

# Get-Started: ii Captain! Using the Invoke-Item CmdLet

✘ A nifty little PowerShell CmdLet to make use of is **Invoke-Item**, which also has the alias **ii**. This CmdLet can be used against files, URLs and other documents and items to invoke the default action against them.

An example where we would want to use this is where we generate some output into a file such as a .txt file, and then we want it to be rendered to the screen using the editor for the .txt extension which is most likely Notepad.exe on your system.

In this case, let's use a simple one-liner to output the path variable (\$env:path) to a text file.

```
$env:path | Out-File MyPath.txt
```



Now that we have the file created, we simply run the Invoke-Item against the file to launch it using the default Open action based on the file type association.

```
Invoke-Item .MyPath.txt
```



The result of the **Invoke-Item** command is that you will see a Notepad windows launch (or whatever your associated application happens to be).



This was a very simple example, but the concept is identical for any registered file type in your system. When running as an interactive script, a very popular purpose for the **Invoke-Item** CmdLet is to launch Internet Explorer to view an HTML file which was created by some other process.

While this is a handy little tool in our PowerShell toolbox, you may find that you want to move to the **Invoke-Expression** if you require more complex command lines and parameters to be available. I'll be putting together a post on that CmdLet in the future which will go into much further detail.

The CmdLet supports relative and literal path references and requires double-quotes for long file and path names where spaces are in the path.

The Get-Help for the Invoke-Item is as follows:



The full TechNet help for the Invoke-Item can be found here:  
<http://technet.microsoft.com/en-us/library/hh849794.aspx>

Happy invoking!