Microsoft Dynamics 365 Customer Engagement

DevOps 1 Day with Lab

Lab 1: Setup Azure Repo and Make Your First Commit

Student Lab Manual

Conditions and Terms of Use

Copyright and Trademarks

Table of Contents

Introduction

In this lab, we will commit our first unpacked Dynamics 365 Customer Engagement solution using Solution Packager.

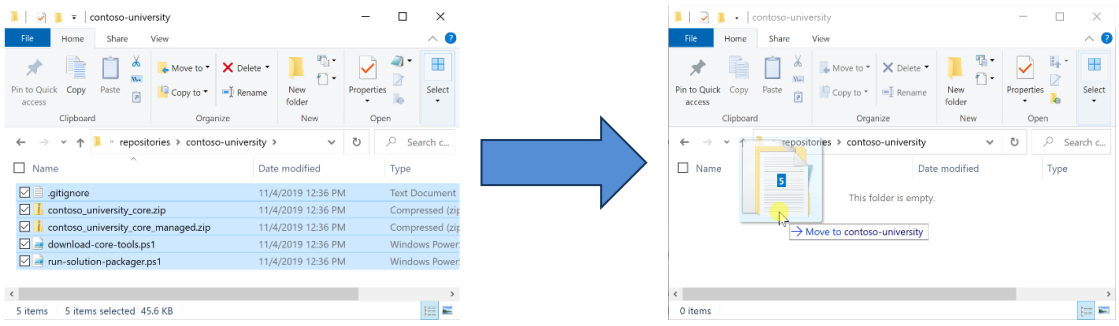
Check-in your first Dynamics 365 Customer Engagement Solution

If you have not already done so, please follow the workshop prerequisites lab to setup your Azure Repo and clone the repository locally to continue.

Note: This lab assumes that you have cloned the Contoso University repository into `C:\Users\<Current User>\Documents\repositories\contoso-university.`

Copy Initial Lab Resources

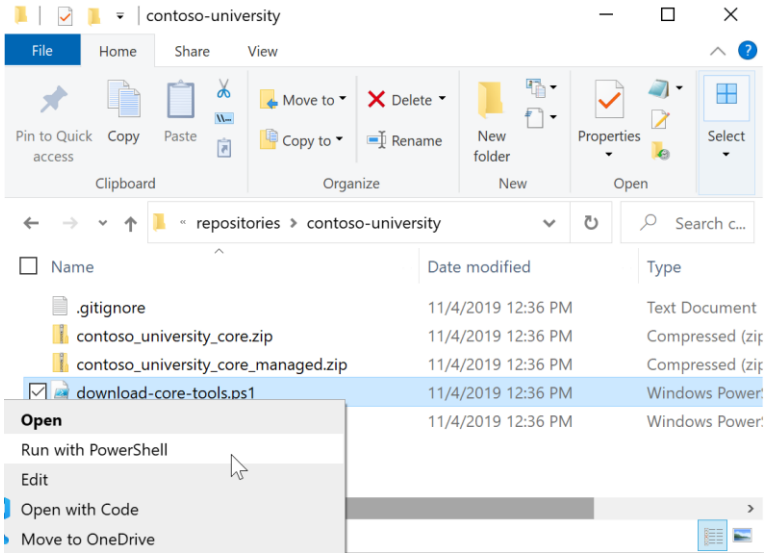
1. Open file explorer (Shortcut keys: Win + E).
2. Navigate to `C:\Users\<Current User>\Documents\repositories\contoso-university.`
3. Open another instance of file explorer.
4. Navigate to the `lab-resources/lab-1/copy-to-repo` folder in the provided supporting materials.



