VISUAL STUDIO CODE LAB

LAB PREREQUISITES

If you want to do this lab using GitHub CodeSpaces:

- Have a GitHub Alias
- Create your own GitHub repository
- Download the samples solution and the PCF control from the lab GitHub repository
- Access to CodeSpaces
- Access to a Power platform environment (new environment is preferred)
    - The type of environment does not matter (It can Trial, Time bound Trial, Sandbox, and Production)
    - Import the Collaboration.zip file into your repository **(Note: do not add the PCF Control, if you plan to use GitHub CodeSpaces, as it has dependencies on .NET4.8 which is not available on Linux)**

If you want to do this lab with GitHub CodeSpaces

- Have a GitHub Alias
- Create your own GitHub repository
- Access to a Power platform environment (new environment is preferred)
    - The type of environment does not matter (It can Trial, Time bound Trial, Sandbox, and Production)
    - Import the Collaboration.zip file into your repository
- Have npm and node installed on your machine

INSTALLATION

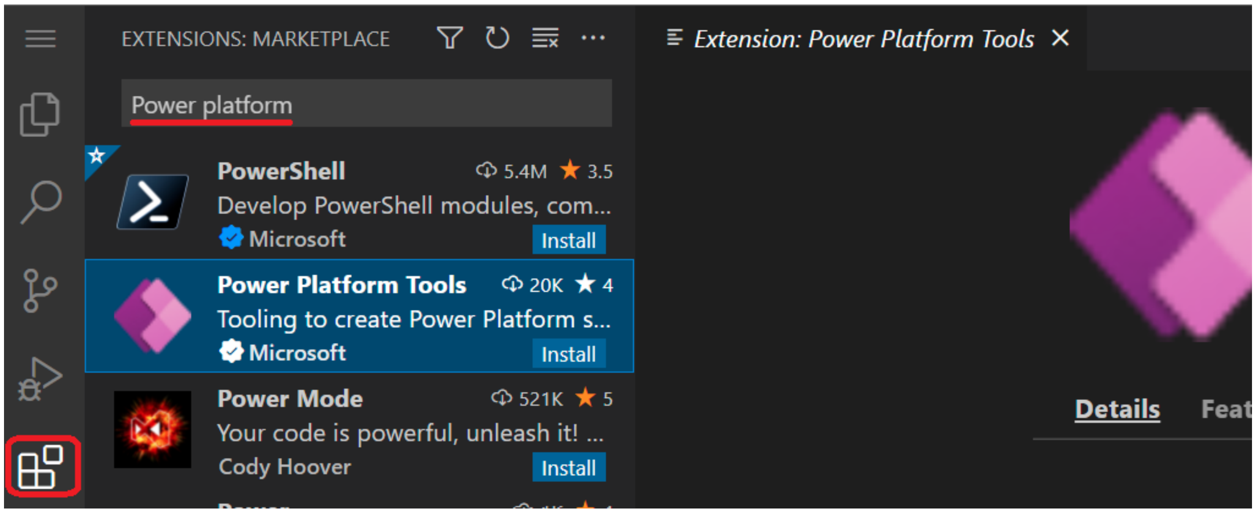
The installation of VS code extension can happen in 3 ways:

1) Install from within the VS Code Extension
2) Install from VS Code Market Place
3) Sideload the VSIX

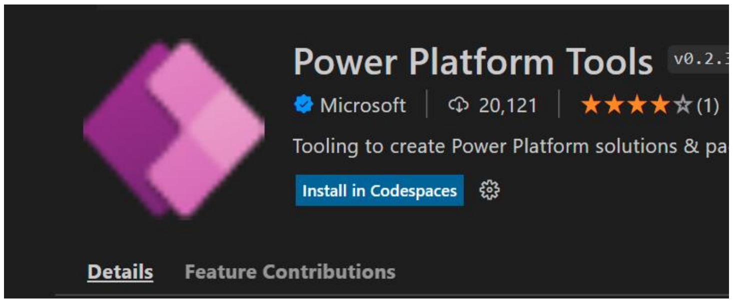
INSTALLATION FROM WITHIN VS CODE

Launch Visual Studio Code or Run GitHub CodeSpace (if you have access to Codespaces)

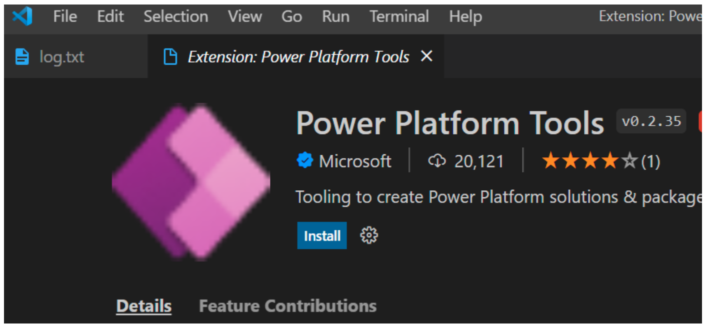
Then click on the extension icon on the Activity bar and search for Power Platform on the Extensions side bar.



If you are installing in GitHub CodeSpace, you will see the following option for installation



or you will see the following in your native VS Code instance

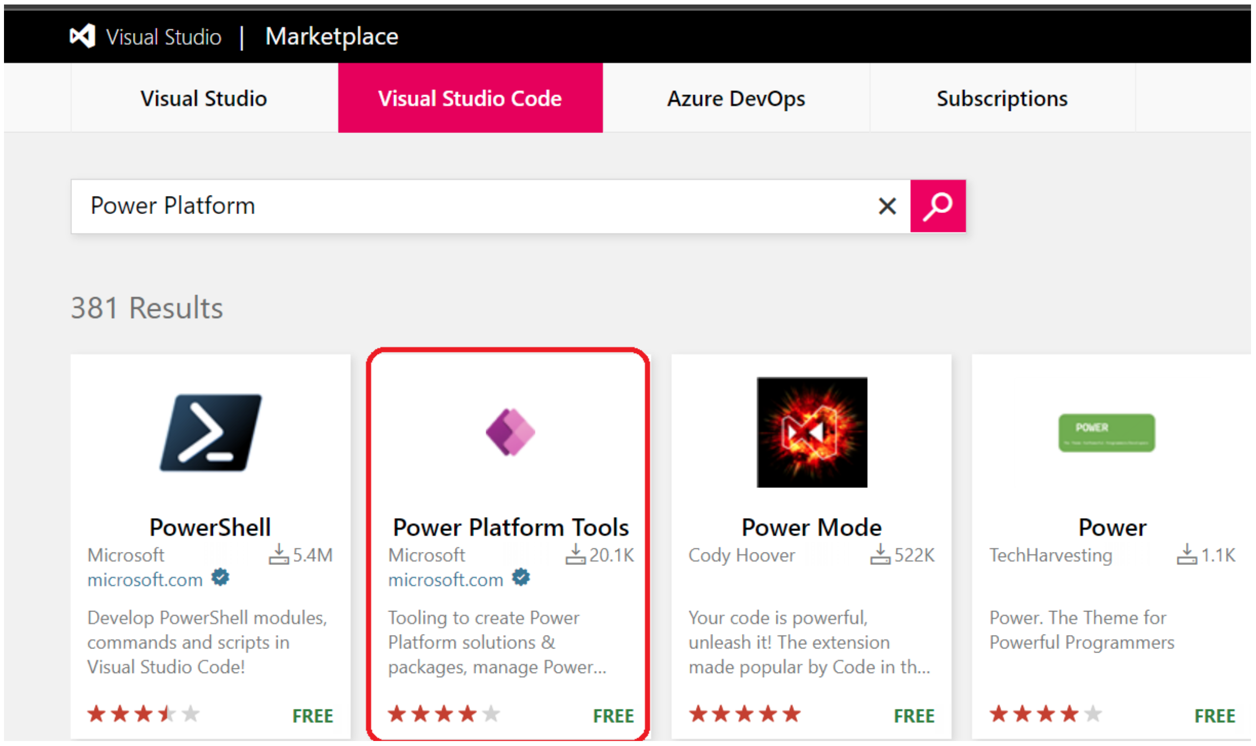


And the installation should be completed.

