

NestedFlow Automation– Initial Setups

Creation Date: 9/29/2023

Table Of Contents

Introduction	1
Accessibility Setups	2
Web Automation Setups	4
GitHub Setups	5
SQL Definitions	10
DataBase Steps	25
Code Setups	30

Introduction

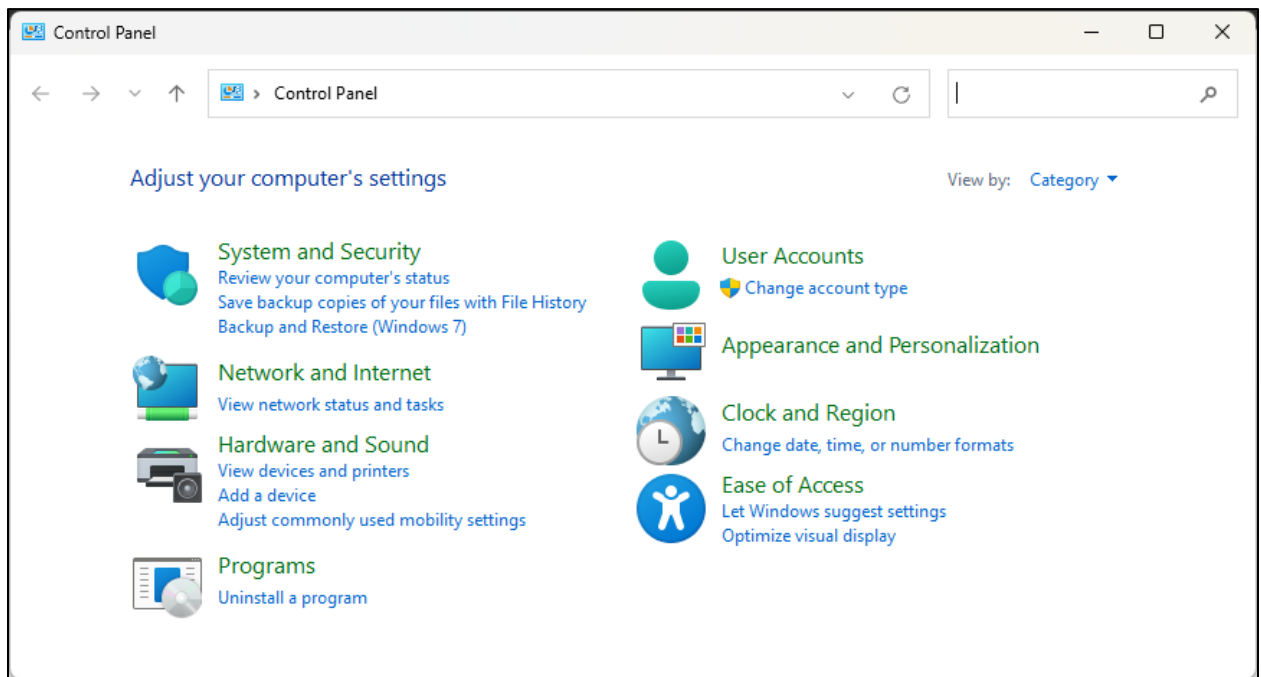
This document depicts in detail the initial setups needed to use the NestedFlow Automation tool effectively. The setups involve:

- Accessibility Setups
- Web Automation Setups
- GitHub Setups (Optional)
- SQL Definitions (Optional)
- Database Setups (Optional)
- Code Setups

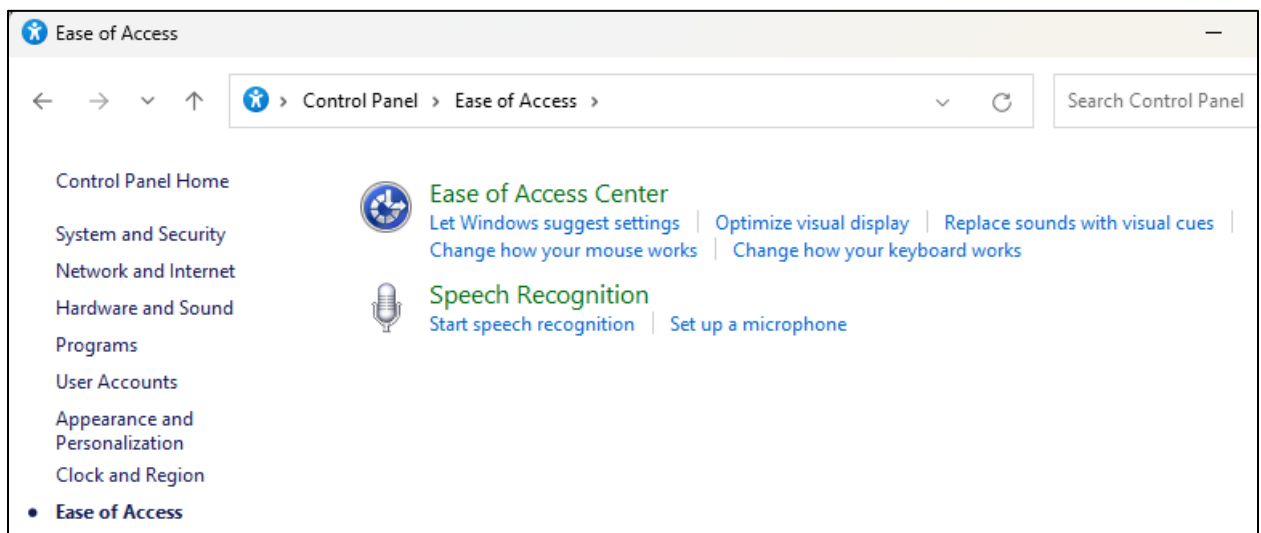
Accessibility Setups

Java legacy desktop applications are automated using Java Access Bridge technology. For enabling that we need to perform the below steps

