Microsoft Dynamics 365

What's new in AL-Go for GitHub

What is AL-Go for GitHub

The plug-and-play DevOps solution
for Business Central development on GitHub

Offering an easy path for Business Central developers to secure CI/CD

Including quality assurance, change tracking, validating code consistency and compatibility

Managing Business Central and ISV updates

Handles collaboration, code reviews and much more

No prior knowledge in docker, PowerShell or yaml needed

No need for a dedicated DevOps engineer

This session is for people using/familiar to AL-Go for GitHub

To learn more about how to get started with AL-Go for GitHub go to https://aka.ms/algoworkshop

Topics

Settings and Updating

Code signing apps

Deployment enhancements

Custom Deployment

Pull Request Status Check

CompilerFolder and Linux

Use GhTokenWorkflow

+ A look into the future…

Settings and upgrading

Where are my settings?
What version of AL-Go am I running?
How do I upgrade?

Where are my Settings?

Three types of settings

Repository settings

Examples: Type, TemplateUrl, workflow schedules, runs-on, environments, branch settings, etc.

Additionally: Repository wide project settings (project settings, which should affect all projects)

Location: .github/AL-Go-Settings.json

Additionally: GitHub variables ALGoOrgSettings and ALGoRepoSettings.

Project specific settings

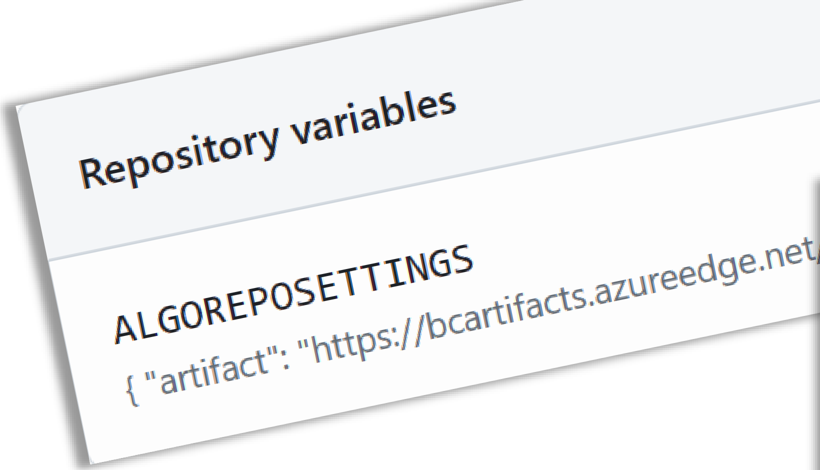
Examples: appFolders, testFolders, appDependencyProbingPaths, projectName, artifact, etc.

Location: .AL-Go/settings.json (or <project>/.AL-Go/settings.json

Workflow specific settings

Examples: artifact, cacheImageName, versioningStrategy

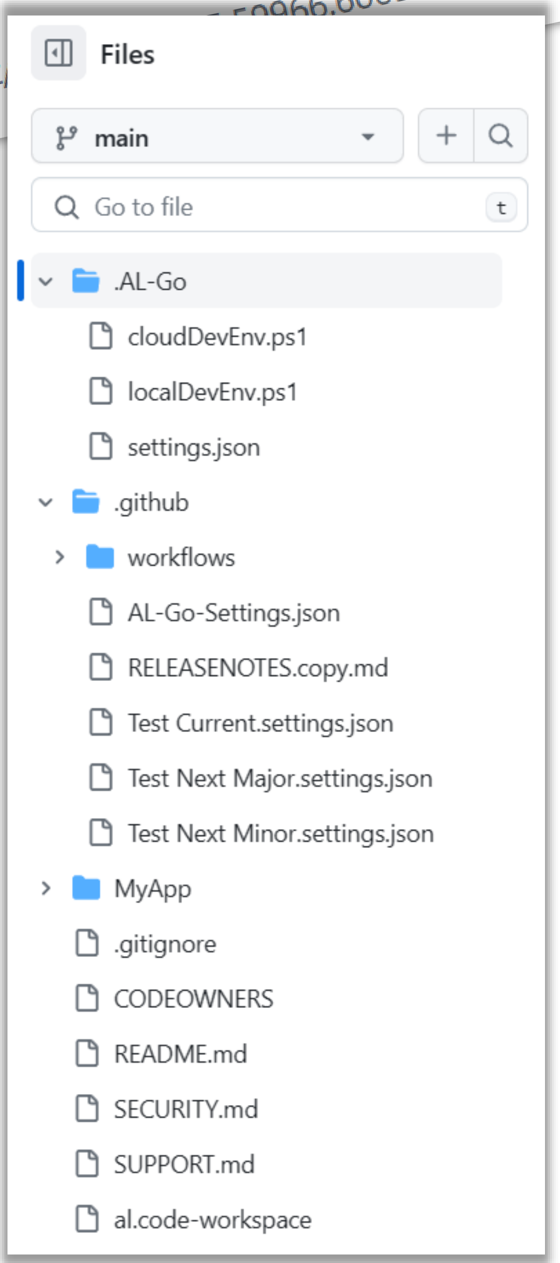
Location: .github/<workflow>.settings.json