INSTALLING FROM VISUAL STUDIO MARKETPLACE

Go to the following URL: https://marketplace.visualstudio.com/

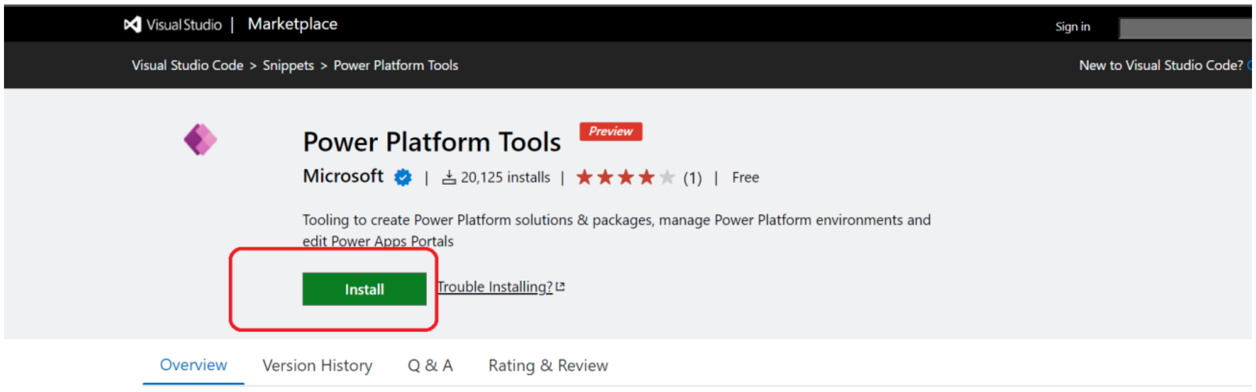
Select Visual Studio code and Search for Power Platform and Select the extension



381 Results

PowerShell	Power Platform Tools	Power Mode	Power
Microsoft 5.4M	Microsoft 20.1K	Cody Hoover 522K	TechHarvesting 1.1K
microsoft.com	microsoft.com		
Develop PowerShell modules, commands and scripts in Visual Studio Code!	Tooling to create Power Platform solutions & packages, manage Power...	Your code is powerful, unleash it! The extension made popular by Code in th...	Power. The Theme for Powerful Programmers
FREE	FREE	FREE	FREE

This will take you to the extensions page and from there you can launch your installation in your local instance of VS Code



Visual Studio | Marketplace
Sign in
Visual Studio Code > Snippets > Power Platform Tools
New to Visual Studio Code?
Preview
Install
Trouble Installing?
Overview
Version History
Q & A
Rating & Review

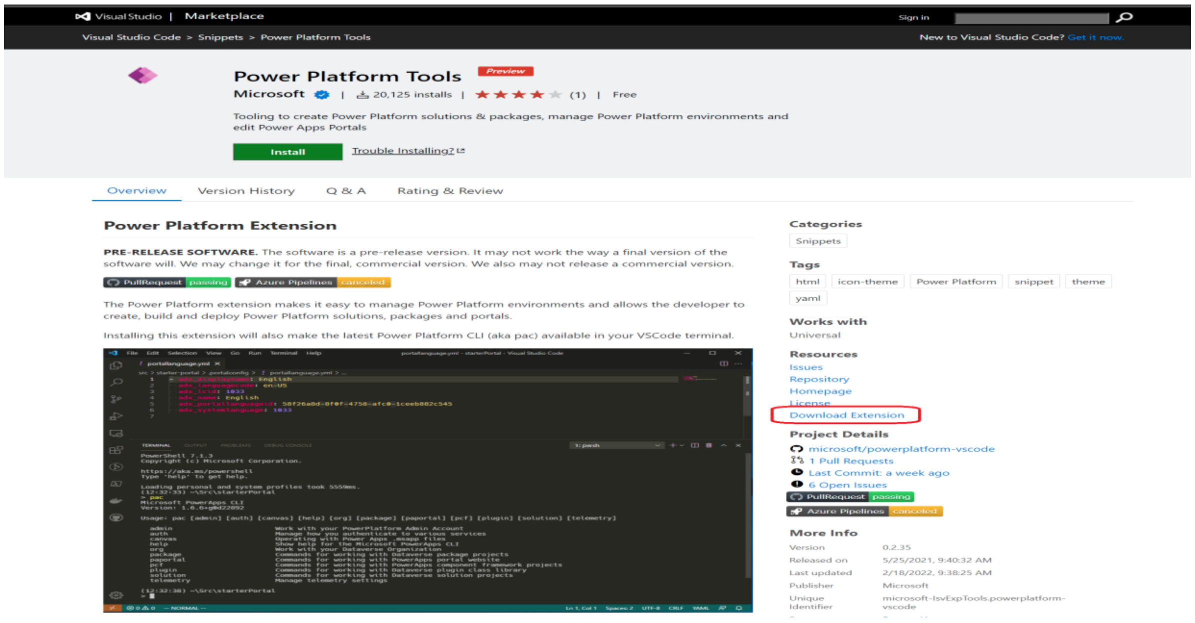
Power Platform Tools

Microsoft | 20,125 installs | (1) | Free

Tooling to create Power Platform solutions & packages, manage Power Platform environments and edit Power Apps Portals

SIDE LOADING AND INSTALLING THE EXTENSION

in some organizations, running an installation process directly off the web is prohibited. In most cases the organization downloads the installation in a secure location, tests it to make sure it does not violate company policy, and then makes it available to the rest of the organization. To do this, from the marketplace site for the extension, as mentioned in the prior step, instead of pressing the install button, you will select the Download extension link from the page.



Visual Studio | Marketplace — Sign in

Visual Studio Code > Snippets > Power Platform Tools — New to Visual Studio Code? Get it now.

Power Platform Tools *Preview*

Microsoft | 20,125 installs | ★★★★☆ (1) | Free

Tooling to create Power Platform solutions & packages, manage Power Platform environments and edit Power Apps Portals

Install — Trouble Installing?

Overview — Version History — Q & A — Rating & Review

Power Platform Extension

PRE-RELEASE SOFTWARE. The software is a pre-release version. It may not work the way a final version of the software will. We may change it for the final, commercial version. We also may not release a commercial version.

PullRequest passing — Azure Pipelines canceled

The Power Platform extension makes it easy to manage Power Platform environments and allows the developer to create, build and deploy Power Platform solutions, packages and portals.

