Microsoft Dynamics 365

Power Platform GitHub Actions

Dynamics 365 FastTrack
Architecture Insights Series

Kieran Petrie
Senior FastTrack Solution Architect

Content

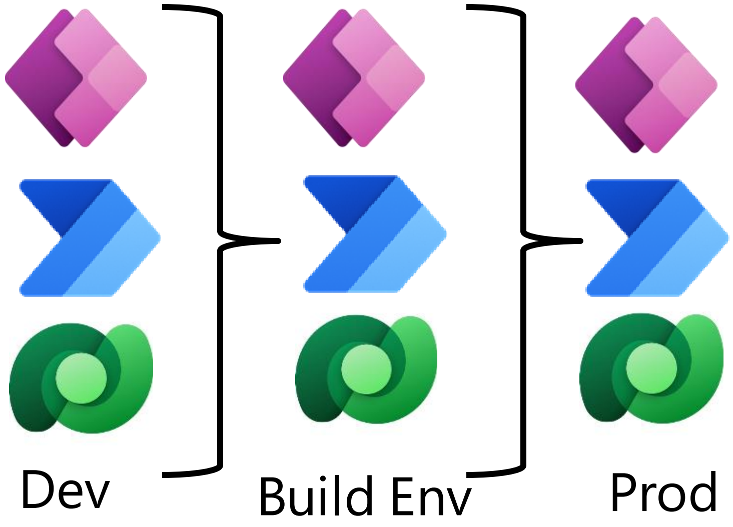
- Power Platform ALM
- Overview of GitHub Actions
- Repository, Branches, Pull Requests, Releases and Workflows
- Configuration & End to End Deployment
- References

Power Platform ALM

Power Platform ALM Basics

Environments

Multiple environments should be used to dev, build and promote to production



Dev

Build Env

Prod

Solutions

Solutions are used to transport apps and components from one environment to another, or to apply a set of customizations to existing apps.

Unmanaged

- Used in Dev Environments
- Can be exported
- Doesn’t remove customizations when deleted

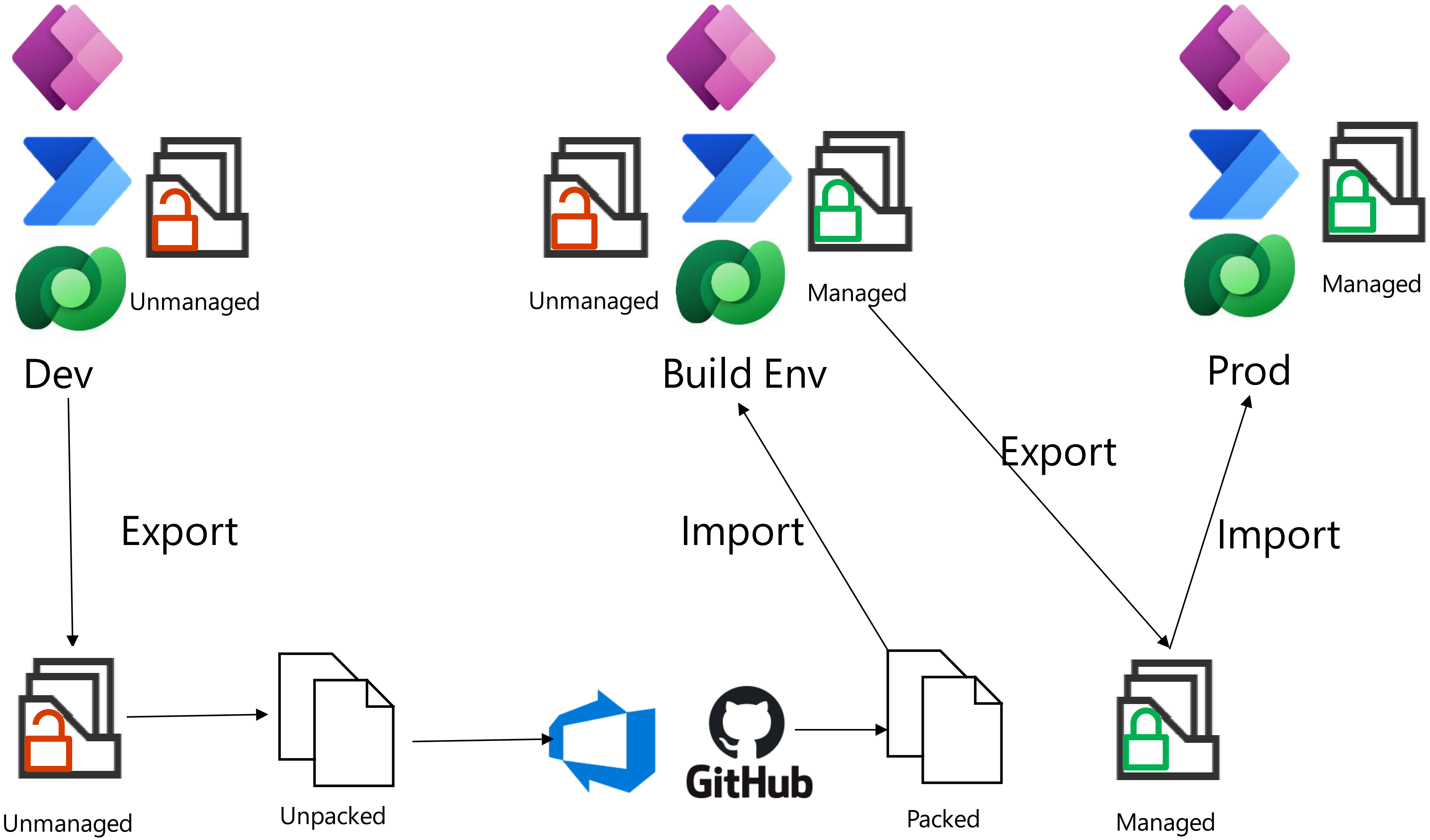
Managed

- Used to deploy to non Dev environments
- Can not edit components in the environment
- Can not be exported
- Deletes customizations when uninstalled

Source Control

A source control system helps organizations achieve healthy ALM because the assets maintained in the source control system are the "single source of truth"—or, in other words, the single point of access and modification for your solutions

Healthy ALM


Unmanaged
Dev
Unmanaged
Managed
Build Env
Managed
Prod
Export
Import
Export
Import
Unmanaged
Unpacked
GitHub
Packed
Managed

Overview of GitHub Actions

Why GitHub Actions?

GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows you to automate your build, test, and deployment pipeline. You can create workflows that build and test every pull request to your repository, or deploy merged pull requests to production.

Automate any workflow
directly from GitHub, with automation for any workflow

Standardize and scale best practices
across your organization

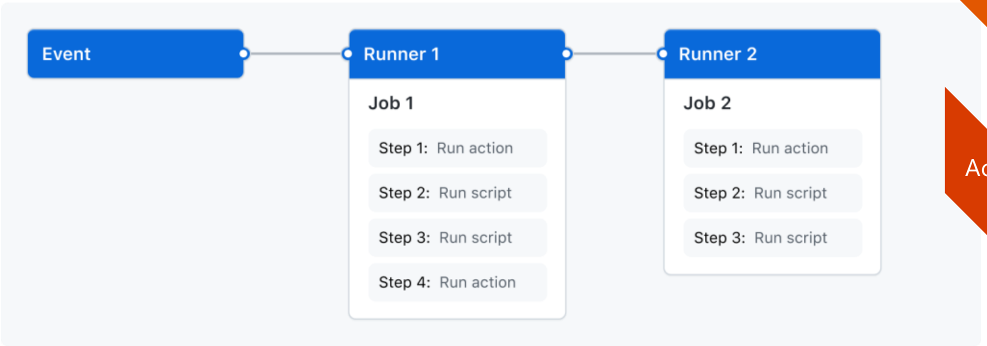
Community standard for automation
Innovate faster with your entire community

Components of GitHub Actions

You can configure a GitHub Actions workflow to be triggered when an event occurs in your repository, such as a pull request being opened or an issue being created.

Your workflow contains one or more jobs which can run in sequential order or in parallel.

Each job will run inside its own virtual machine runner, or inside a container, and has one or more steps that either run a script that you define or run an action.



Event

Runner 1

Job 1

Step 1: Run action

Step 2: Run script

Step 3: Run script

Step 4: Run action

Runner 2

Job 2

Step 1: Run action

Step 2: Run script

Step 3: Run script

Event

- An event is a specific activity in a repository that triggers a workflow run.

Workflow

- A workflow is a configurable automated process that will run one or more jobs.

Job

- A job is a set of *steps* in a workflow that execute on the same runner.

Runner

- A runner is a server that runs your workflows when they're triggered.

Action

- An *action* is a custom application for the GitHub Actions platform that performs a complex but frequently repeated task.

