

# PowerShell Start Process and Get its PID

I needed to start a process on my machine and get its procesID in PowerShell to monitor that it does not run longer than x min. This sounded simple, but was a bit of a challenge with PowerShell (mostly that I could not get the Start-Process cmdlet to return this information).

After playing on Google I stumbled upon the following links to MSDN and it hit me, PowerShell is based on .net, and there are tons of scripts on the net for starting processes with .net. With the following 3 links I was able to get something working.

- <http://msdn.microsoft.com/en-us/library/system.diagnostics.process.aspx>
- <http://msdn.microsoft.com/en-us/library/system.diagnostics.processstartinfo.aspx>
- [http://msdn.microsoft.com/en-us/library/system.diagnostics.processwindowstyle\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/system.diagnostics.processwindowstyle(VS.85).aspx)

After a bit of playing I was able to come up with the following script.

```
$ProcessStartInfo = New-Object System.Diagnostics.ProcessStartInfo "notepad.exe"
$Process           = [System.Diagnostics.Process]::Start($ProcessStartInfo)
$Process.Id
```

Pretty nifty eh, but still not what I wanted. In my case I need to start the process hidden as I do not want users to interact with it, so now I need to figure out how to start the process hidden. From the 3<sup>rd</sup> link I know that I can set the process window style to Hidden, but how do I do this?

I start by piping the \$ProcessStartInfo to the Get-Member cmdlet, and behold I now know the value of the WindowStyle property.

Name	MemberType	Definition
----	-----	-----
Equals	Method	bool Equals(System.Object obj)
GetHashCode	Method	int GetHashCode()
GetType	Method	type GetType()
ToString	Method	string ToString()
Arguments	Property	System.String Arguments {get;set;}
CreateNoWindow	Property	System.Boolean CreateNoWindow {get;set;}
Domain	Property	System.String Domain {get;set;}
EnvironmentVariables	Property	System.Collections.Specialized.StringDict...
ErrorDialog	Property	System.Boolean ErrorDialog {get;set;}
ErrorDialogParentHandle	Property	System.IntPtr ErrorDialogParentHandle {ge...
FileName	Property	System.String FileName {get;set;}
LoadUserProfile	Property	System.Boolean LoadUserProfile {get;set;}
Password	Property	System.Security.SecureString Password {ge...
RedirectStandardError	Property	System.Boolean RedirectStandardError {get...
RedirectStandardInput	Property	System.Boolean RedirectStandardInput {get...

RedirectStandardOutput	Property	System.Boolean RedirectStandardOutput {ge...
StandardErrorEncoding	Property	System.Text.Encoding StandardErrorEncodin...
StandardOutputEncoding	Property	System.Text.Encoding StandardOutputEncodi...
UserName	Property	System.String UserName {get;set;}
UseShellExecute	Property	System.Boolean UseShellExecute {get;set;}
Verb	Property	System.String Verb {get;set;}
Verbs	Property	System.String[] Verbs {get;}
WindowStyle	Property	System.Diagnostics.ProcessWindowStyle Win...
WorkingDirectory	Property	System.String WorkingDirectory {get;set;}

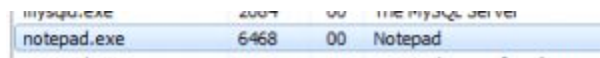
So let's set it.

```
$ProcessStartInfo.WindowStyle = "Hidden"
```

I run this and no errors, this is promising.



A quick check of task manager confirms that I have successfully started the process, and have the correct procesID.



That's all there is to it, albeit it took me a good hour to figure out L. Now I can continue with my script.

Here is the complete code listing to start a process and get its processID.

```
$ProcessStartInfo = New-Object System.Diagnostics.ProcessStartInfo "notepad.exe"
$ProcessStartInfo.WindowStyle = "Hidden"
$Process = [System.Diagnostics.Process]::Start($ProcessStartInfo)
$Process.Id
```