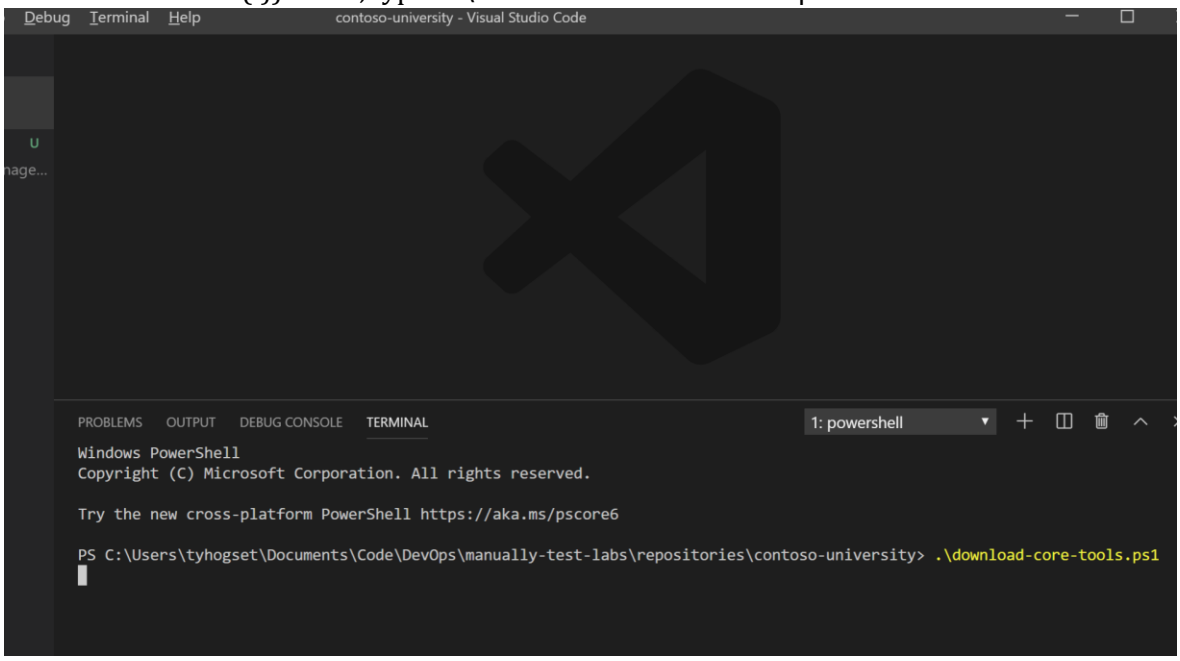
5. Select all the contents from the `copy-to-repo` folder and drag them into the `contoso-university` folder.

Download core tools

1. From your local `contoso-university` repository, right-click on `download-core-tools.ps1` and click "Run with PowerShell."



Alternately, from VS Code, where your repository should be opened, open the terminal (press Control + Backtick (`)). Then, type `.\download-core-tools.ps1` and hit Enter.

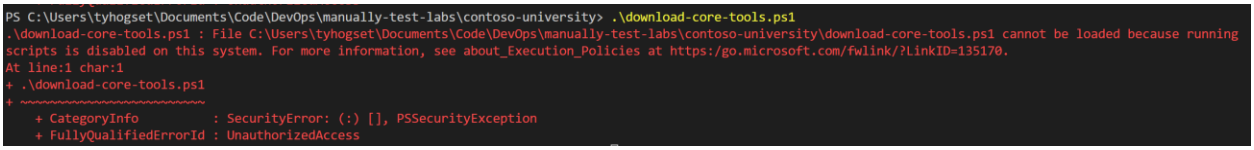


2. The PowerShell script will run, creating a `tools` directory in the `contoso-university` folder.

Note: This script was pulled directly from the Download tools from Nuget page in the PowerApps documentation.

