

Windows 10 - Konfigurations-Skript

WICHTIG: Das Skript wurde ausschliesslich für Windows 10 und Windows Server 2016 erstellt und getestet!

Download:

Skript Sourcecode

Filename: **win10_usability_configuration.ps1**

```
#####  
# Win10 / Server2016 Initial Setup Script  
# Version: 2.6, 2017-07-31  
# Original Source:  
https://github.com/Disassembler0/Win10-Initial-Setup-Script  
# Modded / Updated by Michael, 2017-08-07  
#####  
  
# Choose your preferences: (If you like to modify some stuff..)  
$tweaks = @(  
    ### Privacy Settings ###  
    "DisableTelemetry",           # "EnableTelemetry",  
    "DisableWiFiSense",          # "EnableWiFiSense",  
    # "DisableSmartScreen",       # "EnableSmartScreen",  
    "DisableWebSearch",          # "EnableWebSearch",  
    "DisableStartSuggestions",   # "EnableStartSuggestions",  
    "DisableLocationTracking",   # "EnableLocationTracking",  
    "DisableFeedback",           # "EnableFeedback",  
    "DisableAdvertisingID",       # "EnableAdvertisingID",  
    "DisableCortana",             # "EnableCortana",  
    # "DisableErrorReporting",    # "EnableErrorReporting",  
    "RestrictUpdateP2P",          # "UnrestrictUpdateP2P",  
    "DisableAutoLogger",         # "EnableAutoLogger",  
    "DisableDiagTrack",          # "EnableDiagTrack",  
    "DisableWAPPush",            # "EnableWAPPush",  
  
    ### Service Settings ###  
    # "LowerUAC",                 # "RaiseUAC",  
    # "EnableSharingMappedDrives", # "DisableSharingMappedDrives",  
    "DisableAdminShares",        # "EnableAdminShares",  
    "DisableSMB1",               # "EnableSMB1",
```

```
"SetCurrentNetworkPrivate",      # "SetCurrentNetworkPublic",
# "SetUnknownNetworksPrivate",   # "SetUnknownNetworksPublic",
# "DisableFirewall",            # "EnableFirewall",
# "DisableDefender",            # "EnableDefender",
# "DisableUpdateMSRT",          # "EnableUpdateMSRT",
# "DisableUpdateDriver",        # "EnableUpdateDriver",
"DisableUpdateRestart",          # "EnableUpdateRestart",
"DisableHomeGroups",             # "EnableHomeGroups",
"DisableRemoteAssistance",       # "EnableRemoteAssistance",
"EnableRemoteDesktop",           # "DisableRemoteDesktop",
"DisableAutoplay",               # "EnableAutoplay",
"DisableAutorun",                # "EnableAutorun",
# "DisableDefragmentation",      # "EnableDefragmentation",
# "DisableSuperfetch",           # "EnableSuperfetch",
# "DisableIndexing",             # "EnableIndexing",
# "SetBIOSTimeUTC",              # "SetBIOSTimeLocal",

### UI Settings ###
"DisableActionCenter",           # "EnableActionCenter",
"DisableLockScreen",             # "EnableLockScreen",
# "DisableLockScreenRS1",        # "EnableLockScreenRS1",
"DisableStickyKeys",             # "EnableStickyKeys",
"ShowTaskManagerDetails",        # "HideTaskManagerDetails",
"ShowFileOperationsDetails",     # "HideFileOperationsDetails",
"HideTaskbarSearchBox",          # "ShowTaskbarSearchBox",
"HideTaskView",                  # "ShowTaskView",
# "ShowSmallTaskbarIcons",       # "ShowLargeTaskbarIcons",
"ShowTaskbarTitles",             # "HideTaskbarTitles",
# "HideTaskbarPeopleIcon",       # "ShowTaskbarPeopleIcon",
"ShowTrayIcons",                 # "HideTrayIcons",
"ShowKnownExtensions",           # "HideKnownExtensions",
"ShowHiddenFiles",               # "HideHiddenFiles",
"HideSyncNotifications",         # "ShowSyncNotifications",
"HideRecentShortcuts",           # "ShowRecentShortcuts",
"ExplorerThisPC",                # "ExplorerQuickAccess",
"ShowThisPCOnDesktop",           # "HideThisPCFromDesktop",
"HideDesktopFromThisPC",         # "ShowDesktopInThisPC",
"HideDocumentsFromThisPC",       # "ShowDocumentsInThisPC",
"HideDownloadsFromThisPC",       # "ShowDownloadsInThisPC",
"HideMusicFromThisPC",           # "ShowMusicInThisPC",
"HidePicturesFromThisPC",        # "ShowPicturesInThisPC",
"HideVideosFromThisPC",          # "ShowVideosInThisPC",
# "AddENKeyboard",               # "RemoveENKeyboard",
# "EnableNumlock",              # "DisableNumlock",

### Application Tweaks ###
"DisableOneDrive",               # "EnableOneDrive",
"UninstallOneDrive",             # "InstallOneDrive",
"UninstallMsftBloat",           # "InstallMsftBloat",
```

```
"UninstallThirdPartyBloat",      # "InstallThirdPartyBloat",
# "UninstallWindowsStore",      # "InstallWindowsStore",
"DisableConsumerApps",          # "EnableConsumerApps",
"DisableXboxFeatures",          # "EnableXboxFeatures",
# "UninstallMediaPlayer",      # "InstallMediaPlayer",
# "UninstallWorkFolders",      # "InstallWorkFolders",
# "InstallLinuxSubsystem",      # "UninstallLinuxSubsystem",
# "InstallHyperV",             # "UninstallHyperV",
"SetPhotoViewerAssociation",    # "UnsetPhotoViewerAssociation",
"AddPhotoViewerOpenWith",      # "RemovePhotoViewerOpenWith",
"DisableSearchAppInStore",      # "EnableSearchAppInStore",
"DisableNewAppPrompt",          # "EnableNewAppPrompt",
# "EnableF8BootMenu",          # "DisableF8BootMenu",
# "SetDEPOptOut",              # "SetDEPOptIn",

### Server Specific Settings ###
# "HideServerManagerOnLogin",    # "ShowServerManagerOnLogin",
# "DisableShutdownTracker",      # "EnableShutdownTracker",
# "DisablePasswordPolicy",        # "EnablePasswordPolicy",
# "DisableCtrlAltDelLogin",      # "EnableCtrlAltDelLogin",
# "DisableIEEnhancedSecurity",    # "EnableIEEnhancedSecurity",

### Auxiliary Functions ###
"WaitForKey",
"Restart"
)

#####
#####
#
#           TO GUARANTEE THE FUNCTION, DON'T CHANGE ANYTHING
BELOW THIS LINE:
#####
#####

#####
# Privacy Settings
#####

# Disable Telemetry
Function DisableTelemetry {
    Write-Host "Disabling Telemetry..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection" -
Name "AllowTelemetry" -Type DWord -Value 0
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\DataCollection" -Name
"AllowTelemetry" -Type DWord -Value 0
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Policies\DataCo
llection" -Name "AllowTelemetry" -Type DWord -Value 0
}
```

```
# Enable Telemetry
Function EnableTelemetry {
    Write-Host "Enabling Telemetry..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\DataCollection" -
Name "AllowTelemetry" -Type DWord -Value 3
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\DataCollection" -Name
"AllowTelemetry" -Type DWord -Value 3
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Policies\DataCo
llection" -Name "AllowTelemetry" -Type DWord -Value 3
}

# Disable Wi-Fi Sense
Function DisableWiFiSense {
    Write-Host "Disabling Wi-Fi Sense..."
    If (!(Test-Path
"HKLM:\SOFTWARE\Microsoft\PolicyManager\default\WiFi\AllowWiFiHotSpotReporti
ng")) {
        New-Item -Path
"HKLM:\SOFTWARE\Microsoft\PolicyManager\default\WiFi\AllowWiFiHotSpotReporti
ng" -Force | Out-Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\PolicyManager\default\WiFi\AllowWiFiHotSpotReporti
ng" -Name "Value" -Type DWord -Value 0
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\PolicyManager\default\WiFi\AllowAutoConnectToWiFiS
enseHotspots" -Name "Value" -Type DWord -Value 0
}

# Enable Wi-Fi Sense
Function EnableWiFiSense {
    Write-Host "Enabling Wi-Fi Sense..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\PolicyManager\default\WiFi\AllowWiFiHotSpotReporti
ng" -Name "Value" -Type DWord -Value 1
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\PolicyManager\default\WiFi\AllowAutoConnectToWiFiS
enseHotspots" -Name "Value" -Type DWord -Value 1
}

# Disable SmartScreen Filter
Function DisableSmartScreen {
    Write-Host "Disabling SmartScreen Filter..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" -Name
"SmartScreenEnabled" -Type String -Value "Off"
```

```
Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\AppHost" -Name
"EnableWebContentEvaluation" -Type DWord -Value 0
$edge = (Get-AppxPackage -AllUsers
"Microsoft.MicrosoftEdge").PackageFamilyName
If (!(Test-Path "HKCU:\SOFTWARE\Classes\Local
Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\$edge\MicrosoftEdge\PhishingFilter")) {
    New-Item -Path "HKCU:\SOFTWARE\Classes\Local
Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\$edge\MicrosoftEdge\PhishingFilter" -Force | Out-Null
}
Set-ItemProperty -Path "HKCU:\SOFTWARE\Classes\Local
Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\$edge\MicrosoftEdge\PhishingFilter" -Name "EnabledV9" -Type DWord -Value 0
Set-ItemProperty -Path "HKCU:\SOFTWARE\Classes\Local
Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\$edge\MicrosoftEdge\PhishingFilter" -Name "PreventOverride" -Type DWord -Value 0
}

# Enable SmartScreen Filter
Function EnableSmartScreen {
    Write-Host "Enabling SmartScreen Filter..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" -Name
"SmartScreenEnabled" -Type String -Value "RequireAdmin"
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\AppHost" -Name
"EnableWebContentEvaluation" -ErrorAction SilentlyContinue
    $edge = (Get-AppxPackage -AllUsers
"Microsoft.MicrosoftEdge").PackageFamilyName
    Remove-ItemProperty -Path "HKCU:\SOFTWARE\Classes\Local
Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\$edge\MicrosoftEdge\PhishingFilter" -Name "EnabledV9" -ErrorAction
SilentlyContinue
    Remove-ItemProperty -Path "HKCU:\SOFTWARE\Classes\Local
Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\$edge\MicrosoftEdge\PhishingFilter" -Name "PreventOverride" -ErrorAction
SilentlyContinue
}

# Disable Web Search in Start Menu
Function DisableWebSearch {
    Write-Host "Disabling Bing Search in Start Menu..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Search" -Name
"BingSearchEnabled" -Type DWord -Value 0
    If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\Windows
Search")) {
        New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\Windows
Search" -Force | Out-Null
    }
```

```
}
Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Windows Search" -Name
"DisableWebSearch" -Type DWord -Value 1
}

# Enable Web Search in Start Menu
Function EnableWebSearch {
    Write-Host "Enabling Bing Search in Start Menu..."
