<#

Import-Module "C:\Users\amadmin\Desktop\Auto Reports\Report Scripts\AD\ADCleanup-Disable.psm1" -Verbose -Force

Import-Module "C:\Users\amadmin\Desktop\Auto Reports\Report Scripts\AD\ADCleanup-Alert.psm1" -Verbose -Force

Import-Module "C:\Users\amadmin\Desktop\Auto Reports\Report Scripts\AD\ADCleanup-Report.psm1" -Verbose -Force

#>

$reportDate = Get-Date -Format 'MMMM yyyy'

$location = "Kenya"

$alertLocation = "ICRAF"

$selectObject = @('Name', 'samAccountName', 'ObjectClass', 'AccountExpirationDate', 'lastLogonDate', 'Enabled', 'PasswordNeverExpires')

$ICRAF\_ExcludedOU = @('ICRAF BOT','ICRAF Spouses','Disabled accounts','ICRAF Kenya General Accounts','ICRAF Administrators','ICRAF MFI','ICRAF Meeting Rooms','ICRAF ICT','ICRAF OCS','EC Regreening Project','ICRAF Sharepoint Users','ICRAF Kenya Shared Mailboxes','ICRAF Kenya Service Accounts') #update with KE data

$ICRAFOU = 'OU=ICRAFHUB,DC=CIFOR-ICRAF,DC=ORG'

$ICRAFComputersOU = 'OU=Computers,OU=ICRAFHUB,DC=CIFOR-ICRAF,DC=ORG'

$inactiveDisabledOU = 'OU=Disabled Due To Inactivity,OU=Disabled accounts,OU=ICRAF Kenya,OU=ICRAFHUB,DC=CIFOR-ICRAF,DC=ORG'

$reportDirectory = 'C:\Users\lkavoo\Desktop\Auto Reports\Report Results\AD\ICRAF\'

$reportDirectoryCurrent = $reportDirectory + $reportDate + '\'

$compressDirectory = $reportDirectoryCurrent + '\*'

$compressedDirectory = $reportDirectoryCurrent + 'ADReport.zip'

$ignoreList = Get-Content ($reportDirectory + 'ICRAF-IgnoreList.csv')

$ICRAFReportRecipient = @('A.Mulugeta@cgiar.org','R.Kande@cgiar.org','P.Oyuko@cgiar.org','L.Kavoo@cgiar.org','S.Mueke@cgiar.org') #update with ICRAF Recipients

$ICRAFAlertRecipient = @('ICTILRIICRAFEndpoint@cgiar.org','ILRI-ICRAFICTInfrastructureTeam@cgiar.org','l.kavoo@cgiar.org','S.Gachigi@cgiar.org','L.kavoo@cgiar.org','L.Kavoo@cgiar.org','l.kavoo@cifor-icraf.org','l.kavoo@cgiar.org','l.kavoo@cgiar.org','l.kavoo@cgiar.org','S.mueke@cgiar.org','lkadmin@cifor-icraf.org') #update with KE data

$alertFlag = 0

$excUsername = 'ILRIICRAFSpanningAdmin@cgiar.org'

$excPassword = ConvertTo-SecureString -String 'ycgtpghkghyvfhvv' -AsPlainText -Force #Change to secure mode credential after testing

$excCreds = New-Object System.Management.Automation.PSCredential($excUsername,$excPassword)

$excURI = 'https://outlook.office365.com/powershell-liveid/'

$excSession = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri $excURI -Credential $excCreds -Authentication Basic -AllowRedirection

#Expired Accounts

$ICRAFExpiredAccounts = Search-ADAccount -SearchBase $ICRAFOU -AccountExpired | Where 'DistinguishedName' -NotMatch ($ICRAF\_ExcludedOU -join '|')

#Expired Accounts Above Six Month

$ICRAFExpiredAccounts\_over180 = Search-ADAccount -SearchBase $ICRAFOU -AccountExpired | Where 'DistinguishedName' -NotMatch ($ICRAF\_ExcludedOU -join '|') | Where 'AccountExpirationDate' -lt ((get-date).AddDays(-180))

#Accounts Without Expiry Date

$ICRAFNonExpiringAccounts = Get-ADUser -LDAPFilter '(|(accountExpires=0)(accountExpires=9223372036854775807))' -SearchBase $ICRAFOU | Where 'DistinguishedName' -NotMatch ($ICRAF\_ExcludedOU -join '|')