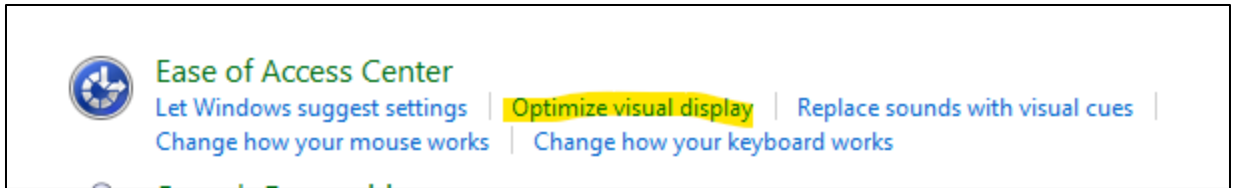
- Navigate to Control Panel



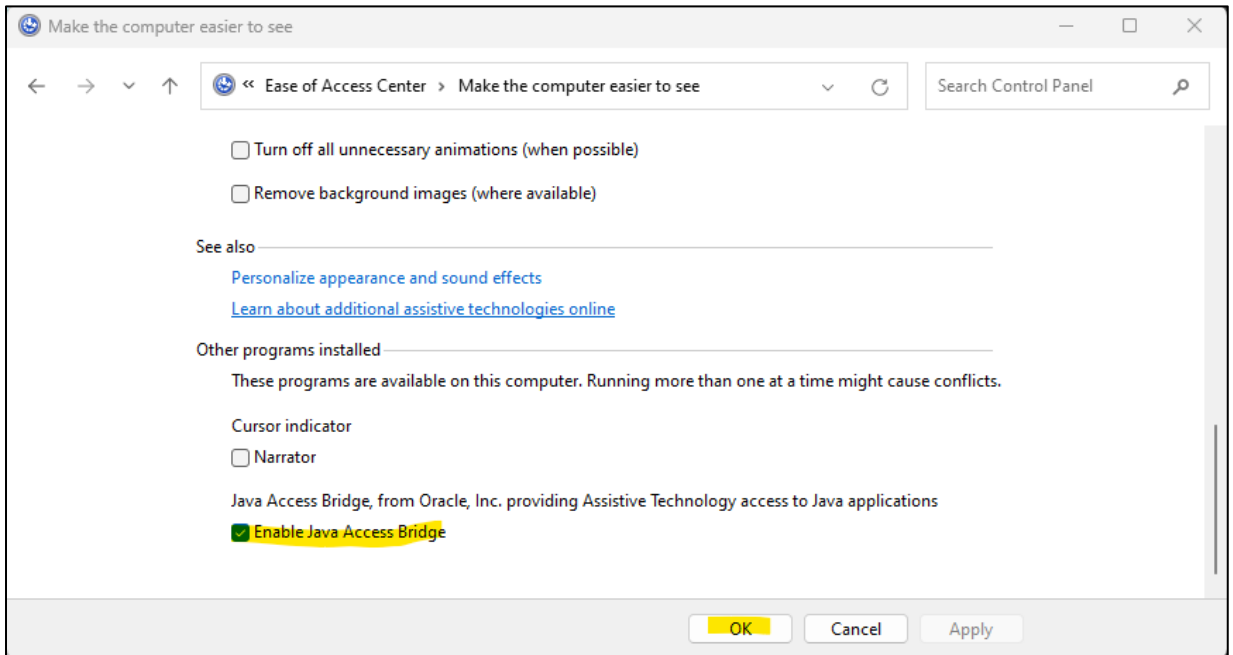
- Click **Ease of Access**



- Click on **Optimal visual display**



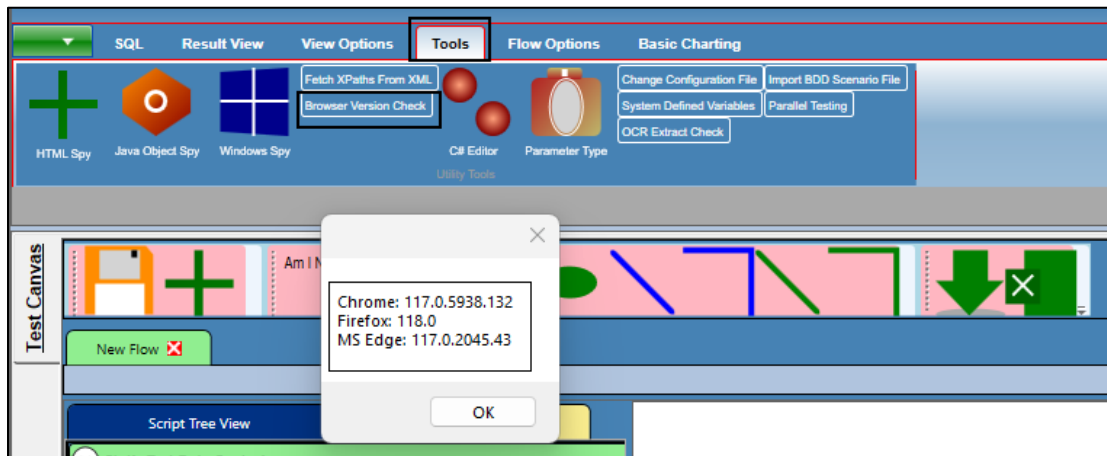
- Scroll down to the bottom of the page and Check “**Enable Java Access Bridge**” and click on **OK** button



Web Automation Setups

NestedFlow Automation tool requires webdriver exe files for automating the HTML UI elements. We must download chromedriver.exe, msedgedriver.exe, IEDriverServer.exe and geckodriver.exe matching the current browser configuration on the machine.





