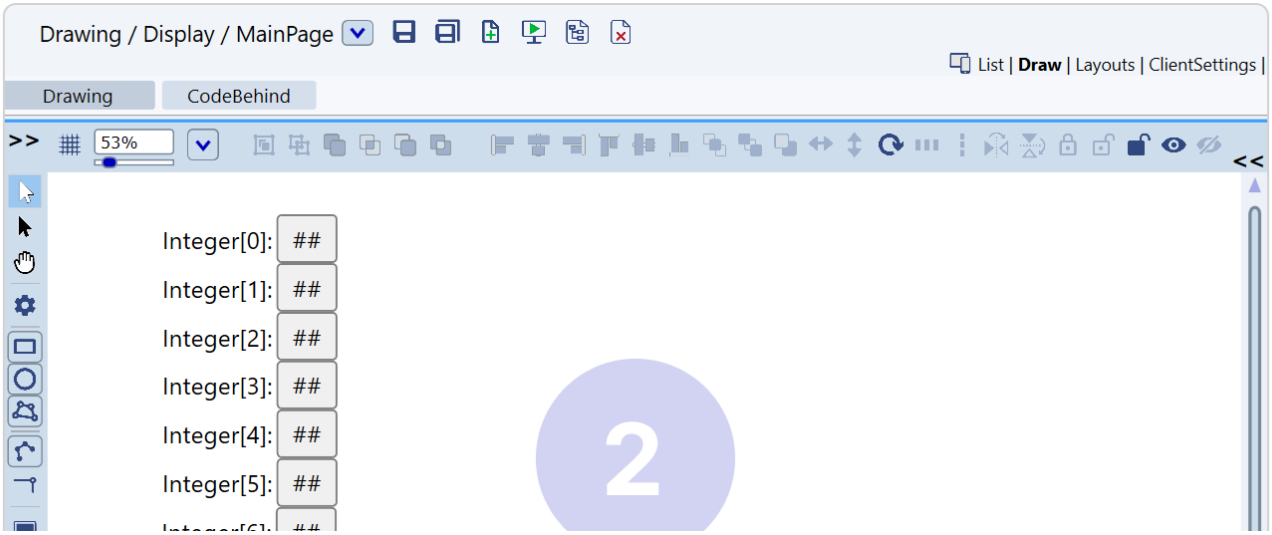

Drawing / Display / MainPage
List | Draw | Layouts | ClientSettings |
Drawing
CodeBehind
53%
Test: ##
2

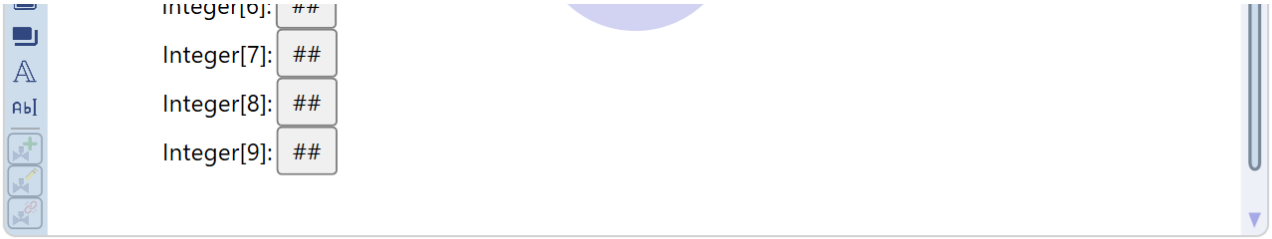


The same logic can be applied for array Tags. **Copy** the Tag row from **Unified Namespace Tags** related to Integer array. All elements should appear formatted when **Pasted** into a page.


Unified Namespace / Tags / Integer
AssetTree | Tags | DataTemplates | Enumerations | TagProviders | ConnectionsMonitor
Map to a Symbol
Edit Properties
Search
Drag columns titles here to group data.
Click here to conclude cell edition.
Path
Name
Type
Array
Parameters
Domain
Description
Test
Text
Custom
Custom
Digital
Digital
Integer
Integer
9
Placeholder
Placeholder
Cut selected row
Copy selected row
Delete selected row
Built-in tag designed to use within symbol definitions


Drawing / Display / MainPage
List | Draw | Layouts | ClientSettings |
Drawing
CodeBehind
53%
Paste
Insert Tag...
Find UI element
1


Drawing / Display / MainPage
List | Draw | Layouts | ClientSettings |
Drawing
CodeBehind
53%
Integer[0]: ##
Integer[1]: ##
Integer[2]: ##
Integer[3]: ##
Integer[4]: ##
Integer[5]: ##
2



Lastly, for a Custom Type Tag, **Copy** the Parent element from **Unified Namespace Tags** and **Paste** it into your page. The child elements will be automatically formatted at the Display.