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Search" -Name
"BingSearchEnabled" -ErrorAction SilentlyContinue
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Windows Search" -Name
"DisableWebSearch" -ErrorAction SilentlyContinue
}

# Disable Start Menu suggestions
Function DisableStartSuggestions {
    Write-Host "Disabling Start Menu suggestions..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -
Name "SystemPaneSuggestionsEnabled" -Type DWord -Value 0
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -
Name "SilentInstalledAppsEnabled" -Type DWord -Value 0
}

# Enable Start Menu suggestions
Function EnableStartSuggestions {
    Write-Host "Enabling Start Menu suggestions..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -
Name "SystemPaneSuggestionsEnabled" -Type DWord -Value 1
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager" -
Name "SilentInstalledAppsEnabled" -Type DWord -Value 1
}

# Disable Location Tracking
Function DisableLocationTracking {
    Write-Host "Disabling Location Tracking..."
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Sensor\Overrides\{BFA794E4-F964-4FDB-90F6-51056BFE4B44}" -
Name "SensorPermissionState" -Type DWord -Value 0
    Set-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\lfsvc\Service\Configuration" -Name
"Status" -Type DWord -Value 0
}
```

```
# Enable Location Tracking
Function EnableLocationTracking {
    Write-Host "Enabling Location Tracking..."
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Sensor\Overrides\{BFA794E4-F964-4FDB-90F6-51056BFE4B44}" -
Name "SensorPermissionState" -Type DWord -Value 1
    Set-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\lfsvc\Service\Configuration" -Name
"Status" -Type DWord -Value 1
}

# Disable Feedback
Function DisableFeedback {
    Write-Host "Disabling Feedback..."
    If (!(Test-Path "HKCU:\SOFTWARE\Microsoft\Siuf\Rules")) {
        New-Item -Path "HKCU:\SOFTWARE\Microsoft\Siuf\Rules" -Force | Out-
Null
    }
    Set-ItemProperty -Path "HKCU:\SOFTWARE\Microsoft\Siuf\Rules" -Name
"NumberOfSIUFInPeriod" -Type DWord -Value 0
}

# Enable Feedback
Function EnableFeedback {
    Write-Host "Enabling Feedback..."
    Remove-ItemProperty -Path "HKCU:\SOFTWARE\Microsoft\Siuf\Rules" -Name
"NumberOfSIUFInPeriod" -ErrorAction SilentlyContinue
}

# Disable Advertising ID
Function DisableAdvertisingID {
    Write-Host "Disabling Advertising ID..."
    If (!(Test-Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\AdvertisingInfo")) {
        New-Item -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\AdvertisingInfo" | Out-Null
    }
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\AdvertisingInfo" -Name
"Enabled" -Type DWord -Value 0
}

# Enable Advertising ID
Function EnableAdvertisingID {
    Write-Host "Enabling Advertising ID..."
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\AdvertisingInfo" -Name
"Enabled" -ErrorAction SilentlyContinue
}

# Disable Cortana
```

```
Function DisableCortana {
    Write-Host "Disabling Cortana..."
    If (!(Test-Path "HKCU:\SOFTWARE\Microsoft\Personalization\Settings")) {
        New-Item -Path "HKCU:\SOFTWARE\Microsoft\Personalization\Settings" -
Force | Out-Null
    }
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Personalization\Settings" -Name
"AcceptedPrivacyPolicy" -Type DWord -Value 0
    If (!(Test-Path "HKCU:\SOFTWARE\Microsoft\InputPersonalization")) {
        New-Item -Path "HKCU:\SOFTWARE\Microsoft\InputPersonalization" -
Force | Out-Null
    }
    Set-ItemProperty -Path "HKCU:\SOFTWARE\Microsoft\InputPersonalization" -
Name "RestrictImplicitTextCollection" -Type DWord -Value 1
    Set-ItemProperty -Path "HKCU:\SOFTWARE\Microsoft\InputPersonalization" -
Name "RestrictImplicitInkCollection" -Type DWord -Value 1
    If (!(Test-Path
"HKCU:\SOFTWARE\Microsoft\InputPersonalization\TrainedDataStore")) {
        New-Item -Path
"HKCU:\SOFTWARE\Microsoft\InputPersonalization\TrainedDataStore" -Force |
Out-Null
    }
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\InputPersonalization\TrainedDataStore" -Name
"HarvestContacts" -Type DWord -Value 0
    If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\Windows
Search")) {
        New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\Windows
Search" -Force | Out-Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Windows Search" -Name
"AllowCortana" -Type DWord -Value 0
}

# Enable Cortana
Function EnableCortana {
    Write-Host "Enabling Cortana..."
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Personalization\Settings" -Name
"AcceptedPrivacyPolicy" -ErrorAction SilentlyContinue
    Set-ItemProperty -Path "HKCU:\SOFTWARE\Microsoft\InputPersonalization" -
Name "RestrictImplicitTextCollection" -Type DWord -Value 0
    Set-ItemProperty -Path "HKCU:\SOFTWARE\Microsoft\InputPersonalization" -
Name "RestrictImplicitInkCollection" -Type DWord -Value 0
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\InputPersonalization\TrainedDataStore" -Name
"HarvestContacts" -ErrorAction SilentlyContinue
```



```
Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Windows Search" -Name
"AllowCortana" -ErrorAction SilentlyContinue
}

# Disable Error reporting
Function DisableErrorReporting {
    Write-Host "Disabling Error reporting..."
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\Windows Error
Reporting" -Name "Disabled" -Type DWord -Value 1
}

# Enable Error reporting
Function EnableErrorReporting {
    Write-Host "Enabling Error reporting..."
    Remove-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\Windows
Error Reporting" -Name "Disabled" -ErrorAction SilentlyContinue
}

# Restrict Windows Update P2P only to local network
Function RestrictUpdateP2P {
    Write-Host "Restricting Windows Update P2P only to local network..."
    If (!(Test-Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\DeliveryOptimization\Config
")) {
        New-Item -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\DeliveryOptimization\Config
" | Out-Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\DeliveryOptimization\Config
" -Name "DODownloadMode" -Type DWord -Value 1
    If (!(Test-Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\DeliveryOptimization")) {
        New-Item -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\DeliveryOptimization" |
Out-Null
    }
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\DeliveryOptimization" -Name
"SystemSettingsDownloadMode" -Type DWord -Value 3
}

# Unrestrict Windows Update P2P
Function UnrestrictUpdateP2P {
    Write-Host "Unrestricting Windows Update P2P to internet..."
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\DeliveryOptimization\Config
" -Name "DODownloadMode" -ErrorAction SilentlyContinue
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\DeliveryOptimization" -Name
```

```
"SystemSettingsDownloadMode" -ErrorAction SilentlyContinue
}

# Remove AutoLogger file and restrict directory
Function DisableAutoLogger {
    Write-Host "Removing AutoLogger file and restricting directory..."
    $autoLoggerDir =
"$env:PROGRAMDATA\Microsoft\Diagnosis\ETLLogs\AutoLogger"
    If (Test-Path "$autoLoggerDir\AutoLogger-Diagtrack-Listener.etl") {
        Remove-Item "$autoLoggerDir\AutoLogger-Diagtrack-Listener.etl"
    }
    icacls $autoLoggerDir /deny SYSTEM:`(OI)`(CI)`F | Out-Null
}

# Unrestrict AutoLogger directory
Function EnableAutoLogger {
    Write-Host "Unrestricting AutoLogger directory..."
    $autoLoggerDir =
"$env:PROGRAMDATA\Microsoft\Diagnosis\ETLLogs\AutoLogger"
    icacls $autoLoggerDir /grant:r SYSTEM:`(OI)`(CI)`F | Out-Null
}

# Stop and disable Diagnostics Tracking Service
Function DisableDiagTrack {
    Write-Host "Stopping and disabling Diagnostics Tracking Service..."
    Stop-Service "DiagTrack" -WarningAction SilentlyContinue
    Set-Service "DiagTrack" -StartupType Disabled
}

# Enable and start Diagnostics Tracking Service
Function EnableDiagTrack {
    Write-Host "Enabling and starting Diagnostics Tracking Service..."
    Set-Service "DiagTrack" -StartupType Automatic
    Start-Service "DiagTrack" -WarningAction SilentlyContinue
}

# Stop and disable WAP Push Service
Function DisableWAPPush {
    Write-Host "Stopping and disabling WAP Push Service..."
    Stop-Service "dmwappushservice" -WarningAction SilentlyContinue
    Set-Service "dmwappushservice" -StartupType Disabled
}

# Enable and start WAP Push Service
Function EnableWAPPush {
    Write-Host "Enabling and starting WAP Push Service..."
    Set-Service "dmwappushservice" -StartupType Automatic
    Start-Service "dmwappushservice" -WarningAction SilentlyContinue
    Set-ItemProperty -Path
```

```
"HKLM:\SYSTEM\CurrentControlSet\Services\dmwappushservice" -Name
"DelayedAutoStart" -Type DWord -Value 1
}

#####
# Service Tweaks
#####

# Lower UAC level
Function LowerUAC {
    Write-Host "Lowering UAC level..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" -Name
"ConsentPromptBehaviorAdmin" -Type DWord -Value 0
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" -Name
"PromptOnSecureDesktop" -Type DWord -Value 0
}

# Raise UAC level
Function RaiseUAC {
    Write-Host "Raising UAC level..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" -Name
"ConsentPromptBehaviorAdmin" -Type DWord -Value 5
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" -Name
"PromptOnSecureDesktop" -Type DWord -Value 1
}

# Enable sharing mapped drives between users
Function EnableSharingMappedDrives {
    Write-Host "Enabling sharing mapped drives between users..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" -Name
"EnableLinkedConnections" -Type DWord -Value 1
}

# Disable sharing mapped drives between users
Function DisableSharingMappedDrives {
    Write-Host "Disabling sharing mapped drives between users..."
