

# Power Platform

## French Summit

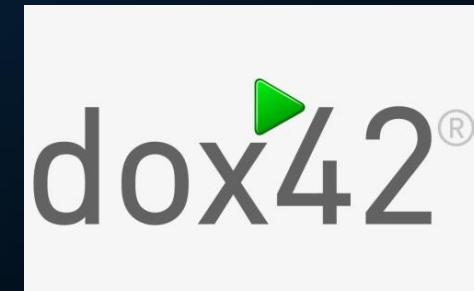
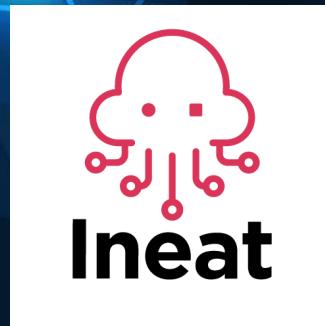


avec le soutien de Microsoft

# MERCI À NOS SPONSORS



Power Platform  
French Summit



# SUIVEZ-NOUS !

Power Platform  
French Summit



<https://www.linkedin.com/company/biz-apps-french-community>



@BizFrench



<https://www.linkedin.com/company/ams-community/>



@aMSComm2



<https://www.linkedin.com/groups/8599174/>



@ClubPowerBI

# INTRODUCING THE POWER APPS COMPONENT FRAMEWORK



# Allan De Castro

CRM Power Platform Consultant

[Paris Power Platform Happy Hour](#)

[Linkedin.com/in/allandecastro/](https://www.linkedin.com/in/allandecastro/)

[Twitter.com/decastroallan](https://twitter.com/decastroallan)

[www.blog.allandecastro.com](http://www.blog.allandecastro.com)



# AGENDA

- WHAT IS THE POWER APPS COMPONENT FRAMEWORK?
- LIVE DEMONSTRATION
- GETTING STARTED WITH PCF (A.K.A. GOODBYE WEB RESOURCES)
- QNA

# WHAT IS THE POWER APPS COMPONENT FRAMEWORK?

# WHAT IS POWER APPS?

Power Platform  
French Summit



**Power BI**  
Business analytics



**Power Apps**  
Application development



**Power Automate**  
Workflow automation



**Power Virtual Agents**  
Intelligent virtual agents

Power Apps Component Framework



Data connectors



AI Builder

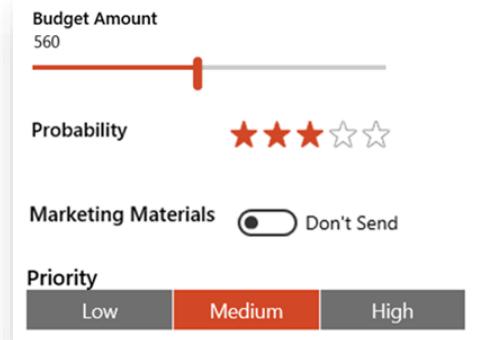


Dataverse

# HISTORY

## First Components introduced in 2016

Set of optimized mobile controls



## Unified Interface -> Components for all

At that time we were talking about Custom Control using an exclusive Microsoft framework

## Power Platform, renamed into Power Apps Component Framework

Custom Control → PCF → Power Apps Component Framework

# DEFINITION

Model Driven App, Canvas Apps and Portal!

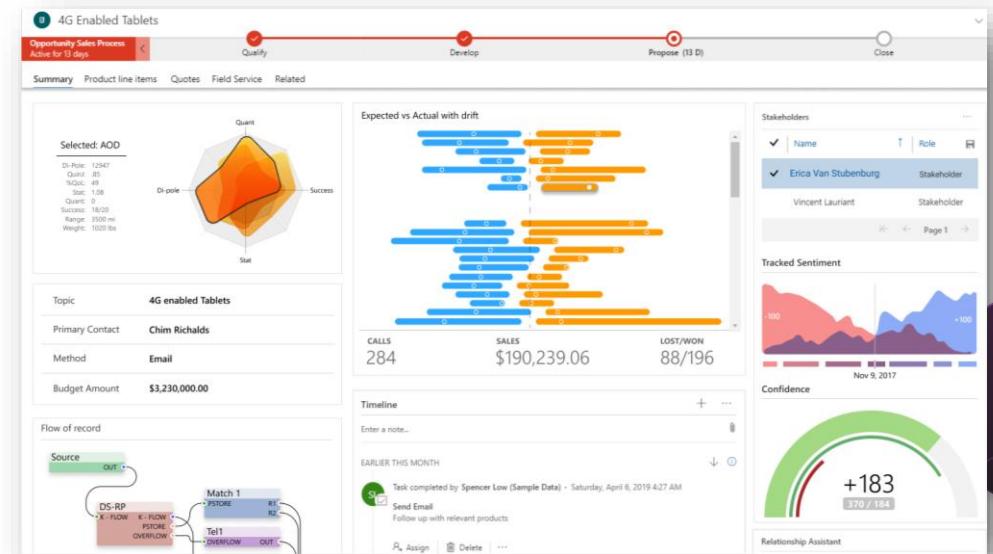
1st and 3rd party developers

Professionally developed controls with well defined contracts, lifecycle and interaction model

Improves performance

Create your own user experience

Modern Tooling (CLI)