Installing this extension will also make the latest Power Platform CLI (aka pac) available in your VSCode terminal.

```
portallanguage.yml - starterPortal - Visual Studio Code
src > starter-portal > .portalconfig > portallanguage.yml > ...
- adx_displayname: English
  adx_languagecode: en-US
  adx_lcid: 1033
  adx_name: English
  adx_portallanguageid: 58f26a8d-8f0f-4758-afc0-1ceeb882c545
  adx_systemlanguage: 1033
```

```
PowerShell 7.1.3
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

Loading personal and system profiles took 5559ms.
(12:32:33) ~\Src\starterPortal
> pac
Microsoft PowerApps CLI
Version: 1.6.6+g0d22892

Usage: pac [admin] [auth] [canvas] [help] [org] [package] [paportal] [pcf] [plugin] [solution] [telemetry]

  admin                      Work with your PowerPlatform Admin Account
  auth                       Manage how you authenticate to various services
  canvas                     Operating with Power Apps .msapp files
  help                       Show help for the Microsoft PowerApps CLI
  org                        Work with your Dataverse Organization
  package                    Commands for working with Dataverse package projects
  paportal                   Commands for working with PowerApps portal website
  pcf                        Commands for working with PowerApps component framework projects
  plugin                     Commands for working with Dataverse plugin class library
  solution                   Commands for working with Dataverse solution projects
  telemetry                  Manage telemetry settings

(12:32:38) ~\Src\starterPortal
>
```

Categories
Snippets

Tags
html — icon-theme — Power Platform — snippet — theme — yaml

Works with
Universal

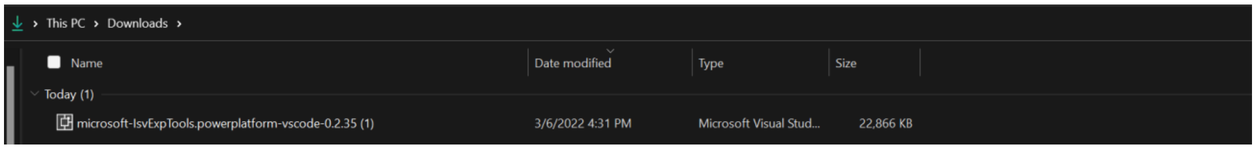
Resources
Issues
Repository
Homepage
License
Download Extension

Project Details
microsoft/powerplatform-vscode
1 Pull Requests
Last Commit: a week ago
6 Open Issues
PullRequest passing
Azure Pipelines canceled

More Info

Version	0.2.35
Released on	5/25/2021, 9:40:32 AM
Last updated	2/18/2022, 9:38:25 AM
Publisher	Microsoft
Unique Identifier	microsoft-IsvExpTools.powerplatform-vscode

This will download a VSIX file, on your workstation, on PC it downloads to the Download folder.

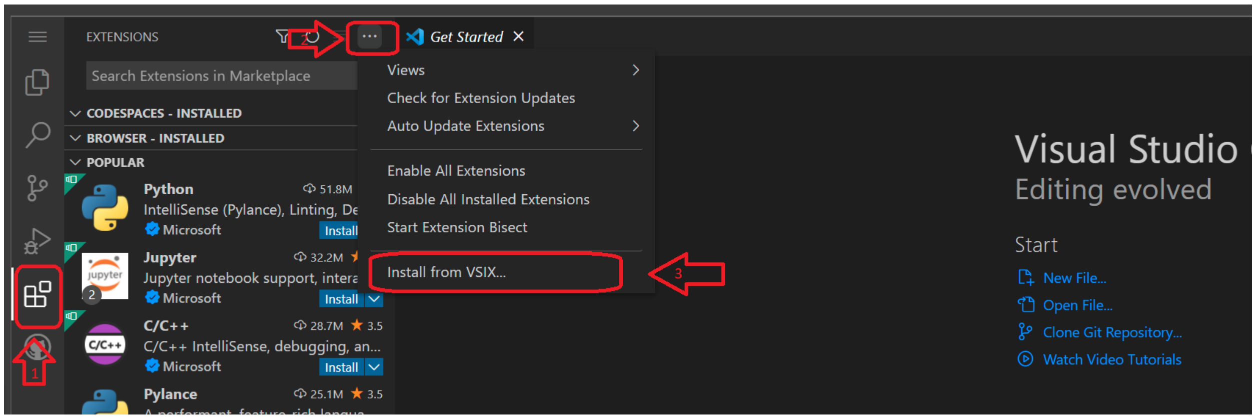


This PC > Downloads >

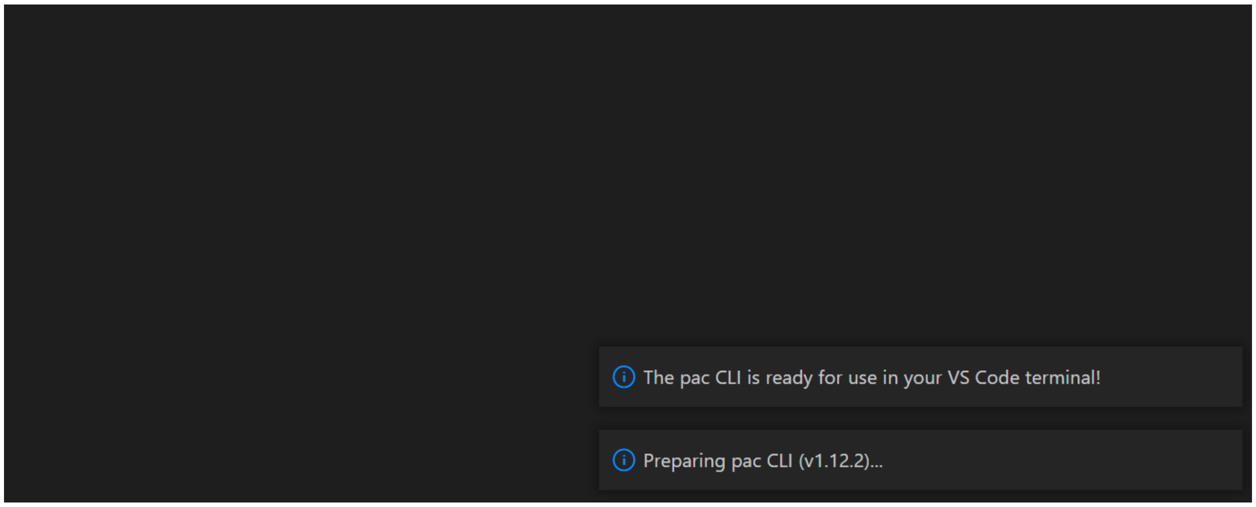
Name	Date modified	Type	Size
Today (1)			
microsoft-IsvExpTools.powerplatform-vscode-0.2.35 (1)	3/6/2022 4:31 PM	Microsoft Visual Stud...	22,866 KB

Once that is downloaded, launch Visual Studio Code on your workstation

1. Select the Extensions icon from the Activity bar
2. Then click on the ellipsis icon on the Extensions side bar
3. Select install from VSIX



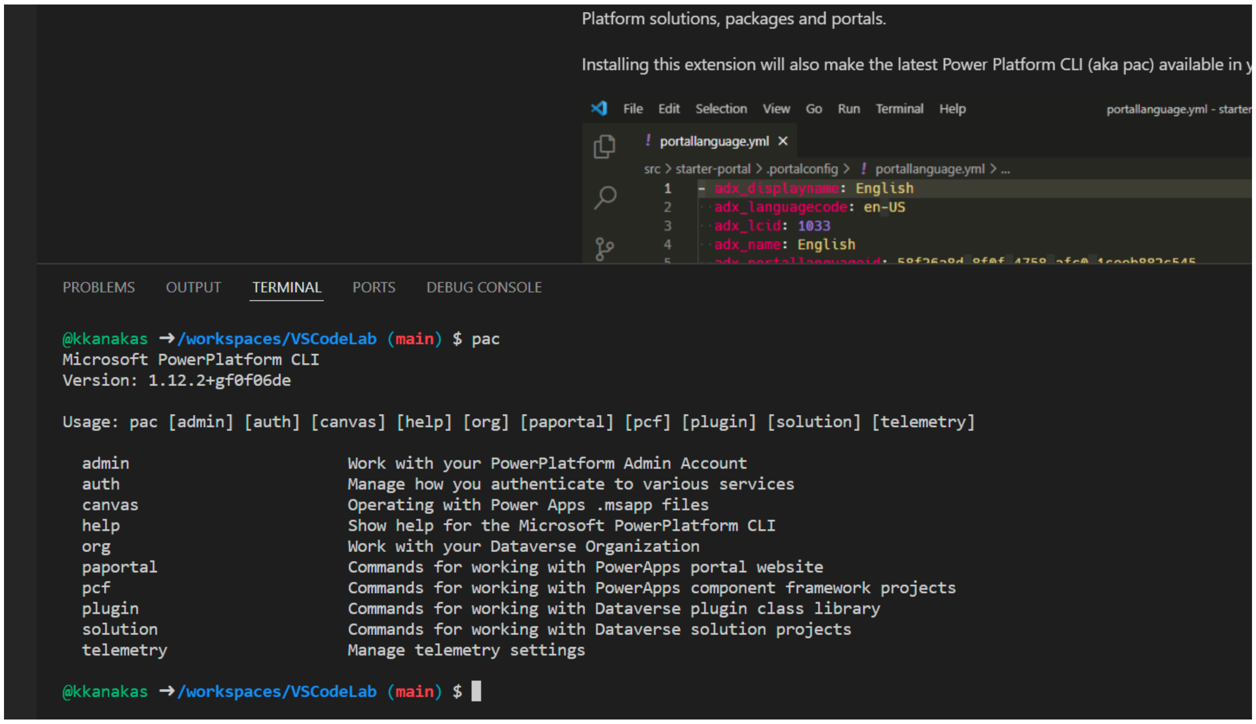
This will now install the Power Platform extension in your Visual Studio Code or GitHub Extension. On the bottom right hand corner, you will see the following message.



