

Get-UserAccountControl for Specific Machine

```
<#
.SYNOPSIS
    Retrieves UserAccountControl values for machine accounts across multiple servers.

.DESCRIPTION
    This script uses Get-ADComputer to retrieve the UserAccountControl property for machine
accounts
    of servers in the provided list and outputs the results to a text file. The script is
compatible
    with all PowerShell versions.

.PARAMETER ServerList
    Path to a text file containing the list of servers, one per line.

.PARAMETER OutputFile
    Path to the output file where results will be written.

.EXAMPLE
    .\Get-MachineAccountControl.ps1 -ServerList "C:\Servers.txt" -OutputFile
"C:\UAC_Results.txt"
#>

param (
    [Parameter(Mandatory=$true)]
    [string]$ServerList,

    [Parameter(Mandatory=$true)]
    [string]$OutputFile
)

# Function to translate UserAccountControl flags into readable format
function Get-UserAccountControlFlags {
    param (
```

```

    [Parameter(Mandatory=$true)]
    [int]$UAC
)

$UACFlags = @{
    0x0001 = "SCRIPT"
    0x0002 = "ACCOUNTDISABLE"
    0x0008 = "HOMEDIR_REQUIRED"
    0x0010 = "LOCKOUT"
    0x0020 = "PASSWD_NOTREQD"
    0x0040 = "PASSWD_CANT_CHANGE"
    0x0080 = "ENCRYPTED_TEXT_PWD_ALLOWED"
    0x0100 = "TEMP_DUPLICATE_ACCOUNT"
    0x0200 = "NORMAL_ACCOUNT"
    0x0800 = "INTERDOMAIN_TRUST_ACCOUNT"
    0x1000 = "WORKSTATION_TRUST_ACCOUNT"
    0x2000 = "SERVER_TRUST_ACCOUNT"
    0x10000 = "DONT_EXPIRE_PASSWORD"
    0x20000 = "MNS_LOGON_ACCOUNT"
    0x40000 = "SMARTCARD_REQUIRED"
    0x80000 = "TRUSTED_FOR_DELEGATION"
    0x100000 = "NOT_DELEGATED"
    0x200000 = "USE_DES_KEY_ONLY"
    0x400000 = "DONT_REQ_PREAUTH"
    0x800000 = "PASSWORD_EXPIRED"
    0x1000000 = "TRUSTED_TO_AUTH_FOR_DELEGATION"
    0x04000000 = "PARTIAL_SECRETS_ACCOUNT"
}

$FlagList = @()

foreach ($Flag in $UACFlags.Keys) {
    if ($UAC -band $Flag) {
        $FlagList += $UACFlags[$Flag]
    }
}

return $FlagList -join ", "
}

```

```

# Verify the server list file exists
if (-not (Test-Path $ServerList)) {
    Write-Error "Server list file not found: $ServerList"
    exit 1
}

# Read the server list
$Servers = Get-Content $ServerList

# Initialize results array
$Results = @()

# Connect to the domain controller for querying
try {
    # Import ActiveDirectory module if needed
    if (-not (Get-Module -Name ActiveDirectory -ErrorAction SilentlyContinue)) {
        Import-Module ActiveDirectory -ErrorAction Stop
    }

    # Process each server
    foreach ($Server in $Servers) {
        $Server = $Server.Trim()
        if ([string]::IsNullOrEmpty($Server)) { continue }

        Write-Host "Processing machine account for: $Server"

        try {
            # Get the server's computer account by name
            # Note: Depending on how the server names are in your list, you might need to
adjust this
            # If your list has FQDNs, you might need to extract just the hostname part
            $ServerName = $Server
            if ($Server.Contains(".")) {
                $ServerName = $Server.Split(".")[0]
            }

            # Get the computer account
            $ComputerAccount = Get-ADComputer -Identity $ServerName -Properties
UserAccountControl, Enabled, SamAccountName, DNSHostName, OperatingSystem -ErrorAction Stop

```

```

$UACFlags = Get-UserAccountControlFlags -UAC $ComputerAccount.UserAccountControl

$Results += [PSCustomObject]@{
    Server = $Server
    SamAccountName = $ComputerAccount.SamAccountName
    DNSHostName = $ComputerAccount.DNSHostName
    OperatingSystem = $ComputerAccount.OperatingSystem
    Enabled = $ComputerAccount.Enabled
    UserAccountControl = $ComputerAccount.UserAccountControl
    UACFlags = $UACFlags
}
}
catch {
    Write-Warning "Error retrieving machine account for $Server`: $_"

    # Add error entry to results
    $Results += [PSCustomObject]@{
        Server = $Server
        SamAccountName = "ERROR"
        DNSHostName = "ERROR"
        OperatingSystem = "ERROR"
        Enabled = "ERROR"
        UserAccountControl = "ERROR"
        UACFlags = $_.Exception.Message
    }
}
}
}
catch {
    Write-Error "Error connecting to Active Directory: $_"
    exit 1
}

# Create output file with header
$Header = @"
=====
Machine Account UserAccountControl Report - Generated $(Get-Date -Format "yyyy-MM-dd
HH:mm:ss")
=====

```

```
"@

$header | Out-File -FilePath $OutputFile -Encoding utf8

# Format and append results to the output file
foreach ($Result in $Results) {
    $OutputText = @"
Server: $($Result.Server)
Computer Account: $($Result.SamAccountName)
DNS Host Name: $($Result.DNSHostName)
Operating System: $($Result.OperatingSystem)
Enabled: $($Result.Enabled)
UserAccountControl: $($Result.UserAccountControl)
Flags: $($Result.UACFlags)
-----

"@

    $OutputText | Out-File -FilePath $OutputFile -Append -Encoding utf8
}

Write-Host "Report completed. Results saved to $OutputFile"
```

Revision #2

Created 2025-02-27 16:18:40 UTC by Ryan

Updated 2025-03-13 00:56:28 UTC by Ryan