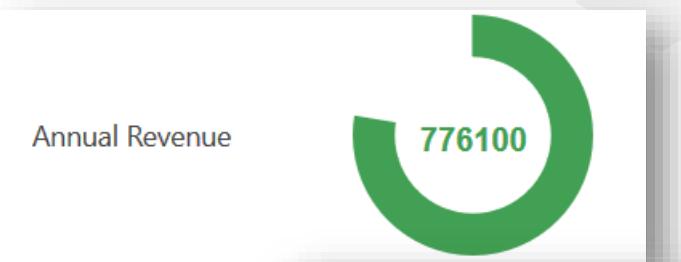


# OUT-OF-THE-BOX COMPONENT

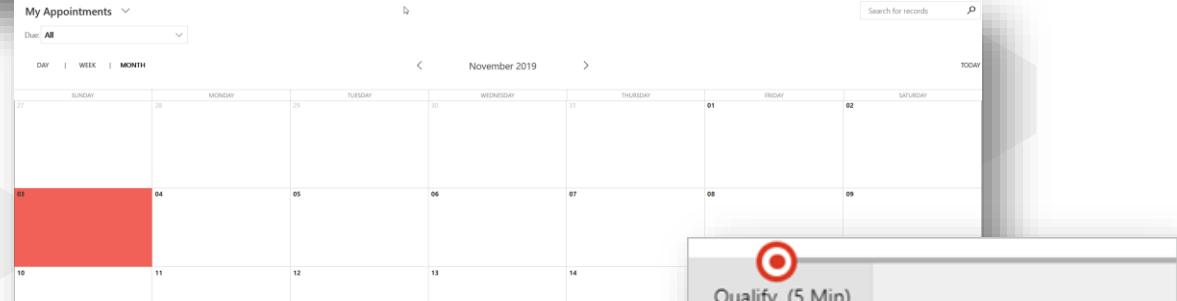
# Power Platform

## French Summit

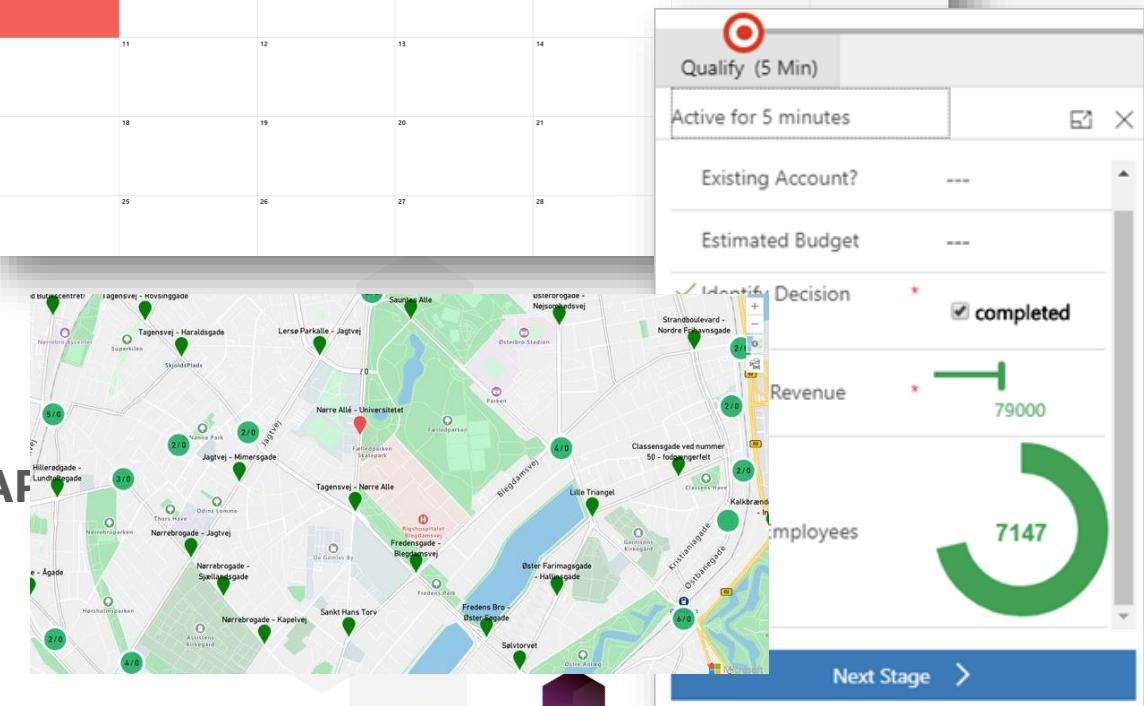
## Field on the form



## DataSet (Entity Grid/SubGrid)



# Business Process step



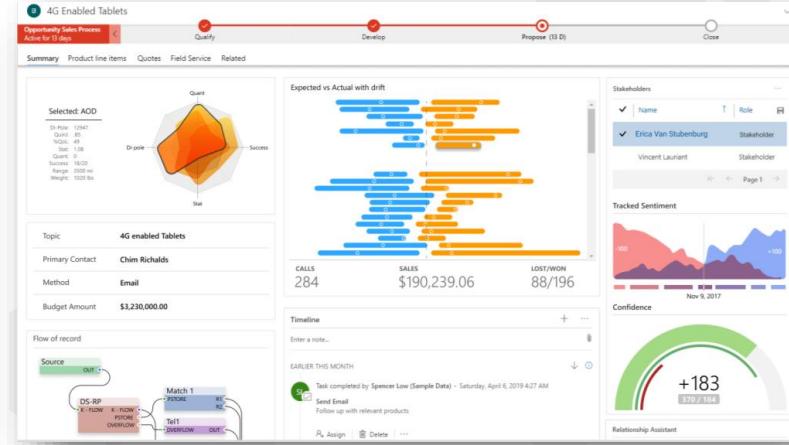
# Display/Use external content (LinkedIn, API, Azure Maps...)

# AVAILABILITY

Power Platform  
French Summit

Public Preview, April 23, 2019

The PowerApps component framework and the PowerApps CLI is now available for public preview

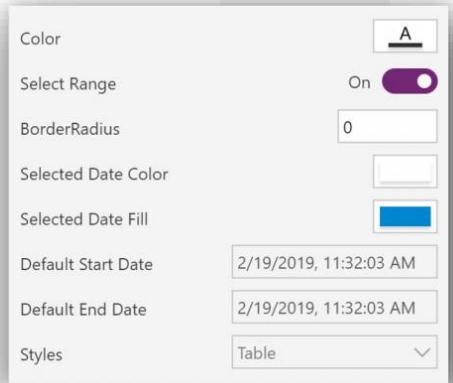
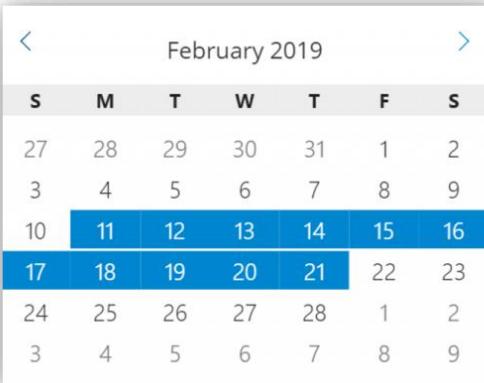


General Availability, October 1, 2019

Announcing the general availability of the PowerApps component framework for model-driven applications and PowerApps CLI

# TWO TYPES OF COMPONENTS

Power Platform  
French Summit



**Custom Canvas Compositions :** Low-code composition using expressions and connectors (Canvas App).

Stage	Summary	Contact	Account	Est. revenue	Est. close date
Qualify	\$3,000	Lesa Byrd	Adventure works	\$3,000	3/31/2019
Develop	New store opened this year - follow up	Marla Chan	Blue Yonder Airline	\$45,000	3/30/2019
Propose	12 order of product CD121234	Milton Philips	Contoso Pharma	\$40,000	3/29/2019
Close	Reorder of all items	Stuart Silas	A. Datum Corp	\$57,000	3/15/2019

