



# Automating Deployments with Azure DevOps

Erin Dempster  
5 Mar 2024

# Erin Dempster

She/Her

**Data Operations Team Lead**  
Trean Corporation



- SQL Server DBA
- Azure Administrator
- Azure DevOps Administrator

<https://www.erindempster.com>

@em\_dempster

<https://www.linkedin.com/in/erindempster>

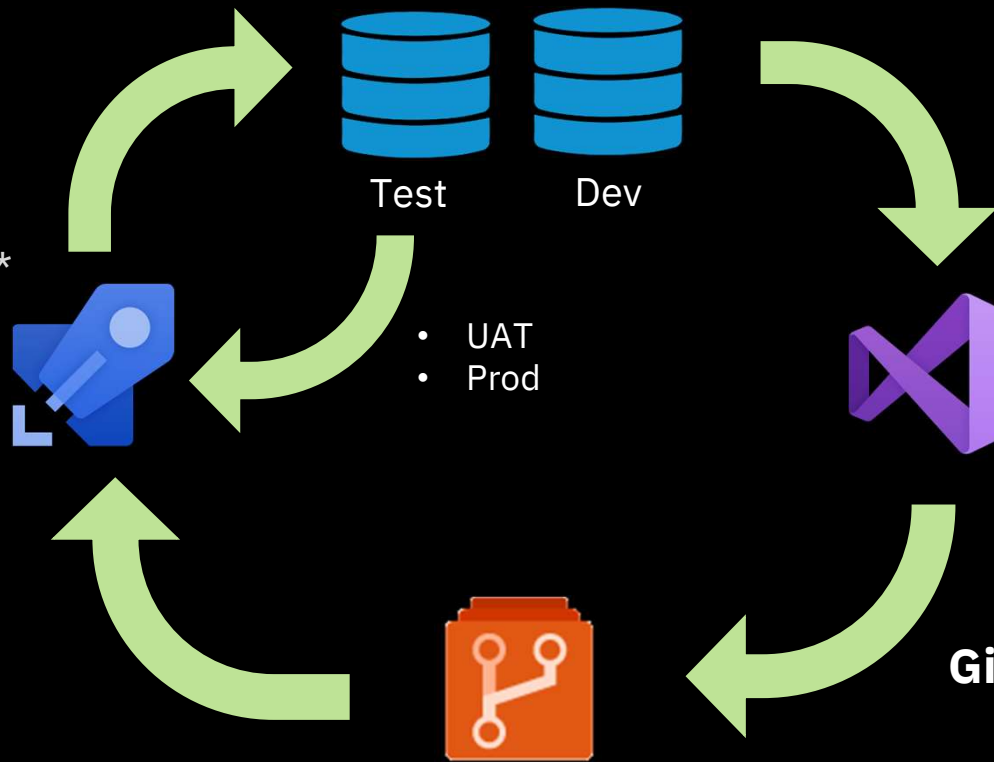
Speaker – PASS Summit, SQL Saturdays +

Author – SQLServerCentral.com

# Agenda

## Azure Pipelines

1. Build
2. Automated Tests\*
3. Approve
4. Deploy
5. Repeat 3&4



## Visual Studio

- Reverse Engineer
- Develop
- Unit Test
- Commit Changes

## Git Client

## Azure Repos

Git Repositories

# Sample Code

*Coming Soon*

<https://github.com/TrueNorthMN/DevOps>

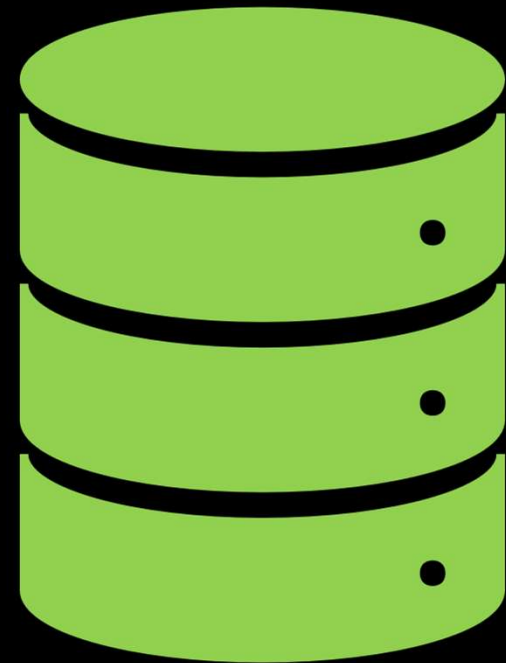


# Introduction to Visual Studio

SQL Database Project

# Visual Studio – SQL Database Project

- Long-Time Support for SQL Databases
- Build Projects with msbuild.exe
- Creates .DACPAC for Deployment
- Define Deployment Settings





