



PowerApps Component Framework Custom Control Development

Aric Levin

Lead Software Architect

Technical Project Manager

Microsoft Power Platform

The low-code platform that spans Office 365, Azure, Dynamics 365, and standalone applications



Power BI

Business analytics



Power Apps

Application development



Power Automate

Process automation

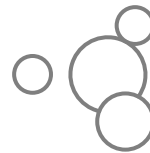


Power Virtual Agents

Intelligent virtual agents



**Data
connectors**



AI Builder

**Common
Data Service**

What is PCF?

- ▶ Framework that allows us to build custom components to be used inside of Model-Driven apps
- ▶ Currently two types of components available:
 - ▶ field
 - ▶ dataset
- ▶ PCF for Model Driven Apps became General Available (GA) to the public on October 1 2019
- ▶ Components can only be used in Unified Interface
- ▶ PCF controls can also be used in Canvas Apps (current in experimental mode, not preview/public)

Prerequisites

- ▶ Node.js
- ▶ PowerApps Command Line Interface
- ▶ Visual Studio
 - ▶ Developer Command Prompt for VS 2017/2019
 - ▶ Developer PowerShell for VS 2019
- ▶ Optional: Visual Studio Code
- ▶ Optional: PCF Custom Control Builder plugin from XRM Toolbox

Node.js

Required to develop custom controls for Model-driven apps

► <http://nodejs.org/en>

Download for Windows (x64)

10.15.3 LTS

Recommended For Most Users

12.2.0 Current

Latest Features

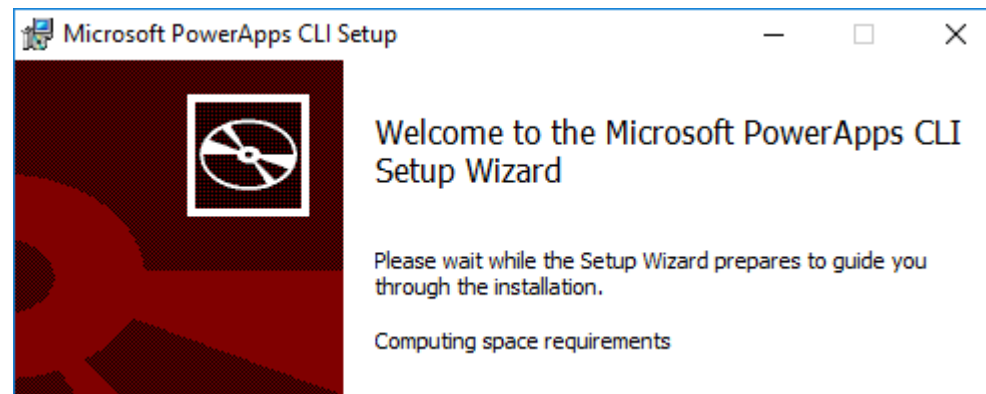
[Other Downloads](#) | [Changelog](#) | [API Docs](#)

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

Power Apps Command Line Interface

Required to create PCF Control solutions and deploy to CDS

► <https://aka.ms/PowerAppsCLI>



Visual Studio Code

You can use Visual Studio code to create your Power Apps Component Framework controls and simplify development efforts

