PoSHServer Documentation

AUTHOR:

YUSUF OZTURK (MVP)

http://www.poshserver.net

Contents

Introduction

PoSHServer is a secure, flexible and lightweight web server to meet your requirements. It's free and open source. You can add or remove features via PowerShell ISE. You just need to know PowerShell scripting language to work on it. PoSHServer supports many features:

- Authentication
  - Basic Authentication
  - Windows Authentication
- PHP
  - PHP 5.3.X
  - PHP 5.4.X
- Security
  - IP Restriction
  - Content Filtering
  - Directory Browsing
  - 404 Custom Error Page
- Logging
  - Advanced Logging
  - Log Parser
- Installation
  - Running as PowerShell Process
  - Running as Background Job
- SSL
  - Self-Signed SSL Certificate
  - Commercial SSL Certificate
- Others
  - Custom Mime Types
  - Background Jobs
  - Get/Post Support
  - Windows Server Core Support

Installation

Before beginning PoSHServer installation, please make sure you have Administrative privileges on server. Because PoSHServer setup requires to add PowerShell module files into C:\Program

Files\PoSHServer directory. If you don't have Administrative privileges, then PoSHServer setup can't access to Program Files directory.

1. If you have privileges, you can start installation by executing PoSHServer.exe file.

Figure 1: Welcome to PoSHServer setup
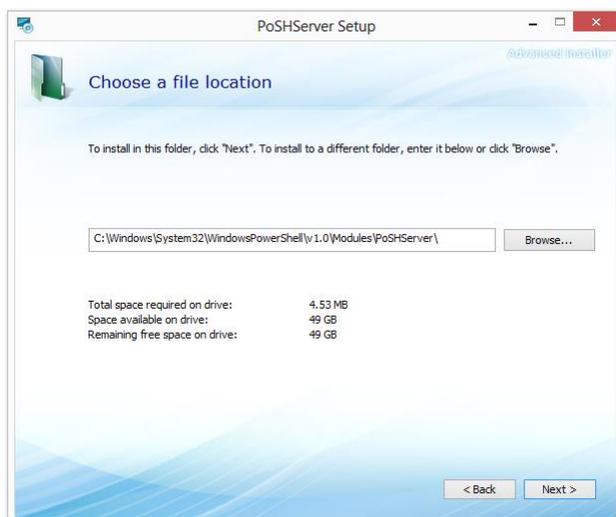
2. Click "Next" to continue installation process.

Figure 2: Choose a file location for PoSHServer setup

By default, you will see your PowerShell modules directory. If your PowerShell modules directory is different than this path, then you should change it with correct path. You websites files could

be in any path but core files of PoSHServer should be in PowerShell modules directory to work without any problem.

3. You can begin PoSHServer installation by clicking "Install".

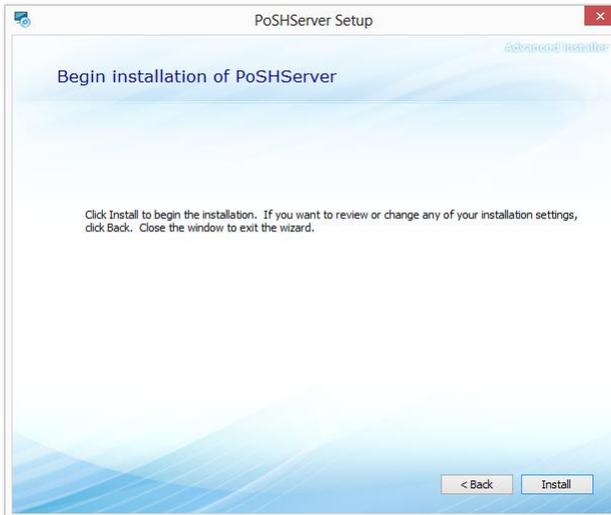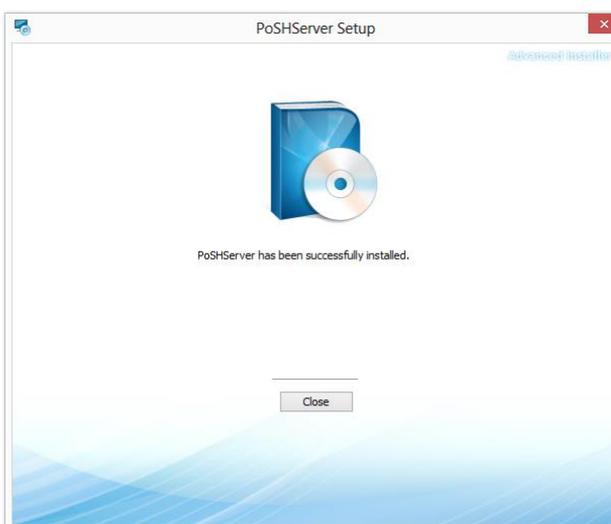**4.** When installation is finished, you can close setup manager.

After installation, you can go to C:\Program Files directory to check if installation is successful. All source codes are under PoSHServer directory. You can use PowerShell ISE to modify source codes for

your requirements. PoSHServer doesn't do any registry changes so you are always free to change or remove files. If something is broken, you just need to past original files.

How to start PoSHServer?

You have two options to start PoSHServer:

1. Run as PowerShell Process
2. Run as Background Job

Running as PowerShell Process is a good way for testing purposes. But if you want your server permanent, then you should run it as background job. So when you restart server, PoSHServer continues to run.

If you want to start PoSHServer as a PowerShell process, just open a PowerShell console and type:

- Start-PoSHServer

That makes PoSHServer to run on that PowerShell session. If you close that PowerShell window, that will stop PoSHServer. You can get examples by typing:

- Get-Help Start-PoSHServer –full
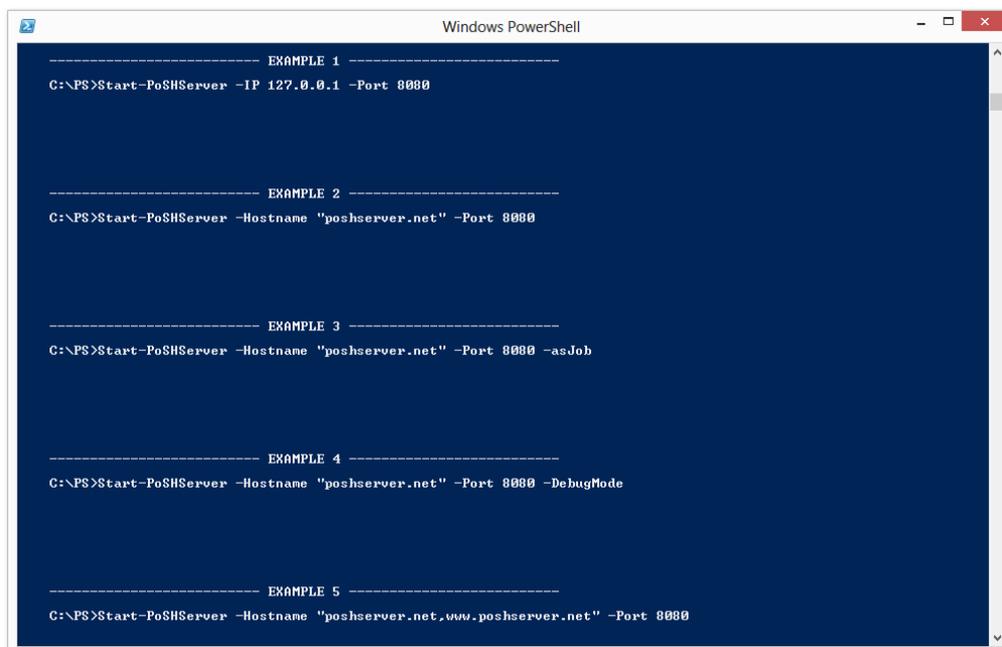
By default, PoSHServer listens port 8080. But you can change it by specifying new port. For example, if you want to publish a website called poshserver.net from port 80:

- Start-PoSHServer -Hostname "poshserver.net,www.poshserver.net" -Port 80

When you try to start PoSHServer, you may see "Please execute PoSH Server with administrative privileges. Aborting" message.
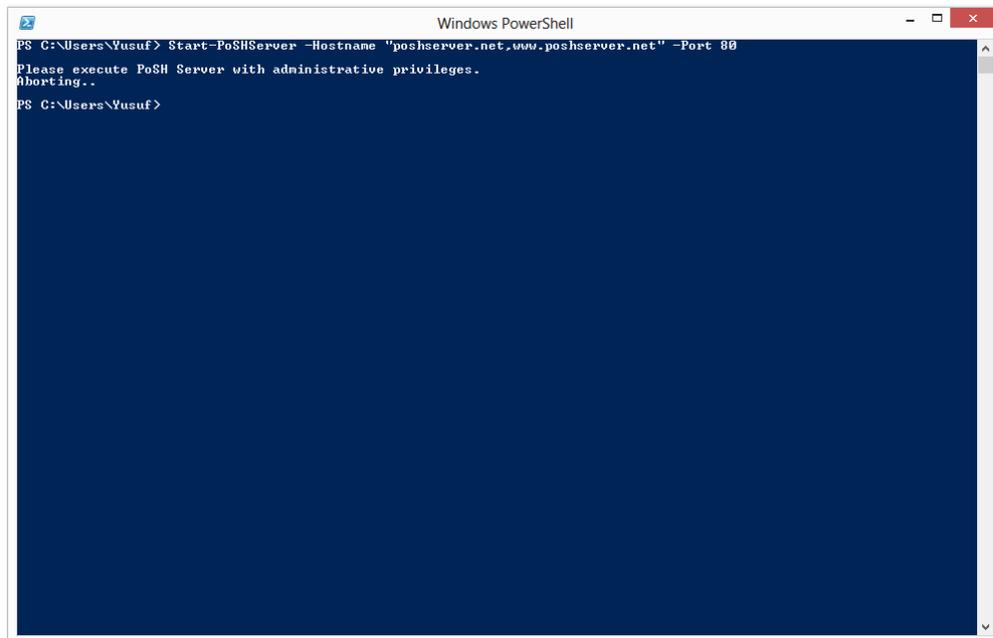
You can fix this issue by clicking "Run as Administrator" on your PowerShell shortcut.
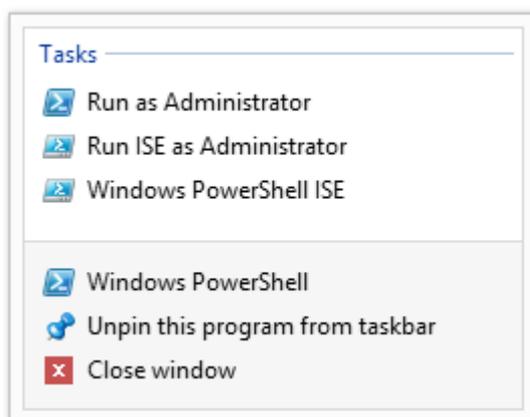
After that you can try to start PoSHServer again.

- Start-PoSHServer -Hostname "poshserver.net,www.poshserver.net" -Port 80

Figure 8: Starting PoSHServer

If you see this welcome message, you can go to your browser to browse your website.

Figure 9: Sample web site template of PoSHServer

Home directory of this sample web site:

- C:\Program Files\PoSHServer\webroot\http

There are many example files in that directory path. You can check how to use PoSHServer to publish PowerShell scripts.

If you want to change home directory of PoSHServer, you can try:

- Start-PoSHServer -Hostname "poshserver.net,www.poshserver.net" -Port 80 –
  HomeDirectory "C:\inetpub\wwwroot"

Warning: Since PoSHServer is running with Administrative privileges, you should be ensure that your website or application is secure. If your PowerShell script accepts parameters from Web, you need to filter incoming requests to block dangerous operations.
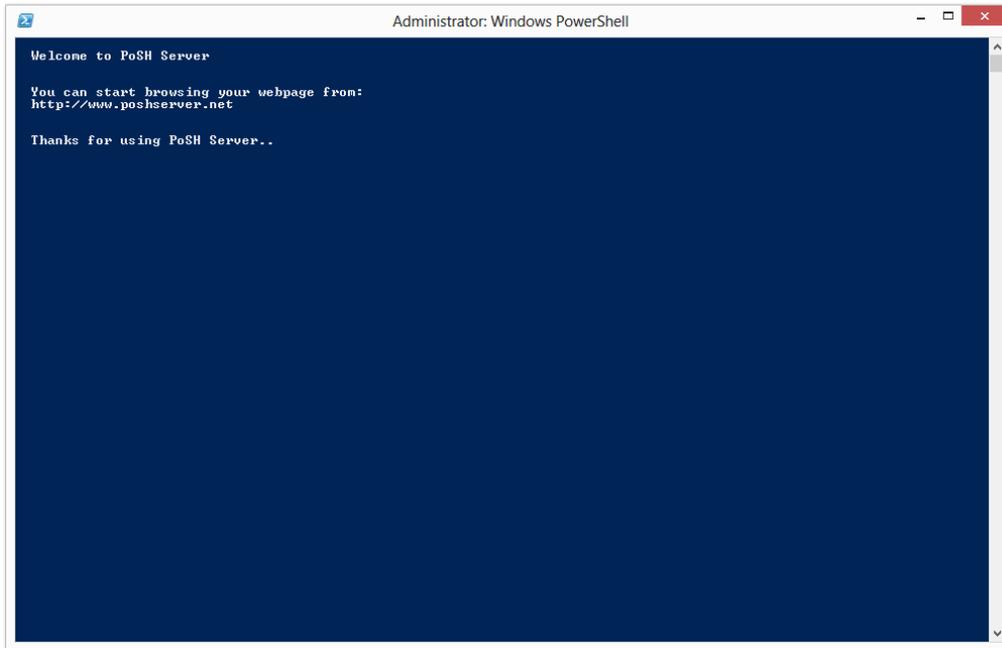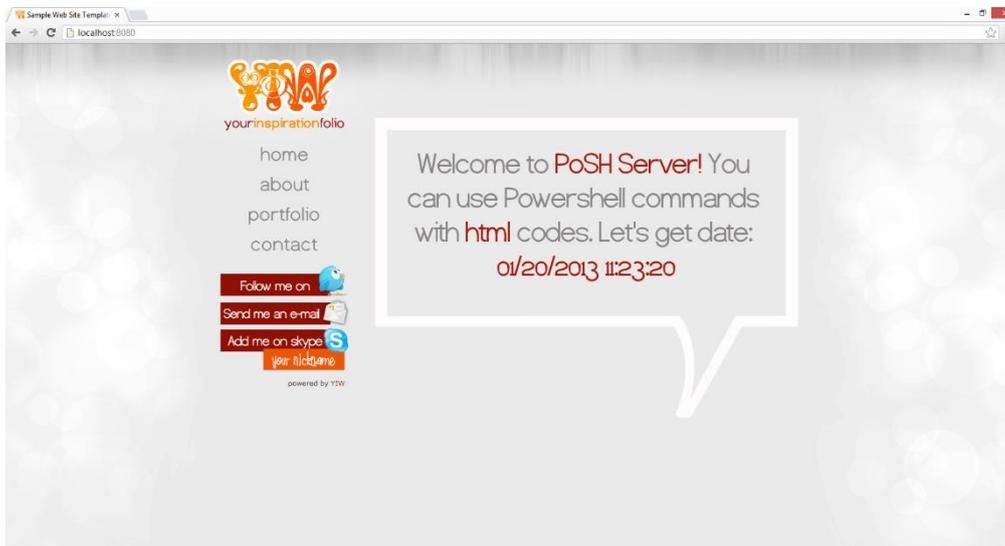
PoSHServer also accepts changing log directory:

- Start-PoSHServer -Hostname "poshserver.net,www.poshserver.net" -Port 80 –
  HomeDirectory "C:\inetpub\wwwroot" –LogDirectory "C:\inetpub\logs"

If you want to start PoSHServer with Self-Signed SSL, you should use –SSL switch:

- Start-PoSHServer -Hostname "poshserver.net,www.poshserver.net" -Port 80 –SSL –SSLIP
  "192.168.2.222" –SSLPort 443

Figure 10: Sample web site template over SSL

If you are ready to start PoSHServer as a background job, just add –asJob parameter at the end.

- Start-PoSHServer -Hostname "poshserver.net,www.poshserver.net" -Port 80 –SSL –SSLIP
  "192.168.2.222" –SSLPort 443 –asJob

When you specify -AsJob parameter, it creates a scheduled job on Windows Task Scheduler. Due to limitation of Task Scheduler, PoSHServer stores hostname, port, SSL IP, SSL port, home directory, log directory, custom config and custom job informations into a text file. Default location of this job file:

- C:\Program Files\PoSHServer\jobs

If you want to check PoSHServer job status, you need to go to Task Scheduler:
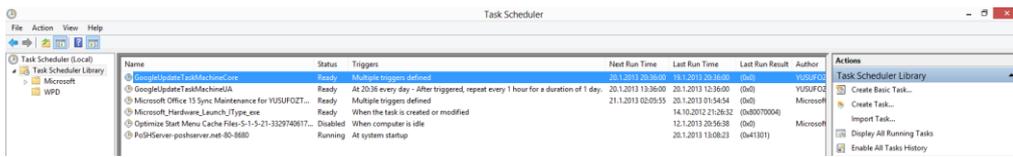
Figure 11: PoSHServer Job at Task Scheduler

You will find your PoSHServer job. Simply click "End" to stop PoSHServer.
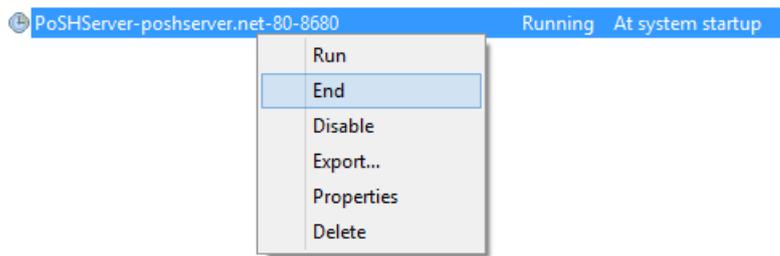
Figure 12: Stopping PoSHServer Job from Task Scheduler

If you try to create similar PoSHServer job, PoSHServer will give you a warning as "This job is already exist. You should run it from Scheduled Jobs". So you need to go Task Scheduler, stop the job first, and then delete it from Task Scheduler to remove this job.
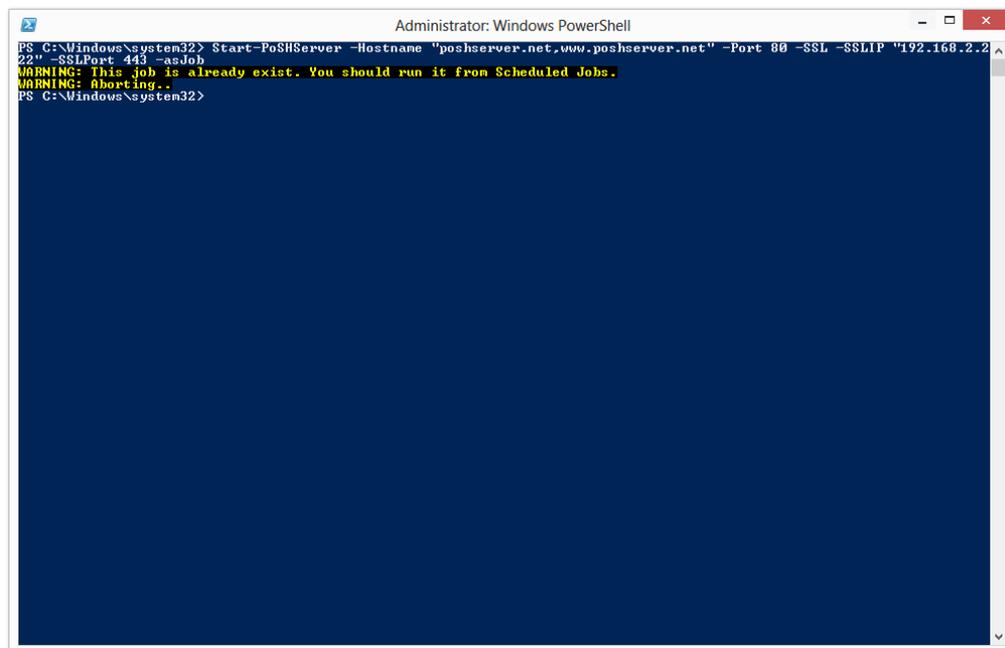
Figure 13: PoSHServer checks jobs from Task Scheduler

After that you won't get any warning. You can create same job again.

If you have a commercial SSL certificate and want to use it with PoSHServer, simply you can provide it via -SSLName switch.

