

Power Apps Component Framework Overview

Cathal Noonan - 25th Nov 2021

Senior Technical Consultant at Codec Ireland

Agenda

- What is PCF
- Brief comparison with HTML Web Resources
- Where we can use PCF
- PCF development

What is PCF?

- Power Apps Component Framework
- Also referred to as "Code Components"
- Pro-code
- Replace existing field controls and grids

Comparison with HTML Web Resources

- HTML WebResources:
 - Usually involve separate files for HTML, CSS, JavaScript
 - May include translations using XML files
 - No specific build process needed unless using TypeScript already
- PCF:
 - Build process needed
 - Written in TypeScript
 - RESX files used for translations

Where can we learn about PCF?

- <https://docs.microsoft.com/en-us/powerapps/developer/component-framework>
 - Official documentation from Microsoft
- <https://pcf.gallery>
 - Pre-built components
 - Created by the community
 - Typically open-source projects, so you can see how they work

Places we can use PCF controls

- Model Driven Apps (Unified Interface only)
- Canvas Apps
- Power Apps Portals (in preview, since March 2021)
 - Need to assign permissions to Read the Web Resource table (entity) in the Web Roles
 - Need to create Entity Form Metadata or Web Form Metadata
- Custom Pages (preview feature)

PCF Development

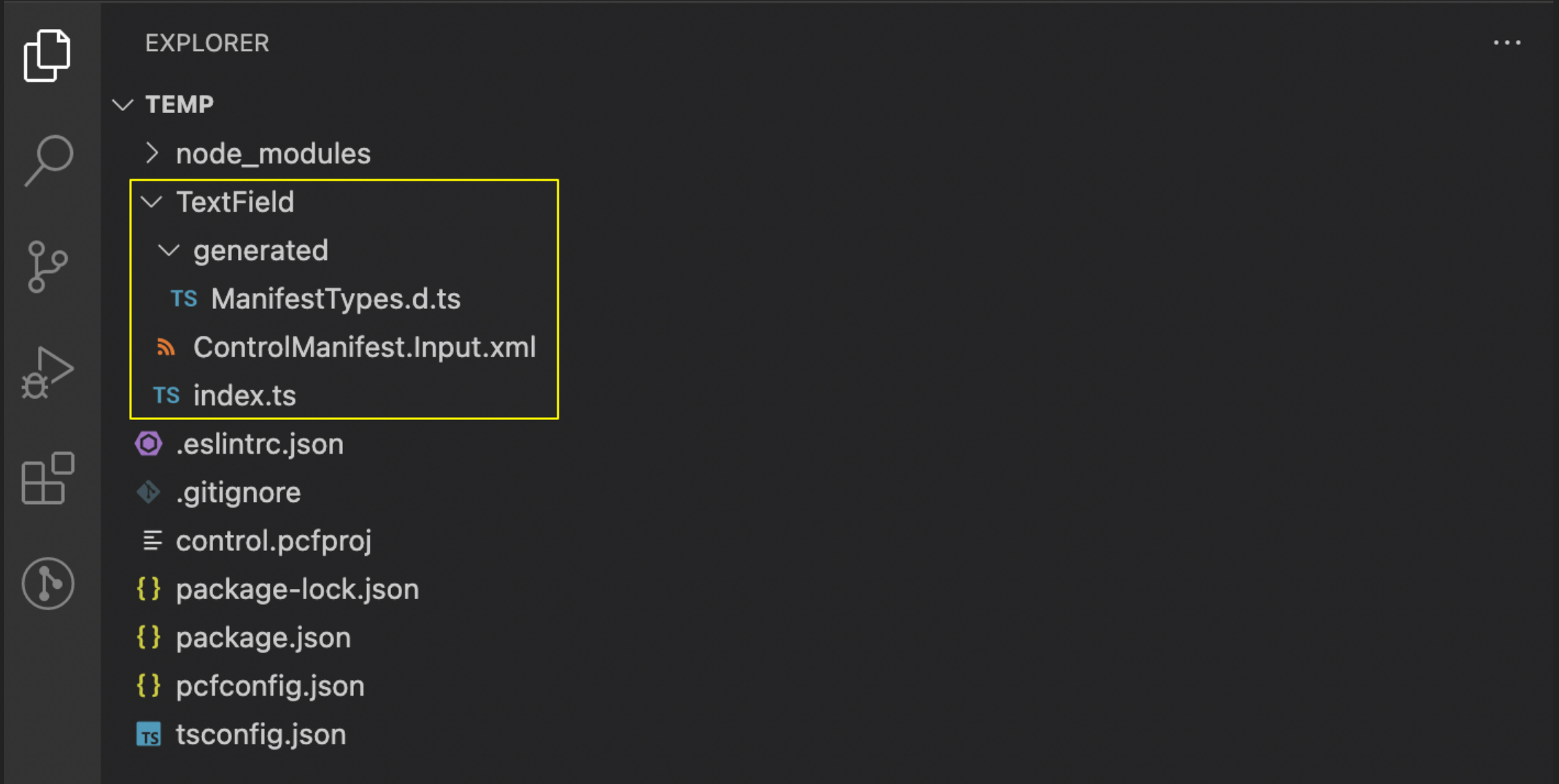
- What software is needed?
 - Node.JS (& npm)
 - dotnet
 - VS Code Extension, or Power Apps Command Line
- Other helpful tools
 - Fiddler AutoResponder, or Charles Proxy

