



# Stone Teams Policies Enforcer (STEPER)

Quick Reference Guide for STEPER V1.0.99  
Andrew Sharrad 09/06/24



## Software License Agreement

### Terms:

“Software” – program code, script, manuals, instructions and any other resources associated with this program, as automatically updated from time to time.

### Agreement:

By using this software, you confirm that you agree to the terms contained in this software license agreement.

No guarantees implied or otherwise are provided with this software.

Use of this software is entirely at your own risk. There are no warranties or guarantees provided as to the functionality, effectiveness, or reliability of the software.

This software is provided in the hope that it will enable System Administrators more effectively manage new users added to their Microsoft M365 online platform, however it is provided WITHOUT ANY WARRANTY and without the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

The author and Stone Technologies shall not in any circumstances have any liability for any losses or damages, including but not limited to:

- (a) loss of profits;
- (b) loss of anticipated savings;
- (c) loss of business opportunity;
- (d) loss of goodwill;
- (e) loss or corruption of data,

which may be suffered by you (or any person claiming under or through you), whether the same are suffered directly or indirectly or are immediate or consequential, and whether the same arise in contract, tort (including negligence) or otherwise howsoever, arising from the use or mis-use of this software or components, including those supplied through any automatic or manual distribution or update process.

The exclusions shall apply to the fullest extent permissible at law, but Stone Technologies does not exclude liability for:

- (a) death or personal injury caused by the negligence of Stone Technologies, its officers, employees, contractors or agents;
- (b) fraud or fraudulent misrepresentation;
- (c) any other liability which may not be excluded by law.

We recommend that you read and use this manual to fully test your intended usage, for you to prove to yourself that the software meets your needs and requirements, and for you to continue to do so on a regular basis.

No guarantees are provided with regards to the availability or effectiveness of future software updates, fixes or additional functionality, or to the availability of software support or assistance.



## Intended Use

**IMPORTANT:** Please make sure you understand this manual and fully test your intended use of the software or changes to your users using the Simulation feature.

Please do not make changes to the STEPER source file. Any changes to the source file are completely unsupported.

## Consent to Receive Updates or Changes to System Functionality Automatically

**CONSENT TO RECEIVE UPDATES:** STEPER includes functionality to update itself and related files or programs over the internet. Updates are provided (but not limited to) providing updates to, increases in or reductions to functionality, or the provision of software improvements or bugfixes.

This update functionality, referred to as “AutoUpdate”, is enabled by default.

If you do not wish to receive these updates, you have the following options:

- At your own risk, follow the instructions in this manual to [disable AutoUpdate](#)
- Alternatively, do not use the STEPER software

**CONSENT TO UPDATE SYSTEM SOFTWARE:** STEPER will install and periodically update the required system Powershell modules and related system connectivity software.

If you do not wish for your system to be modified, you have the following options:

- At your own risk, follow the instructions in this manual to [disable Powershell Updates](#)
- Alternatively, do not use the STEPER software

You may also wish to consider running STEPER in its own virtual machine, so as to avoid unwanted interactions with other applications or software that may use the same Powershell modules.

**CONSENT FOR STEPER TO DISABLE ITSELF AS REQUIRED:** STEPER includes functionality to disable the software as required. Situations where this may occur include but are not limited to requirements for bugfixes, or changes to Microsoft systems, platforms or processes that necessitate that STEPER functionality is disabled.

By default, when STEPER runs it performs an online check to see if it should be disabled.

If you do not wish for this to occur, you have the following options:

- At your own risk, follow the instructions in this manual to [disable the online update check](#) (this will also disable the AutoUpdate functionality, but not Powershell updates)
- Alternatively, do not use this STEPER software



## Introduction

STEPER is designed to ensure that User Licensing, M365 Groups and Teams Packages or Policies are automatically configured or deployed for new users in your M365 environment. It can be used in conjunction with a manual or automatic data import via Microsoft School Data Sync (SDS), or completely stand-alone.

## Manual Contents

<b>SOFTWARE LICENSE AGREEMENT</b> .....	<b>2</b>
<b>INTENDED USE</b> .....	<b>3</b>
<b>CONSENT TO RECEIVE UPDATES OR CHANGES TO SYSTEM FUNCTIONALITY AUTOMATICALLY</b> .....	<b>3</b>
<b>INTRODUCTION</b> .....	<b>4</b>
<b>MANUAL CONTENTS</b> .....	<b>4</b>
<b>CHANGE LOG</b> .....	<b>9</b>
<b>FEATURE OVERVIEW</b> .....	<b>15</b>
<b>FUNCTIONALITY OVERVIEW</b> .....	<b>16</b>
SIMULATION .....	16
PROFILES AND PROFILE NAMES .....	16
LOGGING .....	16
MANAGEMENT OF POWERSHELL.....	16
<b>GENERAL DOS AND DON'TS</b> .....	<b>17</b>
<b>GETTING STARTED</b> .....	<b>18</b>
PRE-REQUISITES .....	18
BASIC INSTALLATION .....	18
PACKAGE CONTENTS .....	18
AZURE AD, SDS (SCHOOL DATA SYNC), AND STEPER – THE PROCESS.....	19
<b>PLANNING AND PROCESSING</b> .....	<b>20</b>
PREVENTING STUDENTS FROM CREATING TEAMS .....	20
IDENTIFYING NEW USERS AND USING LICENSING GROUPS .....	22
CREATING THE ALL STAFF TEAM (ALSO KNOWN AS THE [ALLTEACHERSGROUP]) .....	23
CREATING TEAMS POLICIES.....	24
SECURITY GROUP NESTING, OR RECURSION – IMPLIED VERSUS EXPLICIT MEMBERSHIPS .....	28
<b>TESTING STEPER USING THE POWERSHELL COMMAND LINE</b> .....	<b>31</b>
<b>USING SETTINGS FILES</b> .....	<b>31</b>
CONSOLE OUTPUT .....	32
ELEVATION .....	33
COMMAND LINE OPTIONS.....	34
-SettingsFile.....	34
-MakeChanges .....	34
-UpdateModules .....	34



-Verbose .....	34
<b>CONFIGURING STEPER.....</b>	<b>35</b>
EDITING THE SETTINGS FILE .....	35
SAMPLE SETTINGS FILES .....	35
PROCESSING ORDER .....	36
[GENERAL] .....	37
[CREDENTIALS].....	38
<i>Password Encryption</i> .....	38
[TENANCY].....	39
[GLOBALADMINTEAMSCREATION].....	40
<i>DisableNameOrIDChecking</i> .....	40
SOURCE GROUPS – [TEACHERSOURCEGROUPS] AND [STUDENTSOURCEGROUPS].....	41
<i>Wildcard Processing - What If You aren't using SDS or don't have suitable Teacher or Student Source Groups? ..</i>	42
<i>What's the Difference Between Required, and RequiredOR?</i> .....	42
<i>Recursion</i> .....	43
<i>DisableNameOrIDCheckingForGroupRules</i> .....	43
<i>IncludeDistributionGroupsInGroupRules</i> .....	43
<i>NotRequiredGroups / NotRequiredRecursiveGroups: Excluding Teachers or Students from Processing</i> .....	44
<i>Source Group Domain Filtering</i> .....	45
<i>Removing Students from the Teacher Source Groups / Removing Teachers from the Student Source Groups</i> .....	46
<i>Recording the Users Available for Processing</i> .....	47
<i>Source Groups Settings Summary</i> .....	48
<i>Master List Processing Order</i> .....	49
[LICENSINGGROUPS] .....	50
USING LICENSINGGROUPS .....	50
<i>SleepSecondsAfterApplyingLicensing</i> .....	51
<i>Licensing Group Examples</i> .....	51
<i>DisableNameOrIDChecking</i> .....	51
<i>Creating Licensing Groups</i> .....	52
<i>Not-in Licensing Group (NLG) Master List</i> .....	53
[ALLTEACHERSGROUP] AND [ALLSTUDENTSGROUP] .....	54
<i>Creating Your Own Groups</i> .....	55
<i>AllTeachersGroup and AllStudentsGroup Examples</i> .....	56
<i>Creating AllTeachers Group and AllStudents Group</i> .....	56
[DEPARTMENT] .....	57
<i>Department Examples</i> .....	58
[TEACHERSECURITYGROUPX] AND [STUDENTSECURITYGROUPX].....	59
<i>Removing Stale Members from Security Groups</i> .....	62
<i>SecurityGroupName Matching</i> .....	64
<i>DisableNameOrIDChecking</i> .....	64
<i>Turning Off All Filtered Security Group Processing</i> .....	64
<i>TeacherSecurityGroupX and StudentSecurityGroupX Examples</i> .....	65
[TEACHERUNIFIEDGROUPX] AND [STUDENTUNIFIEDGROUPX].....	66
<i>Removing Stale Members from Unified Groups</i> .....	69
<i>UnifiedGroupName Matching</i> .....	71
<i>DisableNameOrIDChecking</i> .....	71
<i>Turning Off All Filtered Unified Group Processing</i> .....	71
<i>TeacherUnifiedGroupX and StudentUnifiedGroupX Examples</i> .....	72
[TEACHERDISTRIBUTIONGROUPX] AND [STUDENTDISTRIBUTIONGROUPX] .....	73
<i>Removing Stale Members from Distribution Groups</i> .....	76
<i>DistributionGroupName Matching</i> .....	78



<i>DisableNameOrIDChecking</i> .....	78
<i>Turning Off All Filtered Distribution Group Processing</i> .....	78
<i>TeacherDistributionGroupX and StudentDistributionGroupX Examples</i> .....	79
[TEACHERTEAMSPACKAGESX] AND [STUDENTTEAMSPACKAGESX] .....	80
<i>Package Batch Processing Thresholds</i> .....	83
<i>Disabled Users</i> .....	83
<i>License Checking</i> .....	84
<i>Dealing with Problems when Applying Packages</i> .....	84
<i>Turning Off All Package Processing</i> .....	84
<i>TeacherTeamsPackagesX and StudentTeamsPackagesX Examples</i> .....	85
[TEACHERTEAMPOLICIESX] AND [STUDENTTEAMPOLICIESX] .....	86
<i>Policy Batch Processing Thresholds</i> .....	90
<i>Disabled Users</i> .....	90
<i>License Checking</i> .....	90
<i>Dealing with Problems when Applying Policies</i> .....	91
<i>Turning Off All Policy Processing</i> .....	91
<i>TeacherTeamsPoliciesX and StudentTeamsPoliciesX Examples</i> .....	92
<b>APPLYING CHANGES TO ALL USERS / TEACHERS / STUDENTS [OVERRIDE] .....</b>	<b>93</b>
HOW TO OVERRIDE WHICH USERS ARE MODIFIED .....	93
<b>[INFORMATION RETRIEVAL] .....</b>	<b>95</b>
GETALLUSERPOLICYINFORMATION .....	95
GETUSERPACKAGEINFORMATION .....	96
GETUNIQUEUSERDEPARTMENTNAMES .....	96
GETPOLICYJOBS .....	96
<b>POWERSHELL FILTERS.....</b>	<b>97</b>
ATTRIBUTE FILTERS .....	97
REMOTE AND LOCAL FILTERS.....	98
<i>UseLocalAttributeFiltering</i> .....	98
<i>UseAzureADforLocalAttributeFilteringByDefault</i> .....	98
<i>Local and Remote Filtering and Sources</i> .....	99
FILTER FORMATTING OR SYNTAX.....	100
FIELD EXAMPLES.....	100
AVAILABLE ATTRIBUTE FILTER FIELDS .....	101
<b>GROUP RULES .....</b>	<b>102</b>
SETTINGS AVAILABLE .....	104
EXAMPLE .....	104
HOW TO USE THESE SETTINGS .....	105
USING COMMA SEPARATED GROUP LISTS .....	106
<i>Using ObjectIDs</i> .....	106
GROUP RECURSION.....	107
GROUP RULES PROCESS .....	108
<b>ACTIVE DIRECTORY ORGANISATIONAL UNIT (OU) / CANONICAL NAME (CN) RULES .....</b>	<b>109</b>
WHAT IS A CANONICAL NAME? .....	110
SETTINGS AVAILABLE .....	110
EXAMPLES.....	111
<b>COLLECTIONS .....</b>	<b>114</b>



COLLECTIONID PROCESSING BASICS .....	115
GLOBAL SETTINGS AVAILABLE .....	115
FUNCTION SETTINGS AVAILABLE .....	116
COLLECTIONID NUMBERING AND PROCESSING ORDER.....	117
COLLECTIONID BASIC EXAMPLES .....	117
COLLECTIONIDS: GROUPS VS. PACKAGES VS. POLICIES .....	118
DEFAULT PROCESSING .....	119
COLLECTIONIDIGNORETHRESHOLD EXAMPLE .....	120
COLLECTIONIDSTOEXCLUDE LISTS .....	121
COLLECTIONIDS COMPLEX EXAMPLES.....	122
<b>DOMAIN FILTERS.....</b>	<b>125</b>
<b>FUNCTION CONTROL.....</b>	<b>126</b>
<b>RESOLVING PROBLEMS WITH USERS.....</b>	<b>127</b>
METHODS OF RESOLVING PROBLEMS WITH USERS .....	127
METHODS OF RESOLVING STEPER FAILURES .....	127
CHOOSING USER REMEDIATION OR USER RE-PROCESSING.....	128
RECOMMENDATIONS TO AVOID THE NEED FOR REMEDIATION OR RE-PROCESSING.....	128
<b>USER REMEDIATION.....</b>	<b>129</b>
PROCESSING ALL USERS .....	129
PROCESSING NEW USERS ONLY .....	129
ENABLING REMEDIATION - REMOVEFAILEDUSERSFROMLICENSINGGROUP.....	130
REMIATION – REMEDIATING NEW FAILED USERS ONLY, OR ALL FAILED USERS.....	131
WHEN STEPER WON’T REMEDIATE .....	132
REMIATION SETTINGS SUMMARY .....	132
<b>USER RE-PROCESSING .....</b>	<b>133</b>
USER RE-PROCESSING SETTINGS SUMMARY.....	133
ENABLING USER RE-PROCESSING – GLOBAL SETTING.....	134
ENABLING USER RE-PROCESSING – FOR INDIVIDUAL FUNCTIONS.....	135
[AllTeachersGroup] / [AllStudentsGroup].....	136
[Department] .....	136
TeacherSecurityGroup / StudentSecurityGroup / TeacherUnifiedGroup / StudentUnifiedGroup /	
TeacherDistributionGroup / StudentDistributionGroup .....	137
[TeacherTeamsPackagesX] / [StudentTeamsPackagesX] / [TeacherTeamsPoliciesX] / [StudentTeamsPoliciesX]	138
[Remediation] .....	139
<b>OTHER SETTINGS AND FUNCTIONALITY.....</b>	<b>140</b>
[RESULTSEMAIL] .....	140
[POWERSHELL] .....	140
AutoSetTLSSecurity .....	140
DisableMonthlyModuleUpdateChecks.....	140
ForceUpdateModules.....	140
[MISC] .....	141
AutoUpdate.....	142
Azure AD Access Policies .....	143
AzureAD Access Policies – Recommended Settings.....	145
Batch Processing Thresholds.....	146
CheckForUpdates .....	146
Console Warning Suppression.....	146



<i>DontRunIfStepperMayBeRunningTimeOutInMinutes</i> .....	147
<i>IgnoreMissingNETFramework</i> .....	147
<i>Logging</i> .....	148
<i>MaximumBatchSize</i> .....	149
<i>MaxSettingsGroups</i> .....	149
<i>Pauses for Licensing or User Changes</i> .....	150
REFRESHING TEACHER OR STUDENT LICENSING STATUS AFTER ADDING USERS TO LICENSING GROUPS .....	151
<i>UseCSVforUserDataInLogFile</i> .....	152
<i>User Attribute Filters (Local or Remote, AzureAD or CSOnline)</i> .....	152
<b>TROUBLESHOOTING</b> .....	<b>153</b>
<b>SETTINGS REFERENCE</b> .....	<b>155</b>





## Change Log

Date	Version	Changes
8/8/20	1.01	-Azure AD Access Token Lifetime Policy functionality is now aware of multiple instances of STEPER running on the same machine, with different STEPER profiles. For example, running STEPER with a profile of (Teachers) and a profile of (Students) – if the both need to set the Azure AD Access Token Lifetime Policy, the first script to run will set the policy. The second will re-use the existing policy. Whichever script finishes first will leave the policy removal to the second instance
11/8/20	1.02	-Added the ability to optionally specify the source Teacher or Student groups via Object ID, rather than group name
12/8/20	1.03	-Added the ability to optionally specify groups via ID, rather than group name, for [TeacherSecurityGroupX] / [StudentSecurityGroupX] / [TeacherUnifiedGroupX] / [StudentUnifiedGroupX]
13/8/20	1.04	-Minor bug fixes; Added DontCheckOnMicrosoftCOMPrefixWhenRemovingRemediation to Azure AD Access Token Lifetime Policy functionality
13/8/20	1.05	Minor bug fixes; Changed the default profile name to include the tenancy name
16/8/20	1.0.6	-Minor bug fixes; Improved console output when Filtered Security Groups, Unified Groups, Teams Policy Packages or Teams Policies were not found
23/8/20	1.0.7	-Added Strict Name/DisplayName vs. GUID/ID Matching for Filtered Unified Groups -Minor changes to the Teams Policy Batch jobs names and console display
16/9/20	1.0.8	-Added compatibility work around for Windows 10 1803 or older (AutoTLS); Minor bugfixes
17/9/20	1.0.9	-Reliability enhancement for slow response from MS platforms after importing Powershell sessions
18/9/20	1.0.10	-Improvement to work around S4B and Teams Powershell when S4B Module has been installed first
22/9/20	1.011	-Additional work-around for early Server 2016 builds -Minor Improvement to Set Student Grade for progress indication
09/10/20	1.0.12	-Solved issue where Teachers or Students would not be added to licensing groups when the groups were empty -Updated Nuget Package Provider Installation to support 3.0.0.1
11/10/20	1.0.13	-Added Nuget Package Provider upgrade to monthly module update check
14/10/20	1.0.14	-Added additional warning or error if the Enable-CsOnlineSessionForReconnection Powershell command is not available. This is in preparation for the move from the SkypeForBusiness Connector to the Teams 1.1.6 module in a future version of STEPER. Currently the Teams Powershell module does not support Enable-CsOnlineSessionForReconnection
18/10/20	1.0.15	-Added MOTD functionality
21/10/20	1.0.16	-Resolved a new cosmetic issue with missing Unified groups displaying Powershell errors in the console
28/10/20	1.0.17	-Bugfixes; Added Filtered Distribution Group Functionality
28/10/20	1.0.18	-Minor bugfixes; Some refinements to the logging information
30/10/20	1.0.19	-Added the ability to STEPER to create the licensing groups if required
08/11/20	1.0.20	-Added AutoUpdate Functionality; Minor Bugfixes
11/1/21	1.0.21	<ul style="list-style-type: none"> <li>-Minor bugfixes and reliability enhancements</li> <li>-Sets the local process code execution policy to unrestricted to help install new Powershell modules</li> <li>-Added the ability to STEPER to create the All Teachers or All Students unified group if required</li> <li>-Corrected GlobalAdminTeamsCreation setting from AddGlobalAdminToCreateTeamsUnifiedGroup to AddGlobalAdminToCreateTeamsGroup – the name of the setting was incorrect. Old settings will be upgraded automatically</li> <li>-Added the ability to process the Source Groups recursively, to find implied members. Turned off by default.</li> <li>-Improved the manual with sample code for creating licensing groups, Teams policies, preventing students from creating teams, and creating the All Staff Team</li> <li>-Older settings files will now be automatically be upgraded to include some of the newer key settings with the default value set, to enable the user to change these settings faster – for example, added DisableAutoUpdate setting to settings file, if it is missing</li> <li>-Added warning if a Microsoft Teams connection is required and .NET v4 appears to be missing or too low a version</li> <li>-Added warning if the WinRM Client has Basic Authentication disabled, as this may cause Skype For Business Powershell to fail</li> <li>-Added a double check to Teams Packages and Teams Policies to check that the user is licensed before trying to make changes. This licensing check can be skipped by setting SkipUpdatedLicensingCheck=True</li> <li>-Added additional assignment/operation error checking to all functions. For example, STEPER will now capture Teams Package and Policy assignment failures, generating a warning. Assignment failures will be logged</li> <li>-Reverted the update check code to 1.0.19 for reliability purposes, to solve the Autoupdater not running correctly in 1.0.20</li> <li>-Solved an issue where specifically enabling SSL for Email in the INI file would not be effective (UseSSL=True). SSL now enabled by default but can be turned off using UseSSL=False. This change has no impact if you do not already use this setting</li> <li>-Removed support for Legacy/v1 Exchange Powershell Module</li> <li>-Captured any error message from attempting to email the final run results and add this to the log file</li> <li>-Added improved group name checking for TeacherSecurityGroupX/StudentSecurityGroupX, and corresponding DisableNameOrIDChecking setting (for example, preventing a searching for “Year11” also matching “Year11b”</li> <li>-Added a fix to TeacherSecurityGroupX/StudentSecurityGroupX to ensure that only Security/ Mail Enabled Security Groups matched</li> <li>-Added TeacherSecurityGroupX and StudentSecurityGroupX group name checking, on by default, with an option to turn it off (DisableNameorIDChecking=True)</li> <li>-Added GlobalAdminTeamsCreation, All Teachers Unified Group and All Students Unified Group group name checking, on by default, with the option to turn it off (DisableNameorIDChecking=True)</li> </ul>

		<ul style="list-style-type: none"> <li>-Added SourceGroups and LicensingGroups group name checking, on by default, with options to turn it off (DisableTeacherGroupNameorIDChecking=True, DisableStudentGroupNameorIDChecking=True)</li> <li>-Added improved group filtering to all Security-group based functions. Distribution groups will not be included by default. You can override this by specifying the Distribution group by ObjectID, or overriding this with IncludeDistributionGroups on all Group Rule filter-based functions</li> <li>-Added Group Rule filter functionality to all "bulk" functions: TeacherSecurityGroupX/StudentSecurityGroupX, TeacherUnifiedGroupX/StudentUnifiedGroupX, TeacherDistributionGroupX/StudentDistributionGroupX, TeacherTeamsPackagesX/StudentTeamsPackagesX, TeacherTeamsPoliciesX/StudentTeamsPoliciesX. Useful for RM Unify customers.</li> <li>-Added Group Rule filter "test" functionality to InformationRetrieval/GetAllUserPolicyInformation</li> <li>-Changed the default log file location to include "STEPPER Logs" in the pathname</li> </ul>
12/1/21	1.0.22	-Standardised all group name inputs. Separate settings for Group ID are no longer used. Group Names will be checked to see if they contain a GUID/object ID or a Name
12/1/21	1.0.23	-Cosmetic improvements to Log file
13/1/21	1.0.24	-Minor bugfixes
13/1/21	1.0.25	-Minor bugfixes; improvements to the example Powershell files
16/1/21	1.0.26	-Improvements to console output for error messages and Assignment Error tracking for Packages and Policies. Repeated console messages suppressed after default of 20. -Package and Policy Assignment: Detect and report Disabled Users (users Disabled in Teams)
17/1/21	1.0.27	-Improve the reporting for Packages and Policies of Unlicensed or Disabled users, and Assignment Errors
20/1/21	1.0.28	-Minor further improvements to the reporting for Packages and Policies of Unlicensed or Disabled users, and Assignment Errors
28/1/21	1.0.29	-Solved an issue where the search locations for a missing configuration file weren't displayed correctly
28/1/21	1.0.30	-Improved logging of Source and Licensing group names and ObjectIDs; Solved an instance where All Teachers or All Students Unified groups names were mismatched
13/2/21	1.0.31	-Added the ability to remediate from users from Package and Policy Assignment by removing them from the licensing group
14/2/21	1.0.32	-Added a global setting to give the ability to remediate only new users (users that weren't in a licensing group) -SleepSecondsAfterApplyingLicensing now has a default of 600 if not specified in the settings file
23/2/21	1.0.33	-Minor code improvement
28/2/21	1.0.35	<ul style="list-style-type: none"> <li>-Solved an Issue with AllTeachersUnifiedGroup and AllStudentsUnified Group where the users were not added</li> <li>-Solved an issue with Packages and Policies where filtered users should be shown as Not Enabled users, and available for remediation</li> <li>-Added the ability to install the latest PowershellGet Module in preparation for 1.0.36 and support for the new pre-release Teams Powershell module. Installing PowershellGet be disabled by settings DisablePowerShellGetCheck=True under [Powershell]</li> </ul>
02/3/21	1.0.36	<ul style="list-style-type: none"> <li>-Revised PowershellGet Module installer – STEPER will not proceed if PowershellGet version 2.x.x is not installed. This is to ensure the pre-release Teams module can be installed.</li> <li>-Add support for the new Teams Powershell Modules, replacing the SkypeforBusiness Module. Needs TeamsPowershell 1.1.11 minimum, which will be installed by STEPER (needs the latest PowershellGet Module first)</li> <li>-If the new Teams Powershell Module is in use (instead of SFB), STEPER will no longer create an AzureAD policy. However, STEPER still checks AzureAD to cleanup any previous policy</li> <li>-Add an automatic upgrade to settings files under [Powershell] called UseSkypeforBusinessPowerShell, default of False. Skype for Business Powershell use now disabled by default. - Other bugfixes and enhancements</li> </ul>
4/3/21	1.0.37	-Skype for Business Powershell is forced ON, due to some current issues with the new Teams Powershell model. The UseSkypeforBusinessPowerShell setting is ignored in the settings file
5/3/21	1.0.38	-Reinstated Teams Powershell instead of Skype for Business, with a minimum requirement of Teams Powershell 2.x.x
10/3/21	1.0.39	-The option DontPauseForChanges will no longer apply to warnings about previously uncompleted runs of STEPER. This means the warning and 20 second pause will always occur when STEPER previously failed to finish
21/3/21	1.0.40	<ul style="list-style-type: none"> <li>-Improved the error message gathering for the creation and removal of temporary Azure AD policies, plus some message fixes</li> <li>-Improved the message handling dealing with start-up issues (e.g. STEPER already running) and warning about changes being made</li> </ul>
24/3/21	1.0.41	<ul style="list-style-type: none"> <li>-Solved an issue where functions relying on AzureADPreview were not available because the AzureAD module took precedence, even if the AzureADPreview module was reinstalled. Only AzureADPreview supports working with Azure policies</li> <li>-Improved the performance of Get All User Policy Information</li> </ul>
28/3/21	1.0.42	-Updated OnMicrosoftCOMPprefix and PrivateDomain handling. OnMicrosoftCOMPprefix is now only required if SFB is being used
29/3/21	1.0.43	-Further updated the error handling for PrivateDomain. Leaving this at the default now results in a warning instead of an error
15/4/21	1.0.43	-Some corrections to the manual
18/4/21	1.0.44	<ul style="list-style-type: none"> <li>-Some minor corrections and improved consistency to the log output; Solved a minor issue with GetAllUserPolicyInformation output</li> <li>-Added [Misc] UseCSVforUserDataInLogFile, default of False. When set to True, all user log information will be added to the log file in CSV format, to make it easier for processing afterwards</li> <li>-Added [SourceGroups] LogAvailableTeachers and LogAvailableStudents. If set to true, this will record all of the users found in the source groups at the start of user processing</li> </ul>
19/4/21	1.0.45	-Resolved some other minor typos and log entry headings
27/4/21	1.0.46	-Corrected some start-up messages which could incorrectly state that there was a problem with the STEPPER settings
04/5/21	1.0.47	-Added command line and INI override options: -ApplytoAllTeachers, ApplytoAllStudents, ApplytoAll
04/5/21	1.0.48	-Minor bugfixes
09/5/21	1.0.49	-Added synthetic -Whatif support: If used, MakeChanges is always False
14/5/21	1.0.50	-Adding command line debugging to the Log File
23/6/21	1.0.51	-Added command line debugging to the event log. Using -Verbose on the CMD line also turns on LogAvailableTeachers/Students