The pac CLI is ready for use in your VS Code terminal!

Preparing pac CLI (v1.12.2)...

Once the installation is done, launch the terminal interface from the Visual Studio Code interface



Platform solutions, packages and portals.

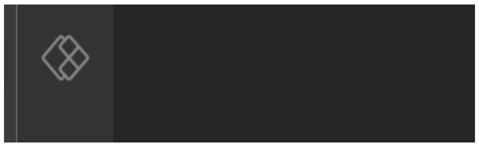
Installing this extension will also make the latest Power Platform CLI (aka pac) available in y

File Edit Selection View Go Run Terminal Help
portallanguage.yml - starter
portallanguage.yml
src > starter-portal > .portalconfig > portallanguage.yml > ...
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```yaml
- adx_displayname: English
  adx_languagecode: en-US
  adx_lcid: 1033
  adx_name: English
```

```
@kkanakas ➜/workspaces/VSCodeLab (main) $ pac
Microsoft PowerPlatform CLI
Version: 1.12.2+gf0f06de

Usage: pac [admin] [auth] [canvas] [help] [org] [paportal] [pcf] [plugin] [solution] [telemetry]

  admin                       Work with your PowerPlatform Admin Account
  auth                        Manage how you authenticate to various services
  canvas                      Operating with Power Apps .msapp files
  help                        Show help for the Microsoft PowerPlatform CLI
  org                         Work with your Dataverse Organization
  paportal                    Commands for working with PowerApps portal website
  pcf                         Commands for working with PowerApps component framework projects
  plugin                      Commands for working with Dataverse plugin class library
  solution                    Commands for working with Dataverse solution projects
  telemetry                   Manage telemetry settings

@kkanakas ➜/workspaces/VSCodeLab (main) $ 
```

and you will see the following icon show-up on the activity panel



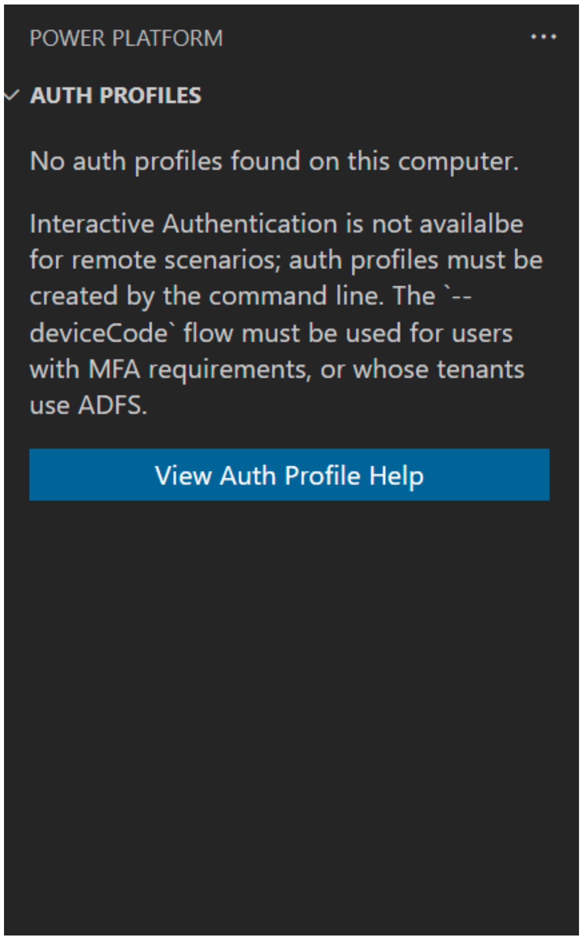
AUTHENTICATNG TO POWER PLATFORM AND LISTING ENVIRONMENTS AND SOLUTIONS

When you click on the Power platform icon on the activity bar, within the side bar, you will 2 panels, they are called

- Authentication Profiles Panel
- Environments and Solution Panel

The Authentication panel will only list the dataverse environments to which you are connected and based on that connection, it will populate the environment and solutions to which you have access to.

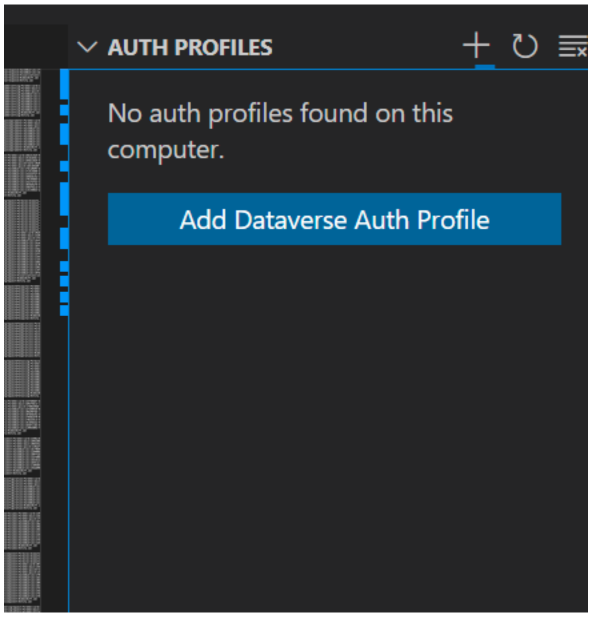
If you don't have any authentication profiles listed, then you will see the following:



No auth profiles found on this computer.

Interactive Authentication is not availalbe for remote scenarios; auth profiles must be created by the command line. The `--deviceCode` flow must be used for users with MFA requirements, or whose tenants use ADFS.

on Codespaces

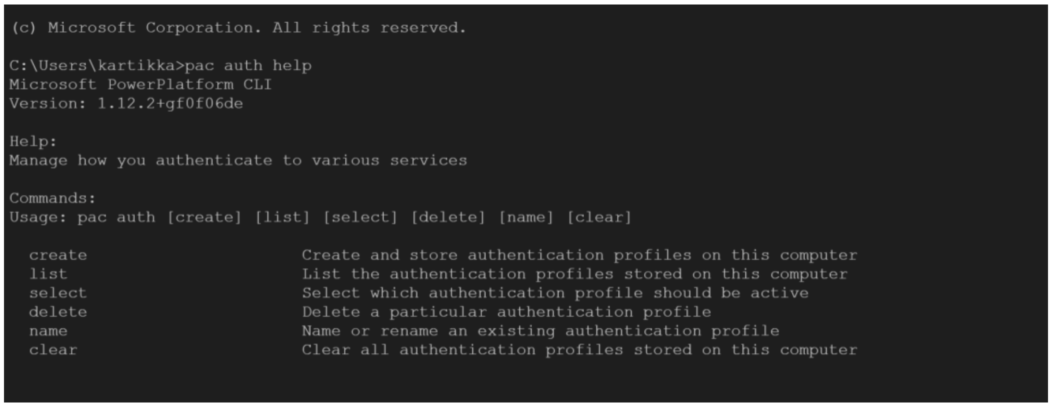

AUTH PROFILES
No auth profiles found on this computer.
Add Dataverse Auth Profile

on your workstation, if you click on the add Dataverse Auth Profile. It will launch the terminal window and show how to create the authentication profile using Power Platform CLI. The command in this case **"pac auth create"**

pac auth create allows to create two kinds of connection profiles

1) Administrator – used to create,list,backup, and remove environments
2) Dataverse – Authenticated to the relevant environment in order to execute solution operations.

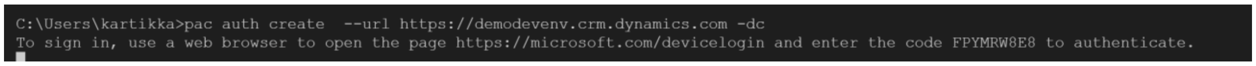
The Visual studio code extension, we focus on the latter.



