How to Back Up and Restore a Dynamics NAV 2015 Database Using SQL and PowerShell:

Prerequisite:

NAV Server Instance: In these instructions, I do not cover how to configure a NAV Server Instances. However, in order for the following instructions to work, it is necessary to have a separate NAV Server Instance dedicated to the testing database/environment. The testing environment's NAV Server Instance needs to be configured properly, and you'll need to know a few of its specific configurations in order to setup the cmdlets explained below. Specifically, you'll need to know the Database Server and Database Name values. This information can be found within the Microsoft Dynamics NAV 2015 Administration application. The name of the testing database used in the following example is TPC NAV Staging.

## High Level Steps - How to Back Up and Restore a Dynamics NAV 2015 Database:

- In **SQL**, create a backup of the production NAV Database
- In **SQL**, delete the database you want to replace (if you're replacing an existing database)
- In **PowerShell**, restore the backup, using the **New-NAVDatabase** cmdlet
- In **Microsoft Dynamics NAV 2015 Administration**, restart the NAV Service
- In **PowerShell**, rename the NAV Companies, as appropriate, using the **Rename-NAVCompany** cmdlet
- Optional Step: in NAV, in the new testing database NAV Companies, change the System Indicator options on the Company Information setup page

## Detailed Steps - How to Create a NAV Database Backup Using SQL:

Because the PowerShell restore process requires a .bak file, I'll first show you how to create this backup file using SQL.

1. Open SQL Server Management Studio
   a. Right click on your production database
   b. Focus on the **Tasks** menu option
   c. Click on the **Back Up…** menu option