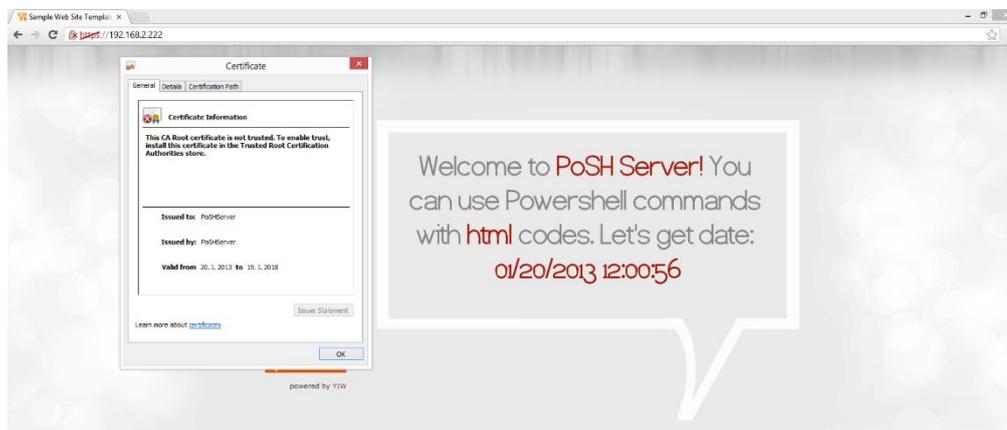- Start-PoSHServer -Hostname "poshserver.net,www.poshserver.net" -Port 80 –SSL –SSLIP "192.168.2.222" –SSLPort 443 –SSLName "poshserver.net" –asJob

You can check your existing SSL certificates from "mmc" by adding "Certificates".
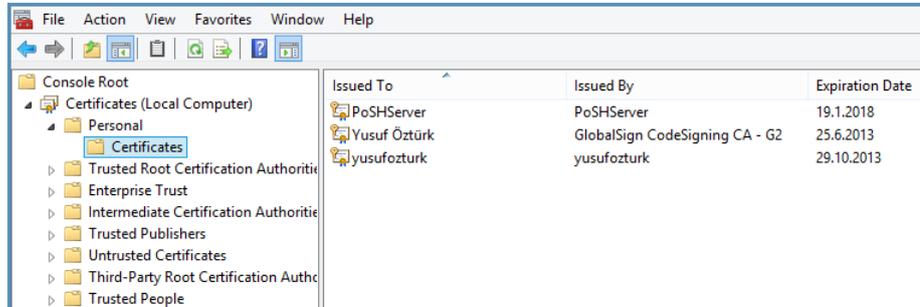
Figure 14: Managing certificates via MMC

You can also see your local web server certificates if you have IIS on your server.
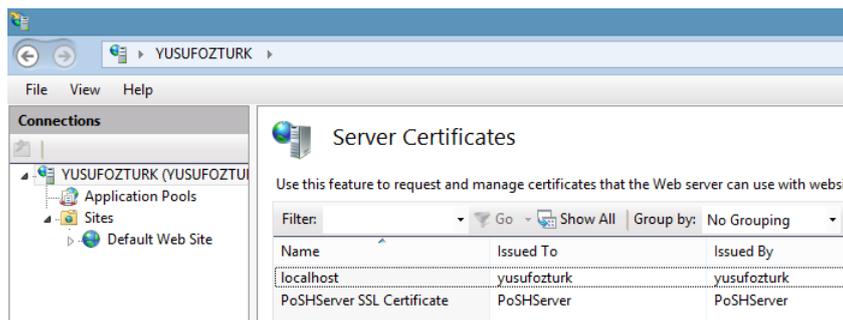
Figure 15: Managing certificates via IIS

If you provide an IP address which is not exist on your server, you will get warning like "X is not exist on your current network configuration". You need to add that IP address into your network configuration on Windows. If you are using NAT, please specify "+" as IP address.
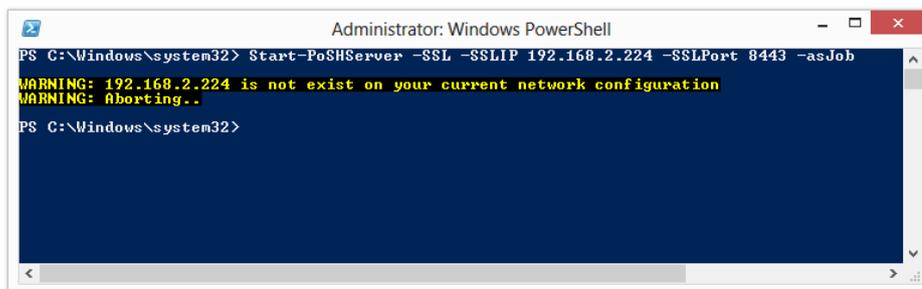
Figure 16: IP address verification

You can add multiple IP addresses for hostname and SSL.

- Start-PoSHServer -Hostname "192.168.2.222,192.168.2.223" -Port 80 –SSL –SSLIP "192.168.2.222,192.168.2.223,192.168.2.224" –SSLPort 443 –asJob

PoSHServer checks all IP addresses and if an IP address is not exist server, gives warning again.

Figure 17: Multiple IP address verification

You can always use –DebugMode switch to see debug output for troubleshooting.

How to run PoSHServer jobs with different user credentials?

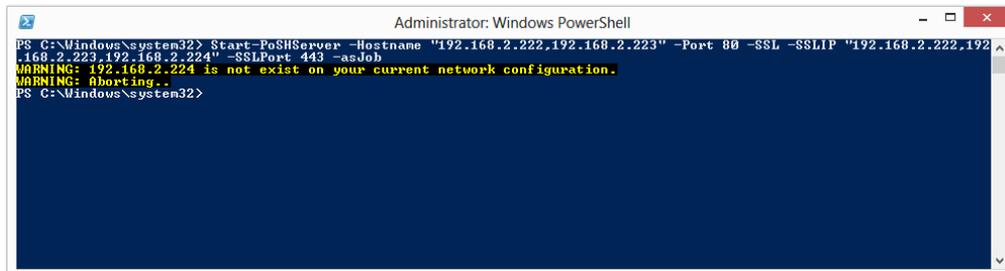PoSHServer uses "SYSTEM" account if you run it as background job. You may need to run PoSHServer with domain account to reach other servers in network or limited privileges on server. In order to do that you just need to use JobUsername and JobPassword parameters with –asJob switch.

- Start-PoSHServer -Hostname "192.168.2.222" -Port 80 –JobUsername "CONTOSO\Administrator" –JobPassword "P@ssw0rd1" –asJob

You will see PoSHServer job under scheduled jobs again. But now it's under another user credentials. If you change your domain password, you need to go to Scheduled Jobs and change user credentials.

After that, you should stop and start PoSHServer again to enable changes.

JobPassword accepts password as clear text. If you want to specify password in secure string, please use –JobCredentials switch to specify username and password as secure string.

- Start-PoSHServer -Hostname "192.168.2.222" -Port 80 –JobCredentials –asJob

PoSHServer will ask you to type username and password.

Username: DOMAIN\username

Password: ******** (SecurePassword)

How to set default document?

Default document for PoSHServer is "index.ps1". You can change it from config file.

- C:\Program Files\PoSHServer\modules\PoSHServer\modules\config.ps1
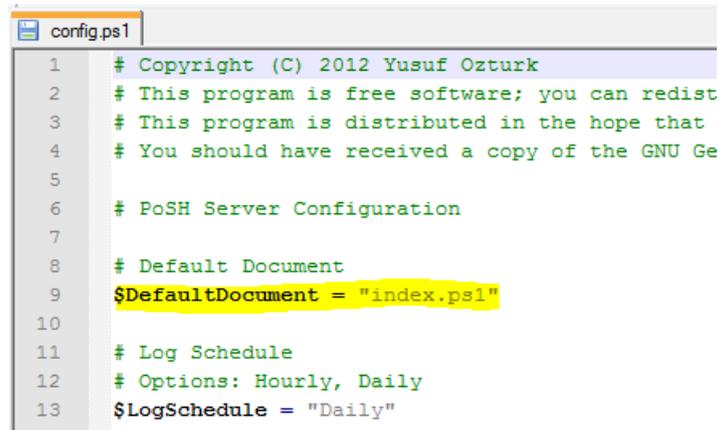
Figure 18: Setting default document

If you want to use "index.html" for default document, simply change that value like this:

- $DefaultDocument = "index.html"

How to change log schedule?

PoSHServer stores all web traffic into log files daily. If you need to parse log files hourly, you can change scheduling from config file.

- C:\Program Files\PoSHServer\modules\PoSHServer\modules\config.ps1
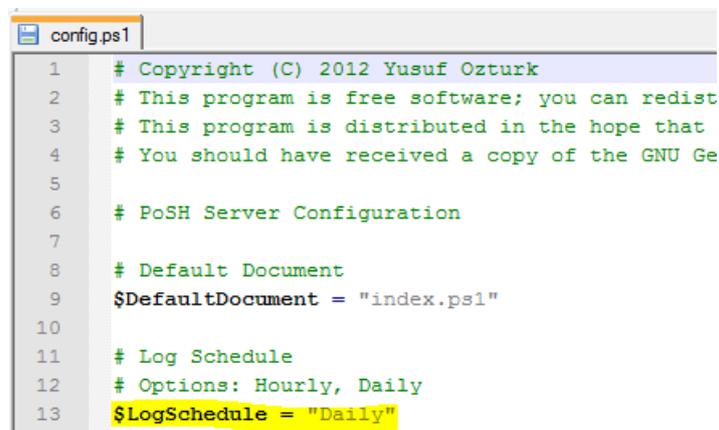
Figure 19: Setting log schedule

If you want to use "hourly" log rotating, simply change that value like this:

- $LogSchedule = "Hourly"

How to enable basic authentication?

You may need to enable basic authentication for different purposes. Go to config file and enable basic authentication first.

- C:\Program Files\PoSHServer\modules\PoSHServer\modules\config.ps1

```
15      # Basic Authentication
16      # Options: On, Off
17      $BasicAuthentication = "Off"
18
```

Figure 20: Enabling basic authentication

If you want to enable basic authentication, simple change that value like this:

- $BasicAuthentication = "On"

After you modify config file, PoSHServer starts asking username and password to guests. You should add a verification into your script.
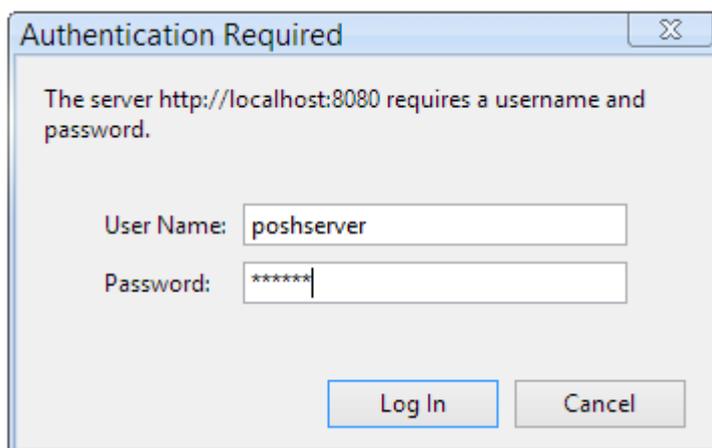
Figure 21: Basic authentication

You can get username and password via this values in your script:

- $PoSHUserName
- $PoSHUserPassword

Let's try it in a test file. I created a sample test file like this:

```
if ($PoSHUserName –eq "poshserver")
{
        if ($PoSHUserPassword –eq "123456")
        {
                $Output = "Authenticated!"
        }
        else
        {
                $Output = "Username or password is not correct!"
        }
}
else
{
        $Output = "Username or password is not correct!"
}

@"
<html>
<center>
<font color="red" size="5">$($Output)</font>
</center>
</html>
"@
```

If username and password is matched, you will see this output:
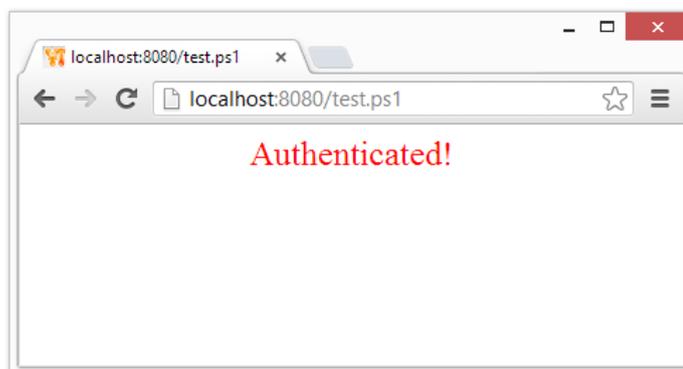
Figure 22: Basic authentication test

Otherwise you will see this output:

Session is persistent per cookie, so you need to clear browser cookies for logout.

How to enable windows authentication?

You may need to enable windows authentication for different purposes. Go to config file and enable windows authentication first.

- C:\Program Files\PoSHServer\modules\PoSHServer\modules\config.ps1


```
18
19    # Windows Authentication
20    # Options: On, Off
21    $WindowsAuthentication = "Off"
22
```

Figure 24: Enabling windows authentication

If you want to enable windows authentication, simply change that value like this:

- $WindowsAuthentication = "On"

You can get username via this value in your script:

- $PoSHUserName

How to enable directory browsing?

You can enable directory browsing on system-wide. Currently PoSHServer doesn't support directory specific config but you can change the source code if you need this support.

Go to config file and enable directory browsing first.

- C:\Program Files\PoSHServer\modules\PoSHServer\modules\config.ps1

```
22
23     # DirectoryBrowsing
24     # Options: On, Off
25     $DirectoryBrowsing = "Off"
26
```

Figure 25: Enabling directory browsing

If you want to enable directory browsing, simply change that value like this:

- $DirectoryBrowsing = "On"

If you remove default document in a directory, you will see the contents.

Figure 26: Directory browsing

How to enable IP restriction?

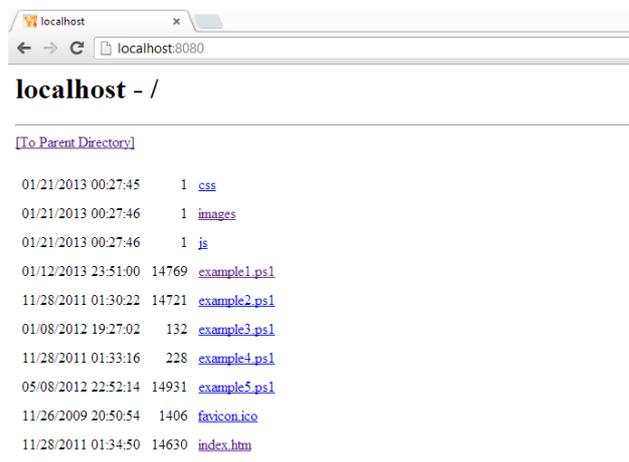You can enable IP restriction to block access from external networks. Go to config file and configure IP restriction options.

- C:\Program Files\PoSHServer\modules\PoSHServer\modules\config.ps1

```
27     # IP Restriction
28     # Options: On, Off
29     $IPRestriction = "Off"
30     $IPWhiteList = "::1 127.0.0.1"
```

Figure 27: IP restriction

If you want to enable IP restriction, simply change that value like this:

- $IPRestriction = "On"

Add your trusted networks IP addresses to whitelist. Currently PoSHServer doesn't support subnets for IP restriction. Example whitelist config:

- $IPWhiteList = "::1 127.0.0.1 192.168.2.222 192.168.2.223"

How to enable content filtering?

You can enable content filtering block access to specific file types. Go to config file and configure content filtering options.

- C:\Program Files\PoSHServer\modules\PoSHServer\modules\config.ps1

```
32    # Content Filtering
33    # Options: On, Off
34    $ContentFiltering = "Off"
35    $ContentFilterBlackList = "audio/mpeg video/mpeg"
```

Figure 28: Content filtering

If you want to content filtering, simply change that value like this:

- $ContentFiltering = "On"

You can edit $ContentFilterBlacklist to block specific file types.

How to enable PHP support?

You should install PHP before using it with PoSHServer. You can download and install PHP via Microsoft WebPI on your server. After that you can specify full path of PHP executable file. Go to:

- C:\Program Files\PoSHServer\modules\PoSHServer\modules\config.ps1

```
37    # PHP Cgi Path
38    $PHPCgiPath = "C:\Program Files (x86)\PHP\v5.3\php-cgi.exe" # WebPI Location
```

Figure 29: PHP Support

You should check the path if it is exist. If your PHP path is different, you should change this value:

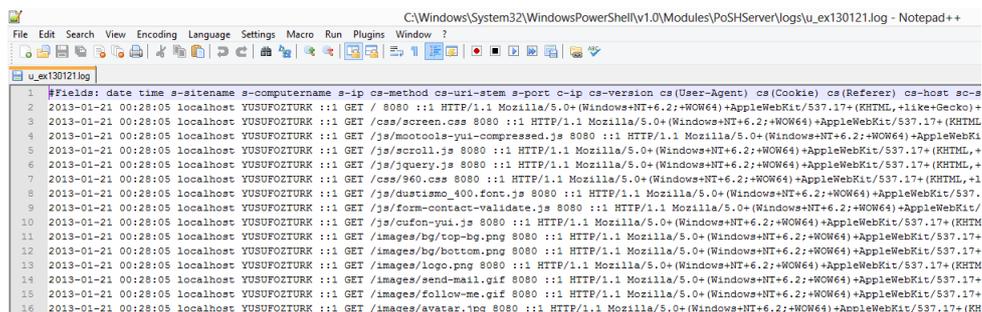- $PHPCgiPath = "C:\Program Files (x86)\PHP\v5.3\php-cgi.exe"

After that PoSHServer will be able to run your PHP scripts. PHP support is limited to GET requests. PoSHServer doesn't support "POST" at the moment. Also you can't use .htaccess and .htpasswd files in PoSHServer. You should convert them to .ps1 files. For example, you will be able to run Wordpress on PoSHServer, but you can't add any post or comments into it because lack of POST support.

Understanding advanced logging and log parser

Logging is designed to be compatible with IIS logging. So you can use any IIS log parser script to analyze PoSHServer logs. Default log directory:

- C:\Program Files\PoSHServer\webroot\logs

You will notice that logging is almost same with IIS.

Figure 30: Contents of log file

You can use AWStats, Webalizer or PoSHStats to analyze these logs. But PoSHServer also has in-built log parser to analyze your log files. Go to PowerShell console and type:

- Start-PoSHLogParser –LogPath C:\Program Files\PoSHServer\logs\u_ex130121.log >> C:\LogOutput.txt

Output file will be like:

```
date          : 2013-01-21
time          : 00:28:05
ssitename     : localhost
scomputername : YUSUFOZTURK
sip           : ::1
csmethod      : GET
```

```
csuristem    : /css/screen.css
sport        : 8080
cip          : ::1
csversion    : HTTP/1.1
csUserAgent  :
Mozilla/5.0+(Windows+NT+6.2;+WOW64)+AppleWebKit/537.17+(KHTML,+like+Gecko)+Chro
me/24.0.1312.52+Safari/5
             37.17
csCookie     : -
csReferer    : http://localhost:8080/
cshost       : ::1:8080
scstatus     : 200
```

How to get cookie value?

PoSHServer adds a cookie for each client. So multiple users can use same PoSHServer session. If you need to get cookie value, just use this:

- $PoSHCookies

That will give you PoSHSessionID for current user session. You can change cookie code from:

- C:\Program Files\PoSHServer\modules\PoSHServer\PoSHServer.ps1
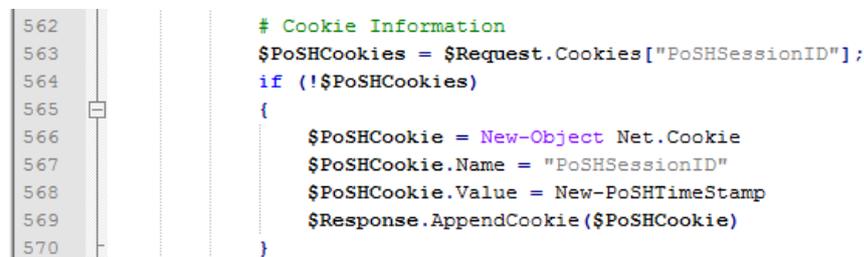
```
562        # Cookie Information
563        $PoSHCookies = $Request.Cookies["PoSHSessionID"];
564        if (!$PoSHCookies)
565        {
566            $PoSHCookie = New-Object Net.Cookie
567            $PoSHCookie.Name = "PoSHSessionID"
568            $PoSHCookie.Value = New-PoSHTimeStamp
569            $Response.AppendCookie($PoSHCookie)
570        }
```

Figure 31: Cookie codes

How to fetch GET values?