```
(c) Microsoft Corporation. All rights reserved.

C:\Users\kartikka>pac auth help
Microsoft PowerPlatform CLI
Version: 1.12.2+gf0f06de

Help:
Manage how you authenticate to various services

Commands:
Usage: pac auth [create] [list] [select] [delete] [name] [clear]

  create                      Create and store authentication profiles on this computer
  list                        List the authentication profiles stored on this computer
  select                      Select which authentication profile should be active
  delete                      Delete a particular authentication profile
  name                        Name or rename an existing authentication profile
  clear                       Clear all authentication profiles stored on this computer
```

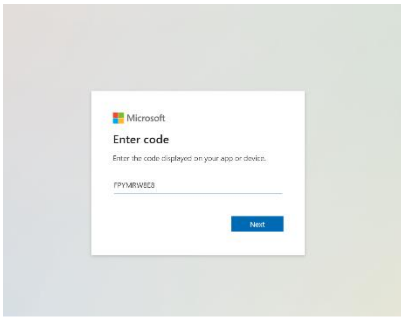
The command to create an authentication profile is **"pac auth create"** this command by default creates Dataverse authentication profile, unless you specify "**admin"** as part of the **--kind or -k** parameter.

You can specify username and password for the connection with the --username or --password, but these are optional. If you don't specify these parameters, the AAD authentication window pops up. You can also authenticate using Service Principal credentials, as in using your client id and secret. You also have the option to use the Device Code authentication flow, if you don't want to show your username and password, when you are doing this lab. In this example we are using the device code authentication

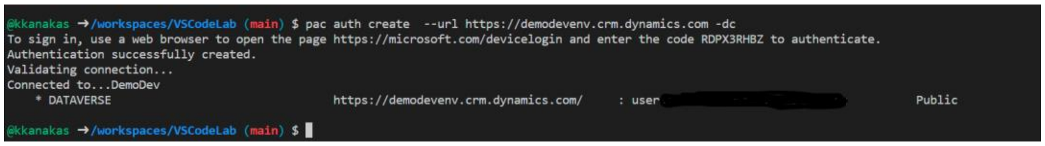


```
C:\Users\kartikka>pac auth create  --url https://demodevenv.crm.dynamics.com -dc
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code FPYMRW8E8 to authenticate.
```

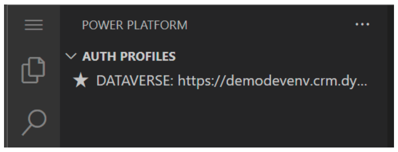
launch a browser session on your workstation, and go to the URL https://microsoft.com/devicelogin and provide the device code.

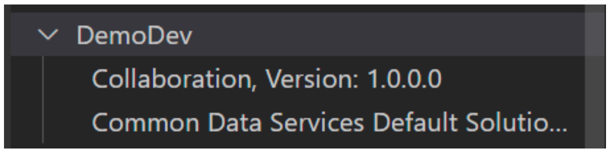


you then proceed to provide the name of the account you want to authenticate with Azure AD credentials and then auth profile token gets updated and the Auth profile side bar shows the authenticated profile with a star against it.



```
@kkanakas ➜/workspaces/VSCodeLab (main) $ pac auth create  --url https://demodevenv.crm.dynamics.com -dc
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code RDPX3RHBZ to authenticate.
Authentication successfully created.
Validating connection...
Connected to...DemoDev
    * DATAVERSE                         https://demodevenv.crm.dynamics.com/     : user[redacted]            Public