30/8/21	1.0.52	<ul style="list-style-type: none"> <li>-Removed support for the now deprecated Skype for Business Powershell Module</li> <li>-The ability to set an AzureAD Access Policy remains for MSTeams Use. Disabled by default, and disabled even if previously enabled</li> <li>-If the new renamed CreateAzureADAccessPolicy setting is used, it is applied more consistently. The policy will be created whenever AzureAD is used. AzureAD will always be enabled if a connection to MSTeams is required</li> <li>-PrivateDomain is no longer used but the setting is not deleted from the settings file</li> <li>-Resolved an issue where AutoSetTLSecurity was not processed properly</li> <li>-Resolved an issue where a failed connection to AzureAD did not mark an AzureAD Access Policy as not being in use, in the registry</li> <li>-Added AssumeAzureADAccessPolicyNotRequiredTimeoutHours (default 24.0) for the removal of AzureAD Access Policy when the registry indicates STEPER should still be running (previously hardcoded at 24)</li> </ul>
07/09/21	1.0.53	<ul style="list-style-type: none"> <li>-Fixed some minor typos; Fixed incorrect messages displayed if package batch assignments fail</li> <li>-The log file always now records the Batch Processing Thresholds in use even if not specified in the settings file</li> <li>-Added the MaximumBatchSize parameter, default 5000. STEPER will now split Package and Policy assignment operations into smaller MaximumBatchSize batches, to meet TeamsPowershell requirements</li> <li>-Added Package Processing Batch Summary to Logfile</li> <li>-Added Policy Processing Batch Summary to Logfile</li> </ul>
09/09/21	1.0.54	<ul style="list-style-type: none"> <li>-Resolved an issue introduced in 1.0.47. Use of the ApplytoAllTeachers, ApplytoAllStudents, ApplytoAll should mean that the licensing groups for Teachers/Students/Both are not required. This bug has been fixed</li> <li>-Added RepeatLogEventThreshold wording when event log messages are suppressed, to make it clearer why they are being suppressed</li> </ul>
12/9/21	1.0.55	<ul style="list-style-type: none"> <li>-Some minor logging improvements</li> <li>-Added Windows Event log entry if the results email is not able to be sent</li> </ul>
04/11/21	1.0.56	<ul style="list-style-type: none"> <li>-Fixed some minor typos and some minor logging improvements</li> </ul>
18/12/21	1.0.57	<ul style="list-style-type: none"> <li>TLS1.2 enforcement for older builds of Windows now occurs much earlier during the process, to ensure a reliable connection to the update server. If 1.0.56 cannot reach the update server, a manual update to 1.0.57 is unfortunately required</li> </ul>
6/3/22	1.0.58	<ul style="list-style-type: none"> <li>-Some minor logging improvements</li> <li>-Resolved an issue where user data was always recorded in CSV format in log files</li> </ul>
26/3/22	1.0.59	<ul style="list-style-type: none"> <li>-Resolved a bug in filter processing for Teams Packages that could cause the wrong filter to be applied</li> </ul>
6/4/22	1.0.60	<ul style="list-style-type: none"> <li>-Resolved an issue in Filtered Unified Groups processing that did not perform fully accurate name matching</li> </ul>
16/4/22	1.0.61	<ul style="list-style-type: none"> <li>-Added default and Collection ID functionality for TeacherSecurityGroupX/StudentSecurityGroupX, TeacherUnifiedGroupX/StudentUnifiedGroupX, TeacherDistributionGroupX/StudentDistributionGroupX, TeacherTeamsPackagesX/StudentTeamsPackagesX, TeacherTeamsPoliciesX/StudentTeamsPoliciesX (settings include "Default", "CollectionID", "CollectionIDIgnoreThreshold" and CollectionIDsToExclude)</li> <li>-Using Collection IDs, it is now possible for different settings to add users to users via TeacherSecurityGroupX/StudentSecurityGroupX, TeacherUnifiedGroupX/StudentUnifiedGroupX, TeacherDistributionGroupX/StudentDistributionGroupX – this gives "or" functionality</li> <li>-Using Collection IDs, you can now apply default packages or individual policies to TeacherTeamsPackagesX/StudentTeamsPackagesX, TeacherTeamsPoliciesX/StudentTeamsPoliciesX</li> <li>-Multiple improvements and fixes to reflect recent changes to Microsoft platforms</li> <li>-Added UseLocalAttributeFiltering functionality (off by default), for all functions that use filters including GetAllUserPolicyInformation</li> <li>-Added compilation check for all filters – filters that are not complete, or don't appear to match requirements will be rejected – e.g. when UseLocalAttributeFiltering=True, filters should contain \$_. when UseLocalAttributeFiltering=False, filters should not contain \$_. </li> <li>-Added the ability to use filters against information from Azure or Teams. Azure supports Local processing only – CSOnline supports Local and Remote</li> <li>-Updated the documentation for the recently reduced Microsoft Teams filter functionality</li> <li>-Improved the amount of settings output and debug in the log file when STEPER does not run due to invalid settings</li> <li>-Cleaned up the console output, also now reflecting the CollectionID</li> <li>-Changed the Policy and Package processing so that Groups are processed earlier in the pipeline, resulting in more accurate output</li> <li>-Added support for RequiredOrGroups and RequiredOrRecursiveGroups in all group processing – the user has to be a member of any of the listed any groups. This adds basic OR group processing.</li> </ul> <p><b>1.0.61 is a major update.</b> It is recommended that you update your filter settings and then run STEPER against all of your users.</p>
25/4/22	1.0.62	<ul style="list-style-type: none"> <li>-Significantly Improve TeamsPolicy processing time when Group filters are used by caching the user results</li> </ul>
26/4/22	1.0.63	<ul style="list-style-type: none"> <li>-Resolve an issue that All Teachers/All Students Unified group processing incorrectly reported an assignment error count</li> </ul>
27/4/22	1.0.64	<ul style="list-style-type: none"> <li>-All plain text GA_Passwords are now automatically encrypted for some protection against plain text password harvesting. As with all back-end systems, the virtual machine or PC running Windows should always be protected against unauthorised access. We recommend that an adequate firewall and anti-virus solution is always used, and that end users do not login to backend systems.</li> </ul>
1/5/22	1.0.65	<ul style="list-style-type: none"> <li>-Improvements and corrections to the log output</li> </ul>
9/5/22	1.0.66	<ul style="list-style-type: none"> <li>-Improve the reliability of handling empty multi-expression settings</li> </ul>
13/5/22	1.0.67	<ul style="list-style-type: none"> <li>-Minor fixes</li> </ul>
24/5/22	1.0.68	<ul style="list-style-type: none"> <li>-Added the User Re-Processing feature, Enabled by Default, as an alternative to RemoveFailedUsersFromLicensingGroup (user Remediation). Includes the GlobalOnFailureApplytoAllUsersOnNextRun and OnFailureApplytoAllUsersOnNextRun settings</li> </ul>
1/6/22	1.0.69	<ul style="list-style-type: none"> <li>-Added ability to exclude Teachers or Students from processing – ExcludedTeachersGroupName and ExcludedStudentsGroupName</li> </ul>
3/6/22	1.0.70	<ul style="list-style-type: none"> <li>-Added protection against looped nested groups; minor reliability enhancements</li> </ul>
4/6/22	1.0.71	<ul style="list-style-type: none"> <li>-Minor performance improvements</li> </ul>
23/6/22	1.0.72	<ul style="list-style-type: none"> <li>-Minor bugfixes</li> </ul>
24/9/22	1.0.73	<ul style="list-style-type: none"> <li>-Support ExchangeOnlineV3; Improvements to Connection error message handling; Improvements to re-processing messages</li> </ul>

25/9/22	1.0.74	-Minor fixes
27/9/22	1.0.75	-Minor fixes and improvements; add warning if unrecognised command line arguments are found; some logging improvements
16/10/22	1.0.76	-Minor fixes and logging improvements. Fix MOTD not visible in logs
01/11/22	1.0.77	-Add the ability for CollectionIDsToExclude to support basic numerical ranges e.g. 5-10
2/11/22	1.0.78	-Minor fixes -CollectionIDstoExclude now supports CollectionID 0 -CollectionIDstoExclude is now not displayed in Summary tables; an amalgamated CollectionIDstoExclude_Display is used instead
3/11/22	1.0.79	-Minor fixes; Manual improvement around user re-processing -User Re-Processing for Filtered Security Groups / Unified Groups / Distribution groups now supports "GroupProblem" for when the group is missing -Running STEPER in Simulation only mode without errors will now not clear user processing requests
5/11/22	1.0.80	-Logging Improvements -Comma separated list handling/reliability improvements
25/12/22	1.0.81	-Added User State Key Table to the Log file
28/02/23	1.0.82	-Add the ability to use ALL users for source Teachers or Students (set the source group name to *) -Improved Support for ExchangeOnlineV3 -Fixed a bug where filtered group processing user removals showed the wrong group name in the summary
4/03/23	1.0.83	-Minor logging improvements -AllTeachersDomainFilter and AllStudentsDomainFilter now support a comma separated list of acceptable user email domains - TeacherSecurityGroupX/StudentSecurityGroupX, TeacherUnifiedGroupX/StudentUnifiedGroupX, TeacherDistributionGroupX/StudentDistributionGroupX: Corrected an issue where mismatched User Principal names (UPN) and Email Addresses caused the user to be filtered out even when CSONline/Azure AD user filtering was not being used. If filtering is being used, the user UPN and email address should match
10/03/23	1.0.84	-Logging improvements and fixes -Removed need for users UserPrincipalName and Email address to match. This improves the reliability of any functions relying on AzureAD or CSONline attribute filtering -Policy and package processing also now similarly perform assignment operations based on user object IDs instead of email addresses -Minor bugfixes -Improve reporting on Get All User Policy Information, with more fields displayed and improved ordering -Added a check for duplicated section names in the config file. This will stop STEPER from processing
15/04/23	1.0.85	-Add Powershell version check to warn if the Powershell version is less than v5 -Add [FunctionalityControl] to disable groups of filtered settings all in one go -Resolved an issue where existing group member matching (for user removals) for filtered groups didn't match on object ID -Further improved reporting on Get All User Policy Information, with more fields displayed -Resolved an issue where users filtered by CollectionID wouldn't be removed from Groups -Added functionality where you can remove unmatching users from Filtered groups by Attribute matching, Group Rules, OU Matching or CollectionID (RemoveUnMatchedMembers=True means all, as before) -Some improvements to how arguments that accept Boolean values process entries – 1 means True, and usually means All -Other fixes and improvements to argument processing -Fixes to GlobalOnFailureApplytoAllUsersOnNextRun where specifying True and then some other options caused unexpected results -Support on-prem OU filtering - AD OU/CN matching rules -Minor fixes -Renamed group settings – settings files will be automatically upgraded: ListofReqdGroups > RequiredGroups, ListofReqdRecursiveGroups > RequiredRecursiveGroups, ListofReqdORGroups > RequiredOrGroups, ListofReqdORRecursiveGroups > RequiredOrRecursiveGroups, ListofNotReqdGroups > NotRequiredGroups, ListofNotReqdRecursiveGroups > NotRequiredRecursiveGroups -Added user-processing in a run when STEPER detects the previous run was not completed
22/04/23	1.0.86	-Renamed group settings – settings files will be automatically upgraded: GroupFilter > Filter -Renamed package settings – settings files will be automatically upgraded: TeamsPackageFilter > Filter -Renamed policy settings – settings files will be automatically upgraded: TeamsPolicyFilter > Filter -Renamed Information Retrieval settings – settings files will be automatically upgraded: GetAllUserPolicyInformationFilter > Filter, GetAllUserPolicyInformationFilterSource > FilterSource -Renamed IncludeDistributionGroups > IncludeDistributionGroupsInGroupRules - settings files will be automatically upgraded -Resolved an issue where Teacher Filtered Unified group attribute Filters did not work -Resolved DisableNameorIDCheckingForGroupRules not working on Filtered Security Groups -Resolved an issue where the user type was properly displayed in package-failure user re-processing -STEPER will now check the usertype when checking users for Collection processing -Added [Collections] IgnoreCollectionIDUserType. With this setting, STEPER will not check the user type when processing Collections -Logging improvements around Source Group Email Domain Filtering and Excluded Source Groups -Improved processing reliability around Excluded Source Groups -Added DontProcessStudentsFoundinTeachersSourceGroup and DontProcessTeachersFoundinStudentsSourceGroup
06/05/23	1.0.87	-Improvement to Get-MSOLRecursive to ensure unique members are processed by ObjectID instead of email address
14/06/23	1.0.88	-Minor bugfix to ADOU processing and ensuring STEPER connects to the required services

04/07/23	1.0.89	-Resolved a bug where ExcludedStudentsSourceGroups and ExcludedTeachersSourceGroups names were not displayed correctly in the log file
12/07/23	1.0.90	-Resolve a bug in CheckSingleUserGroupMemberShips that added to the PS error log but did not affect processing
23/07/23	1.0.91	-Replaced CheckSingleUserGroupMemberShips with a new method of Group Rules processing – GetGroupRuleUsers – this reduces Group Rules processing time by around 90%. The number of potential users that match the Group Rules is now stated in the log -Some logging improvements, including when unmatched members are not to be removed from Filtered Unified / Security / Distribution groups – this is clearly stated in the log -Improved the performance of Get All User Information / Information Retrieval – in the addition to the Group Rules improvements, the SDS Grade/Year/Object Type are now retrieved in one operation, again leading to a performance improvement -Get All User Information / Information Retrieval now does not get Package Information by default, as this is the most time-consuming part of the operation. Use GetUserPackageInformation=True to get the package information -Improved the end-to-end checking of Group Rules processing, to ensure that any errors are recognised and tabulated -Improved the performance of using * for Students Source group or Teachers Source group. Large numbers of users will now be processed in a much shorter period of time
02/08/23	1.0.92	-Improvements and fixes to BuildGroupTable to ensure that empty fields in the Group Rule results aren't shown as "\$null"
08/09/23	1.0.93	-Resolved an issue in Filtered Group/Package/Policy processing that if a group rule is impossible (i.e. the group couldn't be found) an error message about foregroundcolor was incorrectly triggered -Group Rules Problem Group Table: Added IncludeDistributionGroups field -Minor formatting improvements -Filtered Group/Package/Policy group rules processing – Distributiongroups are now included by default -Changed the default email port from 25 to 587
20/09/23	1.0.94	-Improved some of the consistency of the log output -All Package/Policy processing statistics now based off user ObjectID instead of Email Address -Group Rules Problem Group Table: Added DisableNameorIDCheckingForGroupRules field -Package and Policy Processing: Users that weren't licensed, but then added to a licensing group earlier during STEPER processing would still be flagged as Not Licensed users. Package and Policy processing re-checks the status of unlicensed users and reports back the updated licensing status; however any request to process users that were unlicensed at the start of STEPER is honoured -Added [Misc]RefreshTeacherLicensingStatusAfterLicensingGroupChanges and RefreshStudentLicensingStatusAfterLicensingGroupChanges, both default to False. If turned on, users that have been added to the respective licensing group will have their Licensing status re-check. Note that using these functions will stop OnlyApplytoUnlicensedTeachers/ OnlyApplytoUnlicensedStudents from processing users that were unlicensed when STEPER started, but were then added to the licensing group -Change the Package and Policy Batch threshold default settings from 20 to 99999: TeacherTeamsPackageBatchProcessingThreshold, StudentTeamsPackageBatchProcessingThreshold, TeacherTeamsPolicyBatchProcessingThreshold, StudentTeamsPolicyBatchProcessingThreshold -Information retrieval now takes places after the Pause after licensing changes -Improved the processing speed and reliability of Teacher and Student Licensing Groups -Improved the processing speed and reliability of All Teacher and All Student Unified Groups -Improved the performance and clarity of some of the package and policy log outputs -Improved the detail in some of the console output -Improved NumericalListCreator ability to accept lists of numbers with spaces in them, e.g. 1 - 5
27/09/23	1.0.95	-Resolved an issue where the log and on-screen display for "Processing Accumulated Filtered [Security/Unified/Distribution] Group User Removals" showed the incorrect Collection ID or Settings reference (the users were actually processed correctly) -Improved resilience for adjusting TLS settings where necessary. If TLS1.3 or TLS1.2 is not detected, STEPER will attempt to set a TLS1.2 connection -Improve the reliability of Exchange V3 connections on systems where the older V2 components haven't been used
13/12/23	1.0.96	-Resolved an issue with group processing that would cosmetically return member groups in addition to users -Resolved an issue where Student source groups were sometimes processed recursively when this was not requested -Fixed a couple of typos in source group processing -Resolved an issue introduced in 1.0.91 where GetGroupRuleUsers would not process nested/recursive groups
14/12/23	1.0.97	-Reworked [SourceGroups] into [TeacherSourceGroups] and [StudentSourceGroups] with automatic major settings file upgrades: --RequiredGroups / RequiredRecursiveGroups / RequiredOrGroups / RequiredOrRecursiveGroups replaces AllTeachersSourceGroupName and EnableAllTeachersSourceGroupRecursion --NotRequiredGroups / NotRequiredRecursiveGroups replaces ExcludedTeachersGroupName and EnableExcludedTeachersGroupRecursion --LogAvailable Users replaces LogAvailableTeachers and LogavailableStudents --RequiredGroups / RequiredRecursiveGroups / RequiredOrGroups / RequiredOrRecursiveGroups replaces AllStudentsSourceGroupName and EnableAllStudentsSourceGroupRecursion --NotRequiredGroups / NotRequiredRecursiveGroups replaces ExcludedStudentsGroupName and EnableExcludedStudentsGroupRecursion -Added DomainFilter to TeacherSecurityGroupX/StudentSecurityGroupX, TeacherUnifiedGroupX/StudentUnifiedGroupX, TeacherDistributionGroupX/StudentDistributionGroupX and InformationRetrieval/GetAllUserPolicyInformation -As part of the major settings file upgrade, the previous settings file is renamed with a ".Pre118Upgrade.INI" suffix -CheckForUpdates now defaults to True



		-Warnings and Errors are now also shown in the console on completion -Improved the readability of Group Rules log output, and added the NestedGroupTypes field to BuildGroupTable
14/12/23	1.0.98	-Resolved an issue where the Exchange online information connection details weren't logged properly
09/06/24	1.0.99	-Resolved an issue where the setting DontProcessTeachersFoundinStudentSourceGroups wasn't processed properly during settings upgrades or during processing

## Feature Overview

### Source User Groups

- All Teachers Group (which may be from MS School Data Sync (SDS)) (you can specify the group name or object ID)
- All Students Group (which may be from SDS) (you can specify the group name or object ID)
- Filtering Students or Teachers in those groups by Username domain e.g. myschool.co.uk

### Applying Licensing

- Applying licensing to Students or Teachers by adding them to a licensing group as required

### Applying Teachers to the All Staff Team

- Applying Teachers and the Global Admin to the Staff Team Unified Group

### Applying Students to the All Students Team

- Applying Students to the Students Team Unified Group

### Setting the Students Department Field

- Use the SDS Grade field
- Or alternatively use the SDS Graduation field

### Applying Teachers or Students to different Security Groups

- Use a Powershell style filter to select teachers or students and add them to a group
- Additionally, filter students by licensing status, licensing group membership, via group membership rules or on-prem AD location/OU
- Up to 100 Teacher Groups can be configured and 100 Student Groups (although this can be increased)

### Applying Teachers or Students to different Unified Groups

- Use a Powershell style filter to select teachers or students and add them to a group
- Additionally, filter students by licensing status, licensing group membership, via group membership rules or on-prem AD location/OU
- By default, up to 100 Teacher Groups can be configured and 100 Student Groups (although this can be increased)

### Applying Teachers or Students to different Distribution Groups

- Use a Powershell style filter to select teachers or students and add them to a group
- Additionally, filter students by licensing status, licensing group membership, via group membership rules or on-prem AD location/OU
- Up to 100 Teacher Groups can be configured and 100 Student Groups (although this can be increased)

### Applying Teams Policies Packages

- Use a Powershell style filter to select teachers or students and apply a Teams Policy Package
- Additionally, filter students by licensing status, licensing group membership, via group membership rules or on-prem AD location/OU
- Up to 100 Teacher Packages can be configured and 100 Student Packages (although this can be increased)
- Configurable thresholds for batch or immediate package processing

### Applying Teams Policies

- Use a Powershell style filter to select teachers or students and apply up to 8 of the most common Teams policies
- Additionally, filter students by licensing status, licensing group membership, via group membership rules or on-prem AD location/OU
- Up to 100 Teacher policy sets can be configured and 100 Student policy sets (although this can be increased)
- Configurable thresholds for batch or immediate policy processing

### Export Information

- Export information for all users: Basic details, on-prem ID, OU, Policy and Package Information
- Export all user Department Information
- Get existing Policy Jobs



## Functionality Overview

STEPPER is designed to be deployed on a Windows 10 or Windows 2016 or newer server systems. Ideally, it should be run after each MIS system SDS sync. If you use an automatic or scheduled SDS import, you should use a scheduled task to run STEPPER after your MIS SDS sync. In this situation, the machine (or virtual machine) for STEPPER should be left permanently switched on.

### Simulation

By default, STEPPER operates in simulation only mode. No changes are made in this mode, however all items are processed and logged to show you the changes that would be made.

Use simulation mode on any new deployment or if you change the configuration to ensure that STEPPER operates as you expect.

### Profiles and Profile Names

In a large M365 environment, you may need to use more than one run of STEPPER, for example if different schools need different settings. STEPPER uses an INI file for its configuration, allowing you to build multiple configurations. Each configuration file can have a Profile Name included allowing you to easily differentiate different configurations in log files, event logs and email reports.

### Logging

STEPPER records all actions through:

- A detailed Log File (look for the “STEPPER Logs” directory)
- The Windows Event Log
- A CSV file for the All-user Policy and Package Information export
- The Powershell console (not available if you use a scheduled task)
- An email, which contains a ZIPPED copy of the log file, the settings used and the CSV output
- The Registry (HKLM:Software\Stone\M365\STEPPER)

### Management of Powershell

STEPPER will:

- Install all Powershell modules as needed
- Perform a monthly Powershell module update
- Automatically connect and disconnect only the needed Powershell connections



## General Dos and Don'ts

### Do

- ✓ Always use the Simulation feature on a new installation, or when you try some new settings or functionality
- ✓ Always run STEPER using an Administrative Powershell Window using Simulation Mode, so you can see the console output
- ✓ Always check the text file event log. It's a good idea to configure the email settings so the logs get emailed to you – this way you also build up a history of what you've done. The simulation mode event logs show exactly what user changes would be made, without actually making them
- ✓ Always read the manual for any functionality you want to use
- ✓ If you aren't sure of something, please contact support for help
- ✓ Before making large changes to your tenancy, consider using the Information Retrieval functions to list your users existing policies and packages

### Don't

- ✗ Don't close the STEPER console Window whilst its running unless STEPER is specifically giving you the opportunity. For example, with simulation mode turned off, by default there is a 20 second delay which prompts you to close STEPER if you don't want changes to be made
- ✗ Don't modify the STEPER script. It is completely unsupported to do so
- ✗ Don't store your STEPER configuration files somewhere where non-administrators could read them, as they will likely contain your M365 Global Administrator username and password
- ✗ Don't run multiple instances of STEPER on the same tenancy and on the same users at the same time
- ✗ Lastly, don't run multiple instances of STEPER on the same tenancy from different computers at the same time, even if the user actions are completely different



## Getting Started

STEPPER can be deployed in around 10 minutes for most functionality.

### Pre-Requisites

- Select an always on PC or virtual machine, that runs Windows 10 1809 or Server 2016 or newer. Windows Server 2019 or Windows Server 2022 is preferred
- Install .Net Framework 4.8 or a newer version that includes .NET Framework 4.x
- You will need Administrative rights to this PC or virtual machine
- Get a licensed M365 Global Administrator Account. It is recommended to use an account specifically for STEPPER, instead of a regular Global Administrator account that is used by one of the company users. Make sure the account has a complex password

### Basic Installation

- [Download the latest version of STEPPER](#) and extract the files into a local folder such as C:\\_STONE\STEPPER
- The STEPPER package includes a scheduled task XML file, which you may import into Task Scheduler to run STEPPER at 10am and 6pm every day. You must use C:\\_STONE\STEPPER as the folder for all of the extracted files for this task to work
- Edit the STEPPER.INI file for your settings
- Test STEPPER in simulation mode
- When comfortable that STEPPER will make the changes you want, turn off simulation mode

**TIP:** Don't enable or import the scheduled task until you have configured and tested your STEPPER deployment.

### Package Contents

Key Files	Purpose
STEPPER.PS1	Main STEPPER code – this is what you run
STEPPER.XML	Sample Scheduled Task for STEPPER to run from the C:\_STONE\STEPPER folder
Library\CreateAllStaffTeamUnifiedGroup.PS1	Sample Code to <a href="#">Create an All Staff Team</a>
Library\CreateDefaultPolicies.PS1	Sample Code to <a href="#">Create Teams Policies</a>
Library\CreateLicensingGroups.PS1	Sample Code to <a href="#">Create Licensing Groups</a>
Library\PreventStudentsFromCreatingTeams.PS1	Sample Code to <a href="#">Prevent Students from Creating Teams</a>
STEPPER.ini	Default settings file. This sample (which requires you to edit it) contains all of the possible settings available in STEPPER
STEPPER_DFE.INI	This sample settings file references just the functionality used in a Stone DFE deployment. Additional functionality removed or turned off. For example, licensing groups are used, but references to Teams Packages are removed
STEPPER_Stone_DFE.INI	As per STEPPER_DFE.INI, but uses the actual Policy names used in a Stone deployment completed under the DFE Program
STEPPER_TeamsPolicies.INI	This sample settings file contains functionality just to change policy assignments. Licensing groups are not used. All users' policies will be checked to ensure they have the correct assignments

## Azure AD, SDS (School Data Sync), and STEPER – The Process

If you use Azure AD or Azure AD Connect / DirSync to push new users into the cloud:

- Use an SDS sync with your MIS system to then push Student Enrolment information and class information into the cloud. This could be an ongoing, automatic sync. Your MIS system will likely need to have matching user email addresses for matching against the cloud user.
- By default, SDS will put these users into the All Students and All Teachers groups
- STEPER can then be used to license the user, and apply policies etc.
- Configure STEPER to run on a scheduled task daily, after the MIS System SDS sync should have completed. For example, if the SDS sync is run at 8am, run STEPER at 10am
- If the SDS sync is ongoing, consider running STEPER at 6pm every day

If you create users in the cloud manually:

- Run the MIS SDS Sync after any user or class changes
- By default, SDS will put these users into the All Students and All Teachers groups
- STEPER can then be used to license the user, and apply policies etc.
- When the sync is complete, run STEPER

If your MIS system creates users in the cloud automatically:

- Where possible, then configure the MIS system to perform an ongoing SDS sync
- By default, SDS will put these users into the All Students and All Teachers groups
- STEPER can then be used to license the user, and apply policies etc.
- If the SDS sync is ongoing, consider running STEPER at 6pm every day

All Scenarios:

- Check your SDS sync status at <https://sds.microsoft.com>
- Consider using Salamander (free version) for a one-off MIS to SDS sync, if not directly supported by your MIS system
- If you need an automatic ongoing sync between your MIS system and SDS, contact your Stone account manager



## Planning and Processing

### Preventing Students from Creating Teams

We recommend that all of your Global Admins, as well as Teachers, are the only ones able to create new Teams. This will prevent Students from creating their own Teams. This can be done by creating a security group that has exclusive permission to create Teams, and then adding the SDS “All Teachers” security group and the Global Admin as members.

STEPPER does not contain the functionality to do this directly, however it does support ensuring that the Global Admin account used during a run of STEPPER is added to this group, using the [\[GlobalAdminTeamsCreation\] functionality](#).

The Powershell below can be used as a one-off configuration change to:

- Create a group used to control who can create Teams
- Add All Teachers to that group
- Add the Global Admin to that group

We recommend you paste the text below into an Administrative Powershell ISE Window, and then edit the first 16 lines (up until “# Do not edit the lines below”) before running it.

**TIP:** As of STEPPER 1.0.52, the Skype for Business Powershell connector is no longer supported. It is no longer available for download from Microsoft and has been deprecated.



**Sample Code:** This code is also included in the STEPER package in  
Library\PreventStudentsFromCreatingTeams.PS1

```
# Sample Code used prevent Students from Creating Teams
# It adds the Global Admin and the SDS "All Teachers" groups as those that can create new Teams
# Andrew Sharrad 1/12/2020 v1.0

$UserName = "globaladmin@myschool.org.uk"
$Password = "Password99!"

$TeamCreate = "Stone-Allow-Team-Create"
# This group name is the default group name in STEPER.INI under [GlobalAdminTeamsCreation]

$GroupName = $TeamCreate
$AllowGroupCreation = "False"

$TeacherGroup = "All Teachers"
$StudentGroup = "All Students"

# Do not edit the lines below

$GAPassSec = ConvertTo-SecureString -AsPlainText $Password -Force
$GACreds = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $UserName,$GAPassSec

Install-Module AzureADPreview -Confirm:$False -Force -SkipPublisherCheck
Install-Module MSOnline -Confirm:$False -Force -SkipPublisherCheck
Connect-AzureAD -Credential $GACreds
Connect-MsolService -Credential $GACreds

New-MsolGroup -DisplayName $TeamCreate -Description "Group that allows the creation of Teams"
Add-MsolGroupMember -GroupObjectId (Get-MsolGroup -SearchString $TeamCreate).objectId -GroupMemberType Group -
GroupMemberObjectId (Get-MsolGroup -SearchString $TeacherGroup).objectId
Add-MsolGroupMember -GroupObjectId (Get-MsolGroup -SearchString $TeamCreate).objectId -GroupMemberType User -
GroupMemberObjectId (Get-MsolUser -UserPrincipalName $UserName).objectId

$settingsObjectID = (Get-AzureADDirectorySetting | Where-object -Property Displayname -Value "Group.Unified" -
EQ).id
if(!$settingsObjectID) {
    $template = Get-AzureADDirectorySettingTemplate | Where-object {$_.displayname -eq "group.unified"}
    $settingsCopy = $template.CreatedDirectorySetting()
    New-AzureADDirectorySetting -DirectorySetting $settingsCopy
    $settingsObjectID = (Get-AzureADDirectorySetting | Where-object -Property Displayname -Value "Group.Unified" -
EQ).id
}

$settingsCopy = Get-AzureADDirectorySetting -Id $settingsObjectID
$settingsCopy["EnableGroupCreation"] = $AllowGroupCreation

if($GroupName) {
    $settingsCopy["GroupCreationAllowedGroupId"] = (Get-AzureADGroup -SearchString $GroupName).objectId
} else {
    $settingsCopy["GroupCreationAllowedGroupId"] = $GroupName
}

Set-AzureADDirectorySetting -Id $settingsObjectID -DirectorySetting $settingsCopy
(Get-AzureADDirectorySetting -Id $settingsObjectID).values
```



## Identifying New Users and Using Licensing Groups

The settings in the STEPER.INI file are organised roughly into the same order that they are processed by STEPER.

For example, Students are licensed before they then have Teams Policies applied.

Most STEPER functions rely on finding Teachers and Students via Microsoft School Data Sync Groups called All Teachers and All Students. All of these users can be processed for changes, or, in most functions, you can specify that only new users are processed. New users are identified by checking All Teachers and All Students users against licensing groups.

When a new user – such as a Teacher – is processed for the first time, by default they are added to the group specified in the [\[LicensingGroups\]](#) section of the settings file. This prevents the user from being processed the next time, unless you specify that All users are processed.

This approach means that you should apply all of the changes you want to new students or teachers in one run of STEPER.

STEPPER can [create these licensing groups for you](#), or you can use this code below. These groups are ideal for assigning licenses to users, but you do not actually need to do this.

We recommend you paste the text below into an Administrative Powershell ISE Window, and then edit the first 12 lines (up until “# Do not edit the lines below”) before running it.

**Sample Code:** This code is also included in the STEPER package in Library\CreateLicensingGroups.PS1

```
# Sample Code used to Create Licensing Groups for Teachers and Students
# These group are used by STEPER to help identify new users. You can optionally assign licenses via the Azure AD
Console
# Andrew Sharrad 1/12/2020 v1.0

$UserName = "globaladmin@myschool.org.uk"
$Password = "Password99!"
# to create a team group for the students, change the line below to $CreateAllStudentsTeam = $true
$CreateAllStudentsTeam = $false

$A1Faculty = "Stone-Licensing-A1Faculty"
$A1Student = "Stone-Licensing-A1Student"
# Do not edit the lines below

$GAPassSec = ConvertTo-SecureString -AsPlainText $Password -Force
$GACreds = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $UserName,$GAPassSec

Install-Module MSOnline -Confirm:$False -Force -SkipPublisherCheck
Connect-MsolService -Credential $GACreds

New-MsolGroup -DisplayName $A1Faculty -Description "Group that assigns the A1 faculty licenses"
if ($CreateAllStudentsTeam -eq $true) {
    New-MsolGroup -DisplayName $A1Student -Description "Group that assigns the A1 student licenses"
}
```



## Creating the All Staff Team (also known as the [AllTeachersGroup])

You may wish for your school to include an All Staff Team.