#Inactive Accounts

$ICRAFInactiveAccounts = Search-ADAccount -SearchBase $ICRAFOU -AccountInactive -TimeSpan 30.00:00:00 -UsersOnly | Where 'Enabled' -eq 'True' | Where 'DistinguishedName' -NotMatch ($ICRAF\_ExcludedOU -join '|')

#Inactive Accounts For 180 Days

$ICRAFInactiveAccounts\_over180 = Search-ADAccount -SearchBase $ICRAFOU -AccountInactive -TimeSpan 180.00:00:00 -UsersOnly | Where 'Enabled' -eq 'True' | Where 'DistinguishedName' -NotMatch ($ICRAF\_ExcludedOU -join '|')

$ICRAFInactiveAccounts\_over180 = $ICRAFInactiveAccounts\_over180 | Get-ADUser -Properties WhenCreated | Where WhenCreated -lt (Get-Date).AddDays(-30)

#Inactive Computers For 180 Days

$ICRAFInactiveComputers\_over180 = Search-ADAccount -SearchBase $ICRAFComputersOU -AccountInactive -TimeSpan 180.00:00:00 -ComputersOnly | Where 'Enabled' -eq 'True'

#Accounts With Password Set Not To Expire

$ICRAFAccounts\_PasswordsNeverExpire = Search-ADAccount -SearchBase $ICRAFOU -PasswordNeverExpires | Where 'Enabled' -eq 'True' | Where 'DistinguishedName' -NotMatch ($ICRAF\_ExcludedOU -join '|')

#Accounts Disabled for 2 moths and above

$ICRAFDisabledAccounts\_over60 = Search-ADAccount -SearchBase $ICRAFOU -AccountDisabled | Where "LastLogonDate" -lt ((get-date).AddDays(-60))

#Generate Report From Exchange Online Session

Import-PSSession $excSession

$mailboxCount = (Get-Recipient -Filter {(Company -eq 'ICRAF' -and RecipientType -eq 'UserMailbox') -or (City -eq 'Gigiri' -and RecipientType -eq 'UserMailbox')} -ResultSize Unlimited).count

Get-PSSession | Remove-PSSession

$userAccountCount = (Get-ADUser -Filter \* -SearchBase $ICRAFOU).count

$computerCount = (Get-ADComputer -Filter \* -SearchBase $ICRAFComputersOU).count

#Generate Report Object

$ADFullReport = [pscustomobject]@{

'No of MailBox' = $mailboxCount

'No of User Accounts' = $userAccountCount

'No of Computers' = $computerCount

'Expired Accounts' = $ICRAFExpiredAccounts.Count

'Expired Accounts Over 6 Months' = $ICRAFExpiredAccounts\_over180.Count

'Accounts Without Expire Date' = $ICRAFNonExpiringAccounts.Count

'Inactive Accounts' = $ICRAFInactiveAccounts.Count

'Inactive Accounts Over 6 Months' = $ICRAFInactiveAccounts\_over180.Count

'Inactive Computers Over 6 Months' = $ICRAFInactiveComputers\_over180.Count

'Accounts With Non Expiring Passwords' = $ICRAFAccounts\_PasswordsNeverExpire.Count

'Disabled Accounts Over 2 Months' = $ICRAFDisabledAccounts\_over60.Count

}

#Store the report in a string

$reportBody = @"

No of MailBox: $mailboxCount

No of User Accounts: $userAccountCount

No of Computers: $computerCount

Expired Accounts: $($ICRAFExpiredAccounts.Count)

Expired Accounts Over 6 Months: $($ICRAFExpiredAccounts\_over180.Count)

Accounts Without Expire Date: $($ICRAFNonExpiringAccounts.Count)

Inactive Accounts: $($ICRAFInactiveAccounts.Count)

Inactive Accounts Over 6 Months: $($ICRAFInactiveAccounts\_over180.Count)

Inactive Computers Over 6 Months: $($ICRAFInactiveComputers\_over180.Count)

Accounts With Non Expiring Passwords: $($ICRAFAccounts\_PasswordsNeverExpire.Count)

Disabled Accounts Over 2 Months: $($ICRAFDisabledAccounts\_over60.Count)

"@

#Generate Report Files

#Create Directory

mkdir $reportDirectoryCurrent -Force

$ICRAFExpiredAccounts | Select $selectObject | Export-Csv ($reportDirectoryCurrent + 'Expired Accounts.csv') -NoTypeInformation

