

# Asset Modelling



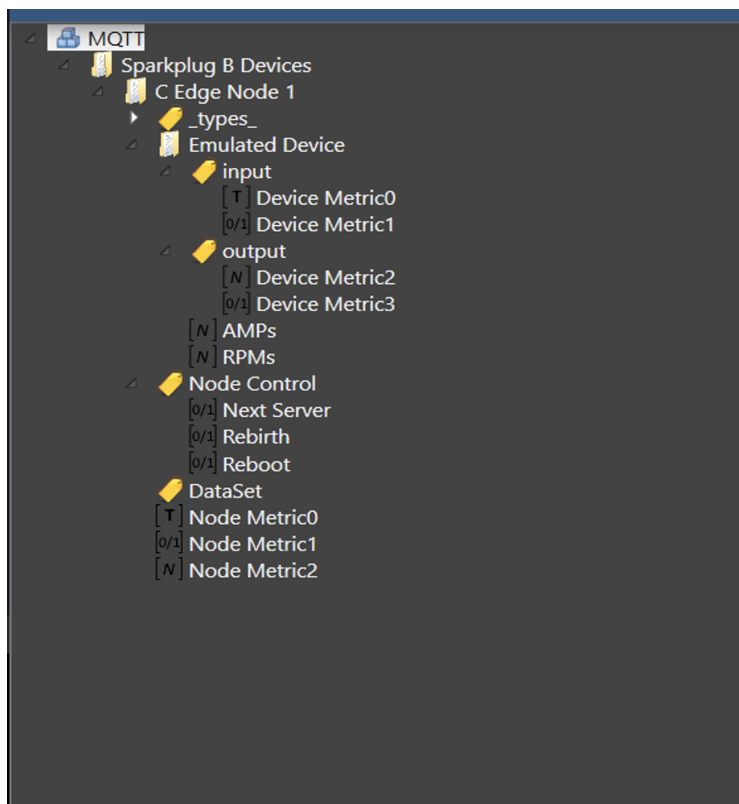
[Watch the video tutorial here](#) (no audio)

## Overview

Asset Modeling exists in both the Engineering Environment and Project Runtime. Assets allow you to configure additional metadata in your project. For example, you can organize the objects in your project into a hierarchy.

This enables you to group tags that are related to each other. The hierarchy may reflect such things as the area of your manufacturing floor or the location of your machinery.

The image below describes an example of a project's hierarchy.



## System Requirements

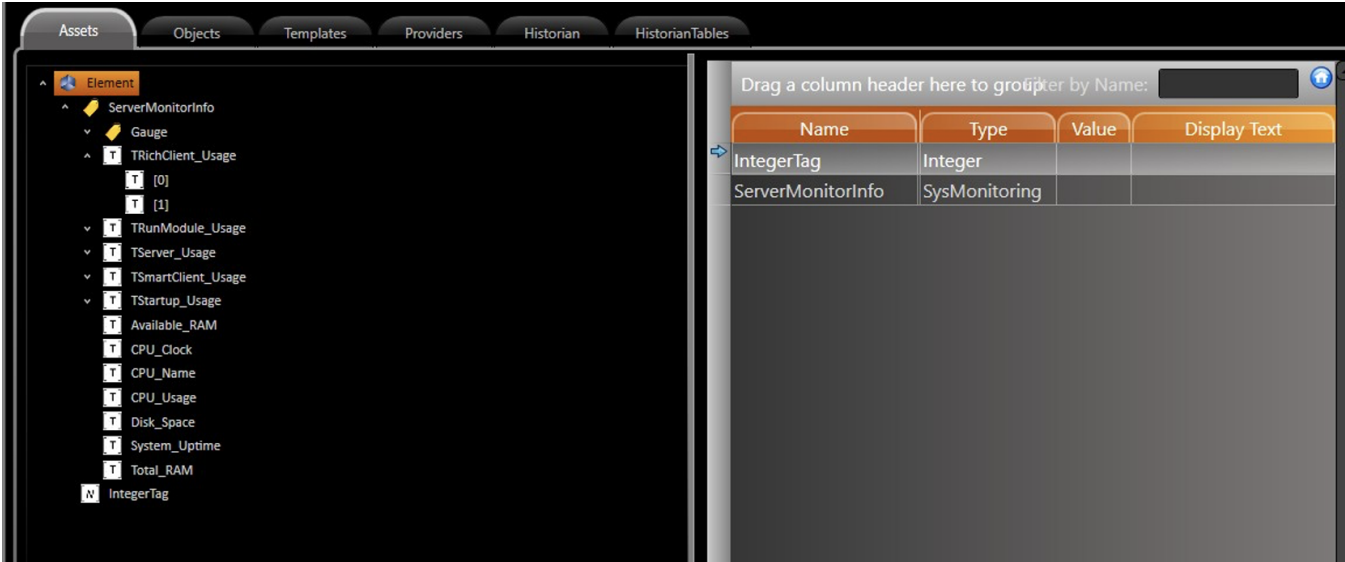
In order to use this feature, you need to make sure your system matches these requirements.