Creating the PCF Project

- `pac pcf init`
 - `--name [-n]`
 - `--namespace [-ns]`
 - `--type [-t]`
 - Field or DataSet
- Command:
`pac pcf init --name TextField --namespace ppug --template field`

Project Structure

```
pac pcf init --name TextField --namespace ppug --template field
```

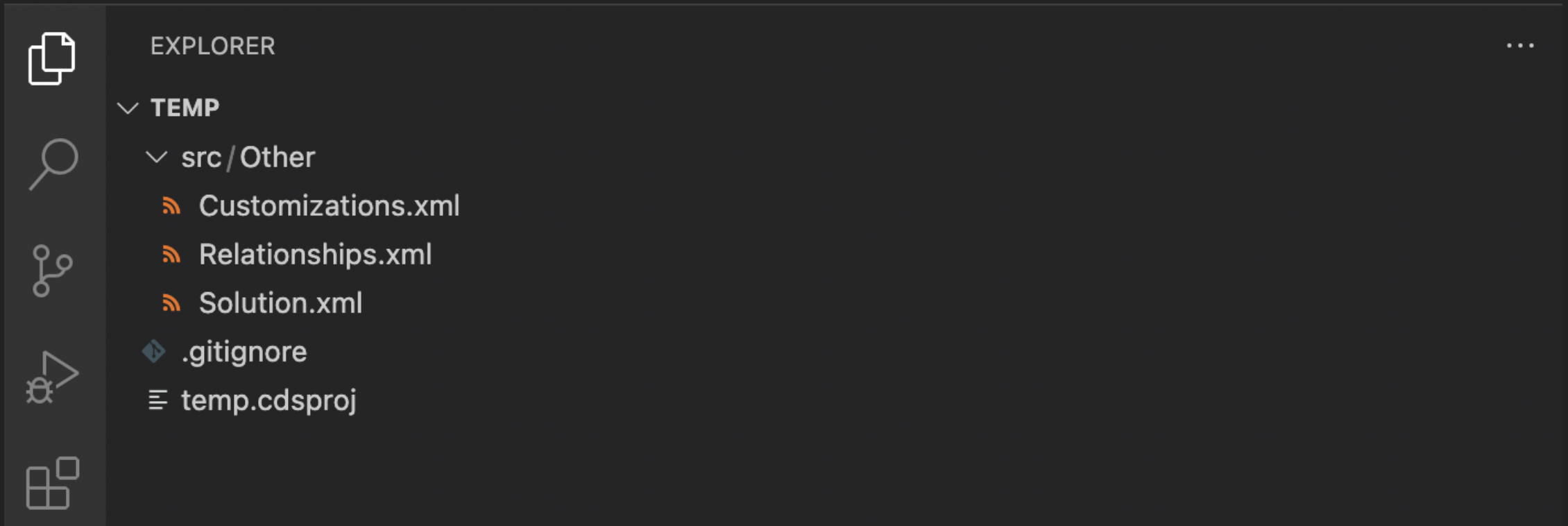


Creating the PCF Solution

- `pac solution init`
 - `--publisher-name [-pn]`
 - `--publisher-prefix [-pp]`
 - `--output-directory [-o]` (optional)
- Command:
`pac solution init --publisher-name PPUG --publisher-prefix ppug`