**Custom Pro Components :** Create more sophisticated component using code (Model Driven App, Canvas App & Portal).

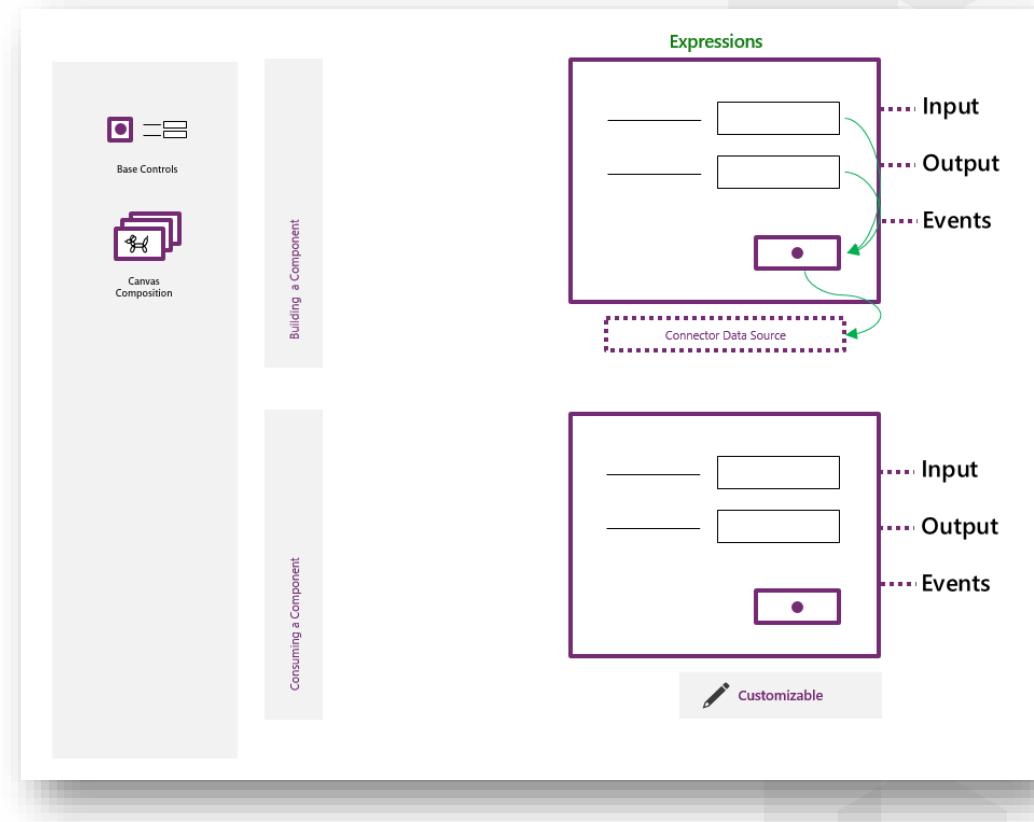
# CUSTOM CANVAS COMPOSITIONS

Power Platform  
French Summit

Allows you to connect the **basic** commands between base control and the connectors using **expressions**.

Use of OOB controls available, enriched with expressions and connected data (using connector) to produce **reusable components**.

Input and output properties to **interact** with the component by passing data (can be used to change the visual aspect).



*Business App Summit 2019: Microsoft PowerApps:  
Building and Extending Applications*

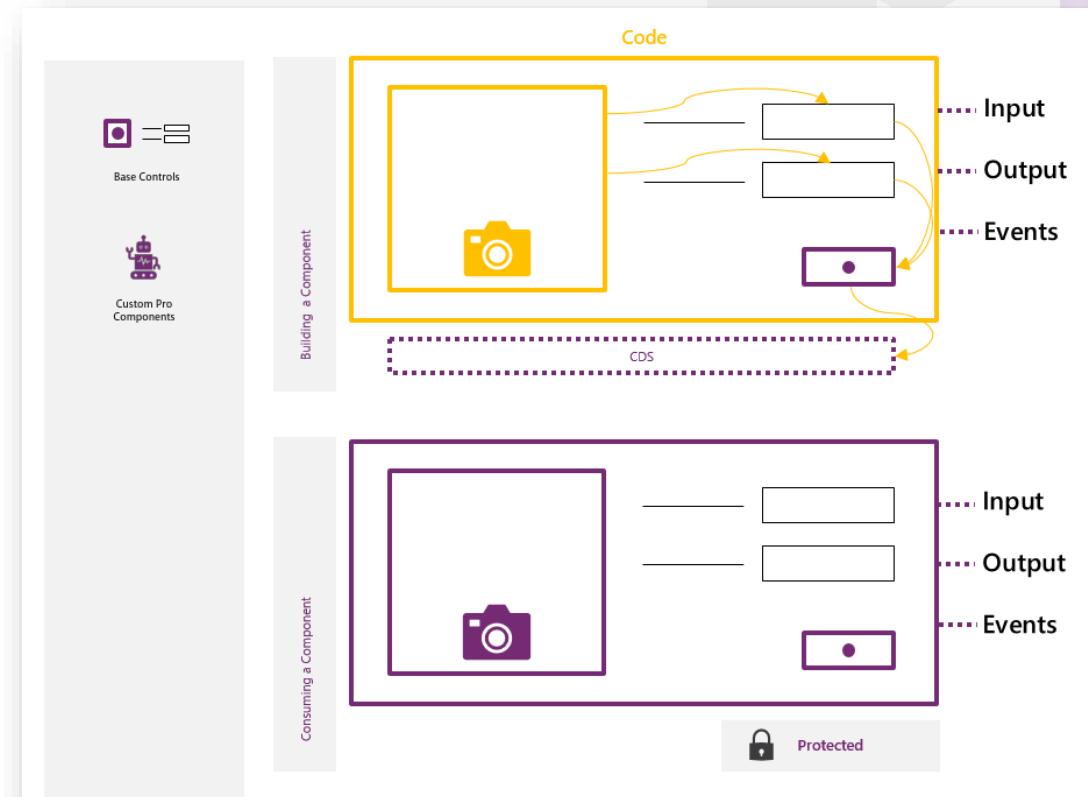
# CUSTOM PRO (DEV) COMPONENTS

Power Platform  
French Summit

Using **code** to connect base controls.

Custom Pro (Dev) Components use input and output properties and can **connect natively to CDS**.

Custom Pro (Dev) Components are consumed as **black box components** to Makers, and can't be **customized through maker experiences**, except for developer designated properties.



Business App Summit 2019: Microsoft PowerApps:  
Building and Extending Applications

# RESUME

Power Platform  
French Summit

An **extensibility** Framework **for professional developer** to create custom component and **user experiences** to be **leveraged by citizen developers**:



## Conception

- Using full HTML, JavaScript, React.. to create custom UI.
- Access to an extensive API Surface.



## Packaging

- PowerApps CLI (Managed & UnManaged).