Repository, Branches, Pull Requests, Releases and Workflows

GitHub Repositories

A GitHub Repository is a location on the GitHub platform that will store all of your project's files and each files revision history.

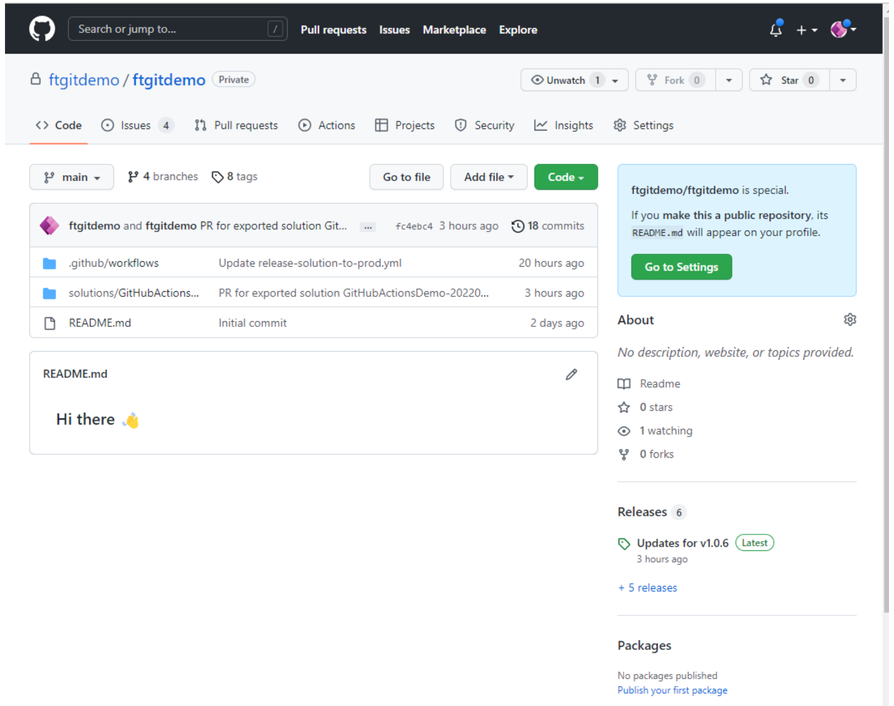
Repositories can be:-

User-owned – You would give people collaborator access so the can collaborate on your project

Organization-owned – Organization members can be given access permissions to collaborate on the repository

Repositories can be used to manage your work and collaborate with others:-

- **Issues** - can collect user feedback, report bugs and organize tasks to be accomplished
- **Discussions** – Ask and answer questions, share information, make announcements and have conversations about a project
- **Pull Requests** – Are used to propose changes to a repository
- **Project Boards** – organize and prioritize your issues and pull requests



ftgitdemo / ftgitdemo Private

ftgitdemo and ftgitdemo PR for exported solution Git... fc4ebc4 3 hours ago 18 commits

.github/workflows	Update release-solution-to-prod.yml	20 hours ago
solutions/GitHubActions...	PR for exported solution GitHubActionsDemo-20220...	3 hours ago
README.md	Initial commit	2 days ago

README.md

Hi there 👋

ftgitdemo/ftgitdemo is special.

If you **make this a public repository**, its `README.md` will appear on your profile.

About

No description, website, or topics provided.

Readme
0 stars
1 watching
0 forks

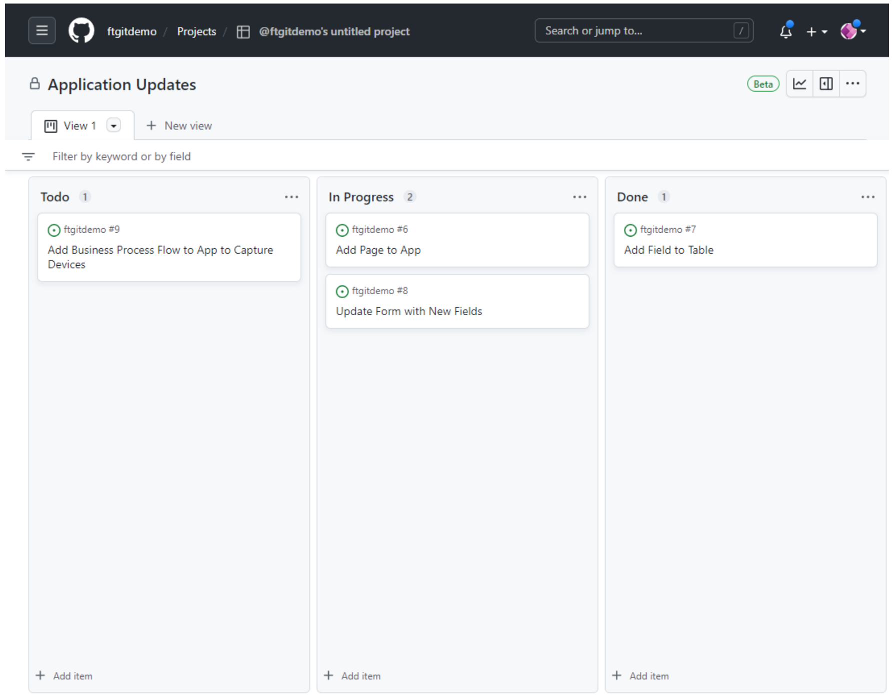
Releases 6

Updates for v1.0.6 Latest
3 hours ago

+ 5 releases

Packages

No packages published
Publish your first package



ftgitdemo / Projects / @ftgitdemo's untitled project
Search or jump to...
Beta
View 1
New view
Filter by keyword or by field
Add item
Add item
Add item

Application Updates

Todo 1
- ftgitdemo #9
  Add Business Process Flow to App to Capture Devices

In Progress 2
- ftgitdemo #6
  Add Page to App
- ftgitdemo #8
  Update Form with New Fields

Done 1
- ftgitdemo #7
  Add Field to Table

Branches

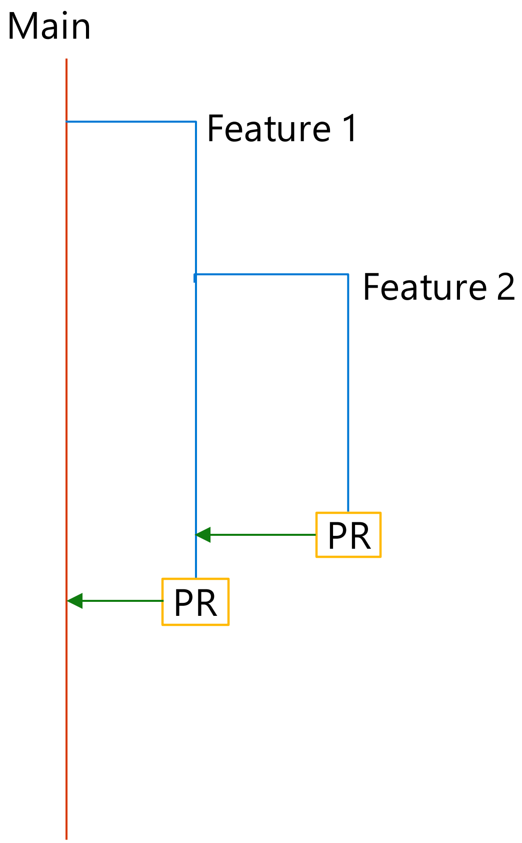
Branches are used to isolate development work without affecting other branches within the repository.

Each repository has one default branch (main) and can have multiple other branches.

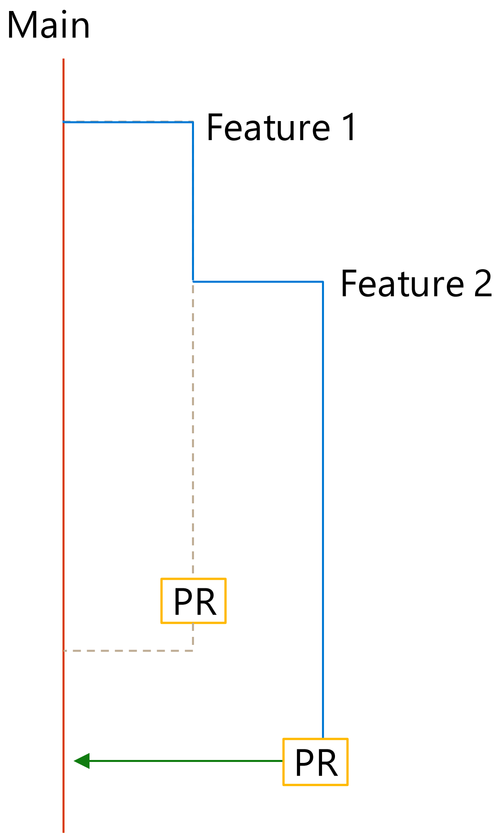
You merge the changes made in one branch into another using a pull request. A branch (head) is always created from an existing branch (base) and typically eventually merged back into the main branch once the features have been completed.