# Visual Studio Installer

Modifying — Visual Studio Community 2022 — 17.8.1


Workloads Individual components Language packs Installation locations


## Gaming (2)


**Game development with Unity**  
Create 2D and 3D games with Unity, a powerful cross-platform development environment. ☐


**Game development with C++**  
Use the full power of C++ to build professional games powered by DirectX, Unreal, or Cocos2d. ☐


## Other Toolsets (5)

**Data storage and processing** ☒  
Connect, develop, and test data solutions with SQL Server, Azure Data Lake, or Hadoop.

**Data science and analytical applications** ☐  
Languages and tooling for creating data science applications, including Python and F#.

**Visual Studio extension development** ☐  
Create add-ons and extensions for Visual Studio, including new commands, code analyzers and tool windows.

**Office/SharePoint development** ☐  
Create Office and SharePoint add-ins, SharePoint solutions, and VSTO add-ins using C#, VB, and JavaScript.

**Linux and embedded development with C++** ☐

## Installation details

- Visual Studio core editor
- ▾ Data storage and processing
  - ▾ Optional
    - ☒ SQL Server Data Tools
    - ☒ Azure Data Lake and Stream Analytics Tools
    - ☒ .NET Framework 4.7.2 development tools
    - ☐ GitHub Copilot
    - ☐ F# desktop language support

# Demo

Visual Studio and Azure DevOps Overview

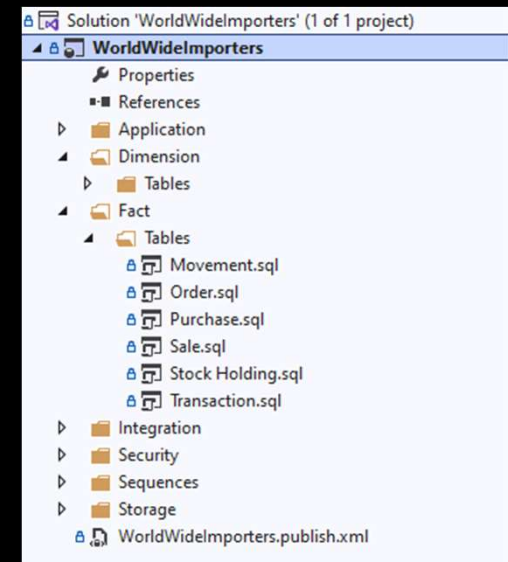




# Introduction to Azure Pipelines

# Azure Pipelines

- Manages code builds and deployments
- Triggered by source code check-ins



# Azure Pipelines

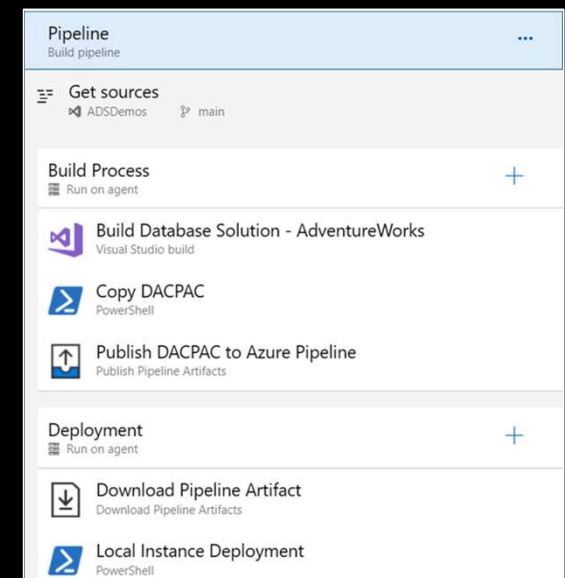
- Many pre-defined tasks
  - Visual Studio Build
  - Publish & Download Artifacts
- Scripting tasks
  - PowerShell
  - BASH

Define **YOUR** workflow in YAML

```
trigger:
- main

pool:
  vmImage: windows-latest

stages:
- stage: buildProcess
  jobs:
  - job: buildDACPAC
    displayName: Build Database DACPAC
    steps:
    - task: VSBUILD@1
      inputs:
        solution: '**\AdventureWorks.sln'
    - task: PublishPipelineArtifact@1
      inputs:
        targetPath: '$(Build.SourcesDirectory)\AdventureWorks'
        artifact: 'Databases'
        publishLocation: 'pipeline'
```