Repository variables

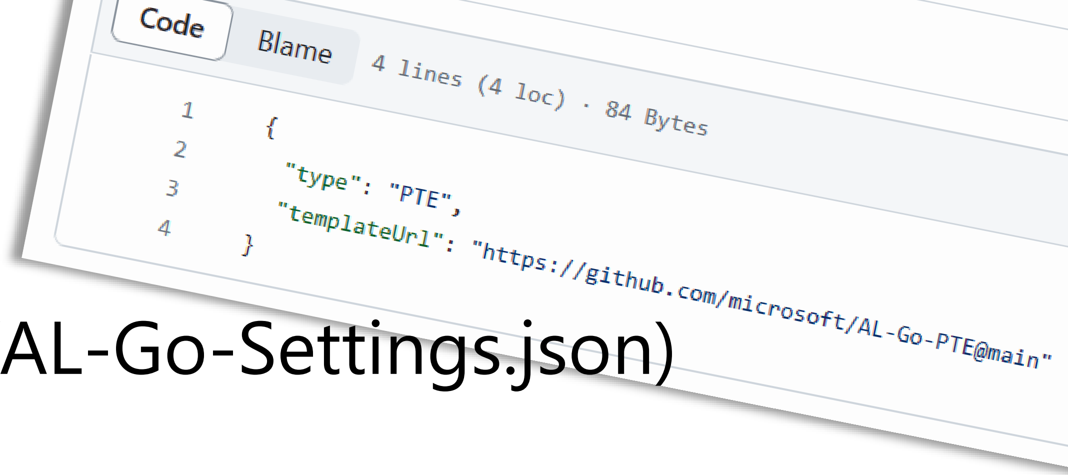
ALGOREPOSETTINGS

{ "artifact": "https://bcartifacts.azureedge.net/ 0966.60650/us" }



.AL-Go
cloudDevEnv.ps1
localDevEnv.ps1
settings.json
.github
workflows
AL-Go-Settings.json
RELEASENOTES.copy.md
Test Current.settings.json
Test Next Major.settings.json
Test Next Minor.settings.json
MyApp
.gitignore
CODEOWNERS
README.md
SECURITY.md
SUPPORT.md
al.code-workspace

What version of AL-Go???



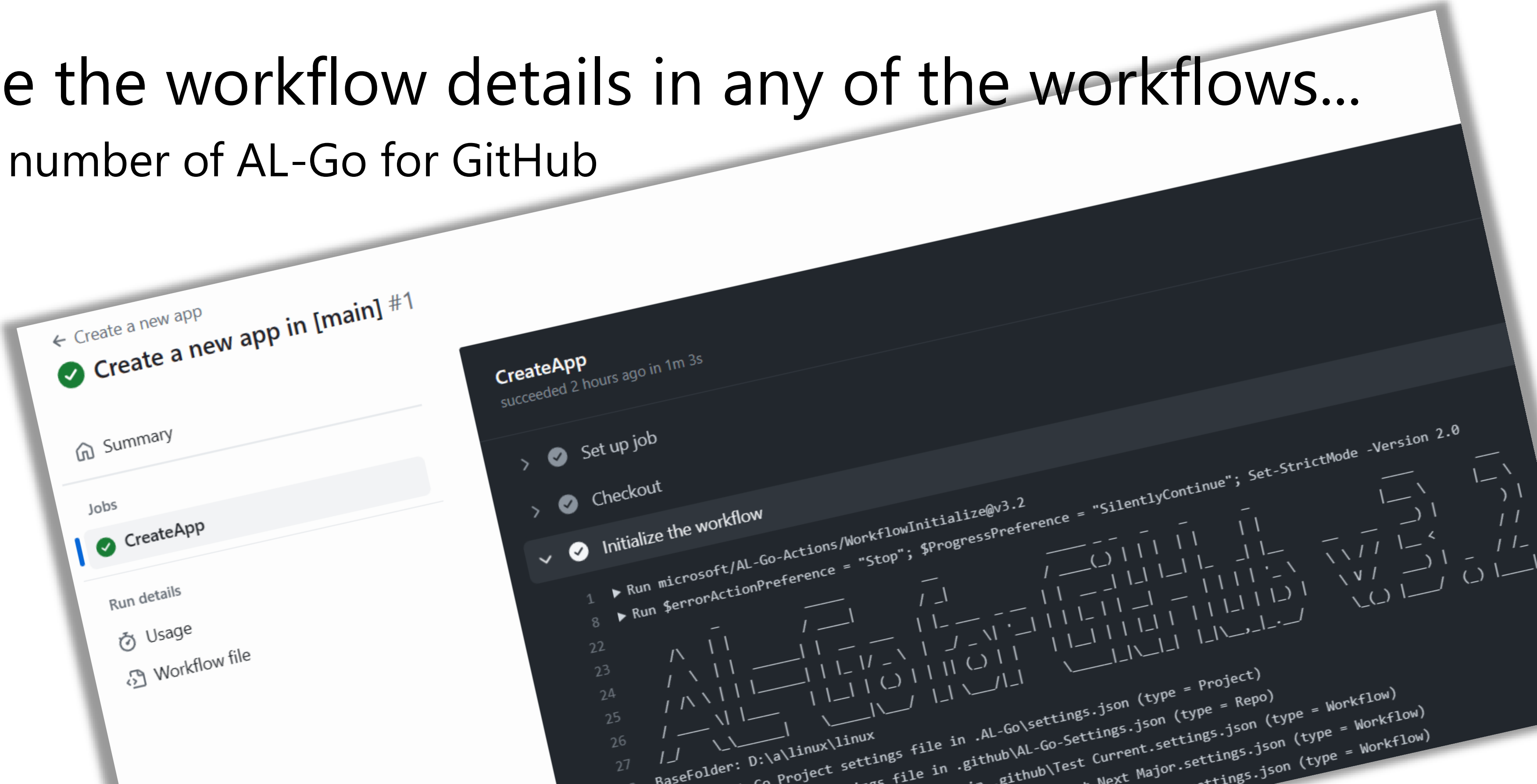
```
{
  "type": "PTE",
  "templateUrl": "https://github.com/microsoft/AL-Go-PTE@main"
}
```