The branching strategy used should be considered carefully to ensure that all projects working on a repository can safely work on and merge changes as required. This simple example below illustrates branching to work on two features in a repository and merging them back into the main branch once complete..

Branches are created from Main when the developer(s) wish to build new features. Once they are happy with their changes, they will create pull requests



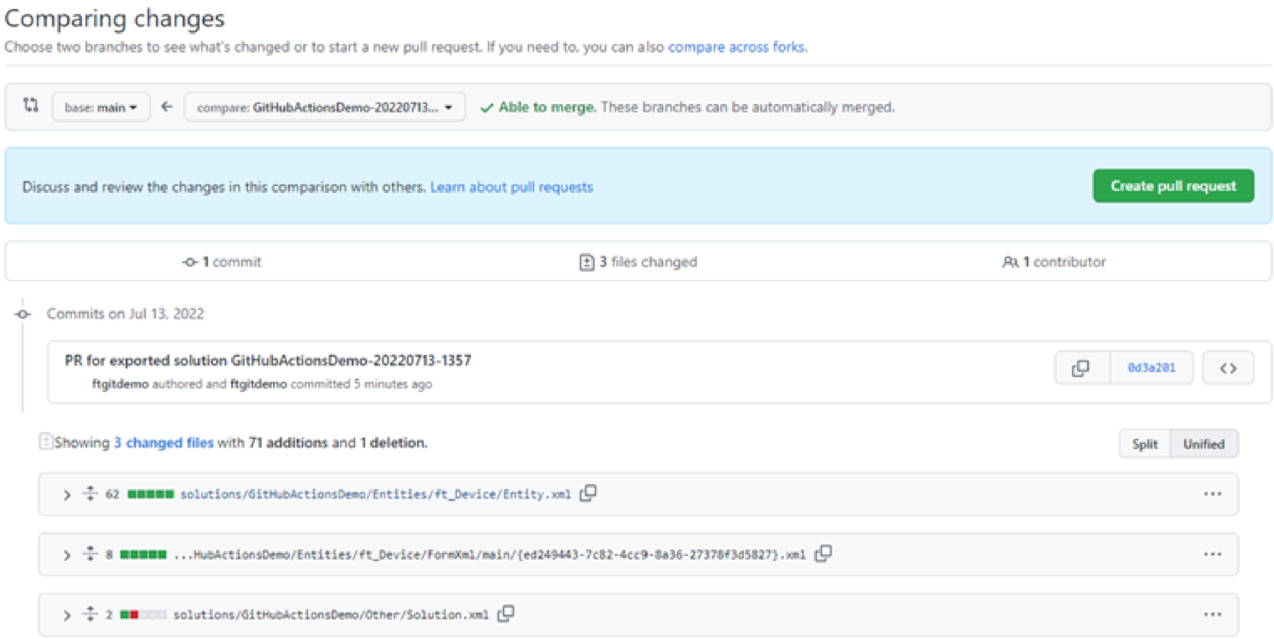
When a pull request is approved the changes are merged into the main branch and the branch can then be deleted. Any further branches are retargeted, and the second pull request may have to deal with merging conflicts



Pull Requests

Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators. Each pull requests has their own discussion forum and must be accepted before the changes can be merged into the base branch.

Contributors can discuss the Pull Request and decide if the changes should be merged with the base branch. By default, any pull request can be merged at any time, unless the head branch conflicts with the base branch. The default “Merge pull request” will take all commits from the feature branch (head) and add them to the base branch in a merge commit. A “Squash and merge” takes all the Pull Requests commits and squashes them into a single commit so instead of seeing all of individual commits from a head branch the commits are combined into one and merged into the base branch.



Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base: main ← compare: GitHubActionsDemo-20220713... ✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. Learn about pull requests

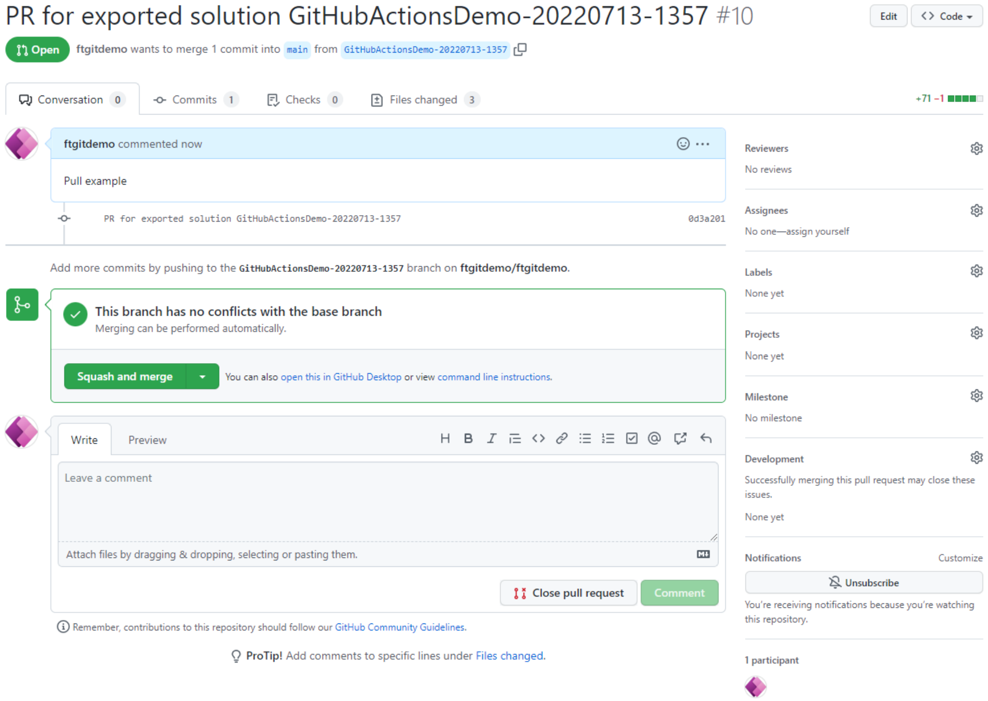
1 commit | 3 files changed | 1 contributor

Commits on Jul 13, 2022

PR for exported solution GitHubActionsDemo-20220713-1357
ftgitdemo authored and ftgitdemo committed 5 minutes ago — 0d3a201

Showing 3 changed files with 71 additions and 1 deletion.

```
62  solutions/GitHubActionsDemo/Entities/ft_Device/Entity.xml
8   ...HubActionsDemo/Entities/ft_Device/FormXml/main/{ed249443-7c82-4cc9-8a36-27378f3d5827}.xml
2   solutions/GitHubActionsDemo/Other/Solution.xml
```



PR for exported solution GitHubActionsDemo-20220713-1357 #10

Open — ftgitdemo wants to merge 1 commit into main from GitHubActionsDemo-20220713-1357

Conversation 0 | Commits 1 | Checks 0 | Files changed 3 | +71 −1

ftgitdemo commented now

Pull example

PR for exported solution GitHubActionsDemo-20220713-1357 — 0d3a201

Add more commits by pushing to the GitHubActionsDemo-20220713-1357 branch on ftgitdemo/ftgitdemo.

This branch has no conflicts with the base branch
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Remember, contributions to this repository should follow our GitHub Community Guidelines.

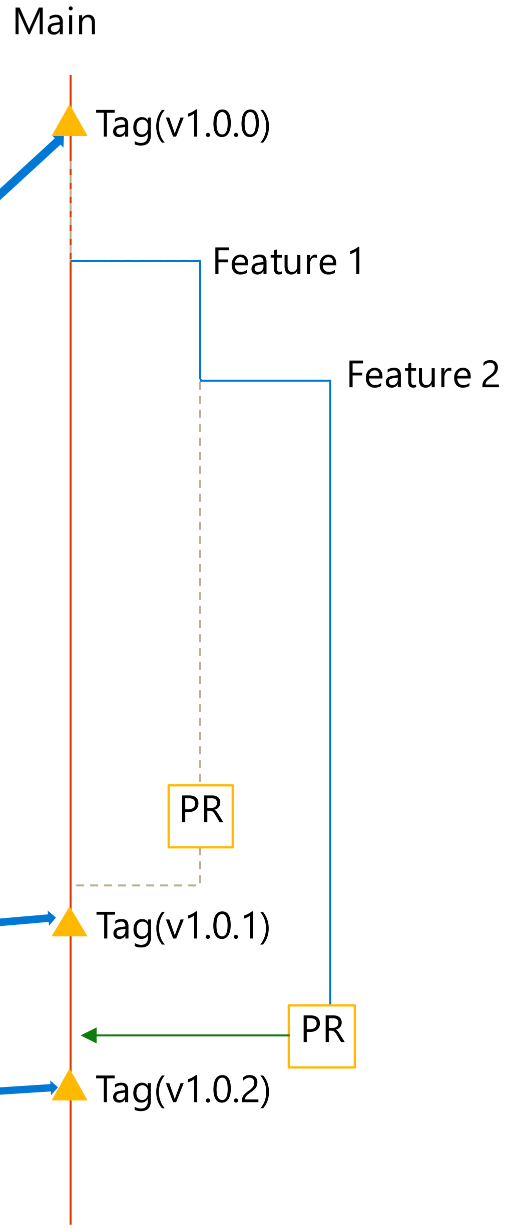
ProTip! Add comments to specific lines under Files changed.