## Ways to Run Pipelines

- Azure-hosted Pipeline Agents
  - Great for building code/build validation
- Self-hosted Pipeline Agents
  - Preferred for deployments

## Pipeline Options

|         | Build Pipeline | Release Pipeline |
|---------|----------------|------------------|
| Classic | No*            | Yes              |
| YAML    | Yes            | Yes              |

Optimal Design - Multi-stage YAML Build and Release Pipeline

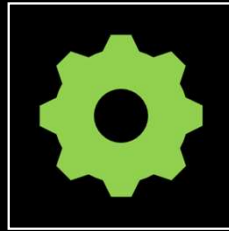
## Azure Pipelines

- YAML Schema - <https://tinyurl.com/4z7x3mkw>
- Task Index - <https://tinyurl.com/56vf436z>
- Pipeline Variables - <https://tinyurl.com/ye2ypvum>

# What is YAML?



“YAML Ain’t Markup  
Language”



Configuration Layout



Used by both Azure DevOps  
and GitHub Actions

# YAML - Azure Pipelines vs GitHub Actions

```
trigger:
- main

pool:
  vmImage: windows-latest

jobs:
- job: buildDACPAC
  displayName: Build Azure Data Studio DACPAC
  steps:
    - task: Settings
    - task: DotNetCoreCLI@2
      inputs:
        command: 'build'
        projects: '**/Adventureworks.sqlproj'
```

```
name: Build AdventureWorks

on:
  push:
    branches: [main]
  workflow_dispatch:

jobs:
  build:
    runs-on: windows-latest
    steps:
      - uses: actions/checkout@v3

      - name: Add MSBuild to PATH
        uses: microsoft/setup-msbuild@v1.0.2

      - name: Build
        working-directory: ${env.GITHUB_WORKSPACE}
        run: msbuild /m /p:Configuration=Release Adventureworks/Adventureworks.sqlproj
```



# Configuration Layout

## Hierarchical

<Pipeline>

Stages

Jobs

Steps

Task

# Basic Pipeline

```
trigger:
- main

pool:
  vmImage: windows-latest

steps:
- task: DotNetCoreCLI@2
  inputs:
    command: 'build'
    projects: '**/Adventureworks.sqlproj'
- task: SqlAzureDacpacDeployment@1
  inputs:
    azureSubscription: 'MVP-Community'
    AuthenticationType: 'servicePrincipal'
    ServerName: '...'
    DatabaseName: 'AdventureWorks'
    deployType: 'DacpacTask'
    DeploymentAction: 'Publish'
    DacpacFile: 'Adventureworks.dacpac'
    IpDetectionMethod: 'AutoDetect'
```

Implied Single Stage with one job

1<sup>st</sup> task – builds the project

2<sup>nd</sup> task – deploys the DACPAC using SQLPackage.exe

Both tasks run in the same agent instance

When to use this scenario?

- Proof of concept
- Testing connections
- Build Validation/Automated Testing

# Explicit Job Definitions

```
trigger:
- main

pool:
  vmImage: windows-latest

jobs:
- job: buildDACPAC
  displayName: Build Azure Data Studio DACPAC
  steps:
    Settings
    - task: DotNetCoreCLI@2
    inputs:
      command: 'build'
      projects: '**/Adventureworks.sqlproj'
- job: deployDACPAC
  displayName: Deploy DACPAC to Azure SQL DB
  steps:
    Settings
    - task: SqlAzureDacpacDeployment@1
    inputs:
      azureSubscription: 'MVP Community'
      AuthenticationType: 'servicePrincipal'
      ServerName: 'endlessreporting.database.windows.net'
      DatabaseName: 'AdventureWorks'
      deployType: 'DacpacTask'
      DeploymentAction: 'Publish'
      DacpacFile: 'Adventureworks.dacpac'
      IpDetectionMethod: 'AutoDetect'
```

Two Jobs each with one task

1<sup>st</sup> job – builds the project

2<sup>nd</sup> job – deploys the DACPAC using SQLPackage.exe

Each job runs in a separate agent instance

# Explicit Job Definitions

```
trigger:
- main

pool:
  vmImage: windows-latest

jobs:
- job: buildDACPAC
  displayName: Build Azure Data Studio DACPAC
  steps:
    - task: Settings
    - task: DotNetCoreCLI@2
      inputs:
        command: 'build'
        projects: '**/Adventureworks.sqlproj'
- job: deployDACPAC
  displayName: Deploy DACPAC to Azure SQL DB
  steps:
    - task: Settings
    - task: SqlAzureDacpacDeployment@1
      inputs:
        azureSubscription: 'MVP Community'
        AuthenticationType: 'servicePrincipal'
        ServerName: 'endlessreporting.database.windows.net'
        DatabaseName: 'AdventureWorks'
        deployType: 'DacpacTask'
        DeploymentAction: 'Publish'
        DacpacFile: 'Adventureworks.dacpac'
        IpDetectionMethod: 'AutoDetect'
```

There is a problem with this pipeline!

