



How to hide and show a filter panel with a custom control and JavaScript

Filters (prompts) tend to take up a lot of valuable space on a report. It is often useful to provide users with the ability to hide and show these filters in a panel. In the sample report titled “Global sales”, we used a Block, a Custom Control, and JavaScript to hide and show a panel of filters. Here’s how...

1. When we created the *Global sales* sample, we began by laying out the general structure of the report with a Table. We created a new report and dragged in a Table object from the Toolbox




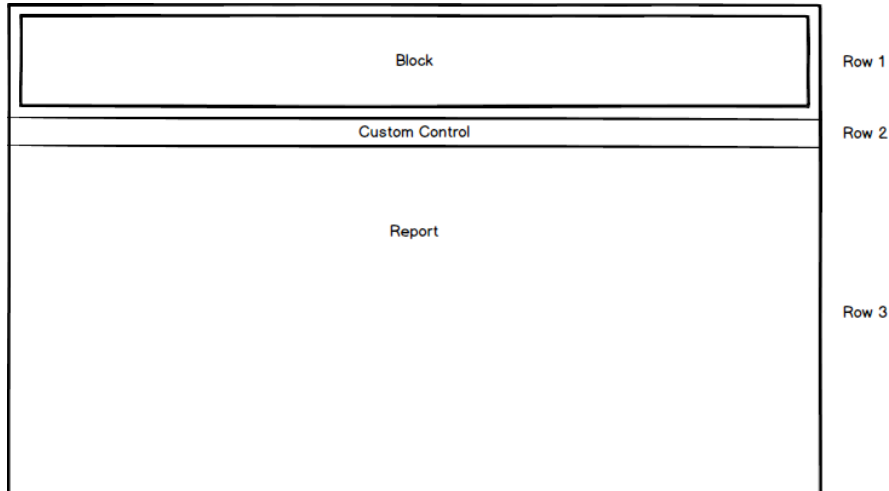
. The Table was configured to have 1 column, 3 rows, and a width of 100%:

Row 1
Row 2
Row 3

2. The filter prompts in the sample report for Year and Product Line are contained within a Block. In fact, it is this Block that we are hiding and showing with JavaScript. We opened the Toolbox and dragged a Block from the *Pinned* section into row 1 of the report:

Block
Row 2
Row 3

3. With the Block selected, we opened the Properties  and set the following:
 - a. Padding: 20px 20px 20px 20px
 - b. Size & overflow: Width: 100%
 - c. Name: FilterPanel
4. From the Toolbox, we opened the *Advanced* section and dragged a Custom Control into row 2 of the report:



5. With the Custom Control selected, we opened the Properties and set the Module path to `/bi/samples/js/HideShowFilterPanel.js`.
 - a. **Note 1:** JavaScript files can be stored anywhere a browser has access to. For example, you can host JavaScript files on your webserver, or you can include JavaScript files under the Cognos install (e.g. `C:\install_location\webcontent\bi\samples\js`)
 - b. **Note 2:** If your installation uses a **gateway** instead of a dispatcher (Easy Install uses a dispatcher by default) then modifications must be made to the location of the Module Path property of the Custom Control:
 - i. Method 1: If your web alias is *ibmcognos*, you will need to set the correct module path to include the web alias, like this:
`/ibmcognos/bi/samples/js/HideShowFilterPanel.js`
 - ii. Method 2: You can also use the full URL with the server name, like this:
`http://[webserver]/[web alias]/bi/samples/js/HideShowFilterPanel.js`
6. Again, in the Properties of the Custom Control, we set the Configuration to:

```
{
  "Control name": "FilterPanel",
  "Orientation": "top",
  "Padding": "0px 30px 0px 30px",
  "Foreground color": "#003868",
  "Hover foreground color": "#EAEAEA",
  "Background color": "#EAEAEA",
  "Hover background color": "#38A8C8",
```

```
    "Border color": "#B1B6BA",  
    "Hover border color": "#EAEAEA",  
    "Font size": "10px"  
  }  
}
```

- **Note:** The sample JavaScript file has been configured to accept several variable values via the Configuration property above. In this way, the report author can make changes to the look and feel of the filter panel without having to modify the source code directly in the JavaScript file.

7. We selected the Table Cell that contained the Custom Control and set the following Properties:
 - Padding: Top 0px Bottom 1px
 - Background color: #003868
 - Horizontal alignment: Center
8. Now the basic structure for hiding and showing a panel of filters is complete! All that is left is to add content to the filter panel and the report. We also added a page header for the report's title.

Tip

In this example, the panel of filters is open (visible) by default when you run the report. You may want it to be closed by default. To do that:

- Select the Block from Step 2 above named FilterPanel and open the Properties
- Set the *Box type* to *None-Block*

Warning: Once you do this, as an author, you will not be able to see or modify the Block in *Page design* or *Page preview* mode. You will need to switch to *Page structure* mode to find the Block and switch the *Box type* property back to Default:

