

## DHCP Server Scope Management Tool

**This PowerShell script enables DHCPv4 scope management and reporting. It allows users to add new scopes, exclude IP ranges, set options, import configurations from CSV, and generate detailed HTML reports. The script presents a user-friendly menu interface and facilitates various DHCPv4 management tasks. A remote DHCP server can also be configured provided that it is part of the domain, enhancing efficiency and organization.**

### Usage

Inputs:

- Computer name - full computer name of the DHCPv4 server (e.g. sherwin-dc01.sherwin-domain.com)
- Scope Name - Specifies the name of the IPv4 scope that is added.
- Start Range - Specifies the starting IP address of the range in the subnet from which IP addresses should be leased or exclude by the DHCP server service.
- End Range - Specifies the ending IP address of the range in the subnet from which IP addresses should be leased or exclude by the DHCP server service.
- Scope Id - Specifies the scope ID, in IPv4 address format for which one or more option values are set.
- Option Id - Specifies the numeric identifier (ID) of the option for which one or more values are set.
- Value - Specifies one or more values to be set for the option.

Outputs:

- Operation 1: Add DHCPv4 Scope in the specified computer name
- Operation 2: Exclude IP address range from the provided Scope ID in a computer
- Operation 3: Set the option ID and value. This can be in Server Level or Scope ID level
- Operation 4: Import DHCPv4 Scope, Excluded IP addresses, and Option settings from the CSV file
- Operation 5: Generate an HTML report

## References

Most of the PowerShell cmdlet that I used in this project came from these links

### DHCP Server Module

<https://learn.microsoft.com/en-us/powershell/module/dhcpserver/?view=windowsserver2022-ps>