The easiest way to do this is to create a Unified Group called “Staff Team”. You can then create the Team itself from inside the Teams Administrative console.

By default, [STEPER adds new Teachers found in the SDS “All Teachers” security group and adds them to the Staff Team Unified group.](#)

The code below will create a Unified Group called “Staff Team” and also enable organisation auditing. Alternatively, [STEPER can create these groups for you.](#)

We recommend you paste the text below into an Administrative Powershell ISE Window, and then edit the first 10 lines (up until “# Do not edit the lines below”) before running it.

**Sample Code:** This code is also included in the STEPER package in  
Library\CreateAllStaffTeamUnifiedGroup.PS1

```
# Sample Code used to Create a Unified Group, for use with the All Staff Team
# Andrew Sharrad 1/12/2020 v1.0

$UserName = "globaladmin@myschool.org.uk"
$Password = "Password99!"

$EnableAuditing = $true

$StaffTeamName = "Staff Team"
# This group name is the default group name in STEPER.INI under [AllTeachersGroup]
# Do not edit the lines below

$GAPassSec = ConvertTo-SecureString -AsPlainText $Password -Force
$GACreds = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $UserName,$GAPassSec

if ($PSVersionTable.PSVersion.Build -le 17134) {
    [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
}
Install-PackageProvider -Name NuGet
Install-Module MSOnline -Confirm:$False -Force -SkipPublisherCheck
Install-Module ExchangeOnlineManagement -Confirm:$False -Force -SkipPublisherCheck

Connect-MsolService -Credential $GACreds
Connect-ExchangeOnline -Credential $GACreds -ShowBanner:$false -ShowProgress:$false -Verbose:$false

if ($EnableAuditing -eq $true) {
    Enable-OrganizationCustomization
    Set-AdminAuditLogConfig -UnifiedAuditLogIngestionEnabled $true
}

New-unifiedGroup -DisplayName "$StaffTeamName" -Alias "StaffTeam" -accessstype Private -
AutoSubscribeNewMembers:$false -Notes "Microsoft Team for all staff collaboration"
```



## Creating Teams Policies

We recommend that you create your own Teams Policies and apply these policies to your users, rather than modifying the default policies.

We also recommend that you use individual Teams Policies as opposed to Teams Packages.

You can create Teams Policies and then use STEPER to assign policies to users.

- If you are running STEPER for the first time, make sure the policy names that you use match those in the STEPER.INI file
- You can use the code to create some DFE approved (June 2020) policies. The policy names in the code match those in the default STEPER.INI file

Policy	Default Policy Names in STEPER.INI	Settings Section
Teachers	Stone_EduTeacher	[TeacherTeamsPolicies1]
Primary School Students	Stone_EduPrimary	[StudentTeamsPolicies1]
Secondary School Students	Stone_EduSecondary	[StudentTeamsPolicies2]
Sixth Form Students	Stone_EduSixth	[StudentTeamsPolicies3]

- If you change the settings in a policy, you do not need to re-apply the policy to users
- If, however, you decide to use a different policy after having already deployed one using STEPER, you will need to change the STEPER.INI file to include the new policy name, and also set `ApplyTeamsPoliciesToAllTeachers=True` / `ApplyTeamsPoliciesToAllStudents=True` (as appropriate) to process all users, not just new users, for one run only

**TIP 1:** The sixth form student policies vary from the secondary student policies only in the allowing of pinning of Apps

We recommend you paste the text below into an Administrative Powershell ISE Window, and then edit the first 24 lines (up until “# Do not edit the lines below”) before running it.



**Sample Code:** This code is also included in the STEPER package in  
Library\CreateDefaultPolicies.PS1

```
# Sample Code used to create Teams Policies for Schools
# Andrew Sharrad 28/08/2021 v1.01

$UserName = "globaladmin@myschool.org.uk"
$Password = "Password99!"

$OnMicrosoftCOMPREFIX="myschool"
#Example onMicrosoftCOMPREFIX=ourtenancyname (without the onmicrosoft.com suffix)

$CreatePrimaryPolicy = $true
$PrimaryPolicyName= "Stone_EduPrimary"

$CreateSecondaryPolicy = $true
$SecondaryPolicyName= "Stone_EduSecondary"

$CreateSixthFormPolicy = $true
$SixthFormPolicyName= "Stone_EduSixth"

$CreateTeacherPolicy = $true
$TeacherPolicyName= "Stone_EduTeacher"

# Do not edit the lines below

$GAPassSec = ConvertTo-SecureString -AsPlainText $Password -Force
$GACreds = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $UserName,$GAPassSec

# Connected to MSTEams
Install-Module MicrosoftTeams -SkipPublisherCheck -AllowClobber
Connect-MicrosoftTeams -Credential $GACreds

if ($CreatePrimaryPolicy -eq $true) {
    $PolicyID = $PrimaryPolicyName
    Write-Host "Creating Primary School Policies"

    New-CsTeamsMessagingPolicy -Identity $PolicyID -AllowOwnerDeleteMessage $true -AllowUserDeleteMessage $false -
    AllowUserEditMessage $true -
    ReadReceiptsEnabledType Everyone -AllowUserChat $false -AllowGiphy $false -GiphyRatingType Strict -
    AllowMemes $false -AllowStickers $true -
    AllowURIPreviews $true -AllowUserTranslation $true -AllowImmersiveReader $true -AllowPriorityMessages
    $false -AudioMessageEnabledType Disabled -
    AllowRemoveUser $false -ChannelsInChatListEnabledType DisabledUserOverride -Description "Messaging Policy
    for Primary School Children"

    New-CsTeamsMeetingPolicy -Identity $PolicyID -AllowMeetNow $false -AllowOutlookAddIn $false -
    AllowChannelMeetingsScheduling $false -AllowPrivateMeetingsScheduling $false -
    AllowTranscription $true -AllowCloudRecording $false -AllowIPVideo $true -MediaBitRateKb 50000 -
    ScreenSharingMode SingleApplication -AllowParticipantGiveRequestControl $false -
    AllowExternalParticipantGiveRequestControl $false -AllowPowerPointSharing $true -
    AllowWhiteboard $true -AllowSharedNotes $false -
    AllowAnonymousUsersToStartMeeting $false -AutoAdmittedUsers EveryoneInCompany -
    AllowPSTNUsersToBypassLobby $false -AllowPrivateMeetNow $false -
    LiveCaptionsEnabledType DisabledUserOverride -MeetingChatEnabledType Enabled -Description "Meeting Policy
    for Primary School Children"

    New-CsTeamsAppSetupPolicy -Identity $PolicyID -AllowSideLoading $false -AllowUserPinning $false -Description
    "AppSetup Policy for Primary School Children"

    New-CsTeamsCallingPolicy -Identity $PolicyID -AllowPrivateCalling $false -AllowCallForwardingToUser $false -
    AllowCallForwardingToPhone $false -AllowVoicemail AlwaysDisabled -
    AllowCallGroups $false -AllowDelegation $false -PreventTollBypass $false -BusyOnBusyEnabledType Disabled -
    AllowWebPSTNCalling $false -
    Description "Calling Policy for Primary School Children"

    New-CsTeamsMeetingBroadcastPolicy -Identity $PolicyID -AllowBroadcastScheduling $false -
    AllowBroadcastTranscription $true -BroadcastAttendeeVisibilityMode EveryoneInCompany -
    BroadcastRecordingMode UserOverride -Description "Broadcast meeting policy for Primary School Children"
```



```
New-CsTeamsChannelsPolicy -Identity $PolicyID -AllowPrivateChannelCreation $false -Description "Teams Policy
for Primary School Children"
write-Host "Primary School policies created"
}

if ($CreateSecondaryPolicy -eq $true) {
    $PolicyID = $SecondaryPolicyName
    write-Host "Creating Secondary School Policies"

    New-CsTeamsMessagingPolicy -Identity $PolicyID -AllowOwnerDeleteMessage $true -AllowUserDeleteMessage $false -
AllowUserEditMessage $true
    -ReadReceiptsEnabledType Everyone -AllowUserChat $false -AllowGiphy $false -GiphyRatingType Strict -
AllowMemes $false -AllowStickers $true
    -AllowUrlPreviews $true -AllowUserTranslation $true -AllowImmersiveReader $true -AllowPriorityMessages
    $false -AudioMessageEnabledType Disabled
    -AllowRemoveUser $false -ChannelsInChatListEnabledType DisabledUserOverride -Description "Messaging Policy
for Secondary School Children"

    New-CsTeamsMeetingPolicy -Identity $PolicyID -AllowMeetNow $false -AllowOutlookAddIn $false -
AllowChannelMeetingsScheduling $false -AllowPrivateMeetingsScheduling $false
    -AllowTranscription $true -AllowCloudRecording $false -AllowIPVideo $true -MediaBitRateKb 50000
    -ScreenSharingMode SingleApplication -AllowParticipantGiveRequestControl $false -
AllowExternalParticipantGiveRequestControl $false -AllowPowerPointSharing $true
    -AllowWhiteboard $true -AllowSharedNotes $false
    -AllowAnonymousUsersToStartMeeting $false -AutoAdmittedUsers EveryoneInCompany -
AllowPSTNUsersToBypassLobby $false -AllowPrivateMeetNow $false
    -LiveCaptionsEnabledType DisabledUserOverride -MeetingChatEnabledType Enabled -Description "Meeting Policy
for Primary School Children"

    New-CsTeamsAppSetupPolicy -Identity $PolicyID -AllowSideLoading $false -AllowUserPinning $false -Description
"AppSetup Policy for Secondary School Children"

    New-CsTeamsCallingPolicy -Identity $PolicyID -AllowPrivateCalling $false -AllowCallForwardingToUser $false -
AllowCallForwardingToPhone $false -AllowVoicemail AlwaysDisabled
    -AllowCallGroups $false -AllowDelegation $false -PreventTollBypass $false -BusyOnBusyEnabledType Disabled
    -AllowWebPSTNCalling $false
    -Description "Calling Policy for Secondary School Children"

    New-CsTeamsMeetingBroadcastPolicy -Identity $PolicyID -AllowBroadcastsScheduling $false -
AllowBroadcastTranscription $true -BroadcastAttendeeVisibilityMode EveryoneInCompany
    -BroadcastRecordingMode UserOverride -Description "Broadcast meeting policy for Secondary School Children"

    New-CsTeamsChannelsPolicy -Identity $PolicyID -AllowPrivateChannelCreation $false -Description "Teams Policy
for Secondary School Children"

    write-Host "Secondary School policies created"
}

if ($CreateSixthFormPolicy -eq $true) {
    $PolicyID = $SixthFormPolicyName
    write-Host "Creating Sixth Form Policies"

    New-CsTeamsMessagingPolicy -Identity $PolicyID -AllowOwnerDeleteMessage $true -AllowUserDeleteMessage $true -
AllowUserEditMessage $true
    -ReadReceiptsEnabledType Everyone -AllowUserChat $false -AllowGiphy $false -GiphyRatingType Strict -
AllowMemes $false -AllowStickers $true
    -AllowUrlPreviews $true -AllowUserTranslation $true -AllowImmersiveReader $true -AllowPriorityMessages
    $false -AudioMessageEnabledType Disabled
    -AllowRemoveUser $false -ChannelsInChatListEnabledType DisabledUserOverride -Description "Messaging Policy
for Sixth Form Students"

    New-CsTeamsMeetingPolicy -Identity $PolicyID -AllowMeetNow $false -AllowOutlookAddIn $false -
AllowChannelMeetingsScheduling $false -AllowPrivateMeetingsScheduling $false
    -AllowTranscription $true -AllowCloudRecording $false -AllowIPVideo $true -MediaBitRateKb 50000
    -ScreenSharingMode SingleApplication -AllowParticipantGiveRequestControl $true -
AllowExternalParticipantGiveRequestControl $false -AllowPowerPointSharing $true
    -AllowWhiteboard $true -AllowSharedNotes $false
    -AllowAnonymousUsersToStartMeeting $false -AutoAdmittedUsers EveryoneInCompany -
AllowPSTNUsersToBypassLobby $false -AllowPrivateMeetNow $false
    -LiveCaptionsEnabledType DisabledUserOverride -MeetingChatEnabledType Enabled -Description "Meeting Policy
for Sixth Form Students"
```



```
New-CsTeamsAppSetupPolicy -Identity $PolicyID -AllowSideLoading $false -AllowUserPinning $true -Description
"AppSetup Policy for Sixth Form Students"

New-CsTeamsCallingPolicy -Identity $PolicyID -AllowPrivateCalling $false -AllowCallForwardingToUser $false -
AllowCallForwardingToPhone $false -AllowVoicemail AlwaysDisabled -
-AllowCallGroups $false -AllowDelegation $false -PreventTollBypass $false -BusyOnBusyEnabledType Disabled
-AllowWebPSTNCalling $false -
-Description "Calling Policy for Sixth Form Students"

New-CsTeamsMeetingBroadcastPolicy -Identity $PolicyID -AllowBroadcastsScheduling $false -
AllowBroadcastTranscription $true -BroadcastAttendeeVisibilityMode EveryoneInCompany -
-BroadcastRecordingMode UserOverride -Description "Broadcast meeting policy for Sixth Form Students"

New-CsTeamsChannelsPolicy -Identity $PolicyID -AllowPrivateChannelCreation $false -Description "Teams Policy
for Sixth Form Students"

Write-Host "Sixth Form policies created"
}

if ($CreateTeacherPolicy -eq $true) {
    $PolicyID = $TeacherPolicyName
    Write-Host "Creating Teacher Policies"

    New-CsTeamsMessagingPolicy -Identity $PolicyID -AllowOwnerDeleteMessage $true -AllowUserDeleteMessage $true -
AllowUserEditMessage $true -
-ReadReceiptsEnabledType Everyone -AllowUserChat $true -AllowGiphy $true -GiphyRatingType Strict -
AllowMemes $false -AllowStickers $true -
-AllowUrlPreviews $true -AllowUserTranslation $true -AllowImmersiveReader $true -AllowPriorityMessages
$true -AudioMessageEnabledType ChatsAndChannels -
-allowRemoveUser $true -ChannelsInChatListEnabledType DisabledUserOverride -Description "Messaging Policy
for Teachers"

    New-CsTeamsMeetingPolicy -Identity $PolicyID -AllowMeetNow $true -AllowOutlookAddIn $true -
AllowChannelMeetingsScheduling $true -AllowPrivateMeetingsScheduling $true -
-AllowTranscription $true -AllowCloudRecording $true -AllowIPVideo $true -MediaBitRateKb 50000 -
-ScreenSharingMode EntireScreen -AllowParticipantGiveRequestControl $true -
AllowExternalParticipantGiveRequestControl $true -AllowPowerPointSharing $true -
-AllowWhiteboard $true -AllowSharedNotes $true -
-AllowAnonymousUsersToStartMeeting $false -AutoAdmittedUsers EveryoneInCompany -
AllowPSTNUsersToBypassLobby $false -AllowPrivateMeetNow $false -
-LiveCaptionsEnabledType DisabledUserOverride -MeetingChatEnabledType Enabled -Description "Meeting Policy
for Teachers"

    #Set-CsTeamsMeetingPolicy -Identity $PolicyID -AllowMeetNow $true -AllowOutlookAddIn $true -
AllowChannelMeetingsScheduling $true -AllowPrivateMeetingsScheduling $true -
# -AllowTranscription $true -AllowCloudRecording $true -AllowIPVideo $true -MediaBitRateKb 50000 -
# -ScreenSharingMode EntireScreen -AllowParticipantGiveRequestControl $true -
AllowExternalParticipantGiveRequestControl $true -AllowPowerPointSharing $true -
# -AllowWhiteboard $true -AllowSharedNotes $true -
# -AllowAnonymousUsersToStartMeeting $false -AutoAdmittedUsers EveryoneInCompany -
AllowPSTNUsersToBypassLobby $false -AllowPrivateMeetNow $false -
# -LiveCaptionsEnabledType DisabledUserOverride -MeetingChatEnabledType Enabled -Description "Meeting
Policy for Teachers"

    New-CsTeamsAppSetupPolicy -Identity $PolicyID -AllowSideLoading $false -AllowUserPinning $true -Description
"AppSetup Policy for Teachers"

    New-CsTeamsCallingPolicy -Identity $PolicyID -AllowPrivateCalling $false -AllowCallForwardingToUser $false -
AllowCallForwardingToPhone $false -AllowVoicemail AlwaysDisabled -
-AllowCallGroups $false -AllowDelegation $false -PreventTollBypass $false -BusyOnBusyEnabledType Disabled
-AllowWebPSTNCalling $true -
-Description "Calling Policy for Teachers"

    New-CsTeamsMeetingBroadcastPolicy -Identity $PolicyID -AllowBroadcastScheduling $false -
AllowBroadcastTranscription $true -BroadcastAttendeeVisibilityMode EveryoneInCompany -
-BroadcastRecordingMode AlwaysEnabled -Description "Broadcast meeting policy for Teachers"

    New-CsTeamsChannelsPolicy -Identity $PolicyID -AllowPrivateChannelCreation $true -Description "Teams Policy
for Teachers"

    Write-Host "Teacher policies created"
}
```



## Security Group Nesting, or Recursion – Implied versus Explicit Memberships

With Microsoft Security groups, members can be directly assigned to a group. For example, UserA can be added to Group1. This direct assignment means that the user is explicitly a member of Group1.

Alternatively, UserB can be directly assigned as a member of SubGroupX. SubGroupX can then be directly assigned as a member of Group1. This means that while UserB is not directly a member of Group1, their membership is implied.

STEPER supports Implied memberships where requested. The process of finding implied memberships via nested groups is known as **Recursion**.

Function or Group Name/Type	STEPER Supports Group Recursion (Implied Membership)?
[TeacherSourceGroups] / [StudentSourceGroups] - Source Groups (All Teachers, All Students)	<p>STEPER supports group recursion for these security groups, to find implied members.</p> <p>This is useful if you are not using SDS and for example, if don't have a single security group for all of your teaching staff. In this situation, you can create a new security group called "All Teachers" and add your existing staff security groups to it, for example "Staff - Lower School" and "Staff - Upper School". STEPER will find the members of "Staff - Lower School" and "Staff - Upper School" for processing.</p> <p><a href="#">Group Filters</a>: Checking if the user matches the "RequiredGroups=", "RequiredOrGroups=" or "NotRequiredGroups" setting - a recursive search is not performed</p> <p>Checking if the user matches the "RequiredRecursiveGroups=", "RequiredOrRecursiveGroups=" or "NotRequiredRecursiveGroups" setting - a recursive search is performed</p>
[SourceGroups] - Excluded Users (ExcludedTeachersGroupName, ExcludedStudentsGroupName)	<p>Yes, however it is disabled by default.</p> <p>STEPER can support group recursion for these security groups, to find implied members.</p> <p>You can <a href="#">enable the recursive search</a> if required, by setting EnableExcludedTeachersGroupRecursion=True and/or EnableExcludedStudentsGroupRecursion=True</p>
[GlobalAdminTeamsCreation]	No. STEPER will always add Global Admins as Explicit members. It will not check for existing membership via a recursive search.
[Licensing Groups]	No. STEPER will always look for new users by seeing if they are explicit members, and directly adding new users as a member of the licensing group if configured to do so.

<p>[AllTeachersGroup] / [AllStudentsGroup]</p>	<p>No, as these are Unified Groups for Teams, not Security Groups</p>
<p>[TeacherSecurityGroupX] / [StudentSecurityGroupX]</p>	<p>Checking to see if the user is a member of the Teacher or Student security group already:</p> <ul style="list-style-type: none"> <li>- A recursive search is not performed. Users that are not already directly, explicitly a member are candidates for adding to the group</li> </ul> <p><a href="#">Group Filters</a>: Checking if the user matches the “RequiredGroups=”, “RequiredOrGroups=” or “NotRequiredGroups” setting - a recursive search is not performed</p> <p>Checking if the user matches the “RequiredRecursiveGroups=”, “RequiredOrRecursiveGroups=” or “NotRequiredRecursiveGroups” setting - a recursive search is performed</p>
<p>[TeacherUnifiedGroupX] / [StudentUnifiedGroupX]</p>	<p>Checking to see if the user is a member of the Unified group already:</p> <ul style="list-style-type: none"> <li>- A recursive search is not performed. Users that are not already directly, explicitly a member are candidates for adding to the group</li> </ul> <p><a href="#">Group Filters</a>: Checking if the user matches the “RequiredGroups=”, “RequiredOrGroups=” or “NotRequiredGroups” setting - a recursive search is not performed</p> <p>Checking if the user matches the “RequiredRecursiveGroups=”, “RequiredOrRecursiveGroups=” or “NotRequiredRecursiveGroups” setting - a recursive search is performed</p>
<p>[TeacherDistributionGroupX] / [StudentDistributionGroupX]</p>	<p>Checking to see if the user is a member of the Distribution group already:</p> <ul style="list-style-type: none"> <li>- A recursive search is not performed. Users that are not already directly, explicitly a member are candidates for adding to the group</li> </ul> <p><a href="#">Group Filters</a>: Checking if the user matches the “RequiredGroups=”, “RequiredOrGroups=” or “NotRequiredGroups” setting - a recursive search is not performed</p> <p>Checking if the user matches the “RequiredRecursiveGroups=”, “RequiredOrRecursiveGroups=” or “NotRequiredRecursiveGroups” setting - a recursive search is performed</p>



<p>[TeacherTeamsPackagesX] / [StudentTeamsPackagesX]</p>	<p><a href="#">Group Filters</a>: Checking if the user matches the “RequiredGroups=”, “RequiredOrGroups=” or “NotRequiredGroups” setting: a recursive search is not performed</p> <p>Checking if the user matches the “RequiredRecursiveGroups=”, “RequiredOrRecursiveGroups=” or “NotRequiredRecursiveGroups” setting:</p> <ul style="list-style-type: none"><li>- A recursive search is performed</li></ul>
<p>[TeacherTeamsPoliciesX] / [StudentTeamsPoliciesX]</p>	<p><a href="#">Group Filters</a>: Checking if the user matches the “RequiredGroups=”, “RequiredOrGroups=” or “NotRequiredGroups” setting: a recursive search is not performed</p> <p>Checking if the user matches the “RequiredRecursiveGroups=”, “RequiredOrRecursiveGroups=” or “NotRequiredRecursiveGroups” setting:</p> <ul style="list-style-type: none"><li>- A recursive search is performed</li></ul>

## Testing STEPER using the Powershell Command Line

It's recommended that before you deploy the scheduled task, you experiment with STEPER using the Powershell command line.

### Steps

- Open an Administrative Powershell prompt
- Change to the directory containing the STEPER files, for example, CD \\_STONE\STEPER
- Use the DIR command to make sure you can see the STEPER.PS1 file, and sample STEPER INI configuration files
- After making changes to the configuration file, try running STEPER. For your first run, ensure that in your INI file, MakeChanges is set to **false**

```
PS C:\_stone\steper> dir

Directory: C:\_stone\steper

Mode                LastWriteTime         Length Name
----                -
-a----             20/07/2020   23:58           20608 steper.ini
-a----             21/07/2020   17:08          420178 STEPER.ps1

PS C:\_stone\steper> .\steper.ps1 -|
  SettingsFile
  MakeChanges
  UpdateModules
  Verbose
```

## Using Settings Files

STEPER is configured using text based INI files.

STEPER will look for STEPER.INI in the same folder that STEPER itself exists, and then the current working directory.

Alternatively, you can specify an alternative STEPER settings file to be used, with the command line option -SettingsFile

```
PS C:\_stone\STEPER> .\STEPER.ps1 -SettingsFile C:\_Stone\STEPER\Alternative.INI
```



## Console Output

STEPER will show its progress through the console output as well as the log file. In simulation model, use the console output to help troubleshoot issues before you deploy the scheduled task and turn off simulation mode.

```
C:\Users\Andrew\Documents\Schools\M365\OngoingSync\ [redacted] \STEPER.ps1 STEPER v1.0.75 (C:\1000\100) Profile: (Untitled - [redacted]) (Elevated)
Stone Teams Policy Enforcer (STEPER): v1.0.75 (C:\1000\100) Andrew Sharrod 27/09/22

Please see the license agreement in the accompanying manual and the notice regarding receiving updates
No Warranty, Implied or Otherwise is supplied with this software

-----
Loading Settings from STEPER.INI
Detailed Log Information being saved in C:\Users\Andrew\Documents\Schools\M365\OngoingSync\ [redacted] \STEPER Logs\STEPER v1.0.75 AS-WORKPC (Untitled - [redacted]) 27-09-2022 225141 Log File.txt
-----

Simulation mode enabled - changes will NOT be made to your users

-----
Processing settings
Using local AzureAD Attribute Filtering by default
Installing and/or updating the required Powershell Modules as needed
Connecting to MSOnline
Global Admin Teams Creation Group Membership
PowerShell [redacted] is already a member of the Teams Creation Security Group Stone-Allow-Team-Create
Getting the Source List of Teachers
Simulation: 139 Teachers from the Source Group [redacted] Staff available For processing
Getting the Source List of Students
Simulation: 1145 Students from the Source Group [redacted] Student available For processing
Connecting to Exchange Online V3
Connecting to AzureAD
Connecting to Microsoft Teams
Adding Teachers to the Licensing Group
There are no Teachers to add to the Licensing Group Stone-licensing-AllFaculty
Adding Students to the Licensing Group
There are no Students to add to the Licensing Group Stone-licensing-AllStudent
Adding Teachers to the All Teachers Unified Group
There are no Teachers to add to the All Teachers Unified Group All Staff
Information Retrieval - Get User Unique Department Information
Get User Unique Department Information: Found 1535 user records with 5 Unique Department Fields (not Filtered)
There are no Filtered Security Groups to Apply
There are no Filtered Unified Groups to Apply
There are no Filtered Distribution Groups to Apply
There are no Teams Packages to Apply
Applying Teams Policies
Simulation: Processing Policy Request 1 of 18: CollectionID: 100 [TeacherTeamsPolicies1] Policy Type: TeamsMeetingBroadcastPolicy Policy Name: Education_Teacher Desc: Standard Stone DFE Teacher Policies
Simulation: Processing Policy Request 2 of 18: CollectionID: 100 [TeacherTeamsPolicies1] Policy Type: TeamsMeetingPolicy Policy Name: Education_Teacher Desc: Standard Stone DFE Teacher Policies
Simulation: Processing Policy Request 3 of 18: CollectionID: 100 [TeacherTeamsPolicies1] Policy Type: TeamsMessagingPolicy Policy Name: Education_Teacher Desc: Standard Stone DFE Teacher Policies
Simulation: Processing Policy Request 4 of 18: CollectionID: 100 [TeacherTeamsPolicies1] Policy Type: TeamsAppSetupPolicy Policy Name: Education_Teacher Desc: Standard Stone DFE Teacher Policies
Simulation: Processing Policy Request 5 of 18: CollectionID: 100 [TeacherTeamsPolicies1] Policy Type: TeamsCallingPolicy Policy Name: Education_Teacher Desc: Standard Stone DFE Teacher Policies
Simulation: Processing Policy Request 6 of 18: CollectionID: 100 [TeacherTeamsPolicies1] Policy Type: TeamsChannelsPolicy Policy Name: Education_Teacher Desc: Standard Stone DFE Teacher Policies
Simulation: Processing Policy Request 7 of 18: CollectionID: 100 [StudentTeamsPolicies2] Policy Type: TeamsMeetingBroadcastPolicy Policy Name: Education_SecondaryStudent Desc: Standard Stone DFE Secondary Policies
Simulation: Building Group Rules Table
Simulation: Group Rules OK: True Problem Groups: 0 Required Groups: 0 Required OR Groups: 5 Not Required Groups: 0
Simulation: Processing Policy Request 8 of 18: CollectionID: 100 [StudentTeamsPolicies2] Policy Type: TeamsMeetingPolicy Policy Name: Education_SecondaryStudent Desc: Standard Stone DFE Secondary Policies
Simulation: Reusing Previous Group Rules Table
Simulation: Processing Policy Request 9 of 18: CollectionID: 100 [StudentTeamsPolicies2] Policy Type: TeamsMessagingPolicy Policy Name: Education_SecondaryStudent Desc: Standard Stone DFE Secondary Policies
Simulation: Reusing Previous Group Rules Table
Simulation: Processing Policy Request 10 of 18: CollectionID: 100 [StudentTeamsPolicies2] Policy Type: TeamsAppSetupPolicy Policy Name: Education_SecondaryStudent Desc: Standard Stone DFE Secondary Policies
Simulation: Reusing Previous Group Rules Table
Simulation: Processing Policy Request 11 of 18: CollectionID: 100 [StudentTeamsPolicies2] Policy Type: TeamsCallingPolicy Policy Name: Education_SecondaryStudent Desc: Standard Stone DFE Secondary Policies
Simulation: Reusing Previous Group Rules Table
Simulation: Processing Policy Request 12 of 18: CollectionID: 100 [StudentTeamsPolicies2] Policy Name: Education_SecondaryStudent Desc: Standard Stone DFE Secondary Policies
Simulation: Reusing Previous Group Rules Table
Simulation: Processing Policy Request 13 of 18: CollectionID: 100 [StudentTeamsPolicies3] Policy Type: TeamsMeetingBroadcastPolicy Policy Name: Education_HigherEducationStudent Desc: Standard Stone DFE Sixth Form Policies
Simulation: Building Group Rules Table
Simulation: Group Rules OK: True Problem Groups: 0 Required Groups: 0 Required OR Groups: 2 Not Required Groups: 0
Simulation: Processing Policy Request 14 of 18: CollectionID: 100 [StudentTeamsPolicies3] Policy Type: TeamsMeetingPolicy Policy Name: Education_HigherEducationStudent Desc: Standard Stone DFE Sixth Form Policies
Simulation: Reusing Previous Group Rules Table
Simulation: Processing Policy Request 15 of 18: CollectionID: 100 [StudentTeamsPolicies3] Policy Type: TeamsMessagingPolicy Policy Name: Education_HigherEducationStudent Desc: Standard Stone DFE Sixth Form Policies
Simulation: Reusing Previous Group Rules Table
Simulation: Processing Policy Request 16 of 18: CollectionID: 100 [StudentTeamsPolicies3] Policy Type: TeamsAppSetupPolicy Policy Name: Education_HigherEducationStudent Desc: Standard Stone DFE Sixth Form Policies
Simulation: Reusing Previous Group Rules Table
Simulation: Processing Policy Request 17 of 18: CollectionID: 100 [StudentTeamsPolicies3] Policy Type: TeamsCallingPolicy Policy Name: Education_HigherEducationStudent Desc: Standard Stone DFE Sixth Form Policies
Simulation: Reusing Previous Group Rules Table
Simulation: Processing Policy Request 18 of 18: CollectionID: 100 [StudentTeamsPolicies3] Policy Type: TeamsChannelsPolicy Policy Name: Education_HigherEducationStudent Desc: Standard Stone DFE Sixth Form Policies
Setting Policy Statistics (this may take a few minutes)
Simulation: 18 Policy Requests Processed with 0 Teacher Policy Changes and 0 Student Policy Changes
Information Retrieval - Get Policy Jobs
Get Policy Jobs: Could not find any CS Batch Policy Assignment Operation Jobs
User Remediation - There are no Users to Remediate by Removal from Licensing Groups
User Re-Processing - There are no User Re-Processing / Apply to All Teachers or Students on Next Run Causes

-----
Closing Connections
Tidy up Log Folder C:\Users\Andrew\Documents\Schools\M365\OngoingSync\ [redacted] \STEPER Logs
Cleaning up log files that are 60 days old and older - C:\Users\Andrew\Documents\Schools\M365\OngoingSync\ [redacted] \STEPER Logs\STEPER v* Log File.txt
Cleaning up log files that are 60 days old and older - C:\Users\Andrew\Documents\Schools\M365\OngoingSync\ [redacted] \STEPER Logs\STEPER v* Log Bundle.zip
Processing ended with no problems detected - review the Application Event Log and Logfile C:\Users\Andrew\Documents\Schools\M365\OngoingSync\ [redacted] \STEPER Logs\STEPER v1.0.75 AS-WORKPC (Untitled - [redacted]) 27-09-2022 225141 Log File.txt
Processing time: 0 mins 57 secs
Sending results email to [redacted]
```





## Elevation

STEPPER requires Administrative rights to run.