$ICRAFExpiredAccounts\_over180 | Select $selectObject | Export-Csv ($reportDirectoryCurrent + 'Expired Accounts Over 6 Months.csv') -NoTypeInformation

$ICRAFNonExpiringAccounts | Select $selectObject | Export-Csv ($reportDirectoryCurrent + 'Accounts Without Expire Date.csv') -NoTypeInformation

$ICRAFInactiveAccounts | Select $selectObject | Export-Csv ($reportDirectoryCurrent + 'Inactive Accounts.csv') -NoTypeInformation

$ICRAFInactiveAccounts\_over180 | Select $selectObject | Export-Csv ($reportDirectoryCurrent + 'Inactive Accounts Over 6 Months.csv') -NoTypeInformation

$ICRAFInactiveComputers\_over180 | Select $selectObject | Export-Csv ($reportDirectoryCurrent + 'Inactive Computers Over 6 Months.csv') -NoTypeInformation

$ICRAFAccounts\_PasswordsNeverExpire | Select $selectObject | Export-Csv ($reportDirectoryCurrent + 'Accounts With Non Expiring Passwords.csv') -NoTypeInformation

$ICRAFDisabledAccounts\_over60 | Select $selectObject | Export-Csv ($reportDirectoryCurrent + 'Disabled Accounts Over 2 Months.csv') -NoTypeInformation

$ADFullReport | Export-Csv ($reportDirectoryCurrent + 'Full Report.csv') -NoTypeInformation

#Compress The Directory

Compress-Archive -Path $compressDirectory -DestinationPath $compressedDirectory -Force

#Send-Report -recipients $ICRAFReportRecipient -attachmentPathZIP $compressedDirectory -reportBody $reportBody

#Update Report Database

$PercentageAccountsWithNonExpiringPasswords = [math]::Round((@($ICRAFAccounts\_PasswordsNeverExpire).count/$userAccountCount),2)

$PercentageAccountsWithoutExpiryDates = [math]::Round((@($ICRAFNonExpiringAccounts).count/$userAccountCount),2)

$PercentageDisabledAccountsOver60Days = [math]::Round((@($ICRAFDisabledAccounts\_over60).count/$userAccountCount),2)

$PercentageDormantComputersOver180Days = [math]::Round((@($ICRAFInactiveComputers\_over180).count/$computerCount),2)

$PercentageDormantUsersOver180Days = [math]::Round((@($ICRAFInactiveAccounts\_over180).count/$userAccountCount),2)

$PercentageExpiredAccountsOver180Days = [math]::Round((@($ICRAFExpiredAccounts\_over180).count/$userAccountCount),2)

$PercentageExpiredAccounts = [math]::Round((@($ICRAFExpiredAccounts).count/$userAccountCount),2)

$body = @{

"Date" = $reportDate;

"Recorded Date" = $weekOfMonthNo;

"Total No of AD Users" = $userAccountCount;

"Total No of AD Computers" = $computerCount;

"Total No of Mailbox" = $mailboxCount;

"Total No of Expired Accounts" = @($ICRAFExpiredAccounts).count;

"Total No of Expired Accounts (Over 180 Days)" = @($ICRAFExpiredAccounts\_over180).count;

"Total No of Dormant Users (Over 180 Days)" = @($ICRAFInactiveAccounts\_over180).count;

"Total No of Dormant Computers (Over 180 Days)" = @($ICRAFInactiveComputers\_over180).count;

"Total No of Disabled Accounts (Over 60 Days)" = @($ICRAFDisabledAccounts\_over60).count;

"Total No of Accounts With Non Expiring Password" = @($ICRAFAccounts\_PasswordsNeverExpire).count;

"Total No of Accounts Without Expiry Dates" = @($ICRAFNonExpiringAccounts).count;

"Percentage of Accounts with Non Expiring Password" = $PercentageAccountsWithNonExpiringPasswords;

"Percentage of Accounts without Expiry Dates" = $PercentageAccountsWithoutExpiryDates;

"Percentage of Disabled Accounts (Over 60 Days)" = $PercentageDisabledAccountsOver60Days;

"Percentage of Dormant Computers (Over 180 Days)" = $PercentageDormantComputersOver180Days;

"Percentage of Dormant Users (Over 180 Days)" = $PercentageDormantUsersOver180Days;