For example, this is your GET:

- http://localhost:8080/index.ps1?name=Yusuf&surname=Ozturk

Then you can fetch "GET" values like this:

- $PoSHQuery.name

- $PoSHQuery.surname

This will gives you values directly.

How to fetch POST values?

For example, this is your POST:

- name=Yusuf&surname=Ozturk

Then you can fetch "POST" values like this:

- $PoSHPost.name

- $PoSHPost.surname

This will gives you values directly.

How to use jscript on PoSHServer?

There is no jscript limitation on PoSHServer.
However you need to use escape characters to make it work.

If you need to use something like this:

- $(function() { $.ajax({

Then you should change your codes like this:

- `$(function() {`$.ajax({

` is a escape character in PowerShell.

How to add custom MimeTypes?

PoSHServer stores MimeTypes information into:

- C:\Program Files\PoSHServer\modules\PoSHServer\modules\functions.ps1

You can change or add extra mime-types to here:

```
205     # Get Mime Types
206     function Get-MimeType
207   □{
208    param ($Extension)
209
210        switch ($Extension)
211   □    {
212            .ps1 {"text/ps1"}
213            .psxml {"text/psxml"}
214            .psapi {"text/psxml"}
215            .posh {"text/psxml"}
216            .html {"text/html"}
217            .htm {"text/html"}
218            .php {"text/php"}
219            .css {"text/css"}
220            .jpeg {"image/jpeg"}
221            .jpg {"image/jpeg"}
222            .gif {"image/gif"}
223            .ico {"image/x-icon"}
224            .flv {"video/x-flv"}
225            .swf {"application/x-shockwave-flash"}
226            .js {"text/javascript"}
227            .txt {"text/plain"}
228            .rar {"application/octet-stream"}
229            .zip {"application/x-zip-compressed"}
230            .rss {"application/rss+xml"}
231            .xml {"text/xml"}
232            .png {"image/png"}
233            .mpg {"video/mpeg"}
234            .mpeg {"video/mpeg"}
235            .mp3 {"audio/mpeg"}
236            .woff {"application/x-font-woff"}
237            default {"text/html"}
238        }
239   └}
```

Figure 32: Custom MimeTypes

How to use config file per Web Site?

You just need to put your config.ps1 file into HomeDirectory. So PoSHServer will read that configuration first, instead of default config.ps1 file. So you can create different config files for web sites. You can enable Basic Authentication on only one web site by using this feature.

How to add custom config file?

It's possible to add your own config files into PoSHServer. You may need to add additional config files due to requirements of your own software. For example, SetLinuxVM uses custom config file for persistent key and values.

Example config file of SetLinuxVM:

```
# Import required modules
$ImportSetLinuxVM = Import-Module SetLinuxVM

# SetLinuxVM VM hosts config
$VMHostsConfig = "$HomeDirectory\config\vmhosts.config"
$VMHostsList = Search-LinuxVMHost
Clear-Content -Path $VMHostsConfig
foreach ($VMHost in $VMHostsList)
{
        Add-Content -Path $VMHostsConfig -Value $VMHost.VMHost
}

# SetLinuxVM RestAPI config
$RestAPIKeyConfig = "$HomeDirectory\config\restapikey.config"
Clear-Content -Path $RestAPIKeyConfig
$NewAPIKey = Get-Random
Add-Content -Path $RestAPIKeyConfig -Value $NewAPIKey
```

If you also need to import a custom config file like this, you can start PoSHServer like this:

- Start-PoSHServer -CustomConfig "C:\config.ps1" –asJob

```
366        # PoSH Server Custom Config
367        if ($CustomConfig)
368        {
369            . $CustomConfig
370        }
```

Figure 33: Config config code

Your config file will be imported into PoSHServer session as the command above.

- C:\Program Files\PoSHServer\modules\PoSHServer\PoSHServer.ps1

How to add custom config file to child processes?

When you use PoSHServer with child processes, your custom config file won't be active in child processes since they are new PowerShell runspaces. So you should also import your config file into child processes with –CustomChildConfig switch. You may need to import different config files into main runspace and child runspaces. For example, SetLinuxVM uses different custom config file in child processes.

If you need to import a custom child config file, you can start PoSHServer like this:

- Start-PoSHServer -CustomChildConfig "C:\childconfig.ps1" -asJob

Your config file will be imported into PoSHServer child processes.

How to add custom scheduled jobs?

PoSHServer has a feature called "background jobs" to run your scripts in a scheduled job. Using this feature, you can schedule any process in any time as long as PoSHServer is running. PoSHStats uses this feature to measure Hyper-V VM usages every an hour and a day.

Default job interval is 5 minutes.
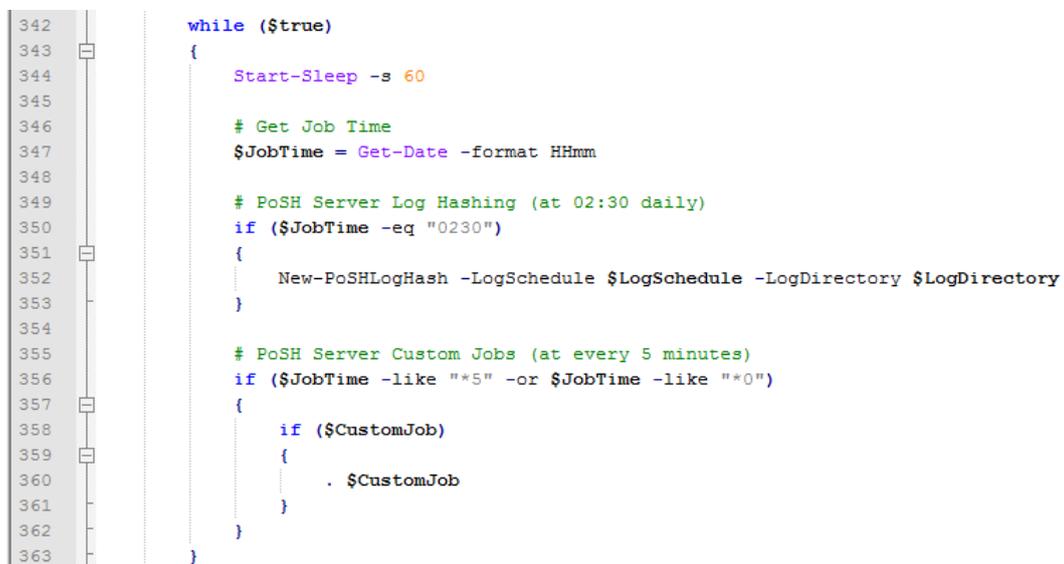
```
342     while ($true)
343     {
344         Start-Sleep -s 60
345
346         # Get Job Time
347         $JobTime = Get-Date -format HHmm
348
349         # PoSH Server Log Hashing (at 02:30 daily)
350         if ($JobTime -eq "0230")
351         {
352             New-PoSHLogHash -LogSchedule $LogSchedule -LogDirectory $LogDirectory
353         }
354
355         # PoSH Server Custom Jobs (at every 5 minutes)
356         if ($JobTime -like "*5" -or $JobTime -like "*0")
357         {
358             if ($CustomJob)
359             {
360                 . $CustomJob
361             }
362         }
363     }
```

Figure 34: Custom jobs

If you need to run your custom job at every 5 mins, you should start your PoSHServer like this:

- Start-PoSHServer -CustomJob "C:\jobs\custom.ps1" –asJob

How to change custom scheduled job interval?

