PepperDash®

# PowerShell Scripts for Crestron Control Processors & Touch Panels

April 2021

Confidential

# Document History

| Date | Document Version | Author | Description |
|---|---|---|---|
| 2021-03-23 | 01.00 | Jonathan Arndt | Initial Draft |
| 2021-04-22 | 01.01 | Raymond Montoya | Updates to draft |
| 2021-04-23 | 01.02 | Jonathan Arndt | Release |

# Contents

# Synopsis

## Purpose

This documentation is written for software, created by PepperDash, for Crestron controlled systems. The purpose of this document is to provide guidance on executing PowerShell scripts that will connect to Crestron control processors and touch panels performing automated functionality detailed in this document.

## Assumptions

It is assumed that the user of this document has the following training and/or expertise:

- Familiarity with Crestron Toolbox software
- Basic understanding of working program and folder structure on Crestron 3-Series or 4-Series control processor and TSW touch panels
- Basic understanding of how to use and run Microsoft Windows PowerShell

## Required Tools

- Windows 7 operating system or higher
- Windows PowerShell (32 or 64 bit), minimum version 5
- Crestron PowerShell Scripting Enterprise Development Kit (EDK)
  https://sdkcon78221.crestron.com/downloads/EDK/EDK_Setup_1.0.5.3.exe
- TCP/IP network connectivity to Crestron control processor or touch panel
- Access to PepperDash Portal
- PowerShell script software package from PepperDash (obtained from PepperDash Portal)

## Recommended Tools

- Microsoft Visual Studio Code
  https://code.visualstudio.com/download

## Software Versions

| Enable Authentication (from default) Script | Release Version | Release Date |
|---|---|---|
| PDT.PowerShell.Authentication.ps1 | v01.00 | 4/23/2021 |

| Load file(s) to Crestron control Processor or Touch Panel | Release Version | Release Date |
|---|---|---|
| PDT.PowerShell.LoadScript.PS1 | v01.00 | 4/23/2021 |

| Send Custom Commands to Processor or Touch Panel | Release Version | Release Date |
|---|---|---|
| PDT.PowerShell.Authentication.ps1 | v01.00 | 4/23/2021 |

# Firmware Requirements

The following equipment must be running the required firmware. PepperDash PowerShell scripts assume Crestron equipment is running latest provided device firmware prior to executing scripts.

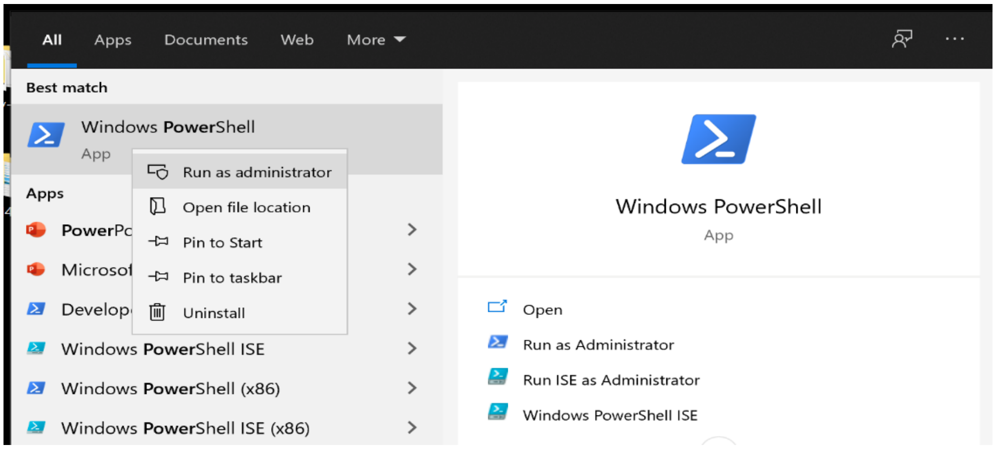
| Manufacturer | Model | Device Type | Crestron PUF | Firmware | Release Date |
|---|---|---|---|---|---|
| Crestron | 3-Series | Processor | N/A | 1.7000.0021 | Jan, 2021 |
| Crestron | TSW | Touch Panel | N/A | 3.000.0014 | Oct, 2020 |

## *General Disclaimer*

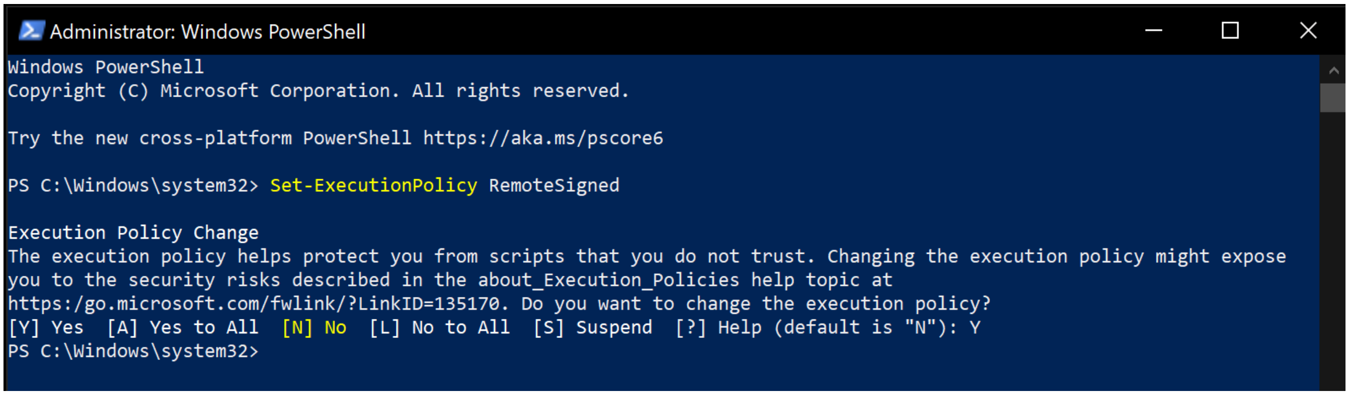
*All scripts written by PepperDash (and Crestron) require the PowerShell Execution Policy set to ‘remote signed’. This allows provided scripts to be ran on a local machine that were originally created on another machine. See instructions below on how to set the execution policy. Note: If you intend on utilizing PowerShell-5 for running attended scripts but later wish to utilize some other version of PowerShell for unattended (example: PowerShell 7), you will need to set the execution policy to ‘remote signed’ for both versions.*

## Setup an Execution Policy

1. Open PowerShell **as an Administrator.**



2. Use the following command (cmdlet) to enable running remotely signed scripts (note: all PepperDash scripts utilize Crestron cmdlets which are signed): “**Set-ExecutionPolicy** RemoteSigned”



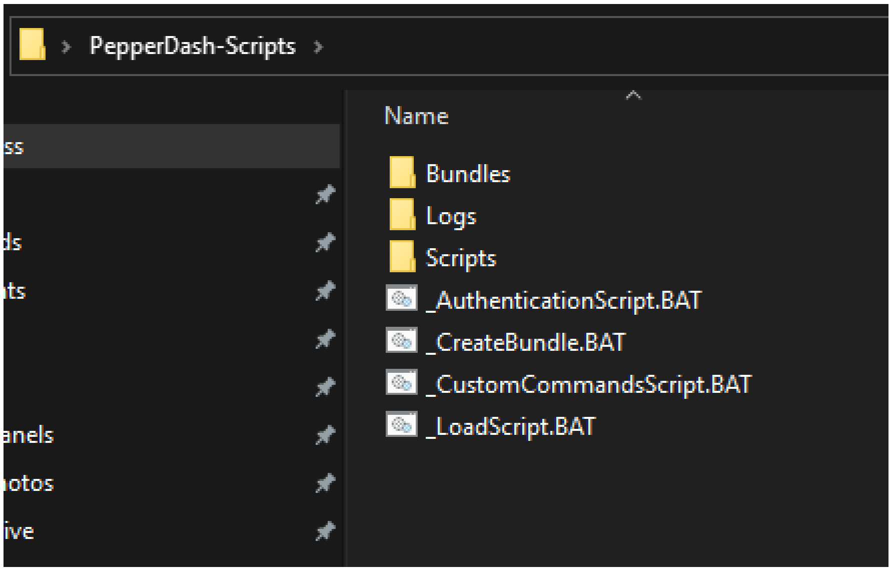
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Set-ExecutionPolicy RemoteSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
https:/go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): Y
PS C:\Windows\system32>
```

3. 3. Enter “Y” in the field to accept the execution policy

## Understanding the Folder Structure

*PowerShell scripts written by PepperDash require implementation of specific folder structure. Details of this structure our outlined below.*

1. Extract the provided ‘PepperDash-Scripts.zip’ file from PepperDash on your desktop. Extracting the provided zip file automatically provides the required folder structure.
2. Within the ‘PepperDash-Scripts’ folder there are three sub-folders: Scripts, Bundles, and Logs. The ‘Scripts’ folder contains the actual PowerShell scripts, the ‘Logs’ folder contains automatically generated logs entries, and the ‘Bundles’ folder contains example address books, custom command files, and is also where a firmware or PUF file would reside should the need arise to script load a firmware file.
3. The root folder of the PepperDash-Scripts will include multiple *.BAT files. These files allow you to run a PepperDash script unattended (also known as ‘headless’). Further documentation describing how to modify the *.BAT files will be covered in the document below.
4. Running PepperDash scripts unattended requires the use of a JSON formatted address book file with the file extension of *.pda (Example: addressBook.pda). The address book file must be a Java-Script-Object-Notation formatted file also known as JSON. Example address books are provided.



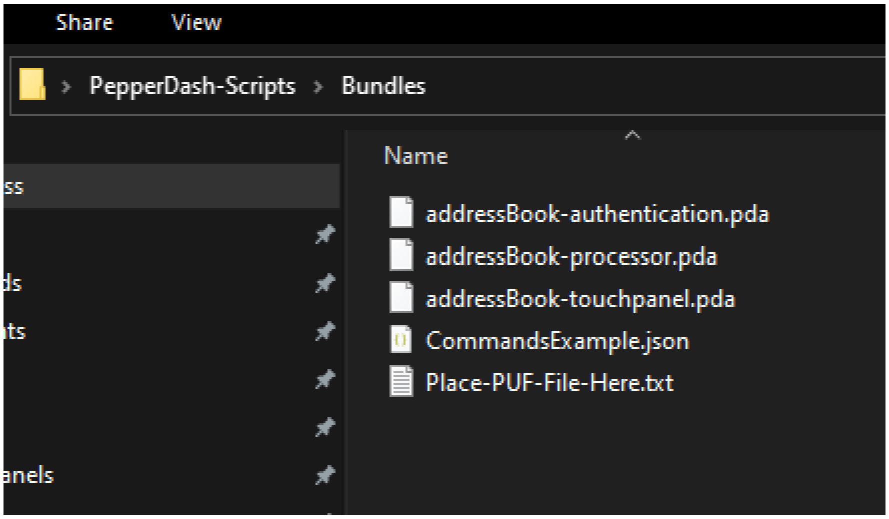
Bundles
Logs
Scripts
_AuthenticationScript.BAT
_CreateBundle.BAT
_CustomCommandsScript.BAT
_LoadScript.BAT

## Update Crestron Control Processor or Touch Panel Firmware (attended)

### *General Disclaimer*

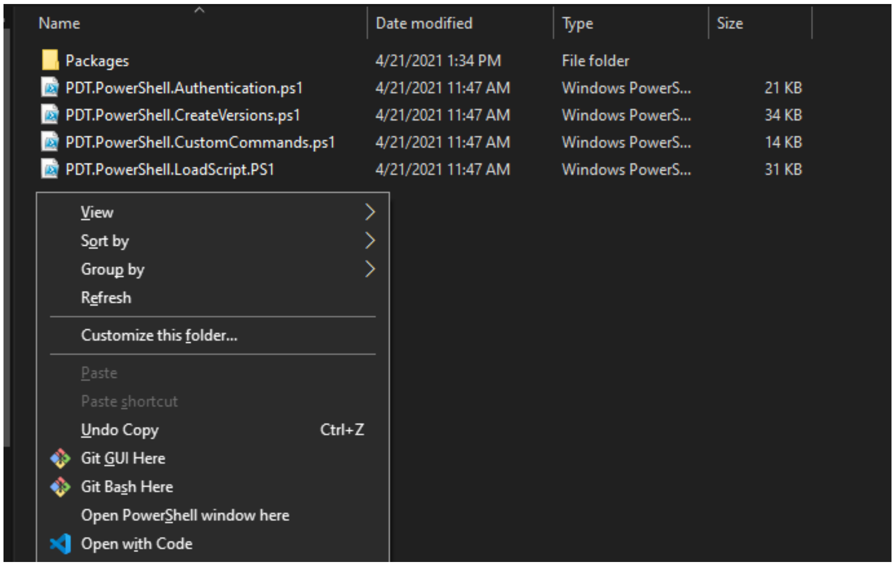
*PepperDash does not recommend updating Crestron control processor or Touchpanel firmware unattended and therefore will not be covered in this document.*

1. Place firmware PUF file in the 'Bundles' folder.



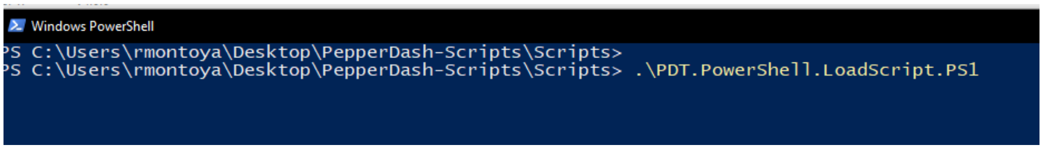
Name
- addressBook-authentication.pda
- addressBook-processor.pda
- addressBook-touchpanel.pda
- CommandsExample.json
- Place-PUF-File-Here.txt

2. Navigate to provided PowerShell script titled, **'PDT.PowerShell.LoadScript.PS1**' (Note: Script file is located within the 'Scripts' folder).
3. If you are running the scripts attended, hold **shift** and **right-click** in the File window (not on the actual script file) and select "Open PowerShell window here".



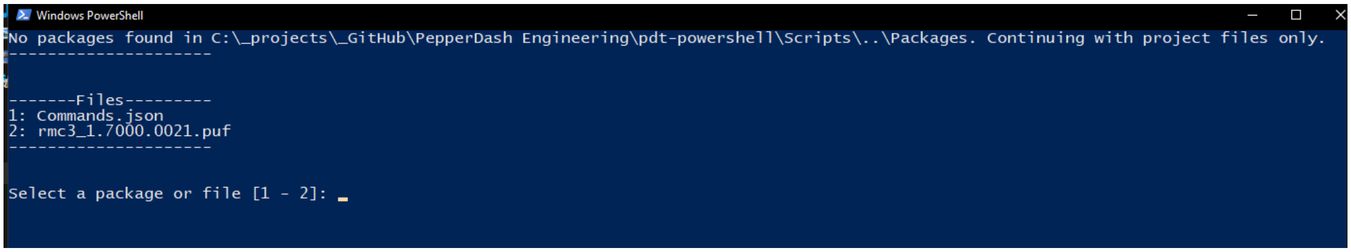
| Name | Date modified | Type | Size |
|---|---|---|---|
| Packages | 4/21/2021 1:34 PM | File folder | |
| PDT.PowerShell.Authentication.ps1 | 4/21/2021 11:47 AM | Windows PowerS... | 21 KB |
| PDT.PowerShell.CreateVersions.ps1 | 4/21/2021 11:47 AM | Windows PowerS... | 34 KB |
| PDT.PowerShell.CustomCommands.ps1 | 4/21/2021 11:47 AM | Windows PowerS... | 14 KB |
| PDT.PowerShell.LoadScript.PS1 | 4/21/2021 11:47 AM | Windows PowerS... | 31 KB |

4. To run the script, type ".\" in the console and the *name of the desired PowerShell script followed by* enter key.



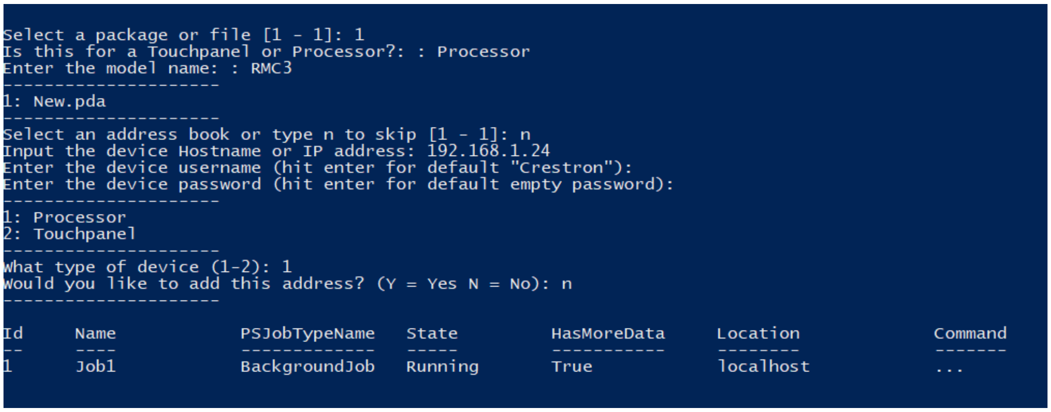
```
PS C:\Users\rmontoya\Desktop\PepperDash-Scripts\Scripts>
PS C:\Users\rmontoya\Desktop\PepperDash-Scripts\Scripts> .\PDT.PowerShell.LoadScript.PS1
```

5. If script discovers multiple files possible to load to the processor or touch panel the script will query for a choice of files to load.