The TemplateUrl in repo settings (.github/AL-Go-Settings.json)

Tells you where AL-Go for GitHub is looking for updates

Look into the Initialize the workflow details in any of the workflows…

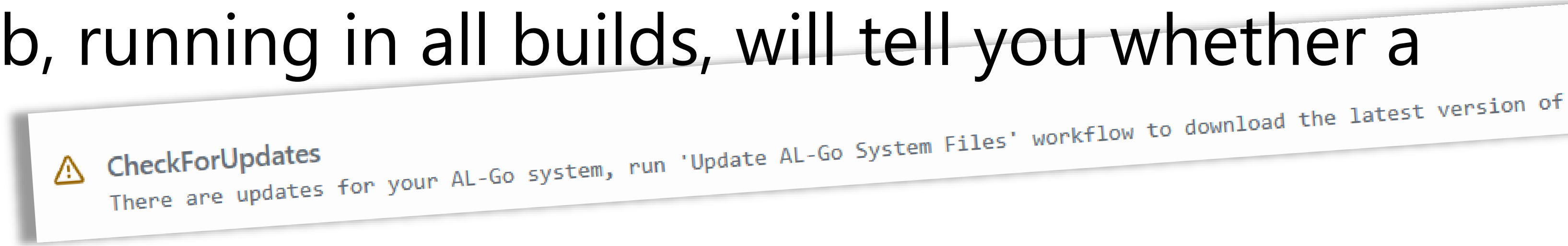
Tells you the actual version number of AL-Go for GitHub



```
▶ Run microsoft/AL-Go-Actions/WorkflowInitialize@v3.2
▶ Run $errorActionPreference = "Stop"; $ProgressPreference = "SilentlyContinue"; Set-StrictMode -Version 2.0
BaseFolder: D:\a\linux\linux
Go Project settings file in .AL-Go\settings.json (type = Project)
ngs file in .github\AL-Go-Settings.json (type = Repo)
in .github\Test Current.settings.json (type = Workflow)
Next Major.settings.json (type = Workflow)
ttings.json (type = Workflow)
```

How do I upgrade???

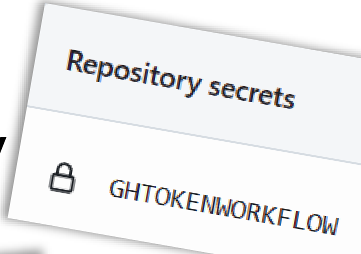
The CheckForUpdates job, running in all builds, will tell you whether a new update is available



⚠ CheckForUpdates

There are updates for your AL-Go system, run 'Update AL-Go System Files' workflow to download the latest version of

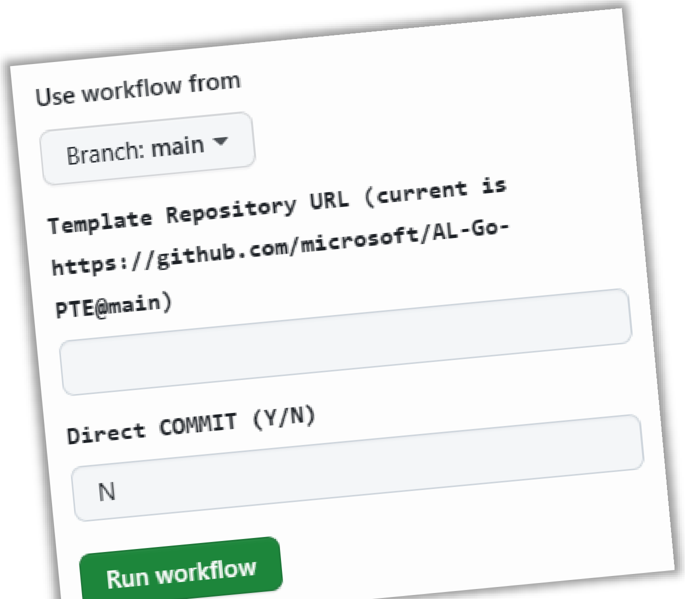
Add a secret to your org or repository called GhTokenWorkflow containing a Personal Access Token, with permissions to modify Workflows



