PowerShell ABC's - Z is for zh-CHT

Joe Pruitt, 2009-09-02

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. For today's letter of "Z", I'll discuss PowerShell's internationalization features and it's support for country cultures like **zh-CHT**.

PowerShell 2.0 added features that make script internationalization relatively simple to implement. The script internationalization features query the user interface culture of the operating system during execution, import the associated translated text strings, and allows you to display them to the user.

To support international text, PowerShell 2.0 includes the following features:

- A DATA section that separates text strings from code instructions.
- New $PsCulture and $PsUICulture automatic variables. $PsCulture stores the name of the UI language used on the system for elements such as date, time, and currency, while the $PsUICulture variable stores the name of the UI language used on the system for user interface elements.
- The ConvertFrom-StringData cmdlet that converts text strings into a dictionary-like hash table.
- The new .psd1 file type that is used to store text strings in language-specific subdirectories of the script directory.
- The Import-LocalizedData cmdlet that imports translated text strings for a specified language into a script at runtime.

Implementing Internationalization

Let's start with the following script that displays a hallway converstation

```
#File foo.ps1
Write-Host "Hi";
Write-Host "How are you?";
Write-Host "I'm great!";
Write-Host "Goodbye";
```

running this script results in the following output:

```
PS D:\dev\powershell\i18n> .\foo.ps1
Hi
How are you?
I'm great!
Goodbye
```

Setting up the internationalization of strings in your scripts involves the following steps:

1. Create subdirectories for each language you would like to support. In my example, I'll support English (en-US) and Traditional Chinese (zh-CHT).

```
PS D:\dev\powershell\i18n> New-Item -ItemType directory en-US > $null
PS D:\dev\powershell\i18n> New-Item -ItemType directory zh-CHT > $null
```

2. In each subdirectory, create a string table for the associated language in a file with the same name as the script and the extension of **.psd1**. The ConvertFrom-StringData cmdlet will convert a string containing one or more "name=value" pairs into a hash table.

```
#File en-US\foo.psd1
# culture="en-US"
# culture="en-US"
ConvertFrom-StringData @'
    S_HI = Hi
    S_HOWAREYOU = How are you?
    S_IMGREAT = I'm great!
    S_GOODBYE = Goodbye
'@

#File zh-CHT\foo.psd1
# culture="en-US"
ConvertFrom-StringData @'
    S_HI = Hi (In Chinese)
    S_HOWAREYOU = How are you? (In Chinese)
    S_IMGREAT = I'm great! (In Chinese)
    S_GOODBYE = Goodbye (In Chinese)
'@
```

Since I'm working on a English based version of Windows, I'll leave it as an exercise to the user to add the actual translated strings into the zh-CHT\foo.psd1 language file.

3. Load the string table with the Import-LocalizedData cmdlet and reference the strings by their key names in the hash table.

```
# foo.ps1
param([string]$uiCulture = ($PsUiCulture));
Write-Host "$uiCulture";
Import-LocalizedData -bindingVariable stringTable -UICulture $uiCulture;

Write-Host $stringTable.S_HI;
Write-Host $stringTable.S_HOWAREYOU;
Write-Host $stringTable.S_IMGREAT;
Write-Host $stringTable.S_GOODBYE;

Write-Host "All Localization Table Entries";
$stringTable
```

By default the Import-LocalizedData cmdlet will query the $PsUiCulture automatic variable but I've added the ability to pass the culture code into the script via the uiCulture parameter. If one is not supplied, the value of the $PsUiCulture variable is used.

The Import-LocalizedData cmdlet will store the hash table in the $stringTable variable. You can act on that as you would any other hash table, except that the contents will be the language specific strings.

```
PS D:\Dev\PowerShell\i18n> .\foo.ps1
en-US
Hi
How Are You?
I'm great!
Goodbye
All Localization Table Entries
```

```
Name                           Value
----                           -----
S_GOODBYE                      Goodbye
S_HI                           Hi
S_IMGREAT                      I'm great!
S_HOWAREYOU                    How Are You?

PS D:\Dev\PowerShell\i18n> .\foo.ps1 zh-CHT
zh-CHT
Hi (In Chinese)
How Are You? (In Chinese)
I'm great! (In Chinese)
Goodbye (In Chinese)
All Localization Table Entries

Name                           Value
----                           -----
S_GOODBYE                      Goodbye (In Chinese)
S_HI                           Hi (In Chinese)
S_IMGREAT                      I'm great! (In Chinese)
S_HOWAREYOU                    How Are You? (In Chinese)
```

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