```
No packages found in C:\_projects\_GitHub\PepperDash Engineering\pdt-powershell\Scripts\..\Packages. Continuing with project files only.
---------------------

-------Files---------
1: Commands.json
2: rmc3_1.7000.0021.puf
---------------------

Select a package or file [1 - 2]: _
```

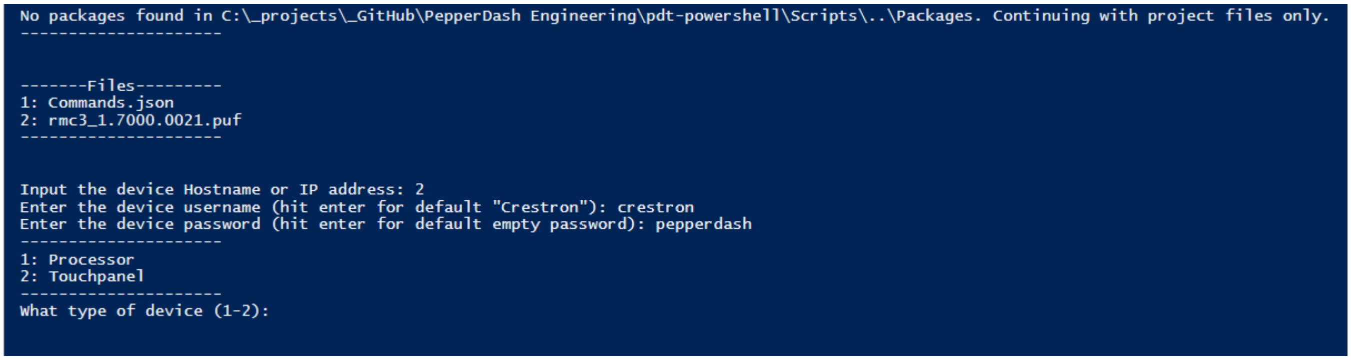
6. Enter a value for the desired file selection. The example above allows for a value of either '1' or '2'. *Note: At this prompt you are not able to load multiple files to a Processor or Touch Panel.*
7. If prompted with "Is this for a Touchpanel or Processor, Enter the device name of either 'Processor' or Touchpanel' exactly as it is shown in the PowerShell. Continue with this for all following questions.



```
Select a package or file [1 - 1]: 1
Is this for a Touchpanel or Processor?: : Processor
Enter the model name: : RMC3
---------------------
1: New.pda
---------------------
Select an address book or type n to skip [1 - 1]: n
Input the device Hostname or IP address: 192.168.1.24
Enter the device username (hit enter for default "Crestron"):
Enter the device password (hit enter for default empty password):
---------------------
1: Processor
2: Touchpanel
---------------------
What type of device (1-2): 1
Would you like to add this address? (Y = Yes N = No): n
---------------------

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Job1            BackgroundJob   Running       True            localhost            ...
```

8. Provide the 'device username' of the processor or touch panel required to connect via SSH. If the device credentials or authentication have not been setup the option to choose the default value of 'Crestron' is allowed by hitting enter when prompted.
9. Provide the 'device password' of the Processor or touch panel required to connect via SSH. If the device credentials or authentication have not been setup the option to choose the default empty value is allowed by hitting enter when prompted.
10. Provide the device type that files will be loaded to. A value of '1' for Processor or '2' for Touchpanel are

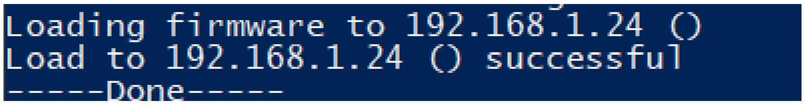
the only allowed values.



```
No packages found in C:\_projects\_GitHub\PepperDash Engineering\pdt-powershell\Scripts\..\Packages. Continuing with project files only.
---------------------


-------Files---------
1: Commands.json
2: rmc3_1.7000.0021.puf
---------------------


Input the device Hostname or IP address: 2
Enter the device username (hit enter for default "Crestron"): crestron
Enter the device password (hit enter for default empty password): pepperdash
---------------------
1: Processor
2: Touchpanel
---------------------
What type of device (1-2):
```

11. The script will prompt asking if the address provided should be added to an existing address book. By default, this value should be ‘N’ for no.

12. Wait for confirmation of successful completion.

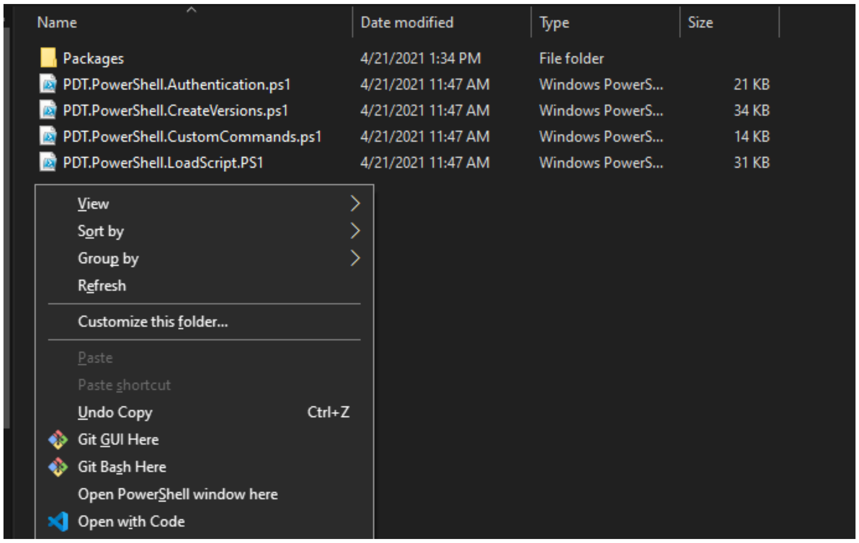