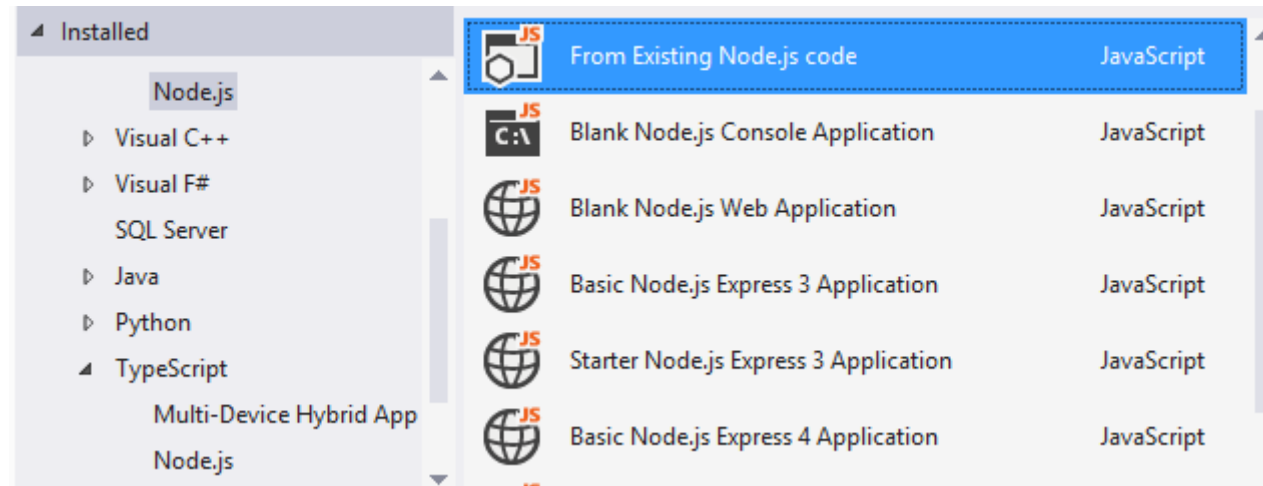
► <https://code.visualstudio.com/>

```
// Does credit card have a value
if (creditCardNumber != null && creditCardNumber.length > 0)
{
    // Is this a Visa card?
    _regEx = new RegExp('^4[0-9]{12}(?:[0-9]{3})?$');
    if (_regEx.test(creditCardNumber))
    {
        isValid = true;
        imageUrl = "https://xyz.blob.core.windows.net/shared/imgs/ico/visa24.png";
    }
}
```


Visual Studio Developer Tools

The Developer Command Prompt or Developer PowerShell for Visual Studio 2017/2019 is required for deployment of PCF custom controls

► <https://visualstudio.microsoft.com/vs/>



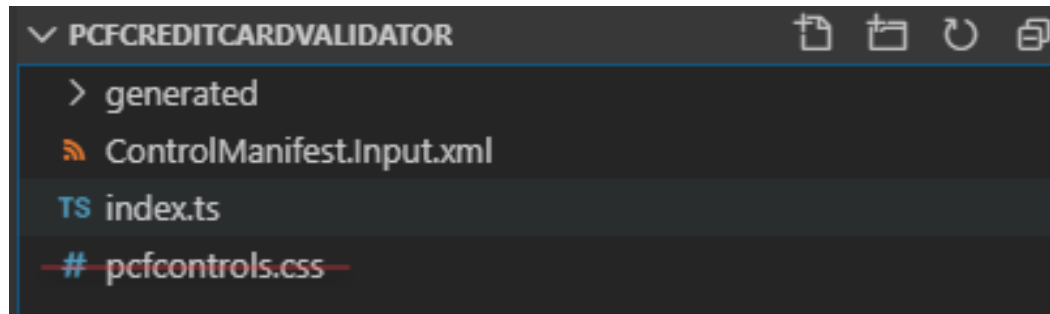
Development Lifecycle

- ▶ Create Required Project files
- ▶ Install Required components
- ▶ Development and Testing
 - ▶ Isolated Testing
 - ▶ Continuous Testing (watch)
- ▶ Solution Creation
 - ▶ Solution Init
 - ▶ Add Reference
 - ▶ MS Build
- ▶ Deploy Solution

Project Creation

Use the `pac pcf init` command to create your new project files

- ▶ `pac pcf init --namespace PCFControls --name CreditCardValidator --template field`
 - ▶ The values for template will be either field or dataset



Install Dependencies

Use the npm install command to add ALL dependencies to your project

- ▶ npm install
- ▶ npm install jquery
- ▶ npm install @types/jquery
 - Ability to install all project dependencies or only required project dependencies
 - Will not add additional code to output, only disk space during development