- If you use a scheduled task, make sure you set an Administrative account to run the task
- If you try and run STEPPER from Windows File explorer, it will automatically spawn an elevated second copy of STEPPER
- If you run STEPPER from a non-elevated Powershell Window, with no command line arguments, it will try and spawn an elevated second copy of STEPPER

If you run STEPPER from a non-elevated Powershell Window, with command line arguments, STEPPER will fail to run, as it will be unable to pass the command line arguments to the spawned copy.

## Command Line Options

### -SettingsFile

Specify the name of the configuration file that you want to use. By default, STEPER will look for STEPER.INI in the same folder that STEPER itself exists, and then the current working directory. If you want to use an alternative file, using -SettingsFile <settings.ini> on the command line

### -MakeChanges

By default, STEPER will run in Simulation only mode. Specifying -MakeChanges on the command line, or [General]MakeChanges=True inside the settings file will make run outside of simulation only mode. In this situation, changes will be made to your users.

```
[General]
;Note that the STEPERProfileName is just for STEPER - it is not related to your SDS Profile Name
STEPPERProfileName=
;By default, STEPER will operate in Simulation mode - it will just list the changes that would be
MakeChanges=False
```

### -UpdateModules

By default, STEPER will install the needed Powershell modules, and then update them once a month. To force an immediate check for updates, use -UpdatedModules or [PowerShell]ForceUpdateModules=True

### -Verbose

Use -Verbose on the command line or [Misc]Verbose=True to make STEPER display more information on the console window during processing.

### -ApplytoAll / -ApplytoAllTeachers / -ApplytoAllStudents

If specified, this global setting makes STEPER apply changes to all users (or all teachers/all students), instead of just those that weren't in the licensing groups. This affects setting the Student Department, Filtered Security Groups, Filtered Unified Groups, Filtered Distribution Groups, Teams Packages and Teams Policies, and overrides the settings in those functions.

You can also specify this in the configuration file under the [Override] section. If set to false in the configuration file, the settings in the individual functions are honoured.

```
[Override]
GlobalOnFailureApplytoAllUsersOnNextRun=True
;The settings below only take effect if set to True
ApplytoAllTeachers=False
ApplytoAllStudents=False
```

See [here](#) for more information.

## Configuring STEPER

It is recommended to use [Notepad++](#) to edit the STEPER.INI file to make your settings changes.

### Editing the Settings File

The settings file is organized into sections, for example [General] and [Credentials].

- Do not make any changes to the section heading names
- Ensure that you don't accidentally put extra spaces before or after your setting
- Many settings are switched on using the answer True or False. Pay attention to ensure this is spelled correctly, or alternatively use 1 for True and 0 for False
- Lines that start with a semi-colon are switched off

### Sample Settings Files

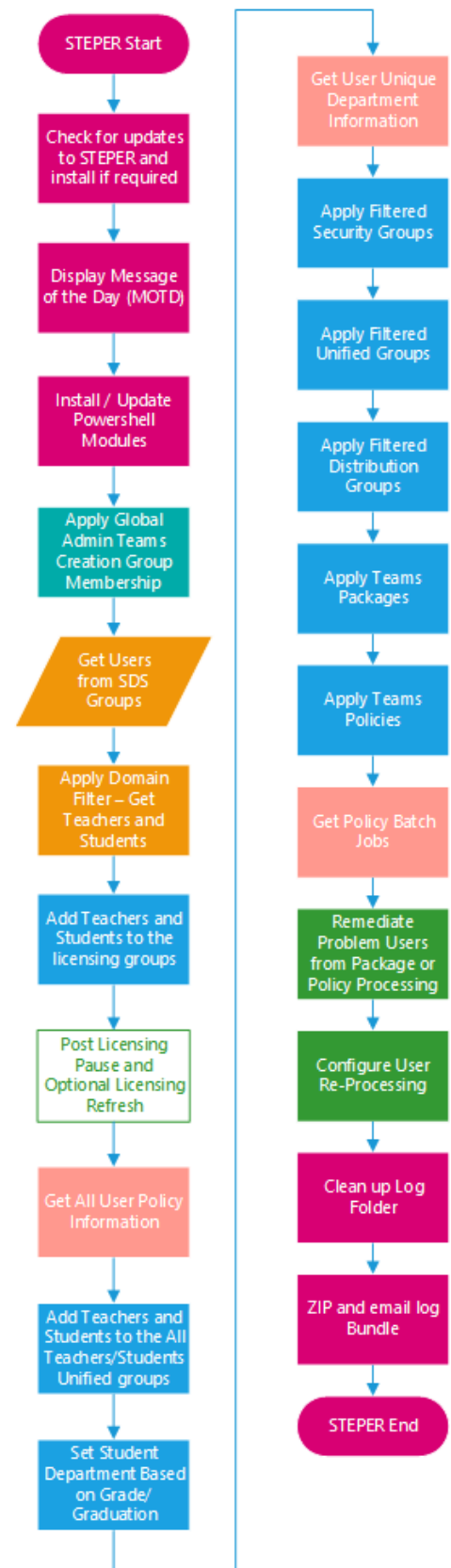
Filename	Intended Use	Requires Editing / Customisation
STEPPER.INI (this is the default configuration file that STEPER will look for)	Contains an example of all possible STEPER actions. Useful if you want to make STEPER make changes to Unified groups or security groups	Yes
STEPPER_DFE.INI	Contains an example of only the groups of settings that are used in a normal DFE deployment, including license groups, All Teachers Group, Student Department Setting, Teacher and Student Policies	
STEPPER_Stone_DFE.INI	As per STEPPER_DFE.INI, but uses the actual Policy Names used in a Stone deployment completed under the DFE Program	
STEPPER_TeamsPolicies.INI	Settings that only enforce DFE standard Team policies. No licensing changes are made	

## Processing Order

STEPPER processes functions in the order shown on the right.

In addition to the processing stages shown, STEPPER will make connections online as requested:

- MSOLService – for Global Admin Teams Creation Group Membership, and to get All Teachers / All Students
- Then STEPPER will connect to Exchangeonline V3, Azure AD, and Teams. This is done in one go, so that any errors are caught before any users are modified, to prevent partial processing.



## [General]

### [General]

```
;Note that the STEPERProfileName is just for STEPER - it is not related to your SDS Profile Name  
STEPPERProfileName=  
;By default, STEPER will operate in Simulation mode - it will just list the changes that would be made  
MakeChanges=False
```

- **STEPPERProfileName** - Enter your STEPER Profile Name. For example, if using different STEPER configurations for different subdomains, change the STEPERProfileName to the subdomain name. The profile name may be left blank and will default to “(Untitled-<onmicrosoftname>)”. It is highly recommended to use profile names of your own if you intend to be using multiple instances of STEPER in a deployment. If you have a complex deployment, it is a good idea for the profile name to give a summary of what STEPER is being asked to do – for example: New Teachers at myschool.org
- **MakeChanges** - By default, this setting is False, meaning that STEPER operates in simulation only mode. No changes are made in this mode, however all items are processed and logged to show you the changes that would be made. Change this setting to True to actually make changes.

**TIP:** Alternatively, from the command line, use the -MakeChanges switch to stop using Simulation mode. This will override the MakeChanges setting in the INI file.

```
PS C:\_stone\STEPPER> .\STEPPER.ps1 -MakeChanges
```

## [Credentials]

### [Credentials]

```
;Note that this Account should have the Global Admin role and should be licensed  
GA_Username=globaladmin@myschool.org.uk  
GA_Password=  
Protect_Password=True
```

- GA\_Username – Specify the Global Administrator (GA) username. This account should be licensed
- GA\_Password – Specify the Global Administrator Password in clear text
- Protect\_Password – By default, STEPER will encrypt the GA Password from clear text on its first run. If you don't want STEPER to encrypt the password, set this to False

**TIP 1:** When you save the configuration, the GA\_Password may be stored in clear text, so it's important that you do not save the completed STEPER INI file on a public or commonly accessible drive or location – or if you do, run STEPER straight away to protect the password. See Password Encryption below.

**TIP 2:** If you use federated access in your tenancy, you may need to specify your global admin users as [globaladmin@myschool.onmicrosoft.com](mailto:globaladmin@myschool.onmicrosoft.com), as the onmicrosoft.com domain is not federated. If you use federated access and use [globaladmin@myschool.org.uk](mailto:globaladmin@myschool.org.uk), STEPER may not be able to connect to online services.

## Password Encryption

From STEPER 1.0.64 by default the clear text GA\_Password will be encrypted, to provide some protection against clear-text password harvesting.

- The GA\_Password field will be emptied
- A new field will be created by STEPER, called ProtectedGA\_Password
- The STEPER.INI file containing the encrypted password can only be used on the same Windows machine. If you move STEPER to a new machine, you will need to fill in GA\_Password again
- The STEPER.INI file containing the encrypted password can be shared between different Windows user accounts on the same PC
- If you need to update the account password, simply fill in GA\_Password again. STEPER will then encrypt the new password

### [Credentials]

```
;Note that this Account should have the Global Admin role and should be licensed  
GA_Username=globaladmin@myschool.org.uk  
GA_Password=  
Protect_Password=True  
ProtectedGA_Password=6738ghfjdnv478nbcvufdchnng745b78hvdijfgsoldkhbUHUFH845jfg944
```

[Tenancy]

```
[Tenancy]
;Example OnMicrosoftCOMPrefix=ourtenancyname (without the onmicrosoft.com suffix)
OnMicrosoftCOMPrefix=myschool
;PrivateDomain is not currently used in STEPER
PrivateDomain=myschool.org.uk
```

These settings determine your basic M365 connection.

The OnMicrosoftCOMPrefix is mandatory if the optional “[CreateAzureADAccessPolicy](#)” setting is used. “CreateAzureADAccessPolicy” is used to set an Azure AD access policy which could be useful in some tenancies where STEPER is being disconnected before it has finished making lengthy changes to Microsoft Teams.

The PrivateDomain setting is not used currently used in STEPER but we recommend it should be configured correctly.



## [GlobalAdminTeamsCreation]

### [GlobalAdminTeamsCreation]

```
AddGlobalAdminToCreateTeamsGroup=True  
CreateTeamsGroupName=Stone-Allow-Team-Create  
DisableNameOrIDChecking=False
```

In the standard Stone DFE deployment, the Global Administrator account is added to a special Security Group which is given permission to in turn create unified groups, which are used to create Teams. The [GlobalAdminTeamsCreation] setting adds the STEPER GA\_Username account to the same special Security Group. This function exists in case the GA account used for STEPER is different to the original account used in the DFE deployment.

To use this function, you will need to specify the name of the special unified group, for example Stone-Allow-Team-Create.

**TIP:** If you have received the standard Stone DFE deployment, The All Teachers SDS group normally already has membership of this group, so you do not need to add new teachers as member of this special unified group

If you have not had a Stone DFE deployment, you can use this code [here](#) to create your own Security group and limit those in your organisation that can create Teams to All Teachers and the Global Admin.

**Note:** These settings were previously called AddGlobalAdminToCreateTeamsUnifiedGroup and CreateTeamsUnifiedGroupName in STEPER 1.0.20 and earlier, which does not accurately reflect the functionality. From STEPER 1.0.21, the settings have been renamed. Any old settings encountered in the settings file will be upgraded to the new names.

## DisableNameOrIDChecking

By default, STEPER will find source groups matched exactly to one of the following:

- Common Name
- DisplayName
- EmailAddress

You can turn off the additional STEPER additional matching protection by adding the setting DisableNameOrIDChecking=True. If you specify a group object ID as the group name, additional matching protection is automatically turned off for security groups.





## Source Groups – [TeacherSourceGroups] and [StudentSourceGroups]

```
[TeacherSourceGroups]
;In most simple situations use RequiredOrGroups, or RequiredOrRecursiveGroups
;Use RequiredOrGroups=* or RequiredOrRecursiveGroups=* to select all Users as Teachers
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=All Teachers
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=True
;Optionally Filter the found Teachers based on their M365 email address domain name - do not include the @ symbol in the filter
DomainFilter=
DontProcessStudentsFoundinTeachersSourceGroup=False
LogAvailableUsers=False

[StudentSourceGroups]
;In most simple situations use RequiredOrGroups, or RequiredOrRecursiveGroups
;Use RequiredOrGroups=* or RequiredOrRecursiveGroups=* to select all Users as Students
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=All Students
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=True
;Optionally Filter the found Students based on their M365 email address domain name - do not include the @ symbol in the filter
DomainFilter=
DontProcessTeachersFoundinStudentsSourceGroup=False
LogAvailableUsers=False
```

STEPPER needs to find unlicensed Teachers and Students that need processing. It primarily does this by looking for users in the All Students and All Teachers SDS groups that are not part of the corresponding Stone DFE licensing groups.

The TeacherSourceGroups and StudentSourceGroups sections allow you to specify the name of the SDS Source Security Groups (e.g. All Teachers and All Students), e.g.:

```
TeacherSourceGroups]
RequiredOrGroups=AllTeachers
```

```
StudentSourceGroups]
RequiredOrGroups=AllStudents
```

**NOTE:** You can specify an object ID in any field that accepts group names. STEPPER will automatically detect when an object ID has been provided. Using an object ID is useful when the group name is used more than once in your tenancy.



## Wildcard Processing - What If You aren't using SDS or don't have suitable Teacher or Student Source Groups?

If you don't have an All Teachers or All Students group that you can use as a source group, then you can ask STEPER to use ALL cloud users as an initial source. **Use with caution!**

To use all cloud users as Teachers, set the Group Name to \*

```
[TeacherSourceGroups]
```

```
RequiredOrGroups=*
```

To use all cloud users as Students, set the Group Name to \*

```
[StudentSourceGroups]
```

```
RequiredOrGroups=*
```

## What's the Difference Between Required, and RequiredOR?

- With Required, a user has to be a member of ALL of those groups.
- With RequiredOR, a user has to be a member of ANY of those groups.

Example:

```
RequiredGroups=
```

```
RequiredRecursiveGroups=
```

```
RequiredOrGroups=Year6,Year7,Year8
```

```
RequiredOrRecursiveGroups=
```

In this example, Students from all three year groups are selected for processing.

Example 2:

```
[StudentSourceGroups]
```

```
RequiredGroups=
```

```
RequiredRecursiveGroups=
```

```
RequiredOrGroups=
```

```
RequiredOrRecursiveGroups=Year6,Year7,Year8
```

In this example, Students from all three year groups and any sub-groups are selected for processing.

Example 3:

```
[StudentSourceGroups]
```

```
RequiredGroups=Year6,Year7,Year8
```

```
RequiredRecursiveGroups=
```

```
RequiredOrGroups=
```

```
RequiredOrRecursiveGroups=
```

In this example, only Students that are members of ALL three groups are selected for processing.



## Recursion

STEPER supports looking for [implied memberships](#) of the source groups, by recursively working through the supplied group names to find users.

Non-Recursive Setting	Recursive Setting
RequiredGroups	RequiredRecursiveGroups
RequiredOrGroups	RequiredOrRecursiveGroups
NotRequiredGroups	NotRequiredRecursiveGroups

## DisableNameOrIDCheckingForGroupRules

By default, STEPER will find source groups matched exactly to one of the following:

- Common Name
- DisplayName
- EmailAddress

You can turn off the additional STEPER additional matching protection by adding the setting `DisableNameOrIDCheckingForGroupRules=True`. If you specify a group object ID as the group name, additional matching protection is automatically turned off for these security groups.

Example:

```
[StudentsSourceGroups]  
RequiredOrGroups=AllStudents  
DisableNameorIDCheckingForGroupRules=True
```

## IncludeDistributionGroupsInGroupRules

Set `IncludeDistributionGroupsInGroupRules=False` (for Teachers) or `IncludeDistributionGroupsInGroupRules=False` (for Students) to make STEPER ignore `DistributionList` groups when looking for matching groups, so that only `Security` or `MailEnabledSecurity` groups will match

Example:

```
[TeacherSourceGroups]  
RequiredOrGroups=AllTeachers  
IncludeDistributionGroupsInGroupRules=False
```



## NotRequiredGroups / NotRequiredRecursiveGroups: Excluding Teachers or Students from Processing

From STEPER 1.0.69, you can now specify groups of Teachers or Students that contain users that should not be processed.

- NotRequiredGroups=
- NotRequiredRecursiveGroups=
- NotRequiredGroups=
- NotRequiredRecursiveGroups=

### Example:

```
TeacherSourceGroups]
RequiredOrGroups=AllTeachers
NotRequiredGroups=All_ExternalStaff
```

In this example, AllTeachers are selected for processing, but then members of AllTeachers that are also members of All\_ExternalStaff are removed from being processed.

You can just use the “NotRequired” settings and leave the “Required” settings blank. In this case, all users will be selected under Required by default:

### Example:

```
[TeacherSourceGroups]
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=
NotRequiredGroups=AllTeachers
NotRequiredRecursiveGroups=
```

In this example all users will be selected, and then All Teachers will be de-selected.



## Source Group Domain Filtering

The TeacherSourceGroups and StudentSourceGroups sections also allow you to filter eligible users based on the domain name portion of their primary username.

When specifying the domain names to be used for filtering, do not include the @ symbol in the filter. The domain filter field allows you to specify a comma separated list of domains.

### Example:

```
[TeacherSourceGroups]
RequiredOrGroups=All Teachers
DomainFilter=subdomain.myschool.org.uk, subdomain2.myschool.org.uk
```

- A Teacher that is a member of All Teachers group, that has with the username [joblogs@myschool.org.uk](mailto:joblogs@myschool.org.uk) will not be included in the Teacher Master List, as their suffix myschool.org.uk does not match subdomain.myschool.org.uk or subdomain2.myschool.org.uk
- A user that is a member of the All Teachers group, with a username of [johnsmith@subdomain.myschool.org.uk](mailto:johnsmith@subdomain.myschool.org.uk) will be added to the Teacher Master List



## Removing Students from the Teacher Source Groups / Removing Teachers from the Student Source Groups

There are also some Teacher and Student specific settings you can use.

[TeacherSourceGroups] Settings	Purpose
DontProcessStudentsFoundinTeacherSourceGroups	Remove Students that are found in the TeacherSourceGroups

[StudentSourceGroups] Settings	Purpose
DontProcessTeachersFoundinStudentSourceGroups	Remove Teachers that are found in the StudentSourceGroups

There may be situation where you use the All Users (\*) source group for some users only have source groups for either Students or Teachers. Use `DontProcessStudentsFoundinTeacherSourceGroups` or `DontProcessTeachersFoundinStudentSourceGroups` to remove these users from the Source Group.

### Example

```
TeacherRequiredOrGroups=All Teachers  
StudentRequiredOrGroups=*  
DontProcessTeachersFoundinStudentsSourceGroup=True
```

In this situation, Teachers found in the “All Users” source group will be removed from processing in the Students Source group.

This allows you to identify Students and Teachers when you only have only one user specific user group:

Teachers Source Group Members:

Bob  
Sue

Students Source Group Members: (All Users)

Student1  
Student2  
Bob  
Sue

Resulting Source users for processing:

Teachers:  
Bob  
Sue

Students:  
Student1  
Student2



## Recording the Users Available for Processing

The LogAvailableUsers setting also allows you to record all of the users that STEPER finds for processing.

Setting	Turned on by Default	Action
LogAvailableUsers	No	Turn these on to record the users in the log file.

### Example:

```
[TeacherSourceGroups]  
RequiredOrRecursiveGroups=AllTeachers  
LogAvailableUsers=True
```

These settings are turned off by default to minimise the size of the log file.



## Source Groups Settings Summary

[TeacherSourceGroups] and [StudentSourceGroups] Settings	Purpose
RequiredGroups	Provide a comma separated list of groups (or object IDs) that the user must belong to. If the group does not exist, processing will stop. The user must belong to all groups in the list.
RequiredRecursiveGroups	Provide a comma separated list of groups (or object IDs) or member groups that the user must belong to. Groups processed recursively, so the user could be a member of the primary group, or a group that belongs to the named group. If the primary group does not exist, processing will stop. The user must belong to all groups (or recursive groups) in the list.
RequiredOrGroups	Provide a comma separated list of groups (or object IDs) that the user must belong to. If the group does not exist, processing will stop. The user must belong at least one of the groups in the list.
RequiredOrRecursiveGroups	Provide a comma separated list of groups (or object IDs) or member groups that the user must belong to. Groups processed recursively, so the user could be a member of the primary group, or a group that belongs to the named group. If the primary group does not exist, processing will stop. The user must belong at least one of the groups or recursive groups in the list.
NotRequiredGroups	Provide a comma separated list of groups (or object IDs) that the user must not belong to. If the group does not exist, by default this is OK, unless you specify DontAllowMissingNotRequiredGroups=True, in which case processing will stop. The user must not belong to any of the groups in the list.
NotRequiredRecursiveGroups	Provide a comma separated list of groups (or object IDs) or member groups that the user must not belong to. Groups processed recursively, so the user must not be a member of the primary group, or a group that belongs to the named group. If the primary group does not exist, by default this is OK, unless you specify DontAllowMissingNotRequiredGroups=True, in which case processing will stop.
DisableGroupNameorIDCheckingForGroupRules	By default, where you provide a group name, STEPER will perform additional checking to ensure that the groups match. For example, "Office" should not match "Office Staff". You can turn off this additional checking by changing this setting to True
DontAllowMissingNotRequiredGroups	With NotRequiredGroups and NotRequiredRecursiveGroups, by default these groups don't need to exist, as the user is required not to be a member to proceed.  By setting DontAllowMissingNotRequiredGroups =True, if these groups don't exist, processing will stop.
IncludeDistributionGroupsInGroupRules	By default, all types of Security group (including Mail Enabled Security groups and Distributiongroups) are included in the search. To help in situations where you have many groups with the same or similar names, you can specify that Distribution groups are not included in the search by changing this setting to False
DomainFilter	Filter eligible users based on the domain name
[TeacherSourceGroups] DontProcessStudentsFoundinTeacherSourceGroups  [StudentSourceGroups] DontProcessTeachersFoundinStudentSourceGroups	Remove Students that are found in the TeacherSourceGroups / Remove Teachers that are found in the StudentSourceGroups
LogAvailableUsers	Record all of the users that STEPER finds for processing



## Master List Processing Order

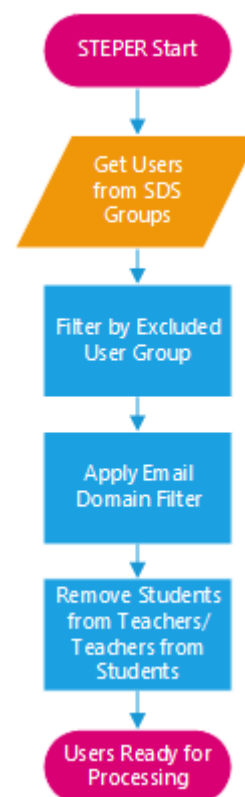
The results from this section form the **Teacher Master List** and **Student Master List**.

### Teachers:

- Get All Teachers that are member of the Teachers Source Group
- If Filtering is enabled, remove any users that don't have a username that matches the Teachers domain filter.

### Students:

- Get All Students that are member of the Students Source Group
- If Filtering is enabled, remove any users that don't have a username that matches the Students domain filter.



## [LicensingGroups]

### [LicensingGroups]

```
AssignTeacherstoLicensingGroup=True
AllTeachersLicensingGroupName=Stone-Licensing-A1Faculty
DisableTeacherGroupNameorIDChecking=False
CreateAllTeachersLicensingGroupIfNotExists=False
;As an additional safety feature, you can specify that each action is only carried out to unlicensed users.
;Or you can choose to apply actions to users irrespective of their licensing status. In this case, if the us
OnlyApplytoUnlicensedTeachers=False
AssignStudentstoLicensingGroup=True
AllStudentsLicensingGroupName=Stone-Licensing-A1Student
DisableStudentGroupNameorIDChecking=False
CreateAllStudentsLicensingGroupIfNotExists=False
OnlyApplytoUnlicensedStudents=False
```

When examining the **Teachers Master List** (SDS All Teachers) and **Students Master List** (SDS All Students), STEPER uses Security groups to find new Teachers or Students.

- A Teacher in the All Teachers group but not in the Security group must be a newly added teacher
- A Teacher in the All Teachers group and in the Security group cannot be an newly added teacher

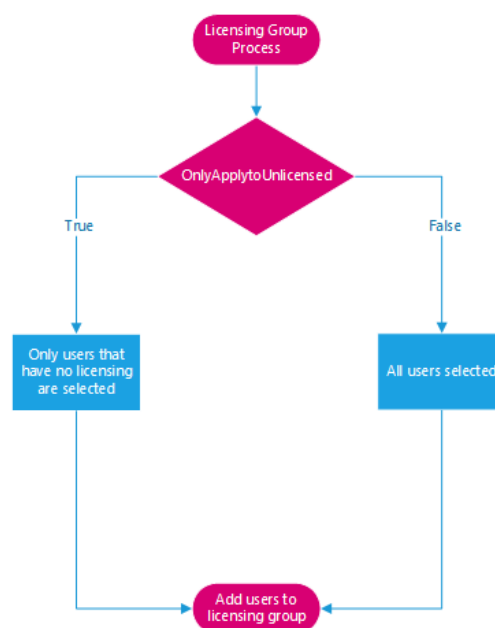
When it comes to applying policies and the other functions later on, whilst you can always ask STEPER later on to process all users (e.g. "ApplyAllStudents...=True", it's usually much faster just to process the newly added users.

The security groups used to determine new users are referred to as licensing groups, as it's easy to use these groups to apply licenses to users through AzureAD.

## Using LicensingGroups

Teachers or Students may already be licensed if this has been done manually. Licensing is cumulative, so adding a teacher to the licensing group (Stone-Licensing-A1Faculty in the example) will not remove any licensing options from the user.

However, as an additional feature, you can specify that users are only added to the licensing group if they are not actually already licensed. To do this, use the `OnlyApplyToUnlicensedTeachers=True` or `OnlyApplyToUnlicensedStudents=True` options. This will mean that users that already have some form of licensing will not have any additional applications or features licensed to them.



## SleepSecondsAfterApplyingLicensing

Microsoft can take some substantial time to actually license a user. You can use the SleepSecondsAfterApplyingLicensing option in the [Misc] section to wait for licensing to take effect before processing the next functions. The default is 600 seconds or 10 minutes.

### [Misc]

;The setting below will introduce a pause after applying licensing groups, if any licensing group changes were made  
SleepSecondsAfterApplyingLicensing=600

## Licensing Group Examples

Aim	Target / Section	Example Settings Required
Make No Licensing Changes	[LicensingGroups]	Teachers: AssignTeacherstoLicensingGroup=False Students: AssignStudentstoLicensingGroup=False
Add Students to the licensing group if they are not already members. Ignore current licensing status.	[LicensingGroups]	AssignStudentstoLicensingGroup=True AllStudentsLicensingGroupName=Stone-Licensing-A1Student OnlyApplytoUnlicensedStudents=False
Add Teachers to the licensing group if they are not already members, but only if they are currently unlicensed.	[LicensingGroups]	AssignTeacherstoLicensingGroup=True AllTeachersLicensingGroupName=Stone-Licensing-A1Faculty OnlyApplytoUnlicensedTeachers=False
Add Students to the licensing group if they are not already members, but include a filter to add more licensing to certain Students		Use [StudentSecurityGroup1] to [StudentSecurityGroup100]

## DisableNameOrIDChecking

By default, STEPER will find source groups matched exactly to one of the following:

- Common Name
- DisplayName
- EmailAddress

You can turn off the additional STEPER additional matching protection by adding the setting DisableTeacherGroupNameorIDChecking=True for Teachers, or DisableStudentGroupNameorIDChecking=True for Students.

If you specify a group object ID as the group name, additional matching protection is automatically turned off for these security groups.