- Open the NestedFlow Automation tool. Navigate to Tools menu and click on **Browser Version Check** menu option:



- Navigate to below websites to download webdriver exes
 - Edge: [Microsoft Edge WebDriver - Microsoft Edge Developer](#)
 - Firefox: [Releases · mozilla/geckodriver \(github.com\)](#)
 - Chrome: [ChromeDriver - WebDriver for Chrome - Downloads \(chromium.org\)](#)
 - If you are using [Chrome version 115 or newer](#), please consult the Chrome for Testing availability dashboard. This page provides convenient JSON endpoints for specific ChromeDriver version downloading.

<https://github.com/GoogleChromeLabs/chrome-for-testing#json-api-endpoints>

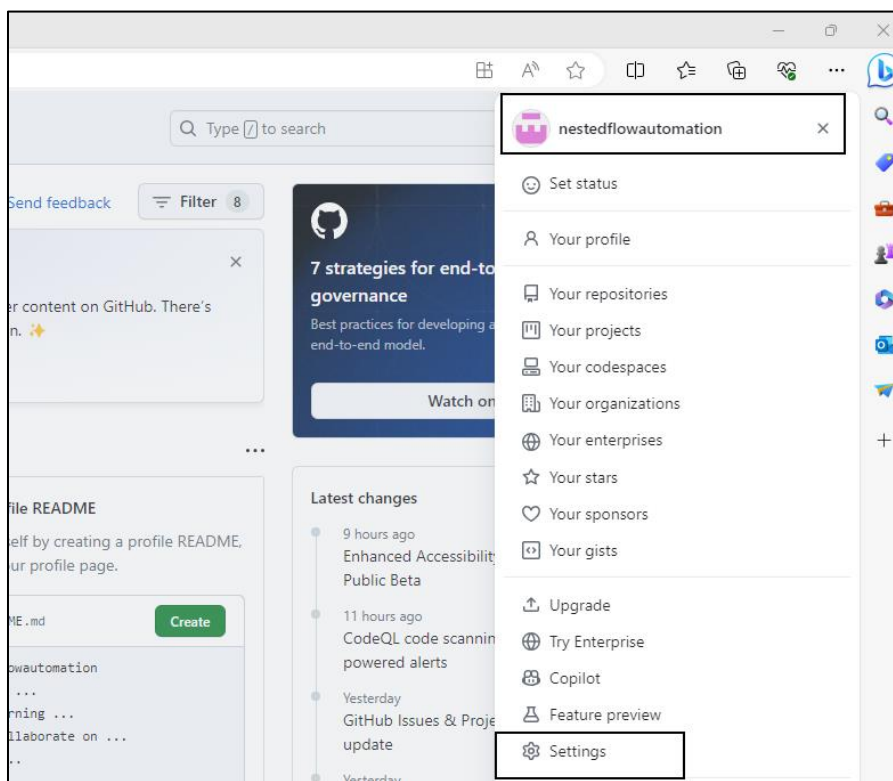
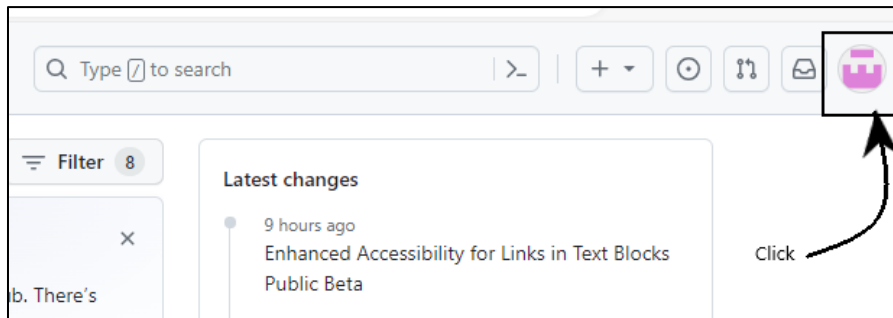
- Download the exes and place in %APPDATA%/one/webdrivers

This PC > Windows (C:) > Users > paile > AppData > Roaming > One > WebDrivers				
<input type="checkbox"/> Name	Date modified	Type	Size	
 chromedriver.exe	8/22/2023 2:00 PM	Application	12,106 KB	
 geckodriver.exe	4/6/2022 1:53 AM	Application	3,652 KB	
 IEDriverServer.exe	12/2/2021 3:32 PM	Application	2,986 KB	
 msedgedriver.exe	9/22/2023 8:32 AM	Application	13,696 KB	

GitHub Setups

This is an optional setup. This is needed only if the test cases are needed to be hosted in GitHub. It requires the GitHub Token to be added in configuration file.

- Navigate to your Github page and login
- Navigate to settings



- Click on the **Developer Settings** in the bottom of the screen

- Click on **Personal access tokens**

- Click on **Tokens (Classic)**

- Choose **Generate new token (classic)**

- If it asks for credentials enter

Confirm access

Signed in as

@nestedflowautomation

Password

[Forgot password?](#)

Confirm

Tip: You are entering [sudo mode](#). After you've performed a sudo-protected action, you'll only be asked to re-authenticate again after a few hours of inactivity.

- Enter the Name for the token

- GitHub Apps
- OAuth Apps
- Personal access tokens
 - Fine-grained tokens
 - Tokens (classic)**

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

GitHubAccess

What's the token for?

Expiration *

30 days

The token will expire on Sun, Oct 29 2023

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo:deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> manage_runners:org	Manage org runners and runner groups
<input type="checkbox"/> admin:public_key	Full control of user public keys

- Set Expiration to suitable value and check all possible checkboxes (both read and write rights are needed for the feature to work)

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

GitHubAccess

What's this token for?