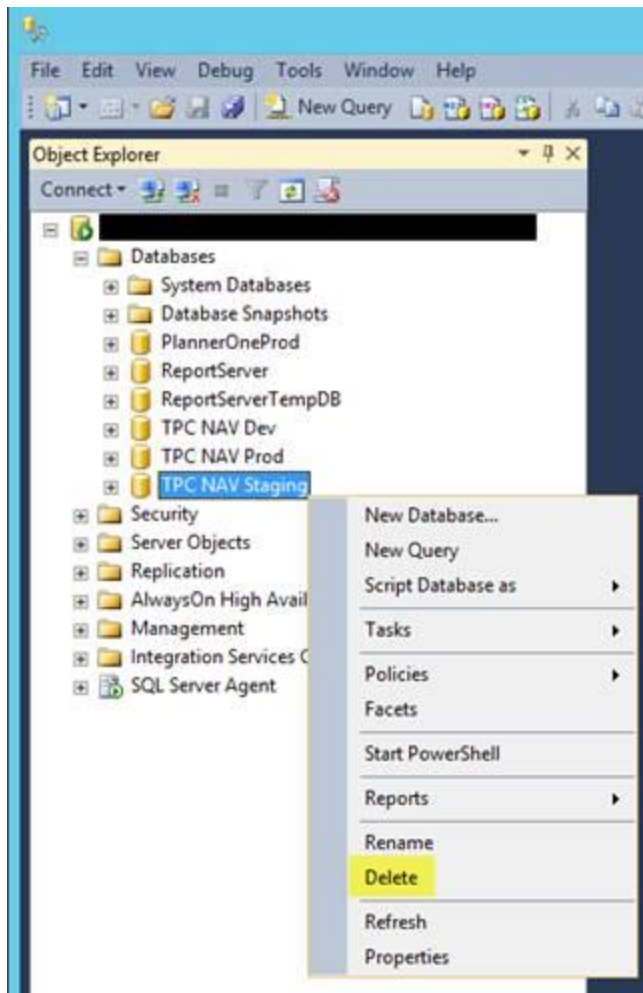The navigation path is shown in the following screenshot.
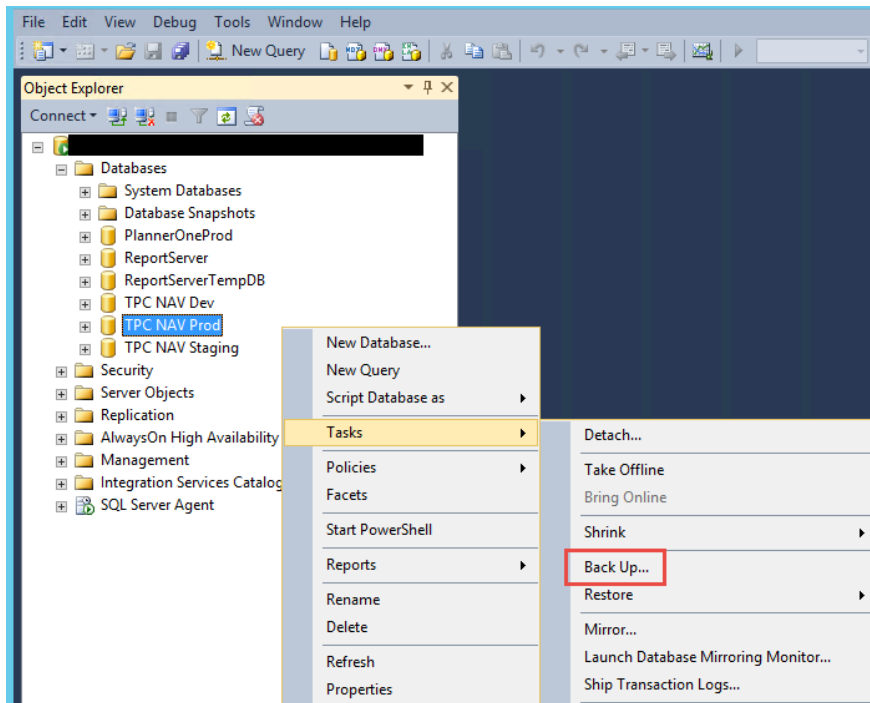
2. **Back Up Database Window:** From the **General** page, I configure the backup as shown in the following screenshot.
   a. Note that you may need to use the **Add** / **Remove** buttons to specify a unique file name and path for the new .bak file you'll create.
        i. **Important:** Make note of the file path used in the Destination section, as you'll need it when constructing your PowerShell cmdlet later in the process (or better yet, just copy/past it somewhere so you've got exactly what you'll need later)
   b. When you're ready to create the backup file, click on the **OK** button.
   c. After the back up process is finished, you now have the .bak file needed for the PowerShell restore.
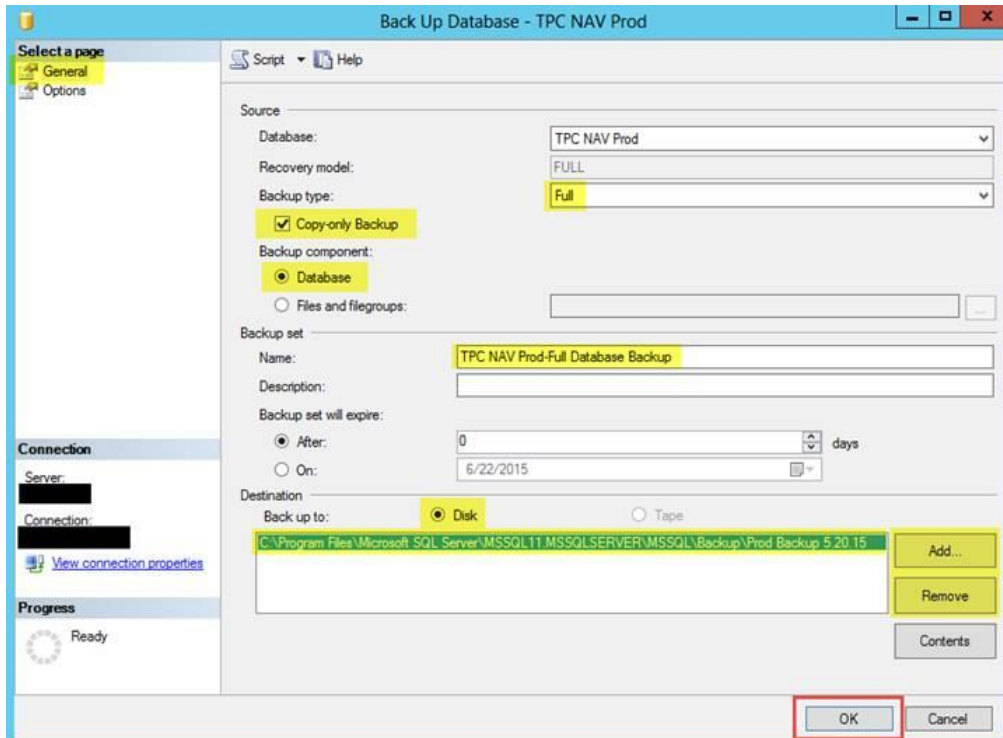
3. If replacing an existing testing database, delete that database.
    a. To do this, open SQL Server Management Studio and do the following:
        i. Right click on the testing database
        ii. Click on the **Delete** menu option