**TIP:** Set DisableTeacherGroupNameorIDChecking=True (for Teachers) or DisableStudentGroupNameorIDChecking=True (for Students) to make STEPER ignore the type of source group. By default, only Security or MailEnabledSecurity groups are permitted. Use these True settings to accept groups of type "DistributionList"

## Creating Licensing Groups

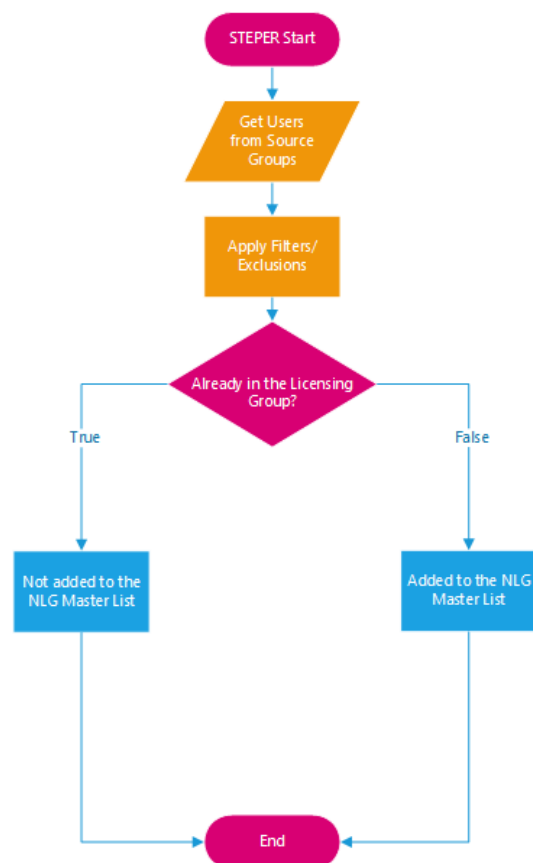
If the licensing groups do not exist, you can ask STEPER to create them by setting `CreateAllTeachersLicensingGroupIfNotExists=True` or `CreateAllStudentsLicensingGroupIfNotExists=True`. These groups will not be created in simulation mode (`MakeChanges=False`) and will also not be created if you set the corresponding `AssignTeacherstoLicensingGroup=False`.

Setting	Situation	Output
[General]MakeChanges=False AssignTeacherstoLicensingGroup=True AllTeachersLicensingGroupName=Stone-Licensing-A1Faculty CreateAllStudentsLicensingGroupIfNotExists=True	Licensing group does not exist	<ul style="list-style-type: none"> <li>Licensing group is not created, as <code>MakeChanges=False</code></li> <li>New users are not added to the group</li> </ul>
[General]MakeChanges=True AssignTeacherstoLicensingGroup=True AllTeachersLicensingGroupName=Stone-Licensing-A1Faculty CreateAllStudentsLicensingGroupIfNotExists=False	Licensing group does not exist	<ul style="list-style-type: none"> <li>Licensing group is not created, as <code>CreateAllStudentsLicensingGroupIfNotExists=False</code></li> <li>New users are not added to the group</li> </ul>
[General]MakeChanges=True AssignTeacherstoLicensingGroup=False AllTeachersLicensingGroupName=Stone-Licensing-A1Faculty CreateAllStudentsLicensingGroupIfNotExists=True	Licensing group does not exist	<ul style="list-style-type: none"> <li>Licensing group is not created, as <code>AssignTeacherstoLicensingGroup=False</code></li> <li>New users are not added to the group</li> </ul>
[General]MakeChanges=True AssignTeacherstoLicensingGroup=True AllTeachersLicensingGroupName=Stone-Licensing-A1Faculty CreateAllStudentsLicensingGroupIfNotExists=True	Licensing group does not exist	<ul style="list-style-type: none"> <li>Licensing group <code>Stone-Licensing-A1Faculty</code> is created</li> <li>New users are added to the group</li> </ul>
[General]MakeChanges=True AssignTeacherstoLicensingGroup=True AllTeachersLicensingGroupName=Stone-Licensing-A1Faculty CreateAllStudentsLicensingGroupIfNotExists=True	Licensing group exists	<ul style="list-style-type: none"> <li>Licensing group is not created as it already exists</li> <li>New users are added to the group</li> </ul>

**TIP:** Alternatively, you can use the code [here](#) to create your own licensing groups

## Not-in Licensing Group (NLG) Master List

Students and Teachers that are not members of the licensing groups will be added to the **Not-in Licensing Group (NLG) Teacher Master list** and **Not-in Licensing Group (NLG) Student Master list**. This is important to bear in mind for subsequent functions.



Teacher Master List or Student Master List	Teachers or Students in the All Teachers or All Students SDS Groups, optionally filtered by domain
NLG Teacher Master List or NLG Student Master List	Teachers or Students in the All Teachers or All Students SDS Groups, optionally filtered by domain, that were not members of the relevant Teacher or Student Licensing Group

## [AllTeachersGroup] and [AllStudentsGroup]

### [AllTeachersGroup]

```
AddAllTeacherstoUnifiedGroup=True
OnlyApplytoUnlicensedTeachers=False
AllTeachersUnifiedGroup=Staff Team
CreateAllTeachersUnifiedGroupIfNotExists=False
DisableNameOrIDChecking=False
```

### [AllStudentsGroup]

```
;Not part of the standard Stone DFE Config
AddAllStudentstoUnifiedGroup=False
OnlyApplytoUnlicensedStudents=False
AllStudentsUnifiedGroup=Student Group
CreateAllStudentsUnifiedGroupIfNotExists=False
DisableNameOrIDChecking=False
```

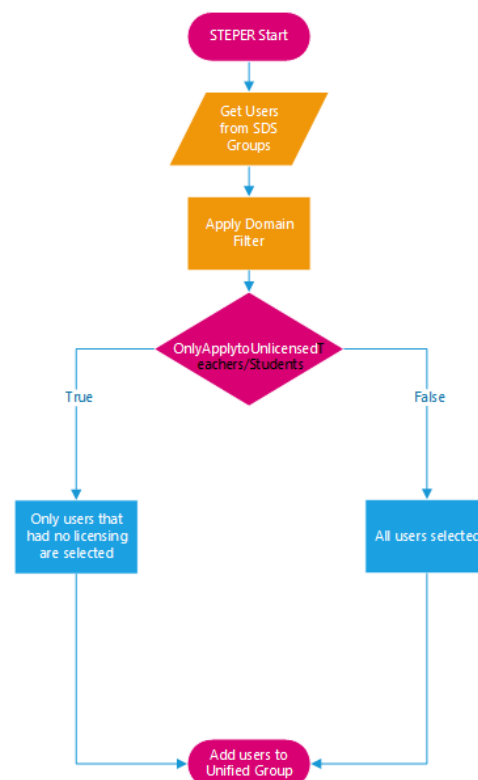
Use this section to add users from the **Teachers Master List** and **Students Master List** to single M365 Unified Groups.

In the standard Stone DFE Deployment, All Teachers are added to an M365 group called Staff Team.

In the standard Stone DFE Deployment, Students are not added to a single Unified group, so this is normally left disable by setting `AddAllStudentstoUnifiedGroup=False`

As an additional feature, you can specify that users are only added to the licensing group if they are not actually already licensed. To do this, use the `OnlyApplyToUnlicensedTeachers` or `OnlyApplyToUnlicensedStudents` options. This will mean that users that were previously licensed (and in use?) will not be added to the Unified group.

**TIP:** These Teachers or Students may now actually be licensed if they were added to the licensing group in the previous step. However, if you use the `OnlyApplyToUnlicensedTeachers` or `OnlyApplyToUnlicensedStudents` options STEPER will remember the users that were not licensed at the start of this process.





## Creating Your Own Groups

You can create your own groups for this function using the code [here](#). Or alternatively, STEPER can create these groups for you, if you specify `CreateAllTeachersLicensingGroupIfNotExists=True` and/or `CreateAllStudentsLicensingGroupIfNotExists=True`.

Note that the Global Admin account used to create these groups is automatically added as a member of the group, and is the owner.

## AllTeachersGroup and AllStudentsGroup Examples

Aim	Target / Section	Example Settings Required
Add Teachers that are missing from the all teachers Staff team into the group, irrespective of licensing status	[AllTeachersGroup]	AddAllTeacherstoUnifiedGroup=True AllTeachersUnifiedGroup=Staff Team
Add Students that are missing from the all students team into the group, but only if they were previously unlicensed	[AllStudentsGroup]	AddAllStudentstoUnifiedGroup=True AllStudentsUnifiedGroup=Student Team OnlyApplytoUnlicensedStudents=True
Add Students that are missing from the all students team into the group, but only if they were not in the student licensing group		Use [StudentUnifiedGroup1] to [StudentUnifiedGroup100]

## Creating AllTeachers Group and AllStudents Group

If the AllTeachersUnifiedGroup does not exist, you can ask STEPER to create it by setting CreateAllTeachersUnifiedGroupIfNotExists=True

Note that the group will not be created in simulation mode (MakeChanges=False) and will also not be created if you set AddAllTeacherstoUnifiedGroup=False

If the AllStudentsUnifiedGroup does not exist, you can ask STEPER to create it by setting CreateAllStudentsUnifiedGroupIfNotExists=True

Note that the group will not be created in simulation mode (MakeChanges=False) and will also not be created if you set AddAllStudentstoUnifiedGroup=False

Setting	Situation	Output
[General]MakeChanges=False AddAllTeacherstoUnifiedGroup=True AllTeachersUnifiedGroup=Staff Team CreateAllTeachersUnifiedGroupIfNotExists=True	Unified group does not exist	<ul style="list-style-type: none"> <li>Unified group is not created, as MakeChanges=False</li> <li>New users are not added to the group</li> </ul>
[General]MakeChanges=True AddAllTeacherstoUnifiedGroup =True AllTeachersUnifiedGroup=Staff Team CreateAllTeachersUnifiedGroupIfNotExists=False	Unified group does not exist	<ul style="list-style-type: none"> <li>Unified group is not created, as CreateAllTeachersUnifiedGroupIfNotExists=False</li> <li>New users are not added to the group</li> </ul>
[General]MakeChanges=True AddAllTeacherstoUnifiedGroup=False AllTeachersUnifiedGroup=Staff Team CreateAllTeachersUnifiedGroupIfNotExists=True	Unified group does not exist	<ul style="list-style-type: none"> <li>Unified group is not created, as AddAllTeacherstoUnifiedGroup=False</li> <li>New users are not added to the group</li> </ul>
[General]MakeChanges=True AddAllTeacherstoUnifiedGroup =True AllTeachersUnifiedGroup=Staff Team CreateAllTeachersUnifiedGroupIfNotExists=True	Unified group does not exist	<ul style="list-style-type: none"> <li>Unified group Staff Team is created</li> <li>New users are added to the group</li> </ul>
[General]MakeChanges=True AddAllTeacherstoUnifiedGroup=True AllTeachersUnifiedGroup=Staff Team CreateAllTeachersUnifiedGroupIfNotExists=True	Unified group exists	<ul style="list-style-type: none"> <li>Unified group is not created as it already exists</li> <li>New users are added to the group</li> </ul>

**TIP:** Alternatively, you can use the code [here](#) to create your own All Teachers Unified Group called “Staff Team”

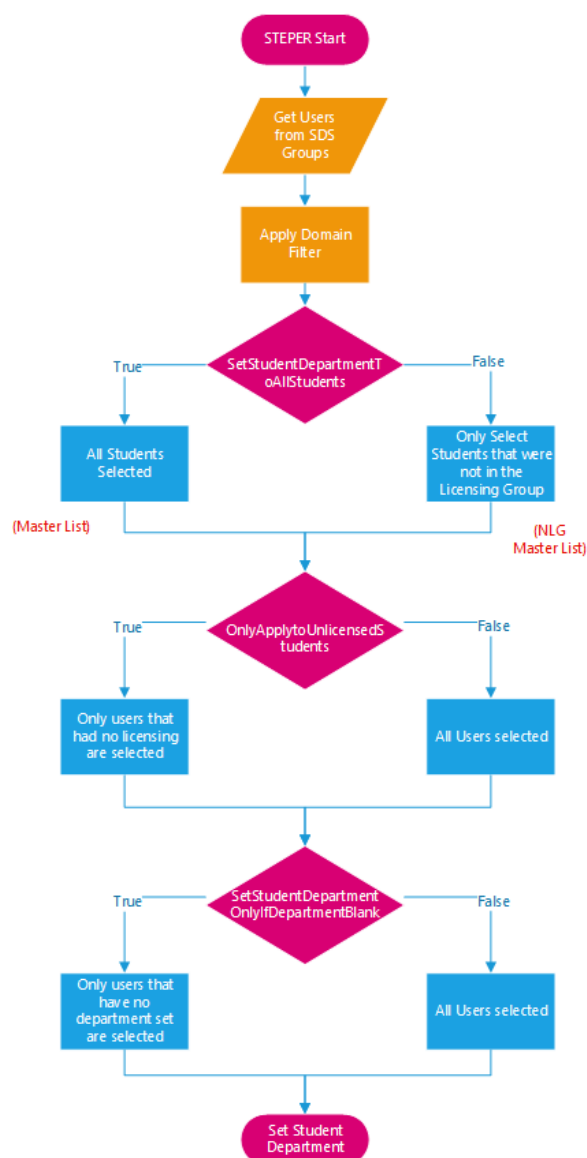


## [Department]

```
[Department]
;To identify Students that need to have this set, i.e. new Students - by default this settir
SetStudentDepartment=False
;By default, SDS Grade information will be used. Alternatively, you can use Graduation Year
UseGraduationYear=False
OnlyApplytoUnlicensedStudents=False
SetStudentDepartmentToAllStudents=False
SetStudentDepartmentOnlyIfDepartmentBlank=True
```

Use this function to apply the SDS Grade information to the Students Department attribute.

- Switch this function on by using `SetStudentDepartment=True`
- If you use the `UseGraduationYear=True` setting, the SDS Graduation Year setting will be used instead of Grade
- By default, Students from the **NLG Student Master list** are used. By using the [SetStudentDepartmenttoAllStudents=True](#) option, Students from the **Student Master list** will be used instead. The Students Master List will have the same number of Students as the NLG Student Master list, or more.
- Additionally, as a safety feature, you can specify that only students that were unlicensed at the start of STEPER are processed, by using the `OnlyApplytoUnlicensedStudents=True` option.
- Finally, you can specify that Students that already have their Department field configured do not have it overwritten, using the option `SetStudentDepartmentOnlyifDepartmentBlank=True` option.





## Reminder:

		Additional Filter:
Student Master List	Students in the All Students SDS Groups, optionally filtered by domain in the [TeacherSourceGroups]/ [StudentSourceGroups] sections	Additionally, filter to only Students that were unlicensed at the start of STEPER, using the OnlyApplytoUnlicensedStudents=True option
NLG Student Master List (default for this function)	Students in the All Students SDS Groups, optionally filtered by domain in the [TeacherSourceGroups]/ [StudentSourceGroups] sections, that were not members of the Student Licensing Group	

## Department Examples

Aim	Target / Section	Example Settings Required
Make No Student Department Changes	[Department]	SetStudentDepartment=False
Set department field on Students that were not in the licensing group. Ignore if they were previously actually licensed or not. Overwrite any existing Department setting.	[Department]	SetStudentDepartment=True SetStudentDepartmentToAllStudents=False OnlyApplytoUnlicensedStudents=False SetStudentDepartmentOnlyIfDepartmentBlank=False
Set department field on Students ALL students. Ignore if they were previously actually licensed or not. Preserve any existing Department setting.	[Department]	SetStudentDepartment=True SetStudentDepartmentToAllStudents=True OnlyApplytoUnlicensedStudents=False SetStudentDepartmentOnlyIfDepartmentBlank=True
Set department field on Students ALL students, but only if they were previously unlicensed. Preserve any existing Department setting.	[Department]	SetStudentDepartment=True SetStudentDepartmentToAllStudents=True OnlyApplytoUnlicensedStudents=True SetStudentDepartmentOnlyIfDepartmentBlank=True



## [TeacherSecurityGroupX] and [StudentSecurityGroupX]

```
[TeacherSecurityGroup1]
;Not part of the standard Stone DFE Config
;STEPPER supports up to 100 different Security Groups for Teachers by default
EnableSecurityGroup=False
;The description is for reporting purposes only
SecurityGroupDescription=Teacher Group1 Description
SecurityGroupName=Security Group 1
;Teachers are selected for Group Processing based on Group Membership and then a Filter
;Group Membership - by default, only New Teachers are processed (i.e. Teachers that were not in
ApplyAllTeachersToGroup=False
;All Teachers Need to be Licensed at this Point (i.e. added to the licensing group) but you can
OnlyApplytoUnlicensedTeachers=False
;Filter - use this to filter Teachers or leave it blank to include the Teachers selected above
;GroupFilter=(Department -eq "ADMINISTRATION")
Filter=
FilterSource=
;Use this to remove Teachers from the group that no longer match the filter. Note that all exist
RemoveUnMatchedMembers=False
;If removing unmatched members, use the setting below to only remove unmatched Teachers
OnlyRemoveUnMatchedTeachers=True
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=True
DomainFilter=
RequiredOrOUs=
NotRequiredOUs=
OnFailureApplytoAllUsersOnNextRun=
Default=False
CollectionID=
CollectionIDIgnoreThreshold=
CollectionIDsToExclude=
```



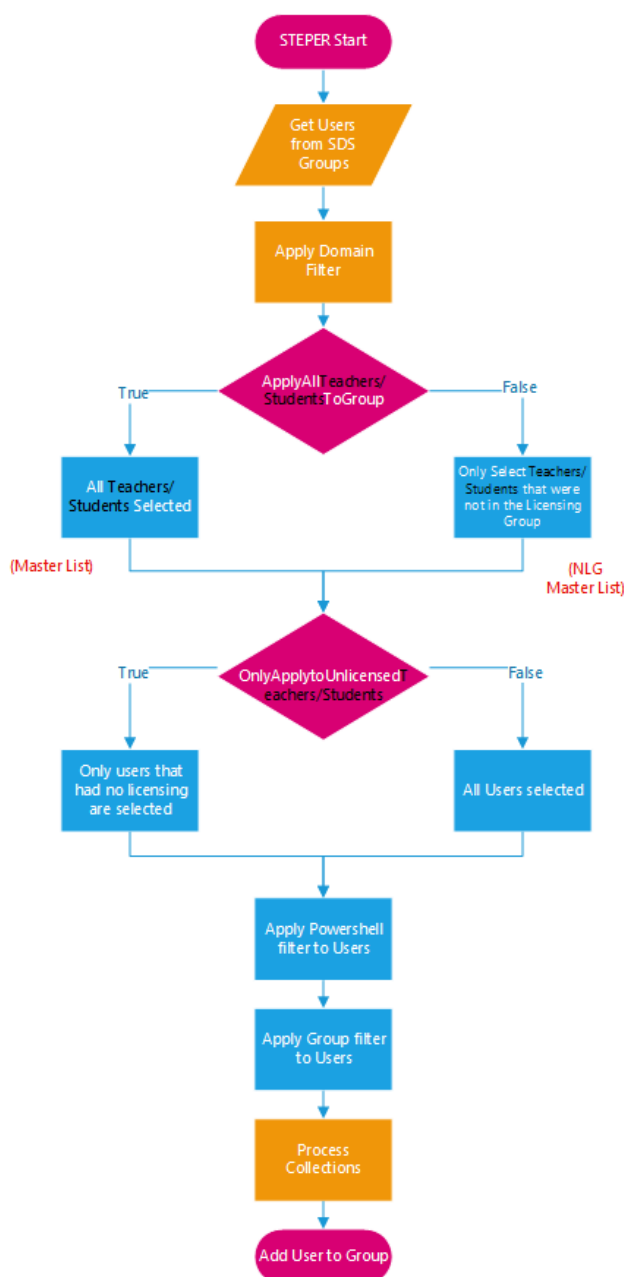
STEPPER supports putting Teachers and Students into Security groups. Whereas the [LicensingGroups] option includes all Teachers or Students, this function allows you to specify a Powershell filter to select Teachers or Students to be put into the Security group.

By default, up to 100 different sets of Teacher settings and 100 different sets of Student settings are available.

	Teacher	Student
1 <sup>st</sup> Settings	[TeacherSecurityGroup1]	[StudentSecurityGroup1]
...		
Last Settings	[TeacherSecurityGroup100]	[StudentSecurityGroup100]

- Switch on the group of settings by using EnableSecurityGroup=True
- Optionally, put a description into the group of settings. This is for your own reference only.
- STEPPER needs to know the name of the Security group or Security group ID to add (or remove) users from. Specify this under SecurityGroupName.
- By default, users from the **NLG Teacher/Student Master** list are used. By using the [ApplyAllTeachersToGroup=True](#) or [ApplyAllStudentsToGroup=True](#) option, Teachers/Students from the **Teacher/Student Master list** will be used instead.

- Additionally, as a safety feature, you can specify that only users that were unlicensed at the start of STEPER are processed, by using the OnlyApplytoUnlicensedTeachers=True or OnlyApplytoUnlicensedStudents=True option. Also see [Refreshing Teacher or Student Licensing Status after Adding Users to Licensing Groups](#)
- You can further select or filter users using the Filter= option. This is a Powershell style [attribute filter](#) string that applies to the entire group of settings.
- You can also filter users based on [Group Rules](#) or their [On-Premise Active Directory location](#) (OU), or via a domain name filter [DomainFilter](#).
- With STEPER 1.0.61 and newer, you can group the application of settings into collections. Settings are processed by collection order. Once a user has been processed in a collection, they are not processed in a higher collection number unless the collection number they were processed in is excluded or under the ignore threshold. See [Collections](#).



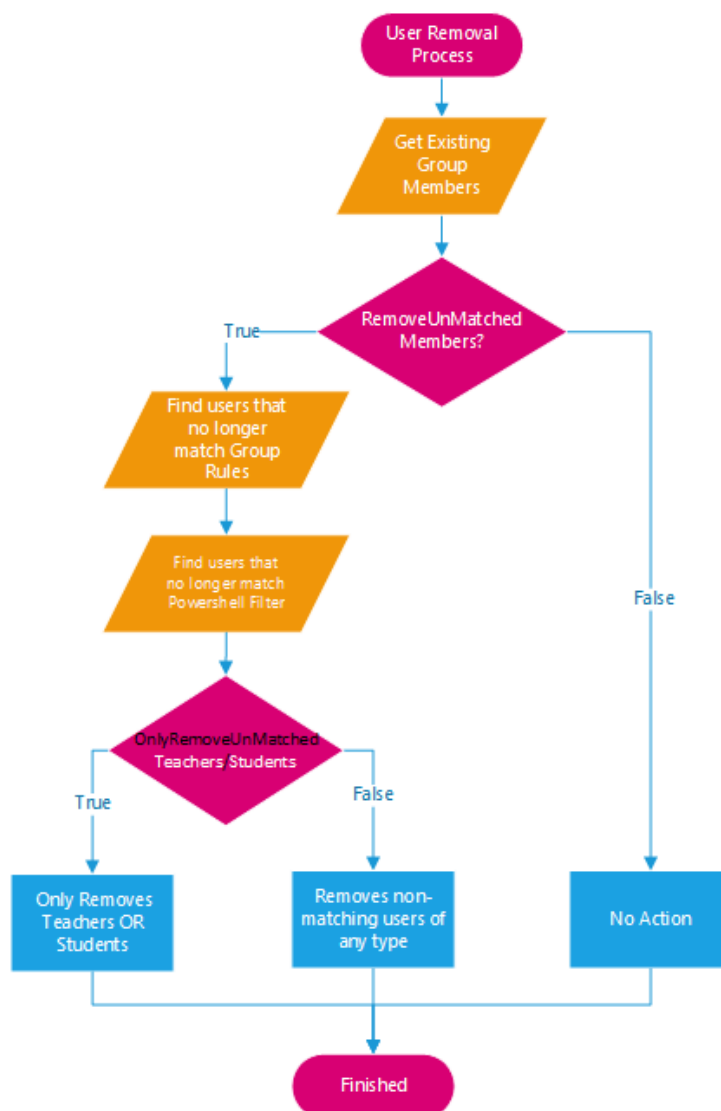
**TIP:** When using Security Groups, don't pad the configuration file section group numbering with zeros. For example:

Good	Bad
[TeacherSecurityGroup1]	[TeacherSecurityGroup01]
[TeacherSecurityGroup20]	[TeacherSecurityGroup020]
[TeacherSecurityGroup100]	[TeacherSecurityGroup099]

## Removing Stale Members from Security Groups

By default, STEPER will not remove users from the Security Group. However, by using the `RemoveUnMatchedMembers=True` option, users that don't match the Group rules, AD/OU Rules and Attribute Filter will be removed from the group. By default, only users of the same type (Teachers/Students) will be removed. Alternatively, use `OnlyRemoveUnMatchedTeachers=False` or `OnlyRemoveUnMatchedStudents=False` to remove other types of users too.

Example 1: Using the settings group [StudentSecurityGroup1], specifying a `Filter=(Department -eq "Year7")` will cause Students matching that filter to be added to the group. With `RemoveUnMatchedMembers=True` option, Students that don't match the filter will be removed from the group. No non-Student users will be removed from the group unless you use the `OnlyRemoveUnMatchedStudents=False` option.



Example 2: Using the settings group [TeacherSecurityGroup1], specifying a `Filter=(Department -eq "Administration")` will cause Teachers matching that filter to be added to the group. With `RemoveUnMatchedMembers=True` option, Teachers that don't match the filter will be removed from the group. No non-Teachers users will be removed from the group unless you use the `OnlyRemoveUnMatchedTeachers=False` option.

**Reminder:** Using the `OnlyRemoveUnMatchedTeachers=False` (under [TeacherSecurityGroupX]) or `OnlyRemoveUnMatchedStudents=False` (under [StudentSecurityGroupX]) has no effect unless you set the corresponding `RemoveUnMatchedMembers=True`



### RemoveUnMatchedMembers Settings

From STEPER 1.0.85, in addition to using RemoveUnMatchedMembers=True, you can now specify specific matching or exclusion reasons for removing a user from the group

Setting	Console Abbreviation	(Internal Boolean Value)	Matching/Exclusion Type
False (default)		0	No users removed from Group
Attribute	A	2	Remove users that don't match the Attribute filter
Group	G	4	Remove users that don't match the Group Rules
OU	O	8	Remove users that don't match the AD OU/CN Rules
Collection	C	16	Remove users Excluded by CollectionID (this means they have already been processed under another Collection unless that previous collection is under the ignore threshold or has been excluded)
1, All or True	AGOC	30	All unmatched users will be removed

Example:

Setting	Result
RemoveUnMatchedMembers=True	All unmatched users will be removed
RemoveUnMatchedMembers=Group,Attribute	Users that don't match the Attribute filter or Group Rules will be removed
RemoveUnMatchedMembers=16	Users Excluded by CollectionID processing



## SecurityGroupName Matching

You can specify an object ID in the GroupName field. STEPER will automatically detect when an object ID has been provided. Using an GUID or object ID is useful when the group name is used more than once in your tenancy.

By default, STEPER will then ensure that the groups meet the name OR ID criteria as an additional level of protection by detecting if a GUID has been provided under SecurityGroupName.

If a Group Name is detected, STEPER will ensure matches are limited to groups based on:

- Common Name
- DisplayName
- EmailAddress

Alternatively, if a GUID is detected in the Group Name field, STEPER will request that group searches are performed against group ObjectIDs

**Note:** Unlike the Filtered Unified Groups and Distribution Groups, M365 allows for searching by name or ObjectID as completely different searches

## DisableNameOrIDChecking

You can turn off the additional STEPER SecurityGroupName additional matching protection by adding the setting `DisableNameOrIDChecking=True` to `[TeacherSecurityGroupX]` or `[TeacherSecurityGroupX]`.

If you specify a group object ID as the group name, additional matching protection is automatically turned off for security groups.

## Turning Off All Filtered Security Group Processing

You can turn off all Filtered Security Group processing for Teachers and Students using the following settings under the `[FunctionControl]` section:

```
TurnOffTeacherFilteredSecurityGroupFunctionality=True  
TurnOffStudentFilteredSecurityGroupFunctionality=True
```

See [here](#) for more information.





