E3 Series

Batch Device Renaming

Introduction

Batch Device Renaming is a PowerShell script for batch processing used to rename selected devices in E^3.Series to save time and reduce the amount of work involved. Renaming is done according to the definition of the new name mask. Instead of `*` and `?` wildcards, this script uses placeholders in brackets `[ ]`. Placeholders can be combined with each other.

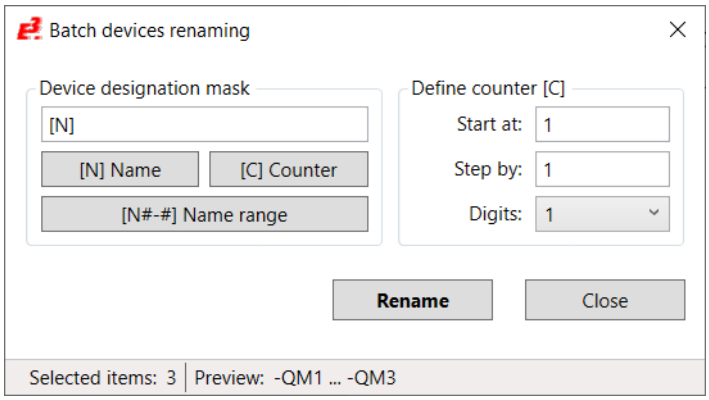

Figure 1 The main window of the script

Devices can be selected in both the Devices tree panel and/or a sheet.

A preview of new names is immediately shown in the status bar of the script window, but the devices are not renamed until the **Rename** button is pressed.

Software Requirements

- **E^3.Series 2016** and higher
- **PowerShell 5.0** and higher

Download

To get the latest version of **Batch Device Renaming** script, please, use the following link:

https://github.com/chukhran/E3.Series-powershell-scripts

Why PowerShell

For comfortable operation, the renaming process is required user interaction, in other words, GUI. There are a few solutions to get a GIU for Add-ons of E^3.Series however, for that need either Add-on should be compiled or new software should be installed. Using PowerShell solves the above problems: ps-scripts can have GUI without any compilation, PowerShell is also installed by default under any modern Windows OS.

PowerShell scripts execution policy

In PowerShell, script execution is disabled by default. The goal of the execution policy mechanism is to reduce the ways that PowerShell can be exploited by an attacker, allowing a user to operate more securely.

You can see the current execution policy by running `Get-ExecutionPolicy`:

```
PS> Get-Executionpolicy
Restricted
```

Here you see that the policy is set to `Restricted`.

To modify it to run `BatchDeviceRenaming.ps1`, run `Set-ExecutionPolicy` in an elevated session and follow the prompts:

```
PS> Set-ExecutionPolicy RemoteSigned
```

or (not recommended):

```
PS> Set-ExecutionPolicy Unrestricted
```

`Set-ExecutionPolicy` command usually must be run by an administrator, because regular users don't have permission to write to that portion of the Registry.

How to run the script

The script can be launched in several ways:

- as an external script from the console window
- as an Add-on from E^3.Series

Launching the script in the console window

The script can be launched either from *Windows PowerShell ISE* or the console window.

