# NAV-RTC-Source-Control

Microsoft Business Solutions NAV Source Control Manager. Built in CAL with GIT as the Source Control Back end.

# WHY?

I created this project because the need in my org existed to have source control for nav developers working on the same database. Typically, NAV developers do not use source control and are not familiar with the tools. I created this tool to assist NAV Developers to use basic Source control functionality in a familiar environment. I was looking around on the internet but could not find one that is free to use and does what I need it to do, so I created my own.

# PLAY NICE

I am providing it “as is” and do not guarantee regular updates on this project. I will help where I can if there are any questions, but please do not expect fast reaction on questions. Please feel free to pull and use the code. Please do not try to Sell it or make a profit off it. You got it free, use it free. If you make any improvements on this, feel free to send me a pull request. It would be nice to make this better for everyone.

# WHAT YOU GET

### GitCmdLet.psm1

Custom Created CmdLet specific for this project containing a mix of NAV, GIT and Home baked PowerShell Commands.

### SourceControlMonitor.sql

SQL Script to Setup Triggers Against the Object Table for Logging any object changes in Navision

### Sample SourceControlSetup.xlsx

Excel file with sample Setup data

### “Source Control NAV RTC.fob” and “Source Control NAV RTC.txt”

Navision object and Text Files with all the source for this tool.

The Included objects Are:

#### Table77777 “SourceControlSetup”

Setup Table with Repo Info, PowerShell Script Locations, Text Editor Path and Compare Tool Path

#### Table77778 “SourceControlMonitor”

Table containing Changes of Objects in the Database

#### Page77775 “Source Control List Part”

The Dashboard List Part Page – Used on Dev Dash

#### Page77776 “Development Dash”

The Development Dashboard – Can be used as a dash and performs same functions as Page77777

#### Page77777 “Source Control”

The Main Application Page – Run This After Setup

#### Page77778 “Source Control Monitor Lines”

The List Page for the Monitoring Table to view the changes

#### Page77779 “SourceControl Setup”

Setup Page to prepare your database for source control

# INSTALL

### Prerequisites

1. If you don’t have it, Install Git for Windows from here: <https://git-scm.com/download/win>
2. Make sure you have a Service tier for your version of your Development database on your local machine. This is technically not required, and you can get away with just having the required PowerShell scripts on your local PC somewhere.
3. Make sure you have the PowershellRunner as an add-in in your RoleTailored client Add-ins folder… typically located here: C:\Program Files (x86)\Microsoft Dynamics NAV\90\RoleTailored Client\Add-ins\PowerShellRunner

If not, copy it from your service tier Add-ins folder. (This Add-in ships with NAV)

1. Make Sure this Add-in is active in your Client under Control Add-ins.
2. Create a Location on your computer as the Parent Directory where your source will be stored. This location will be referred to as **{SourceParentPath}** from here on.

(This readme assumes you already have a GIT repository somewhere. It is explained from the perspective of a new user setting up source control environment on his machine and cloning the existing upstream repo to his local machine. If you do not have a source control repo yet, create one anywhere (github or Devops are good options) and upload your nav objects as text files… using the NAV-ObjectSplitter. This repo will act as your upstream origin for everything further in this document).

### Step 1

Clone this repository to your local computer and extract the files inside.

### Step 2

Copy the 2 Files “GitCmdLet.psm1” and “SourceControlMonitor.sql” to a Location on your computer and save the path for later use. This location will be referred to as **{ScriptPath}** from here on.

### Step 3

Import “Source Control NAV RTC.fob” into your Development database and make sure all objects compile.

# SETUP

## Step 1

Open SQL Server Management Studio and Run the Script “SourceControlMonitor.sql” (which you stored in the {ScriptPath}) against your Development database. You only need to do this once. This script creates triggers against the Object Table to detect changes and then writes them away in the Source Control Monitor Table in Nav. – I’d like to thank Andew Dixon here as I got this script from Mibuso where he uploaded it (And Made some small tweaks of my own). (<https://mibuso.com/downloads/navision-audit-sql-script>)

## Step 2

In NAV Run Page 77779 – “SourceControl Setup” and setup your environment.

In this Repo is an excel File: “Sample SourceControlSetup.xlsx”. This is my setup data and you can use that as reference with real world data.

### How to populate the setup Table:

### User\_ID

This field is Auto populated with the logged in user

### Type

This field consists of the following types:

1. Repo: Select this when specifying a repository location. You can have more than one Repo for different source locations and branches.
2. PS Module: Select this when specifying the locations of the different PowerShell Scripts.
3. Text Editor: Select this when specifying the location of your favourite text editor.
4. Compare Tool: Select this when specifying the location of your favourite compare tool.

For the “PS Module” Type setup lines, you need 3 mandatory scripts setup:

1. C:\Program Files\Microsoft Dynamics NAV\90\Service\NavAdminTool.ps1
2. C:\Program Files (x86)\Microsoft Dynamics NAV\90\RoleTailored Client\Microsoft.Dynamics.Nav.Model.Tools.psd1
3. **{ScriptPath}** \GitCmdLet.psm1

Please Note that the path of 1 and 2 points to the nav install directories… make sure you use the correct version… i.e 80/90/100 etc.

### Name

Unique Descriptive Name for the record you are filling in

### Local Repo Path

The Local Path of the Repo / Script or Editor

For Repo’s…., if it’s the First Time you are setting up the repo on your local machine (i.e. you don’t have a local path that contains the source of your repo), fill in

“{SourceParentPath}\[Remote Repo name]\” here. (See remote repo name further down). Even if ~\[Remote Repo name]\ part of this path does not exist yet, it will be created on first run.

### Local Project Path (Repo Type Only)

The Local Path of the Parent Source Directory… Refer to **{SourceParentPath}**

### Remote Repo Path (Repo Type Only)

The Remote Path of the Repo… (typically http path of remote repo)

### Upstream Repo name (Repo Type Only)

The Remote Repo name… (if you don’t know what this is… use “origin”)

### Branch (Repo Type Only)

The Branch name… (if you don’t know what this is…. use “master”)

### DB Server (Repo Type Only)

DNS Name of the SQL Server that Hosts the Development Database

### DB Name (Repo Type Only)

Database Name in SQL

### Default (Repo Type Only)

Default Repo to Use when opening the Source Control Page

# FIRST TIME RUN

## Step 1

Run the Main Page in NAV (Page77777 – Source Control). If you did your setups correctly you will have no errors and the page should open.

## Step 2

On the Actions Tab, Click on the “Setup Local Repo’s”. This will Clone all the repos in your setup to your local machine (into the {SourceParentPath} you defined). If it Already exists, (i.e. the folder is not empty) it will skip over it.

Once this process is complete, you are ready to use Source Control. One of the First things you would want to do is Check your Repo Status… Click on “Repo Status” and wait for the result to open in your Favourite Text Editor.

You can also setup your dashboard in the respective DB to use the development dash. It has all the functionality directly on the dash that is included in this feature.

# FINAL WORDS

Hope you Enjoy this Tool and that you find it useful. Feel free to leave me a comment or question. Please just remember that I might not respond timeously to all questions.

# Have Fun

/Bert