@kkanakas ➜/workspaces/VSCodeLab (main) $ 
```

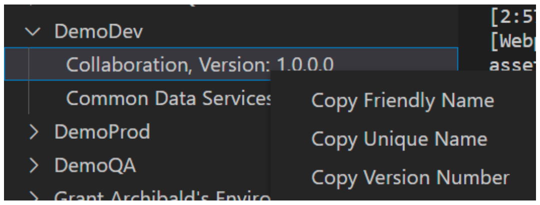



DemoDev
Collaboration, Version: 1.0.0.0
Common Data Services Default Solutio...

In the environments and solution section you will see the solutions that are available in the environment to which you are authenticated too.

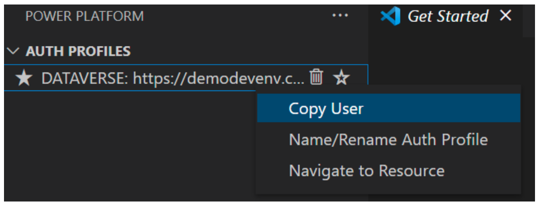
Now you have authenticated Power Platform your Visual Studio Code interface.

When you right-click on the solution you have the following options to select from:



- Copying the friendly name of the solution
- Copying the Unique Name(this is the important one, for command operations)
- Copy the version numbers of the solution

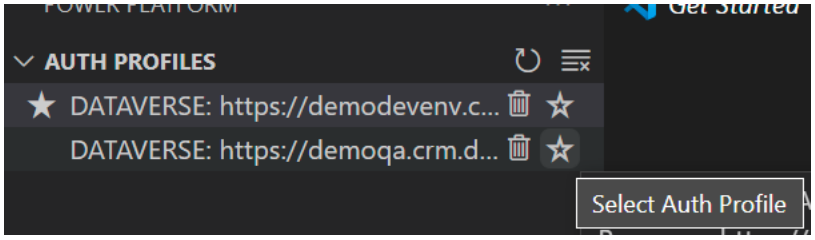
and you right-click on your auth profile you get the following options:



- Copying username used to authenticate to Power Platform
- Name/Rename the Auth profile to a friendly one
- Open a new tab and navigate to the environment url

Now, if you want to delete the profile, you can press the trash icon, and that will delete your authentication profile.

If you have multiple auth profiles, then you can select the start icon against the profile that you want to change to, and that will be the profile that will be in effect.

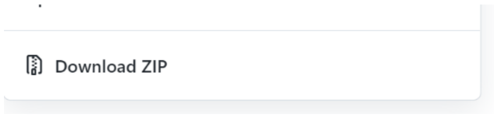

AUTH PROFILES
DATAVERSE: https://demodevenv.c...
DATAVERSE: https://demoqa.crm.d...
Select Auth Profile

This makes it easier to navigate across different auth profiles. To create a second auth profile, just repeat the same steps that you did to create the first one.

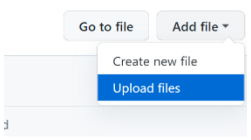
WORKING WITH THE POWER PLATFORM TOOLS EXTENSION FOR VISUAL STUDIO CODE

Now that we have installed the extension, authenticated to the environment, we are going to work with solutions using the Visual Studio Code user interface.

Download the zip files that you need for this project from the GitHub repository.

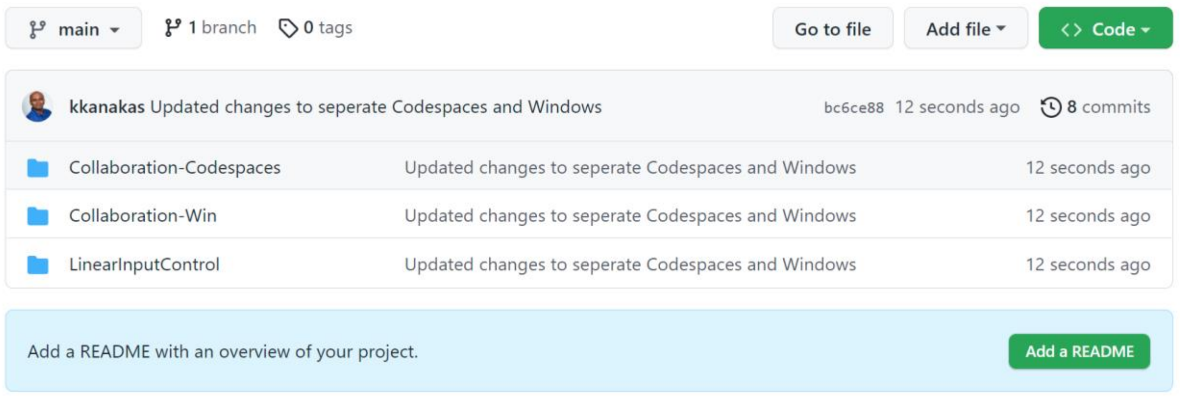


This zip file contains a solution and a PCF control that we are going to use for this project.



In your GitHub repository upload the extracted contents of the zip file. The ability to upload files is under the add files button. This will give you the option to drag files from your workstation and commit them to GitHub.

In your personal GitHub Repository should look like the following:



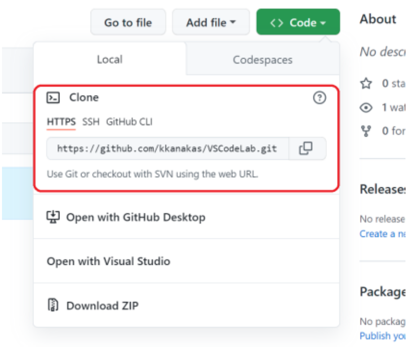
main | 1 branch | 0 tags | Go to file | Add file | Code

kkanakas Updated changes to seperate Codespaces and Windows — bc6ce88 12 seconds ago — 8 commits

Collaboration-Codespaces	Updated changes to seperate Codespaces and Windows	12 seconds ago
Collaboration-Win	Updated changes to seperate Codespaces and Windows	12 seconds ago
LinearInputControl	Updated changes to seperate Codespaces and Windows	12 seconds ago

Add a README with an overview of your project. Add a README

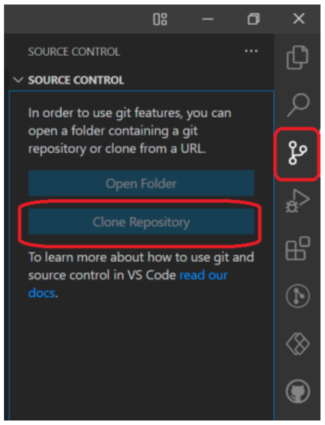
If you are using Visual Studio Code locally on your workstation, then proceed to clone your repository on to your workstation. Under the code button select clone and copy the link to your repository



```
https://github.com/kkanakas/VSCodeLab.git
```

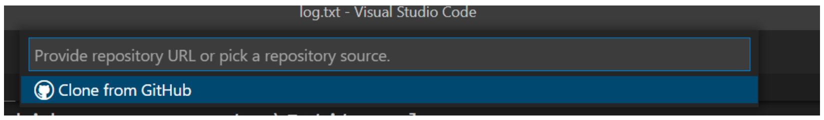
Use Git or checkout with SVN using the web URL.

Then on Visual Studio code, select git from the activity bar and then press the clone repository button in the side bar, then paste the URL in the command palette. If you have not authenticated to GitHub, then Visual Studio Code will prompt for you for your GitHub credentials and after providing that you will be able to clone you repository



In order to use git features, you can open a folder containing a git repository or clone from a URL.

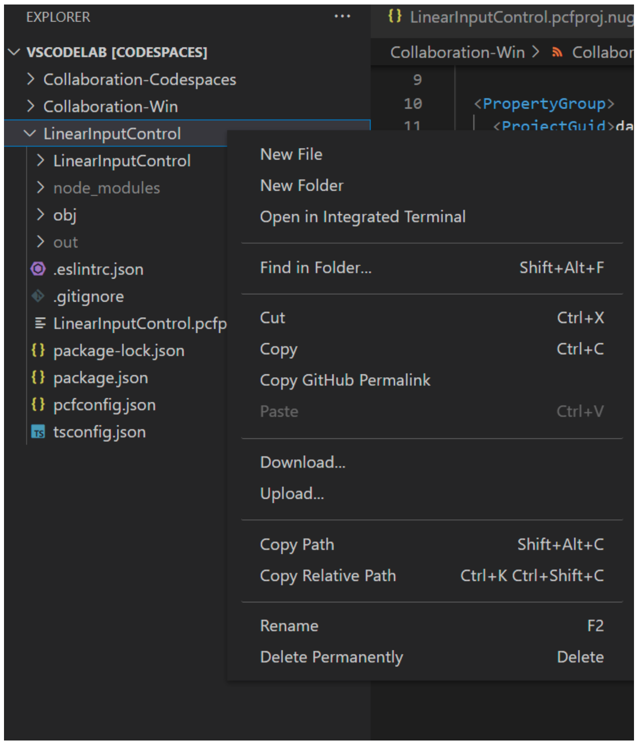
To learn more about how to use git and source control in VS Code read our docs.


log.txt - Visual Studio Code
Provide repository URL or pick a repository source.
Clone from GitHub

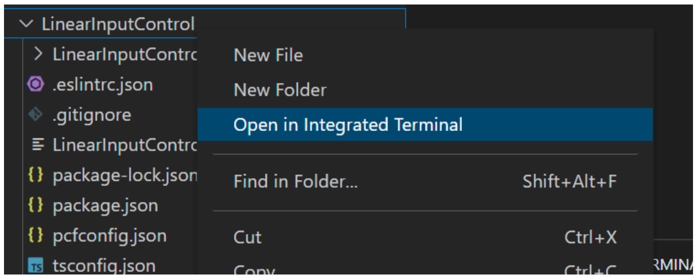
Now that your repository has been successfully cloned. Let us start working with the solutions and controls. **Note: if you are using GitHub codespaces you do not have authenticate to the GitHub as you are already authenticated. All you must do is launch CodeSpaces from the code button.**

WORKING WITH CONTROLS AND SOLUTIONS

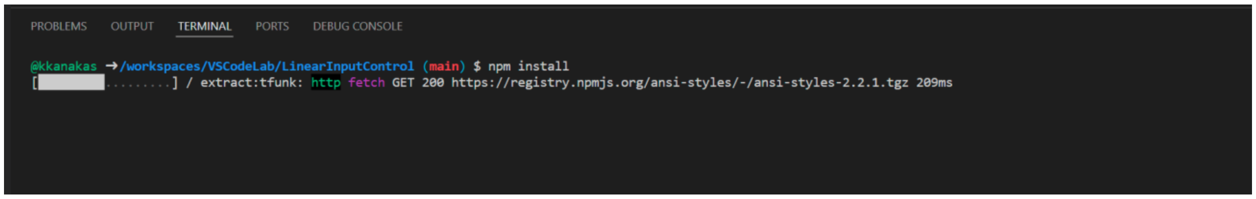
Now that we are in our Visual Studio code environment, Select the explorer icon from the activity bar, and your side bar should look like the following, with the folders completely expanded:



then right click on the linear control folder and select open in integrated terminal



In the terminal window, type the following:

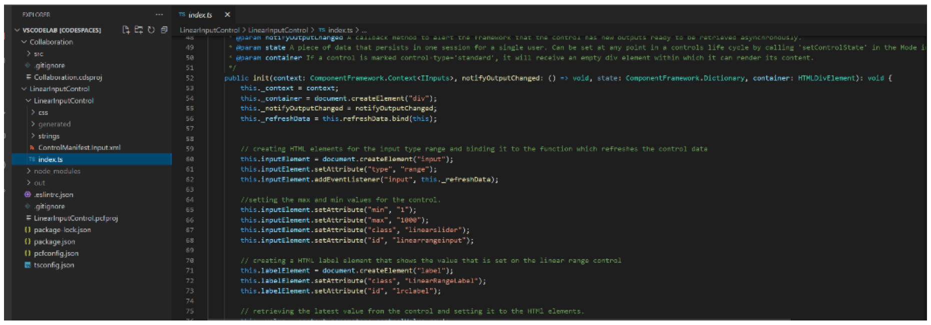