- Product Major Version 9.2
- Enterprise Project family

## Asset Tab - Engineering Configuration

In the Engineering Environment, you should find the **Assets** tab under **Edit > Tags**. If this tab is not available in your project, please check to see if you have the Enterprise project family. (The project will need to be reopened for changes to be applied.)

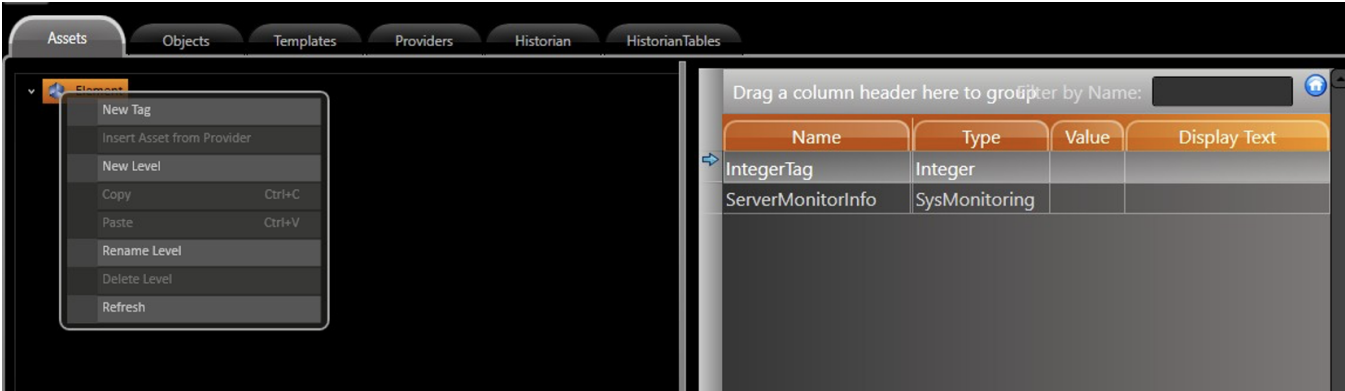
This tab contains a treeview structure that, by default, automatically displays all your project's Tags. They are added to the Root Element called **Element**. This allows you to navigate to the child elements and drill to Arrays or Templates.