Repository secrets

🔒 GHTOKENWORKFLOW

Run the Update AL-Go System Files workflow



Use workflow from

Branch: main ▾

Template Repository URL (current is https://github.com/microsoft/AL-Go-PTE@main)

Direct COMMIT (Y/N)

N

Run workflow

Code Signing apps

With the private key in a HSM key storage

Code Signing anno 2023

New private key storage requirement for Code Signing certificates

AL-Go for GitHub supports Azure KeyVault as an HSM key storage.

Retrieve your certificate in an Azure KeyVault and Setup the connection from AL-Go for GitHub

Deployment enhancements

Deployment done the right way...

GitHub Environments

AL-Go utilizes GitHub Environments and supports branch policies.

Use new setting ExcludeEnvironments to exclude environments from AL-Go Deployment (github_pages and other system environments)

For free SKUs, you can also define environments in settings

!!! Breaking change !!!

The following secrets no longer work (deployment will fail):

<environmentname>_EnvironmentName
(and variations)

<environmentname>_Projects
(and variations)

Replaced by properties in
DeployTo<environmentname>

DeployTo<environmentname>

```
"DeployToQA": {
    "EnvironmentType": "SaaS",
    "EnvironmentName": "QA",
    "Branches": [ "main" ],
    "Projects": "*",
    "SyncMode": "ForceSync",
    "ContinuousDeployment": true,
    "runs-on": "ubuntu-latest"
}
```

Type of environment

Actual name of environment

Branches deploying to environment (can also be defined in GitHub branch policies)

AL-Go Projects to install in the environment

SyncMode when installing in the environment

Continuous deployment?

Which GitHub runner to use for deployment

Link

Custom Deployment

Do it your way!

Custom Deployment

Specify EnvironmentType in Environment settings

EnvironmentType determines mechanism used to deploy

Write Custom Script

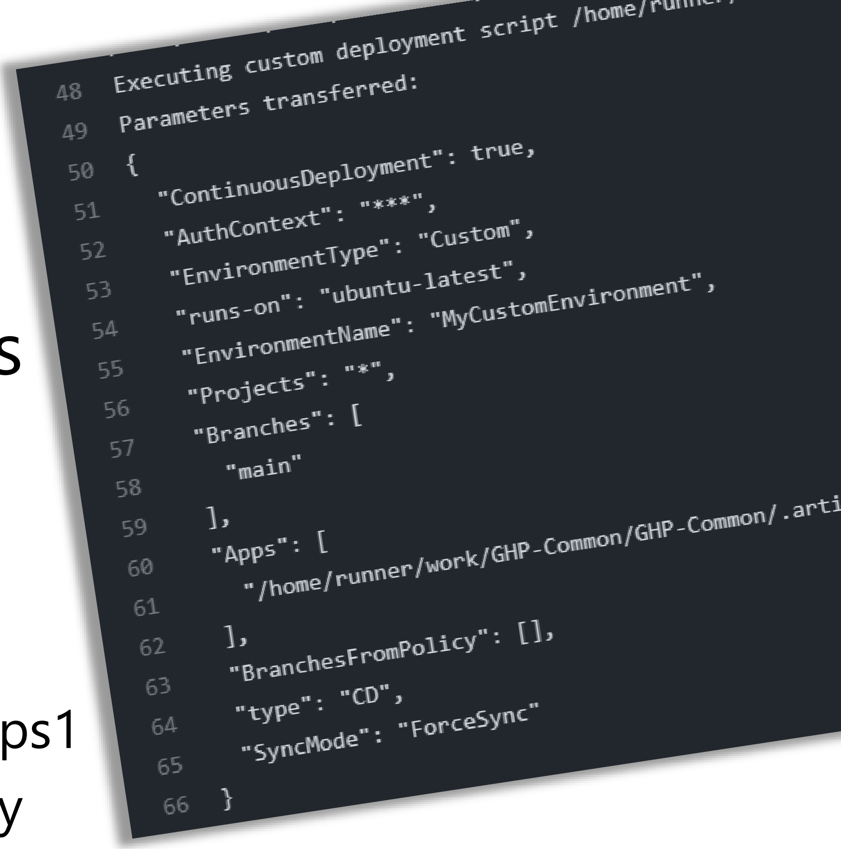
Place a PowerShell script in .github/DeployTo<environmenttype>.ps1