## Installation

- Rich ecosystem of reusable, accessible, and performant UI component.
- Import into CDS Environment



## Configuration

- Configurable by Makers



## Client Rendering

- Both App + Portal
- Adapts according to the configuration
- Improves performance

# EXAMPLES

Power Platform  
French Summit

**Products By Industry**

**Disposable Tableware**

- Food Service
- Disposable Lids**
- Disposable Serveware**
- Disposable Tableware**
- Food Service Baskets
- Heavy Machinery
- Chippers

**Indie Flower** Set font

PowerFont sample ← PCF Component

This is a test ← HTML Text control

EAN13

5 901234 123457

EAN13 customized

Custom Text

**PCF Controls**

Siren 380129866

Is Valid Siren Yes

**France** +33 6 34 55 66 77

**USA** +44 20 8366 1177

**Algeria** +213 634 55 66 77

**Swiss** +41 44 668 18 00

**Italy** +39 365 478 9145

**Portugal** +351 22 040 4000

**Component**

Azure PaaS Azure SaaS Field Services Finance and Operations IoT  
Marketing Power BI Project Service Automation Retail

**Initiative Details** **Milestones** Investment KPI's Document Library Resources / Enablers M&A Digital Related

1 - Make sure you have required items (milk, eggs, flour) 100%

Watch latest favorite series episode 20%

Handle kids requests with love & patience 45%

2 - Mix flour and eggs with 150 ml of milk and a 100%

4 - Let dough stand for some time 40%

3 - Gradually add 350 ml of milk 100%

5 - Pour a ladle into a hot pan and repeat until the dough runs out. 0%

22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7

September 2019 October 2019

**RECENT OPPORTUNITIES**

Interested to buy something!

Qualify Develop Propose Close

I really need to buy something!

Qualify Develop Propose Close

Let me tell you once more: I WANT TO BUY SOMETHING

Qualify Develop Propose Close

Another BPF with one more stage

Qualify Develop Propose Close Approval

**RECENT CASES**

I still haven't been able to buy something

Identify Research Resolve

**OPPORTUNITIES NEXT MONTH**

Qualify Develop Propose Close

# LIVE DEMONSTRATION

- **Community Stuff**
- **Deployment and configuration**
- **PhoneNumberByCountry**
- **VATNumberValidator**
- **BusinessProcessFlowViewer**

# EXAMPLES

Power Platform  
French Summit

PCF Gallery by Guido Preite @crmanswers



The screenshot displays the PCF Gallery application, which is a collection of various Power Platform controls and examples. The controls shown include:

- Notification Control**: A control for managing notifications, showing a camera feed and message history.
- Gantt Progress Timeline**: A Gantt chart showing project timelines and progress.
- CONTACT**: A contact card for "Hello Bartle" with sections for Summary, Details, and Related, and a Connected Users list.
- Image To Text Control**: A control that extracts text from an image, showing a sample image with text like "WE CREATED MYTHS" and "WE ARE NOW REDESIGNING LIFE".
- PCF AppInsights**: A control for monitoring Application Insights data.
- Energy Efficiency Rating Control**: A control for displaying energy efficiency ratings with a slider and bar chart.
- All Tasks - Kanban**: A Kanban board for tracking tasks across columns: Not Started, In Progress, and Waiting on someone else.
- Status Reason Kanban**: A Kanban board for tracking status reasons.
- Immersive Reader Control**: A control for reading text aloud, showing a sample sentence: "The quick brown fox jumps over the lazy dog. The cow then moos. But what does the fox say?"
- RECENT OPPORTUNITIES**: A list of recent opportunities with stages: Quality, Develop, Propose, and Close.
- Business Process Flow Viewer**: A viewer for Business Process Flows (BPFs).
- YouTube Player**: A control for embedding YouTube videos.
- Attachment Manager**: A control for managing attachments, showing a list of files.
- Rich Text Editor with Chart & UML Designer**: A control for rich text editing with chart and UML diagram design capabilities.



**DEMO**

# EXAMPLES

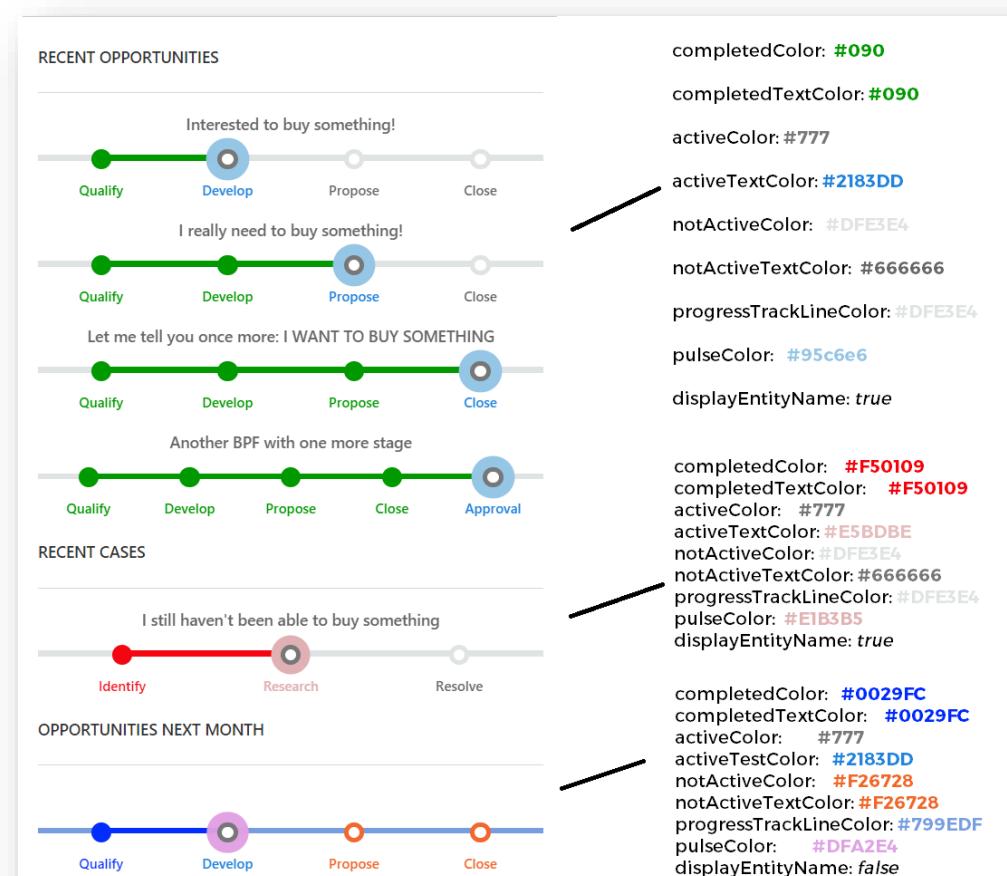
## Business Process Flow Viewer

Displays BPFs with all process stages OR categories of entities from a grid/views.

### Benefits:

- Improves information's visualization
- Fully configurable
- Supports any entities (OOB, custom...)
- Supports any BPFs (OOB, custom...)
- FREE!

