

Responsive Dashboards

Overview

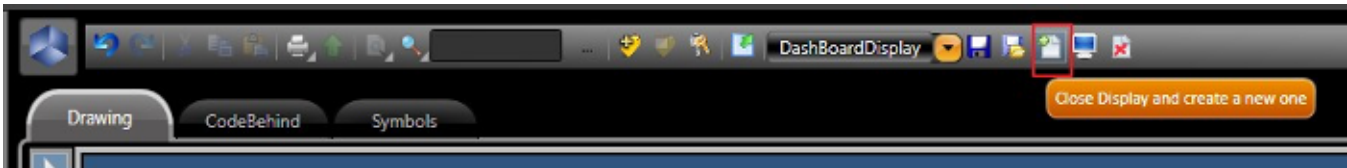
A dashboard is a type of graphical user interface which provides an at-a-glance view of key performance indicators (KPIs) relevant to a particular objective.

A data dashboard is the most efficient way to track multiple data sources because it provides a central location for businesses to monitor and analyze performance. The data is displayed in the form of tables, line charts, bar charts, and gauges.

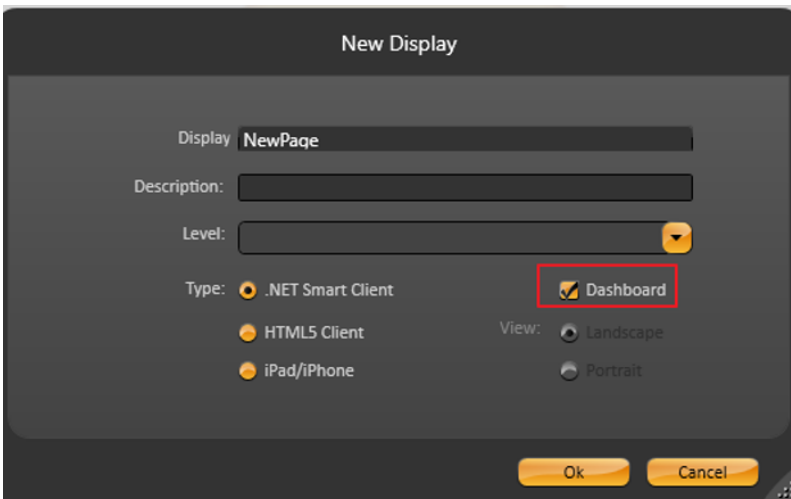
Responsive dashboard displays have the ability to respond dynamically to any display size, which means the display will rearrange to format content based on the size and shape of the user's display.


Creating a Display

To create a dashboard display, you need to navigate to the *Draw Editor*, and click on the Create Display button.



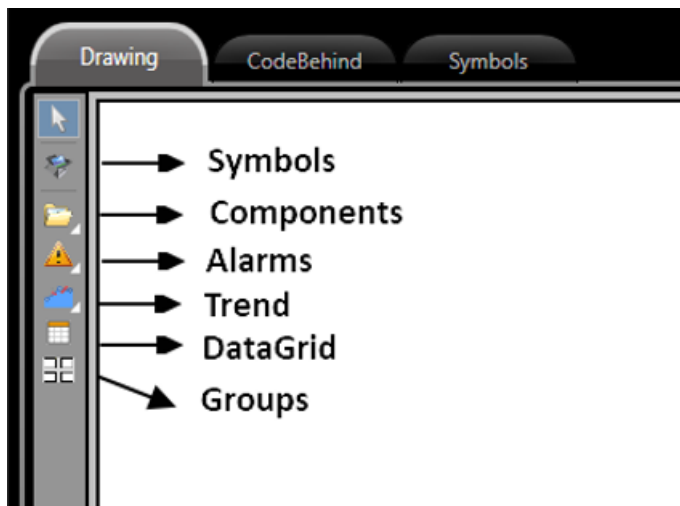
In the popup window, enable the **Dashboard** checkbox, fill in the parameters as desired, and click on **OK**.



 Dashboards are available for both .Net and HTML5 display types.