$parameters will contain all environment settings + apps to deploy

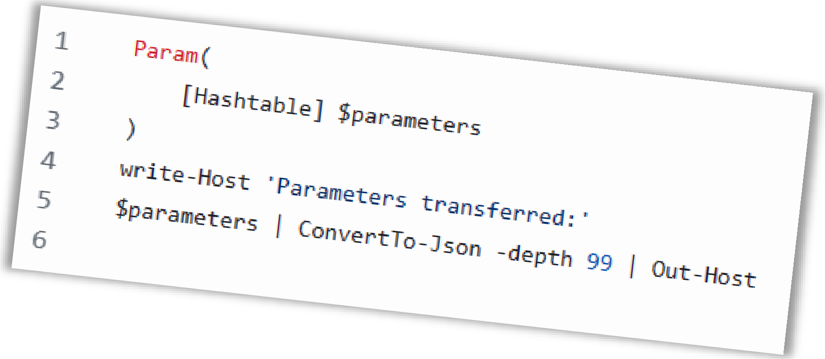
Write your own script !!!!! use the ALOps-External-Deployer PowerShell module

Optionally use Custom GitHub Runner

Environment specific GitHub runner,
which can be on the other side of a firewall



```
Executing custom deployment script /home/runne
Parameters transferred:
{
  "ContinuousDeployment": true,
  "AuthContext": "***",
  "EnvironmentType": "Custom",
  "runs-on": "ubuntu-latest",
  "EnvironmentName": "MyCustomEnvironment",
  "Projects": "*",
  "Branches": [
    "main"
  ],
  "Apps": [
    "/home/runner/work/GHP-Common/GHP-Common/.arti
  ],
  "BranchesFromPolicy": [],
  "type": "CD",
  "SyncMode": "ForceSync"
}
```



```
Param(
    [Hashtable] $parameters
)
write-Host 'Parameters transferred:'
$parameters | ConvertTo-Json -depth 99 | Out-Host
```