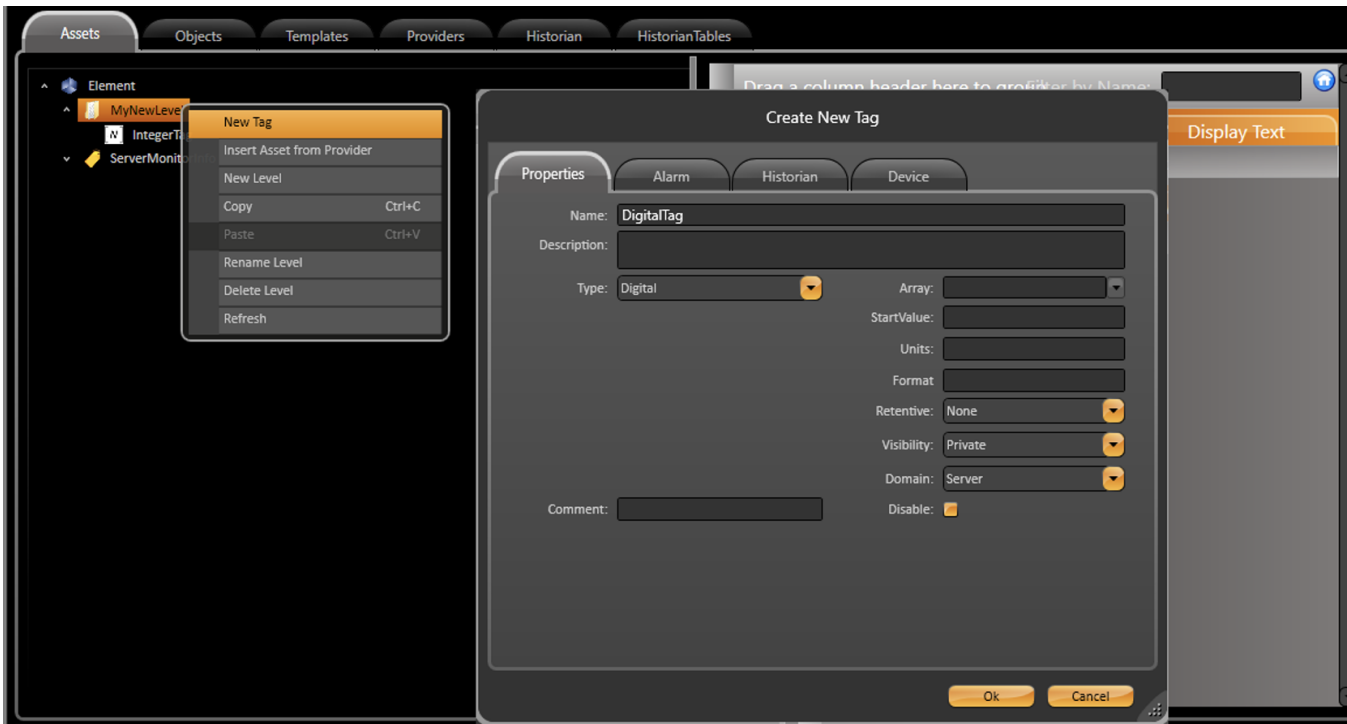
Once you have the project running, you will be able to monitor the Tag values directly from the Engineering Environment through the **Value** column.

The **Display Text** column is also used in Runtime to give an alternative/friendly name to your element in the AssetControl component.

This AssetTree is not limited to the defined structure of the Tags . You can create your own architecture. To do so, right-click on a Level. In this example, it is the Root Element, and select the New Level option.



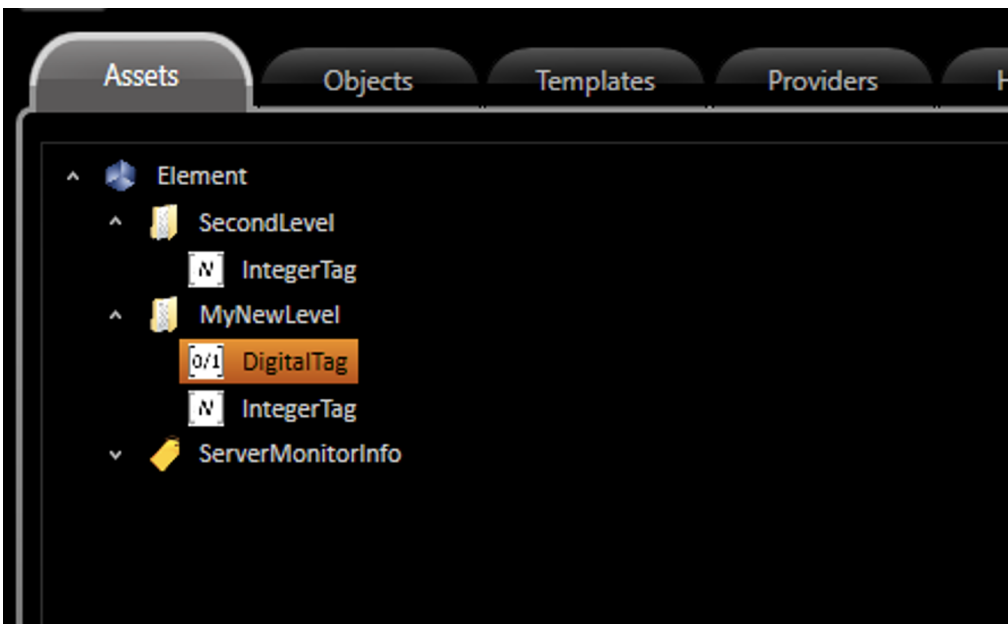
After you give the level an appropriate name, you can start adding Tags to it. You can move (click and drag) an existing Tag from the Root or create a new one directly from the New Tag button.



