

PowerShell ABC's - H is for Here-String



Joe Pruitt, 2008-30-12

Welcome to this addition of the PowerShell ABC's where you'll find 26 posts detailing a component of the PowerShell scripting language, one letter at a time. Today's letter is the letter "H". For "H", I'm going to talk about a feature that PowerShell Guru [Jeffrey Snover](#) loves talking about each time I see him: Here-Strings.



Before we get to *here-strings*, it's probably a good idea to go over the different kinds of string literals in PowerShell. There are actually 4 different kinds of string literals - single-quoted strings, double-quoted strings, single-quoted here-strings, and double-quoted here-strings. All of these types are stored in the same internal format of 16-bit Unicode characters and are directly implemented by the .NET System.String type.

Some additional features that strings have due to their System.String base, is that they can be arbitrarily long and that they are *immutable*, meaning that the contents can be copied but cannot be changed.

Single and Double-Quoted strings

Literal strings can contain any character with the exception of an unquoted closing quote character.

For double-quoted strings, to embed a closing quote character, you have to either quote it with a *backtick* character (```) or provide two adjacent quotes which become a single literal quote in the string. Also, for variable substitutions, only double-quote strings will process those substitutions.

In single-quoted strings, providing double quotes is the only way to embed a literal quote in the string. In single-quoted strings, the backtick is not treated as a special character so it cannot be used for escaping newlines or other special characters. In single-quoted strings, variable substitution is not performed.

Here are some examples of embedded quotes and variable substitutions:

```
PS> $s = "My name is `\"Joe\"\", what's yours?"
PS> $s
My name is "Joe", what's yours?
PS> "He said: $s"
He said: My name is "Joe", what's yours?
PS> 'He said: $s'
He said: $s
```

Here-Strings

A here-string is used to embed large chunks of text inline in a script into a single string literal. This is very similar to a feature in C#, but the difference is that in PowerShell the here-string begins at the end of a line and the terminating sequence must be at the beginning of a line. In C#, the string terminates with the first non-doubled closing quote.

```
PS> $a = @"
one
two
"@
PS> $a
one
two
```

Here strings have the following special quoting rules: First, here-strings begin with @ and here-strings end with @. The content of the here-string is all of the lines between the beginning and ending quotes, but not the lines the quotes are on. Because of these rules, other special characters such as quotes that cause problems with regular strings have no problems.

Here is an example from [Windows PowerShell in Action by Bruce Payette](#):

```
PS> $a = @"
One is "1"
Two is '2'
Three is $(2+1)
The date is "$(Get-Date)"
"@ + "A trailing line"
PS> $a
One is "1"
Two is '2'
Three is 3
The date is "12/30/2008 11:31:35"A trailing line
```

Here-strings come in very useful when writing long scripts that contain text output you would normally have to write on multiple output statements.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com