```
Loading firmware to 192.168.1.24 ()
Load to 192.168.1.24 () successful
-----Done-----
```

13. Any errors during the script process will be added to an error text file within the ‘Log’ folder.

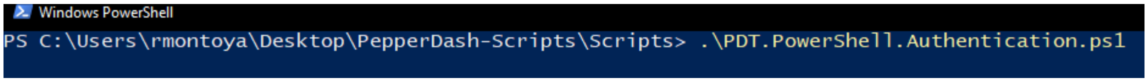
## Execute Authentication Script (attended)

1. Start Windows PowerShell and navigate to the path of where the scripts are stored or hold **shift** and **right-click** in the window where the PowerShell script exists. Select the “Open PowerShell window here” option.



| Name | Date modified | Type | Size |
|---|---|---|---|
| Packages | 4/21/2021 1:34 PM | File folder | |
| PDT.PowerShell.Authentication.ps1 | 4/21/2021 11:47 AM | Windows PowerS... | 21 KB |
| PDT.PowerShell.CreateVersions.ps1 | 4/21/2021 11:47 AM | Windows PowerS... | 34 KB |
| PDT.PowerShell.CustomCommands.ps1 | 4/21/2021 11:47 AM | Windows PowerS... | 14 KB |
| PDT.PowerShell.LoadScript.PS1 | 4/21/2021 11:47 AM | Windows PowerS... | 31 KB |

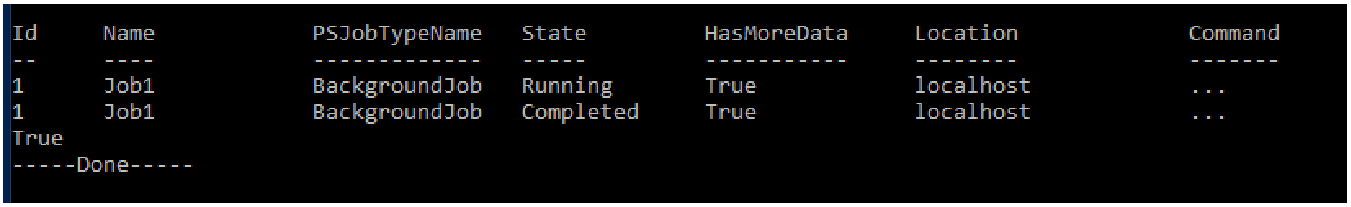
2. To execute a PowerShell script from the PowerShell command line, type in the full name of the script with the prefix “.\”. See example below.



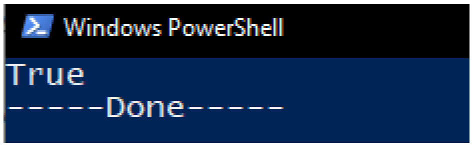
```
PS C:\Users\rmontoya\Desktop\PepperDash-Scripts\Scripts> .\PDT.PowerShell.Authentication.ps1
```

3. Enter the new username desired for the device after the prompt, ‘NewUsername’.
4. Enter the new password desired for the device after the prompt, ‘NewPassword”.
5. If using an address book enter the desired address book from the list or type ‘n’ to skip this selection.
6. If using an address book enter the desired address or ‘a’ for all or ‘n’ to skip.
7. If no address book file (*.pda) is discovered by the script (located in the ‘Bundles’ folder) enter the device hostname or IP address.
8. Enter the device current username (Note: If authentication has not yet been setup on the device, the username is ‘Crestron’). Hit enter for the script to enter the default ‘Crestron’ username for you.
9. Enter the device current password (Note: If authentication has not yet been setup on the device, the password is blank or empty). Hit enter for the script to enter the default empty password for you.
   Script will provide visual indication of the following: Ongoing job running in the background, indication of when the job has completed, and will automatically generate logs showing status of authentication script for each device the script connects to.

10. Wait for confirmation of successful completion.

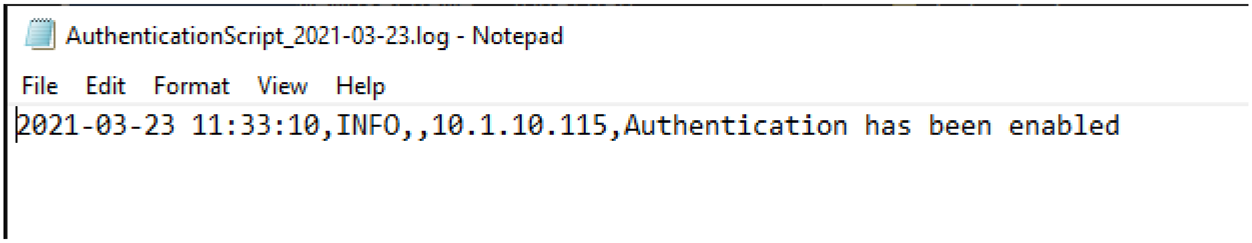


```
Id     Name          PSJobTypeName   State        HasMoreData     Location             Command
--     ----          -------------   -----        -----------     --------             -------
1      Job1          BackgroundJob   Running      True            localhost            ...
1      Job1          BackgroundJob   Completed    True            localhost            ...
True
-----Done-----
```



```
True
-----Done-----
```

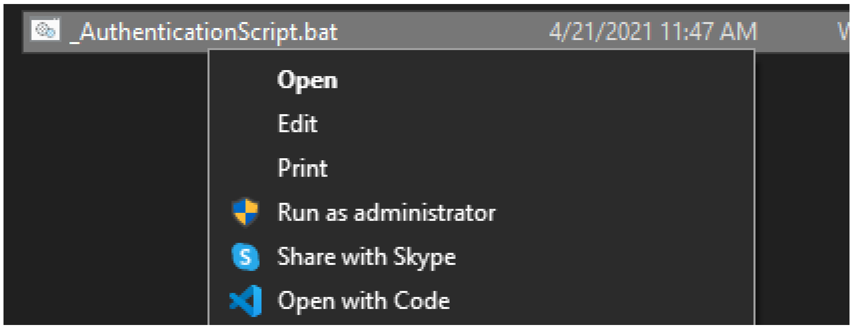
11. Check the ‘Log’ folder for any errors that may have occurred. Typical log entry looks like example below.



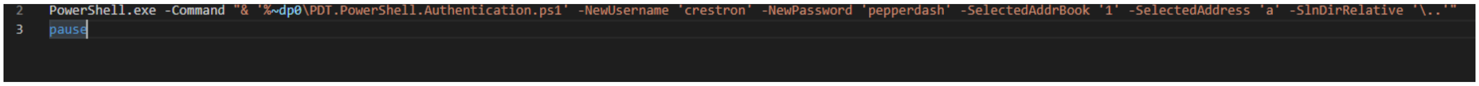
```
2021-03-23 11:33:10,INFO,,10.1.10.115,Authentication has been enabled
```

## Execute Authentication Script (unattended)

1. Navigate to the *_AuthenticationScript.bat* file. Note: PepperDash recommends editing JSON and BAT files using a free Microsoft tool called, Visual Studio Code. See download link under ‘Recommended Tools’.
2. Right click the file and select “Open with Code” or Visual Studio Code.

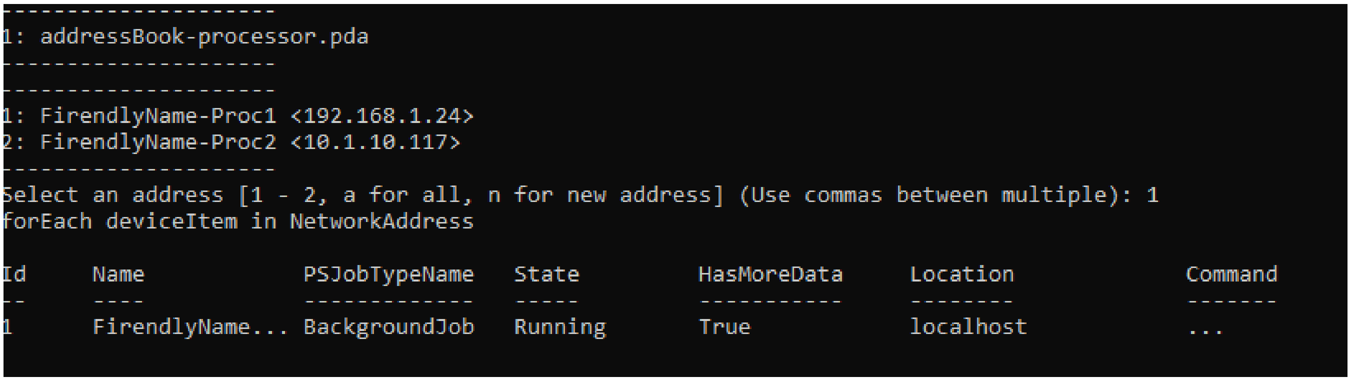


3. While inside the editor, input all information within the single quotes that pertains to the devices you will be working with. Note: PepperDash recommends removing all but a single address book within the 'Bundles' folder preventing utilization of wrong address book variables ("SelectAddrBook" and "SelectAddress").

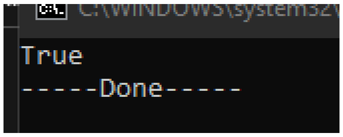


