# Add registry

REG ADD "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\LogonUI\Background" /V "OEMBackground" /T REG\_DWORD /D "1" /F

# Remove registry

REG delete "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\OEMInformation" /V "Manufacturer" /F

# Set Registry for all users

[scriptblock]$HKCURegistrySettings = {

Set-RegistryKey -Key 'HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\' -Name 'WallpaperStyle' -Value '2' -Type String -SID $UserProfile.SID}

Invoke-HKCURegistrySettingsForAllUsers -RegistrySettings $HKCURegistrySettings

# Uninstall application base on file version

  if (**Test-Path** "$env:ProgramFiles\SpellRight\SpellRight.exe")

  {

   $SpellRighVersion = [System.Diagnostics.FileVersionInfo]::GetVersionInfo("$env:ProgramFiles\SpellRight\SpellRight.exe").FileVersion

Write-Log "SpellRight installed - version found: $SpellRighVersion"

if ($appVersion -ne $SpellRighVersion)

{

Execute-Process *-Path* "taskkill.exe" *-Parameters* "/im oribicom.exe /f" *-ContinueOnError* $true *-WindowStyle* 'Hidden'

**Copy-Item** -Path "$dirsupportfiles\INSTALL.LOG" -Destination "$env:TEMP" -Force

Execute-Process *-Path* "$dirsupportfiles\Unwise.exe" *-Parameters* "/S `"$env:TEMP\INSTALL.LOG`"" *-ContinueOnError* $true *- WindowStyle* 'Hidden'

    }

}

# Reboot pending workaround

# Check reboot pending – fix

**REG** delete "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager" /V "PendingFileRenameOperations" /F

if (**Test-Path** "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update\RebootRequired")

{

# Back up Registry key

Execute-Process *-Path* "reg.exe" *-Parameters* "export `"HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update\RebootRequired`" `"$env:TEMP\Regbackup.reg`" /y"

**Remove-Item** -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update\RebootRequired" -Recurse -Force

}

# Restore registry for reboot pending

if (**Test-Path** "$env:TEMP\Regbackup.reg")

{

Execute-Process *-Path* "reg.exe" *-Parameters* "import `"$env:TEMP\Regbackup.reg`""

**Remove-Item** -Path "$env:TEMP\Regbackup.reg" -Force

}

# Give user permissions to folder (NTFS)

$FullPath = "${env:ProgramFiles(x86)}" + "\RCO Security AB"

**icacls** ("$FullPath") /Grant "\*S-1-5-32-545:(OI)(CI)M" /T

**icacls** ("$FullPath") /Grant "\*S-1-5-32-545:M" /T

# Give user permissions to registry

**Start-Process** -FilePath "$dirsupportfiles\SetACL\SetACL.exe" -ArgumentList "-on `"hklm\SYSTEM\CurrentControlSet\Services\IPPTTapDevice`" -ot reg -actn ace -ace `"n:S-1-5-32-545;p:full`" -rec yes" -NoNewWindow -Wait -PassThru -ErrorAction SilentlyContinue

**Start-Process** -FilePath "$dirsupportfiles\SetACL\SetACL.exe" -ArgumentList "-on `"hklm\SYSTEM\CurrentControlSet\Services\evserial`" -ot reg -actn ace -ace `"n:S-1-5-32-545;p:full`" -rec yes" -NoNewWindow -Wait -PassThru -ErrorAction SilentlyContinue

**Start-Process** -FilePath "$dirsupportfiles\SetACL\SetACL.exe" -ArgumentList "-on `"hklm\SYSTEM\CurrentControlSet\Services\VSBC`" -ot reg -actn ace -ace `"n:S-1-5-32-545;p:full`" -rec yes" -NoNewWindow -Wait -PassThru -ErrorAction SilentlyContinue

#copy serversetting-file to each user with profile on computer

$ProfilePaths = Get-UserProfiles | Select-Object -ExpandProperty 'ProfilePath'

ForEach ($Profile in $ProfilePaths)

{

New-Item -Path "$Profile\AppData\Roaming\ABBSG" -Name ip800xAExcelInterface -ItemType Directory

Copy-File -Path "$dirSupportFiles\server.setting" -Destination “$Profile\AppData\Roaming\ABBSG\ip800xAExcelInterface\server.setting"

}