Expiration

No expiration (selected) | The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> writepackages	Upload packages to GitHub Package Registry
<input type="checkbox"/> readpackages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> deletepackages	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> writeorg	Read and write org and team membership, read and write org projects
<input type="checkbox"/> readorg	Read org and team membership, read org projects
<input type="checkbox"/> manage_runnersorg	Manage org runners and runner groups
<input checked="" type="checkbox"/> admin:public_key	Full control of user public keys

- Click on Generate Token

☒ write:ssh_signing_key

☒ read:ssh_signing_key

Generate token **Cancel**

Note: Organization Github setups will also give option to configure SSO in the token setup without which it will not work

- Click on Copy token icon next to the token and it will copy to clipboard

Personal access tokens (classic)

Generate new token | Revoke all

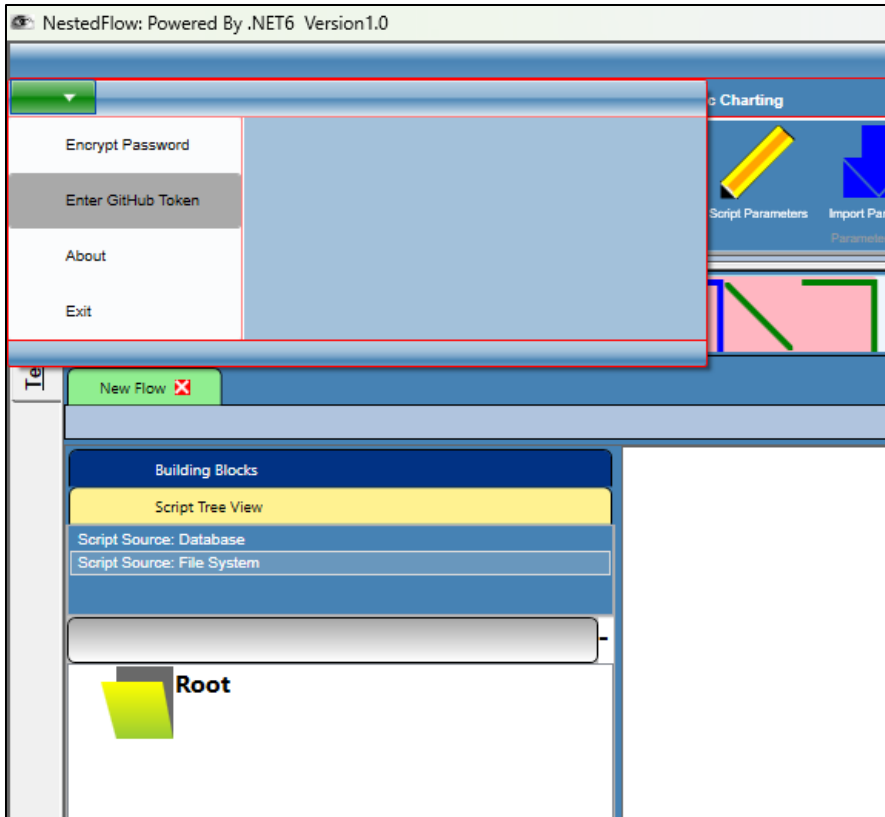
Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

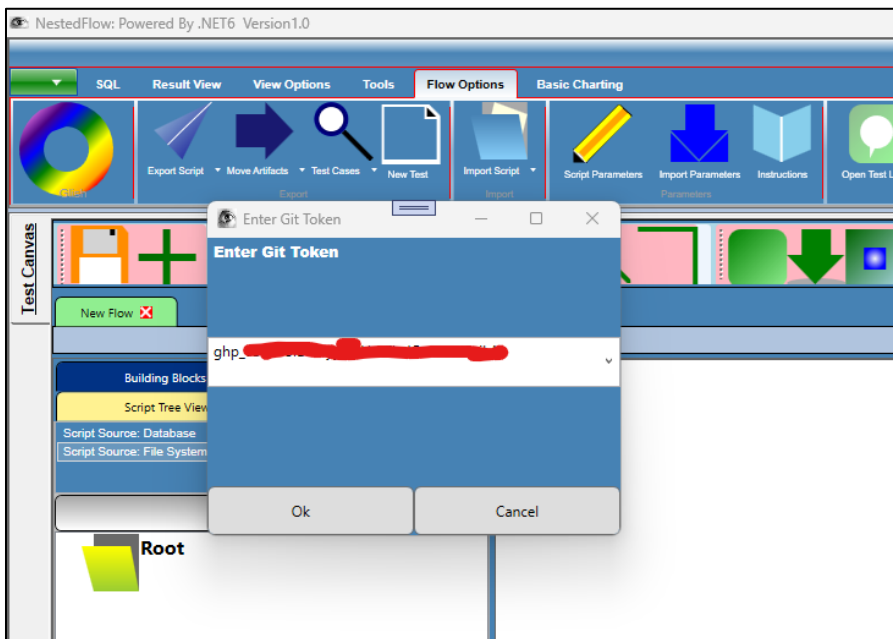
✓ ghp_...	Delete
GHTOKEN — admin:enterprise, admin:ssh_key, admin:org, admin:org_hook, admin:public_key, Last used within the last 2 weeks	Delete
admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, copilot, deletepackages, delete_repo, gist, notifications, project, repo, user, workflow, write:discussion, writepackages	
⚠ This token has no expiration date.	