Paste Tags with Symbols

When pasting Tags, instead of the TextIO property, it's possible to define a Symbol, that will be inserted in the displays, with the biding to the selected tag.

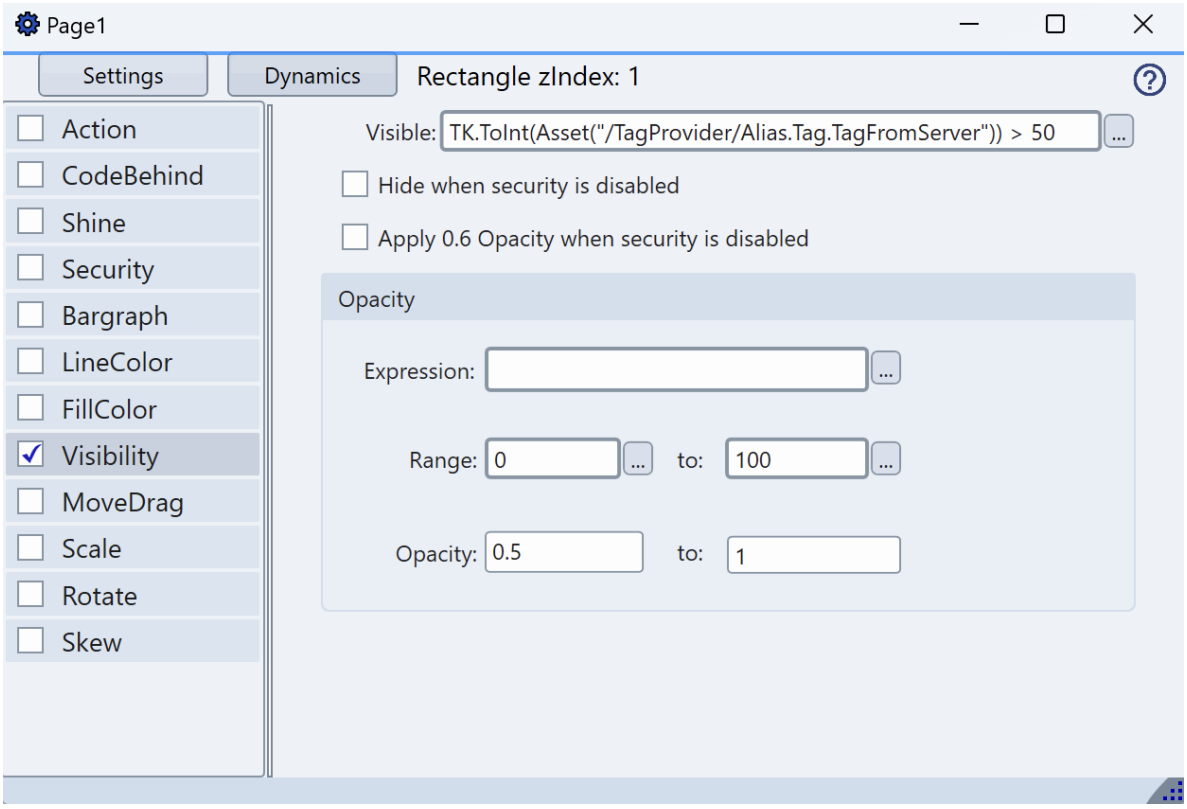
That configuration is described at the page Smart Symbols.

Using Assets in Displays

In Portable/HTML5 displays, Assets cannot be treated as dynamic objects due to limitations in NetStandard 2.0. To use them them in in Graphical Components, you must first convert them to the required data type (e.g., integer, string, digital, DataTable) using methods such as `TK.ToInt`, `TK.ToString`, etc.

In this example, we converted the Asset to an integer using TK.ToInT to determine if the object will be visible.

In WPF-only displays, which are based on .NET Framework 4.8, Asset calls return values that can be handled dynamically, without the need to explicitly define the data type.



In this section...