<https://github.com/allandecastro/BusinessProcessFlowViewer>



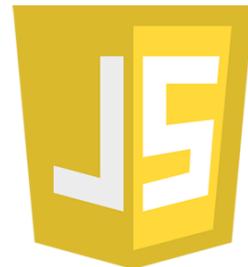
# GETTING STARTED WITH PCF (A.K.A. GOODBYE WEB RESOURCES)

- Technologies
- Approach
- Tools & Architecture

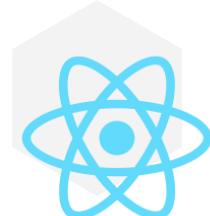
# TECHNOLOGIES

Power Platform  
French Summit

Many **advanced technologies** to take these components to the **next level!**



TypeScript

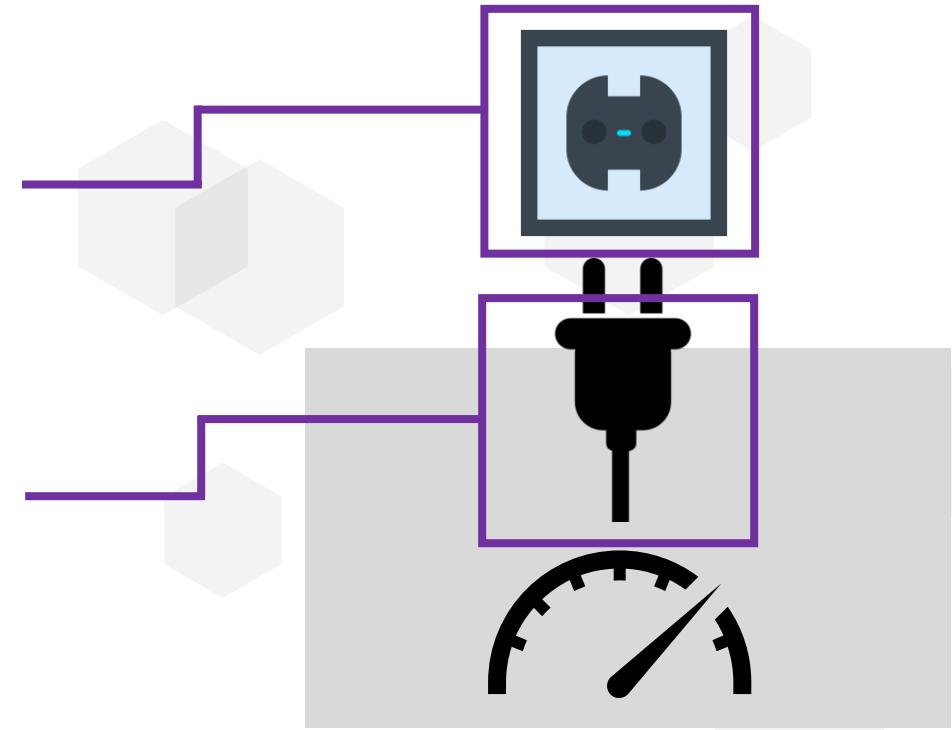


# INTERFACE

## Framework Interface aka Socket

Interface API SDK provides methods for control lifecycle, data/metadata access and more.

Component framework implement interface and functional logic.



# THINKING BEFORE ACTING

What problem  
do I want to  
answer ?



What is the  
added value  
of building a  
component?



What features  
must be  
implemented  
to take full  
advantage of  
this  
component?



What types of  
data will it  
use?  
Will it be used  
on a field or  
on a dataset?



What  
parameters do  
I want to keep  
free to  
modify?



External  
services ?



# EQUIP YOURSELF!

Power Platform  
French Summit

Install Npm or Node.js, one will come with the other in any case!

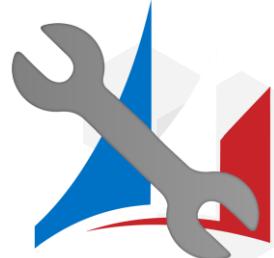


Make sure you have the .NET Framework 4.6.2 Developer Pack.



Install Microsoft PowerApps CLI.

Note: XrmToolBox plugin exists PCF Custom Control Builder.



# POWER APPS CLI

Power Platform  
French Summit

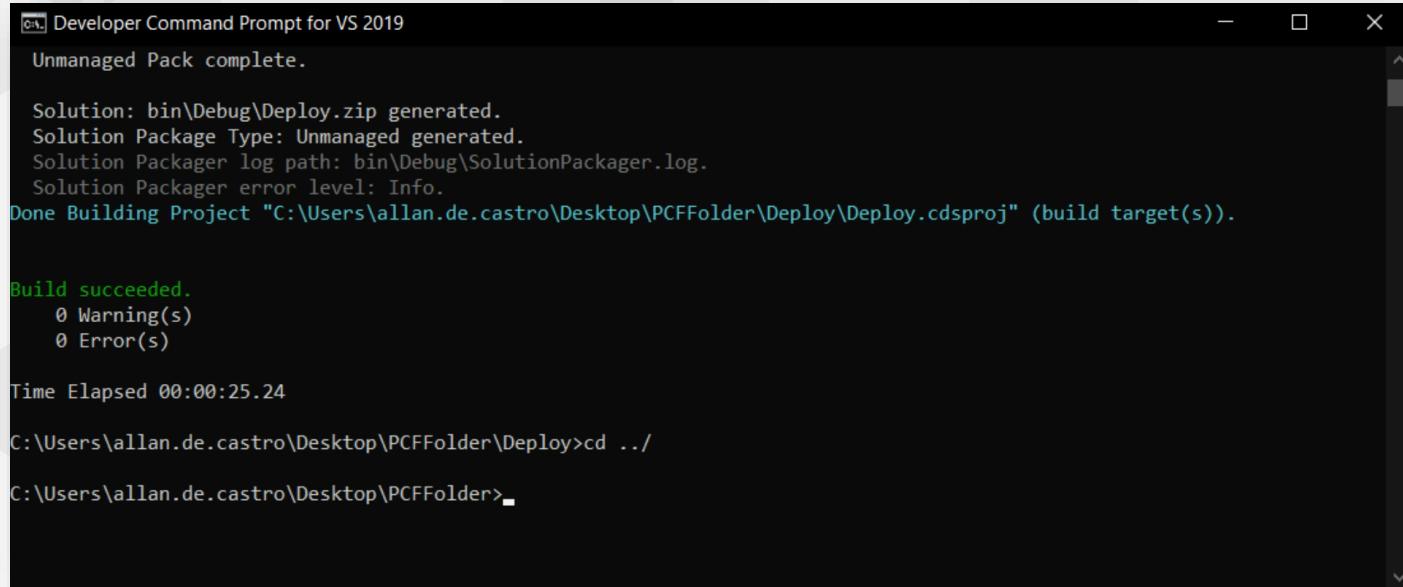
A unique tool that initially focuses on developing **custom controls, plugins and web resources.**

Command Line Interface which enables developers to be more efficient in their PCF development !