```
@kkanakas →/workspaces/VSCodeLab/LinearInputControl (main) $ npm install
[████████.........] / extract:tfunk: http fetch GET 200 https://registry.npmjs.org/ansi-styles/-/ansi-styles-2.2.1.tgz 209ms
```

This will install the appropriate node libraries needed for your PCF control. If there are things that need to be fixed, please run **"npm audit fix"** to fix any vulnerabilities.

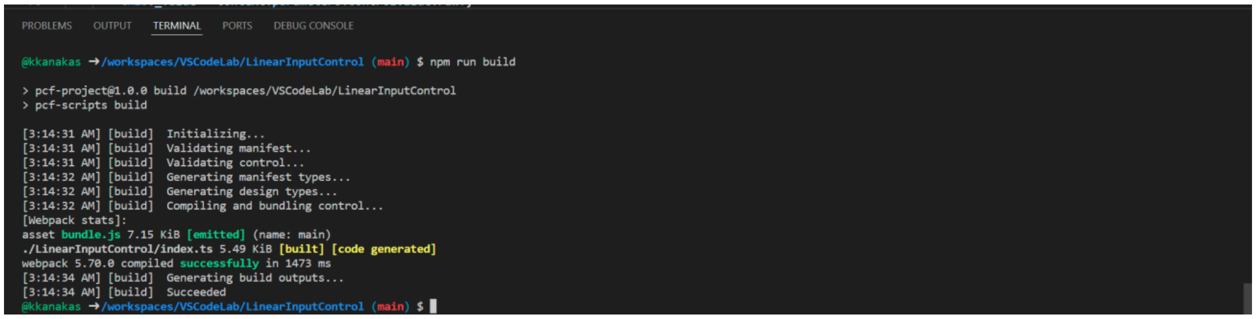
Once you have done those operations, now let us look at the control, and then proceed to build it.

Select the explorer icon and expand the LinerInputControl folder which contained within the LinearInputControl folder



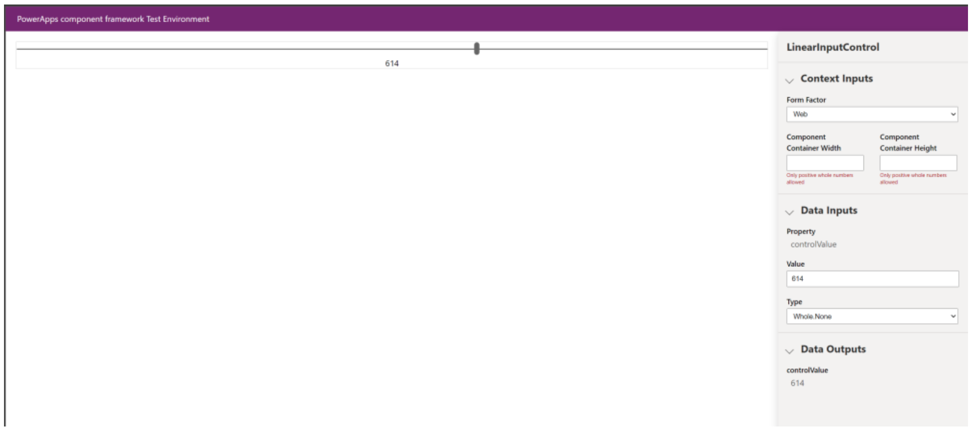
```
 * @param state A piece of data that persists in one session for a single user. Can be set at any point in a controls life cycle by calling 'setControlState' in the Mode i
 * @param container If a control is marked control-type='standard', it will receive an empty div element within which it can render its content.
 */
public init(context: ComponentFramework.Context<IInputs>, notifyOutputChanged: () => void, state: ComponentFramework.Dictionary, container: HTMLDivElement): void {
    this._context = context;
    this._container = document.createElement("div");
    this._notifyOutputChanged = notifyOutputChanged;
    this._refreshData = this.refreshData.bind(this);


    // creating HTML elements for the input type range and binding it to the function which refreshes the control data
    this.inputElement = document.createElement("input");
    this.inputElement.setAttribute("type", "range");
    this.inputElement.addEventListener("input", this._refreshData);

    //setting the max and min values for the control.
    this.inputElement.setAttribute("min", "1");
    this.inputElement.setAttribute("max", "1000");
    this.inputElement.setAttribute("class", "linearslider");
    this.inputElement.setAttribute("id", "linearrangeinput");

    // creating a HTML label element that shows the value that is set on the linear range control
    this.labelElement = document.createElement("label");
    this.labelElement.setAttribute("class", "LinearRangeLabel");
    this.labelElement.setAttribute("id", "lrclabel");

    // retrieving the latest value from the control and setting it to the HTMl elements.
```

Select the index.ts file and you can see that this is a slider control. Now when we proceed to build it, it will take these typescript files and create the appropriate javascript for the web control. So now, in the terminal window type **npm run build** and you should see the following:



```
@kkanakas →/workspaces/VSCodeLab/LinearInputControl (main) $ npm run build

> pcf-project@1.0.0 build /workspaces/VSCodeLab/LinearInputControl
> pcf-scripts build