Reviewers: No reviews
Assignees: No one—assign yourself
Labels: None yet
Projects: None yet
Milestone: No milestone
Development: Successfully merging this pull request may close these issues. None yet
Notifications: You're receiving notifications because you're watching this repository.
1 participant

Releases

Releases are deployable software iterations that can be packaged and made available to be used by others. Releases are based on Tags which mark a specific point in your repository's history, a tag date may be different than a release date since they can be created at different times. This means once a repository has been tagged a release can contain the version of the files at the point in time the tag was created. This allows versions of the software package to be recreated at anytime using the correct tag which will ensure the correct files and versions of the files are included.

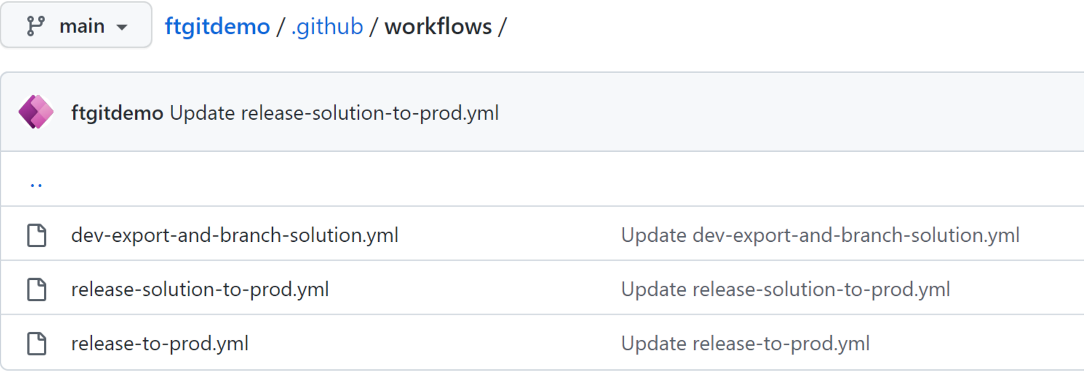
	Release
Jan	Tag(v1.0.0)
Feb	Release v1 [Tag(v1.0.0)]
March	
April	Tag(v1.0.1)
May	Tag(v1.0.2) Release v2 [Tag(v1.0.1)]
June	
July	Release v3 [Tag(v1.0.2)]



Workflows

Workflows are defined by a YAML file checked in to your repository, are stored in the **.github/workflows** folder and can have multiple workflow files, each of which can perform a different set of tasks.

For example, you can have one workflow to export your customisations from development as an unmanaged solution and add them to a new branch and another to convert this solution to managed and deploy to downstream environments



main ▾ ftgitdemo / .github / workflows /

ftgitdemo Update release-solution-to-prod.yml

..

dev-export-and-branch-solution.yml	Update dev-export-and-branch-solution.yml
release-solution-to-prod.yml	Update release-solution-to-prod.yml
release-to-prod.yml	Update release-to-prod.yml

GitHub Actions for Microsoft Power Platform

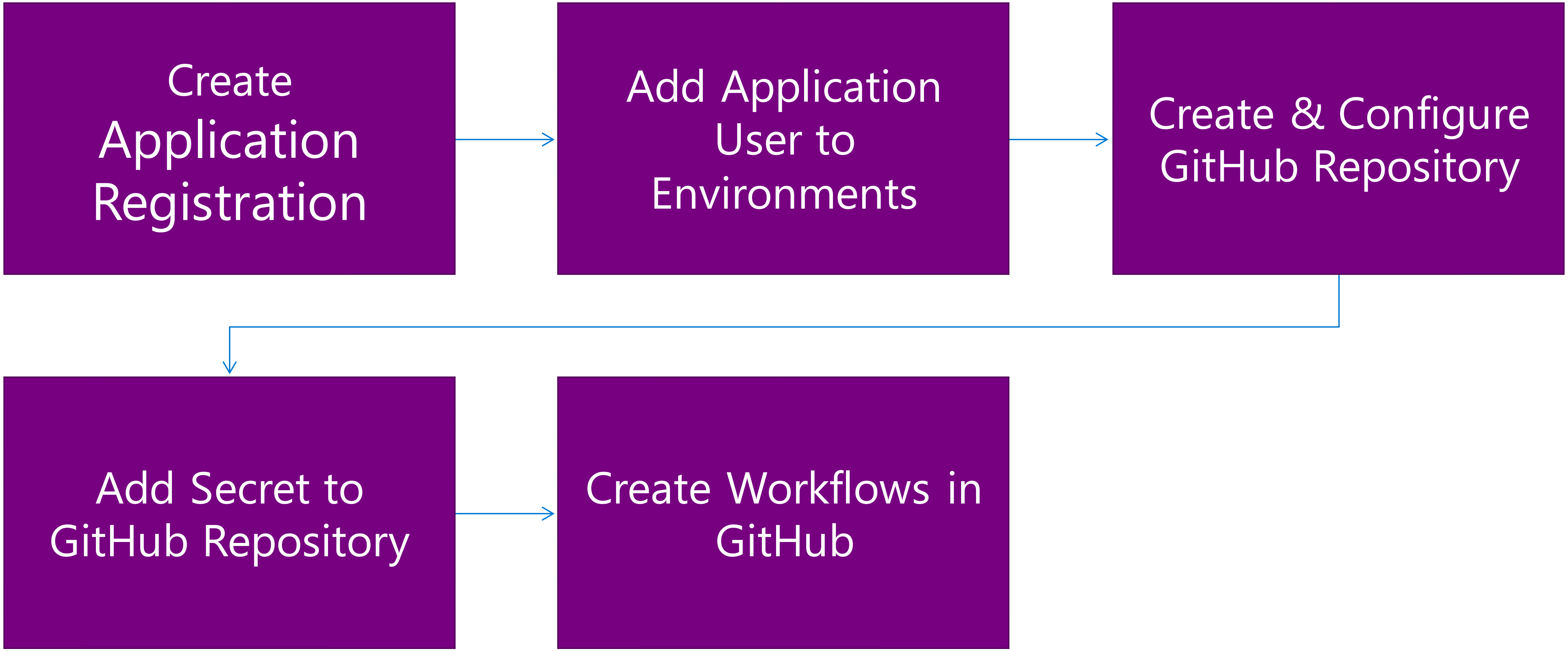
You can use GitHub Actions for Microsoft Power Platform along with any other available GitHub Actions to compose your build and release workflows.

GitHub Actions for Microsoft Power Platform include the following capabilities:

- Importing and exporting application metadata (also known as solutions) that contain various platform components such as canvas apps, model-driven apps, desktop flows, Power Virtual Agents chatbots, AI Builder models, customer engagement apps and connectors between development environments and source control.
- Deploying to downstream environments.
- Provisioning or de-provisioning environments
- Performing static analysis checks against solutions by using Power Apps solution checker.