DACPAC is not shared between jobs

# Publish DACPAC as a Pipeline Artifact

```
- job: buildDACPAC
- displayName: Build Azure Data Studio DACPAC
- steps:
  Settings
  - task: DotNetCoreCLI@2
  - inputs:
    command: 'build'
    projects: '**/Adventureworks.sqlproj'
  Settings
  - task: PublishPipelineArtifact@1
  - inputs:
    targetPath: '$(Pipeline.Workspace)\AdventureWorks\bin\debug\Adventureworks.dacpac'
    artifact: 'Databases'
    publishLocation: 'pipeline'
```

← Artifacts

Published

| Name                    | Size  |
|-------------------------|-------|
| 📁 Databases             | 76 KB |
| 📄 AdventureWorks.dacpac | 76 KB |

# Explicit Stage Declaration

Errors 1

✖ Publishing to database 'AWTest' on server 'localhost'. Initializing deployment (Start) Initializing deployment (Failed) \*\*\* Changes to connecti...  
DeployDACPAC to SQL Server • SqlDacpacDeploymentOnMachineGroup

[View documentation for troubleshooting failed runs](#)

Jobs

| Name                         | Status  | Duration |
|------------------------------|---------|----------|
| ✔ Build Database DACPAC      | Success | ⌚ 45s    |
| ✖ DeployDACPAC to SQL Server | Failed  | ⌚ 25s    |

Stages Jobs

✔ buildProcess

1 job completed 1m 30s

📁 1 artifact

✖ deployProcess

Failed 36s

✖ DeployDACPAC to SQL Ser... 2...

Rerun failed jobs

Rerun all jobs

```
trigger:
- main

pool:
  vmImage: windows-latest

stages:
- stage: buildProcess
  jobs:
  - job: buildDACPAC
    displayName: Build Database DACPAC
    steps:

- stage: deployProcess
  jobs:
  - job: deployDACPACToTest
    displayName: DeployDACPAC to SQL Server
    pool: WorkshopPool
    steps:
```

## Agent Setup

- Define a Pipeline Agent Pool
- Generate a Personal Access Token (PAT)
- Install Agent
- Configure Agent

# Self-Hosted Pipeline Agent

```
Administrator: Command Prompt
C:\Agents>config

Azure Pipelines
agent v3.230.0 (commit 38c1c98)

>> Connect:
Enter server URL > https://dev.azure.com/
Enter authentication type (press enter for PAT) >
Enter personal access token > *****
Connecting to server ...

>> Register Agent:
Enter agent pool (press enter for default) > WorkshopPool
Enter agent name (press enter for WIN11TEMP) > VS 2022 VM
Scanning for tool capabilities.
Connecting to the server.
Successfully added the agent
Testing agent connection.
Enter work folder (press enter for _work) >
2024-02-05 14:03:46Z: Settings Saved.
Enter run agent as service? (Y/N) (press enter for N) > Y
Enter enable SERVICE SID TYPE UNRESTRICTED for agent service (Y/N) (press enter for N) >
Enter User account to use for the service (press enter for NT AUTHORITY\NETWORK SERVICE) >
Enter Password for the account azuread\erindempster > *****
Error reported in diagnostic logs. Please examine the log for more details.
- C:\Agents\diag\Agent_20240305-140252-utc.log
Granting file permissions to 'azuread\erindempster'.
Service vstsagent.endlessreporting.WorkshopPool.VS 2022 VM successfully installed
Service vstsagent.endlessreporting.WorkshopPool.VS 2022 VM successfully set recovery option
Service vstsagent.endlessreporting.WorkshopPool.VS 2022 VM successfully set to delayed auto start
Service vstsagent.endlessreporting.WorkshopPool.VS 2022 VM successfully configured
Enter whether to prevent service starting immediately after configuration is finished? (Y/N) (press enter for N) > Y
```





# Common Database Tasks

# Common Tasks for Databases

## Building DACPACs

- VSBuild@1 – Build Visual Studio project with MSBuild
- DotNetCoreCLI@2 – Build Azure Data Studio project