Link

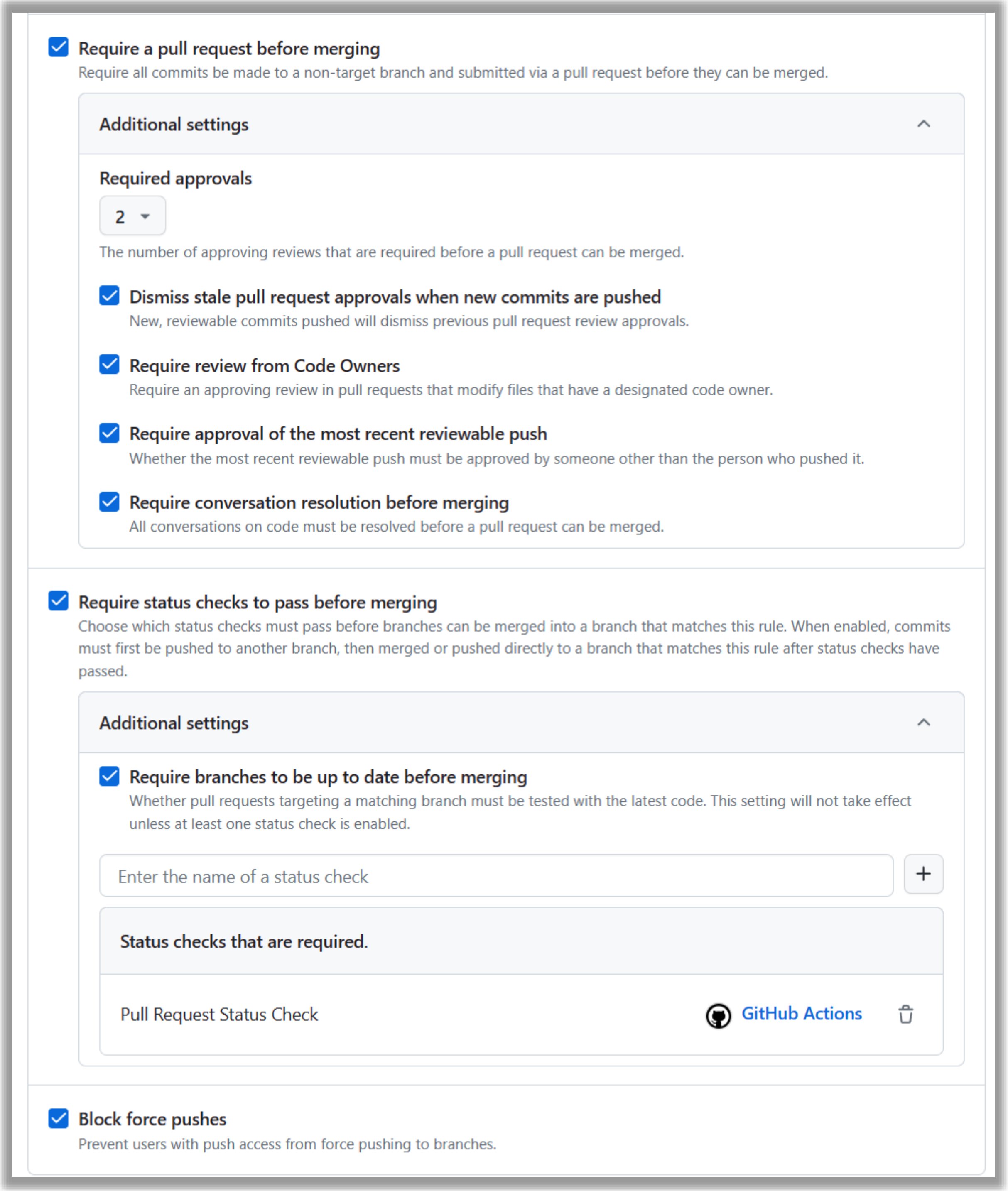
Pull Request Status Check

Require successful PR build before merging...

Pull Request Status Check

When setting up rulesets…

Use this job to determine whether or not the build was successful and the PR can be merged.



☑ Require a pull request before merging

Require all commits be made to a non-target branch and submitted via a pull request before they can be merged.

Additional settings

Required approvals

2

The number of approving reviews that are required before a pull request can be merged.

☑ Dismiss stale pull request approvals when new commits are pushed

New, reviewable commits pushed will dismiss previous pull request review approvals.

☑ Require review from Code Owners

Require an approving review in pull requests that modify files that have a designated code owner.

☑ Require approval of the most recent reviewable push

Whether the most recent reviewable push must be approved by someone other than the person who pushed it.

☑ Require conversation resolution before merging

All conversations on code must be resolved before a pull request can be merged.

☑ Require status checks to pass before merging

Choose which status checks must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

Additional settings

☑ Require branches to be up to date before merging

Whether pull requests targeting a matching branch must be tested with the latest code. This setting will not take effect unless at least one status check is enabled.

Enter the name of a status check +

Status checks that are required.

Pull Request Status Check — GitHub Actions

☑ Block force pushes

Prevent users with push access from force pushing to branches.

CompilerFolder and Linux

We are on a mission to make things faster, easier and cheaper for you!

CompilerFolder and Linux

Today, the RunPipeline job is a do-it-all job. It builds, publishes and runs tests and performance tests, using a docker container for all

This will be split into a build and a test job, where the build part can run on the faster and cheaper linux runners using CompilerFolders

CompilerFolders is a new functionality, which allows AL-Go to build apps without containers and allows GitHub to cache artifacts.

GitHub hosted Linux runners are half the price of Windows runners and ~33% faster

Link

CompilerFolder and Linux

With this release you can enable your builds to use the compilerFolder feature and linux by modifying settings:

```
"doNotPublishApps": true,
"useCompilerFolder": true,
"runs-on": "ubuntu-latest",
"githubRunner": "ubuntu-latest"
```

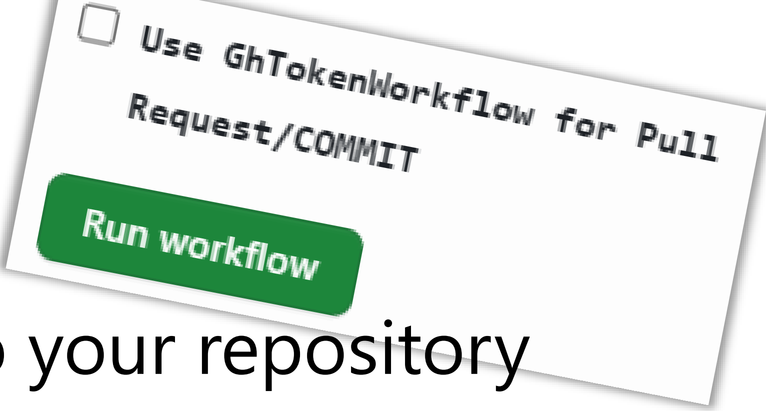
Note that the doNotPublishApps setting will disable running tests.

Link

Use GhTokenWorkflow

Overcoming the limitations of the GITHUB_TOKEN

Use GhTokenWorkflow?



☐ Use GhTokenWorkflow for Pull Request/COMMIT

New option for all workflows, which pushes data to your repository

- Create release, Increment Version Number and Create new apps, test apps or perf test apps

Issue 555: A workflow using the default GitHub token cannot invoke other workflows (i.e. Pull Request Build or Commit build)

- See more

Using GhTokenWorkflow

- Not dependent on the "Allow GitHub Actions to create and approve pull requests" setting
- Commit or Pull Request Build are automatically invoked based on direct COMMIT setting

Link

A look into the future…

What’s cooking…
No promises, no dates…

BCPT Test Result viewer

Pull Request: https://github.com/microsoft/AL-Go/pull/619

Feel free to comment or make suggestions!