## TeacherSecurityGroupX and StudentSecurityGroupX Examples

Aim	Target / Section	Example Settings Required
Make no changes to Teachers Security Groups	[TeacherSecurityGroup1] to [TeacherSecurityGroup100]	EnableSecurityGroup=False
Make no changes to Students Security Groups	[StudentSecurityGroup1] to [StudentSecurityGroup100]	EnableSecurityGroup=False
Add all Teachers that were not in the teachers licensing group to the Email_Licensing_Group. Ignore if they were previously actually licensed or not.  Don't check the group for members that should be removed.	[TeacherSecurityGroup1] to [TeacherSecurityGroup100]	EnableSecurityGroup=True SecurityGroupDescription=Adding All Teachers to Email Licensing Group SecurityGroupName=Email_Licensing_Group ApplyAllTeachersToGroup=False OnlyApplytoUnlicensedTeachers=False Filter=  RemoveUnMatchedMembers=False
Add all Teachers to the S-Drive Security Group. Ignore if they were previously actually licensed or not.  Don't check the group for members that should be removed.	[TeacherSecurityGroup1] to [TeacherSecurityGroup100]	EnableSecurityGroup=True SecurityGroupDescription=Adding All Teachers to S Drive SecurityGroupName=S-Drive ApplyAllTeachersToGroup=True OnlyApplytoUnlicensedTeachers=False Filter=  RemoveUnMatchedMembers=False
To see what user fields are available for filters	[InformationRetrieval]	To Get GetAllUserPolicyInformation=True Or GetUniqueUserDepartmentNames=True
Add all Year 1 Teachers to the Year 1 security Group. Make sure that any teachers that are no longer Year 1 Teachers are removed from the group.	[TeacherSecurityGroup1] to [TeacherSecurityGroup100]	EnableSecurityGroup=True SecurityGroupDescription=Adding Year 1 Teachers to SG Year 1 SecurityGroupName=SG Year 1 ApplyAllTeachersToGroup=True OnlyApplytoUnlicensedTeachers=False Filter=(Department -eq "Year 1")  RemoveUnMatchedMembers=True OnlyRemoveUnMatchedTeachers=True
Add sixth form Students to the Sixth form security Group which has an object ID of d3c8c330-6559-4189-996f-6b2ead2e1a57. Make sure that any users that are no longer part of the sixth form are removed from the group.	[StudentSecurityGroup1] to [StudentSecurityGroup100]	SecurityUnifiedGroup=True SecurityGroupDescription=Adding Sixth form to Security Group SecurityGroupName=d3c8c330-6559-4189-996f-6b2ead2e1a57 ApplyAllStudentsToGroup=True OnlyApplytoUnlicensedStudents=False Filter=(Department -like "Year 12*") -or (Department -like "Year 13*")  RemoveUnMatchedMembers=True OnlyRemoveUnMatchedStudents=False



## [TeacherUnifiedGroupX] and [StudentUnifiedGroupX]

```
[TeacherUnifiedGroup1]
;Not part of the standard Stone DFE Config
;STEPPER supports up to 100 different Unified Groups for Teachers by default
EnableUnifiedGroup=False
;The description is for reporting purposes only
UnifiedGroupDescription=Group1 Description
UnifiedGroupName=Group1
;Teachers are selected for Group Processing based on Group Membership and then a Filter
;Group Membership - by default, only New Teachers are processed (i.e. Teachers that were not in
;The User needs to be licensed at this point for this function to work
ApplyAllTeachersToGroup=False
;All Teachers Need to be Licensed at this Point (i.e. added to the licensing group) but you can
OnlyApplytoUnlicensedTeachers=False
;Filter - use this to filter Teachers or leave it blank to include the Teachers selected above
;GroupFilter=(Department -eq "ADMINISTRATION")
Filter=
FilterSource=
;Use this to remove Teachers from the group that no longer match the filter. Note that all exist
RemoveUnMatchedMembers=False
;If removing unmatched members, use the setting below to only remove unmatched Teachers
OnlyRemoveUnMatchedTeachers=True
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=True
DomainFilter=
RequiredOrOUs=
NotRequiredOUs=
OnFailureApplytoAllUsersOnNextRun=
Default=False
CollectionID=
CollectionIDIgnoreThreshold=
CollectionIDsToExclude=
```



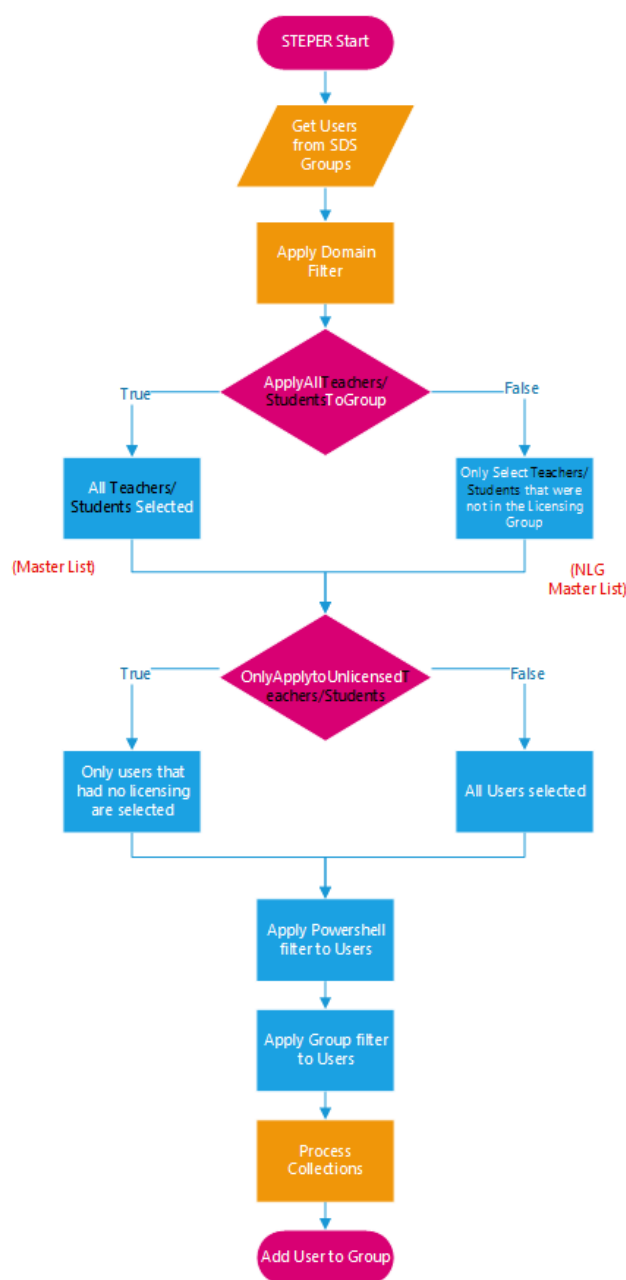
STEPPER supports putting Teachers and Students into Unified groups. Whereas the AllTeachersGroup and AllStudents group option included all teachers or students, this function allows you to specify a Powershell filter to select Teachers or Students to be put into the group.

By default, up to 100 different sets of Teacher settings and 100 different sets of Student settings are available.

	Teacher	Student
1 <sup>st</sup> Settings	[TeacherUnifiedGroup1]	[StudentUnifiedGroup1]
...		
Last Settings	[TeacherUnifiedGroup100]	[StudentUnifiedGroup100]

- Switch on the group of settings by using EnableUnifiedGroup=True
- Optionally, put a description into the group of settings. This is for your own reference only.
- STEPPER needs to know the name or ID of the M365 group to add (or remove) users from. Specify this under UnifiedGroupName.
- By default, users from the **NLG Teacher/Student Master** list are used. By using the [ApplyAllTeachersToGroup=True](#) or [ApplyAllStudentsToGroup=True](#) option, Teachers/Students from the **Teacher/Student Master list** will be used instead.

- Additionally, as a safety feature, you can specify that only users that were unlicensed at the start of STEPER are processed, by using the `OnlyApplytoUnlicensedTeachers=True` or `OnlyApplytoUnlicensedStudents=True` option. Also see [Refreshing Teacher or Student Licensing Status after Adding Users to Licensing Groups](#)
- You can further select or filter users using the `Filter=` option. This is a Powershell style [attribute filter](#) string that applies to the entire group of settings.
- You can also filter users based on [Group Rules](#) or their [On-Premise Active Directory location](#) (OU), or via a domain name filter [DomainFilter](#).
- With STEPER 1.0.61 and newer, you can group the application of settings into collections. Settings are processed by collection order. Once a user has been processed in a collection, they are not processed in a higher collection number unless the collection number they were processed in is excluded or under the ignore threshold. See [Collections](#).



**TIP:** When using Unified Groups, don't pad the configuration file section group numbering with zeros. For example:

Good	Bad
[TeacherUnifiedGroup1]	[TeacherUnifiedGroup01]
[TeacherUnifiedGroup20]	[TeacherUnifiedGroup020]
[TeacherUnifiedGroup100]	[TeacherUnifiedGroup099]

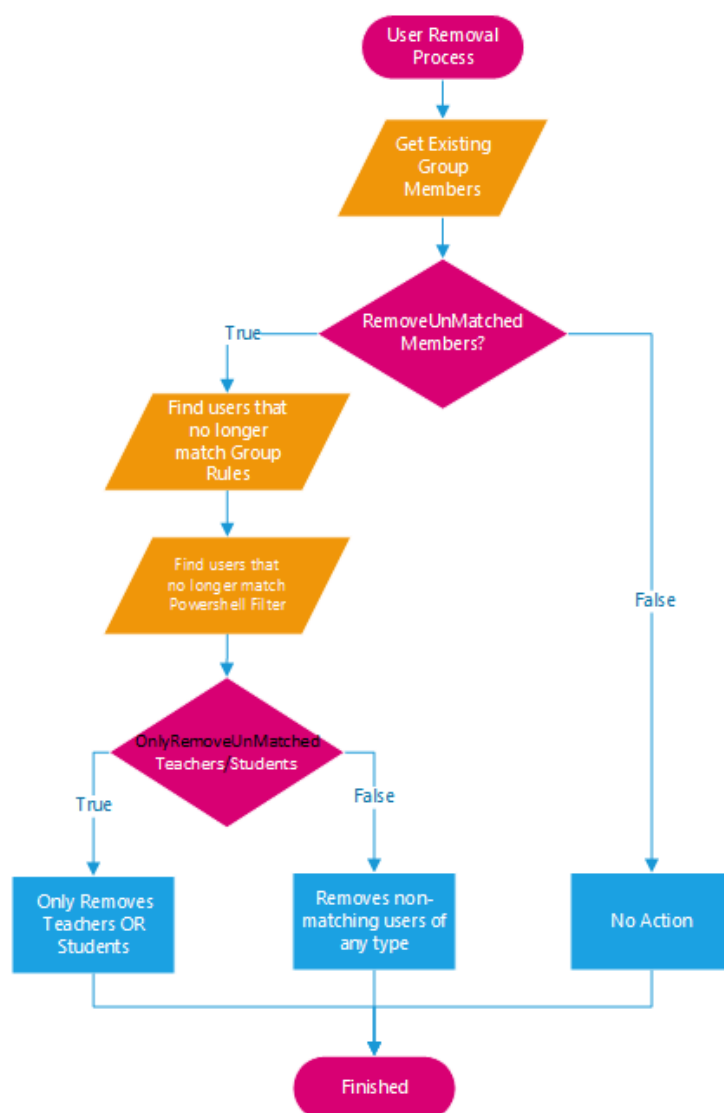
## Removing Stale Members from Unified Groups

By default, STEPER will not remove users from the Unified Group. However, by using the `RemoveUnMatchedMembers=True` option, users that don't match the Group rules, AD/OU Rules and Attribute Filter will be removed from the group. By default, only users of the same type (Teachers/Students) will be removed.

Alternatively, use

`OnlyRemoveUnMatchedTeachers=False` or `OnlyRemoveUnMatchedStudents=False` to remove other types of users too.

Example 1: Using the settings group [StudentUnifiedGroup1], specifying a `Filter=(Department -eq "Year7")` will cause Students matching that filter to be added to the group. With `RemoveUnMatchedMembers=True` option, Students that don't match the filter will be removed from the group. No non-Student users will be removed from the group unless you use the `OnlyRemoveUnMatchedStudents=False` option.



Example 2: Using the settings group [TeacherUnifiedGroup1], specifying a `Filter=(Department -eq "Administration")` will cause Teachers matching that filter to be added to the group. With `RemoveUnMatchedMembers=True` option, Teachers that don't match the filter will be removed from the group. No non-Teachers users will be removed from the group unless you use the `OnlyRemoveUnMatchedTeachers=False` option.

**Reminder:** Using the `OnlyRemoveUnMatchedTeachers=False` (under [TeacherUnifiedGroupX]) or `OnlyRemoveUnMatchedStudents=False` (under [StudentUnifiedGroupX]) has no effect unless you set the corresponding `RemoveUnMatchedMembers=True`



### *RemoveUnMatchedMembers Settings*

From STEPER 1.0.85, in addition to using RemoveUnMatchedMembers=True, you can now specify specific matching or exclusion reasons for removing a user from the group

Setting	Console Abbreviation	(Internal Boolean Value)	Matching/Exclusion Type
False (default)		0	No users removed from Group
Attribute	A	2	Remove users that don't match the Attribute filter
Group	G	4	Remove users that don't match the Group Rules
OU	O	8	Remove users that don't match the AD OU/CN Rules
Collection	C	16	Remove users Excluded by CollectionID (this means they have already been processed under another Collection unless that previous collection is under the ignore threshold or has been excluded)
1, All or True	AGOC	30	All unmatched users will be removed

Example:

Setting	Result
RemoveUnMatchedMembers=True	All unmatched users will be removed
RemoveUnMatchedMembers=Group,Attribute	Users that don't match the Attribute filter or Group Rules will be removed
RemoveUnMatchedMembers=16	Users Excluded by CollectionID processing



### UnifiedGroupName Matching

M365 will select groups based on the supplied Unified Group Name or Unified Group ID. M365 does not differentiate between these searches.

By default, STEPER will then ensure that the groups meet the name OR ID criteria as an additional level of protection by detecting if a GUID has been provided under UnifiedGroupName. Using an GUID or object ID is useful when the group name is used more than once in your tenancy.

If a Group Name is detected, STEPER will ensure matches are limited to groups based on:

- Alias
- DisplayName
- Name
- PrimarySMTPAddress

If a Group Object ID is detected, STEPER will ensure matches are limited to groups based on:

- ExternalDirectoryObjectID
- GUID
- ID

### DisableNameOrIDChecking

You can turn off the additional STEPER UnifiedGroupName additional matching protection by adding the setting `DisableNameOrIDChecking=True` to `[TeacherUnifiedGroupX]` or `[StudentUnifiedGroupX]`

### Turning Off All Filtered Unified Group Processing

You can turn off all Filtered Unified Group processing for Teachers and Students using the following settings under the `[FunctionControl]` section:

```
TurnOffTeacherFilteredUnifiedGroupFunctionality=True  
TurnOffStudentFilteredUnifiedGroupFunctionality=True
```

See [here](#) for more information.

## TeacherUnifiedGroupX and StudentUnifiedGroupX Examples

Aim	Target / Section	Example Settings Required
Make no changes to Teachers Unified Groups	[TeacherUnifiedGroup1] to [TeacherUnifiedGroup100]	EnableUnifiedGroup=False
Make no changes to Students Unified Groups	[StudentUnifiedGroup1] to [StudentUnifiedGroup100]	EnableUnifiedGroup=False
Add all Teachers that were not in the teachers licensing group to the School Team. Ignore if they were previously actually licensed or not.  Don't check the group for members that should be removed.	[TeacherUnifiedGroup1] to [TeacherUnifiedGroup100]	EnableUnifiedGroup=True UnifiedGroupDescription=Adding All Teachers to School Team UnifiedGroupName=School Team ApplyAllTeachersToGroup=False OnlyApplytoUnlicensedTeachers=False Filter=  RemoveUnMatchedMembers=False
Add all Teachers to the School Team. Ignore if they were previously actually licensed or not.  Don't check the group for members that should be removed.	[TeacherUnifiedGroup1] to [TeacherUnifiedGroup100]	EnableUnifiedGroup=True UnifiedGroupDescription=Adding All Teachers to School Team UnifiedGroupName=School Team ApplyAllTeachersToGroup=True OnlyApplytoUnlicensedTeachers=False Filter=  RemoveUnMatchedMembers=False
To see what user fields are available for filters	[InformationRetrieval]	To Get GetAllUserPolicyInformation=True Or GetUniqueUserDepartmentNames=True
Add all Year 1 Teachers to the Year 1 Team. Make sure that any teachers that are no longer Year 1 Teachers are removed from the group.	[TeacherUnifiedGroup1] to [TeacherUnifiedGroup100]	EnableUnifiedGroup=True UnifiedGroupDescription=Adding Year 1 Teachers to Group UnifiedGroupName=Year 1 ApplyAllTeachersToGroup=True OnlyApplytoUnlicensedTeachers=False Filter=(Department -eq "Year 1")  RemoveUnMatchedMembers=True OnlyRemoveUnMatchedTeachers=True
Add sixth form Students to the Sixth form Group, which has an ID of SixthForm_2378f6aa-0014-44f2-8075-80780803d59c. Make sure that any users that are no longer part of the sixth form are removed from the group.	[StudentUnifiedGroup1] to [StudentUnifiedGroup100]	EnableUnifiedGroup=True UnifiedGroupDescription=Adding Sixth form to Group UnifiedGroupName=SixthForm_2378f6aa-0014-44f2-8075-80780803d59c ApplyAllStudentsToGroup=True OnlyApplytoUnlicensedStudents=False Filter=(Department -like "Year 12*") -or (Department -like "Year 13*")  RemoveUnMatchedMembers=True OnlyRemoveUnMatchedStudents=False





## [TeacherDistributionGroupX] and [StudentDistributionGroupX]

```
[TeacherDistributionGroup1]
;Not part of the standard Stone DFE Config
;STEPPER supports up to 100 different Distribution Groups for Teachers by default
EnableDistributionGroup=False
;The description is for reporting purposes only
DistributionGroupDescription=Teacher Group1 Description
DistributionGroupName=Distribution Group 1
;Teachers are selected for Group Processing based on Group Membership and then a Filter
;Group Membership - by default, only New Teachers are processed (i.e. Teachers that were not in
ApplyAllTeachersToGroup=False
;All Teachers Need to be Licensed at this Point (i.e. added to the licensing group) but you can
OnlyApplytoUnlicensedTeachers=False
;Filter - use this to filter Teachers or leave it blank to include the Teachers selected above
;GroupFilter=(Department -eq "ADMINISTRATION")
Filter=
FilterSource=
;Use this to remove Teachers from the group that no longer match the filter. Note that all exist
RemoveUnMatchedMembers=False
;If removing unmatched members, use the setting below to only remove unmatched Teachers
OnlyRemoveUnMatchedTeachers=True
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=True
DomainFilter=
RequiredOrOUs=
NotRequiredOUs=
OnFailureApplytoAllUsersOnNextRun=
Default=False
CollectionID=
CollectionIDIgnoreThreshold=
CollectionIDsToExclude=
```



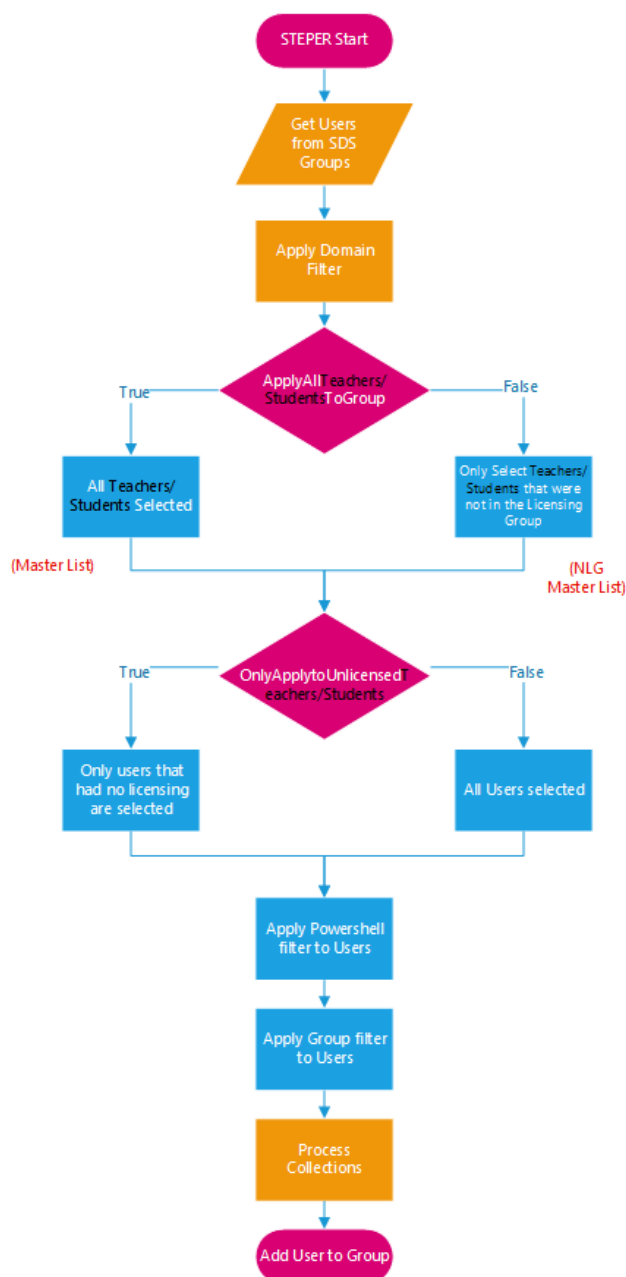
STEPPER supports putting Teachers and Students into Distribution groups. This function allows you to specify a Powershell filter to select Teachers or Students to be put into the group.

[By default](#), up to 100 different sets of Teacher settings and 100 different sets of Student settings are available.

	Teacher	Student
1 <sup>st</sup> Settings	[TeacherDistributionGroup1]	[StudentDistributionGroup1]
...		
Last Settings	[TeacherDistributionGroup100]	[StudentDistributionGroup100]

- Switch on the group of settings by using EnableDistributionGroup=True
- Optionally, put a description into the group of settings. This is for your own reference only.
- STEPER needs to know the name or ID of the Distribution group to add (or remove) users from. Specify this under DistributionGroupName.
- By default, users from the **NLG Teacher/Student Master** list are used. By using the [ApplyAllTeachersToGroup=True](#) or [ApplyAllStudentsToGroup=True](#) option, Teachers/Students from the **Teacher/Student Master** list will be used instead.

- Additionally, as a safety feature, you can specify that only users that were unlicensed at the start of STEPER are processed, by using the `OnlyApplytoUnlicensedTeachers=True` or `OnlyApplytoUnlicensedStudents=True` option. Also see [Refreshing Teacher or Student Licensing Status after Adding Users to Licensing Groups](#)
- You can further select or filter users using the `Filter=` option. This is a Powershell style [attribute filter](#) string that applies to the entire group of settings.
- You can also filter users based on [Group Rules](#) or their [On-Premise Active Directory location](#) (OU), or via a domain name filter [DomainFilter](#).
- With STEPER 1.0.61 and newer, you can group the application of settings into collections. Settings are processed by collection order. Once a user has been processed in a collection, they are not processed in a higher collection number unless the collection number they were processed in is excluded or under the ignore threshold. See [Collections](#).



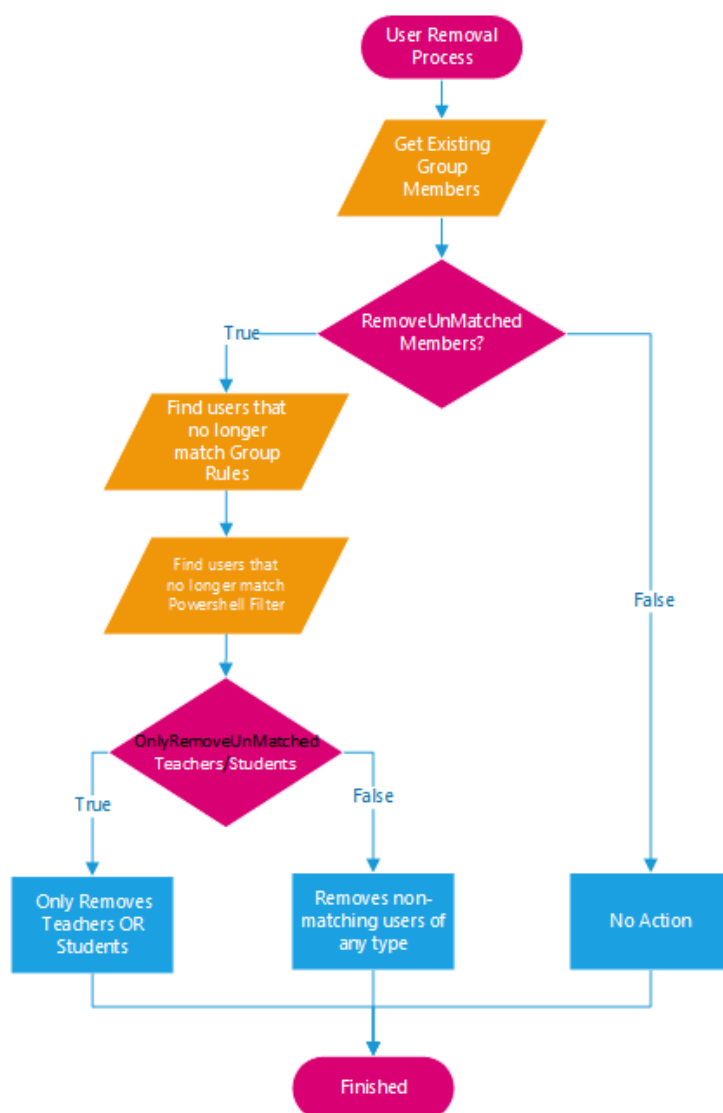
**TIP:** When using Distribution Groups, don't pad the configuration file section group numbering with zeros. For example:

Good	Bad
[TeacherDistributionGroup1]	[TeacherDistributionGroup01]
[TeacherDistributionGroup20]	[TeacherDistributionGroup020]
[TeacherDistributionGroup100]	[TeacherDistributionGroup099]

## Removing Stale Members from Distribution Groups

By default, STEPER will not remove users from the Distribution Group. However, by using the `RemoveUnMatchedMembers=True` option, users that don't match the Group rules, AD/OU Rules and Attribute Filter will be removed from the group. By default, only users of the same type (Teachers/Students) will be removed. Alternatively, use `OnlyRemoveUnMatchedTeachers=False` or `OnlyRemoveUnMatchedStudents=False` to remove other types of users too.

Example 1: Using the settings group [StudentDistributionGroup1], specifying a `Filter=(Department -eq "Year7")` will cause Students matching that filter to be added to the group. With `RemoveUnMatchedMembers=True` option, Students that don't match the filter will be removed from the group. No non-Student users will be removed from the group unless you use the `OnlyRemoveUnMatchedStudents=False` option.



Example 2: Using the settings group [TeacherDistributionGroup1], specifying a `Filter=(Department -eq "Administration")` will cause Teachers matching that filter to be added to the group. With `RemoveUnMatchedMembers=True` option, Teachers that don't match the filter will be removed from the group. No non-Teachers users will be removed from the group unless you use the `OnlyRemoveUnMatchedTeachers=False` option.

**Reminder:** Using the `OnlyRemoveUnMatchedTeachers=False` (under [TeacherDistributionGroupX]) or `OnlyRemoveUnMatchedStudents=False` (under [StudentDistributionGroupX]) has no effect unless you set the corresponding `RemoveUnMatchedMembers=True`



### RemoveUnMatchedMembers Settings

From STEPER 1.0.85, in addition to using RemoveUnMatchedMembers=True, you can now specify specific matching or exclusion reasons for removing a user from the group

Setting	Console Abbreviation	(Internal Boolean Value)	Matching/Exclusion Type
False (default)		0	No users removed from Group
Attribute	A	2	Remove users that don't match the Attribute filter
Group	G	4	Remove users that don't match the Group Rules
OU	O	8	Remove users that don't match the AD OU/CN Rules
Collection	C	16	Remove users Excluded by CollectionID (this means they have already been processed under another Collection unless that previous collection is under the ignore threshold or has been excluded)
1, All or True	AGOC	30	All unmatched users will be removed

Example:

Setting	Result
RemoveUnMatchedMembers=True	All unmatched users will be removed
RemoveUnMatchedMembers=Group,Attribute	Users that don't match the Attribute filter or Group Rules will be removed
RemoveUnMatchedMembers=16	Users Excluded by CollectionID processing



## DistributionGroupName Matching

Exchange Online will select groups based on the supplied Distribution Group Name or Distribution Group ID. Exchange Online does not differentiate between these searches.

By default, STEPER will then ensure that the groups meet the name OR ID criteria as an additional level of protection by detecting if a GUID has been provided under DistributionGroupName. Using an GUID or object ID is useful when the group name is used more than once in your tenancy.

If a Group Name is detected, STEPER will ensure matches are limited to groups based on:

- Alias
- DisplayName
- Name
- PrimarySMTPAddress
- SamAccountName

If a Group Object ID is detected, STEPER will ensure matches are limited to groups based on:

- ExchangeObjectId
- ExternalDirectoryObjectID
- GUID

## DisableNameOrIDChecking

You can turn off this additional STEPER DistributionGroupName additional matching protection by adding the setting `DisableNameOrIDChecking=True` to `[TeacherDistributionGroupX]` or `[StudentDistributionGroupX]`

## Turning Off All Filtered Distribution Group Processing

You can turn off all Filtered Distribution Group processing for Teachers and Students using the following settings under the `[FunctionControl]` section:

```
TurnOffTeacherFilteredDistributionGroupFunctionality=True  
TurnOffStudentFilteredDistributionGroupFunctionality=True
```

See [here](#) for more information.

## TeacherDistributionGroupX and StudentDistributionGroupX Examples

Aim	Target / Section	Example Settings Required
Make no changes to Teachers Distribution Groups	[TeacherDistributionGroup1] to [TeacherDistributionGroup100]	EnableDistributionGroup=False
Make no changes to Students Unified Groups	[StudentDistributionGroup1] to [StudentDistributionGroup100]	EnableDistributionGroup=False
Add all Teachers that were not in the teachers licensing group to the Teacher Distribution Group. Ignore if they were previously actually licensed or not.  Don't check the group for members that should be removed.	[TeacherDistributionGroup1] to [TeacherDistributionGroup100]	EnableDistributionGroup=True DistributionGroupDescription=Adding All Teachers to Group DistributionGroupName=Teacher ApplyAllTeachersToGroup=False OnlyApplytoUnlicensedTeachers=False Filter=  RemoveUnMatchedMembers=False
Add all Teachers to the School Team. Ignore if they were previously actually licensed or not.  Don't check the group for members that should be removed.	[TeacherDistributionGroup1] to [TeacherDistributionGroup100]	EnableDistributionGroup=True DistributionGroupDescription=Adding All Teachers to School Team DistributionGroupName=School Team ApplyAllTeachersToGroup=True OnlyApplytoUnlicensedTeachers=False Filter=  RemoveUnMatchedMembers=False
To see what user fields are available for filters	[InformationRetrieval]	To Get GetAllUserPolicyInformation=True Or GetUniqueUserDepartmentNames=True
Add all Year 1 Teachers to the Year 1 Team. Make sure that any teachers that are no longer Year 1 Teachers are removed from the group.	[TeacherDistributionGroup1] to [TeacherDistributionGroup100]	EnableDistributionGroup=True DistributionGroupDescription=Adding Year 1 Teachers to Group DistributionGroupName=Year 1 ApplyAllTeachersToGroup=True OnlyApplytoUnlicensedTeachers=False Filter=(Department -eq "Year 1")  RemoveUnMatchedMembers=True OnlyRemoveUnMatchedTeachers=True
Add sixth form Students to the Sixth form Group, which has an ID of 2378f6aa-0014-44f2-8075-80780803d59c. Make sure that any users that are no longer part of the sixth form are removed from the group.	[StudentDistributionGroup1] to [StudentDistributionGroup100]	EnableDistributionGroup=True DistributionGroupDescription=Adding Sixth form to Group DistributionGroupName=2378f6aa-0014-44f2-8075-80780803d59c ApplyAllStudentsToGroup=True OnlyApplytoUnlicensedStudents=False Filter=(Department -like "Year 12*") -or (Department -like "Year 13*")  RemoveUnMatchedMembers=True OnlyRemoveUnMatchedStudents=False



## [TeacherTeamsPackagesX] and [StudentTeamsPackagesX]

### [TeacherTeamsPackages1]

```
;Not part of the standard Stone DFE Config
;STEPPER supports up to 100 Teacher Policy Packages by default
ApplyTeamsPackage=False
TeamsPackageDescription=TeacherDesc1
TeamsPackage=Education_Teacher
OnlyChangeTeamsPackageIfBlank=True
;Teachers are selected for Package Processing based on Group Membership and then a Filter
;Group Membership - by default, only New Teachers are processed (i.e. Teachers that were not in
;The User needs to be licensed at this point for this function to work
ApplyTeamsPackagestoAllTeachers=False
;All Teachers Need to be Licensed at this Point (i.e. added to the licensing group) but you can
OnlyApplytoUnlicensedTeachers=False
;Filter - use this to filter Teachers or leave it blank to include the Teachers selected above
;TeamsPackageFilter=(Department -eq "ADMINISTRATION") -or (Department -eq "Systems Services")
Filter=
FilterSource=
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=True
DomainFilter=
RequiredOrOUs=
NotRequiredOUs=
RemoveFailedUsersFromLicensingGroup=
OnFailureApplytoAllUsersOnNextRun=
Default=False
CollectionID=
CollectionIDIgnoreThreshold=
CollectionIDsToExclude=
```





STEPPER supports apply Teams Packages to Teachers and Students. This function allows you to specify a Powershell filter to select Teachers or Students.