Control templates (field and dataset).

Build-in validations to catch issues prior to deployment.

Can be used as part of your automated processes.



```
Developer Command Prompt for VS 2019
Unmanaged Pack complete.

Solution: bin\Debug\Deploy.zip generated.
Solution Package Type: Unmanaged generated.
Solution Packager log path: bin\Debug\SolutionPackager.log.
Solution Packager error level: Info.
Done Building Project "C:\Users\allan.de.castro\Desktop\PCFFolder\Deploy\Deploy.cdsproj" (build target(s)).

Build succeeded.
0 Warning(s)
0 Error(s)

Time Elapsed 00:00:25.24

C:\Users\allan.de.castro\Desktop\PCFFolder\Deploy>cd ../
C:\Users\allan.de.castro\Desktop\PCFFolder>
```

# XRMTOOLBOX PLUGIN BY DANISH

Power Platform  
French Summit



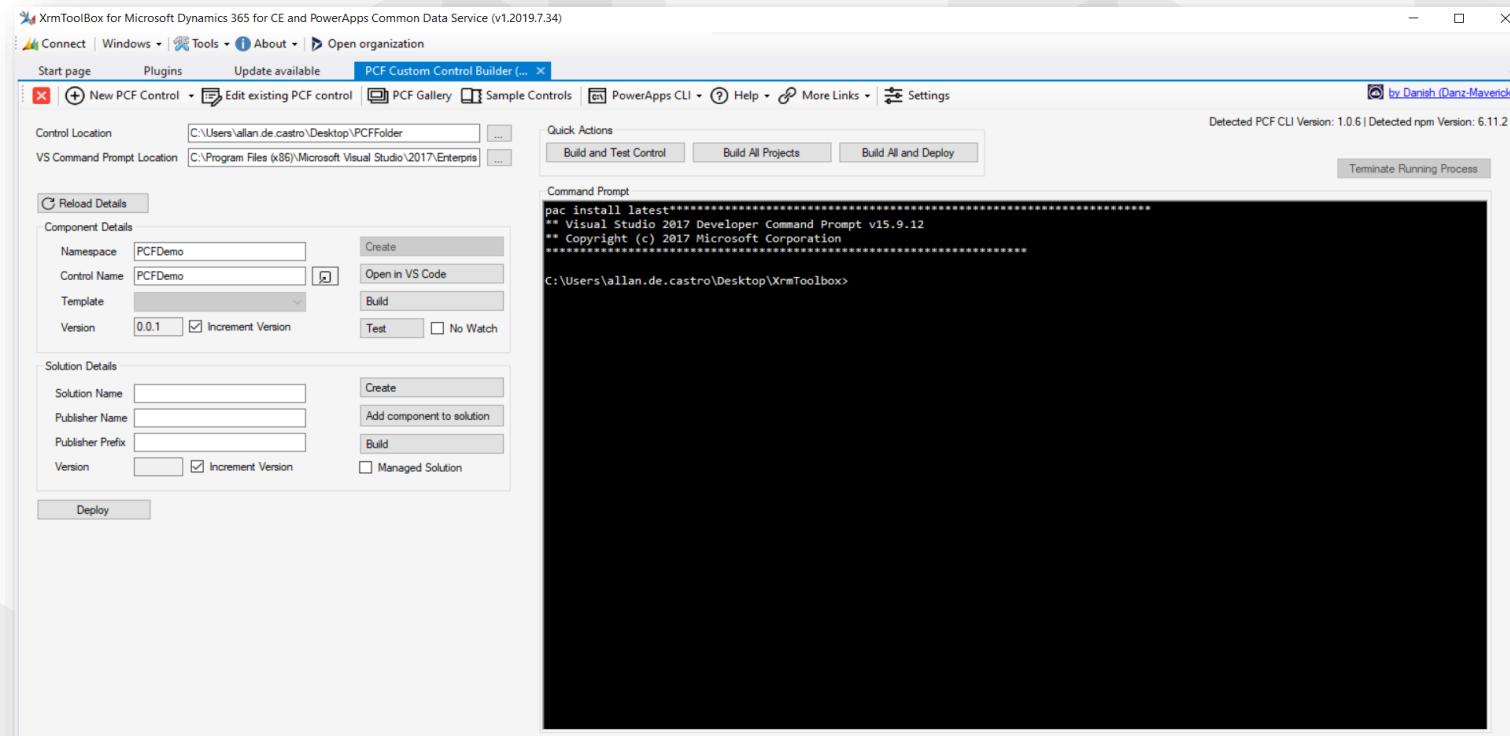
## PCF Custom Control Builder

Version: 2.2019.10.22

Author: Danish Naglekar

Easily create, build and deployment solution for your custom control using PCF.

Facilitates the **creation**, **building** and **deployment** of custom PCF controls by eliminating the need to write commands.



# WHAT MAKES A CUSTOM COMPONENT?

## Manifest Input File

Control Definition:

- Name
- Version
- Properties
- Resource files

## Component Implementation

Code:

- TypeScript / JavaScript
- User Interface
- Functionalities

## Resource Files

Static Resources

- CSS
- Images
- ...

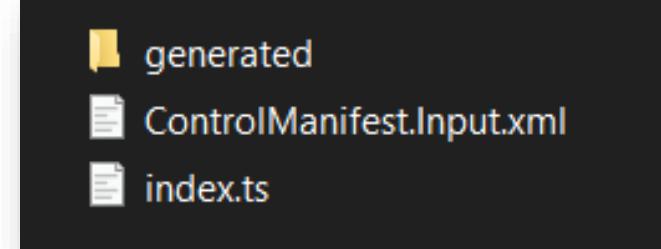
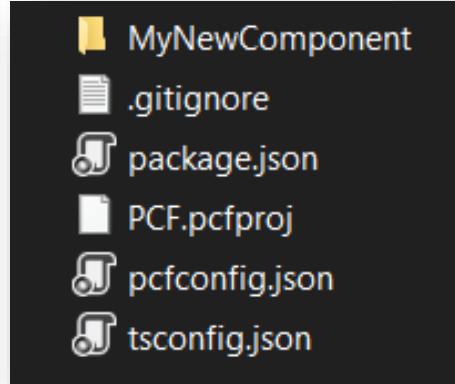
# CREATING A COMPONENT

```
pac pcf init --namespace MyNewComponentNameSpace --name  
MyNewComponent --template field
```

Namespace for this component

Name of the component

Type of PCF project



Note: npm install to ensure to get required project dependencies.