[3:14:31 AM] [build]  Initializing...
[3:14:31 AM] [build]  Validating manifest...
[3:14:31 AM] [build]  Validating control...
[3:14:32 AM] [build]  Generating manifest types...
[3:14:32 AM] [build]  Generating design types...
[3:14:32 AM] [build]  Compiling and bundling control...
[Webpack stats]:
asset bundle.js 7.15 KiB [emitted] (name: main)
./LinearInputControl/index.ts 5.49 KiB [built] [code generated]
webpack 5.70.0 compiled successfully in 1473 ms
[3:14:34 AM] [build]  Generating build outputs...
[3:14:34 AM] [build]  Succeeded
@kkanakas →/workspaces/VSCodeLab/LinearInputControl (main) $
```

And then we are going to test it, by using the command **npm run start**. This launch this web control in a browser under component framework testing environment. You slide the control back and forth and see the values change in the Data Input section.



Form Factor: Web

Component Container Width:
Only positive whole numbers allowed

Component Container Height:
Only positive whole numbers allowed

Property: controlValue

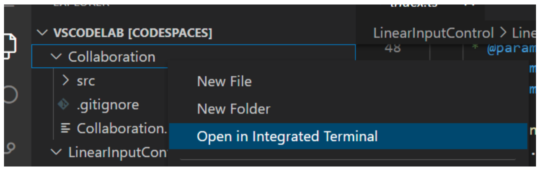
Value: 614

Type: Whole.None

controlValue: 614

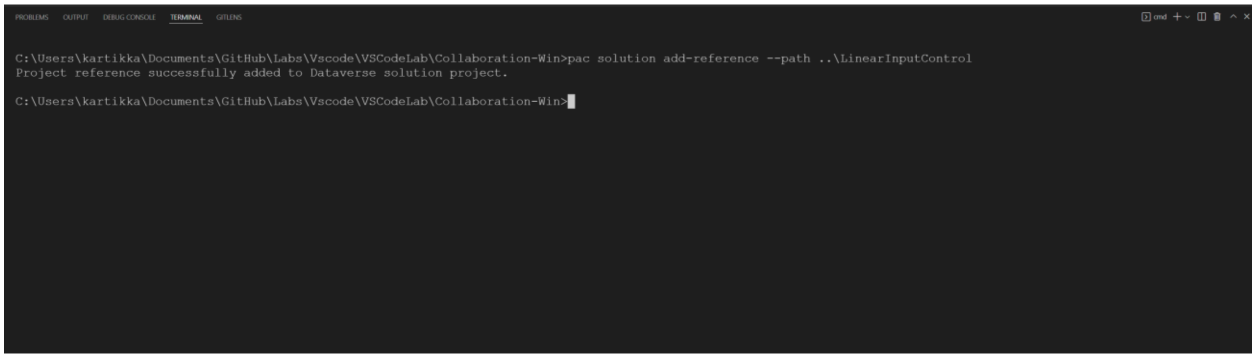
Now, we know our control works properly, let go ahead add it to our solution.

On the Explorer window, select the collaboration folder and right-click and select open in integrated terminal



NOTE: Ignore this step if you are using GitHub CodeSpaces, do not add the pcf control to the solution only windows users can do this step.

In the integrated terminal, type the **"pac solution add-reference --path <path to the pcf control>"**

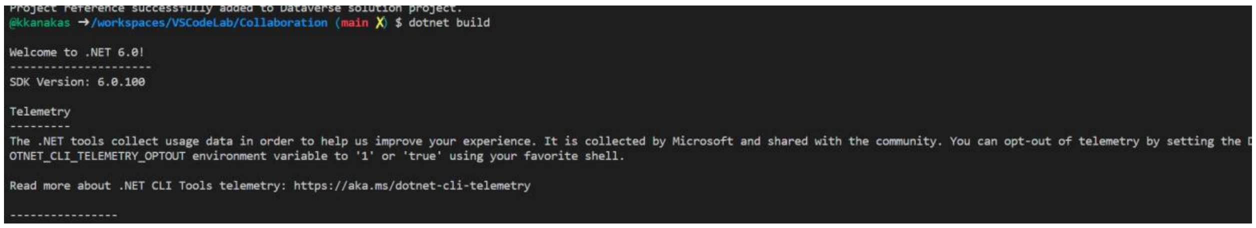


```
C:\Users\kartikka\Documents\GitHub\Labs\Vscode\VSCodeLab\Collaboration-Win>pac solution add-reference --path ..\LinearInputControl
Project reference successfully added to Dataverse solution project.

C:\Users\kartikka\Documents\GitHub\Labs\Vscode\VSCodeLab\Collaboration-Win>
```

this will add the pcf control to the solution

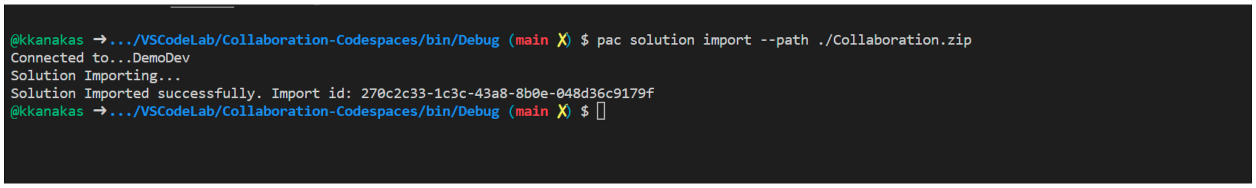
NOTE: for Codespaces folks start from here to continue.

Now proceed to build the solution by typing "**dotnet build**", this will start building the zip file needed to import into your Power Platform environment.



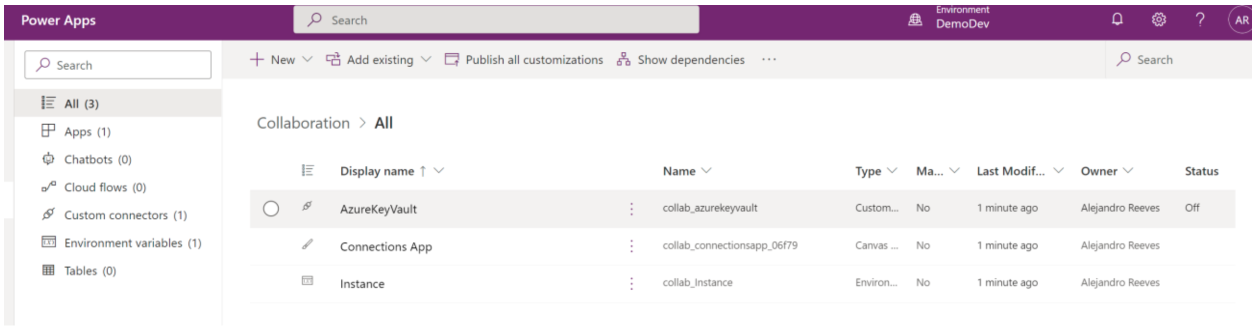
```
Project reference successfully added to Dataverse solution project.
@kkanakas →/workspaces/VSCodeLab/Collaboration (main ✗) $ dotnet build

Welcome to .NET 6.0!
---------------------
SDK Version: 6.0.100

Telemetry
---------
The .NET tools collect usage data in order to help us improve your experience. It is collected by Microsoft and shared with the community. You can opt-out of telemetry by setting the D
OTNET_CLI_TELEMETRY_OPTOUT environment variable to '1' or 'true' using your favorite shell.

Read more about .NET CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

----------------
```

Once the project is built navigate to the bin\Debug folder under the Collaboration folder and run the solution import command back into our dev environment.



```
@kkanakas →.../VSCodeLab/Collaboration-Codespaces/bin/Debug (main ✗) $ pac solution import --path ./Collaboration.zip
Connected to...DemoDev
Solution Importing...
Solution Imported successfully. Import id: 270c2c33-1c3c-43a8-8b0e-048d36c9179f
@kkanakas →.../VSCodeLab/Collaboration-Codespaces/bin/Debug (main ✗) $ 
```

And you confirm if your solution updated in your environment using the Power Platform Web UI



Power Apps | Environment DemoDev

Collaboration > All

Display name	Name	Type	Ma...	Last Modif...	Owner	Status
AzureKeyVault	collab_azurekeyvault	Custom...	No	1 minute ago	Alejandro Reeves	Off
Connections App	collab_connectionsapp_06f79	Canvas ...	No	1 minute ago	Alejandro Reeves	
Instance	collab_Instance	Environ...	No	1 minute ago	Alejandro Reeves	

You have now successfully used the Power Platform Visual Studio Code Extension.