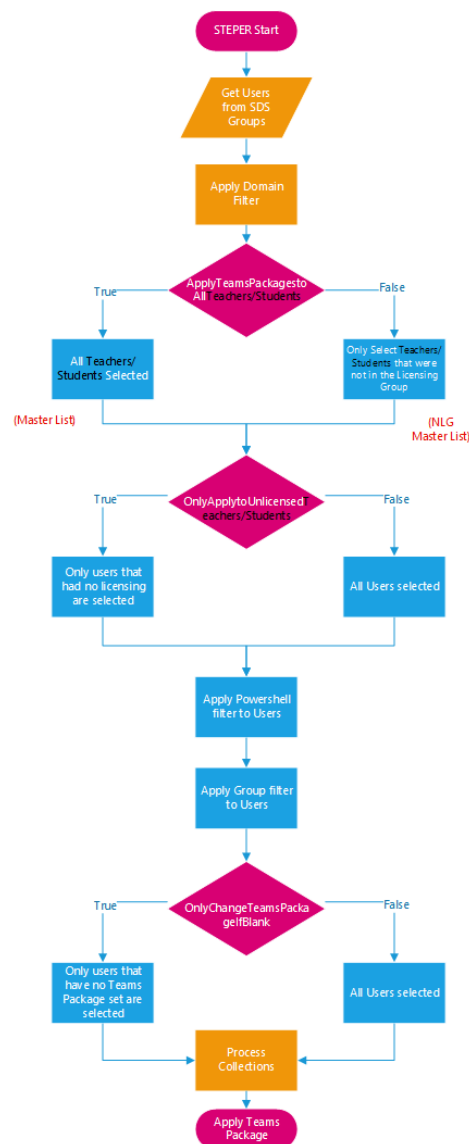
Teams Packages are not part of the Stone standard DFE deployment.

[By default](#), up to 100 different sets of Teacher settings and 100 different sets of Student settings are available.

	Teacher	Student
1 <sup>st</sup> Settings	[TeacherTeamsPackages1]	[StudentTeamsPackages1]
...		
Last Settings	[TeacherTeamsPackages100]	[StudentTeamsPackages100]

- Switch on the group of settings by using `ApplyTeamsPackage=True`
- Optionally, put a description into the group of settings. This is for your own reference only
- STEPPER needs to know the name of the Teams Package to apply. Specify this under `TeamsPackage`
- By default, users from the **NLG Teacher/Student Master** list are used. By using the [ApplyTeamsPackagesToAllTeachers=True](#) or [ApplyTeamsPackagesToAllStudents=True](#) option, Teachers/Students from the **Teacher/Student Master** list will be used instead.

- Additionally, as a safety feature, you can specify that only users that were unlicensed at the start of STEPER are processed, by using the `OnlyApplytoUnlicensedTeachers=True` or `OnlyApplytoUnlicensedStudents=True` option. Also see [Refreshing Teacher or Student Licensing Status after Adding Users to Licensing Groups](#)
- You can specify that users that already have a Teams Package configured do not have it overwritten, using the option `OnlyChangeTeamsPackageIfBlank=True` option.
- You can further select or filter users using the `Filter=` option. This is a Powershell style [attribute filter](#) string that applies to the entire group of settings.
- You can also filter users based on [Group Rules](#) or their [On-Premise Active Directory location](#) (OU), or via a domain name filter [DomainFilter](#).
- With STEPER 1.0.61 and newer, you can group the application of settings into collections. Settings are processed by collection order. Once a user has been processed in a collection, they are not processed in a higher collection number unless the collection number they were processed in is excluded or under the ignore threshold. See [Collections](#).



Note that the users Teams Package setting will not be changed if the package doesn't exist.

When STEPER gets a user's online properties through M365, this does not include the assigned Teams package. This has to be determined with an additional call per user, adding some processing time. For each Teams package, STEPER will display the time left remaining firstly in finding users, and then the time left in applying changes.

**TIP:** When using Teams Packages, don't pad the configuration file section group numbering with zeros. For example:

Good	Bad
[TeacherTeamsPackages1]	[TeacherTeamsPackages01]
[TeacherTeamsPackages20]	[TeacherTeamsPackages020]
[TeacherTeamsPackages100]	[TeacherTeamsPackages099]



## Package Batch Processing Thresholds

```
[Misc]
;The setting below will introduce a pause after applying licensing groups, if any licensing group changes were made
SleepSecondsAfterApplyingLicensing=600
;And/or - use the setting below to introduce a pause before Teams Policies are applied
SleepSecondsBeforeApplyingPolicies=
DontPauseForChanges=False
;Set the threshold to 0 to always use Batch Policy Processing
TeacherTeamsPackageBatchProcessingThreshold=99999
StudentTeamsPackageBatchProcessingThreshold=99999
TeacherTeamsPolicyBatchProcessingThreshold=99999
StudentTeamsPolicyBatchProcessingThreshold=99999
RemediateSkypeAccessTokenTimeout=True
;Prevent Modifying Azure AD Policies when Azure AD Token Lifetime Policies already exist
ApplySkypeAccessTokenTimeoutRemediationWhenPoliciesAlreadyExist=False
ThrowWarningInsteadofErrorOnRemediationFailure=False
DontCheckOnMicrosoftCOMPrefixWhenRemovingRemediation=False
CheckforUpdates=True
DisableAutoUpdate=False
Verbose=True
```

Packages can be applied to users directly (user by user) or by submitting a package batch job.

The [Misc]TeacherTeamsPackageBatchProcessingThreshold and [Misc]StudentTeamsPackageBatchProcessingThreshold allow you to control when packages are applied to users directly, or when a batch job is used.

- If the number of users to be modified is less than the threshold, direct assignment is used. This will take some time, depending on M365 performance.
- If the number of users to be modified is the same or more than the threshold, a batch job will be submitted. This is quickly submitted, but the users are only modified in the background by M365.

**Recommendation:** Set this to a high number such as 99999 to ensure that changes are always applied as quickly as possible

## Disabled Users

To assign a Teams Package in M365, the user should be Enabled in Teams. If the user is Disabled, you cannot apply packages to the user.

Before attempting to assign a Teams Package, STEPER will check that the user is now Enabled. If they are not, the user will not be processed.

You can skip this check by specifying the following in the package settings:

```
ProcessDisabledUsers=True
```

With this setting, when a disabled user is found, STEPER will attempt to process them anyway, which may generate an error message.



## License Checking

To assign a Teams Package in M365, the user should be licensed. In a normal STEPER deployment, you will have licensed the users either through the licensing group, or via an alternative method.

Before attempting to assign a Teams Package, STEPER will check that the user is now licensed. If they are not, the user will not be processed.

You can skip this check by specifying the following in the package settings:

```
SkipUpdatedLicensingCheck=True
```

With this setting, when an unlicensed user is found, STEPER will attempt to process them anyway, which may generate an error message.

## Dealing with Problems when Applying Packages

STEPER can remediate users that can't have packages applied (e.g. because they are disabled or not licensed) or where packages fail to apply (e.g. because of a Microsoft problem) to ensure that they are processed on the next STEPER run. See [User Remediation](#).

## Turning Off All Package Processing

You can turn off all Package processing for Teachers and Students using the following settings under the [FunctionControl] section:

```
TurnOffTeacherPackageFunctionality=True  
TurnOffStudentPackageFunctionality =True
```

See [here](#) for more information.

## TeacherTeamsPackagesX and StudentTeamsPackagesX Examples

Aim	Target / Section	Example Settings Required
Make no changes to Teachers Teams Packages	[TeacherTeamsPackages1] to [TeacherTeamsPackages100]	ApplyTeamsPackage=False
Make no changes to Students Teams Packages	[StudentTeamsPackages1] to [StudentTeamsPackages8]	ApplyTeamsPackage=False
Give all new Year 12 Students the Education_HigherEducationStudent Teams Package. Ignore previous licensing status	[StudentTeamsPackages1] to [StudentTeamsPackages100]	ApplyTeamsPackage=True TeamsPackageDescription=Applying HE Package TeamsPackage=Education_HigherEducationStudent OnlyChangeTeamsPackageIfBlank=False ApplyTeamsPackagesToAllStudents=False OnlyApplytoUnlicensedStudents=False Filter=(Department -like "Year 12*")
Give all Year 12 and Year 13 Students the Education_HigherEducationStudent Teams Package, but only if they were previously unlicensed	[StudentTeamsPackages1] to [StudentTeamsPackages100]	ApplyTeamsPackage=True TeamsPackageDescription=Applying HE Package TeamsPackage=Education_HigherEducationStudent OnlyChangeTeamsPackageIfBlank=False ApplyTeamsPackagesToAllStudents=True OnlyApplytoUnlicensedStudents=True Filter=(Department -like "Year 12*" -or Department -like "Year 13*")
Give all Year 12 Students the Education_HigherEducationStudent Teams Package, but only if they have no package assigned already. Ignore previous licensing status	[StudentTeamsPackages1] to [StudentTeamsPackages100]	ApplyTeamsPackage=True TeamsPackageDescription=Applying HE Package TeamsPackage=Education_HigherEducationStudent OnlyChangeTeamsPackageIfBlank=True ApplyTeamsPackagesToAllStudents=True OnlyApplytoUnlicensedStudents=False Filter=(Department -like "Year 12*")
To see what user fields are available for filters	[InformationRetrieval]	To Get GetAllUserPolicyInformation=True Or GetUniqueUserDepartmentNames=True



## [TeacherTeamsPoliciesX] and [StudentTeamsPoliciesX]

```
[TeacherTeamsPolicies1]
;STEPPER supports up to 100 different sets of Teacher Policies by default
ApplyTeamsPolicies=False
;The description is for reporting purposes only
TeamsPoliciesDescription=Standard Stone DFE Teacher Policies
;Leave the Policy Names below blank if you dont want them changed
TeamsMessagingPolicy=Stone_EduTeacher
OnlyChangeTeamsMessagingPolicyIfBlank=True
TeamsMeetingPolicy=Stone_EduTeacher
OnlyChangeTeamsMeetingPolicyIfBlank=True
TeamsAppSetupPolicy=Stone_EduTeacher
OnlyChangeTeamsAppSetupPolicyIfBlank=True
TeamsCallingPolicy=Stone_EduTeacher
OnlyChangeTeamsCallingPolicyIfBlank=True
TeamsMeetingBroadcastPolicy=Stone_EduTeacher
OnlyChangeTeamsMeetingBroadcastPolicyIfBlank=True
TeamsChannelsPolicy=Stone_EduTeacher
OnlyChangeTeamsChannelsPolicyIfBlank=True
;The Policy Below is not part the standard DFE Rollout
TeamsAppPermissionPolicy=
OnlyChangeTeamsAppPermissionPolicyIfBlank=True
;The Policy Below is not part the standard DFE Rollout
TeamsFeedbackPolicy=
OnlyChangeTeamsFeedbackPolicyIfBlank=True
;Teachers are selected for Policy Processing based on Group Membership and then a Filter
;Group Membership - by default, only New Teachers are processed (i.e. Teachers that were not in
;The User needs to be licensed at this point for this function to work
ApplyTeamsPoliciesToAllTeachers=False
;All Teachers Need to be Licensed at this Point (i.e. added to the licensing group) but you can
OnlyApplytoUnlicensedTeachers=False
;Filter - use this to filter Teachers or leave it blank to include the Teachers selected above
;TeamsPolicyFilter=(Department -eq "ADMINISTRATION") -or (Department -eq "Systems Services")
Filter=
FilterSource=
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=True
DomainFilter=
RequiredOrOUs=
NotRequiredOUs=
RemoveFailedUsersFromLicensingGroup=
OnFailureApplytoAllUsersOnNextRun=
Default=False
CollectionID=
CollectionIDIgnoreThreshold=
CollectionIDsToExclude=
```



STEPPER supports apply Teams Policies to Teachers and Students. This function allows you to specify a Powershell filter to select Teachers or Students. [By default](#), up to 100 different sets of Teacher settings and 100 different sets of Student settings are available.

	Teacher	Student
1 <sup>st</sup> Settings	[TeacherTeamsPolicies1]	[StudentTeamsPolicies1]
...		
Last Settings	[TeacherTeamsPolicies100]	[StudentTeamsPolicies100]

- Switch on the group of settings by using `ApplyTeamsPolicies=True`
- Optionally, put a description into the group of settings. This is for your own reference only

The following Teams Policies can be controlled by STEPPER:

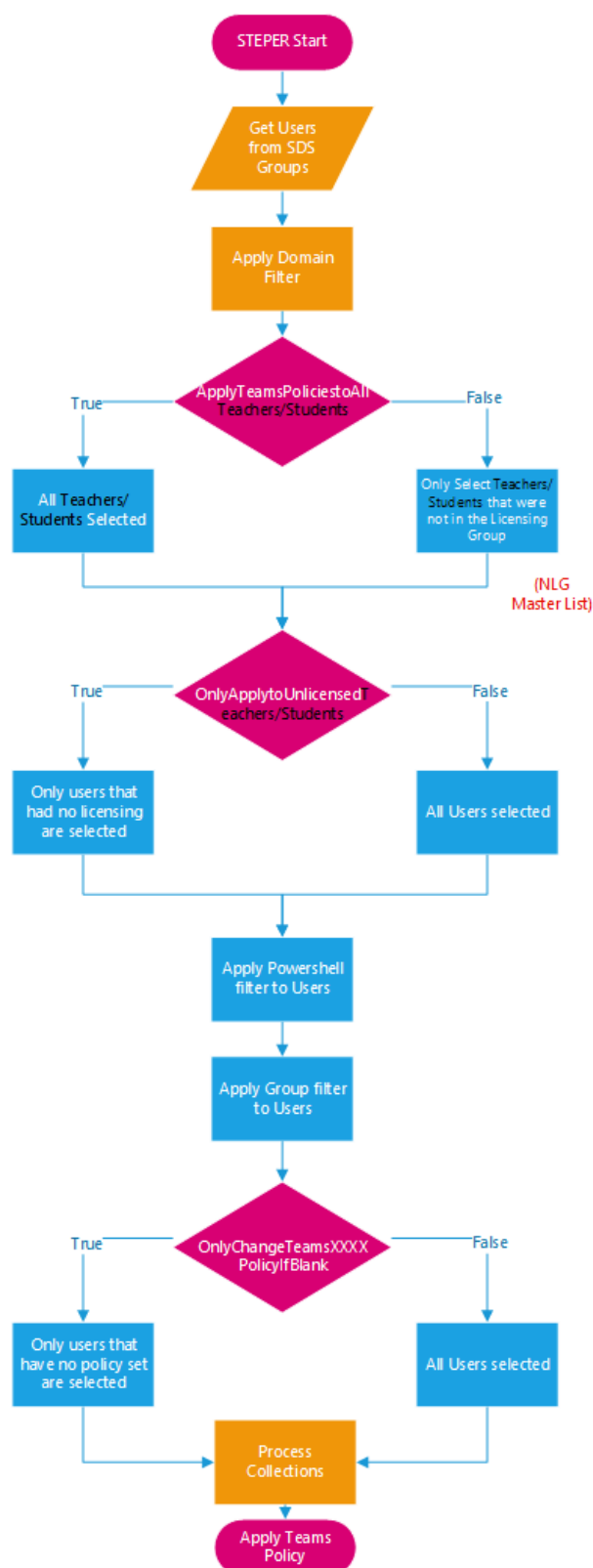
TeamsMessagingPolicy	TeamsMeetingPolicy	TeamsAppSetupPolicy
TeamsCallingPolicy	TeamsMeetingBroadcastPolicy	TeamsChannelsPolicy
TeamsAppPermissionPolicy	TeamsFeedbackPolicy	

The following policy name defaults are part of the standard Stone DFE deployment:

- Primary School Students - Stone\_EduPrimary
- Secondary School Students - Stone\_EduSecondary
- Sixth Form Students - Stone\_EduSixth
- Teachers - Stone\_EduTeacher

By default, users from the **NLG Teacher/Student Master list** are used. By using the [ApplyTeamsPoliciestoAllTeachers=True](#) or [ApplyTeamsPoliciestoAllStudents=True](#) option, Teachers/Students from the **Teacher/Student Master list** will be used instead.

- Additionally, as a safety feature, you can specify that only users that were unlicensed at the start of STEPER are processed, by using the `OnlyApplytoUnlicensedTeachers=True` or `OnlyApplytoUnlicensedStudents=True` option. Also see [Refreshing Teacher or Student Licensing Status after Adding Users to Licensing Groups](#)
- For each of the 8 supported policies, you need to specify the policy name if you want it enforced. If you leave the policy name blank, users will not receive the setting. For each policy, you can specify that users that already have that policy configured should not have it overwritten.
- You can further select or filter users using the `Filter=` option. This is a Powershell style [attribute filter](#) string that applies to the entire group of settings.
- You can also filter users based on [Group Rules](#) or their [On-Premise Active Directory location](#) (OU), or via a domain name filter [DomainFilter](#).
- With STEPER 1.0.61 and newer, you can group the application of settings into collections. Settings are processed by collection order. Once a user has been processed in a collection, they are not processed in a higher collection number unless the collection number they were processed in is excluded or under the ignore threshold. See [Collections](#).







Example:

Setting TeamsAppSetupPolicy=Stone\_EduTeacher and OnlyChangeTeamsAppSetupPolicyIfBlank=True will cause users that match the licensing selections and Powershell filter to be configured to use the Stone\_EduTeacher TeamsAppSetupPolicy, but only if their TeamsAppSetupPolicy is currently not configured, or blank.

**TIP:** When using Teams Policies, don't pad the configuration file section group numbering with zeros. For example:

Good	Bad
[TeacherTeamsPolicies1]	[TeacherTeamsPolicies01]
[TeacherTeamsPolicies20]	[TeacherTeamsPolicies020]
[TeacherTeamsPolicies100]	[TeacherTeamsPolicies099]



## Policy Batch Processing Thresholds

```
[Misc]  
.....  
TeacherTeamsPolicyBatchProcessingThreshold=20  
StudentTeamsPolicyBatchProcessingThreshold=20
```

Policy can be applied to users directly (user by user) or by submitting a policy batch job.

The [Misc]TeacherTeamsPolicyBatchProcessingThreshold and [Misc]StudentTeamsPolicyBatchProcessingThreshold allow you to control when policies are applied to users directly, or when a batch job is used.

- If the number of users to be modified is less than the threshold, direct assignment is used. This will take some time, depending on M365 performance.
- If the number of users to be modified is the same or more than the threshold, a batch job will be submitted. This is quickly submitted, but the users are only modified in the background by M365.

If 10 users need all 8 supported policies to be changed, this is 8 requests of 10 users. By default, the policy batch processing threshold of 20 users will mean that STEPER will apply all of the policies to each user directly.

**TIP:** Use [InformationRetrieval][GetPolicyJobs](#)=True to see the list of ongoing batch policy jobs

## Disabled Users

To assign a Teams Policy in M365, the user should be Enabled in Teams. If the user is Disabled, you cannot apply policies to the user.

Before attempting to assign a Teams Policy, STEPER will check that the user is now Enabled. If they are not, the user will not be processed.

You can skip this check by specifying the following in the policy settings:  
ProcessDisabledUsers=True

With this setting, when a disabled user is found, STEPER will attempt to process them anyway, which may generate an error message.

## License Checking

To assign a Teams Policy in M365, the user should be licensed. In a normal STEPER deployment, you will have licensed the users either through the licensing group, or via an alternative method.



Before attempting to assign a Teams Policy, STEPER will check that the user is now licensed. If they are not, the user will not be processed.

You can skip this check by specifying the following in the policy settings:

```
SkipUpdatedLicensingCheck=True
```

With this setting, when an unlicensed user is found, STEPER will attempt to process them anyway, which may generate an error message.

### Dealing with Problems when Applying Policies

STEPPER can remediate users that can't have policies applied (e.g. because they are disabled or not licensed) or where policies fail to apply (e.g. because of a Microsoft problem) to ensure that they are processed on the next STEPPER run. See [User Remediation](#).

### Turning Off All Policy Processing

You can turn off all Policy processing for Teachers and Students using the following settings under the [FunctionControl] section:

```
TurnOffTeacherPolicyFunctionality=True  
TurnOffStudentPolicyFunctionality =True
```

See [here](#) for more information.



## TeacherTeamsPoliciesX and StudentTeamsPoliciesX Examples

Aim	Target / Section	Example Settings Required
Make no changes to Teachers Teams Packages	[TeacherTeamsPolicies1] to [TeacherTeamsPolicies100]	ApplyTeamsPolicies=False
Make no changes to Students Teams Packages	[StudentTeamsPolicies1] to [StudentTeamsPolicies100]	ApplyTeamsPolicies=False
Give all new Secondary School Students the standard Stone DFE Policies. Ignore previous licensing status	[StudentTeamsPolicies1] to [StudentTeamsPolicies100]	<p>ApplyTeamsPolicies=True  TeamsPoliciesDescription=Standard Stone DFE Secondary Policies  TeamsMessagingPolicy=Stone_EduSecondary  OnlyChangeTeamsMessagingPolicyIfBlank=False  TeamsMeetingPolicy=Stone_EduSecondary  OnlyChangeTeamsMeetingPolicyIfBlank=False  TeamsAppSetupPolicy=Stone_EduSecondary  OnlyChangeTeamsAppSetupPolicyIfBlank=False  TeamsCallingPolicy=Stone_EduSecondary  OnlyChangeTeamsCallingPolicyIfBlank=False  TeamsMeetingBroadcastPolicy=Stone_EduSecondary  OnlyChangeTeamsMeetingBroadcastPolicyIfBlank=False  TeamsChannelsPolicy=Stone_EduSecondary  OnlyChangeTeamsChannelsPolicyIfBlank=False</p> <p>ApplyTeamsPoliciesToAllStudents=False  OnlyApplyToUnlicensedStudents=False  Filter=(Department -eq "Year 07") -or (Department -eq "Year 08") -or (Department -eq "Year 09") -or (Department -eq "Year 10") -or (Department -eq "Year 11")</p>
Give all Secondary School Students the standard Stone DFE Policies, but only if they don't have a policy set. Ignore previous licensing status	[StudentTeamsPolicies1] to [StudentTeamsPolicies100]	<p>ApplyTeamsPolicies=True  TeamsPoliciesDescription=Standard Stone DFE Secondary Policies  TeamsMessagingPolicy=Stone_EduSecondary  OnlyChangeTeamsMessagingPolicyIfBlank=True  TeamsMeetingPolicy=Stone_EduSecondary  OnlyChangeTeamsMeetingPolicyIfBlank=True  TeamsAppSetupPolicy=Stone_EduSecondary  OnlyChangeTeamsAppSetupPolicyIfBlank=True  TeamsCallingPolicy=Stone_EduSecondary  OnlyChangeTeamsCallingPolicyIfBlank=True  TeamsMeetingBroadcastPolicy=Stone_EduSecondary  OnlyChangeTeamsMeetingBroadcastPolicyIfBlank=True  TeamsChannelsPolicy=Stone_EduSecondary  OnlyChangeTeamsChannelsPolicyIfBlank=True</p> <p>ApplyTeamsPoliciesToAllStudents=True  OnlyApplyToUnlicensedStudents=False  Filter=(Department -eq "Year 07") -or (Department -eq "Year 08") -or (Department -eq "Year 09") -or (Department -eq "Year 10") -or (Department -eq "Year 11")</p>
Give all new Primary School Students the standard Stone DFE Policies, but only the students you haven't already manually licensed and processed	[StudentTeamsPolicies1] to [StudentTeamsPolicies100]	<p>ApplyTeamsPolicies=True  TeamsPoliciesDescription=Standard Stone DFE Secondary Policies  TeamsMessagingPolicy=Stone_EduPrimary  OnlyChangeTeamsMessagingPolicyIfBlank=False  TeamsMeetingPolicy=Stone_EduPrimary  OnlyChangeTeamsMeetingPolicyIfBlank=False  TeamsAppSetupPolicy=Stone_EduPrimary  OnlyChangeTeamsAppSetupPolicyIfBlank=False  TeamsCallingPolicy=Stone_EduPrimary  OnlyChangeTeamsCallingPolicyIfBlank=False  TeamsMeetingBroadcastPolicy=Stone_EduPrimary  OnlyChangeTeamsMeetingBroadcastPolicyIfBlank=False  TeamsChannelsPolicy=Stone_EduPrimary  OnlyChangeTeamsChannelsPolicyIfBlank=False</p> <p>ApplyTeamsPoliciesToAllStudents=True  OnlyApplyToUnlicensedStudents=True  Filter=(Department -eq "Year 01") -or (Department -eq "Year 02") -or (Department -eq "Year 03") -or (Department -eq "Year 04") -or (Department -eq "Year 05") -or (Department -eq "Year 06")</p>
To see what user fields are available for filters	[InformationRetrieval]	<p>To Get  GetAllUserPolicyInformation=True  Or  GetUniqueUserDepartmentNames=True</p>



## Applying changes to All Users / Teachers / Students [Override]

Many functions have a setting that allows you to specify whether or not all users are worked on, or alternatively just users that were not in the licensing group.

Function	Supports Applying to All Teachers / All Students, or just those that were not in the Licensing Group	Can be overridden to Apply to All Teachers / All Students using the global [Override] setting or command line
GlobalAdminTeamsCreation	N/A	No
Licensing Groups	N/A	
AllTeachersGroup / AllStudentsGroup	N/A	
Set Student Department	Yes	Yes
TeacherSecurityGroup / StudentSecurityGroup	Yes	
TeacherUnifiedGroup / StudentUnifiedGroup	Yes	
TeacherDistributionGroup / StudentDistributionGroup	Yes	
TeacherTeamsPackages / StudentTeamsPackages	Yes	
TeacherTeamsPolicies / StudentTeamsPolicies	Yes	

If your user estate has inconsistent settings, or if you want to make sure all users are checked, modified and up-to-date, then use the Override below.

### How to Override which Users Are Modified

This can be done either via the command line or using the [Override] section in the configuration file.

#### Command Line

Switch	Effect
-ApplytoAll	Overrides the setting for Teachers and Students
-ApplytoAllTeachers	Overrides the setting for Teachers
-ApplytoAllStudents	Overrides the setting for Students

#### Configuration File

```
[Override]
GlobalOnFailureApplytoAllUsersOnNextRun=True
;The settings below only take effect if set to True
ApplytoAllTeachers=False
ApplytoAllStudents=False
```

[Override]	Effect
ApplytoAllTeachers=True	Overrides the setting for Teachers
ApplytoAllStudents=True	Overrides the setting for Students

Turning on any of these settings overrides the per function setting in the configuration file. Note that settings ApplytoAllTeachers=False has no effect, and ApplytoAllStudents=False also has no effect.

**TIP:** Using this override, either via the command line, or by the configuration file, may significantly increase processing time, depending on the number of users you have, so don't leave it enabled unless you need to.



**Example 1:**

```
[TeacherTeamsPolicies1]  
ApplyTeamsPoliciesToAllTeachers=False
```

This by default would only apply changes to Teachers that were not in the licensing group. By specifying `-ApplytoAllTeachers` on the command line, or by setting `[Override]ApplytoAllTeachers=True`, this all Teachers will be processed including those already in the licensing group.

**Example 2:**

```
[TeacherTeamsPolicies1]  
ApplyTeamsPoliciesToAllTeachers=True
```

In this situation, the override setting will have no effect, even if you set `[Override]ApplytoAllTeachers=False`, as the override setting only works when set it to True. In this instance, the function specific setting is already set tot True.

**Example 3:**

```
[StudentTeamsPackages1]  
ApplyTeamsPackagestoAllStudents=True
```

This by default would only apply changes to Students that were not in the licensing group. By using the command line below, All Students (and Teachers) would be processed including those already in the licensing group.

```
.\STEPPER.PS1 -ApplyToAll
```

**NOTE:** The [User Problem Re-Processing](#) feature for [user problem resolution](#) makes use of the `Override / Apply to All Functionality`

## [Information Retrieval]

```
[InformationRetrieval]
;Result below recorded in CSV file based on the Log File Location
GetAllUserPolicyInformation=False
;Note that users that are not licensed will appear to have been filtered out
Filter=
FilterSource=
GetUserPackageInformation=False
GetUniqueUserDepartmentNames=True
;Note that GetPolicyJobs is Tenant wide
GetPolicyJobs=True
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=True
DomainFilter=
RequiredOrOUs=
NotRequiredOUs=
```

### GetAllUserPolicyInformation

This gets all user policy and package information and is performed after adding users to the licensing groups.

- All users in the tenancy will be reported on
- The SourceGroups filters for Students and Teachers will be used to show users that match them
- An additional filter [Filter](#) is available for testing purposes. You can change the attribute filter source too using [FilterSource](#).
- You can also test [Group Rules filters](#) or [On-Premise Active Directory location](#) (OU) matching, or a domain name filter [DomainFilter](#).
- The 8 supported Teams Policies will be displayed for all users
- The assigned Teams Policy Package will be displayed for all users

This information is exported to a CSV file in the log file directory and included in the results email.