You can create new Tags from the Assets Tab. You can also add Alarms as well as add it to the Historian and to communication Points directly from this dialog.

You can add Dynamic Data from the TagProvider inside your treeview structure. For more information, please refer to the [TagProvider](#) documentation.

We improved this component in this version of FactoryStudio by adding the ability to have the same Tag in more than one level.



In this configuration, the Tags are not exactly the same. Instead, they are objects from the same Tag Instance. This can be seen in **Edit > Tags > Objects** where two instances of the IntegerTag are on a table but are mapped to their own Level.

Drag a column header here to group		
Name	Type	Level
IntegerTag	Integer	SecondLevel
DigitalTag	Digital	MyNewLevel
IntegerTag	Integer	MyNewLevel
ServerMonitorInfo	SysMonitoring	

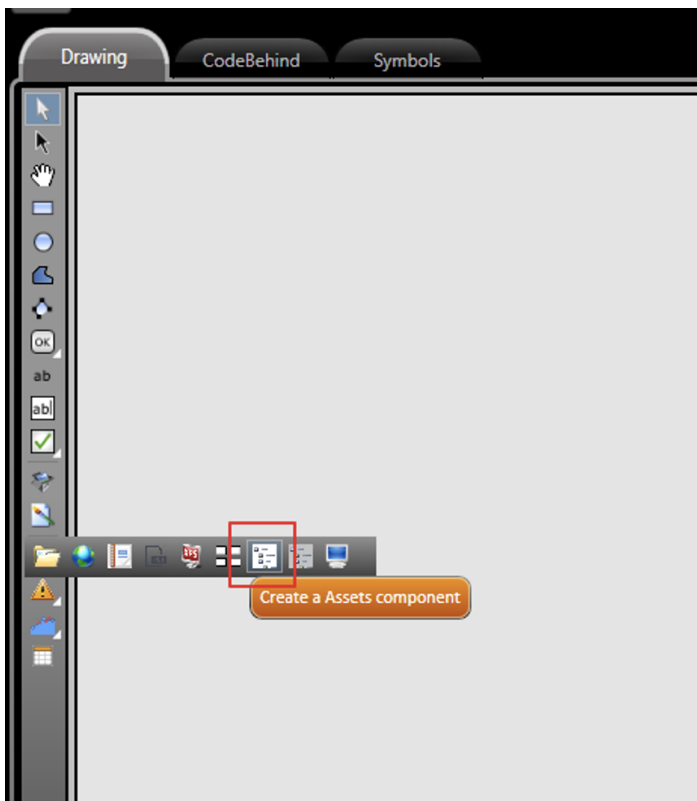
You can copy/paste an entire Level, or just one Asset, from a source Level to a destination one. By doing so, all Tags (and their Alarms, Historian, and Device configurations) will be copied as well.

With this information, your system model can be built in an Asset format, which can be displayed in Runtime with the Assets Control.

## Draw Environment

### Asset Control

This component allows you to view the same structure in Runtime that is defined in the Engineering Environment. To do so, go to your Draw Environment, and select the [AssetControl](#) component from the toolbar.



Add the component to your display, and customize it according to the settings described below.

### AssetsControl Configuration

Control name:

Binding: TwoWay

SelectedAsset:

SelectedLevel:

☐ Show only tags  
☒ Show tags on root  
☒ Expand tags  
☐ Show client tags

☐ Prefer display text  
☐ Localizable

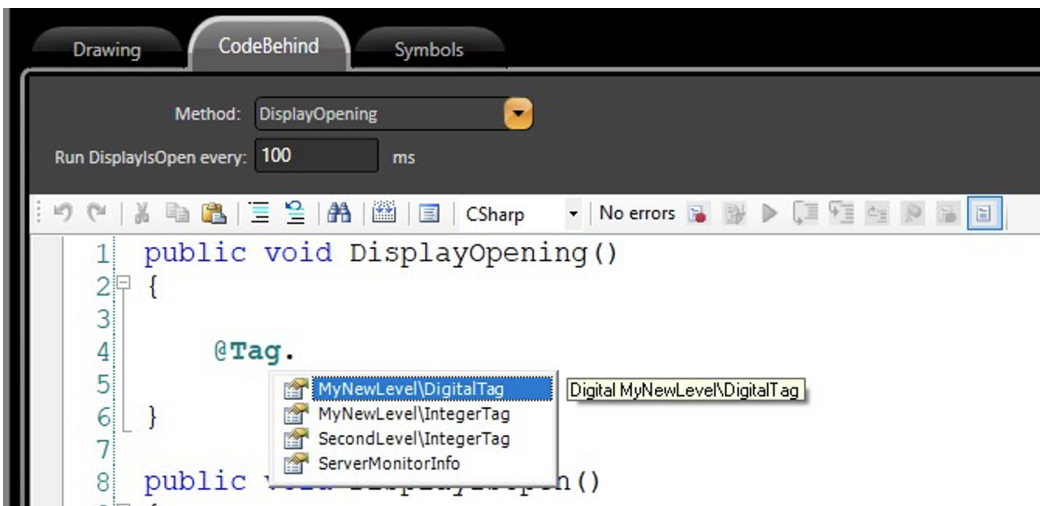
Ok
Cancel

The parameters are described as follows:

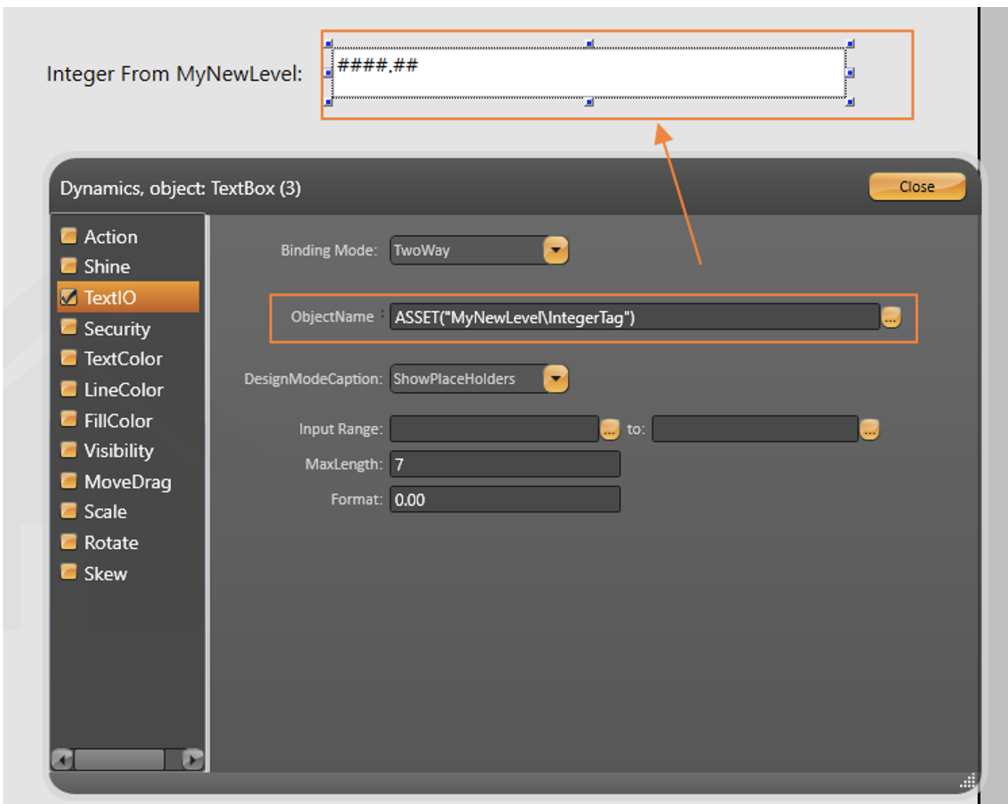
- **ControlName:** Element's reference name so its properties can be accessed and manipulated in CodeBehind.
- **SelectedAsset:** Tag that receives the selected Asset in Runtime.
- **SelectedLevel:** Tag that receives the selected Level in Runtime.
- **Show Only Tags:** Flag that indicates if the Asset Tree will be composed of only Tags (if true) or other project objects (such as Displays).
- **Show Tags On Root:** By default, all common type Tags (Integer, Digital, Text, etc) are added to the Root. By unchecking this option you can filter these elements out.
- **Expand Tags:** Flag that indicates if multi-level Tags can be expanded (Templates, Arrays). If unchecked, only the main Tag will be in the Asset Tree.
- **Show Client Tags:** Flag to define if client domain Tags will be displayed in Runtime. If unchecked, only server domain Tags will show up.
- **Prefer Display Text:** Flag to indicate if the text that will show up in the Tree is the TagName or its Alias (defined in **Edit > Assets > DisplayText**). If checked and the column is empty, the TagName will be used.
- **Localizable:** If checked, Tag Names (or aliases) can be translated to different languages according to your project dictionary's settings.