# MANAGE PROPERTIES & RESOURCES

```
<?xml version="1.0" encoding="utf-8" ?>
<manifest>
  <control namespace="BusinessProcessFlowViewer" constructor="BusinessProcessFlowViewer" version="1.0.0" display-name="Business Process Flow Viewer">
    <data-set name="dataSet" display-name-key="DataSet View Used" cds-data-set-options="displayCommandBar:true" />
    <property name="parametersBPF" display-name-key="parametersBPF" description-key="This parameter is used to define the parameters for the Business Process Flow viewer." />
    <property name="completedColor" display-name-key="completedColor" description-key="This Hex Color define the color for completed tasks." />
    <property name="completedTextColor" display-name-key="completedTextColor" description-key="This Hex Color define the text color for completed tasks." />
    <property name="activeColor" display-name-key="activeColor" description-key="This Hex Color define the background color for active tasks." />
    <property name="activeTextColor" display-name-key="activeTextColor" description-key="This Hex Color define the text color for active tasks." />
    <property name="notActiveColor" display-name-key="notActiveColor" description-key="This Hex Color define the background color for not active tasks." />
    <property name="notActiveTextColor" display-name-key="notActiveTextColor" description-key="This Hex Color define the text color for not active tasks." />
    <property name="progressTrackLineColor" display-name-key="progressTrackLineColor" description-key="This Hex Color define the color for the progress track line." />
    <property name="pulseColor" display-name-key="pulseColor" description-key="This Hex Color define the pulse color for the progress track line." />
    <property name="displayEntityName" display-name-key="displayEntityName" description-key="This value is used to display the entity name in the process flow." />
    <resources>
      <code path="index.ts" order="1"/>
      <css path="css/style.css" order="1" />
      <img path='img/preview.png' />
    </resources>
    <feature-usage>
      <uses-feature name="WebAPI" required="true" />
    </feature-usage>
  </control>
</manifest>
```

Control	Web	Phone	Tablet	
Read-only Grid (default)	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	
BusinessProcessFlowViewer	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<span>X</span>
<a href="#">Add Control...</a>				
BusinessProcessFlowViewer				
activeTextColor *	#2183DD (SingleLine.Text)	<span style="color: #2183DD;">#2183DD</span>	<span style="color: #2183DD;">#2183DD</span>	<span style="color: #2183DD;">#2183DD</span>
notActiveColor *	#DFE3E4 (SingleLine.Text)	<span style="color: #DFE3E4;">#DFE3E4</span>	<span style="color: #DFE3E4;">#DFE3E4</span>	<span style="color: #DFE3E4;">#DFE3E4</span>
notActiveTextColor *	#666666 (SingleLine.Text)	<span style="color: #666666;">#666666</span>	<span style="color: #666666;">#666666</span>	<span style="color: #666666;">#666666</span>
progressTrackLineColor *	#DFE3E4 (SingleLine.Text)	<span style="color: #DFE3E4;">#DFE3E4</span>	<span style="color: #DFE3E4;">#DFE3E4</span>	<span style="color: #DFE3E4;">#DFE3E4</span>
pulseColor	#95c6e6 (SingleLine.Text)	<span style="color: #95c6e6;">#95c6e6</span>	<span style="color: #95c6e6;">#95c6e6</span>	<span style="color: #95c6e6;">#95c6e6</span>
displayEntityName	true (SingleLine.Text)	<span style="color: green;">true</span>	<span style="color: green;">true</span>	<span style="color: green;">true</span>

# LIFE CYCLE OF A COMPONENT (INDEX.TS)

**init:** Initializes the control. Build the skeleton of its DOM and possibly call a logic to initialize what your component should do.

**updateView:** Called when the value of a property are changed (dataset refreshed..).

**destroy:** Called when we decide to remove the component from the DOM.

**getOutputs:** Called when the output values are modified (usually the data is sent back to the application here, e.g a field that is modified).

```
/* initialize component instance*/
init(context, notifyOutputChanged, state, container)

/* update call when value in the context changes */
updateView(context)

/* control is to be removed from the DOM tree*/
destroy()

/* updated values for the bound properties*/
getOutputs()
```