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

- Open NestedFlow Automation tool. In the Ribbon menu choose Enter Github Token



- Enter Token and click Ok button



This step will update configuration file with the token

SQL Definitions

Tool allows the scripts/object files and batch files to be stored in MS SQL database. If you opt for that, you need to create a schema and below tables to make it work

- Create Schema:

Creates schema needed to hold all the tables

```
CREATE schema NESTEDFLOWAUTOMATION authorization dbo;
```

- Create NewDataSet Table

Created for future enhancement purposes

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[newdataset]
(
    [SetName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [TableName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Creation_Date] DATETIME NULL
, [LastUpdate_Date] DATETIME NULL
, [CreatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [UpdatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Active] VARCHAR(1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create NestedFlowFlowAttributes table

Stores test step images. Used to create test case knowledge document

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowFlowAttributes]
(
    [FlowName] VARCHAR(3000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [StepName] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [BitMap] VARBINARY(MAX) NULL
);
```

- Create NestedFlowMailingList table

Created for future enhancement purposes

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[OneRoofMailingList]
(
    [mailID] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [mailType] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [creation_date] DATETIME NULL
, [Createdby] VARCHAR(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create NestedFlowMailingData table

Created for future enhancement purposes

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowMailingData]
(
    [mailID] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [MailDisp] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Host] VARCHAR(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Port] INT NULL
);
```

- Create NestedFlowgoldcopyTable table

Useful to generate custom datasets which can be used for testing

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowgoldcopytable]
(
    [DataSetName] VARCHAR(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [DataSetID] VARCHAR(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Active] VARCHAR(1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Creation_Date] DATETIME NULL
, [Created_by] VARCHAR(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [last_updated_by] VARCHAR(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [last_update_date] DATETIME NULL
);
```

Create index on it

```
CREATE NONCLUSTERED INDEX [NestedFlowgoldcopyindex] ON
[NESTEDFLOWAUTOMATION].[NestedFlowgoldcopytable] ([DataSetID] ASC)
```

- Create NestedFlowgoldcopyseeddata table

Useful to generate custom datasets which can be used for testing

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowgoldcopyseeddata]
(
    [DataSetID] VARCHAR(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute1] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute2] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute3] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute4] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute5] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute6] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute7] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute8] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute9] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute10] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute11] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute12] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute13] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute14] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute15] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute16] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute17] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute18] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute19] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute20] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute21] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute22] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute23] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute24] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute25] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute26] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute27] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute28] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute29] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute30] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attribute31] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

, [ATTRIBUTE459] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE460] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE461] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE462] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE463] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE464] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE465] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE466] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE467] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE468] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE469] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE470] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE471] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE472] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE473] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE474] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE475] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE476] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE477] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE478] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE479] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE480] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE481] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE482] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE483] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE484] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE485] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE486] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE487] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE488] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE489] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE490] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE491] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE492] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE493] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE494] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE495] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE496] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE497] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE498] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE499] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ATTRIBUTE500] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
)
;

```

- Create NestedFlowseedatacolumnMap table
Stores custom dataset definitions

```

CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowseedatacolumnMap]
(
    [DataSetID] VARCHAR(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Attrname] VARCHAR(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ColumnName] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);

```

- Create NestedFlowExecutionRecords table
Created for future enhancement purposes

```

CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowExecutionRecords]

```



```
(
    [RunID] VARCHAR(150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [RunName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [seqid] INT NULL
, [LogType] VARCHAR(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [LogDate] VARCHAR(150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [LogText] VARCHAR(MAX) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create NestedFlowExecution table
Created for future enhancement purposes

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NextedFlowExecution]
(
    [RunID] VARCHAR(150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [RunName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [RunBy] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ScriptName] VARCHAR(5000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [startdate] VARCHAR(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [enddate] VARCHAR(100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create NestedFlowDirectory table

Manages Directory tree

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowDirectory]
(
    [Level] INT NULL
, [Parent] VARCHAR(4000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [FileID] VARCHAR(4000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Name] VARCHAR(4000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [NodeType] VARCHAR(4000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [UniqueStr] VARCHAR(7000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Checkedout] VARCHAR(20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [CheckedoutBy] VARCHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create NestedFlowDirectoryHelp table

Manages Directory tree

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowDirectoryHelp]
(
    [HelperName] VARCHAR(5000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [HelperText] XML NULL
);
```

- Create NestedFlowDBKeys table
Currently unused. Created for future enhancement purposes

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowDBKeys]
(
    [dbUser] VARCHAR(500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [dbPass] VARCHAR(500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create NestedFlowdotNetScript table
Stores C# scripts

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowdotNetScript]
(
    [ScriptID] INT NULL
    , [ScriptName] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [ScriptMode] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [ScriptType] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [ExecOptions] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [ScriptText] VARCHAR(MAX) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [ExtraReferences] VARCHAR(MAX) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [ExtraCS] VARCHAR(MAX) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [Parameters] VARCHAR(MAX) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [Creation_Date] DATETIME NULL
    , [LastUpdate_Date] DATETIME NULL
    , [CreatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [UpdatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create NestedFlowdataSet table
Created for future enhancement purposes

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowDataSet]
(
    [SetName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [TableName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [Creation_Date] DATETIME NULL
    , [LastUpdate_Date] DATETIME NULL
    , [CreatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [UpdatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [Active] VARCHAR(1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create NestedFlowNestedFlowObjectInfo table
Used to store Automation object information optionally

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowObjectInfo]
(
    [PageName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [PageText] XML NULL
    , [Creation_Date] DATETIME NULL
    , [LastUpdate_Date] DATETIME NULL
    , [CreatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [UpdatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create NestedFlowFlowInfo table
Stores Canvas test flows

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowFlowInfo]
(
    [FlowName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [FlowXML] XML NULL
    , [Creation_Date] DATETIME NULL
    , [LastUpdate_Date] DATETIME NULL
    , [CreatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [UpdatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create NestedFlowBatchExecList table
Stores serial batch execution information

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowBatchExecList]
(
    [FlowName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [FlowXML] XML NULL
, [Creation_Date] DATETIME NULL
, [LastUpdate_Date] DATETIME NULL
, [CreatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [UpdatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create UserInfo table
Created for user access management if needed