Available Elements

The following elements are available in *Dashboard* displays:



- **Symbols:** User made symbols and built-in symbols, located in the Symbol library, can be used in dashboard displays.

Graphic elements such as buttons, textboxes, and checkboxes are not available by default in dashboard displays. To use these graphic elements in your dashboard display, you need to create a symbol with these components and use the symbol in your dashboard display.

- **Components:** The external components are listed below.

Bar Chart	Calculator	Web Browser
Pie Chart	Circular Panel	Assets/RemoteAssets Configuration
Advanced Pie Chart	File Explorer	Child Display
3D Pipe Viewer	External DLLs	Page Selector
Report Preview	XPS/PDF Viewer	
Map/GMap Control	Report Viewer	

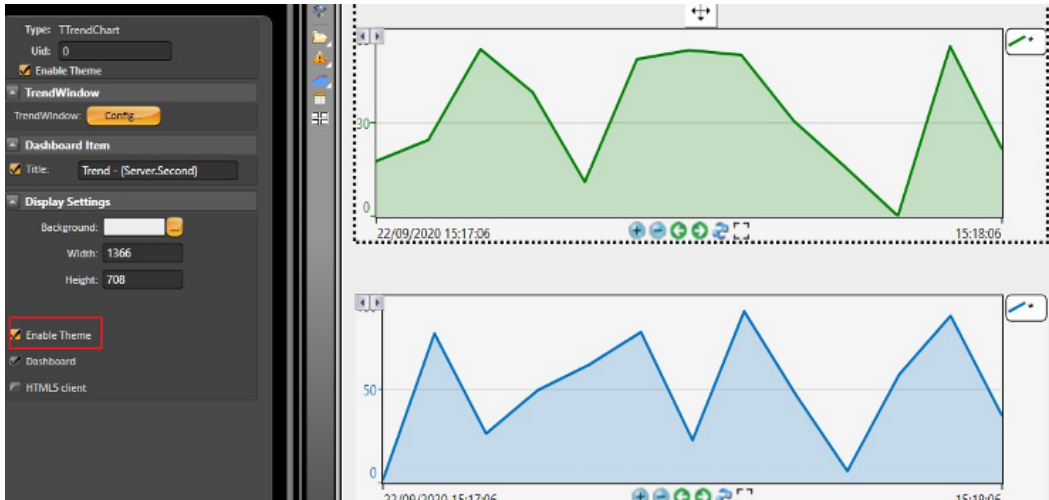
- **Alarms:** Alarm Window and Alarm Area
- **Trend:** Trend Window and Drilling
- **DataGrid:** DataGrid
- **Groups:** You can combine different elements into a single Group.

Display Editor

When the dashboard option is enabled, the Draw Editor's layout is configured in blocks.

To insert an element, select it and drag it onto the page. You should see a highlighted rectangle in the location where it will be placed.

To enable this feature, select the *Enable Theme* checkbox in the left panel.



CodeBehind Callback (Only for Advanced Customization)

The properties of dashboard items can be modified through the callback added to the CodeBehind using the syntax below:

```
* .NET:
public void OnDashboardCustomItem(TDashboardBorder item)
{
    item.TitleElement.TextAlignment = TextAlignment.Center; item.TitleElement.FontStyle = FontStyles.Italic; item.
    TitleElement.FontSize = 20; item.TitleElement.Foreground = Brushes.Black;
}

* HTML5:
this.OnDashboardCustomItem = function(item)
{
    //item.TitleElement.FontStyle = 1;
    //item.TitleElement.HorizontalAlignment = 1;
    //item.TitleElement.FontSize = 18;
};
```

Using the .Net callback described above, our titles look something like this picture in runtime.



The Title also supports Tags and Project Objects by placing them between curly brackets. Curly brackets are used so runtime can handle the element as a project variable, instead of as text. See some examples below.

```
Title: {Client.DateTime}
Title: {Tag.MyTextTag}
Title: {Server.ComputerName}
```



In runtime, the elements can be maximized by clicking on the button indicated by the (1) below or resized by dragging the button indicated by the (2) below.