# LIFE CYCLE OF A COMPONENT (INDEX.TS)

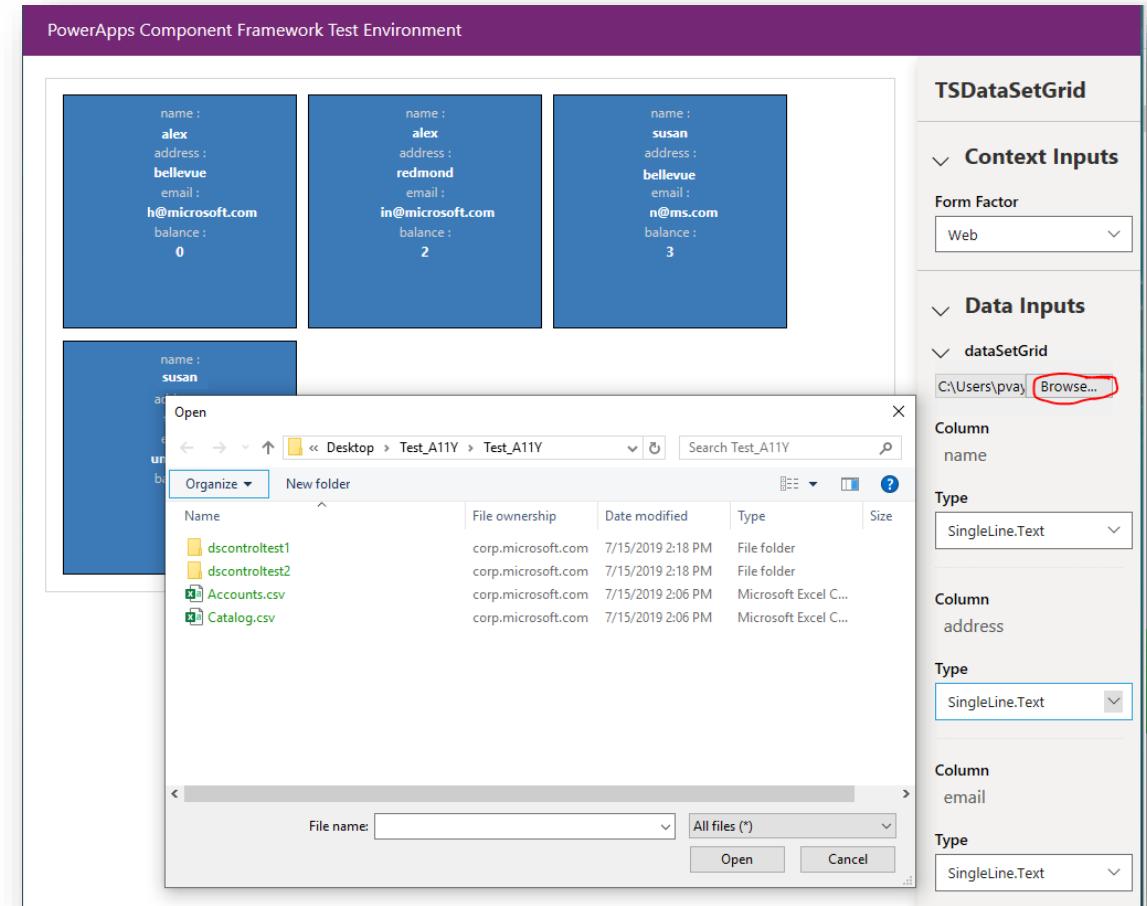


# BUILD & DEBUG

**npm run build**

**npm start watch** (or you can use  
start npmstart to open a new  
window)

Possibility to use fiddler:  
Debugging Canvas App Component

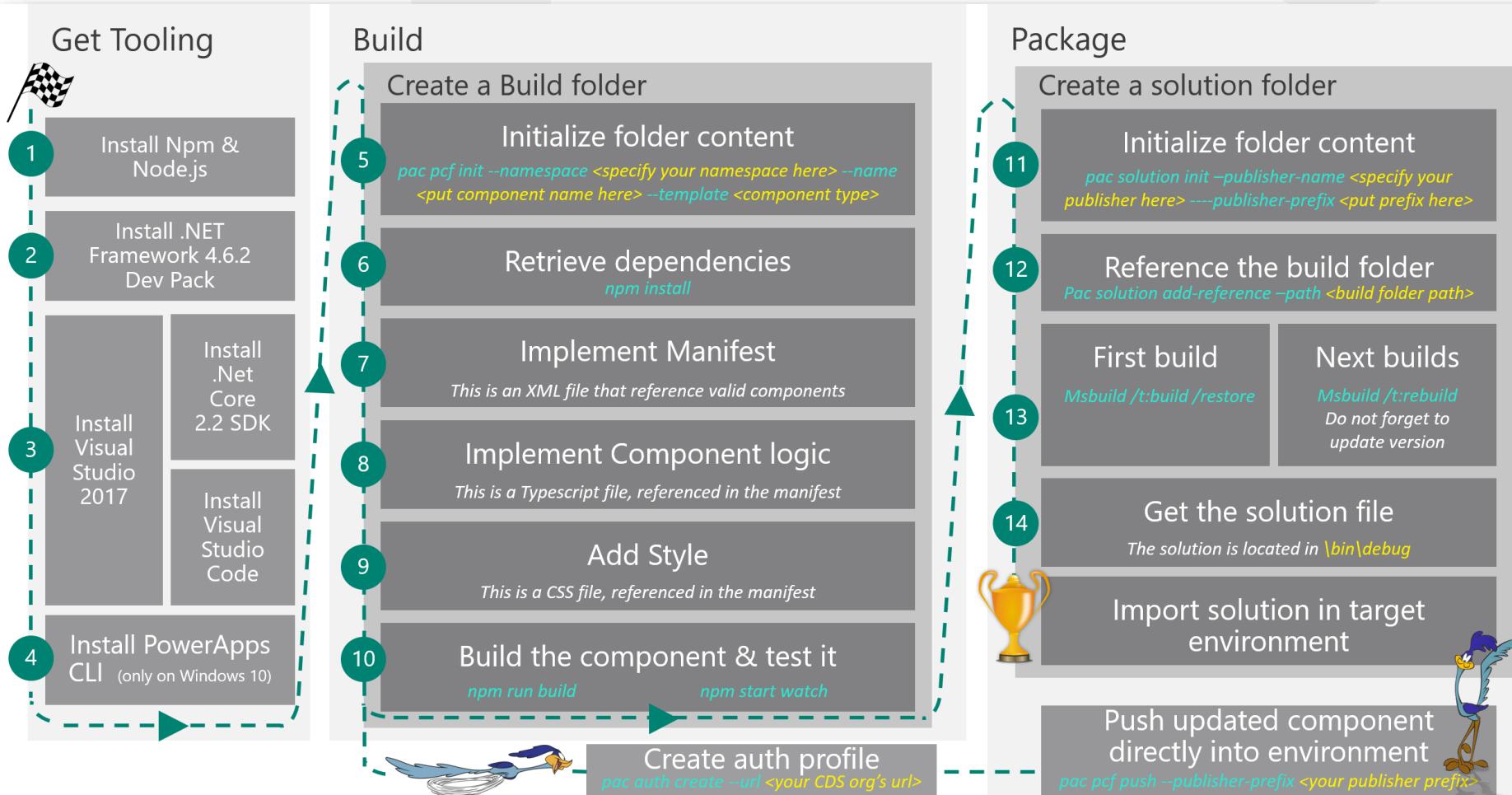


# PACKAGE A COMPONENT & IMPORT THE SOLUTION

Use CLI to create a solution (Managed/UnManaged) containing the custom control.

1. **cd** <solution folder>
2. **pac solution init** --publisher-name <enter your publisher name> --publisher-prefix <enter your publisher prefix> -- template <field or dataset>
3. **pac solution add-reference** --path <path of your PowerApps component framework project on disk>
4. **msbuild** /t:build /restore (msbuild must be in the path)
5. **msbuild** /t:build /restore **/p:configuration=Release** (If you want to build a managed package)
6. **Import** the solution into CDS Environment using the UI.

# HOW TO BUILD A PCF COMPONENT



Credit : Nicolas Kirrmann

# LINKS / REFERENCES 1/2

**PCF forum:** <https://aka.ms/PCFForum>

**Documentation:** <https://aka.ms/PCFDocs>

**Feature ideas:** <https://aka.ms/PCFIdea>

**Community components/blogs/demos:** <https://aka.ms/PCFDemos>

**Business App Summit 2019:** Microsoft PowerApps: Building and Extending Applications

**Ignite 2019:** Using Power Apps component framework to build reusable custom user experiences.

**Ignite 2020 :** Amplify your pro developer skills with Power Platform

# LINKS / REFERENCES 2/2

**Implementing Security:** <http://develop1.net/public/post/2019/11/21/time-to-add-some-finishing-touches-to-your-pcf-controls>

**Debugging PCF for Canvas App:** <http://develop1.net/public/post/2020/04/30/debugging-canvas-app-pcf-components-with-fiddler>

**Using CSS in PCF:** <https://benediktbergmann.eu/2020/01/04/pcf-css-for-model-driven-apps-mda/>

**Articles Series on my blog** (<https://blog.allandecastro.com>)

**Prelude:** [Getting started with PowerApps Component Framework](#)

**Episode 1:** [First Steps!](#)

**Episode 2:** [Focus on ... Control Manifest file!](#)

**Episode 3:** [Focus on ... Architecture & Component's life cycle!](#)

**Episode 4:** [Walkthrough to create your first PCF \(based on a field\)!](#)

MERCI POUR VOTRE  
ATTENTION.