```
PowerShell.exe -Command "& '%~dp0\PDT.PowerShell.Authentication.ps1' -NewUsername 'crestron' -NewPassword 'pepperdash' -SelectedAddrBook '1' -SelectedAddress 'a' -SlnDirRelative '\..'"
pause
```

4. Upon completion of editing the variables in the *.bat file, the file is ready to be ran unattended from either any Windows 'Task Scheduler' or from simply double-clicking the *.bat file manually.
5. To run manually, return to the folder where the *.bat file exists, and double click the *.bat file.
6. A command window will open showing the same prompts as the attended script. However, when using the *.bat file, the script enters the values automatically.
7. You will see the script in progress, then a completed "---Done---" Message



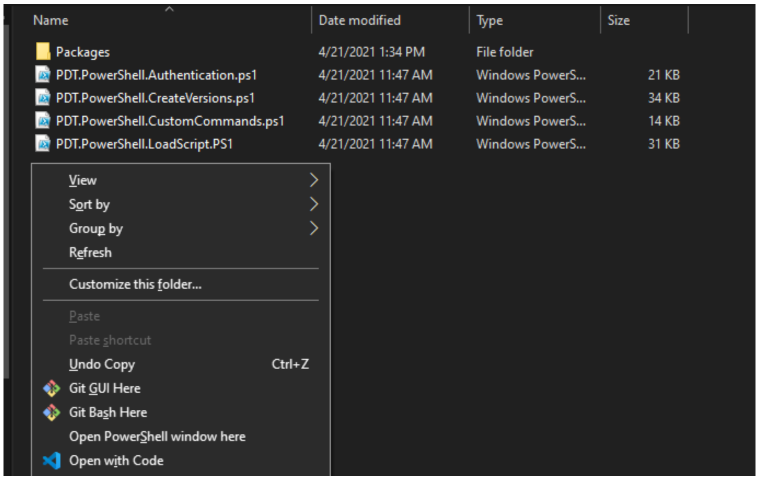
```
---------------------
1: addressBook-processor.pda
---------------------
---------------------
1: FirendlyName-Proc1 <192.168.1.24>
2: FirendlyName-Proc2 <10.1.10.117>
---------------------
Select an address [1 - 2, a for all, n for new address] (Use commas between multiple): 1
forEach deviceItem in NetworkAddress

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      FirendlyName... BackgroundJob   Running       True            localhost            ...
```



```
True
-----Done-----
```

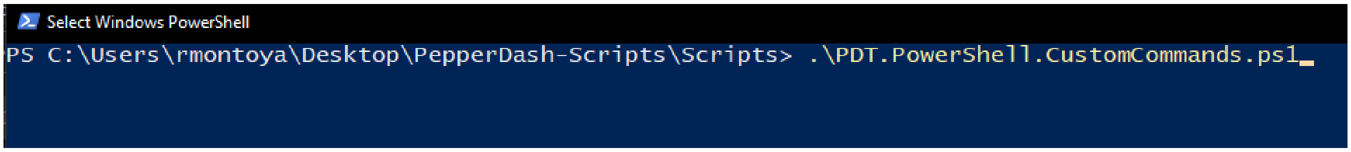
## Execute Custom Commands Script (attended)

1. Start Windows PowerShell and navigate to the path of where the scripts are stored or hold **shift** and **right-click** in the window where the PowerShell script exists. Select the “Open PowerShell window here” option.



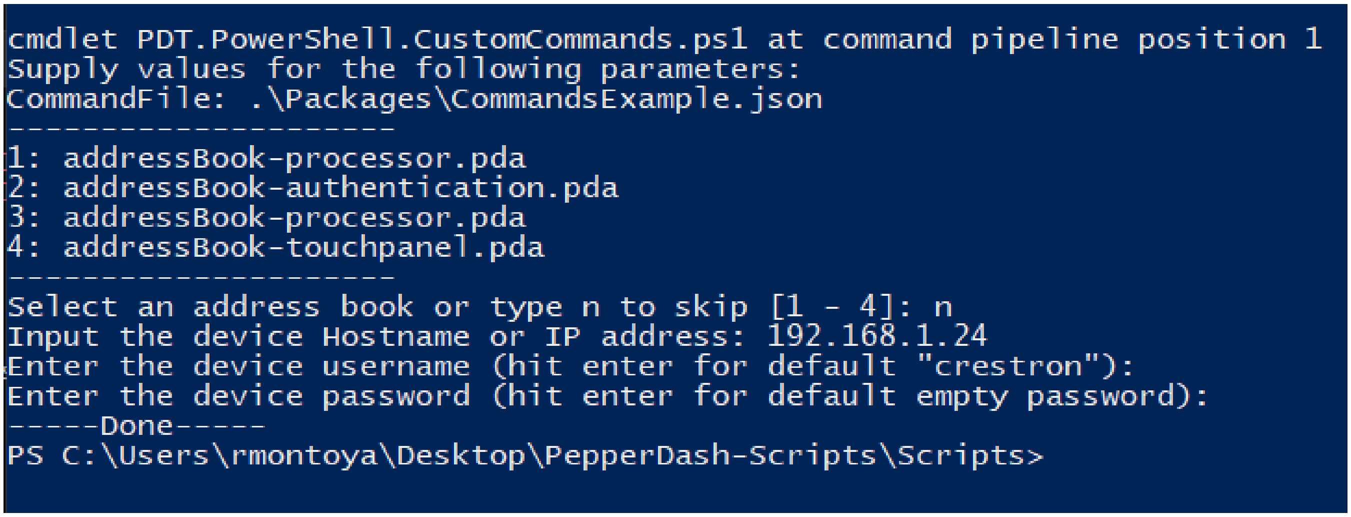
| Name | Date modified | Type | Size |
|---|---|---|---|
| Packages | 4/21/2021 1:34 PM | File folder | |
| PDT.PowerShell.Authentication.ps1 | 4/21/2021 11:47 AM | Windows PowerS... | 21 KB |
| PDT.PowerShell.CreateVersions.ps1 | 4/21/2021 11:47 AM | Windows PowerS... | 34 KB |
| PDT.PowerShell.CustomCommands.ps1 | 4/21/2021 11:47 AM | Windows PowerS... | 14 KB |
| PDT.PowerShell.LoadScript.PS1 | 4/21/2021 11:47 AM | Windows PowerS... | 31 KB |

2. To execute a PowerShell script from the PowerShell command line, type in the full name of the script with the prefix “.\”. See example below.



