Designing and Implementing Custom Visuals for Power BI

Setup Time: 60 minutes

Lab Folder: C:\Student\Modules\06_CustomVisuals

Overview: In this lab, you will continue to develop custom visuals using the Power BI Custom Visual CLI Tool (PBIVIZ) that you began to work with in the previous lab. Now that you know how to create new projects using the PBIVIZ utility and to integrate the D3 library, this lab will focus on moving ahead and learning the fundamentals of designing and implementing a useful custom visual for Power BI.

Unlike the previous lab where you created three different custom visual projects, this lab focuses on a single custom visual project named **barchart** that you will work on throughout all the exercise of this lab. You will learn how to define visual capabilities and how to consume data from inside a Power BI dataset. You will also learn how to add a custom property

In the final exercise, you will work through the steps to package the **barchart** custom visual as a PBIVIZ package file which can deployed directly into the scope of a Power BI workspace or within the scope of a PBIX project file.

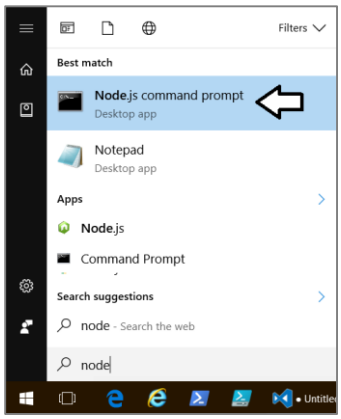
Prerequisites: This lab assumes you've already installed Node.JS and Visual Studio Code as described in setup.docx.

Exercise 1: Create a New Project for the Barchart Custom Visual

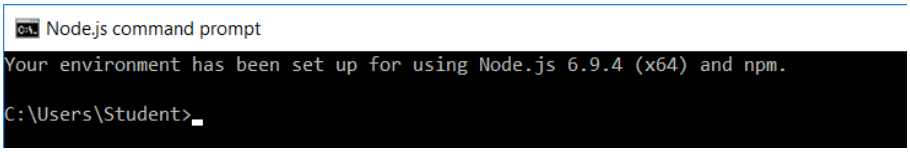
In this exercise, you will begin the process of developing a new custom visual by creating a new project named **barchart** with the PBIVIZ utility and integrating the D3 library.

In this exercise, you will create a new visual named **viz03** which uses D3 to implement a simple Power BI visual.

1. Create a new visual project named **barchart**.
   a) Using the Windows Start menu, launch the **Node.js command prompt**.



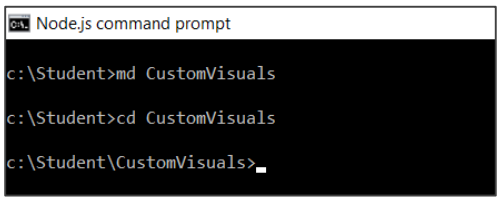
   b) You should now have an open Node.js command prompt.



   c) Type and execute the following command to make the current directory back to **C:\Student\CustomVisuals**

```
cd C:\Student\CustomVisuals
```

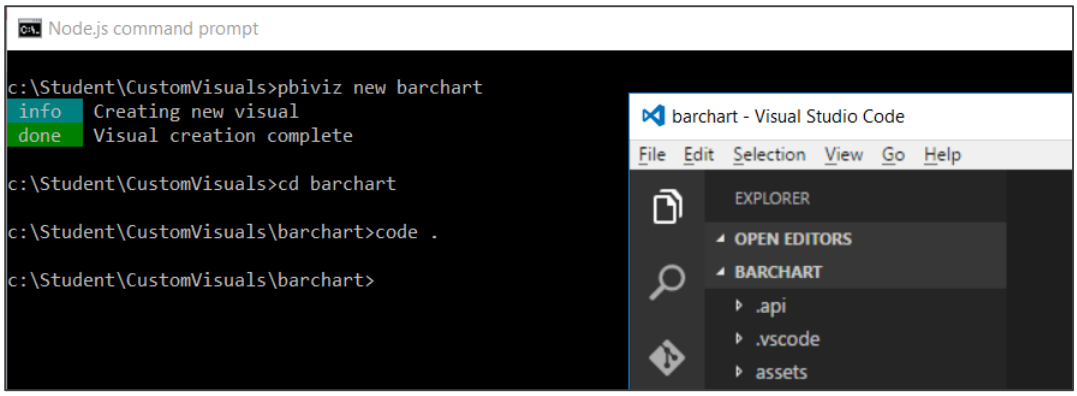
If a directory with the path of **C:\Student\CustomVisuals** does not already exist, you should create it.



d) Type and execute the following three commands to create a new project named **barchart** and open it in Visual Studio Code.

```
pbiviz new barchart
cd barchart
code .
```

e) You should now have a new project named **barchart** that is open in Visual Studio Code.



