

Checking for the Existence of a String Value

What can you do with the `Select-String` cmdlet? Well, one thing you can do is determine whether or not a specific string value exists in a text file. For example, suppose the file `C:\Scripts\Test.txt` is a log file that contains the following information:

```
[crayon-6a9d5deb3b955642884011/]
```

You'd like to be able to quickly scan the contents of the file and see whether the word `Failed` appears anywhere. If it does, that means one of your operations failed; if it doesn't, that means all of your operations succeeded. (And yes, seeing as how we're talking about failed operations we do hope you're not a surgeon.) Here's how you can do that:

```
[crayon-6a9d5deb3b95b113760606/]
```

What we're doing here is using the `Get-Content` cmdlet to retrieve the contents of the file `C:\Scripts\Test.txt`. We're then piping those contents to the `Select-String` cmdlet and asking `Select-String` to search for the target string. By adding the `-quiet` parameter we get back a `True` if the string is found and nothing if the string is not found. If we leave off the `-quiet` parameter then Windows PowerShell returns each line in the text file that includes the target string:

```
[crayon-6a9d5deb3b95e630264976/]
```

Another `Select-String` parameter that you might find useful is `-casesensitive`, which performs a case-sensitive search of, in this case, the text file. This particular command will return nothing, meaning that the target string `Failed` could not be found:

[crayon-6a9d5deb3b95f185371606/]

Why couldn't Failed be found in the text file? That's easy: based on letter case, there's no such string in the file. The file contains the string failed with a lowercase f, while the target search string is Failed with an uppercase F. If you change the target string to failed (or remove the -casesensitive parameter) the command returns True.