```
PS C:\Users\rmontoya\Desktop\PepperDash-Scripts\Scripts> .\PDT.PowerShell.CustomCommands.ps1_
```

3. The script will be looking for the JSON formatted ‘CommandFile’. Enter the file name of the *.json custom commands file to be sent to Processor or Touchpanel. An example of a custom command JSON formatted file is provided by PepperDash. Note: Custom commands JSON file must be located in the ‘Bundles’ folder.
4. Select an address book to pull the IP Address from [1 – 4] or “n” to skip.
5. If skipped, enter the IP Address of the processor you want to run the commands on.
6. Enter the Device Username (Or hit enter if the default username is “crestron”).
7. Enter the Device password (Or hit enter if the default username is empty “”).
8. Select enter and wait for a successful completion message.
9. Check the ‘Log’ folder for any error occurred during script process.

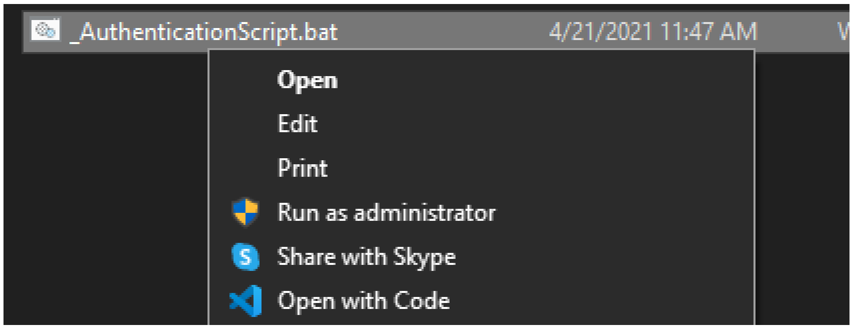


```
cmdlet PDT.PowerShell.CustomCommands.ps1 at command pipeline position 1
Supply values for the following parameters:
CommandFile: .\Packages\CommandsExample.json
---------------------
1: addressBook-processor.pda
2: addressBook-authentication.pda
3: addressBook-processor.pda
4: addressBook-touchpanel.pda
---------------------
Select an address book or type n to skip [1 - 4]: n
Input the device Hostname or IP address: 192.168.1.24
Enter the device username (hit enter for default "crestron"):
Enter the device password (hit enter for default empty password):
-----Done-----
PS C:\Users\rmontoya\Desktop\PepperDash-Scripts\Scripts>
```

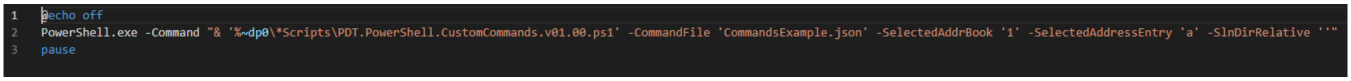
## Execute Custom Commands Script (unattended)

Navigate to the _CustomCommandsScript.bat file. Note: PepperDash recommends editing JSON and BAT files using a free Microsoft tool called, Visual Studio Code. See download link under ‘Recommended Tools’.

8. Right click the file and select “Open with Code” or Visual Studio Code.

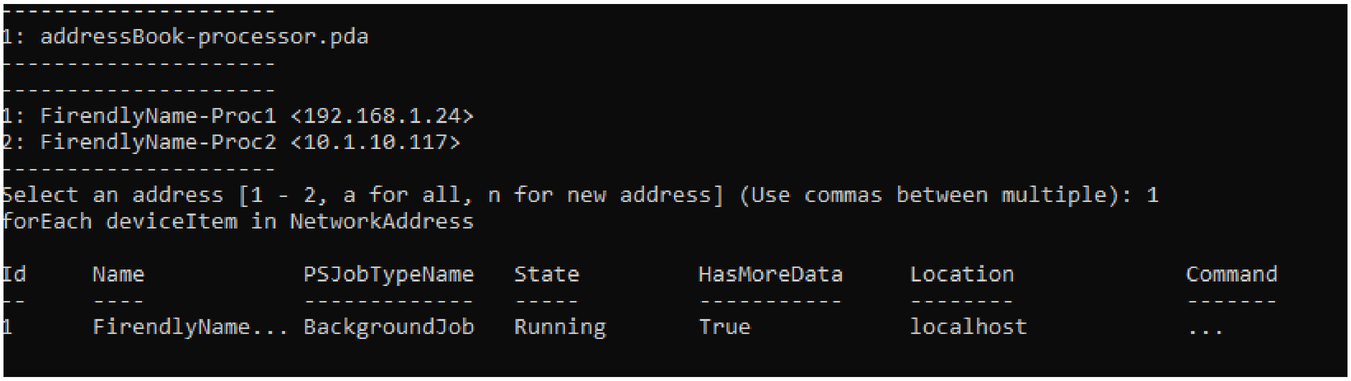


9. While inside the editor, input the file path where the Custom Commands script “PDT.PowerShell.CustomCommands.vXX.XX.PS1” exists.

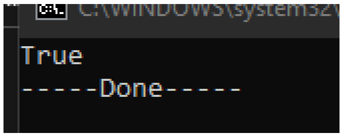