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" -Name
"EnableLinkedConnections" -ErrorAction SilentlyContinue
}

# Disable implicit administrative shares
Function DisableAdminShares {
    Write-Host "Disabling implicit administrative shares..."
```

```
Set-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" -Name
"AutoShareWks" -Type DWord -Value 0
}

# Enable implicit administrative shares
Function EnableAdminShares {
    Write-Host "Enabling implicit administrative shares..."
    Remove-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" -Name
"AutoShareWks" -ErrorAction SilentlyContinue
}

# Disable obsolete SMB 1.0 protocol
Function DisableSMB1 {
    Write-Host "Disabling SMB 1.0 protocol..."
    Set-SmbServerConfiguration -EnableSMB1Protocol $false -Force
}

# Enable obsolete SMB 1.0 protocol
Function EnableSMB1 {
    Write-Host "Enabling SMB 1.0 protocol..."
    Set-SmbServerConfiguration -EnableSMB1Protocol $true -Force
}

# Set current network profile to private (allow file sharing, device
discovery, etc.)
Function SetCurrentNetworkPrivate {
    Write-Host "Setting current network profile to private..."
    Set-NetConnectionProfile -NetworkCategory Private
}

# Set current network profile to public (deny file sharing, device
discovery, etc.)
Function SetCurrentNetworkPublic {
    Write-Host "Setting current network profile to public..."
    Set-NetConnectionProfile -NetworkCategory Public
}

# Set unknown networks profile to private (allow file sharing, device
discovery, etc.)
Function SetUnknownNetworksPrivate {
    Write-Host "Setting unknown networks profile to private..."
    If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows
NT\CurrentVersion\NetworkList\Signatures\010103000F0000F0010000000F0000F0C96
7A3643C3AD745950DA7859209176EF5B87C875FA20DF21951640E807D7C24")) {
        New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows
NT\CurrentVersion\NetworkList\Signatures\010103000F0000F0010000000F0000F0C96
7A3643C3AD745950DA7859209176EF5B87C875FA20DF21951640E807D7C24" -Force | Out-
```

```
Null
}
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows
NT\CurrentVersion\NetworkList\Signatures\010103000F0000F0010000000F0000F0C96
7A3643C3AD745950DA7859209176EF5B87C875FA20DF21951640E807D7C24" -Name
"Category" -Type DWord -Value 1
}

# Set unknown networks profile to public (deny file sharing, device
discovery, etc.)
Function SetUnknownNetworksPublic {
    Write-Host "Setting unknown networks profile to public..."
    Remove-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows
NT\CurrentVersion\NetworkList\Signatures\010103000F0000F0010000000F0000F0C96
7A3643C3AD745950DA7859209176EF5B87C875FA20DF21951640E807D7C24" -Name
"Category" -ErrorAction SilentlyContinue
}

# Disable Firewall
Function DisableFirewall {
    Write-Host "Disabling Firewall..."
    If (!(Test-Path
"HKLM:\SOFTWARE\Policies\Microsoft\WindowsFirewall\StandardProfile")) {
        New-Item -Path
"HKLM:\SOFTWARE\Policies\Microsoft\WindowsFirewall\StandardProfile" -Force |
Out-Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\WindowsFirewall\StandardProfile" -Name
"EnableFirewall" -Type DWord -Value 0
}

# Enable Firewall
Function EnableFirewall {
    Write-Host "Enabling Firewall..."
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\WindowsFirewall\StandardProfile" -Name
"EnableFirewall" -ErrorAction SilentlyContinue
}

# Disable Windows Defender
Function DisableDefender {
    Write-Host "Disabling Windows Defender..."
    If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows Defender")) {
        New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows Defender"
-Force | Out-Null
    }
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows
Defender" -Name "DisableAntiSpyware" -Type DWord -Value 1
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" -Name "SecurityHealth"
```

```
-ErrorAction SilentlyContinue
}

# Enable Windows Defender
Function EnableDefender {
    Write-Host "Enabling Windows Defender..."
    Remove-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows Defender" -Name "DisableAntiSpyware" -ErrorAction SilentlyContinue
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" -Name "SecurityHealth" -Type ExpandString -Value "`"%ProgramFiles%\Windows Defender\MSASCuiL.exe`""
}

# Disable offering of Malicious Software Removal Tool through Windows Update
Function DisableUpdateMSRT {
    Write-Host "Disabling Malicious Software Removal Tool offering..."
    If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\MRT")) {
        New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\MRT" | Out-Null
    }
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\MRT" -Name "DontOfferThroughWUAU" -Type DWord -Value 1
}

# Enable offering of Malicious Software Removal Tool through Windows Update
Function EnableUpdateMSRT {
    Write-Host "Enabling Malicious Software Removal Tool offering..."
    Remove-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\MRT" -Name "DontOfferThroughWUAU" -ErrorAction SilentlyContinue
}

# Disable offering of drivers through Windows Update
Function DisableUpdateDriver {
    Write-Host "Disabling driver offering through Windows Update..."
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\DriverSearching" -Name "SearchOrderConfig" -Type DWord -Value 0
    If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate")) {
        New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate" | Out-Null
    }
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate" -Name "ExcludeWUDriversInQualityUpdate" -Type DWord -Value 1
}

# Enable offering of drivers through Windows Update
Function EnableUpdateDriver {
    Write-Host "Enabling driver offering through Windows Update..."
```

```
Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\DriverSearching" -Name
"SearchOrderConfig" -Type DWord -Value 1
Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate" -Name
"ExcludeWUDriversInQualityUpdate" -ErrorAction SilentlyContinue
}

# Disable Windows Update automatic restart
Function DisableUpdateRestart {
    Write-Host "Disabling Windows Update automatic restart..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\WindowsUpdate\UX\Settings" -Name "UxOption" -Type
DWord -Value 1
    If (!(Test-Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU")) {
        New-Item -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU" -Force | Out-
Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU" -Name
"NoAutoRebootWithLoggedOnUsers" -Type DWord -Value 1
}

# Enable Windows Update automatic restart
Function EnableUpdateRestart {
    Write-Host "Enabling Windows Update automatic restart..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\WindowsUpdate\UX\Settings" -Name "UxOption" -Type
DWord -Value 0
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU" -Name
"NoAutoRebootWithLoggedOnUsers" -ErrorAction SilentlyContinue
}

# Stop and disable Home Groups services - Not applicable to Server
Function DisableHomeGroups {
    Write-Host "Stopping and disabling Home Groups services..."
    Stop-Service "HomeGroupListener" -WarningAction SilentlyContinue
    Set-Service "HomeGroupListener" -StartupType Disabled
    Stop-Service "HomeGroupProvider" -WarningAction SilentlyContinue
    Set-Service "HomeGroupProvider" -StartupType Disabled
}

# Enable and start Home Groups services - Not applicable to Server
Function EnableHomeGroups {
    Write-Host "Starting and enabling Home Groups services..."
    Set-Service "HomeGroupListener" -StartupType Manual
    Set-Service "HomeGroupProvider" -StartupType Manual
    Start-Service "HomeGroupProvider" -WarningAction SilentlyContinue
}
```

```
}

# Disable Remote Assistance - Not applicable to Server (unless Remote
Assistance is explicitly installed)
Function DisableRemoteAssistance {
    Write-Host "Disabling Remote Assistance..."
    Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Remote
Assistance" -Name "fAllowToGetHelp" -Type DWord -Value 0
}

# Enable Remote Assistance - Not applicable to Server (unless Remote
Assistance is explicitly installed)
Function EnableRemoteAssistance {
    Write-Host "Enabling Remote Assistance..."
    Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Remote
Assistance" -Name "fAllowToGetHelp" -Type DWord -Value 1
}

# Enable Remote Desktop w/o Network Level Authentication
Function EnableRemoteDesktop {
    Write-Host "Enabling Remote Desktop w/o Network Level Authentication..."
    Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal
Server" -Name "fDenyTSConnections" -Type DWord -Value 0
    Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-Tcp" -Name "UserAuthentication" -Type DWord -Value 0
}

# Disable Remote Desktop
Function DisableRemoteDesktop {
    Write-Host "Disabling Remote Desktop..."
    Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal
Server" -Name "fDenyTSConnections" -Type DWord -Value 1
    Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-Tcp" -Name "UserAuthentication" -Type DWord -Value 1
}

# Disable Autoplay
Function DisableAutoplay {
    Write-Host "Disabling Autoplay..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\AutoplayHandlers"
-Name "DisableAutoplay" -Type DWord -Value 1
}

# Enable Autoplay
Function EnableAutoplay {
    Write-Host "Enabling Autoplay..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\AutoplayHandlers"
```



```
-Name "DisableAutoplay" -Type DWord -Value 0
}

# Disable Autorun for all drives
Function DisableAutorun {
    Write-Host "Disabling Autorun for all drives..."
    If (!(Test-Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer")) {
        New-Item -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer" | Out-
Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer" -Name
"NoDriveTypeAutoRun" -Type DWord -Value 255
}

# Enable Autorun for removable drives
Function EnableAutorun {
    Write-Host "Enabling Autorun for all drives..."
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer" -Name
"NoDriveTypeAutoRun" -ErrorAction SilentlyContinue
}

# Disable scheduled defragmentation task
Function DisableDefragmentation {
    Write-Host "Disabling scheduled defragmentation..."
    Disable-ScheduledTask -TaskName
"\Microsoft\Windows\Defrag\ScheduledDefrag" | Out-Null
}

# Enable scheduled defragmentation task
Function EnableDefragmentation {
    Write-Host "Enabling scheduled defragmentation..."