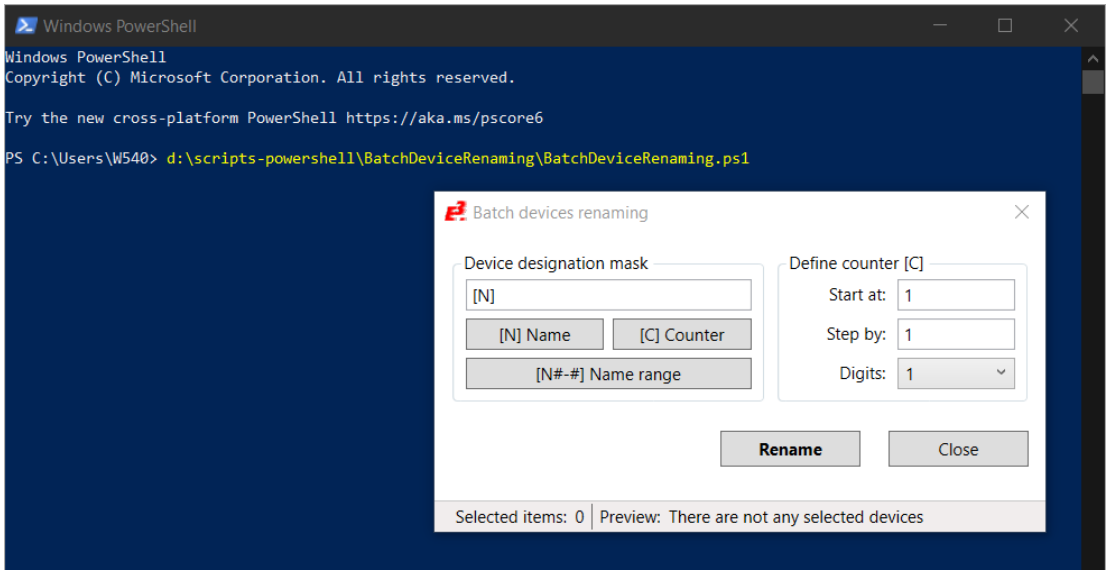

Figure 2 Launching the script from ps-console directly

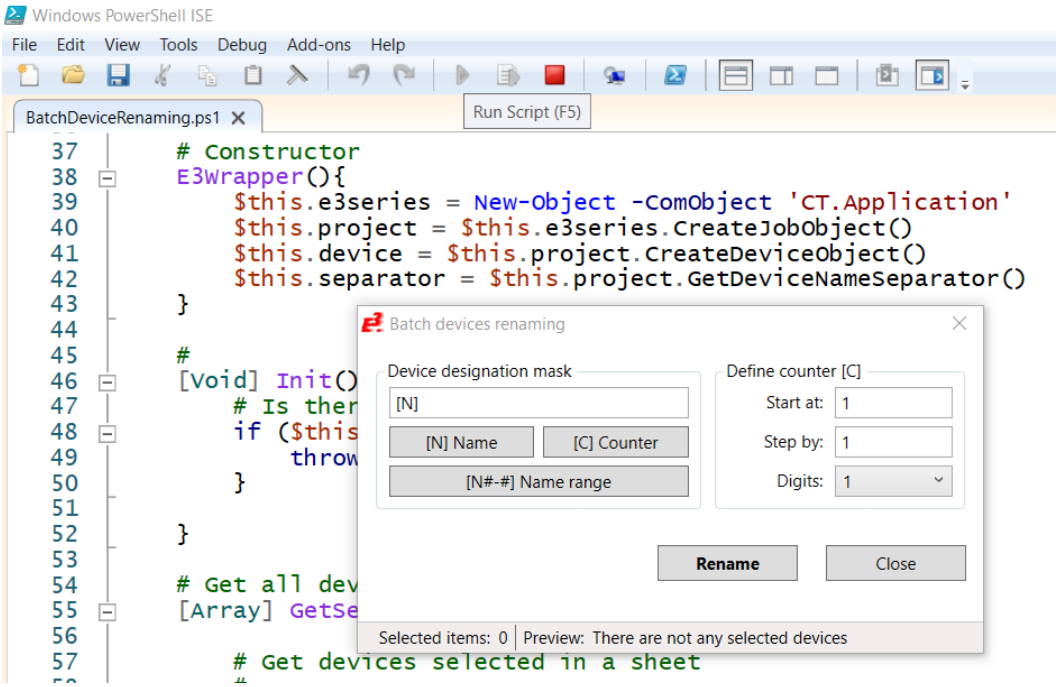

Figure 3 Launching the script from Windows PowerShell ISE

They're the easiest ways but are not always convenient.

Launcher for PowerShell scripts

PowerShell is a task automation solution made up of a command-line shell, a scripting language. When the ps-script is launching, a console window shows up always even using the keys `powershell.exe -WindowStyle Hidden -NoLogo ...` . In the last case, a console window will still flash.

To hide a console window, PowerShell has to be launched through a process that does not itself have a console window. It is known that WSH scripts do not have a console window, so VBScript can be used to launch `powershell.exe` in a hidden window.

It was created `launcher.vbs` script to run `BatchDeviceRenaming.ps1` script as an Add-on of E^3.Series.