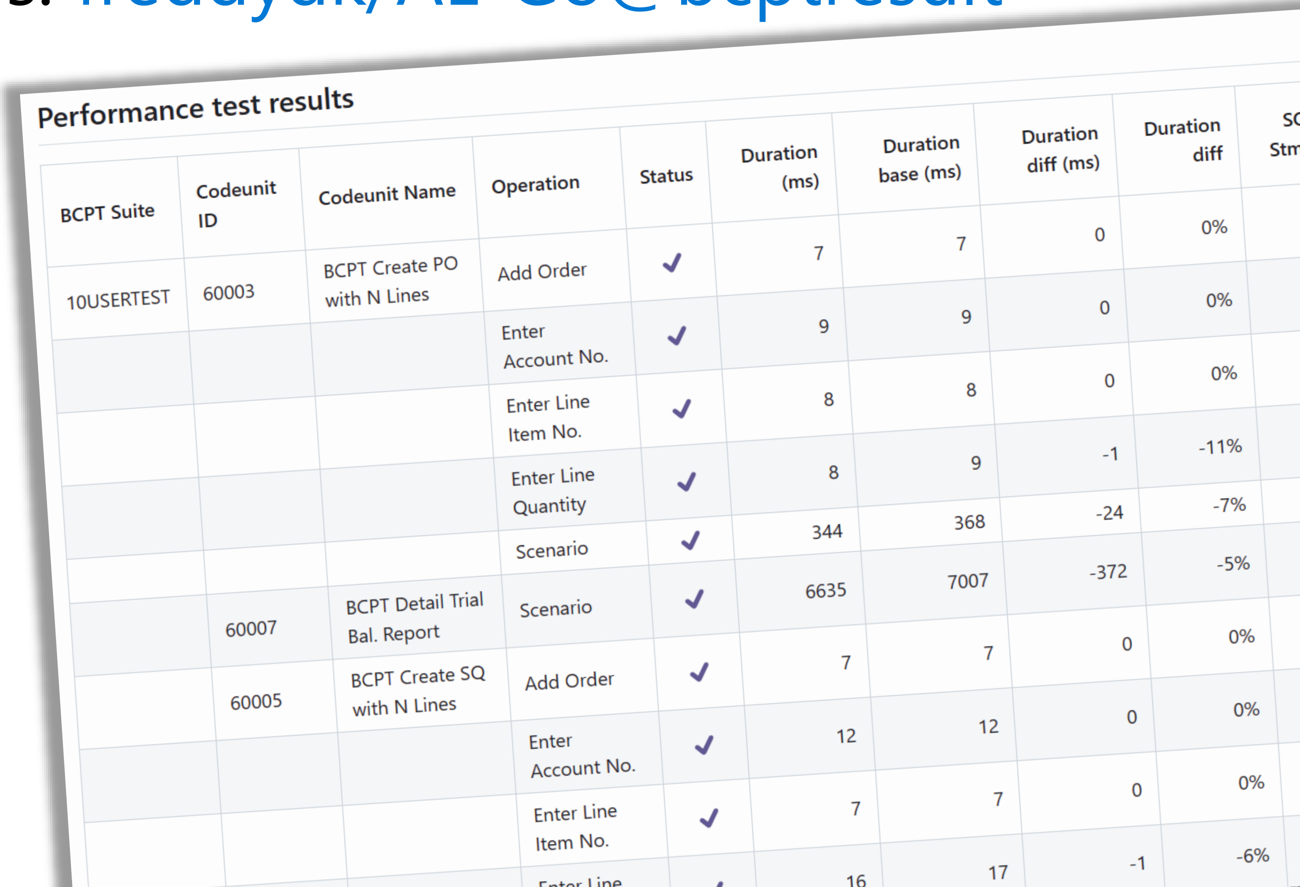
Template repository supporting this: freddydk/AL-Go@bcptresult

Try it out – give us feedback!

Test repository: freddydk/bcpttest

Define baseline and thresholds

Give warnings or errors based on performance degradations



Performance test results

BCPT Suite	Codeunit ID	Codeunit Name	Operation	Status	Duration (ms)	Duration base (ms)	Duration diff (ms)	Duration diff	SC Stm
10USERTEST	60003	BCPT Create PO with N Lines	Add Order	✓	7	7	0	0%	
			Enter Account No.	✓	9	9	0	0%	
			Enter Line Item No.	✓	8	8	0	0%	
			Enter Line Quantity	✓	8	9	-1	-11%	
			Scenario	✓	344	368	-24	-7%	
	60007	BCPT Detail Trial Bal. Report	Scenario	✓	6635	7007	-372	-5%	
	60005	BCPT Create SQ with N Lines	Add Order	✓	7	7	0	0%	
			Enter Account No.	✓	12	12	0	0%	
			Enter Line Item No.	✓	7	7	0	0%	
			Enter Line	✓	16	17	-1	-6%	

Customize AL-Go for GitHub

Pull Request: https://github.com/microsoft/AL-Go/pull/715

Feel free to comment or make suggestions!

Template repository supporting this: freddydk/AL-Go@custom

Try it out – give us feedback!