End to End Deployment

Setup



Create Application Registration

Add Application User to Environments

Create & Configure GitHub Repository

Add Secret to GitHub Repository

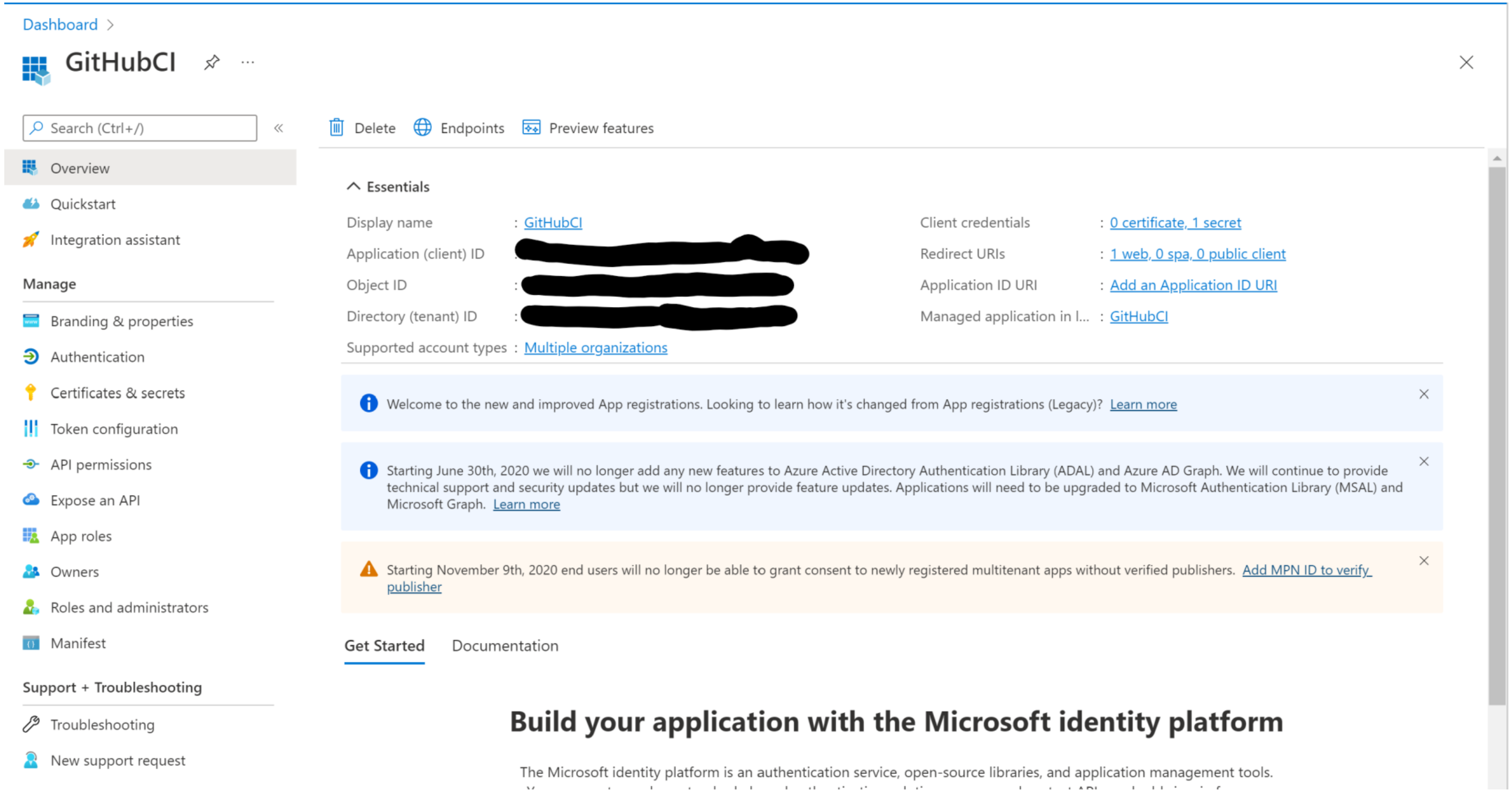
Create Workflows in GitHub

Create Application Registration

The GitHib Actions will execute the commands against your environments using an Application User.

- First Create the Application User in Azure Active Directory and give it permissions to Dataverse ([Tutorial: Register an app with Azure Active Directory (Microsoft Dataverse) - Power Apps | Microsoft Docs]())

- Create a secret for the application user



Display name : GitHubCI

Application (client) ID : [redacted]

Object ID : [redacted]

Directory (tenant) ID : [redacted]

Supported account types : Multiple organizations

Client credentials : 0 certificate, 1 secret

Redirect URIs : 1 web, 0 spa, 0 public client

Application ID URI : Add an Application ID URI

Managed application in l... : GitHubCI

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

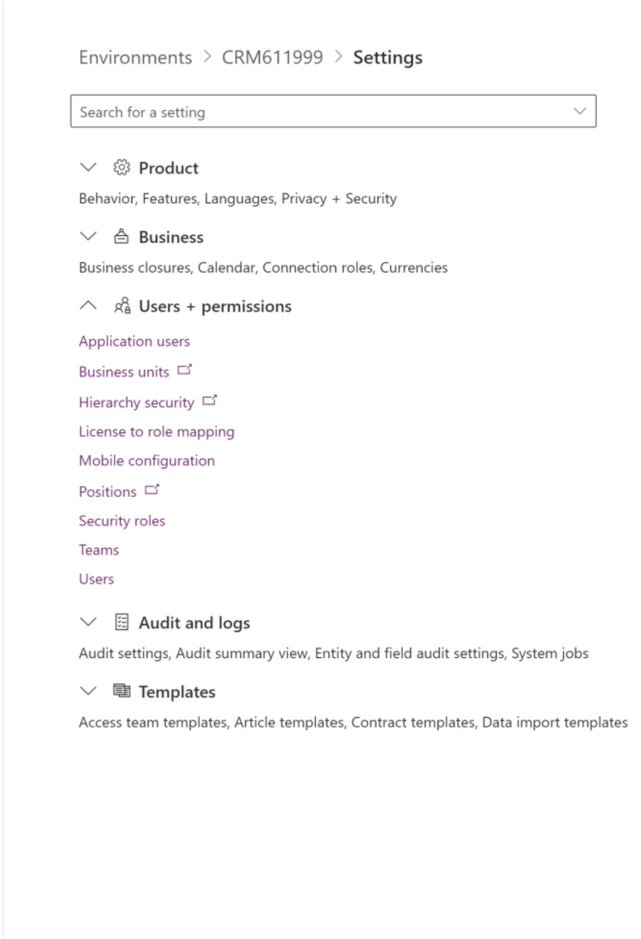
Starting November 9th, 2020 end users will no longer be able to grant consent to newly registered multitenant apps without verified publishers. Add MPN ID to verify publisher

Build your application with the Microsoft identity platform

The Microsoft identity platform is an authentication service, open-source libraries, and application management tools.

Add Application User to Environments

- [Manage application users in the Power Platform admin center - Power Platform | Microsoft Docs]()
- Provide security roles to be able to export and import solutions (Environment Maker, Solution Checker, System Customizer)



Environments > CRM611999 > Settings

Search for a setting

Product

Behavior, Features, Languages, Privacy + Security

Business

Business closures, Calendar, Connection roles, Currencies

Users + permissions

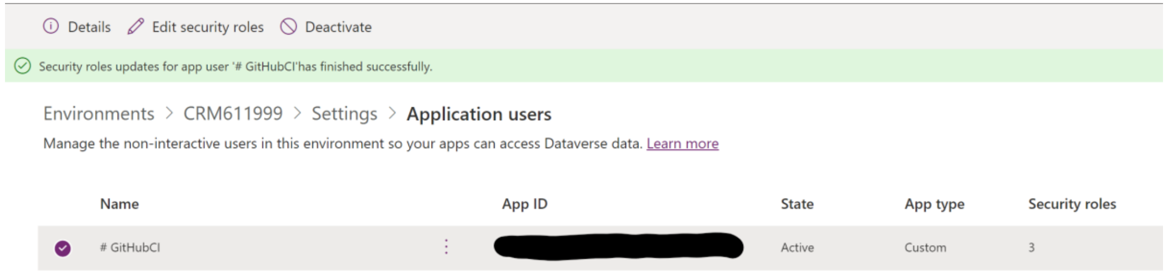
- Application users
- Business units
- Hierarchy security
- License to role mapping
- Mobile configuration
- Positions
- Security roles
- Teams
- Users

Audit and logs

Audit settings, Audit summary view, Entity and field audit settings, System jobs

Templates

Access team templates, Article templates, Contract templates, Data import templates



Security roles updates for app user '# GitHubCI'has finished successfully.