    Enable-ScheduledTask -TaskName
"\Microsoft\Windows\Defrag\ScheduledDefrag" | Out-Null
}

# Stop and disable Superfetch service - Not applicable to Server
Function DisableSuperfetch {
    Write-Host "Stopping and disabling Superfetch service..."
    Stop-Service "SysMain" -WarningAction SilentlyContinue
    Set-Service "SysMain" -StartupType Disabled
}

# Start and enable Superfetch service - Not applicable to Server
Function EnableSuperfetch {
    Write-Host "Starting and enabling Superfetch service..."
    Set-Service "SysMain" -StartupType Automatic
    Start-Service "SysMain" -WarningAction SilentlyContinue
}
```

```
}

# Stop and disable Windows Search indexing service
Function DisableIndexing {
    Write-Host "Stopping and disabling Windows Search indexing service..."
    Stop-Service "WSearch" -WarningAction SilentlyContinue
    Set-Service "WSearch" -StartupType Disabled
}

# Start and enable Windows Search indexing service
Function EnableIndexing {
    Write-Host "Starting and enabling Windows Search indexing service..."
    Set-Service "WSearch" -StartupType Automatic
    Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\WSearch"
-Name "DelayedAutoStart" -Type DWord -Value 1
    Start-Service "WSearch" -WarningAction SilentlyContinue
}

# Set BIOS time to UTC
Function SetBIOSTimeUTC {
    Write-Host "Setting BIOS time to UTC..."
    Set-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Control\TimeZoneInformation" -Name
"RealTimeIsUniversal" -Type DWord -Value 1
}

# Set BIOS time to local time
Function SetBIOSTimeLocal {
    Write-Host "Setting BIOS time to Local time..."
    Remove-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Control\TimeZoneInformation" -Name
"RealTimeIsUniversal" -ErrorAction SilentlyContinue
}

#####
# UI Tweaks
#####

# Disable Action Center
Function DisableActionCenter {
    Write-Host "Disabling Action Center..."
    If (!(Test-Path "HKCU:\SOFTWARE\Policies\Microsoft\Windows\Explorer")) {
        New-Item -Path "HKCU:\SOFTWARE\Policies\Microsoft\Windows\Explorer"
| Out-Null
    }
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Policies\Microsoft\Windows\Explorer" -Name
```

```
"DisableNotificationCenter" -Type DWord -Value 1
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\PushNotifications" -Name
"ToastEnabled" -Type DWord -Value 0
}

# Enable Action Center
Function EnableActionCenter {
    Write-Host "Enabling Action Center..."
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Policies\Microsoft\Windows\Explorer" -Name
"DisableNotificationCenter" -ErrorAction SilentlyContinue
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\PushNotifications" -Name
"ToastEnabled" -ErrorAction SilentlyContinue
}

# Disable Lock screen
Function DisableLockScreen {
    Write-Host "Disabling Lock screen..."
    If (!(Test-Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Personalization")) {
        New-Item -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Personalization" | Out-Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Personalization" -Name
"NoLockScreen" -Type DWord -Value 1
}

# Enable Lock screen
Function EnableLockScreen {
    Write-Host "Enabling Lock screen..."
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Personalization" -Name
"NoLockScreen" -ErrorAction SilentlyContinue
}

# Disable Lock screen (Anniversary Update workaround) - Applicable to RS1 or
newer
Function DisableLockScreenRS1 {
    Write-Host "Disabling Lock screen using scheduler workaround..."
    $service = New-Object -com Schedule.Service
    $service.Connect()
    $task = $service.NewTask(0)
    $task.Settings.DisallowStartIfOnBatteries = $false
    $trigger = $task.Triggers.Create(9)
    $trigger = $task.Triggers.Create(11)
    $trigger.StateChange = 8
    $action = $task.Actions.Create(0)
    $action.Path = "reg.exe"
```

```
$action.Arguments = "add
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\LogonUI\SessionData /t REG_DWORD /v AllowLockScreen /d 0 /f"
    $service.GetFolder("\").RegisterTaskDefinition("Disable LockScreen",
$task, 6, "NT AUTHORITY\SYSTEM", $null, 4) | Out-Null
}

# Enable Lock screen (Anniversary Update workaround) - Applicable to RS1 or newer
Function EnableLockScreenRS1 {
    Write-Host "Enabling Lock screen (removing scheduler workaround)..."
    Unregister-ScheduledTask -TaskName "Disable LockScreen" -Confirm:$false -ErrorAction SilentlyContinue
}

# Disable Sticky keys prompt
Function DisableStickyKeys {
    Write-Host "Disabling Sticky keys prompt..."
    Set-ItemProperty -Path "HKCU:\Control Panel\Accessibility\StickyKeys" -Name "Flags" -Type String -Value "506"
}

# Enable Sticky keys prompt
Function EnableStickyKeys {
    Write-Host "Enabling Sticky keys prompt..."
    Set-ItemProperty -Path "HKCU:\Control Panel\Accessibility\StickyKeys" -Name "Flags" -Type String -Value "510"
}

# Show Task Manager details
Function ShowTaskManagerDetails {
    Write-Host "Showing task manager details..."
    If (!(Test-Path "HKCU:\Software\Microsoft\Windows\CurrentVersion\TaskManager")) {
        New-Item -Path "HKCU:\Software\Microsoft\Windows\CurrentVersion\TaskManager" -Force | Out-Null
    }
    $preferences = Get-ItemProperty -Path "HKCU:\Software\Microsoft\Windows\CurrentVersion\TaskManager" -Name "Preferences" -ErrorAction SilentlyContinue
    If (!$preferences) {
        $taskmgr = Start-Process -WindowStyle Hidden -FilePath taskmgr.exe -PassThru
        While (!$preferences) {
            Start-Sleep -m 250
            $preferences = Get-ItemProperty -Path "HKCU:\Software\Microsoft\Windows\CurrentVersion\TaskManager" -Name "Preferences" -ErrorAction SilentlyContinue
        }
    }
}
```

```
    }
    Stop-Process $taskmgr
}
$preferences.Preferences[28] = 0
Set-ItemProperty -Path
"HKCU:\Software\Microsoft\Windows\CurrentVersion\TaskManager" -Name
"Preferences" -Type Binary -Value $preferences.Preferences
}

# Hide Task Manager details
Function HideTaskManagerDetails {
    Write-Host "Hiding task manager details..."
    $preferences = Get-ItemProperty -Path
"HKCU:\Software\Microsoft\Windows\CurrentVersion\TaskManager" -Name
"Preferences" -ErrorAction SilentlyContinue
    If ($preferences) {
        $preferences.Preferences[28] = 1
        Set-ItemProperty -Path
"HKCU:\Software\Microsoft\Windows\CurrentVersion\TaskManager" -Name
"Preferences" -Type Binary -Value $preferences.Preferences
    }
}

# Show file operations details
Function ShowFileOperationsDetails {
    Write-Host "Showing file operations details..."
    If (!(Test-Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\OperationStatusMan
ager")) {
        New-Item -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\OperationStatusMan
ager" | Out-Null
    }
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\OperationStatusMan
ager" -Name "EnthusiastMode" -Type DWord -Value 1
}

# Hide file operations details
Function HideFileOperationsDetails {
    Write-Host "Hiding file operations details..."
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\OperationStatusMan
ager" -Name "EnthusiastMode" -ErrorAction SilentlyContinue
}

# Hide Taskbar Search button / box
Function HideTaskbarSearchBox {
    Write-Host "Hiding Taskbar Search box / button..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Search" -Name
```

```
"SearchboxTaskbarMode" -Type DWord -Value 0
}

# Show Taskbar Search button / box
Function ShowTaskbarSearchBox {
    Write-Host "Showing Taskbar Search box / button..."
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Search" -Name
"SearchboxTaskbarMode" -ErrorAction SilentlyContinue
}

# Hide Task View button
Function HideTaskView {
    Write-Host "Hiding Task View button..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"ShowTaskViewButton" -Type DWord -Value 0
}

# Show Task View button
Function ShowTaskView {
    Write-Host "Showing Task View button..."
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"ShowTaskViewButton" -ErrorAction SilentlyContinue
}

# Show small icons in taskbar
Function ShowSmallTaskbarIcons {
    Write-Host "Showing small icons in taskbar..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"TaskbarSmallIcons" -Type DWord -Value 1
}

# Show large icons in taskbar
Function ShowLargeTaskbarIcons {
    Write-Host "Showing large icons in taskbar..."
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"TaskbarSmallIcons" -ErrorAction SilentlyContinue
}

# Show titles in taskbar
Function ShowTaskbarTitles {
    Write-Host "Showing titles in taskbar..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"TaskbarGlomLevel" -Type DWord -Value 1
}
```

```
}

# Hide titles in taskbar
Function HideTaskbarTitles {
    Write-Host "Hiding titles in taskbar..."
    Remove-ItemProperty -Path
    "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
    "TaskbarGlomLevel" -ErrorAction SilentlyContinue
}

# Hide Taskbar People icon
Function HideTaskbarPeopleIcon {
    Write-Host "Hiding People icon..."
    If (!(Test-Path
    "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\People"))
    {
        New-Item -Path
        "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\People" |
        Out-Null
    }
    Set-ItemProperty -Path
    "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\People" -
    Name "PeopleBand" -Type DWord -Value 0
}

# Show Taskbar People icon
Function ShowTaskbarPeopleIcon {
    Write-Host "Showing People icon..."
    Remove-ItemProperty -Path
    "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\People" -
    Name "PeopleBand" -ErrorAction SilentlyContinue
}

# Show all tray icons
Function ShowTrayIcons {
    Write-Host "Showing all tray icons..."
    Set-ItemProperty -Path
    "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" -Name
    "EnableAutoTray" -Type DWord -Value 0
}

# Hide tray icons as needed
Function HideTrayIcons {
    Write-Host "Hiding tray icons..."
    Remove-ItemProperty -Path
    "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" -Name
    "EnableAutoTray" -ErrorAction SilentlyContinue
}

# Show known file extensions
Function ShowKnownExtensions {
```

```
Write-Host "Showing known file extensions..."
Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"HideFileExt" -Type DWord -Value 0
}

# Hide known file extensions
Function HideKnownExtensions {
    Write-Host "Hiding known file extensions..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"HideFileExt" -Type DWord -Value 1
}

# Show hidden files
Function ShowHiddenFiles {
    Write-Host "Showing hidden files..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"Hidden" -Type DWord -Value 1
}

# Hide hidden files
Function HideHiddenFiles {
    Write-Host "Hiding hidden files..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"Hidden" -Type DWord -Value 2
}

# Hide sync provider notifications
Function HideSyncNotifications {
    Write-Host "Hiding sync provider notifications..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"ShowSyncProviderNotifications" -Type DWord -Value 0
}

# Show sync provider notifications
Function ShowSyncNotifications {
    Write-Host "Showing sync provider notifications..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"ShowSyncProviderNotifications" -Type DWord -Value 1
}

# Hide recently and frequently used item shortcuts in Explorer
Function HideRecentShortcuts {
    Write-Host "Hiding recent shortcuts..."
}
```



```
Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" -Name
"ShowRecent" -Type DWord -Value 0
Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" -Name
"ShowFrequent" -Type DWord -Value 0
}

# Show recently and frequently used item shortcuts in Explorer
Function ShowRecentShortcuts {
    Write-Host "Showing recent shortcuts..."
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" -Name
"ShowRecent" -ErrorAction SilentlyContinue
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" -Name
"ShowFrequent" -ErrorAction SilentlyContinue
}

# Change default Explorer view to This PC
Function ExplorerThisPC {
    Write-Host "Changing default Explorer view to This PC..."
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"LaunchTo" -Type DWord -Value 1
}

# Change default Explorer view to Quick Access
Function ExplorerQuickAccess {
    Write-Host "Changing default Explorer view to Quick Access..."
    Remove-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced" -Name
"LaunchTo" -ErrorAction SilentlyContinue
}

# Show This PC shortcut on desktop
Function ShowThisPCOnDesktop {
    Write-Host "Showing This PC shortcut on desktop..."
    If (!(Test-Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\C
lassicStartMenu")) {
        New-Item -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\C
lassicStartMenu" -Force | Out-Null
    }
    Set-ItemProperty -Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\C
lassicStartMenu" -Name "{20D04FE0-3AEA-1069-A2D8-08002B30309D}" -Type DWord
-Value 0
    If (!(Test-Path
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\N
```

```
ewStartPanel")) {  
    New-Item -Path  
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\N  
ewStartPanel" -Force | Out-Null  
}  
Set-ItemProperty -Path  
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\N  
ewStartPanel" -Name "{20D04FE0-3AEA-1069-A2D8-08002B30309D}" -Type DWord -  
Value 0  
}  
  
# Hide This PC shortcut from desktop  
Function HideThisPCFromDesktop {  
    Write-Host "Hiding This PC shortcut from desktop..."  
    Remove-ItemProperty -Path  
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\C  
lassicStartMenu" -Name "{20D04FE0-3AEA-1069-A2D8-08002B30309D}" -ErrorAction  
SilentlyContinue  
    Remove-ItemProperty -Path  
"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\N  
ewStartPanel" -Name "{20D04FE0-3AEA-1069-A2D8-08002B30309D}" -ErrorAction  
SilentlyContinue  
}  
  
# Hide Desktop icon from This PC  
Function HideDesktopFromThisPC {  
    Write-Host "Hiding Desktop icon from This PC..."  
    Set-ItemProperty -Path  
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions  
\{B4BFCC3A-DB2C-424C-B029-7FE99A87C641}\PropertyBag" -Name "ThisPCPolicy" -  
Type String -Value "Hide"  
    Set-ItemProperty -Path  
"HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Folder  
Descriptions\{B4BFCC3A-DB2C-424C-B029-7FE99A87C641}\PropertyBag" -Name  
"ThisPCPolicy" -Type String -Value "Hide"  
}  
  
# Show Desktop icon in This PC  
Function ShowDesktopInThisPC {  
    Write-Host "Showing Desktop icon in This PC..."  
    Set-ItemProperty -Path  
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions  
\{B4BFCC3A-DB2C-424C-B029-7FE99A87C641}\PropertyBag" -Name "ThisPCPolicy" -  
Type String -Value "Show"  
    Set-ItemProperty -Path  
"HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Folder  
Descriptions\{B4BFCC3A-DB2C-424C-B029-7FE99A87C641}\PropertyBag" -Name  
"ThisPCPolicy" -Type String -Value "Show"  
}
```

```
# Hide Documents icon from This PC
Function HideDocumentsFromThisPC {
    Write-Host "Hiding Documents icon from This PC..."
    Set-ItemProperty -Path
    "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\{f42ee2d3-909f-4907-8871-4c22fc0bf756}\PropertyBag" -Name "ThisPCPolicy" -
    Type String -Value "Hide"
    Set-ItemProperty -Path
    "HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Folder
    Descriptions\{f42ee2d3-909f-4907-8871-4c22fc0bf756}\PropertyBag" -Name
    "ThisPCPolicy" -Type String -Value "Hide"
}

# Show Documents icon in This PC
Function ShowDocumentsInThisPC {
    Write-Host "Showing Documents icon in This PC..."
    Set-ItemProperty -Path
    "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\{f42ee2d3-909f-4907-8871-4c22fc0bf756}\PropertyBag" -Name "ThisPCPolicy" -
    Type String -Value "Show"
    Set-ItemProperty -Path
    "HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Folder
    Descriptions\{f42ee2d3-909f-4907-8871-4c22fc0bf756}\PropertyBag" -Name
    "ThisPCPolicy" -Type String -Value "Show"
}

# Hide Downloads icon from This PC
Function HideDownloadsFromThisPC {
    Write-Host "Hiding Downloads icon from This PC..."
    Set-ItemProperty -Path
    "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\{7d83ee9b-2244-4e70-b1f5-5393042af1e4}\PropertyBag" -Name "ThisPCPolicy" -
    Type String -Value "Hide"
    Set-ItemProperty -Path
    "HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Folder
    Descriptions\{7d83ee9b-2244-4e70-b1f5-5393042af1e4}\PropertyBag" -Name
    "ThisPCPolicy" -Type String -Value "Hide"
}

# Show Downloads icon in This PC
Function ShowDownloadsInThisPC {
    Write-Host "Showing Downloads icon in This PC..."
    Set-ItemProperty -Path
    "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\{7d83ee9b-2244-4e70-b1f5-5393042af1e4}\PropertyBag" -Name "ThisPCPolicy" -
    Type String -Value "Show"
    Set-ItemProperty -Path
    "HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Folder
    Descriptions\{7d83ee9b-2244-4e70-b1f5-5393042af1e4}\PropertyBag" -Name
    "ThisPCPolicy" -Type String -Value "Show"
}
```

```
# Hide Music icon from This PC
Function HideMusicFromThisPC {
    Write-Host "Hiding Music icon from This PC..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions
\{a0c69a99-21c8-4671-8703-7934162fcf1d}\PropertyBag" -Name "ThisPCPolicy" -
Type String -Value "Hide"
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Folder
Descriptions\{a0c69a99-21c8-4671-8703-7934162fcf1d}\PropertyBag" -Name
"ThisPCPolicy" -Type String -Value "Hide"
}

# Show Music icon in This PC
Function ShowMusicInThisPC {
    Write-Host "Showing Music icon in This PC..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions
\{a0c69a99-21c8-4671-8703-7934162fcf1d}\PropertyBag" -Name "ThisPCPolicy" -
Type String -Value "Show"
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Folder
Descriptions\{a0c69a99-21c8-4671-8703-7934162fcf1d}\PropertyBag" -Name
"ThisPCPolicy" -Type String -Value "Show"
}

# Hide Pictures icon from This PC
Function HidePicturesFromThisPC {
    Write-Host "Hiding Pictures icon from This PC..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions
\{0ddd015d-b06c-45d5-8c4c-f59713854639}\PropertyBag" -Name "ThisPCPolicy" -
Type String -Value "Hide"
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Folder
Descriptions\{0ddd015d-b06c-45d5-8c4c-f59713854639}\PropertyBag" -Name
"ThisPCPolicy" -Type String -Value "Hide"
}

# Show Pictures icon in This PC
Function ShowPicturesInThisPC {
    Write-Host "Showing Pictures icon in This PC..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions
\{0ddd015d-b06c-45d5-8c4c-f59713854639}\PropertyBag" -Name "ThisPCPolicy" -
Type String -Value "Show"
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Folder
Descriptions\{0ddd015d-b06c-45d5-8c4c-f59713854639}\PropertyBag" -Name
```

```
"ThisPCPolicy" -Type String -Value "Show"
}

# Hide Videos icon from This PC
Function HideVideosFromThisPC {
    Write-Host "Hiding Videos icon from This PC..."
    Set-ItemProperty -Path
    "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\{35286a68-3c57-41a1-bbb1-0eae73d76c95}\PropertyBag" -Name "ThisPCPolicy" -
    Type String -Value "Hide"
    Set-ItemProperty -Path
    "HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Folder
    Descriptions\{35286a68-3c57-41a1-bbb1-0eae73d76c95}\PropertyBag" -Name
    "ThisPCPolicy" -Type String -Value "Hide"
}

# Show Videos icon in This PC
Function ShowVideosInThisPC {
    Write-Host "Showing Videos icon in This PC..."
    Set-ItemProperty -Path
    "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\{35286a68-3c57-41a1-bbb1-0eae73d76c95}\PropertyBag" -Name "ThisPCPolicy" -
    Type String -Value "Show"
    Set-ItemProperty -Path
    "HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Folder
    Descriptions\{35286a68-3c57-41a1-bbb1-0eae73d76c95}\PropertyBag" -Name
    "ThisPCPolicy" -Type String -Value "Show"
}

# Add secondary en-US keyboard
Function AddENKeyboard {
    Write-Host "Adding secondary en-US keyboard..."
    $langs = Get-WinUserLanguageList
    $langs.Add("en-US")
    Set-WinUserLanguageList $langs -Force
}

# Remove secondary en-US keyboard
Function RemoveENKeyboard {
    Write-Host "Removing secondary en-US keyboard..."
    $langs = Get-WinUserLanguageList
    Set-WinUserLanguageList ($langs | ? {$_.LanguageTag -ne "en-US"}) -Force
}

# Enable NumLock after startup
Function EnableNumlock {
    Write-Host "Enabling NumLock after startup..."
