# Documentation for Active Directory Audit Script

## Purpose of the Script

This script automatically collects and exports data from Active Directory (AD) into CSV files. It gathers information on computers, users, groups and members, and organizational units (OUs) within the AD Domain, and saves them to a specified location for auditing or reporting purposes.

## Prerequisites

**Before Running the Script**:

- Ensure you have the necessary access and permissions to read Active Directory data.
- You must have **PowerShell** installed on your computer.
- This script was developed for **PowerShell version 5.1**. Unexpected results may occur in other versions.
- You must have **PowerShell** Set-ExecutionPolicy set to Unrestricted
- The **Active Directory module** should be available and accessible on your system. If it is not installed, please contact your IT administrator.
- Ensure the folder or path where you want the results saved exists and is writable.

## How to Run the Script

**Download the Script**: Save the PowerShell script to a location on your computer.

**Open PowerShell**:

a) Press Windows + X and choose **Windows PowerShell** (Admin).
b) Navigate to the folder where the script is located using the cd command.

**Run the Script**:

a) In PowerShell, type .\AD-Audit.ps1 and press Enter to execute the script.
b) A parameter can be given for where the generated files are to be saved. E.g. .\AD-Audit.ps1 -filepath C:\tmp

**Confirm the Execution**:

The script will display messages to the PowerShell terminal about its progress, indicating whether the operations were successful or if there were any errors.

**Important Notes**:

- If you do not have permission to run the script, contact your IT administrator.
- If no parameter value is given at execution, a default value of the home directory for the user executing the script will be used.

## Expected Output

When running the script, it will create and save 4 CSV files with AD Audit data and a log file the either the default location or a user defined path on the machine.

1. AD User Objects – [scanTime].csv
2. AD Computer Objects – [scanTime].csv

3. AD Group Member Objects – [scantime].csv
4. OU Hierarchy – [scantime].csv

AD-Audit – [scanTime].log

The script will display coloured (red for error, green for success, yellow for processing) progress and success messages to the PowerShell window, such as:

- [+] SUCCESS: $filePath Is Real
- [-] ERROR: Given path Does not exist: [file path]
- [+] SUCCESS: Writable: [file path]
- [-] ERROR: Not writable: [file path]
- [+] Success: ActiveDirectory module imported"
- [-] Error: Code 400: ActiveDirectory module not imported.
- Active Directory Audit script started
- Creating AD [Object type] data and saving to file
- SUCCESS: Exported AD Computer data to [file path].
- ERROR: Failed to write AD User data to [file path].

The script will write log messages to a log file. If an error occurs, it logs the message to the file. This helps in keeping track of the script's progress, especially when troubleshooting.

Log Messages:

- "Active Directory Audit script started"
- "Success: ActiveDirectory module imported"
- "Error: Failed to write AD data to CSV"
- "Script completed successfully"
- "Error: Code 400: ActiveDirectory module not imported."
- "Error Writing AD [Object type] data to [csvFile]"
- "Exported AD [Object type] data to [csvFile]"
- "Active Directory Audit script completed."

# Troubleshooting

## Common Issues & Solutions:

**Error Codes and Descriptions**

Throughout the script, errors are handled with clear messages that will be displayed in the PowerShell when something goes wrong.

**Error Message**:

[-] ERROR: Given path Does not exist: [file path]

- **Cause**:
  Path Does Not Exist Error. The path provided by the user does not exist or not found.
- **Solution**:
  Ensure that the folder path you provided as the -filepath parameter at run time

exists. If not, ensure the correct path is provided or create the directory where the script is expected to write files.

**Error Message**:

[-] ERROR: Not writable: [file path]

**Cause**:
Path Not Writable Error. The directory provided exists, but the script does not have permission to write files to it.

**Solution**:
Ensure the folder path provided as the -filepath parameter at run time is writable. If the issue is permission-related, ask your administrator to grant write access or choose a different folder where you have permissions.

**Error Message**:

[-] Error: Code 400: ActiveDirectory module not imported

**Cause**:
Active Directory Module Not Imported. The script failed to import the Active Directory module, which is required to interact with AD data.

**Solution**:
Confirm that the Active Directory PowerShell module is installed on your system. If it is missing, contact IT Support for assistance in installing it.

**Error Message**:

[-] ERROR: Failed to write AD [data type] data to [path]

**Cause**:
This error occurs if the script encounters an issue creating and writing the AD object output data (like AD User, Computer, or Group data) to the CSV file.

**Solution**:

- Check that the prerequisite requirements have been meet.
- Check for permission issues on the specified directory or drive
- Ensure the disk is not full and that there are no conflicts with existing files in the target directory.
- Check if the file is already open or locked by another application.
- Try running the script again with elevated privileges (Run as Administrator).

## How to Stop the Script (If required)

If the script is running and you need to stop it, simply press Ctrl + C in the PowerShell window. This will immediately halt the execution of the script

## Default Values

The script has default values which defines where the output files will be saved if the user does not specify a location, the names of the output files and timestamp formats used

**Default File Path**

**Parameter**: [string]$filePath = $HOME

**Description:**
The default value for the -filePath variable to the user's home directory. If the user doesn't define the -filepath parameter at run time, this default file path (e.g. C:\Users\Username) to save the output files will be used.

**Timestamp Format for File Naming**

**Value:** $scanTime = (Get-Date).toString(“yyyyMMddHHmmss”)

**Description:**
This generates a timestamp in the format yyyyMMddHHmmss (e.g., 20241126132811) which is used in the filenames of the output CSV and log files. This ensures that each time the script runs, the generated files will have unique names based on the date and time of execution.

**Effect:**
This prevents overwriting files by appending the timestamp to each file name, making it easy to track when the audit data was collected.

**File Names**

**Value:**

- $csvFile = “[file path]\AD Computer Objects – [scanTime].csv”
- $csvFile = “[file path]\AD User Objects – [scanTime].csv"
- $csvFile = “[file path]\AD Group Member Objects – [scanTime].csv”
- $csvFile = “[file path]\OU Hierarchy – [scanTime].csv”
- $logFile = “[file path]\AD-Audit – [scanTime].log”

**Description:**
This generates the individual file outputs from the script audit process to either the default or user defined file path a timestamp in the file. This ensures that each time the script runs, there is consistency in file output names each time the script is run with unique names based on the date and time of execution.

**Effect:**
This prevents overwriting files by appending the timestamp to each file name, making it easy to track when the audit data was collected. It makes searching for files across the local system easier as there is consistency.

## FAQ

**How Often Should I Run This Script?**
You can run this script periodically as needed, for example, during audits or for system checks.

**Can I Change the default File Location?**
Yes, you can modify the $filePath in the script to specify a different folder or location for saving the exported CSV files.

**Contact Information**:
If you encounter issues, contact IT support or the person who provided the script for assistance.