

# Count-AllDomainControllers

```
#####  
# AUTHOR   : Ryan Mutschler  
# DATE     : 8-15-2014  
# EDIT     : 8-15-2014  
# COMMENT  : Domain Controller Counter  
# This script prompts for a list of domains and counts the number of domain controllers in  
# each domain  
# Compatible with all PowerShell versions  
# Created with assistance from claude.ai  
#  
# VERSION  : 1      (Initial release)  
#####  
  
function Get-DomainControllers {  
    param (  
        [Parameter(Mandatory=$true)]  
        [string]$DomainName  
    )  
  
    try {  
        Write-Host "Searching for domain controllers in $DomainName..." -ForegroundColor Cyan  
  
        # Use .NET DirectoryServices to query the domain  
        $context = New-Object  
System.DirectoryServices.ActiveDirectory.DirectoryContext("Domain", $DomainName)  
        $domain = [System.DirectoryServices.ActiveDirectory.Domain]::GetDomain($context)  
  
        # Get all domain controllers  
        $domainControllers = $domain.DomainControllers  
  
        # Return the domain controllers
```

```

        return $domainControllers
    }
    catch [System.DirectoryServices.ActiveDirectory.ActiveDirectoryServerDownException] {
        Write-Host "Error: Cannot connect to domain $DomainName. The domain may not exist or
is not accessible." -ForegroundColor Red
        return $null
    }
    catch {
        $errorMessage = $_.Exception.Message
        Write-Host ("Error occurred while querying domain " + $DomainName + ": " +
$errorMessage) -ForegroundColor Red
        return $null
    }
}

function Main {
    Clear-Host
    Write-Host "===== " -ForegroundColor Green
    Write-Host "      Domain Controller Counter Tool      " -ForegroundColor Green
    Write-Host "===== " -ForegroundColor Green
    Write-Host ""

    # Prompt for domains
    Write-Host "Enter domain names (one per line). Press Enter on a blank line when finished:"
-ForegroundColor Yellow
    $domainList = @()

    while ($true) {
        $domain = Read-Host
        if ([string]::IsNullOrEmpty($domain)) {
            break
        }
        $domainList += $domain
    }

    # Validate that domains were entered
    if ($domainList.Count -eq 0) {
        Write-Host "No domains entered. Exiting script." -ForegroundColor Red
        return
    }
}

```

```

Write-Host ""
Write-Host "=====" -ForegroundColor Green
Write-Host "      Domain Controller Results      " -ForegroundColor Green
Write-Host "=====" -ForegroundColor Green

# Create a result table
$results = @()

# Process each domain
foreach ($domainName in $domainList) {
    $dcs = Get-DomainControllers -DomainName $domainName

    if ($null -ne $dcs) {
        $dcCount = $dcs.Count

        # Create result object
        $resultObj = New-Object PSObject -Property @{
            DomainName = $domainName
            DCCount = $dcCount
            Status = "Connected"
        }

        $results += $resultObj

        # Display domain controllers
        Write-Host ""
        Write-Host "Domain: $domainName - $dcCount Domain Controller(s) found" -
ForegroundColor Green

        if ($dcCount -gt 0) {
            foreach ($dc in $dcs) {
                Write-Host "  - $($dc.Name) (Site: $($dc.SiteName))" -ForegroundColor
White
            }
        }
    }
    else {
        # Create result object for failed connection
        $resultObj = New-Object PSObject -Property @{

```

```

        DomainName = $domainName
        DCCount = 0
        Status = "Failed to connect"
    }

    $results += $resultObj
}

# Display summary
Write-Host ""
Write-Host "=====" -ForegroundColor Green
Write-Host "          Summary          " -ForegroundColor Green
Write-Host "=====" -ForegroundColor Green

$format = "{0,-30} {1,-10} {2,-20}"
Write-Host ($format -f "Domain", "DC Count", "Status") -ForegroundColor Yellow
Write-Host ($format -f "-----", "-----", "-----") -ForegroundColor Yellow

foreach ($result in $results) {
    $statusColor = if ($result.Status -eq "Connected") { "Green" } else { "Red" }
    Write-Host ($format -f $result.DomainName, $result.DCCount, $result.Status) -
ForegroundColor $statusColor
}

Write-Host ""
Write-Host "Script completed." -ForegroundColor Cyan
}

# Run the main function
Main

```

Revision #1

Created 2025-03-14 20:01:20 UTC by Ryan

Updated 2025-03-14 20:03:26 UTC by Ryan