Restricted Execution Policy

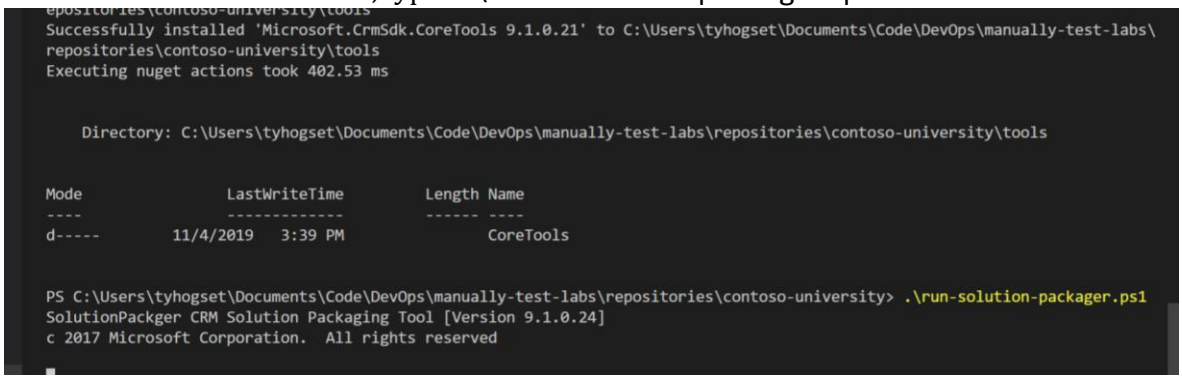
Depending on your system's policies, you may be unable to run ps1 scripts. Running the steps above, you will get an error like this:



If that's the case, you can copy the contents of copy-to-repo-restricted into your repository. This contains a copy of the tools folder that will be downloaded in this step, along with batch file, which you should be able to run without issue.

Unpack solution files

1. From the terminal in VS Code, type `.\run-solution-packager.ps1` and hit Enter.



   Note: If you are not able to run ps1 scripts on your machine, type `.\run-solution-packager.bat` instead.

2. This script (shown below) will take the two `.zip` files in the repository folder and distribute their contents into a newly created `contents` folder.

```
.\tools\CoreTools\SolutionPackager.exe `
    /action Extract `
    /zipfile .\contoso_university_core.zip `
    /folder .\contents `
    /packagetype Both
```

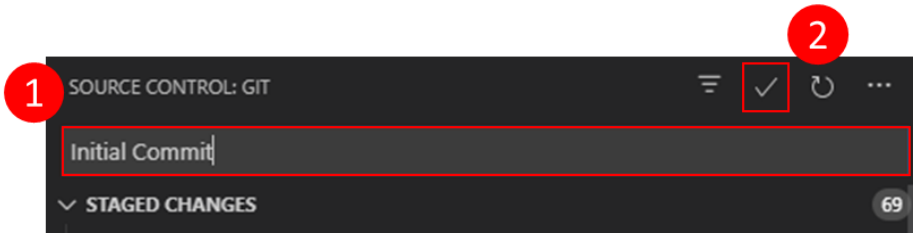
   Here, we've selected a `packagetype` of `Both`, so that we can later deploy this solution into either a development (unmanaged) or downstream (managed) environment.

   For information on SolutionPackager, visit the SolutionPackager tool page in the PowerApps documentation.