If you enable custom scheduled jobs on PoSHServer, default interval is 5 minutes. You can change job intervals with following values:

1: 1 minute
5: 5 minutes
10: 10 minutes
20: 20 minutes
30: 30 minutes
60: 60 minutes

So if you want to change scheduled job interval with 10 minutes, you should use CustomJobSchedule parameter:

- Start-PoSHServer -CustomJob "C:\jobs\custom.ps1" –CustomJobSchedule 10 –asJob

Minimum interval is 1 minute, Maximum interval is 1 hour. If your script requires different job intervals, you should write a custom scheduling in your custom job script.

PoSHServer has a parameter validation for CustomJobSchedule, so you can't use different values than 1, 5, 10, 20, 30 and 60. If you try to use different value, PoSHServer will give error:

```
Start-PoSHServer : Cannot validate argument on parameter 'CustomJobSchedule'. The argument "22" does not belong to the
set "1,5,10,20,30,60" specified by the ValidateSet attribute. Supply an argument that is in the set and then try the co
mmand again.
At line:1 char:37
+ Start-PoSHServer -CustomJobSchedule 22
+                                     ~~
    + CategoryInfo          : InvalidData: (:) [Start-PoSHServer], ParameterBindingValidationException
    + FullyQualifiedErrorId : ParameterArgumentValidationError,Start-PoSHServer
```

How to increase child processes for better multithreading?

PoSHServer supports 3 child processes by default. But this is not a limitation. You can increase the number of child processes any time. You just need to go and modify:

- C:\Program Files\PoSHServer\modules\PoSHServer\PoSHServer.ps1

```
705          # Invoke PoSH Server Multithread Process - Thread 1
706          Invoke-AsyncHTTPRequest -ScriptBlock $ScriptBlock -Listener $Li
707
708          # Invoke PoSH Server Multithread Process - Thread 2
709          Invoke-AsyncHTTPRequest -ScriptBlock $ScriptBlock -Listener $Li
710
711          # Invoke PoSH Server Multithread Process - Thread 3
712          Invoke-AsyncHTTPRequest -ScriptBlock $ScriptBlock -Listener $Li
```

Figure 35: Child processes for multithreading

You can increase multithreads by adding extra thread commands into part above. Using more child processes causes more memory and cpu utilization on your server.

Standalone PoSHServer Exe Package

After v3.4, PoSHServer includes a standalone .exe file for quick and easy test drives. In packages directory:

C:\Program Files\PoSHServer\packages

You will see:

- PoSHServer-Standalone.exe: Main executable file

Just copy exe file into any http directory and start PoSHServer-Standalone.exe with run-as administrator. That directory will be home directory of PoSHServer.

If you don't want to put exe file into http directory, you can move it upper level. Just keep \http directory like this structure:

- C:\Website
    - o PoSHServer-Standalone.exe
    - o http

PoSHServer will check \http directory and use it as home directory.

If you don't provide any config.ps1 file, PoSHServer uses default configuration for server initialization.

```
# Default Document
$DefaultDocument = "index.ps1"

# PHP Cgi Path
$PHPCgiPath = ($env:PATH).Split(";") | Select-String "PHP"
$PHPCgiPath = [string]$PHPCgiPath + "\php-cgi.exe"
```

If you want to change default document or PHP-cgi.exe location, then you should put a config.ps1 file into same directory with PoSHServer-Standalone.exe file.

After config file, directory structure will be like:

- C:\Website
    - o PoSHServer-Standalone.exe
    - o Config.ps1
    - o http

If you do any changes on config file, you should restart PoSHServer-Standalone.exe file to apply changes. Otherwise PoSHServer won't detect changes.

Uninstallation

If you use PoSHServer as a background job, first go to task scheduler and "end" PoSHServer task.
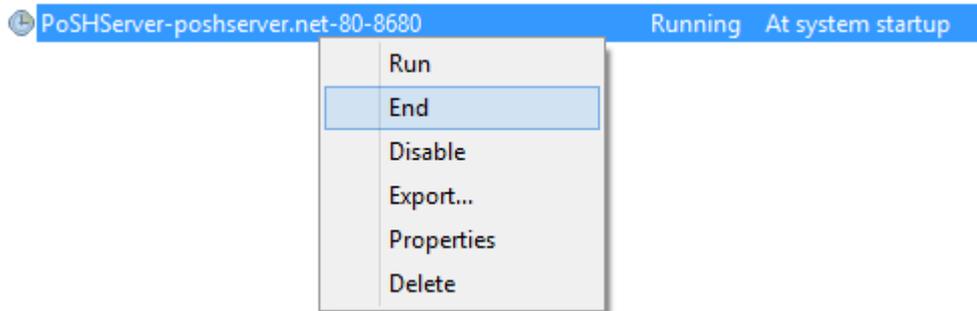
Figure 36: Ending PoSHServer task

And then delete PoSHServer task. If you don't stop running PoSHServer task, then PoSHServer setup won't be able to remove PoSHServer directories completely.

Now go to "Add/Remove Programs" to uninstall PoSHServer completely. Select PoSHServer in the program list and click "Uninstall" from top menu.
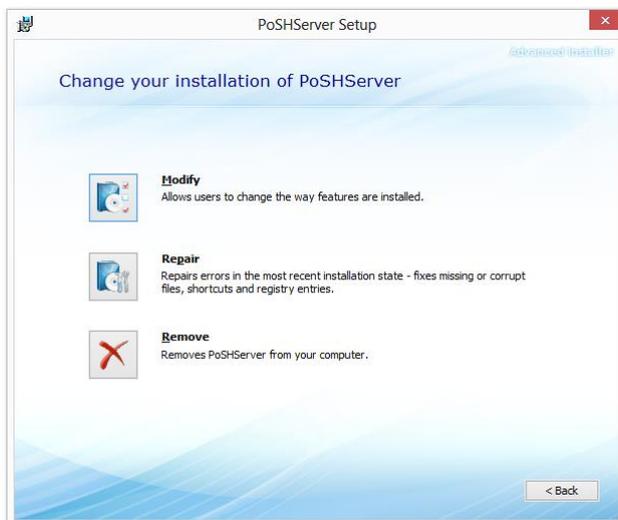

Figure 37: Removing PoSHServer

You can start removing PoSHServer by selecting remove option.