```
@echo off
PowerShell.exe -Command "& '%~dp0\*Scripts\PDT.PowerShell.CustomCommands.v01.00.ps1' -CommandFile 'CommandsExample.json' -SelectedAddrBook '1' -SelectedAddressEntry 'a' -SlnDirRelative ''"
pause
```

10. While inside the editor, input all information within the single quotes that pertains to the devices you will be working with. Note: PepperDash recommends removing all but a single address book within the 'Bundles' folder preventing utilization of wrong address book variables ("CommandFile", "SelectAddrBook" and "SelectedAddressEntry"). Custom commands JSON file must be located within the 'Bundles' folder.
11. Upon completion of editing the variables in the *.bat file, the file is ready to be ran unattended from either any Windows 'Task Scheduler' or from simply double-clicking the *.bat file manually.
12. To run manually, return to the folder where the *.bat file exists, and double click the _*.bat file.
13. A command window will open showing the same prompts as the attended script. However, when using the *.bat file, the script enters the values automatically.
14. You will see the script in progress, then a completed "---Done---" Message.

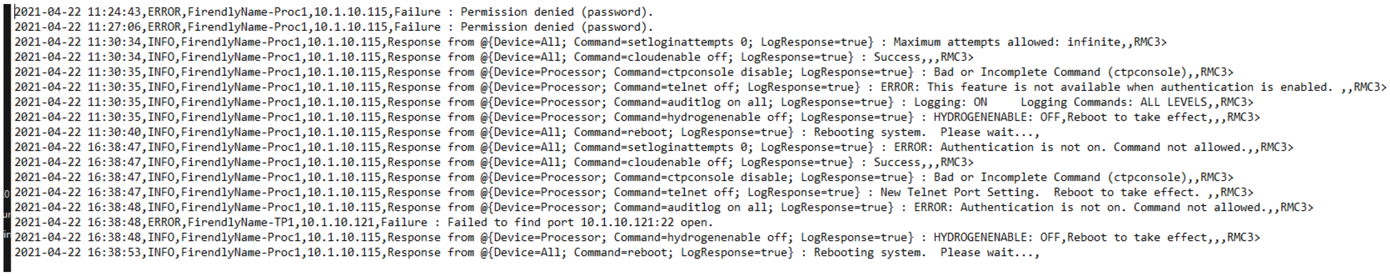


```
--------------------
1: addressBook-processor.pda
--------------------
--------------------
1: FirendlyName-Proc1 <192.168.1.24>
2: FirendlyName-Proc2 <10.1.10.117>
--------------------
Select an address [1 - 2, a for all, n for new address] (Use commas between multiple): 1
forEach deviceItem in NetworkAddress

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      FirendlyName... BackgroundJob   Running       True            localhost            ...
```



```
True
-----Done-----
```

15. Navigate the 'Log' folder and open the CustomCommand.txt file to view any errors that may have occurred. Example of typical error log file entry is below.



```
2021-04-22 11:24:43,ERROR,FirendlyName-Proc1,10.1.10.115,Failure : Permission denied (password).
2021-04-22 11:27:06,ERROR,FirendlyName-Proc1,10.1.10.115,Failure : Permission denied (password).
2021-04-22 11:30:34,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=All; Command=setloginattempts 0; LogResponse=true} : Maximum attempts allowed: infinite,,RMC3>
2021-04-22 11:30:34,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=All; Command=cloudenable off; LogResponse=true} : Success,,,RMC3>
2021-04-22 11:30:35,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=Processor; Command=ctpconsole disable; LogResponse=true} : Bad or Incomplete Command (ctpconsole),,RMC3>
2021-04-22 11:30:35,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=Processor; Command=telnet off; LogResponse=true} : ERROR: This feature is not available when authentication is enabled. ,,RMC3>
2021-04-22 11:30:35,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=Processor; Command=auditlog on all; LogResponse=true} : Logging: ON     Logging Commands: ALL LEVELS,,RMC3>
2021-04-22 11:30:35,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=Processor; Command=hydrogenenable off; LogResponse=true} : HYDROGENENABLE: OFF,Reboot to take effect,,,RMC3>
2021-04-22 11:30:40,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=All; Command=reboot; LogResponse=true} : Rebooting system.  Please wait...,
2021-04-22 16:38:47,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=All; Command=setloginattempts 0; LogResponse=true} : ERROR: Authentication is not on. Command not allowed.,,RMC3>
2021-04-22 16:38:47,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=All; Command=cloudenable off; LogResponse=true} : Success,,,RMC3>
2021-04-22 16:38:47,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=Processor; Command=ctpconsole disable; LogResponse=true} : Bad or Incomplete Command (ctpconsole),,RMC3>
2021-04-22 16:38:47,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=Processor; Command=telnet off; LogResponse=true} : New Telnet Port Setting.  Reboot to take effect. ,,RMC3>
2021-04-22 16:38:48,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=Processor; Command=auditlog on all; LogResponse=true} : ERROR: Authentication is not on. Command not allowed.,,RMC3>
2021-04-22 16:38:48,ERROR,FirendlyName-TP1,10.1.10.121,Failure : Failed to find port 10.1.10.121:22 open.
2021-04-22 16:38:48,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=Processor; Command=hydrogenenable off; LogResponse=true} : HYDROGENENABLE: OFF,Reboot to take effect,,,RMC3>
2021-04-22 16:38:53,INFO,FirendlyName-Proc1,10.1.10.115,Response from @{Device=All; Command=reboot; LogResponse=true} : Rebooting system.  Please wait...,
```

# Notices

## Notice of Ownership and Copyright

The material in which this notice appears is the property of PepperDash Technology Corporation, which claims copyright under the laws of the United States of America in the entire body of material and in all parts thereof, regardless of the use to which it is being put. Any use, in whole or in part, of this material by another party without the express written permission of PepperDash Technology Corporation is prohibited. PepperDash Technology Corporation reserves all rights under applicable laws.

## Notice of Confidentiality

The material in which this notice appears is confidential to PepperDash Technology Corporation. If you have been provided a copy of this material, it is with the understanding that you will not share it with others without the express written consent of PepperDash Technology Corporation.

## Notice of Trademark and Servicemark

PepperDash®, Sentegy®, AVUX® and AV360® are the marks of PepperDash Technology Corporation, registered with the U.S. Patent and Trademark Office. PepperDash Portal™, PepperDash Essentials™, and PepperDash Connect™ are the unregistered marks of PepperDash Technology Corporation. Any use, in whole or in part, of these marks by another party without the express written permission of PepperDash Technology Corporation is prohibited.

A higher level of control™