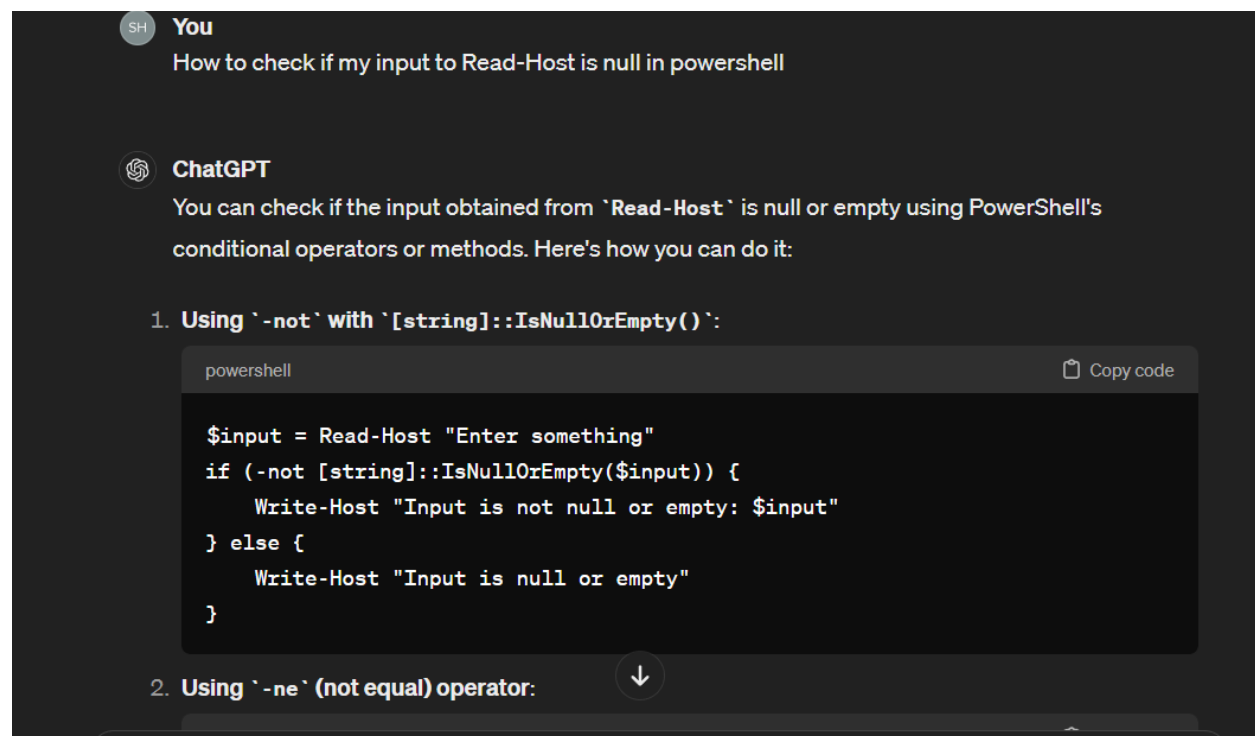
### Converting to HTML format

<https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/convertto-html?view=powershell-7.4>

Some of the advanced commands came from Chat-GPT

### Check if input is null

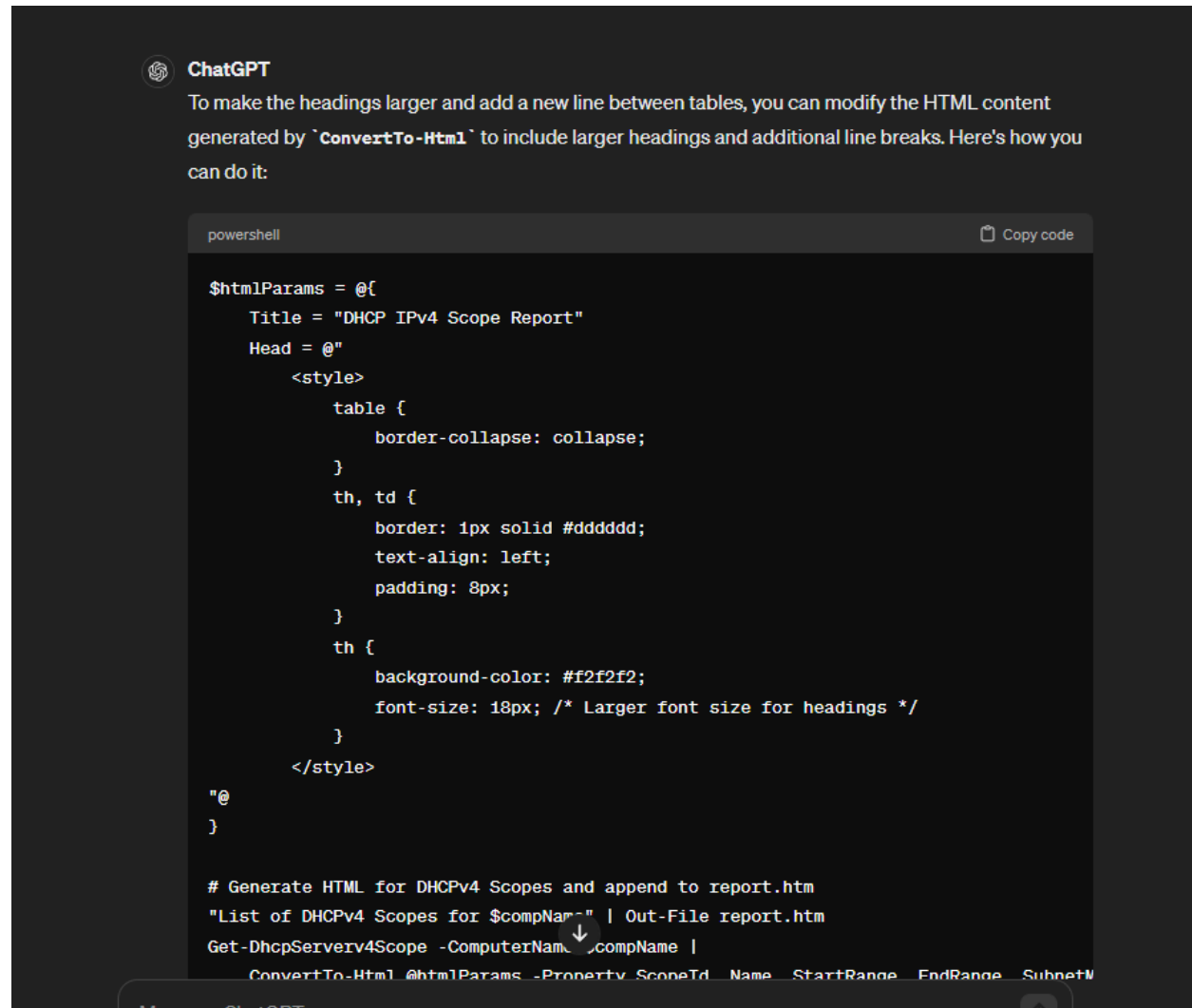
I use the AI (Chat-GPT) to help me with more advanced powershell scripts