Launching the script as an Add-on

Add-ons can be added in the **Customize** dialog.

- Select the **Tools** ⇨ **Customize...** command from the main menu bar, and
- Switch to the **Add-ons** tab.

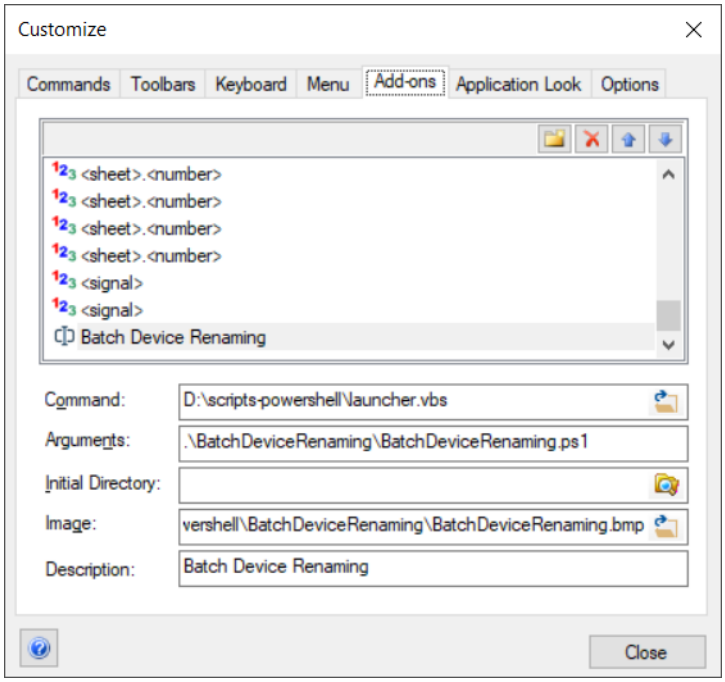

Figure 4 The Customize dialog — Add-ons

To add a new add-on, select the folder-icon in the field's right-hand margin.

Command: Click on the button in the right-hand margin to open the **Open** dialog and select `launcher.vbs` script as a launcher.

Arguments: Enter either the full or a relative path of the PowerShell script.

Image: Click on this button to open the **Open** dialog and select `BatchDeviceRenaming.bmp` file.

Description: The defined description is displayed in the status bar and as an icon tooltip.

Interface description

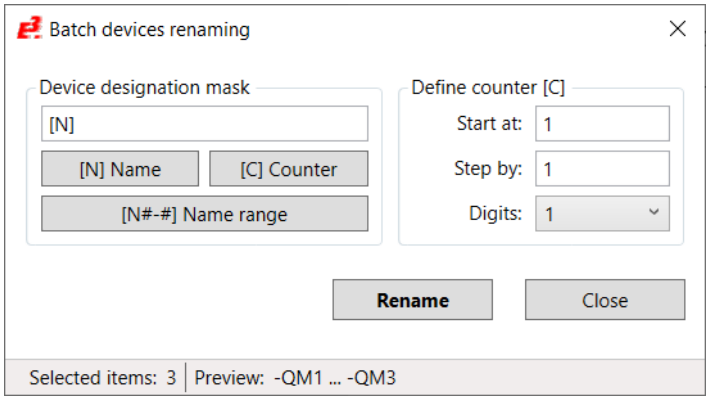

Figure 5 The main window of the script

Device designation mask

With this field, you can create a definition for a new device name. The buttons below allow inserting placeholders for the original name, parts of the original name, and a counter. Placeholders are always in brackets `[ ]`, while all other letters (without brackets) will be placed in the new name without a change.

Here is a description of all available placeholders:

`[N]` An original device name;

`[N#-#]` A part of the original name.
Examples:
`[N1-1]` – The first character of the original name;
`[N2-5]` – Characters 2 to 5 from the original name (totals 4 characters). The first letter is accessed with 1;

`[C]` A counter, as defined in the Define counter field.

Define counter [C]

Allows defining the counter for the `[C]` placeholder(s).

Start at: Number of the first device.;

Step by: The counter is increased by this value.;

Digits: Width of the counter field. If the value is more than 1, the script will insert leading zeros to get a fixed width number field.

Examples

Selected items	Renaming options	Result