    If (!(Test-Path "HKU:")) {
        New-PSDrive -Name HKU -PSProvider Registry -Root HKEY_USERS | Out-
        Null
    }
}
```

```
Set-ItemProperty -Path "HKU:\.DEFAULT\Control Panel\Keyboard" -Name
"InitialKeyboardIndicators" -Type DWord -Value 2147483650
Add-Type -AssemblyName System.Windows.Forms
If (!(([System.Windows.Forms.Control]::IsKeyLocked('NumLock')))) {
    $wsh = New-Object -ComObject WScript.Shell
    $wsh.SendKeys('{NUMLOCK}')
}
}

# Disable NumLock after startup
Function DisableNumlock {
    Write-Host "Disabling NumLock after startup..."
    If (!(Test-Path "HKU:")) {
        New-PSDrive -Name HKU -PSProvider Registry -Root HKEY_USERS | Out-
Null
    }
    Set-ItemProperty -Path "HKU:\.DEFAULT\Control Panel\Keyboard" -Name
"InitialKeyboardIndicators" -Type DWord -Value 2147483648
Add-Type -AssemblyName System.Windows.Forms
If ([System.Windows.Forms.Control]::IsKeyLocked('NumLock')) {
    $wsh = New-Object -ComObject WScript.Shell
    $wsh.SendKeys('{NUMLOCK}')
}
}

#####
# Application Tweaks
#####

# Disable OneDrive
Function DisableOneDrive {
    Write-Host "Disabling OneDrive..."
    If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\OneDrive")) {
        New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\OneDrive"
| Out-Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\OneDrive" -Name
"DisableFileSyncNGSC" -Type DWord -Value 1
}

# Enable OneDrive
Function EnableOneDrive {
    Write-Host "Enabling OneDrive..."
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\OneDrive" -Name
"DisableFileSyncNGSC" -ErrorAction SilentlyContinue
}
```

```
}

# Uninstall OneDrive - Not applicable to Server
Function UninstallOneDrive {
    Write-Host "Uninstalling OneDrive..."
    Stop-Process -Name OneDrive -ErrorAction SilentlyContinue
    Start-Sleep -s 3
    $onedrive = "$env:SYSTEMROOT\SysWOW64\OneDriveSetup.exe"
    If (!(Test-Path $onedrive)) {
        $onedrive = "$env:SYSTEMROOT\System32\OneDriveSetup.exe"
    }
    Start-Process $onedrive "/uninstall" -NoNewWindow -Wait
    Start-Sleep -s 3
    Stop-Process -Name explorer -ErrorAction SilentlyContinue
    Start-Sleep -s 3
    Remove-Item "$env:USERPROFILE\OneDrive" -Force -Recurse -ErrorAction SilentlyContinue
    Remove-Item "$env:LOCALAPPDATA\Microsoft\OneDrive" -Force -Recurse -ErrorAction SilentlyContinue
    Remove-Item "$env:PROGRAMDATA\Microsoft OneDrive" -Force -Recurse -ErrorAction SilentlyContinue
    Remove-Item "$env:SYSTEMDRIVE\OneDriveTemp" -Force -Recurse -ErrorAction SilentlyContinue
    If (!(Test-Path "HKCR:")) {
        New-PSDrive -Name HKCR -PSProvider Registry -Root HKEY_CLASSES_ROOT | Out-Null
    }
    Remove-Item -Path "HKCR:\CLSID\{018D5C66-4533-4307-9B53-224DE2ED1FE6}" -Recurse -ErrorAction SilentlyContinue
    Remove-Item -Path "HKCR:\Wow6432Node\CLSID\{018D5C66-4533-4307-9B53-224DE2ED1FE6}" -Recurse -ErrorAction SilentlyContinue
}

# Install OneDrive - Not applicable to Server
Function InstallOneDrive {
    Write-Host "Installing OneDrive..."
    $onedrive = "$env:SYSTEMROOT\SysWOW64\OneDriveSetup.exe"
    If (!(Test-Path $onedrive)) {
        $onedrive = "$env:SYSTEMROOT\System32\OneDriveSetup.exe"
    }
    Start-Process $onedrive -NoNewWindow
}

# Uninstall default Microsoft applications
Function UninstallMsftBloat {
    Write-Host "Uninstalling default Microsoft applications..."
    Get-AppxPackage "Microsoft.3DBuilder" | Remove-AppxPackage
    Get-AppxPackage "Microsoft.BingFinance" | Remove-AppxPackage
    Get-AppxPackage "Microsoft.BingNews" | Remove-AppxPackage
    Get-AppxPackage "Microsoft.BingSports" | Remove-AppxPackage
}
```

```
Get-AppxPackage "Microsoft.BingWeather" | Remove-AppxPackage
Get-AppxPackage "Microsoft.Getstarted" | Remove-AppxPackage
Get-AppxPackage "Microsoft.MicrosoftOfficeHub" | Remove-AppxPackage
Get-AppxPackage "Microsoft.MicrosoftSolitaireCollection" | Remove-
AppxPackage
Get-AppxPackage "Microsoft.Office.OneNote" | Remove-AppxPackage
Get-AppxPackage "Microsoft.People" | Remove-AppxPackage
Get-AppxPackage "Microsoft.SkypeApp" | Remove-AppxPackage
Get-AppxPackage "Microsoft.Windows.Photos" | Remove-AppxPackage
Get-AppxPackage "Microsoft.WindowsAlarms" | Remove-AppxPackage
Get-AppxPackage "Microsoft.WindowsCamera" | Remove-AppxPackage
Get-AppxPackage "microsoft.windowscommunicationsapps" | Remove-
AppxPackage
Get-AppxPackage "Microsoft.WindowsMaps" | Remove-AppxPackage
Get-AppxPackage "Microsoft.WindowsPhone" | Remove-AppxPackage
Get-AppxPackage "Microsoft.WindowsSoundRecorder" | Remove-AppxPackage
Get-AppxPackage "Microsoft.ZuneMusic" | Remove-AppxPackage
Get-AppxPackage "Microsoft.ZuneVideo" | Remove-AppxPackage
Get-AppxPackage "Microsoft.AppConnector" | Remove-AppxPackage
Get-AppxPackage "Microsoft.ConnectivityStore" | Remove-AppxPackage
Get-AppxPackage "Microsoft.Office.Sway" | Remove-AppxPackage
Get-AppxPackage "Microsoft.Messaging" | Remove-AppxPackage
Get-AppxPackage "Microsoft.CommsPhone" | Remove-AppxPackage
Get-AppxPackage "Microsoft.MicrosoftStickyNotes" | Remove-AppxPackage
Get-AppxPackage "Microsoft.OneConnect" | Remove-AppxPackage
Get-AppxPackage "Microsoft.WindowsFeedbackHub" | Remove-AppxPackage
Get-AppxPackage "Microsoft.MinecraftUWP" | Remove-AppxPackage
Get-AppxPackage "Microsoft.MicrosoftPowerBIForWindows" | Remove-
AppxPackage
Get-AppxPackage "Microsoft.NetworkSpeedTest" | Remove-AppxPackage
Get-AppxPackage "Microsoft.MSPaint" | Remove-AppxPackage
Get-AppxPackage "Microsoft.Microsoft3DViewer" | Remove-AppxPackage
Get-AppxPackage "Microsoft.RemoteDesktop" | Remove-AppxPackage
}

# Install default Microsoft applications
Function InstallMsftBloat {
    Write-Host "Installing default Microsoft applications..."
    Get-AppxPackage -AllUsers "Microsoft.3DBuilder" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "Microsoft.BingFinance" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "Microsoft.BingNews" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "Microsoft.BingSports" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
```



```
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.BingWeather" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.Getstarted" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.MicrosoftOfficeHub" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.MicrosoftSolitaireCollection" |
ForEach {Add-AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.Office.OneNote" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.People" | ForEach {Add-AppxPackage
-DisableDevelopmentMode -Register "$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.SkypeApp" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.Windows.Photos" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.WindowsAlarms" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.WindowsCamera" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.windowscommunicationsapps" |
ForEach {Add-AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.WindowsMaps" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.WindowsPhone" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.WindowsSoundRecorder" | ForEach
{Add-AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.ZuneMusic" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.ZuneVideo" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
    Get-AppxPackage -AllUsers "Microsoft.AppConnector" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$(($_.InstallLocation)\AppXManifest.xml)"
```

```
Get-AppxPackage -AllUsers "Microsoft.ConnectivityStore" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "Microsoft.Office.Sway" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "Microsoft.Messaging" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "Microsoft.CommsPhone" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "Microsoft.MicrosoftStickyNotes" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "Microsoft.OneConnect" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "Microsoft.WindowsFeedbackHub" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "Microsoft.MinecraftUWP" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "Microsoft.MicrosoftPowerBIForWindows" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "Microsoft.NetworkSpeedTest" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "Microsoft.MSPaint" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "Microsoft.Microsoft3DViewer" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "Microsoft.RemoteDesktop" | ForEach {Add-AppxPackage -DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
}
# In case you have removed them for good, you can try to restore the files using installation medium as follows
# New-Item C:\Mnt -Type Directory | Out-Null
# dism /Mount-Image /ImageFile:D:\sources\install.wim /index:1 /ReadOnly /MountDir:C:\Mnt
# robocopy /S /SEC /R:0 "C:\Mnt\Program Files\WindowsApps" "C:\Program Files\WindowsApps"
# dism /Unmount-Image /Discard /MountDir:C:\Mnt
# Remove-Item -Path C:\Mnt -Recurse
```

```
# Uninstall default third party applications
function UninstallThirdPartyBloat {
    Write-Host "Uninstalling default third party applications..."