## Generating the HTML report

This can be done thru a text file but an HTML is easier to read (at least for me)

I got the idea of embedding the styling to ConvertTo-Html cmdlet from the AI (Chat-GPT)



The screenshot shows a ChatGPT chat window with a dark theme. The ChatGPT logo is on the left. The message from ChatGPT says: "To make the headings larger and add a new line between tables, you can modify the HTML content generated by 'ConvertTo-Html' to include larger headings and additional line breaks. Here's how you can do it:". Below this is a code block with a PowerShell script. The code defines HTML parameters for a report titled "DHCP IPv4 Scope Report". It includes CSS styles for a table with a collapsed border, and for table headers (th) with a larger font size (18px) and a light blue background. The script then uses the 'ConvertTo-Html' cmdlet to generate the HTML report from DHCP scope data. A cursor is visible pointing to the 'compName' parameter in the 'Get-DhcpServerv4Scope' command. At the bottom, there is a text input field with the placeholder "Message ChatGPT" and a send button.

```
powershell Copy code

$htmlParams = @{
    Title = "DHCP IPv4 Scope Report"
    Head = @"
        <style>
            table {
                border-collapse: collapse;
            }
            th, td {
                border: 1px solid #dddddd;
                text-align: left;
                padding: 8px;
            }
            th {
                background-color: #f2f2f2;
                font-size: 18px; /* Larger font size for headings */
            }
        </style>
    @"
}

# Generate HTML for DHCPv4 Scopes and append to report.htm
"List of DHCPv4 Scopes for $compName" | Out-File report.htm
Get-DhcpServerv4Scope -ComputerName $compName |
    ConvertTo-Html @htmlParams -Property ScopeId Name StartRange EndRange SubnetV4
```

Loop thru each scope IDs.

I asked the AI on how to use arrays/list in PowerShell. This is for the function of setting the option at Scope ID level since there are multiple Scope IDs per DHCP Server.



**You**

how to save all the scope id to a list and loop thru it



**ChatGPT**

You can create an array to store all the scope IDs, and then loop through this array to perform actions for each scope ID. Here's how you can do it:

powershell

Copy code

```
# Define an empty array to store scope IDs
$scopeIDs = @()

# Retrieve all scope IDs and store them in the array
Get-DhcpServerv4Scope | ForEach-Object {
    $scopeIDs += $_.ScopeId
}

# Loop through the array of scope IDs
foreach ($scopeID in $scopeIDs) {
    # Perform actions for each scope ID
    Write-Host "Processing scope ID: $scopeID"
}
```



## Expanding the value of Value object of Get-DhcpServerv4OptionValue

I use this because I cannot parse the Value object to HTML.



**You**

how to add expand property in the Value object in this script

```
Get-DhcpServerv4OptionValue -ComputerName $compName |  
  ConvertTo-Html @htmlParams -Property ScopeId, OptionId, Name, Value
```



**ChatGPT**

To include the expanded `Value` property in the HTML output, you can use the `Select-Object` cmdlet to create a custom property where the `Value` property is expanded. Here's how you can do it:

powershell

Copy code

```
Get-DhcpServerv4OptionValue -ComputerName $compName |  
  ForEach-Object {  
    foreach ($val in $_.Value) {  
      $_ | Select-Object ScopeId, OptionId, Name, @{Name="Value"; Expression={$val}}  
    }  
  } |  
  ConvertTo-Html @htmlParams | Out-File $reportName
```