Solution Structure

```
pac solution init --publisher-name PPUG --publisher-prefix ppug
```



Solution XML

pac solution init --publisher-name PPUG --publisher-prefix ppug

🔥 Solution.xml ✕

src > Other > 🔥 Solution.xml > 🔑 xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <ImportExportXml version="9.1.0.643" SolutionPackageVersion="9.1" languagecode="1033" generatedBy="CrmLive"
3  <SolutionManifest>
4      <!-- Unique Name of Cds Solution-->
5      <UniqueName>temp</UniqueName>
6      <LocalizedNames>
7          <!-- Localized Solution Name in language code -->
8          <LocalizedName description="temp" languagecode="1033" />
9      </LocalizedNames>
10     <Descriptions />
11     <Version>1.0</Version>
12     <!-- Solution Package Type: Unmanaged(0)/Managed(1)/Both(2)-->
13     <Managed>2</Managed>
14     <Publisher>
15         <!-- Unique Publisher Name of Cds Solution -->
16         <UniqueName>PPUG</UniqueName>
17         <LocalizedNames>
18             <!-- Localized Cds Publisher Name in language code-->
19             <LocalizedName description="PPUG" languagecode="1033" />
20         </LocalizedNames>
```

Add the project to the solution

- Change directory into the folder containing the solution
- `pac solution add-reference`
 - `--path [-p]`
- Command:
`pac solution add-reference --path ../control`

Reference added to solution

pac solution add-reference --path ../control

EXPLORER

TEMP

- control
- node_modules
- TextField
- .eslintrc.json
- .gitignore
- control.pcfproj
- package-lock.json
- package.json
- pcfconfig.json
- tsconfig.json

solution

- src/Other
 - Customizations.xml
 - Relationships.xml
 - Solution.xml
- .gitignore
- solution.cdsproj

solution.cdsproj

```
35 <ExcludeDirectories Include="$(MSBuildThisFileDirectory)\*.cdsprc
36 <ExcludeDirectories Include="$(MSBuildThisFileDirectory)\*.sln" /
37 </ItemGroup>
38
39 <ItemGroup>
40 <None Include="$(MSBuildThisFileDirectory)\*" Exclude="@(<Exclude
41 <Content Include="$(SolutionPackageZipFilePath)">
42 <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>
43 </Content>
44 </ItemGroup>
45
46 <ItemGroup>
47 <ProjectReference Include="../control/control.pcfproj" />
48 </ItemGroup>
49
50 <Import Project="$(MSBuildToolsPath)\Microsoft.Common.targets" />
51 <Import Project="$(PowerAppsTargetsPath)\Microsoft.PowerApps.Visual
52
53 </Project>
54
```

Building and deploying the solution

- Change directory into the solution folder
- For unmanaged solution
 - `dotnet build`
- For managed solution
 - `dotnet build -p:Configuration=Release`

Demo

TS index.ts X

control > TextField > TS index.ts > ...

```
1  import { IInputs, IOutputs } from "../generated/ManifestTypes";
2
3  export class TextField implements ComponentFramework.StandardControl<IInputs, IOutputs> {
4
5      public init(context: ComponentFramework.Context<IInputs>,
6          notifyOutputChanged: () => void,
7          state: ComponentFramework.Dictionary,
8          container: HTMLDivElement): void {
9
10         // Add control initialization code
11     }
12
13     public updateView(context: ComponentFramework.Context<IInputs>): void {
14         // Add code to update control view
15     }
16
17     public getOutputs(): IOutputs {
18         return {
19         };
20     }
21
22     public destroy(): void {
23         // Add code to cleanup control if necessary
24     }
25 }
```

Home

Main

Accounts

Signatures

Important Things

←

↗

Save

Save & Close

+ New

Deactivate

Del

This is our PCF - Saved

Important Thing

General

Related

Name

This is our PCF

Demo

- Creating the project & solution
- Building out the source code
- Deploying the solution
- Tools for Local Development

Thank you for listening!

Slides & source code from the demo

<https://github.com/cathalnoonan/ppug-belfast-202110>