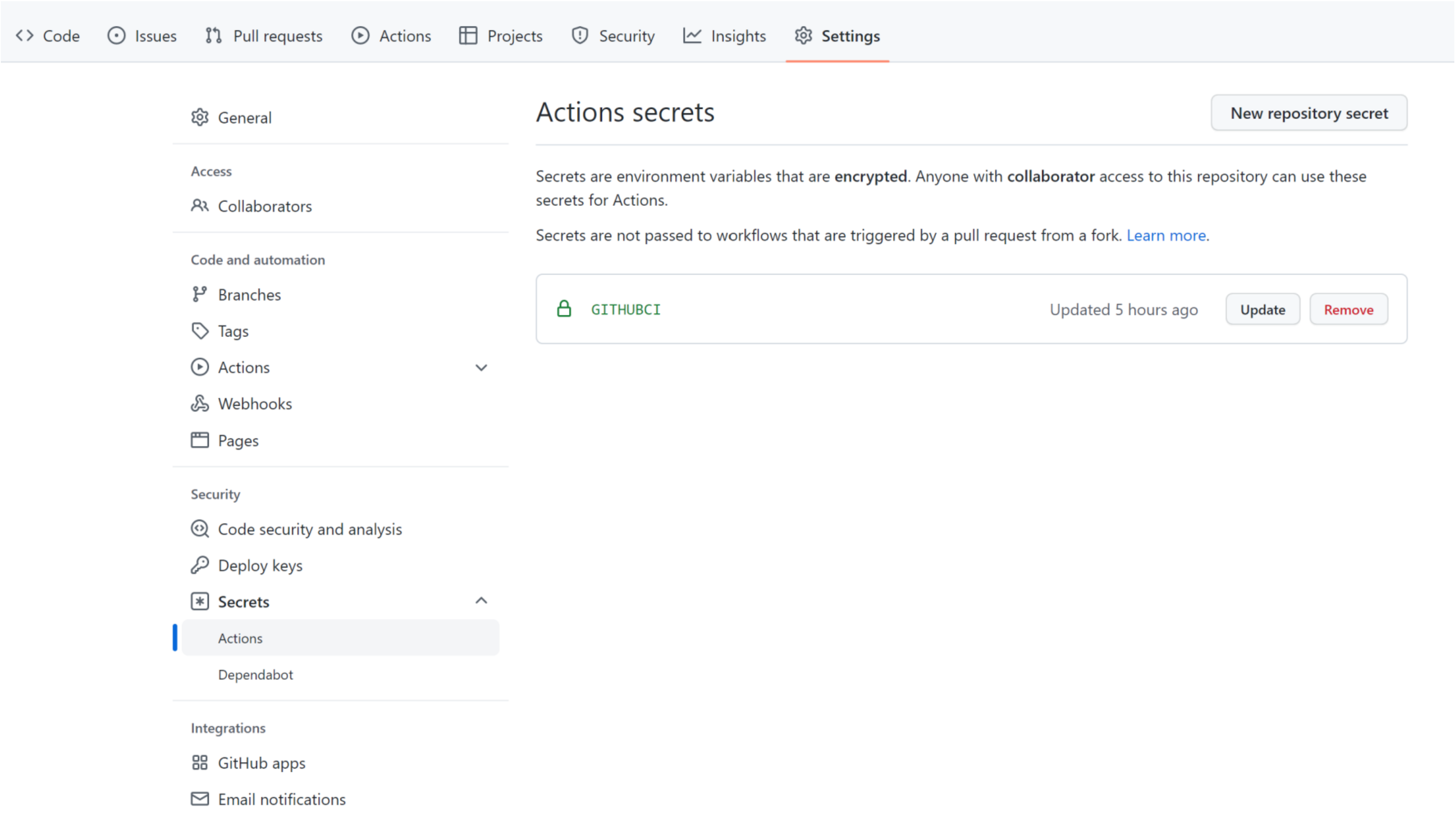
Environments > CRM611999 > Settings > Application users

Manage the non-interactive users in this environment so your apps can access Dataverse data. Learn more

Name	App ID	State	App type	Security roles
# GitHubCI	[redacted]	Active	Custom	3

Create & Configure GitHub Repository

- https://github.com/join
- Add the secret value for the application user to your GitHub secrets



Actions secrets

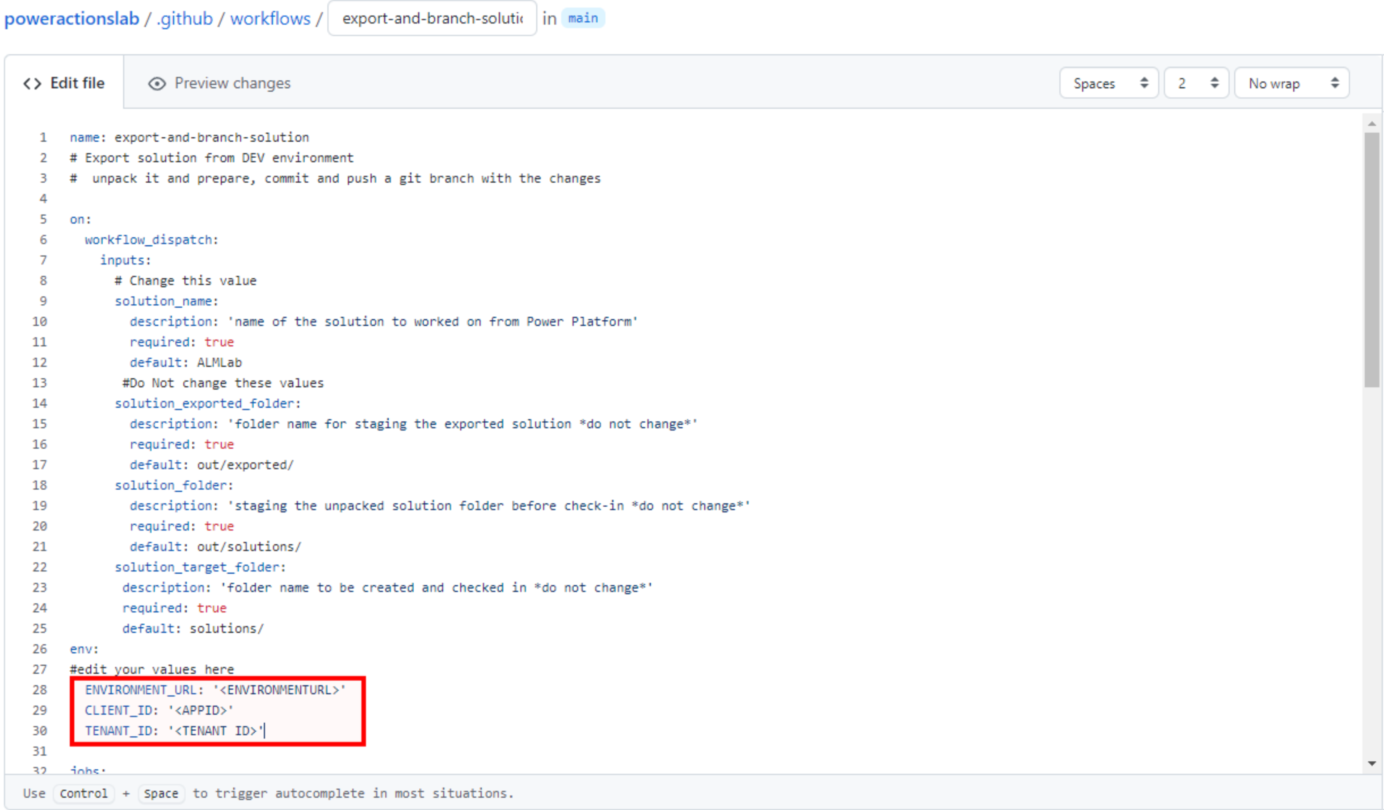
Secrets are environment variables that are **encrypted**. Anyone with **collaborator** access to this repository can use these secrets for Actions.

Secrets are not passed to workflows that are triggered by a pull request from a fork. Learn more.

GITHUBCI Updated 5 hours ago

Create Workflows in GitHub

- Create the workflow to export and unpack the unmanaged solution file to a new branch (copy from https://github.com/microsoft/powerplatform-actions-lab/blob/main/sample-workflows/export-and-branch-solution-with-spn-auth.yml to a new workflow file). Detailed instructions Tutorial: Automate solution deployment using GitHub Actions for Microsoft Power Platform - Power Platform | Microsoft Docs


poweractionslab / .github / workflows / export-and-branch-soluti in main
Edit file
Preview changes
Spaces
2
No wrap

```yaml
name: export-and-branch-solution
# Export solution from DEV environment
#  unpack it and prepare, commit and push a git branch with the changes

on:
  workflow_dispatch:
    inputs:
      # Change this value
      solution_name:
        description: 'name of the solution to worked on from Power Platform'
        required: true
        default: ALMLab
       #Do Not change these values
      solution_exported_folder:
        description: 'folder name for staging the exported solution *do not change*'
        required: true
        default: out/exported/
      solution_folder:
        description: 'staging the unpacked solution folder before check-in *do not change*'
        required: true
        default: out/solutions/
      solution_target_folder:
       description: 'folder name to be created and checked in *do not change*'
       required: true
       default: solutions/
env:
#edit your values here
  ENVIRONMENT_URL: '<ENVIRONMENTURL>'
  CLIENT_ID: '<APPID>'
  TENANT_ID: '<TENANT ID>'

jobs:
```

Use Control + Space to trigger autocomplete in most situations.

