```powershell
#Send-SharedUserUpdateToWVUM.ps1
#Written by: Jeff Brusoe
#Last Updated: November 11, 2020

#This file sends updated shared user email addresses back to WVUM.

[CmdletBinding()]
param (
    [ValidateNotNullOrEmpty()]
    [string]$WVUMOutputDirectory="\\vm-itwebnew\utl_file_txt\SailPoint_Feed\Shared\"
)

#Configure Environment
try {

    Set-HSCEnvironment -ErrorAction Stop
    Connect-HSCExchangeOnlineV1 -ErrorAction Stop

    if (Test-Path $WVUMOutputDirectory) {
        Write-Output "WVUM Output directory exists"
    }
    else {
        Write-Warning "WVUM Output directory doesn't exist"
        Invoke-HSCExitCommand -ErrorCount $Error.Count
    }
}
catch {
    Write-Warning "Unable to configure environment"
    Invoke-HSCExitCommand -ErrorAction $Error.Count
}

#Configure output file paths
$FileName = (Get-Date -format yyyy-MM-dd-HH-mm) + "-HSCSharedUserEmails.csv"
$GitHubOutputFile = "$PSScriptRoot\Logs\$FileName"
$WVUMOutputFile = "$WVUMOutputDirectory\$FileName"

Write-Output "GitHub Output File: $GitHubOutputFile"
Write-Output "WVUM Output File: $WVUMOutputFile"

#Get O365 SQL instance connection string
try {
    $SQLPassword = Get-HSCSQLPassword -Verbose

    $ConnectionStringParams = @{
        DataSource = "hscpowershell.database.windows.net"
        Database = "HSCPowerShell"
        UserName = "HSCPowerShell"
        SQLPassword = $SQLPassword
    }

    $ConnectionString = Get-HSCSQLConnectionString @ConnectionStringParams
}
catch {
    Write-Warning "Unable to generate SQL connection string"
    Invoke-HSCExitCommand -ErrorCount $Error.Count
}

#Get SharedUsers with HSC Email addresses
try {
    $SelectQuery = "Select * from SharedUserTable where HSCEmail = 1"

    $SharedUserParams = @{
        Query = $SelectQuery
        ConnectionString = $ConnectionString
        ErrorAction = "Stop"
        Verbose = $true
    }
    $SharedUsers = Invoke-SQLCmd @SharedUserParams
}
catch {
    Write-Warning "Unable to query DB"
    Invoke-HSCExitCommand -ErrorCount $Error.Count
}

$HSCEmailCount = 1
foreach ($SharedUser in $SharedUsers)
{
    if ($SharedUser.email -like "*.edu.*") {
        $SharedUserEmail = $SharedUser.email -replace "edu.","edu"
    }
    else {
        $SharedUserEmail = $SharedUser.email
    }

    $SharedUserEmail = $SharedUserEmail.Trim()

    Write-Output $("WVUID: " + $SharedUser.wvu_id)
    Write-Output $("uid: " + $SharedUser.uid)
    Write-Output $("Email From File: $SharedUserEmail")
    Write-Output $("HSC Email: " + $SharedUser.HSCEmail)
    Write-Output $("HSC Email Count: $HSCEmailCount")
    $HSCEmailCount++

    try {

        $PrimarySMTPParams = @{
            UserNames = $SharedUser.uid
            ErrorAction = "SilentlyContinue"
            Verbose = $true
        }
        $ADUserInfo = Get-HSCPrimarySMTPAddress @PrimarySMTPParams

        $PrimarySMTPAddress = $ADUserInfo.PrimarySMTPAddress

        Write-Output "Primary SMTP Address: $PrimarySMTPAddress"

        if ($PrimarySMTPAddress -AND ($PrimarySMTPAddress.Trim() -eq $SharedUserEmail)) {
            Write-Output "AD and file emails match"
        }
        else {
            Write-Output "AD and file emails don't match"
        }

    }
    catch {
        Write-Warning "Unable to pull primary SMTP address"
    }

    #Prepare to write to file
    try {
        $HSSharedUserObject = New-Object -TypeName PSObject -ErrorAction Stop

        $AddMemberParams = @{
            MemberType = "NoteProperty"
            Name = "WVUID"
            Value = $SharedUser.wvu_id
            ErrorAction = "Stop"
        }

        Write-Verbose "Adding WVUID to HSSharedUserObject"
        $HSSharedUserObject | Add-Member @AddMemberParams

        Write-Verbose "Adding UID to HSSharedUserObject"
        $AddMemberParams["Name"] = "UID"
        $AddMemberParams["Value"] = $SharedUser.uid
        $HSSharedUserObject | Add-Member @AddMemberParams

        Write-Verbose "Adding EmailFromFile to HSSharedUserObject"
        $AddMemberParams["Name"] = "EmailFromFile"
        $AddMemberParams["Value"] = "$ShardUserEmail"
        $HSSharedUserObject | Add-Member @AddMemberParams

        Write-Verbose "Adding ADPrimarySMTPAddress to HSSharedUserObject"
        $AddMemberParams["Name"] = "ADPrimarySMTPAddress"
        $AddMemberParams["Value"] = $PrimarySMTPAddress
        $HSSharedUserObject | Add-Member @AddMemberParams

        Write-Verbose "Adding WVUM Employee ID to HSSharedUserObject"
        $WVUMEmployeeID = Get-WVUMEnterpriseID -SAMAccountName $SharedUser.uid -Verbose
        Write-Verbose "WVUMEmployeeID: $WVUMEmployeeID"

        $AddMemberParams["Name"] = "WVUEmployeeID"
        $AddMemberParams["Value"] = $WVUMEmployeeID
        $HSSharedUserObject | Add-Member @AddMemberParams

        #Export HSSharedUserObject to output files
        $HSSharedUserObject | Export-Csv $GitHubOutputFile -Append -NoTypeInformation -ErrorAction Stop
        $HSSharedUserObject | Export-Csv $WVUMOutputFile -Append -NoTypeInformation -ErrorAction Stop
    }
    catch {
        Write-Warning "Unable to create HSSharedUserObject"
    }

    Write-Output "****************************"
}

Invoke-HSCExitCommand -ErrorCount $Error.Count
```