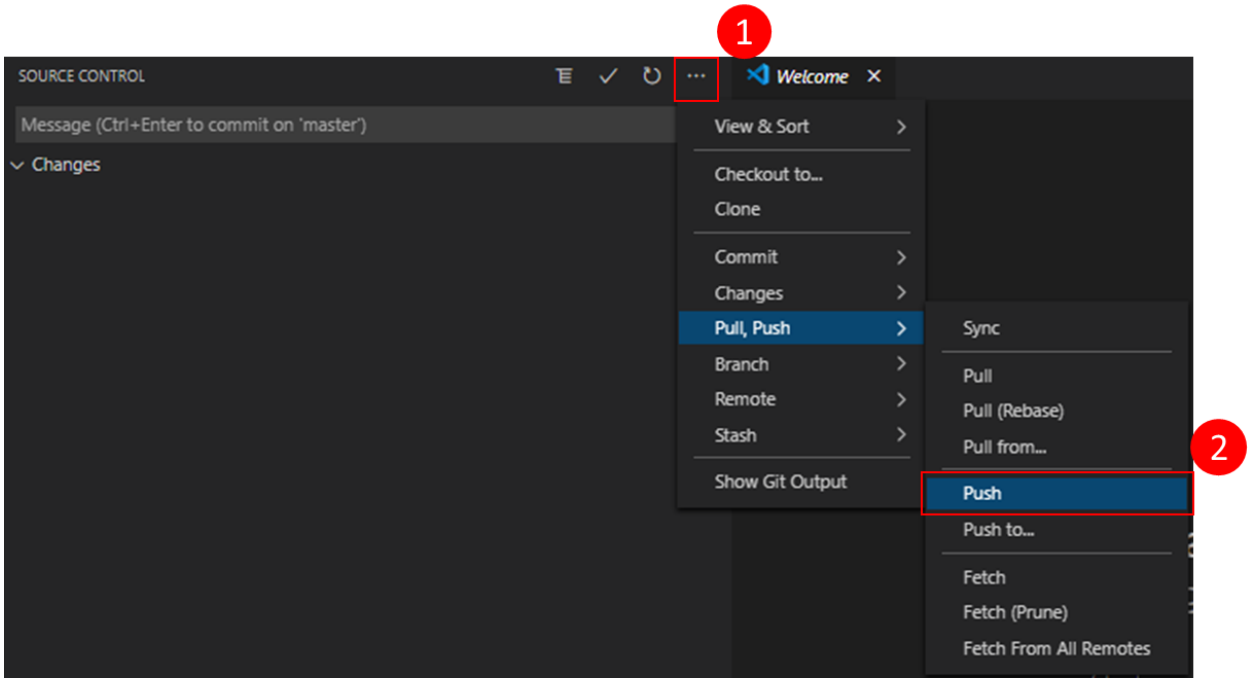
Check unpacked solution(s) into repository using VS Code

1. Open Visual Studio Code (VS Code).
2. Click the source control button  (Shortcut Ctrl+Shift+G).
3. Click the + symbol to stage all changes. If the + symbol is not visible, hover over changes, then click the button.

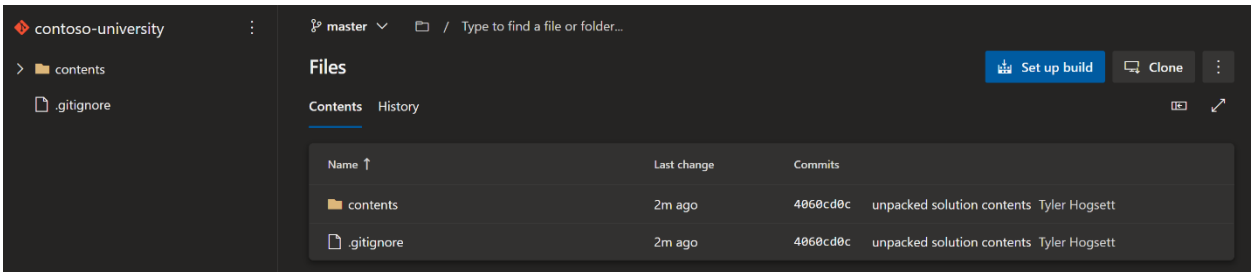
4. Enter a commit message into the **Message** box, then click the check mark to commit the changes.



5. Click the ellipsis (…), click **Pull, Push**, then **Push** to send the committed changes to your Azure DevOps repository.



After you have pushed your changes, if you open your project in Azure DevOps and navigate to your Repo, you should see your changes reflected there.



Congratulations! You have now committed your first unpacked solution into version control and pushed those changes to Azure Repos for safe keeping. You can use later in our build and release pipelines.