Looking at the error in Image 3, the issue is clear:

\*\*ERROR: argument --package-file: expected one argument\*\*

The problem is that the `$packageFile` variable is not being set properly. Looking at your code, you're using:

- `$packageFile = $env:ZipPath` (line 5 in Image 1)

But this environment variable doesn't exist in the Classic Release Pipeline context. Here's the corrected code:

```powershell

# --- variables you normally set at the top ---

$batchAccount = "lvbatchdev"

$resourceGroup = "lv-batch-dev"

$appName = "ccdc-batch-jobs"

$version = "$(Build.BuildId)" # or use "latest"

# Find the ZIP file from the artifact

$dropFolder = "$(System.DefaultWorkingDirectory)/\_CCDC-Batch-Jobs-CI/drop"

$packageFile = Get-ChildItem -Path $dropFolder -Filter "\*.zip" | Select-Object -First 1

if ($null -eq $packageFile) {

Write-Error "No ZIP file found in $dropFolder"

exit 1

}

Write-Host "Found package file: $($packageFile.FullName)"

# 1 - point CLI at the correct Batch account

az batch account set `

--name $batchAccount `

--resource-group $resourceGroup

# 2 - (optional) create the application if it doesn't exist yet

$appExists = az batch application show `

--application-name $appName `

--name $batchAccount `

--resource-group $resourceGroup `

--only-show-errors `

--output none 2>$null

if ($LASTEXITCODE -ne 0) {

Write-Host "Application doesn't exist, creating..."

az batch application create `

--application-name $appName `

--name $batchAccount `

--resource-group $resourceGroup

}

# 3 - upload & activate the new package version

Write-Host "Creating application package..."

az batch application package create `

--application-name $appName `

--name $batchAccount `

--resource-group $resourceGroup `

--package-file $packageFile.FullName `

--version-name $version

if ($LASTEXITCODE -ne 0) {

Write-Error "Failed to create application package"

exit 1

}

# 4 - activate the package

Write-Host "Activating package..."

az batch application package activate `

--application-name $appName `

--name $batchAccount `

--resource-group $resourceGroup `

--version-name $version `

--format "zip"

# 5 - set as default version

Write-Host "Setting default version..."

az batch application set `

--application-name $appName `

--name $batchAccount `

--resource-group $resourceGroup `

--default-version $version

Write-Host "Application package registered successfully!"

```

The key changes:

1. \*\*Properly find the ZIP file\*\* from the artifact drop folder

2. \*\*Use the full path\*\* of the file with `$packageFile.FullName`

3. \*\*Check if the file exists\*\* before trying to use it

4. \*\*Use consistent parameter names\*\* (`--application-name` not `--application-id`)

This should fix the "expected one argument" error because now `--package-file` will receive the actual file path instead of an empty variable.​​​​​​​​​​​​​​​​