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[UserInfo]
(
    [UserName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ActiveFlag] VARCHAR(10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Creation_Date] DATETIME NULL
, [LastUpdate_Date] DATETIME NULL
, [CreatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [UpdatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create RoleInfo table
Stores allowed user roles

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[RoleInfo]
(
    [RoleName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Creation_Date] DATETIME NULL
, [LastUpdate_Date] DATETIME NULL
, [CreatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [UpdatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create UserRole table
Stores user level roles

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[UserRole]
(
    [UserName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [RoleName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [ActiveFlag] VARCHAR(10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [Creation_Date] DATETIME NULL
, [LastUpdate_Date] DATETIME NULL
, [CreatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
, [UpdatedBy] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create NestedFlowFlowExecInfo table
Created for future enhancement purposes


```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowFlowExecInfo]
(
    [OrigFlowName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [FlowName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [FlowXML] XML NULL
    , [CreationDate] DATETIME NULL
    , [LastUpdateDate] DATETIME NULL
    , [CreatedBy] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [UpdatedBy] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [status] VARCHAR(4000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

Creating Index for the table

```
CREATE NONCLUSTERED INDEX [idx_lastname] ON
[NESTEDFLOWAUTOMATION].[NestedFlowFlowExecInfo] ([FlowName] ASC);
```

- Create NestedFlowFlowInfoExec table
Created for future enhancement purposes

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowFlowInfoExec]
(
    [FlowName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [UserName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [CreationDate] DATETIME NULL
    , [LastUpdateDate] DATETIME NULL
    , [CreatedBy] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [UpdatedBy] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create NestedFlowWebActions table
Created for future enhancement purposes

```
CREATE TABLE [NESTEDFLOWAUTOMATION].[NestedFlowWebActions]
(
    [ActionIndex] INT NULL
    , [username] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [CreationDate] DATETIME NULL
    , [ReferBy] VARCHAR(50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [ReferText] VARCHAR(2500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [ActionName] VARCHAR(2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
    , [Actionval] CHAR(1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
);
```

- Create Required Sequences

```
CREATE SEQUENCE NESTEDFLOWAUTOMATION.LogSequence
    START WITH 1
    INCREMENT BY 1;

CREATE SEQUENCE NESTEDFLOWAUTOMATION.ScriptSequence
```

```
START WITH 1
INCREMENT BY 1;

CREATE SEQUENCE NESTEDFLOWAUTOMATION.ActionSequence
START WITH 1
INCREMENT BY 1;