The navigation path is shown in the following screenshot.

a. **Delete Object Window:** From the **General** page, I configure the delete process as shown in the following screenshot. When you're ready, click **OK**.
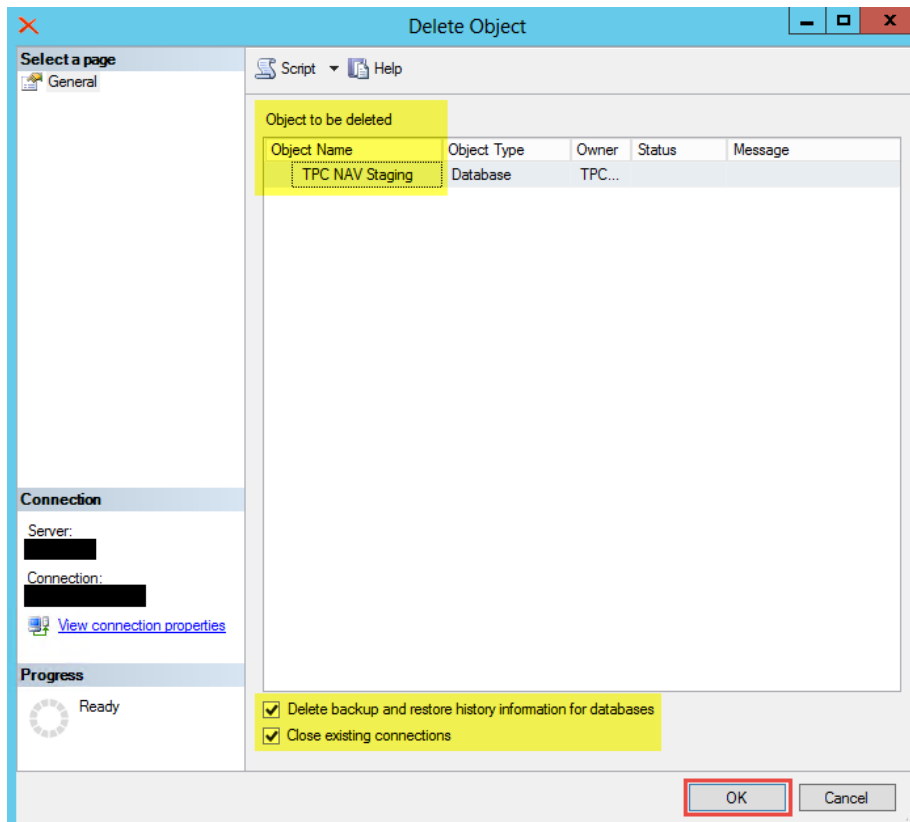
Detailed Steps - How to Restore a NAV Database Using PowerShell:

NOTE: For cmdlets on items 1-3 provided below, you can simply copy and paste them into PowerShell, and run them. Cmdlets shown in items 4 and 6 need to be configured with specific data like your NAV Server Instance, Server Name, Database Name, file paths, etc.

1. Open Windows PowerShell ISE.
    a. **Note:** You'll need to run PowerShell as an administrator. One way to do this is to hold down the Shift key, right click the PowerShell icon, and selecting the 'Run as administrator' option. Answer 'Yes' to any prompts that are presented.

2. If PowerShell has never been used before, you'll need to run through the following one time setup. You'll need to answer 'Yes' to the warning/prompt that is presented.
    a. Run the following cmdlet:

    Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

    b. Run the following cmdlet to see if your previous cmdlet worked properly.

    Get-ExecutionPolicy -List

    If the initial cmdlet worked properly, PowerShell will present the following information, stating that the CurrentUser ExecutionPolicy = RemoteSigned

```
PS C:\Windows\system32> Get-ExecutionPolicy -List

                    Scope                              ExecutionPolicy
                    -----                              ---------------
            MachinePolicy                                    Undefined
               UserPolicy                                    Undefined
                  Process                                    Undefined
              CurrentUser                                   RemoteSigned
             LocalMachine                                    Undefined
```

1. Next, you need to load the NAV administration cmdlets into PowerShell by executing the following cmdlets.
   a. **Note:** You'll need to run these two import cmdlets every time you open PowerShell, as PowerShell does not save these cmdlets.
   b. **Note:** The file location paths used in the cmdlets below are the default file location paths used when installing NAV. It's possible that these files might be in a different location, but check these file paths first.