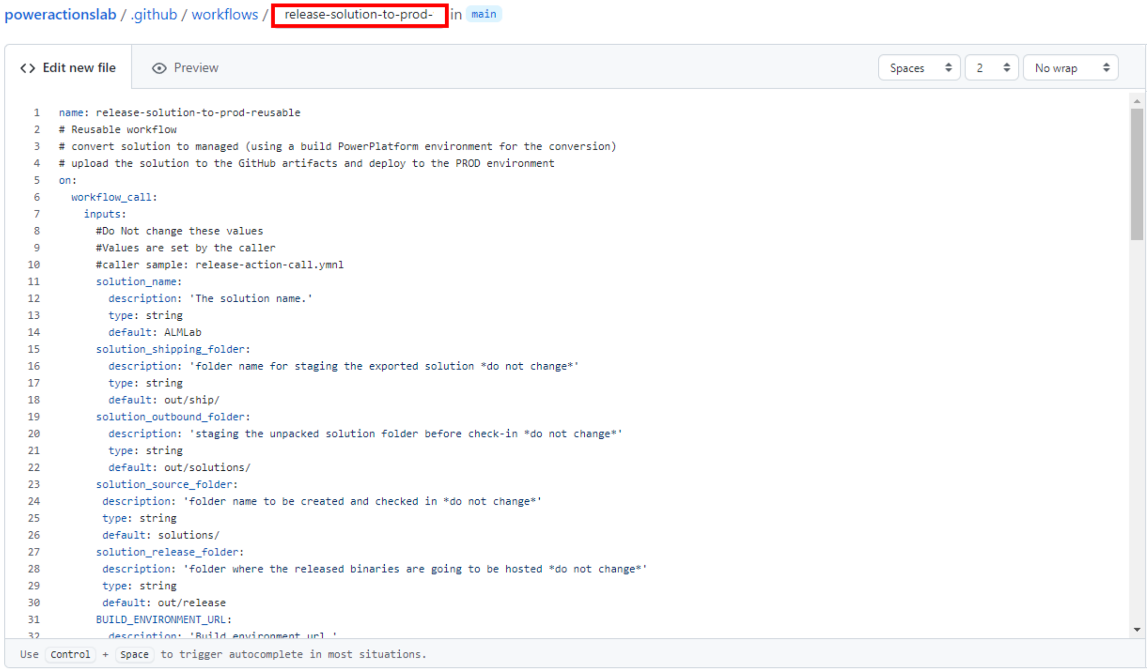
- Environment URL – 'https://devenv.crm.dynamics.com/
- Client ID – 'Copy from AAD App Registration'
- Tenant ID – 'Copy from AAD'

- Update the default value for solution_name
- Update the secret name to match the one created in GitHub
- client-secret: ${{ secrets.YOURSECRETNAME }}

- Look for 'name: export-solution action' and add the async attribute 'run-asynchronously: true'

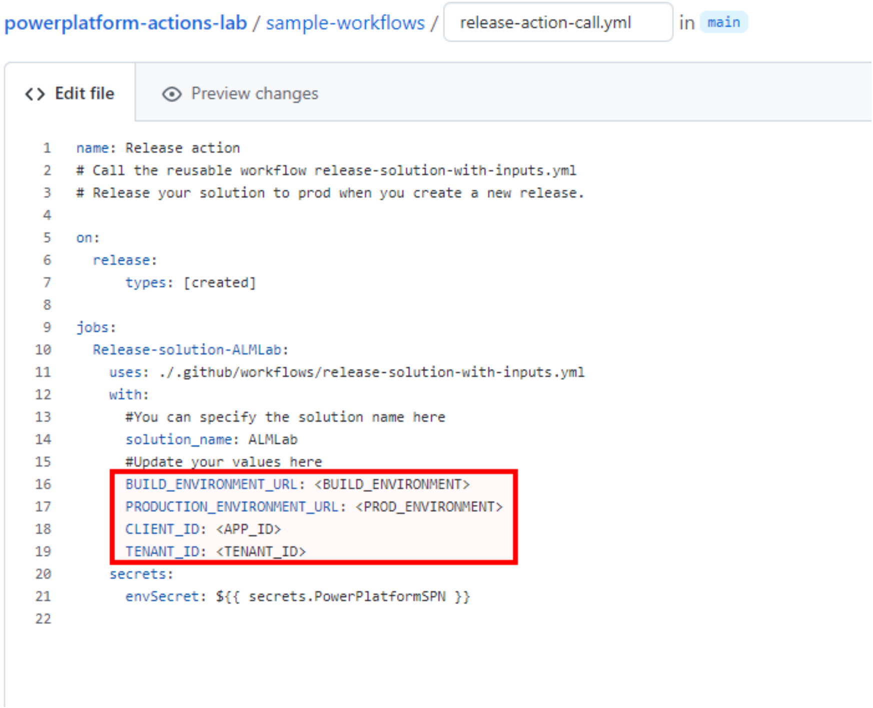
Create Workflows in GitHub continued

- Create the reusable workflow to release the solution file to a new env (copy from https://github.com/microsoft/powerplatform-actions-lab/blob/main/sample-workflows/release-solution-to-prod-with-inputs.yml to a new workflow file)
- Call the reusable workflow on a release event (copy from https://github.com/microsoft/powerplatform-actions-lab/blob/main/sample-workflows/release-action-call.yml)


poweractionslab / .github / workflows / release-solution-to-prod- in main
<> Edit new file
Preview
Spaces
2
No wrap

```yaml
name: release-solution-to-prod-reusable
# Reusable workflow
# convert solution to managed (using a build PowerPlatform environment for the conversion)
# upload the solution to the GitHub artifacts and deploy to the PROD environment
on:
  workflow_call:
    inputs:
      #Do Not change these values
      #Values are set by the caller
      #caller sample: release-action-call.ymnl
      solution_name:
        description: 'The solution name.'
        type: string
        default: ALMLab
      solution_shipping_folder:
        description: 'folder name for staging the exported solution *do not change*'
        type: string
        default: out/ship/
      solution_outbound_folder:
        description: 'staging the unpacked solution folder before check-in *do not change*'
        type: string
        default: out/solutions/
      solution_source_folder:
       description: 'folder name to be created and checked in *do not change*'
       type: string
       default: solutions/
      solution_release_folder:
       description: 'folder where the released binaries are going to be hosted *do not change*'
       type: string
       default: out/release
      BUILD_ENVIRONMENT_URL:
        description: 'Build environment url.'
```

Use Control + Space to trigger autocomplete in most situations.


powerplatform-actions-lab / sample-workflows / release-action-call.yml in main
<> Edit file
Preview changes

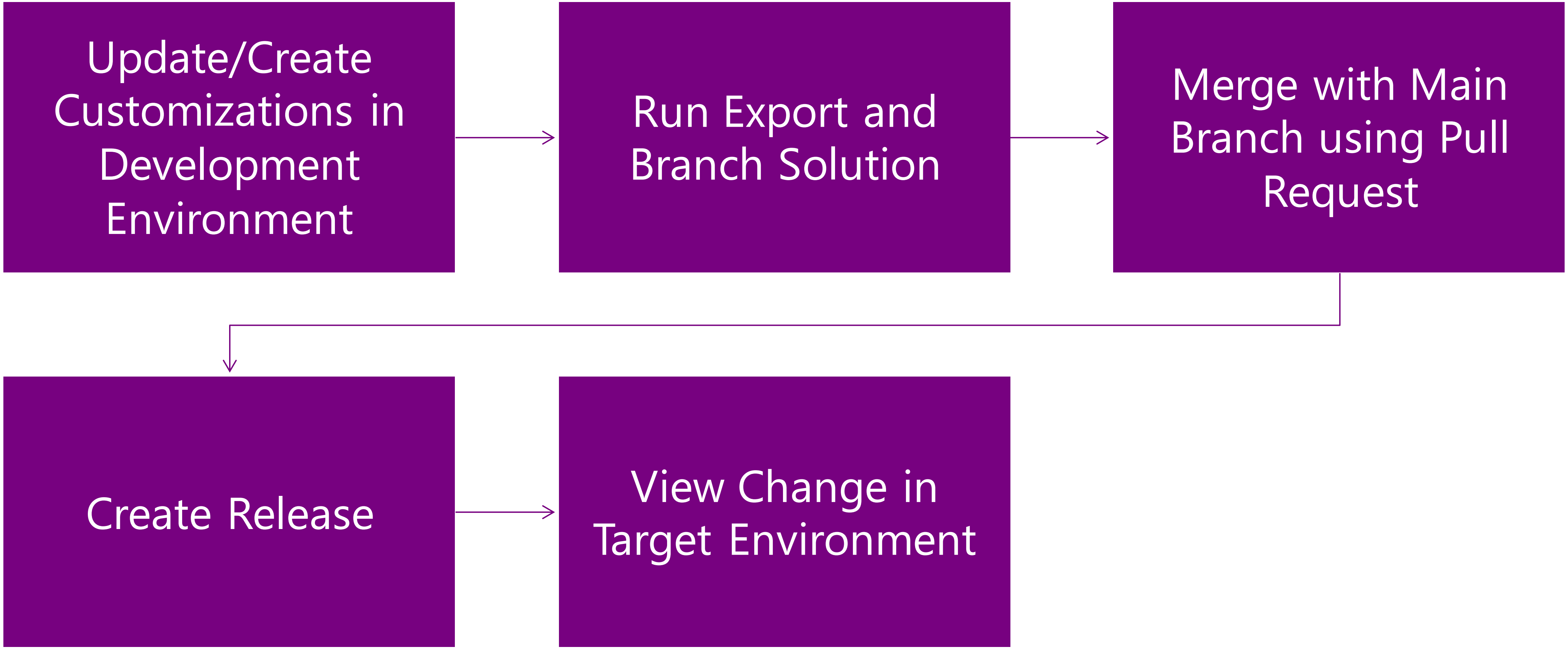
```yaml
name: Release action
# Call the reusable workflow release-solution-with-inputs.yml
# Release your solution to prod when you create a new release.

on:
  release:
      types: [created]

jobs:
  Release-solution-ALMLab:
    uses: ./.github/workflows/release-solution-with-inputs.yml
    with:
      #You can specify the solution name here
      solution_name: ALMLab
      #Update your values here
      BUILD_ENVIRONMENT_URL: <BUILD_ENVIRONMENT>
      PRODUCTION_ENVIRONMENT_URL: <PROD_ENVIRONMENT>
      CLIENT_ID: <APP_ID>
      TENANT_ID: <TENANT_ID>
    secrets:
      envSecret: ${{ secrets.PowerPlatformSPN }}