create Sequence NESTEDFLOWAUTOMATION.SSNSEQ
start with 1
MINVALUE 1
MAXVALUE 2860816
increment by 1;
```

DataBase Steps

We need to update configuration file for enabling the tool to connect to MS SQL database in which the above objects are created. This is done to store entire test artifacts in the database which will help us enhance the test storage/execution/deployment activities. This is optional and can be skipped if the database option is not available. Because tool supports test cases in

- Local Windows machine
- Database
- GitHub

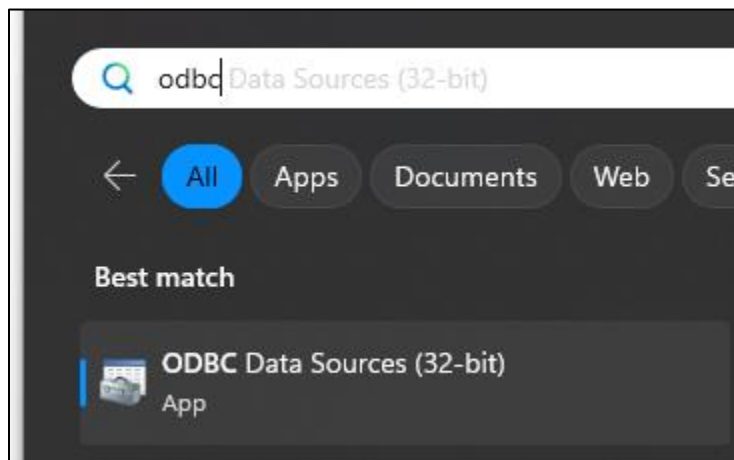
For documenting this we have created a locally hosted SQL server (but tool can connect to remote MS SQL too)

Connection details:

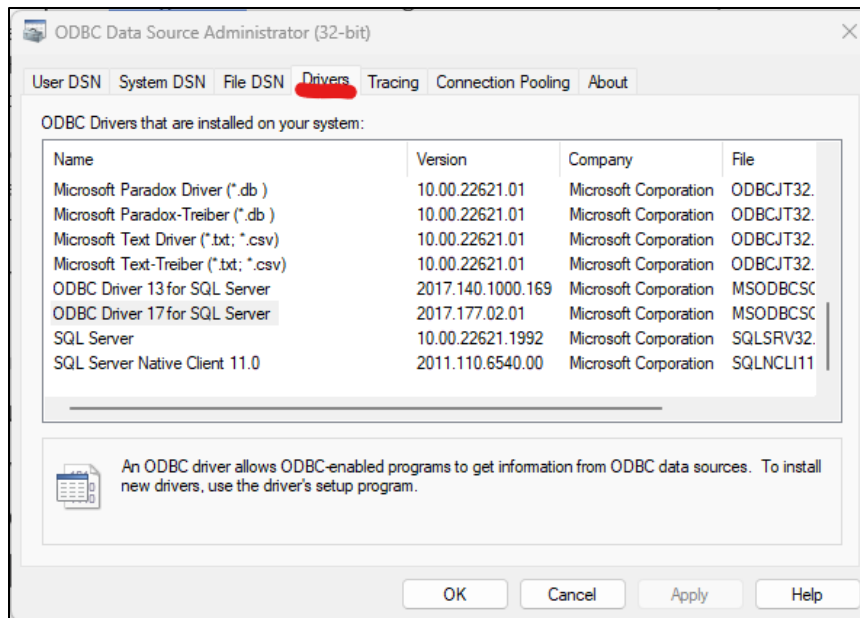
Schema: NESTEDFLOWAUTOMATION

ODBC Driver:

Search **ODBC Data Sources (32-bit)** in the windows programs and open it



Navigate to Drivers Tab



In this case the right driver is {ODBC Driver 17 for SQL Server}

Tree Mode: Script Source: Database

Database Name: Name of the database where the schema is created

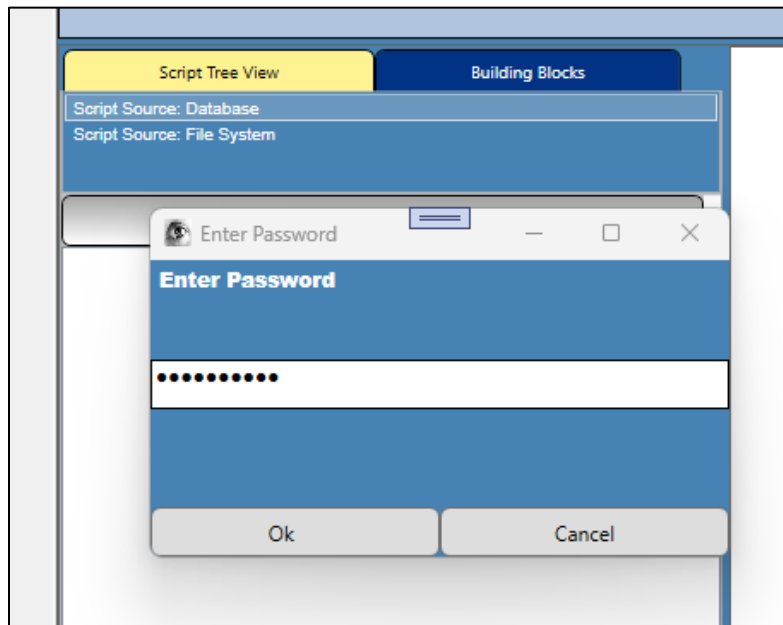
DBUser: Blank if windows authentication is enabled

DBVault (Password): Blank if windows authentication is enabled

Navigate to %APPDATA%/one and open AppConfig.xml and perform the changes



If you provide DBUser (When you don't want the windows authentication to work but use another generic user) , and leave the password blank, As soon as you login password entry will be prompted

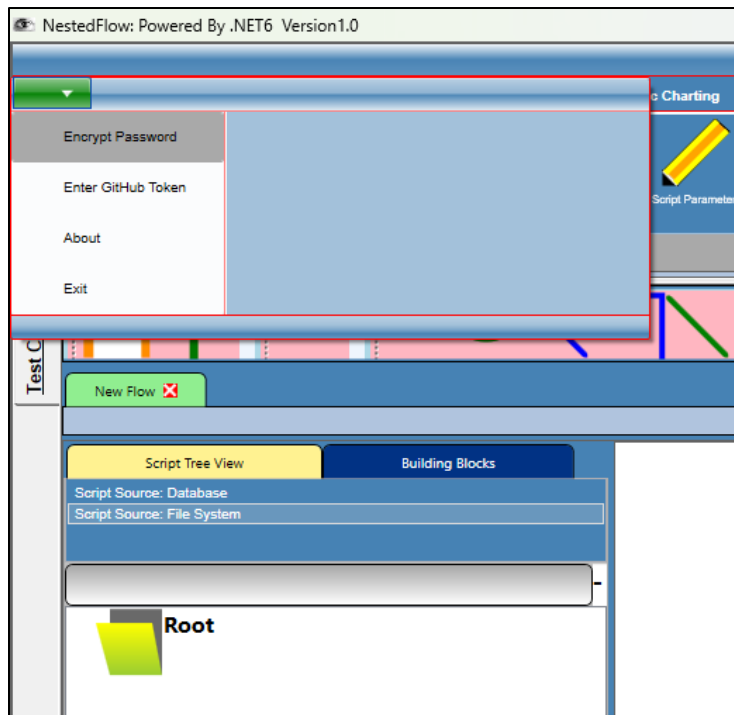


If you enter right password and click on **Ok** script tree will be displayed

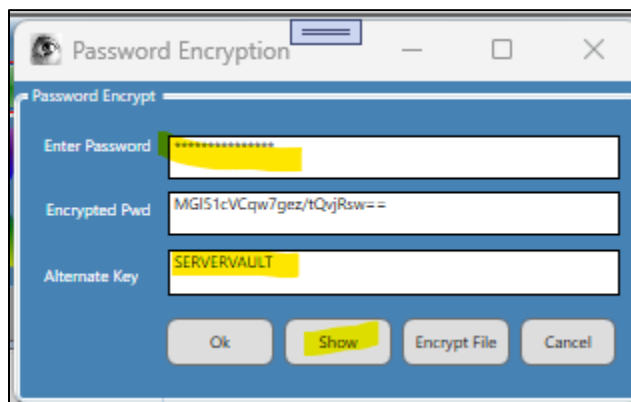
You can also provide encrypted DB password in the config file if you don't want the Enter Password window to pop up every time. For that, create a new tag DBVault in the appconfig.xml

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NET CORE,Version=v6" />
    <ToolVersion Version="1.0" />
    <Tree Mode="Script Source: Database" />
    <BasePath Path="" />
    <ALM Link="" />
    <LOG Path="C:\Users\paile\AppData\Roaming\One\Logs" />
    <OBJ Path="C:\Users\paile\AppData\Roaming\One\Objects" />
    <Repository Path="C:\Users\paile\AppData\Roaming\One\" />
    <LastScript></LastScript>
    <Favourites>Code,MERGE,INLINEFLOW</Favourites>
    <Firefox Path="" />
    <DBServer Server="LAPTOP-BVO747DM\SQLEXPRESS" />
    <DBUser User="xuser" />
    <DBName Name="Master" />
    <Schema Name="NESTEDFLOW." />
    <DBDriver Driver="{SQL Server}" />
    <DBVault vault="" />
    <GitHubKey key="" />
  </startup>
</configuration>
```

Invoke **Encrypt Password** menu in ribbon controls



Enter the Password. Enter Alternate Key = **SERVERVAULT** and click on **Show**.