   Import-Module 'C:\Program Files\Microsoft Dynamics NAV\80\service\NAVAdminTool.ps1' -Verbose
   Import-Module 'C:\Program Files (x86)\Microsoft Dynamics NAV\80\RoleTailored Client\Microsoft.Dynamics.Nav.Model.Tools.psd1' -Verbose

3. In PowerShell, create the restored database by run the following cmdlet: **New-NAVDatabase**.
   a. **Example cmdlet:**

   New-NAVDatabase -DatabaseName "TPC NAV Staging" -FilePath "C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup\Prod Backup 5.20.15" -DatabaseServer SERVERNAME -DestinationPath "C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\Staging DATA" -Timeout 5000 -verbose

   NOTES:
   -DatabaseName this is the name of the database you'll create. This value needs to match the Database Name value specified in the NAV Server Instance you want to use for running this separate database.
   -FilePath this is the file path that specifies the exact location of the .bak file you're trying to restore.
   -DatabaseServer this is the address of the server to which you're trying to restore the database.
   -DestinationPath this is the file location (on the -DatabaseServer) where you'll be putting the new database.

4. In the **Microsoft Dynamics NAV 2015 Administration** application, restart the NAV Server Instance that's associated to the new NAV database. This step is not strictly needed, but it's solved problems I've run into before, so I've included it in my standard process.
   a. Highlight your NAV Server Instance and run the **Restart** option.
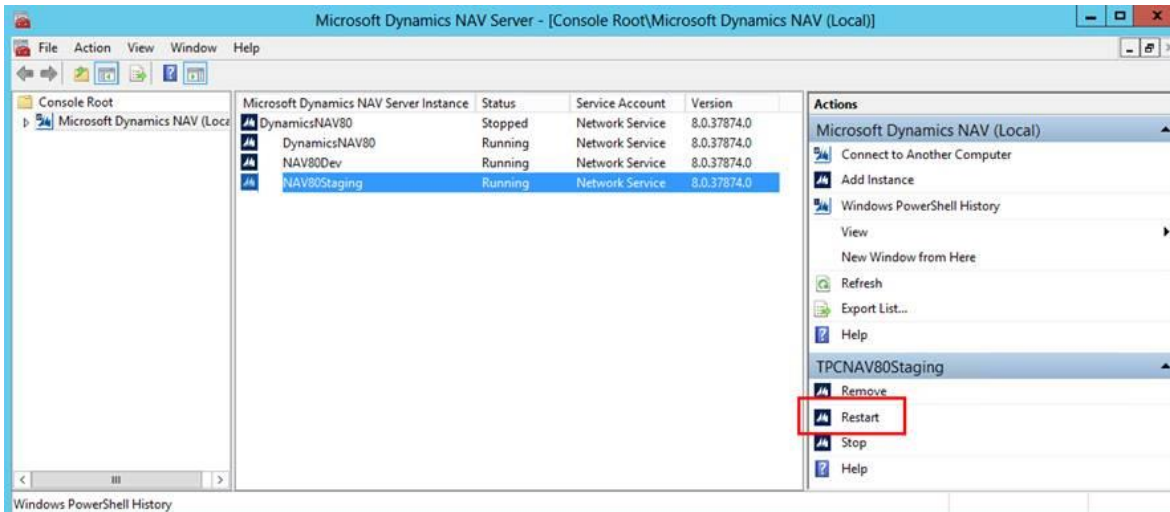
6. In PowerShell, rename the appropriate NAV Companies by using the following cmdlet: **Rename-NAVCompany**. This step is to help users easily distinguish between the production database and the testing database.
   a. **Example cmdlet:**

      Rename-NAVCompany -CompanyName TPC -NewCompanyName "TPC - Staging" - ServerInstance NAV80Staging -verbose

      NOTES:
      -CompanyName this is the name of the NAV Company you want to rename, as it existed in the production database. In the example above, the Company name is as follows: TPC
      -NewCompanyName this is the new name you want your NAV Company to have. In the example above, the new Company name is as follows: TPC - Staging
      -ServerInstance this is the name of the NAV Server Instance associated to the testing database you've setup

7. **Optional Setup:** In NAV, you can use the Company Information page's System Indicator Style field to change the color of the system indicator bar displayed at the top right of all NAV pages. I use this to make it easy for users to differentiate their production and testing databases. When the System Indicator Style field is set to 'Accent2,' the System Indicator is shaded blue, as show below. In my testing database, I set the value to 'Accent1,' which makes the System Indicator red. I like to do this, because it's an easy way for users to know if they're in the production or testing database. If it's blue, they're in production. If it's red, they're in testing.