MyWebUniversity

python PowerShell FPDF Library PDF generator

Full credit is given to the above companies including
the OS that this PDF file was generated!

PowerShell Get-Help on command 'Export-Console'

PS C:\Users\wahid> Get-Help Export-Console

NAME

Export-Console

SYNOPSIS

Exports the names of snap-ins in the current session to a console file.

SYNTAX

```
Export-Console [[-Path] <System.String>] [-Force] [-NoClobber] [-Confirm]
[-WhatIf] [<CommonParameters>]
```

DESCRIPTION

The `Export-Console` cmdlet exports the names of the Windows PowerShell snap-ins in the current session to a Windows PowerShell console file (.psc1). You can use this cmdlet to save the snap-ins for use in future sessions.

To add the snap-ins in the .psc1 console file to a session, start Windows PowerShell (PowerShell.exe) at the command line by using Cmd.exe or another Windows PowerShell session, and then use the PSConsoleFile parameter of PowerShell.exe to specify the console file.

For more information about Windows PowerShell snap-ins, see about_PSSnapins (About/about_PSSnapins.md).

PARAMETERS

-Force <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet overwrites the data in a console file without warning, even if the file has the read-only attribute. The read-only attribute is changed and is not reset when the command finishes.

-NoClobber <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet does not overwrite an existing console file. By default, if a file occurs in the specified path, `Export-Console` overwrites the file without warning.

-Path <System.String>

Specifies a path and file name for the console file (`*.psc1`). Enter an optional path and name. Wildcard characters are not permitted.

If you specify only a file name, `Export-Console` creates a file that has that name and the .psc1 file name extension in the current directory.

This parameter is required unless you have opened Windows PowerShell with the PSConsoleFile parameter or exported a console file during the current session. It is also required when you use the NoClobber parameter to prevent the current console file from being overwritten.

If you omit this parameter, `Export-Console` overwrites the console file that was used most recently in this session. The path of the most recently used console file is stored in the value of the $ConsoleFileName automatic variable. For more information, see about_Automatic_Variables (About/about_Automatic_Variables.md).

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

Example 1: Export the names of snap-ins in the current session

```
PS C:\> Export-Console -Path $pshome\Consoles\ConsoleS1.psc1
```

This command exports the names of Windows PowerShell snap-ins in the current session to the `ConsoleS1.psc1` file in the Consoles folder of the Windows PowerShell installation folder, `$pshome`.

Example 2: Export the names of snap-ins to the most recent console file

```
Export-Console
```

This command exports the names of Windows PowerShell snap-ins from current session to the Windows PowerShell console file that was most recently used in the current session. It overwrites the previous file contents.

If you have not exported a console file during the current session, you are prompted for permission to continue and then prompted for a file name.

-- Example 3: Add a snap-in and export the names of snap-ins --

```
Add-PSSnapin NewPSSnapin
```

```
Export-Console -path NewPSSnapinConsole.psc1
powershell.exe -PsConsoleFile NewPsSnapinConsole.psc1
```

These commands add the NewPSSnapin Windows PowerShell snap-in to the current session, export the names of Windows PowerShell snap-ins in the current session to a console file, and then start a Windows PowerShell session with the console file.

The first command uses the `Add-PSSnapin` cmdlet to add the NewPSSnapin snap-in to the current session. You can only add Windows PowerShell snap-ins that are registered on your system.

The second command exports the Windows PowerShell snap-in names to the `NewPSSnapinConsole.psc1` file.

The third command starts Windows PowerShell with the `NewPSSnapinConsole.psc1` file. Because the console file includes the Windows PowerShell snap-in name, the cmdlets and providers in the snap-in are available in the current session.

- Example 4: Export names of snap-ins to a specified location -

```
PS C:\> export-console -path Console01
PS C:\> notepad console01.psc1
<?xml version="1.0" encoding="utf-8"?>
<PSConsoleFile ConsoleSchemaVersion="1.0">
  <PSVersion>2.0</PSVersion>
  <PSSnapIns>
    <PSSnapIn Name="NewPSSnapin" />
  </PSSnapIns>
</PSConsoleFile>
```

This command exports the names of the Windows PowerShell snap-ins in the current session to the `Console01.psc1` file in the current directory.

The second command displays the contents of the `Console01.psc1` file in Notepad.

------- Example 5: Determine the console file to update -------

```
powershell.exe -PSConsoleFile Console01.psc1
Add-PSSnapin MySnapin
Export-Console NewConsole.psc1
$ConsoleFileName
Add-PSSnapin SnapIn03
Export-Console
```

This example shows how to use the `$ConsoleFileName` automatic variable to determine the console file that will be updated if you use `Export-Console` without a Path parameter value.

The first command uses the PSConsoleFile parameter of PowerShell.exe to open Windows PowerShell with the `Console01.psc1` file.

The second command uses the `Add-PSSnapin` cmdlet to add the MySnapin Windows PowerShell snap-in to the current session.

The third command uses the `Export-Console` cmdlet to export the names of all the Windows PowerShell snap-ins in the session to the `NewConsole.psc1` file.

The fourth command displays the `$ConsoleFileName` variable. It contains the most recently used console file. The sample output shows that NewConsole.ps1 is the most recently used file.

The fifth command adds SnapIn03 to the current console.

The sixth command uses the `Export-Console` cmdlet without a Path parameter. This command exports the names of all the Windows PowerShell snap-ins in the current session to the most recently used file, `NewConsole.psc1`.

REMARKS

To see the examples, type: "get-help Export-Console -examples".

For more information, type: "get-help Export-Console -detailed".

For technical information, type: "get-help Export-Console -full".

For online help, type: "get-help Export-Console -online"