Indirect template repository: freddydk/MyIndirectPteTemplate

Customized

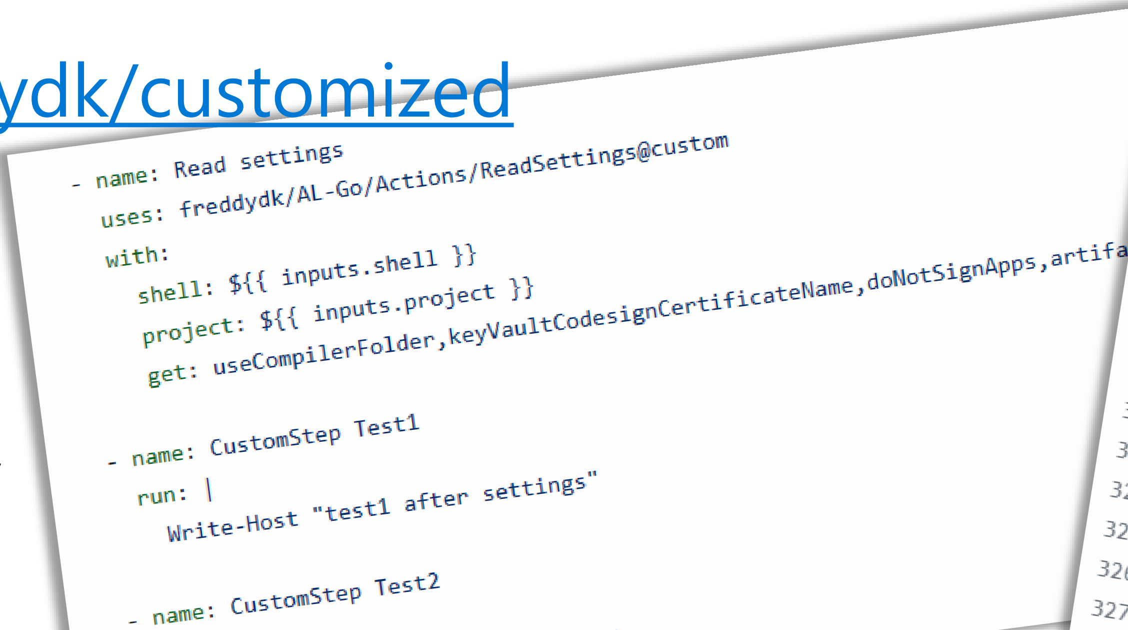
Test repository: freddydk/customized

Supports:

Custom Jobs, Custom Steps

Indirect template repository

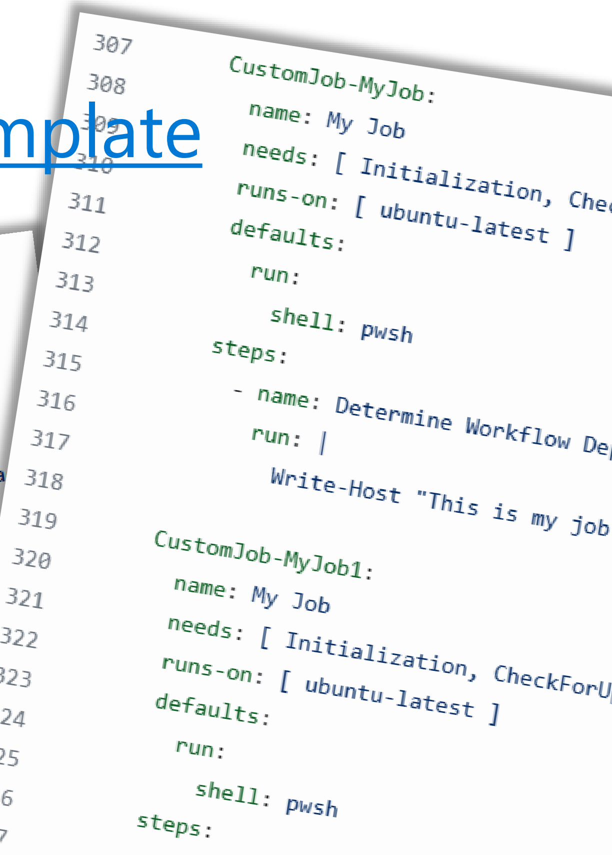
+ a few other fixes☺



```
- name: Read settings
  uses: freddydk/AL-Go/Actions/ReadSettings@custom
  with:
    shell: ${{ inputs.shell }}
    project: ${{ inputs.project }}
    get: useCompilerFolder,keyVaultCodesignCertificateName,doNotSignApps,artifa

- name: CustomStep Test1
  run: |
    Write-Host "test1 after settings"

- name: CustomStep Test2
```



```
  CustomJob-MyJob:
    name: My Job
    needs: [ Initialization, Chec
    runs-on: [ ubuntu-latest ]
    defaults:
      run:
        shell: pwsh
    steps:
      - name: Determine Workflow De
        run: |
          Write-Host "This is my job

  CustomJob-MyJob1:
    name: My Job
    needs: [ Initialization, CheckForU
    runs-on: [ ubuntu-latest ]
    defaults:
      run:
        shell: pwsh
    steps:
```

Thank you