### CodeBehind and IO Components

You can also access the Asset elements from the CodeBehind by navigating through the Tag Namespace. IntelliSense will display the Tags and their respective Levels.

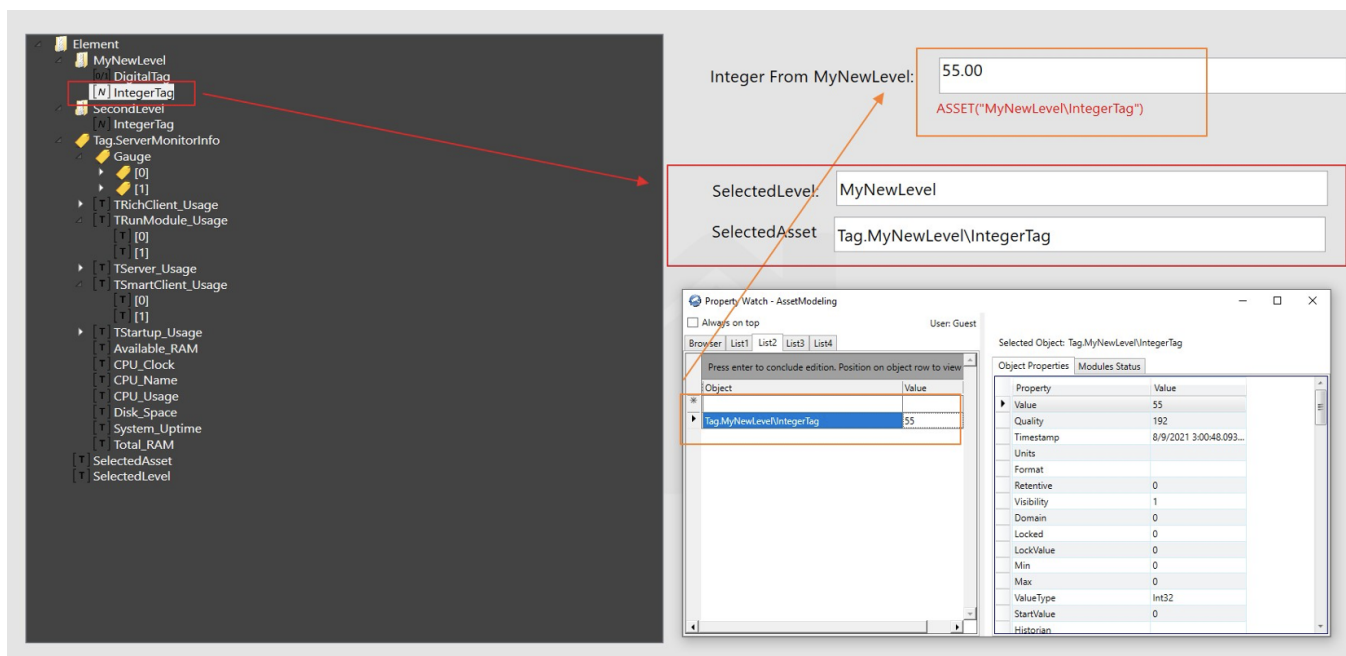


Similar to the TagProvider approach, the elements from the Asset View can be accessed with the syntax: **ASSET(" ? CompleteAssetName ?")** either in a TextBox element or the CodeBehind.



## Runtime Behavior

In runtime, your Asset Tree and its components will look like the image below.



The IO element mapped with the **ASSET(" CompleteAssetName ")** syntax will be resolved similar to the TK.GetObjectValue and will display the current value of the specific Tag it was mapped to.

The SelectedAsset and SelectedLevel Tags will also change according to the element you click on.