```

Execution



Update/Create Customizations in Development Environment

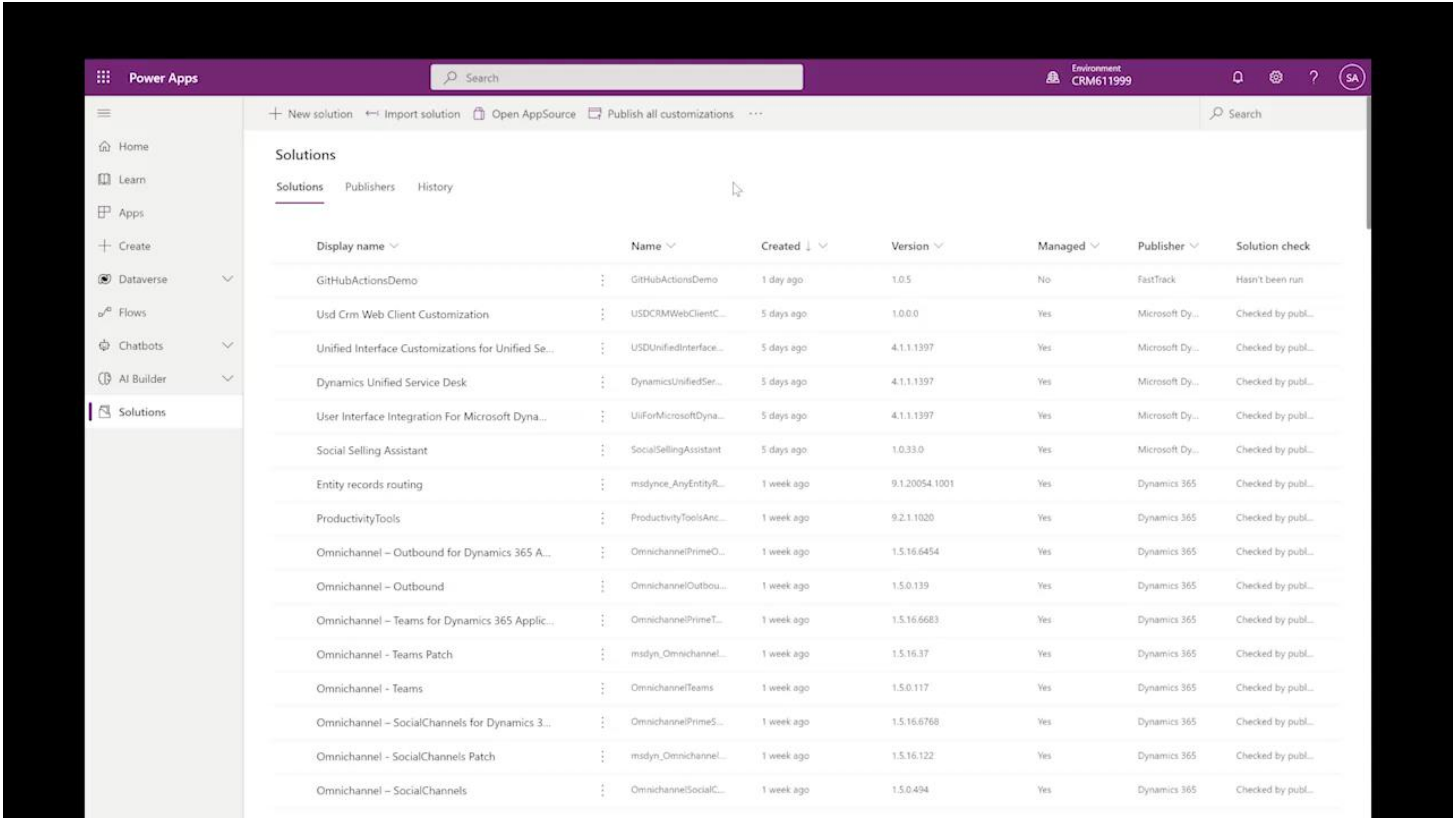
Run Export and Branch Solution

Merge with Main Branch using Pull Request

Create Release

View Change in Target Environment

Execution of end to end release



Solutions

Solutions | Publishers | History

Display name	Name	Created ↓	Version	Managed	Publisher	Solution check
GitHubActionsDemo	GitHubActionsDemo	1 day ago	1.0.5	No	FastTrack	Hasn't been run
Usd Crm Web Client Customization	USDCRMWebClientC...	5 days ago	1.0.0.0	Yes	Microsoft Dy...	Checked by publ...
Unified Interface Customizations for Unified Se...	USDUnifiedInterface...	5 days ago	4.1.1.1397	Yes	Microsoft Dy...	Checked by publ...
Dynamics Unified Service Desk	DynamicsUnifiedSer...	5 days ago	4.1.1.1397	Yes	Microsoft Dy...	Checked by publ...
User Interface Integration For Microsoft Dyna...	UiiForMicrosoftDyna...	5 days ago	4.1.1.1397	Yes	Microsoft Dy...	Checked by publ...
Social Selling Assistant	SocialSellingAssistant	5 days ago	1.0.33.0	Yes	Microsoft Dy...	Checked by publ...
Entity records routing	msdynce_AnyEntityR...	1 week ago	9.1.20054.1001	Yes	Dynamics 365	Checked by publ...
ProductivityTools	ProductivityToolsAnc...	1 week ago	9.2.1.1020	Yes	Dynamics 365	Checked by publ...
Omnichannel – Outbound for Dynamics 365 A...	OmnichannelPrimeO...	1 week ago	1.5.16.6454	Yes	Dynamics 365	Checked by publ...
Omnichannel – Outbound	OmnichannelOutbou...	1 week ago	1.5.0.139	Yes	Dynamics 365	Checked by publ...
Omnichannel – Teams for Dynamics 365 Applic...	OmnichannelPrimeT...	1 week ago	1.5.16.6683	Yes	Dynamics 365	Checked by publ...
Omnichannel - Teams Patch	msdyn_Omnichannel...	1 week ago	1.5.16.37	Yes	Dynamics 365	Checked by publ...
Omnichannel - Teams	OmnichannelTeams	1 week ago	1.5.0.117	Yes	Dynamics 365	Checked by publ...
Omnichannel – SocialChannels for Dynamics 3...	OmnichannelPrimeS...	1 week ago	1.5.16.6768	Yes	Dynamics 365	Checked by publ...
Omnichannel - SocialChannels Patch	msdyn_Omnichannel...	1 week ago	1.5.16.122	Yes	Dynamics 365	Checked by publ...
Omnichannel – SocialChannels	OmnichannelSocialC...	1 week ago	1.5.0.494	Yes	Dynamics 365	Checked by publ...

Resources

Useful Resources

Create and manage environments in the Power Platform admin center - Power Platform | Microsoft Docs
Solution concepts - Power Platform | Microsoft Docs
GitHub Actions for Microsoft Power Platform - Power Platform | Microsoft Docs
Manage application users in the Power Platform admin center - Power Platform | Microsoft Docs
microsoft/powerplatform-actions: Power Platform GitHub Actions
Repositories - GitHub Docs
Learn yaml in Y Minutes (learnxinyminutes.com)
Microsoft Customer Digital Experiences | Power Platform & GitHub: Pro-Developer Experience
Hello-Branch-Merge Video
Microsoft Power Platform CLI - Power Platform | Microsoft Docs
Microsoft Power Platform Build Tools for Azure DevOps - Power Platform | Microsoft Docs