## Common Tasks for Databases

### Artifact Management

- PublishBuildArtifacts@1 – Publish Files to Pipeline
- DownloadPipelineArtifact@2 – Download Files from Pipeline

## Common Tasks for Databases

### Deployment Tasks

- SqlAzureDacpacDeployment@1 – Deploy to Azure SQL DB
- AzureCLI@2 – Execute PowerShell scripts in Azure context
- PowerShell@2 – Execute PowerShell scripts

# Deploy DACPAC with PowerShell

- Requires
  - SQLPackage.exe

```
$connectionString = "server=myserver;database=mydatabase;authentication=Active Directory MSI;
```

```
. 'C:\Program Files\Microsoft SQL Server\160\DAC\bin\sqlpackage.exe'`  
  /action:Publish`  
  /sourceFile:$($filePath)\$databaseName.dacpac`  
  /profile:$($filePath)\$databaseName.publish.xml`  
  /targetConnectionString:"$connectionString"
```

# Deploy DACPAC with PowerShell

- Requires
  - SQLPackage.exe

```
$connectionString = "server=myserver;database=mydatabase;authentication=Active Directory MSI; encrypt=yes;TrustServerCertificate=no"

$dacOutput = . 'C:\Program Files\Microsoft SQL Server\160\DAC\bin\sqlpackage.exe' `
    /action:Publish `
    /sourceFile:$($filePath)\$databaseName.dacpac `
    /profile:$($filePath)\$databaseName.publish.xml `
    /targetConnectionString:"$connectionString"

2>&1

$errors = @()
$logRemaining = $false

for ($i = 0; $i -lt $dacOutput.Count; $i++)
{
    #
    # SQLPackage.exe will fail on errors, so errors are towards the end of the output. Once errors start to appear,
    # grab the rest of the output lines to include additional information.
    #
    if ($dacOutput[$i].ToString().StartsWith("Error") -or $logRemaining -eq $true)
    {
        $errors += $dacOutput[$i]
        $logRemaining = $true
    }
}
```



# Logging Bugs with PowerShell

```
- task: AzureCLI@2
  displayName: Deploy Warehouse Database Changes
  inputs:
    scriptType: ps
    azureSubscription: $(azureSubscription)
    scriptPath: $(Build.SourcesDirectory)/PowerShell/DBDeployWith
    powerShellErrorActionPreference: continue
    arguments: -databaseName "Warehouse" -fullInstanceName $(SQLI
  env:
    AZURE_DEVOPS_EXT_PAT: $(System.AccessToken)
```

```
az extension add --name azure-devops
```

```
$assignBugTo = $env:Build_RequestedFor
```

```
az boards work-item create --title "Deployment Error for $databaseName in $environment"
  --type "Automation Bug"
  --project "$projectName"
  --assigned-to $assignBugTo
  --fields "Microsoft.VSTS.TCM.ReproSteps=$errorString"
  --organization "https://dev.azure.com/<myOrg>/"
```

 AUTOMATION BUG 8351\*

8351 Deployment Error for Warehouse in AutomatedTesting

 Erin Dempster

 3 Comments [Add Tag](#)

 Save

State  Closed

Area BlueMonkey

Reason  Moved to state Closed

Iteration BlueMonkey\Sprint 60

## Repro Steps

Error SQL72014: Framework Microsoft SqlClient Data Provider: Msg 207, Level 16, State 1, Procedure usp\_DimClaim, Line 111 Invalid column name 'ClaimsOfficeName'.

Error SQL72045: Script execution error. The executed script:  
CREATE PROCEDURE [Dim].[usp\_DimClaim]  
@PartnerID VARCHAR (10), @PartnerName VARCHAR (50), @StartDateTime

# Connection Strings

Entra ID (Azure AD) Managed Identity

```
$connectionString = "server=myserver;database=mydatabase;authentication=Active Directory MSI;
```

Windows Authentication

```
$connectionstring = "server=myserver; database=mydatabase; Integrated Security=SSPI;"
```



## Upcoming Sessions in Southeast US

- SQL Saturday Atlanta – April 19<sup>th</sup> & 20<sup>th</sup>
  - ½ Day Pre-Con – Automating Deployments with Azure DevOps
    - Friday 8 – noon
  - General Session – Intro to Automated Deployments with GitHub
- SQL Saturday Jacksonville – May 4<sup>th</sup>
  - Intro to Automated Deployments with Azure DevOps
  - Intro to Automated Deployments with GitHub