    Get-AppxPackage "9E2F88E3.Twitter" | Remove-AppxPackage
    Get-AppxPackage "king.com.CandyCrushSodaSaga" | Remove-AppxPackage
    Get-AppxPackage "4DF9E0F8.Netflix" | Remove-AppxPackage
    Get-AppxPackage "Drawboard.DrawboardPDF" | Remove-AppxPackage
    Get-AppxPackage "D52A8D61.FarmVille2CountryEscape" | Remove-AppxPackage
    Get-AppxPackage "GAMELOFTSA.Asphalt8Airborne" | Remove-AppxPackage
    Get-AppxPackage "flaregamesGmbH.RoyalRevolt2" | Remove-AppxPackage
    Get-AppxPackage "AdobeSystemsIncorporated.AdobePhotoshopExpress" |
Remove-AppxPackage
    Get-AppxPackage "ActiproSoftwareLLC.562882FEEB491" | Remove-AppxPackage
    Get-AppxPackage "D5EA27B7.Duolingo-LearnLanguagesforFree" | Remove-
AppxPackage
    Get-AppxPackage "Facebook.Facebook" | Remove-AppxPackage
    Get-AppxPackage "46928bounde.EclipseManager" | Remove-AppxPackage
    Get-AppxPackage "A278AB0D.MarchofEmpires" | Remove-AppxPackage
}

# Install default third party applications
Function InstallThirdPartyBloat {
    Write-Host "Installing default third party applications..."
    Get-AppxPackage -AllUsers "9E2F88E3.Twitter" | ForEach {Add-AppxPackage
-DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "king.com.CandyCrushSodaSaga" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "4DF9E0F8.Netflix" | ForEach {Add-AppxPackage
-DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "Drawboard.DrawboardPDF" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "D52A8D61.FarmVille2CountryEscape" | ForEach
{Add-AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "GAMELOFTSA.Asphalt8Airborne" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "flaregamesGmbH.RoyalRevolt2" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers
"AdobeSystemsIncorporated.AdobePhotoshopExpress" | ForEach {Add-AppxPackage
-DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "ActiproSoftwareLLC.562882FEEB491" | ForEach
{Add-AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "D5EA27B7.Duolingo-LearnLanguagesforFree" |
ForEach {Add-AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
```

```
Get-AppxPackage -AllUsers "Facebook.Facebook" | ForEach {Add-AppxPackage
-DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "46928bunde.EclipseManager" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
Get-AppxPackage -AllUsers "A278AB0D.MarchofEmpires" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
}

# Uninstall Windows Store
Function UninstallWindowsStore {
    Write-Host "Uninstalling Windows Store..."
    Get-AppxPackage "Microsoft.DesktopAppInstaller" | Remove-AppxPackage
    Get-AppxPackage "Microsoft.WindowsStore" | Remove-AppxPackage
}

# Install Windows Store
Function InstallWindowsStore {
    Write-Host "Installing Windows Store..."
    Get-AppxPackage -AllUsers "Microsoft.DesktopAppInstaller" | ForEach
{Add-AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "Microsoft.WindowsStore" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
}

# Disable installation of consumer experience applications
Function DisableConsumerApps {
    Write-Host "Disabling installation of consumer experience
applications..."
    If (!(Test-Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\CloudContent")) {
        New-Item -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\CloudContent" -Force | Out-Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\CloudContent" -Name
"DisableWindowsConsumerFeatures" -Type DWord -Value 1
}

# Enable installation of consumer experience applications
Function EnableConsumerApps {
    Write-Host "Enabling installation of consumer experience
applications..."
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\CloudContent" -Name
"DisableWindowsConsumerFeatures" -ErrorAction SilentlyContinue
}
```

```
}

# Disable Xbox features
Function DisableXboxFeatures {
    Write-Host "Disabling Xbox features..."
    Get-AppxPackage "Microsoft.XboxApp" | Remove-AppxPackage
    Get-AppxPackage "Microsoft.XboxIdentityProvider" | Remove-AppxPackage
    Get-AppxPackage "Microsoft.XboxSpeechToTextOverlay" | Remove-AppxPackage
    Get-AppxPackage "Microsoft.XboxGameOverlay" | Remove-AppxPackage
    Set-ItemProperty -Path "HKCU:\System\GameConfigStore" -Name
"GameDVR_Enabled" -Type DWord -Value 0
    If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\GameDVR")) {
        New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\GameDVR" |
Out-Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\GameDVR" -Name "AllowGameDVR" -
Type DWord -Value 0
}

# Enable Xbox features
Function EnableXboxFeatures {
    Write-Host "Enabling Xbox features..."
    Get-AppxPackage -AllUsers "Microsoft.XboxApp" | ForEach {Add-AppxPackage
-DisableDevelopmentMode -Register "$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "Microsoft.XboxIdentityProvider" | ForEach
{Add-AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "Microsoft.XboxSpeechToTextOverlay" | ForEach
{Add-AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Get-AppxPackage -AllUsers "Microsoft.XboxGameOverlay" | ForEach {Add-
AppxPackage -DisableDevelopmentMode -Register
"$($_.InstallLocation)\AppXManifest.xml"}
    Set-ItemProperty -Path "HKCU:\System\GameConfigStore" -Name
"GameDVR_Enabled" -Type DWord -Value 1
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\GameDVR" -Name "AllowGameDVR" -
ErrorAction SilentlyContinue
}

# Uninstall Windows Media Player
Function UninstallMediaPlayer {
    Write-Host "Uninstalling Windows Media Player..."
    Disable-WindowsOptionalFeature -Online -FeatureName "WindowsMediaPlayer"
-NoRestart -WarningAction SilentlyContinue | Out-Null
}

# Install Windows Media Player
Function InstallMediaPlayer {
    Write-Host "Installing Windows Media Player..."
```

```
Enable-WindowsOptionalFeature -Online -FeatureName "WindowsMediaPlayer"
-NoRestart -WarningAction SilentlyContinue | Out-Null
}

# Uninstall Work Folders Client - Not applicable to Server
Function UninstallWorkFolders {
    Write-Host "Uninstalling Work Folders Client..."
    Disable-WindowsOptionalFeature -Online -FeatureName "WorkFolders-Client"
-NoRestart -WarningAction SilentlyContinue | Out-Null
}

# Install Work Folders Client - Not applicable to Server
Function InstallWorkFolders {
    Write-Host "Installing Work Folders Client..."
    Enable-WindowsOptionalFeature -Online -FeatureName "WorkFolders-Client"
-NoRestart -WarningAction SilentlyContinue | Out-Null
}

# Install Linux Subsystem - Applicable to RS1 or newer, not applicable to
Server yet
Function InstallLinuxSubsystem {
    Write-Host "Installing Linux Subsystem..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\AppModelUnlock" -Name
"AllowDevelopmentWithoutDevLicense" -Type DWord -Value 1
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\AppModelUnlock" -Name
"AllowAllTrustedApps" -Type DWord -Value 1
    Enable-WindowsOptionalFeature -Online -FeatureName "Microsoft-Windows-
Subsystem-Linux" -NoRestart -WarningAction SilentlyContinue | Out-Null
}

# Uninstall Linux Subsystem - Applicable to RS1 or newer, not applicable to
Server yet
Function UninstallLinuxSubsystem {
    Write-Host "Uninstalling Linux Subsystem..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\AppModelUnlock" -Name
"AllowDevelopmentWithoutDevLicense" -Type DWord -Value 0
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\AppModelUnlock" -Name
"AllowAllTrustedApps" -Type DWord -Value 0
    Disable-WindowsOptionalFeature -Online -FeatureName "Microsoft-Windows-
Subsystem-Linux" -NoRestart -WarningAction SilentlyContinue | Out-Null
}

# Install Hyper-V - Not applicable to Home
Function InstallHyperV {
    Write-Host "Installing Hyper-V..."
```

```
If ((Get-WmiObject -Class "Win32_OperatingSystem").Caption -like
"*Server*") {
    Install-WindowsFeature -Name "Hyper-V" -IncludeManagementTools -
WarningAction SilentlyContinue | Out-Null
} Else {
    Enable-WindowsOptionalFeature -Online -FeatureName "Microsoft-Hyper-
V-All" -NoRestart -WarningAction SilentlyContinue | Out-Null
}
}

# Uninstall Hyper-V - Not applicable to Home
Function UninstallHyperV {
    Write-Host "Uninstalling Hyper-V..."
    If ((Get-WmiObject -Class "Win32_OperatingSystem").Caption -like
"*Server*") {
        Uninstall-WindowsFeature -Name "Hyper-V" -IncludeManagementTools -
WarningAction SilentlyContinue | Out-Null
    } Else {
        Disable-WindowsOptionalFeature -Online -FeatureName "Microsoft-
Hyper-V-All" -NoRestart -WarningAction SilentlyContinue | Out-Null
    }
}

# Set Photo Viewer association for bmp, gif, jpg, png and tif
Function SetPhotoViewerAssociation {
    Write-Host "Setting Photo Viewer association for bmp, gif, jpg, png and
tif..."
    If (!(Test-Path "HKCR:")) {
        New-PSDrive -Name HKCR -PSProvider Registry -Root HKEY_CLASSES_ROOT
| Out-Null
    }
    ForEach ($type in @("Paint.Picture", "giffile", "jpegfile", "pngfile"))
    {
        New-Item -Path $("HKCR:\$type\shell\open") -Force | Out-Null
        New-Item -Path $("HKCR:\$type\shell\open\command") | Out-Null
        Set-ItemProperty -Path $("HKCR:\$type\shell\open") -Name "MuiVerb" -
Type ExpandString -Value "@%ProgramFiles%\Windows Photo
Viewer\photoviewer.dll,-3043"
        Set-ItemProperty -Path $("HKCR:\$type\shell\open\command") -Name
"(Default)" -Type ExpandString -Value "%SystemRoot%\System32\rundll32.exe
`"%ProgramFiles%\Windows Photo Viewer\PhotoViewer.dll`,
ImageView_Fullscreen %1"
    }
}

# Unset Photo Viewer association for bmp, gif, jpg, png and tif
Function UnsetPhotoViewerAssociation {
    Write-Host "Unsetting Photo Viewer association for bmp, gif, jpg, png
and tif..."