Encrypted Pwd field will display the encrypted password which can be entered in the config file

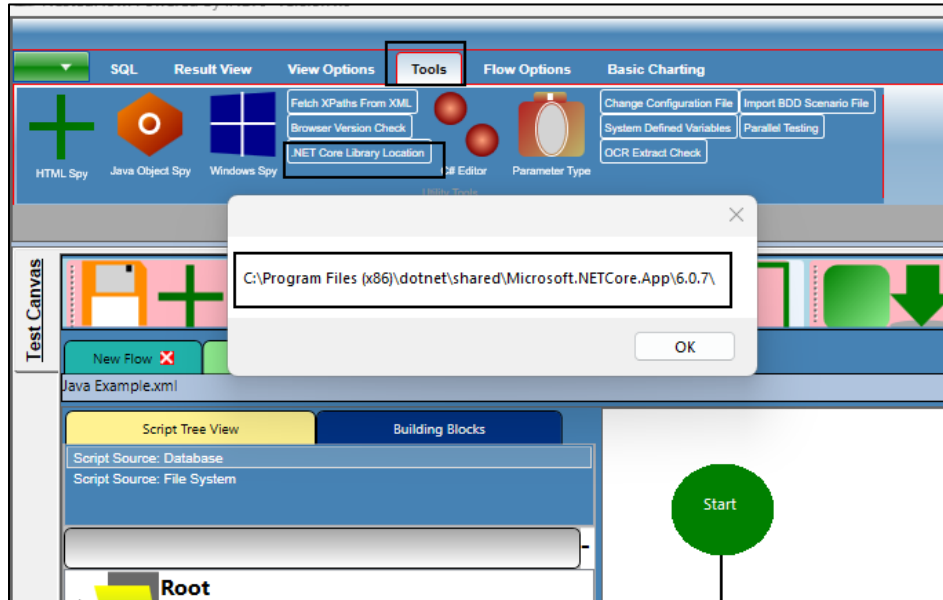
```

<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NET CORE,Version=v6" />
    <ToolVersion Version="1.0" />
    <Tree Mode="Script Source: Database" />
    <BasePath Path="" />
    <ALM Link="" />
    <LOG Path="C:\Users\paile\AppData\Roaming\One\Logs" />
    <OBJ Path="C:\Users\paile\AppData\Roaming\One\Objects" />
    <Repository Path="C:\Users\paile\AppData\Roaming\One\" />
    <LastScript></LastScript>
    <Favourites>Code,MERGE,INLINEFLOW</Favourites>
    <Firefox Path="" />
    <DBServer Server="LAPTOP-BVO747DM\SQLEXPRESS" />
    <DBUser User="xuser" />
    <DBName Name="Master" />
    <Schema Name="NESTEDFLOW." />
    <DBDriver Driver="{SQL Server}" />
    <DBVault vault="MG151cVCqw7gez/tQvjRsw==" />
    <GitHubKey key="" />
  </startup>
</configuration>

```

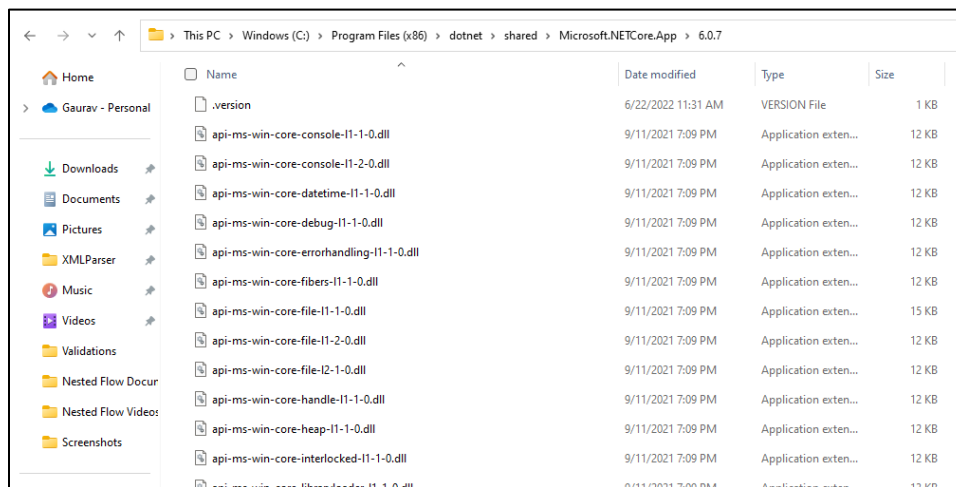
Code Setups

NestedFlow Automation tool allows custom scripting in C# to address corner cases which are not possible to perform in scriptless way. The code follows the .NET Core standards and you can find the reference location of the libraries from:



Navigate to Tools → .NET Core Library Location

Copy All the files from that location to %APPDATA%/One/CoreApp



Note: We can as well use the standard location but If you need to use external or custom libraries you cannot use locations which only admins have access to in a machine

Add all DLLs in the NestedFlow Automation tool's binary location in Program Files (86) should be added to %APPDATA%/One/CoreApp

All the custom DLLs you are using for the scripting should also be added to the location %APPDATA%/One/CoreApp