"Percentage of Expired Accounts (Over 180 Days)" = $PercentageExpiredAccountsOver180Days;

"Percentage of Expired Accounts" = $PercentageExpiredAccounts;

}

$body

<#

$uri = "https://prod-223.westeurope.logic.azure.com:443/workflows/4787a965cace4f6e95024f4d133d973c/triggers/manual/paths/invoke?api-version=2016-06-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=oWLeCyWtPVOfLHN1eCCX6iawNeiPyWg8qU8H0Ki2qUI"

Invoke-RestMethod -Uri $uri -Method Post -Body ($body | ConvertTo-Json) -ContentType "application/json"

#End of Update

if($ICRAFInactiveAccounts\_over180.count -notmatch '0')

{

#Send First Alert

$firstNotficationDelay = (((get-date).AddDays(6)) - (get-date)).totalseconds

$reportString = $ICRAFInactiveAccounts\_over180.Name

$reportString = $reportString -join "`n"

#Generate Inactive Accounts List To CSV

$inactiveUsers\_Report = $reportDirectory + 'ICRAF-InActive Users Over 180Days.csv'

$ICRAFInactiveAccounts\_over180 | Export-Csv -Path $inactiveUsers\_Report -NoTypeInformation

#Send alert about disabled users. Change the recipients parameter with alert variable.

Send-Alert -recipients $ICRAFAlertRecipient -reportDirectory $inactiveUsers\_Report -inactiveUsers $reportString -alertFlag $alertFlag -repLocation $alertLocation

#Pausing Script For 6 Days

Start-Sleep -Seconds $firstNotficationDelay

#Send Reminder

$alertFlag = 1

$ICRAFInactiveAccounts\_over180 = Search-ADAccount -SearchBase $ICRAFOU -AccountInactive -TimeSpan 180.00:00:00 -UsersOnly | Where 'Enabled' -eq 'True' | Where 'DistinguishedName' -NotMatch ($ICRAF\_ExcludedOU -join '|')

$ICRAFInactiveAccounts\_over180 = $ICRAFInactiveAccounts\_over180 | Get-ADUser -Properties WhenCreated | Where WhenCreated -lt (Get-Date).AddDays(-30)

$reportString = $ICRAFInactiveAccounts\_over180.Name

$reportString = $reportString -join "`n"

$secondNotificationDelay = ((get-date).AddDays(1) - (get-date)).totalseconds

#Generate Inactive Accounts List To CSV

$inactiveUsers\_Report = $reportDirectory + 'ICRAF-InActive Users Over 180Days.csv'

$ICRAFInactiveAccounts\_over180 | Export-Csv -Path $inactiveUsers\_Report -NoTypeInformation

Send-Alert -recipients $ICRAFAlertRecipient -reportDirectory $inactiveUsers\_Report -inactiveUsers $reportString -alertFlag $alertFlag -repLocation $alertLocation

#Pausing Script For 1 Day

Start-Sleep -Seconds $secondNotificationDelay

#Send Notice of Disabled Accounts

$alertFlag = 2

$ICRAFInactiveAccounts\_over180 = Search-ADAccount -SearchBase $ICRAFOU -AccountInactive -TimeSpan 180.00:00:00 -UsersOnly | Where 'Enabled' -eq 'True' | Where 'DistinguishedName' -NotMatch ($ICRAF\_ExcludedOU -join '|')

$ICRAFInactiveAccounts\_over180 = $ICRAFInactiveAccounts\_over180 | Get-ADUser -Properties WhenCreated | Where WhenCreated -lt (Get-Date).AddDays(-30)

$reportString = $ICRAFInactiveAccounts\_over180.Name

$reportString = $reportString -join "`n"

#Generate Inactive Accounts List To CSV

$inactiveUsers\_Report = $reportDirectory + 'ICRAF-InActive Users Over 180Days.csv'

$ICRAFInactiveAccounts\_over180 | Export-Csv -Path $inactiveUsers\_Report -NoTypeInformation

Send-Alert -recipients $ICRAFAlertRecipient -reportDirectory $inactiveUsers\_Report -inactiveUsers $reportString -alertFlag $alertFlag -repLocation $alertLocation

DisableAccounts -inactiveUsers $ICRAFInactiveAccounts\_over180 -disabledOU $inactiveDisabledOU

}

else

{

Write-Output "No Inactive Accounts Found"

}

#>