    If (!(Test-Path "HKCR:")) {
        New-PSDrive -Name HKCR -PSProvider Registry -Root HKEY_CLASSES_ROOT
```

```
| Out-Null
}
Remove-Item -Path "HKCR:\Paint Picture\shell\open" -Recurse -ErrorAction SilentlyContinue
Remove-ItemProperty -Path "HKCR:\giffile\shell\open" -Name "MuiVerb" -ErrorAction SilentlyContinue
Set-ItemProperty -Path "HKCR:\giffile\shell\open" -Name "CommandId" -Type String -Value "IE.File"
Set-ItemProperty -Path "HKCR:\giffile\shell\open\command" -Name "(Default)" -Type String -Value "`"$env:SystemDrive\Program Files\Internet Explorer\iexplore.exe`" %1"
Set-ItemProperty -Path "HKCR:\giffile\shell\open\command" -Name "DelegateExecute" -Type String -Value "{17FE9752-0B5A-4665-84CD-569794602F5C}"
Remove-Item -Path "HKCR:\jpegfile\shell\open" -Recurse -ErrorAction SilentlyContinue
Remove-Item -Path "HKCR:\pngfile\shell\open" -Recurse -ErrorAction SilentlyContinue
}

# Add Photo Viewer to "Open with..."
Function AddPhotoViewerOpenWith {
    Write-Host "Adding Photo Viewer to `"$Open with...`""
    If (!(Test-Path "HKCR:")) {
        New-PSDrive -Name HKCR -PSProvider Registry -Root HKEY_CLASSES_ROOT
    }
    New-Item -Path "HKCR:\Applications\photoviewer.dll\shell\open\command" -Force | Out-Null
    New-Item -Path "HKCR:\Applications\photoviewer.dll\shell\open\DropTarget" -Force | Out-Null
    Set-ItemProperty -Path "HKCR:\Applications\photoviewer.dll\shell\open" -Name "MuiVerb" -Type String -Value "@photoviewer.dll,-3043"
    Set-ItemProperty -Path "HKCR:\Applications\photoviewer.dll\shell\open\command" -Name "(Default)" -Type ExpandString -Value "%SystemRoot%\System32\rundll32.exe \"%ProgramFiles%\Windows Photo Viewer\PhotoViewer.dll`, ImageView_Fullscreen %1"
    Set-ItemProperty -Path "HKCR:\Applications\photoviewer.dll\shell\open\DropTarget" -Name "Clsid" -Type String -Value "{FFE2A43C-56B9-4bf5-9A79-CC6D4285608A}"
}

# Remove Photo Viewer from "Open with..."
Function RemovePhotoViewerOpenWith {
    Write-Host "Removing Photo Viewer from `"$Open with...`""
    If (!(Test-Path "HKCR:")) {
        New-PSDrive -Name HKCR -PSProvider Registry -Root HKEY_CLASSES_ROOT
    }
    Out-Null
}
```



```
}
    Remove-Item -Path "HKCR:\Applications\photoviewer.dll\shell\open" -
Recurse -ErrorAction SilentlyContinue
}

# Disable search for app in store for unknown extensions
Function DisableSearchAppInStore {
    Write-Host "Disabling search for app in store for unknown extensions..."
    If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\Explorer")) {
        New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\Explorer"
    | Out-Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Explorer" -Name
"NoUseStoreOpenWith" -Type DWord -Value 1
}

# Enable search for app in store for unknown extensions
Function EnableSearchAppInStore {
    Write-Host "Enabling search for app in store for unknown extensions..."
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Explorer" -Name
"NoUseStoreOpenWith" -ErrorAction SilentlyContinue
}

# Disable 'How do you want to open this file?' prompt
Function DisableNewAppPrompt {
    Write-Host "Disabling 'How do you want to open this file?' prompt..."
    If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\Explorer")) {
        New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\Explorer"
    | Out-Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Explorer" -Name "NoNewAppAlert" -
Type DWord -Value 1
}

# Enable 'How do you want to open this file?' prompt
Function EnableNewAppPrompt {
    Write-Host "Enabling 'How do you want to open this file?' prompt..."
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Explorer" -Name "NoNewAppAlert" -
ErrorAction SilentlyContinue
}

# Enable F8 boot menu options
Function EnableF8BootMenu {
    Write-Host "Enabling F8 boot menu options..."
    bcdedit /set `{current`} bootmenupolicy Legacy | Out-Null
}
```

```
# Disable F8 boot menu options
Function DisableF8BootMenu {
    Write-Host "Disabling F8 boot menu options..."
    bcdedit /set `{current`} bootmenupolicy Standard | Out-Null
}

# Set Data Execution Prevention (DEP) policy to OptOut
Function SetDEPOptOut {
    Write-Host "Setting Data Execution Prevention (DEP) policy to OptOut..."
    bcdedit /set `{current`} nx OptOut | Out-Null
}

# Set Data Execution Prevention (DEP) policy to OptIn
Function SetDEPOptIn {
    Write-Host "Setting Data Execution Prevention (DEP) policy to OptIn..."
    bcdedit /set `{current`} nx OptIn | Out-Null
}

#####
# Server specific Tweaks
#####

# Hide Server Manager after login
Function HideServerManagerOnLogin {
    Write-Host "Hiding Server Manager after login..."
    If (!(Test-Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Server\ServerManager")) {
        New-Item -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Server\ServerManager" -Force |
Out-Null
    }
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Server\ServerManager" -Name
"DoNotOpenAtLogon" -Type DWord -Value 1
}

# Hide Server Manager after login
Function ShowServerManagerOnLogin {
    Write-Host "Showing Server Manager after login..."
    Remove-ItemProperty -Path
"HKLM:\SOFTWARE\Policies\Microsoft\Windows\Server\ServerManager" -Name
"DoNotOpenAtLogon" -ErrorAction SilentlyContinue
}

# Disable Shutdown Event Tracker
Function DisableShutdownTracker {
    Write-Host "Disabling Shutdown Event Tracker..."
}
```

```
If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows
NT\Reliability")) {
    New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows
NT\Reliability" -Force | Out-Null
}
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows
NT\Reliability" -Name "ShutdownReasonOn" -Type DWord -Value 0
}

# Enable Shutdown Event Tracker
Function EnableShutdownTracker {
    Write-Host "Enabling Shutdown Event Tracker..."
    Remove-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows
NT\Reliability" -Name "ShutdownReasonOn" -ErrorAction SilentlyContinue
}

# Disable password complexity and maximum age requirements
Function DisablePasswordPolicy {
    Write-Host "Disabling password complexity and maximum age
requirements..."
    $tmpfile = New-TemporaryFile
    secedit /export /cfg $tmpfile /quiet
    (Get-Content $tmpfile).Replace("PasswordComplexity = 1",
"PasswordComplexity = 0").Replace("MaximumPasswordAge = 42",
"MaximumPasswordAge = -1") | Out-File $tmpfile
    secedit /configure /db "$env:SYSTEMROOT\security\database\local.sdb"
/cfg $tmpfile /areas SECURITYPOLICY | Out-Null
    Remove-Item -Path $tmpfile
}

# Enable password complexity and maximum age requirements
Function EnablePasswordPolicy {
    Write-Host "Enabling password complexity and maximum age
requirements..."
    $tmpfile = New-TemporaryFile
    secedit /export /cfg $tmpfile /quiet
    (Get-Content $tmpfile).Replace("PasswordComplexity = 0",
"PasswordComplexity = 1").Replace("MaximumPasswordAge = -1",
"MaximumPasswordAge = 42") | Out-File $tmpfile
    secedit /configure /db "$env:SYSTEMROOT\security\database\local.sdb"
/cfg $tmpfile /areas SECURITYPOLICY | Out-Null
    Remove-Item -Path $tmpfile
}

# Disable Ctrl+Alt+Del requirement before login
Function DisableCtrlAltDelLogin {
    Write-Host "Disabling Ctrl+Alt+Del requirement before login..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" -Name
"DisableCAD" -Type DWord -Value 1
}
```

```
# Enable Ctrl+Alt+Del requirement before login
Function EnableCtrlAltDelLogin {
    Write-Host "Enabling Ctrl+Alt+Del requirement before login..."
    Set-ItemProperty -Path
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" -Name
"DisableCAD" -Type DWord -Value 0
}

# Disable Internet Explorer Enhanced Security Configuration (IE ESC)
Function DisableIEEnhancedSecurity {
    Write-Host "Disabling Internet Explorer Enhanced Security Configuration
(IE ESC)..."
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Active Setup\Installed
Components\{A509B1A7-37EF-4b3f-8CFC-4F3A74704073}" -Name "IsInstalled" -Type
DWord -Value 0
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Active Setup\Installed
Components\{A509B1A8-37EF-4b3f-8CFC-4F3A74704073}" -Name "IsInstalled" -Type
DWord -Value 0
}

# Enable Internet Explorer Enhanced Security Configuration (IE ESC)
Function EnableIEEnhancedSecurity {
    Write-Host "Enabling Internet Explorer Enhanced Security Configuration
(IE ESC)..."
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Active Setup\Installed
Components\{A509B1A7-37EF-4b3f-8CFC-4F3A74704073}" -Name "IsInstalled" -Type
DWord -Value 1
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Active Setup\Installed
Components\{A509B1A8-37EF-4b3f-8CFC-4F3A74704073}" -Name "IsInstalled" -Type
DWord -Value 1
}

#####
# Auxiliary Functions
#####

Function WaitForKey {
    Write-Host
    Write-Host "Press any key to continue..." -ForegroundColor Black -
BackgroundColor White
    [Console]::ReadKey($true) | Out-Null
}

Function Restart {
    Write-Host "Restarting..."
    Restart-Computer
}
```

```
#####  
# Parse parameters and apply tweaks  
#####  
  
# Ask for elevated privileges if required  
If  
(!([Security.Principal.WindowsPrincipal][Security.Principal.WindowsIdentity]  
::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole]"Administrat  
or")) {  
    Start-Process powershell.exe "-NoProfile -ExecutionPolicy Bypass -File  
`"$PSCommandPath`" $args" -WorkingDirectory $pwd -Verb RunAs  
    Exit  
}  
  
# Load function names from command line arguments or a preset file  
If ($args) {  
    $tweaks = $args  
    If ($args[0].ToLower() -eq "-preset") {  
        $tweaks = Get-Content $args[1] -ErrorAction Stop | ForEach {  
$_.Trim() } | Where { $_ -ne "" -and $_[0] -ne "#" }  
    }  
}  
  
# Call the desired tweak functions  
$tweaks | ForEach { Invoke-Expression $_ }
```

Last update: **2017/09/04 13:56**