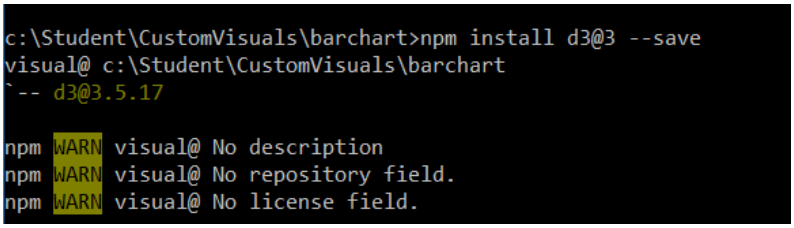
2. Add the NPM package for the D3 library and the D3 typed definition files.

a) Return to the Node.js command prompt.

b) Type and execute the following command to install the D3 JavaScript library.

```
npm install d3@3 --save
```

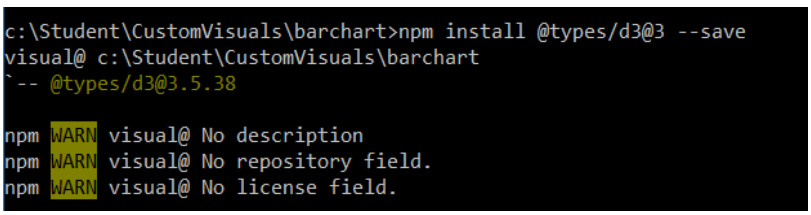
c) Wait for the **npm install** command to complete.



d) Type and execute the following command to install the typed definition files for the D3 library.

```
npm install @types/d3@3 --save
```

e) Wait for the **npm install** command to complete.



3. Modify the **externalJS** setting in the project's **pbiviz.json** file to reference the JavaScript file for the D3 library.
   a) Return to Visual Studio Code.
   b) Open the **pbiviz.json** file and locate the **externalJS** setting.
   c) Update the array for the **externalJS** setting by adding the path to **d3.js**.

```
"externalJS": [
  "node_modules/d3/d3.js"
],
```

   d) Save and close **pbiviz.json**.
4. Modify the **files** setting in the project's **tsonfig.json** file to include a reference to the D3 typed definition file for named **index.d.ts**.
   a) Open the **tsconfig.json** file.
   b) Move to the bottom of **tsconfig.json** and locate the files property.

```
  },
  "files": [
    ".api/v1.5.0/PowerBI-visuals.d.ts",
    "src/visual.ts"
  ]
}
```

   c) Add a new entry into the array for the **files** property using the following project-relative path to the D3 typed definition file.

```
node_modules/@types/d3/index.d.ts
```

   d) Your update to **tsconfig.json** should match the following screenshot.

   e) Save and close **tsconfig.json**.

Now you are now ready to begin programing your TypeScript code using the D3 library.

5. Modify the contents of **capabilities.json**.
   a) Open and copy the contents of the following file.

```
C:\Student\Modules\06_CustomVisuals\Lab\StarterFiles\capabilities.json.txt
```

   b) Open the **capabilities.json** file in your project and replace the contents with what is in the Windows clipboard.
6. Modify the **visual.ts** file.
   a) Open and copy the contents of the following file.

```
C:\Student\Modules\06_CustomVisuals\Lab\StarterFiles\visual.ts.txt
```

   b) Open the **visual.ts** file in your project and replace the contents with what is in the Windows clipboard.
7. Test out your new visual on a Power BI report.
   a) Return to the Node.js command prompt and run the following command to start a new debugging session.

```
pbiviz start
```

   b) Test out the visual from within the Power BI service.