FortyPoundHead KNOWLEDGEBASE HOME CONTACT DISCLAIMER PRIVACYPOLICY TERMS OF USE

Resolving IP to Hostname with PowerShell

By dwirch | 2016-01-15 0 Comment

Getting the hostname from an IP address (or vice versa) is no great magic, and can be done easily with PowerShell, by simply leveraging .Net to do the work.

We've all done something like this to get resolve an IP address:

[System.Net.Dns]::GetHostEntry("69.69.95.133").HostName

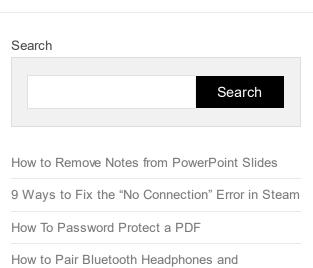
Or this to get an IP address from a hostname:

[System.Net.DNS] :: GetHostAddresses ("www.fortypoundhead.com"). IPAddressToString [System.Net.DNS] :: GetHostAddressEduction [System.Net.DNS] :: GetHostAddressEduc

But did you ever notice that sea of red when you specify an invalid IP address or hostname? When you do that, you'll get a big,red error message similar to the following:

I recently got asked how that nonsense could be avoided. Well, the answer is to simply encapsulate the process in a function that wraps up the .Net method to get the data back. It's pretty simple, and here is the code:

```
[CmdletBinding()]
Param(
[Parameter(Mandatory=$True,Position=1)]
[string]$IPAddress
# noise if we fail
$ErrorActionPreference = "silentlycontinue"
$Result = $null
# Pass the IP to .Net for name resolution.
$result = [System.Net.Dns]::gethostentry($IPAddress)
# process the results
If ($Result)
{
$MyResult = [string]$Result.HostName
}
Else
{
```



How to Make Windows 11 Look Like Windows 10

Earphones

Manage Consent

X

To provide the best experiences, we use technologies like cookies to store and/or access device information. Consenting to these technologies will allow us to process data such as browsing behavior or unique IDs on this site. Not consenting or withdrawing consent, may adversely affect certain features and functions.

Accept

Deny

View preferences

<u>Privacy Policy</u> <u>Privacy Policy</u> <u>Privacy Policy</u>

```
$MyResult = "unresolved"
}
# Send it to the output
$MyResult
Alternatively, you could also plug this directly into a standalone script. For example, lets say that you
have a list of IP addresses in a file that you want to resolve to host names. No problem:# Get list
from file, initialize empty array
$ListOfIPs=Get-content ".\hosts.txt"
$ResultList = @()
# roll through the list, resolving as we
# go with the .Net DNS resolver
foreach ($IP in $ListOfIPs)
{
# We don't want to see any errors as we go, right?
$ErrorActionPreference = "silentlycontinue"
$Result = $null
# status to user, just so they know that something
# is still happening, then pass the current IP
# to .Net for name resolution.
write-host "resolving $IP"
$result = [System.Net.Dns]::gethostentry($IP)
# Enter into the array, with the returned results.
If ($Result)
{
$ResultList += "$IP," + [string]$Result.HostName
}
Else
$ResultList += "$IP,unresolved"
# send it out to a file, and inform the user we are done
$ResultList | Out-File .\resolved.txt
write-host "name resolution complete"
Again, pretty straightforward.
```

First, it grabs the list of IP addresses from a file called *hosts.txt*. Next, the scr resolving the IP addresses (if it can), storing the results in \$ResultList. Final another file called *resolved.txt*

I hope this is useful to someone, and drop a note in the comments if you have

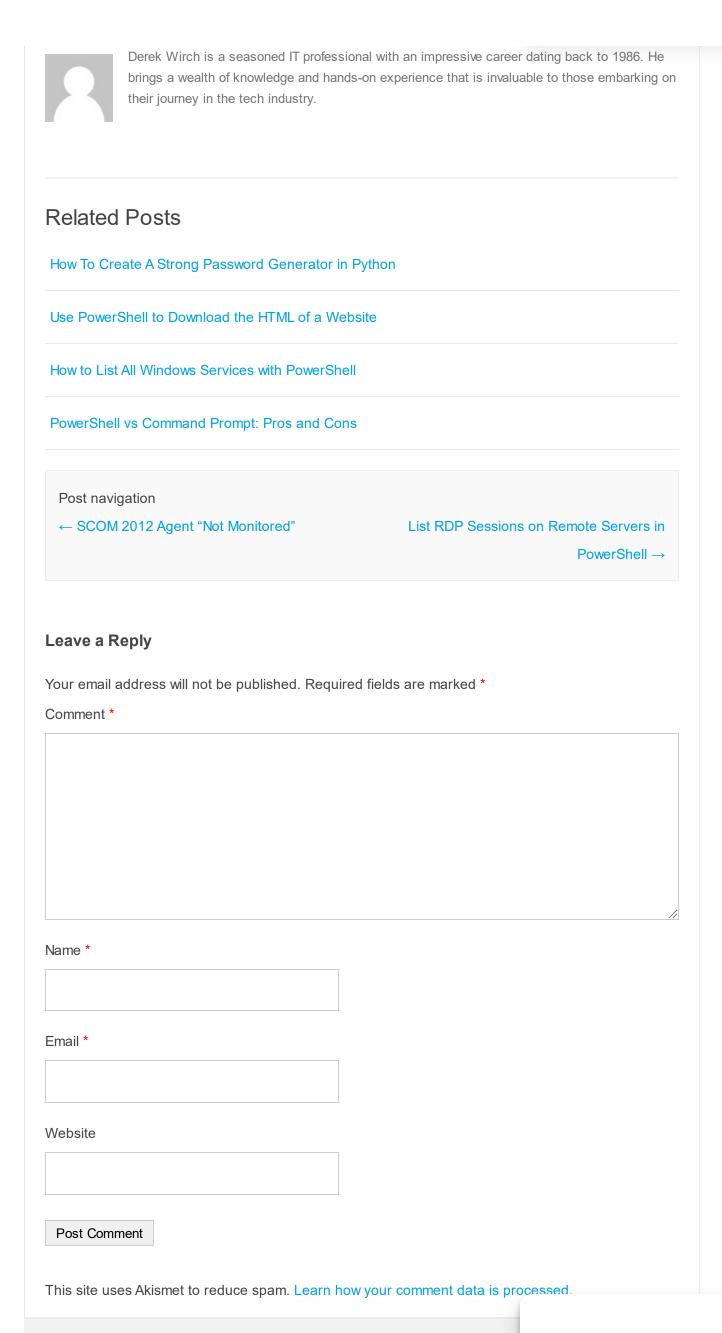
Powershell Scripting

Author: dwirch in

Manage Consent

To provide the best experiences, we use technologies like cookies to store and/or access device information. Consenting to these technologies will allow us to process data such as browsing behavior or unique IDs on this site. Not consenting or withdrawing consent, may adversely affect certain features and functions.

<u>Privacy Policy</u> <u>Privacy Policy</u> <u>Privacy Policy</u>



Manage Consent

×

© 2024

To provide the best experiences, we use technologies like cookies to store and/or access device information. Consenting to these technologies will allow us to process data such as browsing behavior or unique IDs on this site. Not consenting or withdrawing consent, may adversely affect certain features and functions.

<u>Privacy Policy</u> <u>Privacy Policy</u> <u>Privacy Policy</u>