Update Manifest File

Set the properties and resources that make up your project within your manifest file

```
<control namespace="PCFControls" constructor="PCFCreditCardValidator" version="0.0.1" display-name="PCF Credit Card Validator">
  <property name="CreditCardNumber" display-name-key="PCFCreditCardValidator_CreditCardNumber">
    <resources>
      <code path="index.ts" order="1"/>
      <css path="pcfcontrols.css" order="1" />
    </resources>
    <feature-usage>
      <uses-feature name="Utility" required="true" />
      <uses-feature name="WebAPI" required="true" />
    </feature-usage>
  </property>
</control>
```

Development

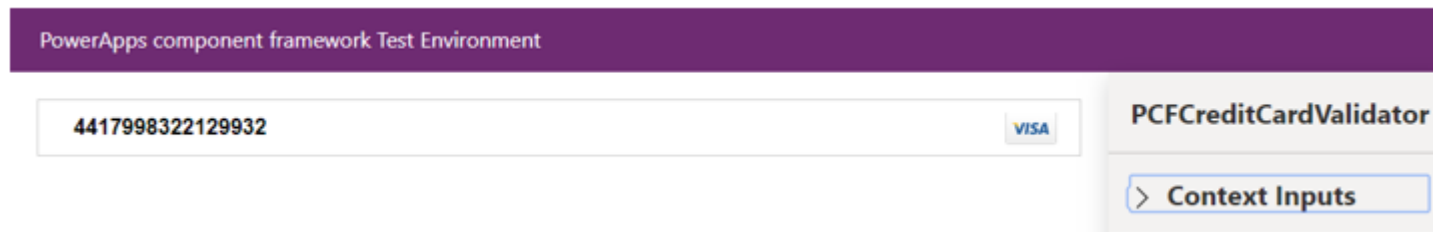
Add required code to the Typescript file

- ▶ variable declaration
 - ▶ Content, notifyOutputChanged, container, HTML elements, event listener, other public variables
- ▶ init method
 - ▶ Initialization code for typescript variables, binding events and skinning the control
- ▶ updateView method
 - ▶ Validation and displaying of data back to the user interface
- ▶ getOutput method
 - ▶ Called prior to control receiving new data
- ▶ Event handler method
 - ▶ Custom validation code when the event happens

Build and Testing

Compile and test the control using the Test Harness

- ▶ npm run build
 - ▶ Compile the control files to be tested
- ▶ npm start
 - ▶ npm start watch for continuous debugging/recompilation/testing



Deploy

Build the solution files and deploy to your CDS environment

- ▶ Create the solution files and add the required references

- ▶ `pac solution init --publisher-name [NAME] --publisher-prefix [PREFIX]`
- ▶ `pac solution add-reference --path [PATH TO PROJECT FILE]`

- ▶ Build the solution files

- ▶ `msbuild /t:restore`
- ▶ `Msbuild`

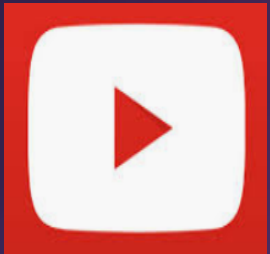
- ▶ Import the packages solution

- ▶ Configure the solution properties

Demonstration



www.ariclevin.com



www.youtube.com/channel/UCzdg6V5kkjiBufYnvJT4XWQ



github.com/ariclevin



www.linkedin.com/in/ariclevin

Business Applications Communities

Connect · Learn · Share · Inspire

Join the Microsoft Business Applications Communities where you can connect with peers and experts and earn recognition for your contribution. Get answers to complex questions, learn from engaging discussions, read informative blogs, view webinars, and find product use examples in galleries.



<https://community.dynamics.com>



<https://community.powerbi.com>



<https://community.powerapps.com>



<https://community.flow.microsoft.com>