GetAllUserPolicyInformation processing takes some time to do, so it is unlikely you will include this function in day-to-day processing.



### **GetUserPackageInformation**

From version 1.0.91 onwards, STEPER no longer gets User Policy Package information by default, due to the amount of time that it takes to get this information from the online platform.

If you still need the User Package information, set GetUserPackageInformation=True

### **GetUniqueUserDepartmentNames**

Unique user department names are stored in the log file. This information is useful in building Powershell filters to apply functions to users.

### **GetPolicyJobs**

This shows outstanding TeamsPolicy batch jobs that are outstanding across the tenancy. It does not show Package batch jobs.



## Powershell Filters

### Attribute Filters

Many functions support the use of Powershell filters, to allow you to select particular users based on their attributes or settings.

Function	Supports Attribute Filters	Supports Group Rules	Supports AD OU/CN Rules	Supports Collections	Supports Domain Filters
SourceGroups	No	Yes	No	No	Yes
GlobalAdminTeamsCreation	No	No	No	No	No
Licensing Groups	No	No	No	No	No
AllTeachersGroup / AllStudentsGroup	No	No	No	No	No
Set Student Department	No	No	No	No	No
Information Retrieval	Yes	Yes	Yes	No	Yes
TeacherSecurityGroup / StudentSecurityGroup	Yes	Yes	Yes	Yes	Yes
TeacherUnifiedGroup / StudentUnifiedGroup	Yes	Yes	Yes	Yes	Yes
TeacherDistributionGroup / StudentDistributionGroup	Yes	Yes	Yes	Yes	Yes
TeacherTeamsPackages / StudentTeamsPackages	Yes	Yes	Yes	Yes	Yes
TeacherTeamsPolicies / StudentTeamsPolicies	Yes	Yes	Yes	Yes	Yes

```
[StudentTeamsPolicies3]
...
Filter=
FilterSource=
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=False
RequiredOrOUs=
NotRequiredOUs=
RemoveFailedUsersFromLicensingGroup=
OnFailureApplytoAllUsersOnNextRun=
Default=False
CollectionID=
CollectionIDIgnoreThreshold=
CollectionIDsToExclude=
```

v



## Remote and Local Filters

In STEPER 1.0.60 and earlier, STEPER only supported remote filters. The filter string was passed to the Microsoft platform and only the filtered results returned.

As of March 2022, Microsoft have reduced the fields available for filters, and the syntax is also being reduced.

So, from STEPER 1.0.61, you can specify that results are filtered locally instead. This may provide a more consistent filtering experience, however the attributes for filtering against still need to be available in the data. The downside is that more data has to be passed from the Microsoft platform to STEPER, which may slow down processing.

The default continues to be Remote filtering.

If you use Local filtering, you can also [change the attribute source from Team to AzureAD](#).

## UseLocalAttributeFiltering

To enable local filtering, in the [Misc] Section, enable UseLocalAttributeFiltering=True  
This is a global setting and applies to all functions that use Attribute filters.

If you do not specify this setting, the default of remote filtering is used.

## UseAzureADforLocalAttributeFilteringByDefault

STEPPER 1.0.60 and earlier only used user information from Microsoft Teams to filter user attributes.

From STEPPER 1.0.61, you can specify that results are filtered using AzureAD (get-azureaduser) instead of Teams (get-csonlineuser) locally instead. The fields and information available between the two vary.

The default continues to be to use Teams / CSONline as the attribute source.

To use AzureAD instead, you must enable Local Attribute Filtering in [Misc]:  
UseLocalAttributeFiltering=True

And then either:

- Set UseAzureADforLocalAttributeFilteringByDefault=True in [Misc]
- Or, in each function that supports attribute filtering, set FilterSource=AzureAD

If you set UseAzureADforLocalAttributeFilteringByDefault=True, you can make individual functions use CSONline by setting FilterSource=CSONline



## Local and Remote Filtering and Sources

Filtering Source	Supports Local Filtering	Supports Remote Filtering
CSONline (default)	Yes	Yes (default for CSONline)
AzureAD	Yes (the only option for AzureAD)	No

Using AzureAD as the Attribute filter is limited to Local filtering only.

To use AzureAD filtering:

- Set UseLocalAttributeFilterings=True in [Misc]
- Set UseAzureADforLocalAttributeFilteringByDefault=True, or inside individual functions set FilterSource=AzureAD
- Update all of your filters to the [correct syntax](#).

[Misc]

```
UseLocalAttributeFiltering=True  
UseAzureADforLocalAttributeFilteringByDefault=False
```



## Filter Formatting or Syntax

Remote and Local filters have different syntax.

### Example Remote Filters:

- `Filter=(Department -eq "ADMINISTRATION") -or (Department -eq "Systems Services")`
- `(Department -eq "Year 01") -or (Department -eq "Year 02") -or (Department -eq "Year 03") -or (Department -eq "Year 04") -or (Department -eq "Year 05") -or (Department -eq "Year 06")`

Notice that online filters should not have the attribute name preceded by an `$_`. Using any Remote Filters that contain `$_` will cause an error.

### Example Local Filters: (UseLocalAttributeFiltering=True)

- `Filter=($_.Department -eq "ADMINISTRATION") -or ($_.Department -eq "Systems Services")`
- `($.Department -eq "Year 01") -or ($.Department -eq "Year 02") -or ($.Department -eq "Year 03") -or ($.Department -eq "Year 04") -or ($.Department -eq "Year 05") -or ($.Department -eq "Year 06")`

Notice that local filters need to have the attribute name preceded by an `$_`. Using any Local Filters that don't contain `$_` will cause an error.

**TIP:** Each of the supported functions also includes a separate field for optionally selecting users that were unlicensed when STEPER began, e.g. `ApplyAllTeachersToGroup=False`, as well as current licensing status e.g. `OnlyApplytoUnlicensedTeachers=False`, as well as [group based filtering](#)

## Field Examples

CSONline	AzureAD
Department	Department
Description	Description
Office	PhysicalDeliveryOfficeName
State	State

## Available Attribute Filter Fields

### *CSONline*

**NOTE:** As of March 2022, Microsoft have reduced the fields available for online filters, and the syntax is also being reduced. For more information on online filter functionality, see “-Filter” [here](#).

- Available fields for remote filters include:
- accountEnabled
- AssignedPlan
- CountryAbbreviation
- department
- displayName
- EnterpriseVoiceEnabled
- ExternalAccessPolicy
- givenName
- Identity
- lineUri
- OnlineAudioConferencingRoutingPolicy
- OnlineDialOutPolicy
- OnlineVoicemailPolicy
- OnlineVoiceRoutingPolicy
- ownerUrn
- TeamsAppPermissionPolicy
- TeamsAppSetupPolicy
- TeamsAudioConferencingPolicy
- TeamsCallHoldPolicy
- TeamsCallingPolicy
- TeamsCallParkPolicy
- TeamsChannelsPolicy
- TeamsComplianceRecordingPolicy
- TeamsCortanaPolicy
- TeamsEducationAssignmentsAppPolicy
- TeamsEmergencyCallingPolicy
- TeamsEmergencyCallRoutingPolicy
- TeamsFeedbackPolicy
- TeamsIPPhonePolicy
- TeamsMeetingBrandingPolicy
- TeamsMeetingBroadcastPolicy
- TeamsMeetingPolicy
- TeamsMessagingPolicy
- TeamsMobilityPolicy
- TeamsNotificationAndFeedsPolicy
- TeamsShiftsAppPolicy
- TeamsShiftsPolicy
- TeamsSurvivableBranchAppliancePolicy
- TeamsSyntheticAutomatedCallPolicy
- TeamsTargetingPolicy
- TeamsTemplatePermissionPolicy
- TeamsUpdateManagementPolicy
- TeamsUpgradeOverridePolicy
- TeamsUpgradePolicy
- TeamsVdiPolicy
- TeamsVerticalPackagePolicy
- TeamsVideoInteropServicePolicy
- TenantDialPlan
- Title
- UsageLocation
- UserDirSyncEnabled
- userPrincipalName

## Group Rules

From STEPER 1.0.21, you can now specify some basic Group Rules on all of the functions that support Attribute Filters. This is useful in some situations like RM Unify where the user attributes may not be available in the cloud.

Function	Supports Attribute Filters	Supports Group Rules	Supports AD OU/CN Rules	Supports Collections	Supports Domain Filters
SourceGroups	No	Yes	No	No	Yes
GlobalAdminTeamsCreation	No	No	No	No	No
Licensing Groups	No	No	No	No	No
AllTeachersGroup / AllStudentsGroup	No	No	No	No	No
Set Student Department	No	No	No	No	No
Information Retrieval	Yes	Yes	Yes	No	Yes
TeacherSecurityGroup / StudentSecurityGroup	Yes	Yes	Yes	Yes	Yes
TeacherUnifiedGroup / StudentUnifiedGroup	Yes	Yes	Yes	Yes	Yes
TeacherDistributionGroup / StudentDistributionGroup	Yes	Yes	Yes	Yes	Yes
TeacherTeamsPackages / StudentTeamsPackages	Yes	Yes	Yes	Yes	Yes
TeacherTeamsPolicies / StudentTeamsPolicies	Yes	Yes	Yes	Yes	Yes

```
[StudentTeamsPolicies3]
```

```
...
```

```
Filter=
```

```
FilterSource=
```

```
RequiredGroups=
```

```
RequiredRecursiveGroups=
```

```
RequiredOrGroups=
```

```
RequiredOrRecursiveGroups=
```

```
NotRequiredGroups=
```

```
NotRequiredRecursiveGroups=
```

```
DisableNameorIDCheckingForGroupRules=False
```

```
DontAllowMissingNotRequiredGroups=False
```

```
IncludeDistributionGroupsInGroupRules=True
```

```
RequiredOrOUs=
```

```
NotRequiredOUs=
```

```
RemoveFailedUsersFromLicensingGroup=
```

```
OnFailureApplytoAllUsersOnNextRun=
```

```
Default=False
```

```
CollectionID=
```

```
CollectionIDIgnoreThreshold=
```

```
CollectionIDsToExclude=
```



## Settings available

```
RequiredGroups=  
RequiredRecursiveGroups=  
RequiredOrGroups=  
RequiredOrRecursiveGroups=  
NotRequiredGroups=  
NotRequiredRecursiveGroups=  
DisableNameorIDCheckingForGroupRules=  
DontAllowMissingNotRequiredGroups=  
IncludeDistributionGroupsInGroupRules=
```

**NOTE:** The Group rules settings available for [TeacherSourceGroups] and [StudentSourceGroups] are very similar, with some small changes. See Source Group Processing [here](#).

## Example

For example, you may want to differentiate secondary school and sixth form students.

```
[StudentTeamsPolicies2]  
;STEPPER supports up to 100 different sets of Student Policies  
ApplyTeamsPolicies=False  
;The description is for reporting purposes only  
TeamsPoliciesDescription=Standard Stone DFE Secondary Policies  
;Leave the Policy Names below blank if you dont want them changed  
TeamsMessagingPolicy=Stone_EduSecondary  
...  
Filter=  
NotRequiredGroups=Sixth Form
```

```
[StudentTeamsPolicies3]  
;STEPPER supports up to 100 different sets of Student Policies  
ApplyTeamsPolicies=False  
;The description is for reporting purposes only  
TeamsPoliciesDescription=Standard Stone DFE Sixth Form Policies  
;Leave the Policy Names below blank if you dont want them changed  
TeamsMessagingPolicy=Stone_EduSixth  
...  
Filter=  
RequiredGroups=Sixth Form
```



## How to use these Settings

Setting	Purpose
RequiredGroups	Provide a comma separated list of groups (or object IDs) that the user must belong to. If the group does not exist, processing will stop for this function. The user must belong to all groups in the list.
RequiredRecursiveGroups	Provide a comma separated list of groups (or object IDs) or member groups that the user must belong to. Groups processed recursively, so the user could be a member of the primary group, or a group that belongs to the named group. If the primary group does not exist, processing will stop for this function. The user must belong to all groups (or recursive groups) in the list.
RequiredOrGroups	Provide a comma separated list of groups (or object IDs) that the user must belong to. If the group does not exist, processing will stop for this function. The user must belong to at least one of the groups in the list.
RequiredOrRecursiveGroups	Provide a comma separated list of groups (or object IDs) or member groups that the user must belong to. Groups processed recursively, so the user could be a member of the primary group, or a group that belongs to the named group. If the primary group does not exist, processing will stop for this function. The user must belong to at least one of the groups or recursive groups in the list.
NotRequiredGroups	Provide a comma separated list of groups (or object IDs) that the user must not belong to. If the group does not exist, by default this is OK, unless you specify <code>DontAllowMissingNotRequiredGroups=True</code> , in which case processing will stop for this function. The user must not belong to any of the groups in the list.
NotRequiredRecursiveGroups	Provide a comma separated list of groups (or object IDs) or member groups that the user must not belong to. Groups processed recursively, so the user must not be a member of the primary group, or a group that belongs to the named group. If the primary group does not exist, by default this is OK, unless you specify <code>DontAllowMissingNotRequiredGroups=True</code> , in which case processing will stop for this function.
DisableNameorIDCheckingForGroupRules	By default, where you provide a group name, STEPER will perform additional checking to ensure that the groups match. For example, "Office" should not match "Office Staff". You can turn off this additional checking with <code>DisableNameorIDCheckingForGroupRules=True</code>
DontAllowMissingNotRequiredGroups	With <code>NotRequiredGroups</code> and <code>NotRequiredRecursiveGroups</code> , by default these groups don't need to exist, as the user is required not to be a member to proceed.  By setting <code>DontAllowMissingNotRequiredGroups=True</code> , if these groups don't exist, processing will stop for this function.
IncludeDistributionGroupsInGroupRules	By default, all types of Security group (including Mail Enabled Security groups and Distributiongroups) are included in the search. To help in situations where you have many groups with the same or similar names, you can specify that Distribution groups are not included in the search by setting <code>IncludeDistributionGroupsInGroupRules=False</code>

**TIP:** If you wanted to modify all users except those in certain groups, leave `RequiredGroups` / `RequiredRecursiveGroups` / `RequiredOrGroups` / `RequiredOrRecursiveGroups` empty, and just put in the groups of users you want excluded in `NotRequiredGroups` or `NotRequiredRecursiveGroups`

## Using Comma Separated Group Lists

These six settings accept comma separated lists. Unlike other settings in STEPER, they are designed to be give one or more group names or object IDs, instead of always a single group name.

```
RequiredGroups=  
RequiredRecursiveGroups=  
RequiredOrGroups=  
RequiredOrRecursiveGroups=  
NotRequiredGroups=  
NotRequiredRecursiveGroups=
```

For example:

```
RequiredGroups=IT Support
```

- ✓ This will work just fine, as STEPER can see that you don't have any commas, so it is just one group name we are dealing with here.

```
RequiredGroups=IT Support, Office 365, Year11
```

- ✗ This won't work. As you have both a comma (indicating more than one group) and spaces, you must enclose names that include spaces inside quotation marks.

```
RequiredGroups="IT Support","Office 365",Year11
```

- ✓ This will work just fine. As you have a comma STEPER knows you are supplying more than one group. All group names that include a space or special characters should be enclosed in quotation marks.

## Using ObjectIDs

Where you have more than one group with the same name (as can happen when you have groups of different types), the settings that accept comma separated lists also accept object IDs.

```
RequiredGroups="IT Support",ac6d9975-f638-4ceb-8f51-4da5003584b8
```

- ✓ This will work just fine, as STEPER will recognise the second group as a GUID, or object ID

## Group Recursion

The comma separated lists below support group recursion:

RequiredRecursiveGroups=  
 RequiredOrRecursiveGroups=  
 NotRequiredRecursiveGroups=

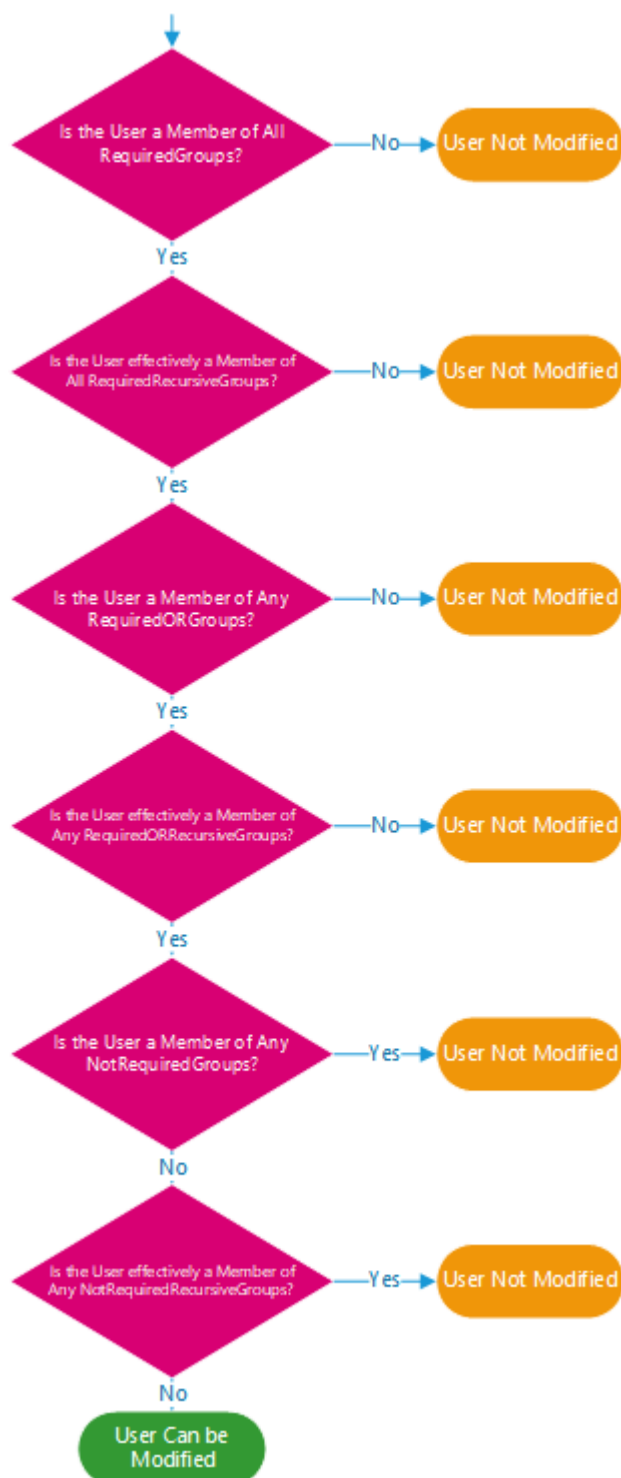
Setting	Recursive Example	Output
RequiredRecursiveGroups=All Staff	"All Staff" group contains: All Admin Staff All Teachers All Site Staff	If a user is a member of either: <ul style="list-style-type: none"> <li>• All Staff</li> <li>• All Admin Staff</li> <li>• All Teachers</li> <li>• All Site Staff</li> </ul> ...Then the rule will apply to them and they can be processed
RequiredRecursiveGroups=All Staff	"All Staff" group is a member of the School_A group	The user can be processed if they are a member of "All Staff" (or one of the groups that's a member of "All Staff").  They cannot be a processed if they are only a member of School_A, because School_A is not a member of "All Staff"
NotRequiredRecursiveGroups=Pupils	"Pupils" Contains: Upper_School Lower_School	If the users if a member of either: <ul style="list-style-type: none"> <li>• Pupils</li> <li>• Upper_School</li> <li>• Lower_School</li> </ul> ...Then the rule will apply to them and they cannot be processed, as this is a negative requirement rule
NotRequiredRecursiveGroups=Sixth Form	"Sixth Form" group is a member of: Pupils	The rule will apply to the user if they are a member of Sixth Form (or one of the groups that's a member of Sixth Form), in which case they cannot be processed, as this is a negative requirement rule  The rule will not apply to the user if they are only a member of Pupils, because Pupils is not a member of "Sixth Form"

If the result of any rule is that the user should not be processed, then they will not be processed. The user has to meet all of the requirements. These lists of rules can contain an unlimited number of groups or object IDs.

## Group Rules Process

This chart illustrates how the group settings are processed and applied for:

- StudentSecurityGroupX / TeacherSecurityGroupX
- StudentUnifiedGroupX / TeacherUnifiedGroupX
- StudentDistributionGroupX / TeacherDistributionGroupX
- StudentDistributionGroupX / TeacherDistributionGroupX
- StudentPackagesX / TeacherPackagesGroupX
- StudentPoliciesX / TeacherPoliciesGroupX
- GetAllUserPolicyInformation



The same process and settings are available under [InformationRetrieval] > GetAllUserPolicyInformation to allow you to test your group rules against all users.



## Active Directory Organisational Unit (OU) / Canonical Name (CN) Rules

From STEPER 1.0.85, you can now specify some basic on-prem AD OU/CN matching rules on all of the functions that support Attribute Filters.

Function	Supports Attribute Filters	Supports Group Rules	Supports AD OU/CN Rules	Supports Collections	Supports Domain Filters
SourceGroups	No	Yes	No	No	Yes
GlobalAdminTeamsCreation	No	No	No	No	No
Licensing Groups	No	No	No	No	No
AllTeachersGroup / AllStudentsGroup	No	No	No	No	No
Set Student Department	No	No	No	No	No
Information Retrieval	Yes	Yes	Yes	No	Yes
TeacherSecurityGroup / StudentSecurityGroup	Yes	Yes	Yes	Yes	Yes
TeacherUnifiedGroup / StudentUnifiedGroup	Yes	Yes	Yes	Yes	Yes
TeacherDistributionGroup / StudentDistributionGroup	Yes	Yes	Yes	Yes	Yes
TeacherTeamsPackages / StudentTeamsPackages	Yes	Yes	Yes	Yes	Yes
TeacherTeamsPolicies / StudentTeamsPolicies	Yes	Yes	Yes	Yes	Yes

```
[StudentTeamsPolicies3]
```

```
...
Filter=
FilterSource=
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=False
RequiredOrOUs=
NotRequiredOUs=
RemoveFailedUsersFromLicensingGroup=
OnFailureApplytoAllUsersOnNextRun=
Default=False
CollectionID=
CollectionIDIgnoreThreshold=
CollectionIDsToExclude=
```



## What is a Canonical Name?

First, STEPER converts the on-prem user OU LDAP DN Name and puts it into canonical name format.

For example:

`cn=JDoe,ou=Intake2020,ou=Students,dc=MySchool,dc=com`

This is converted to the canonical format below.

`MySchool.com/Students/Intake2020/`

This represents the on-premise OU that the user object resides in, and this format makes it easy to process users using wildcards and a tree structure.

## Settings Available

Setting	Purpose
RequiredOrOUs	Provide a comma separated list of OUs / Canonical Names for matching. Can include a prefix or suffix of * for pattern matching. All users are processed against the required list first, but then can be removed from the users to be processed by the NotRequiredOUs list below.
NotRequiredOUs	Users that match the NotRequiredOUs list are then removed from the list of users that can be processed. Can include a prefix or suffix of * for pattern matching.

These settings do not support matching against specific user objects – only against OUs.



## Examples

### *Example 1 – All Students in an OU and sub-OUs*

RequiredOrOUs=MySchool.com/Students/\*

This would match against the following users:

MySchool.com/Students/Intake2015/Deleted  
MySchool.com/Students/Intake2020/  
MySchool.com/Students/Intake2021/  
MySchool.com/Students/

It would not match against:

MySchool.com/Students\_Old/  
MySchool.com/Second\_Campus/Students/

### *Example 2 – Use of a wild-card prefix*

RequiredOrOUs=\*/Students/\*

This would match against the following users:

MySchool.com/Students/Intake2015/Deleted  
MySchool.com/Students/Intake2020/  
MySchool.com/Students/Intake2021/  
MySchool.com/Students/  
MySchool.com/Second\_Campus/Students/

It would not match against:

MySchool.com/Students\_Old/

Use of the \* symbol before /Students means that you don't need to know the entire canonical name. You can find the canonical name from Active director or use the [Information Retrieval](#) function.



### *Example 3 – Split Name Matching*

RequiredOrOUs=\*/Intake\*

This would match against the following users:

MySchool.com/Students/Intake2015/Deleted

MySchool.com/Students/Intake2020/

MySchool.com/Students/Intake2021/

It would not match against:

MySchool.com/Students/

MySchool.com/Second\_Campus/Students/

MySchool.com/Students\_Old/

### *Example 4 – Excluding some users*

RequiredOrOUs=MySchool.com/Students/\*

NotRequiredOUs=\*/Deleted/\*

Users are matched against the RequiredOrOUs list first, and then removed from processing if they match any of the NotRequiredOUs.

This would match against the following users:

MySchool.com/Students/Intake2020/

MySchool.com/Students/Intake2021/

MySchool.com/Students/

It would not match against:

MySchool.com/Students/Intake2015/Deleted

MySchool.com/Second\_Campus/Students/

MySchool.com/Students\_Old/





### *Example 5 – Lists*

RequiredOrOUs=\*/Intake\*,\*/Students\_Old/\*  
NotRequiredOUs=\*/Deleted/\*

This would match against the following users:

MySchool.com/Students/Intake2020/  
MySchool.com/Students/Intake2021/  
MySchool.com/Students\_Old/

It would not match against:

MySchool.com/Students/  
MySchool.com/Students/Intake2015/Deleted

### *Example 6 – Just Excluding Some Users*

RequiredOrOUs=  
NotRequiredOUs=\*/Deleted/\*

In this example, as NotRequiredOUs has been provided, RequiredOrOUs has been assumed to be all users, or \*.

This would match against the following users:

MySchool.com/Students/Intake2020/  
MySchool.com/Students/Intake2021/  
MySchool.com/Students/  
MySchool.com/Students\_Old/

It would not match against:

MySchool.com/Students/Intake2015/Deleted

## Collections

From STEPER 1.0.61, the concept of collections is supported on all of the functions that support Attribute Filters, with the exception of [\[InformationRetrieval\]](#). CollectionIDs allow you to build up some basic processing rules.

Function	Supports Attribute Filters	Supports Group Rules	Supports AD OU/CN Rules	Supports Collections	Supports Domain Filters
SourceGroups	No	Yes	No	No	Yes
GlobalAdminTeamsCreation	No	No	No	No	No
Licensing Groups	No	No	No	No	No
AllTeachersGroup / AllStudentsGroup	No	No	No	No	No
Set Student Department	No	No	No	No	No
Information Retrieval	Yes	Yes	Yes	No	Yes
TeacherSecurityGroup / StudentSecurityGroup	Yes	Yes	Yes	Yes	Yes
TeacherUnifiedGroup / StudentUnifiedGroup	Yes	Yes	Yes	Yes	Yes
TeacherDistributionGroup / StudentDistributionGroup	Yes	Yes	Yes	Yes	Yes
TeacherTeamsPackages / StudentTeamsPackages	Yes	Yes	Yes	Yes	Yes
TeacherTeamsPolicies / StudentTeamsPolicies	Yes	Yes	Yes	Yes	Yes

### [StudentTeamsPolicies3]

```

...
Filter=
FilterSource=
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=False
RequiredOrOUs=
NotRequiredOUs=
RemoveFailedUsersFromLicensingGroup=
OnFailureApplytoAllUsersOnNextRun=
Default=False
CollectionID=
CollectionIDIgnoreThreshold=
CollectionIDsToExclude=

```

## CollectionID Processing Basics

When users are processed, they are matched against the Attribute filters and Group filters. Users that meet those rules will be added to the group or have policies applied etc. STEPER will then record the collection ID used to perform those actions.

So, in the next batch of users, if a user again matches the Attribute filters and Group filters, they won't be added to the next group if the new collection ID is higher than the collection ID previously associated with that user, unless the previous collection ID is specifically excluded, or the previous collection ID is at or below the ignore threshold.

## Global Settings Available

Setting	Default Value	Purpose
[Collections] <b>IgnoreCollectionIDUserType</b>	False	A user should never be a Student and a Teacher. However, from STEPER 1.0.86 onwards, the user Type is checked during Collection ID processing – for example in the situation when a user is in both Student and Teacher groups. To ignore the user type and potentially apply more settings to these users, set IgnoreCollectionIDUserType=True



## Function Settings Available

Setting	Default Value	Purpose
Default	False	Enforces default processing. Processed last. The setting or action will take place if the user hasn't already matched an action.
CollectionID	100 (forced to 10000 if Default=True)	<p>Each setting or action can be given a Collection ID. This can be used more than once. Minimum value of 0, maximum 10000. Settings/Actions are processed in CollectionID order and then settings number.</p> <p>Once a user has met the group/filter/OU rules and is processed against a collection, the user can't be processed under a higher collection number unless the previously matched collection is ignored using CollectionIDIgnoreThreshold or CollectionIDstoExclude</p>
CollectionIDIgnoreThreshold	0 (forced to 0 if Default=True)	<p>User actions made on collection IDs of this threshold and below will be ignored when looking at what has already been done to each user.</p> <ul style="list-style-type: none"> <li>• If the user has been processed on a collection with an ID above this threshold, the setting or action will not take place</li> <li>• If the user has not previously been processed, or has been processed on a collection with an ID at or below this threshold, the setting or action will take place</li> </ul>
CollectionIDsToExclude	[empty]	<p>Similar to CollectionIDIgnoreThreshold, but with CollectionIDsToExclude you can give a comma separated list of CollectionIDs to ignore, including basic numerical ranges</p> <ul style="list-style-type: none"> <li>• If the user has not previously been processed, or has been processed on a collection with an ID on this list, the setting or action will take place</li> <li>• If the user has been processed on a collection with an ID <b>not</b> in this list or above the CollectionIDIgnoreThreshold, the setting or action will not take place</li> </ul>



## CollectionID Numbering and Processing Order

Collections work by assigning each collection of settings an ID. If not specified, the default collection ID is 100. Collection IDs can range from 0 to 10000.

The lower the collection ID, the greater the priority. Settings with the lowest collection ID will be processed first. After the collection ID, settings will be processed by settings number, e.g. [TeacherDistributionGroup1] = 1, [TeacherDistributionGroup2] = 2

Once a user has been processed, they will not be processed again in another unless either:

- The Collection ID is the same
- The Collection ID they were processed in is below the CollectionIDIgnoreThreshold in the collection being evaluated
- The Collection ID they were processed in is in "CollectionsIDstoExclude" in the collection being evaluated

All CollectionID settings are completely independent of each other across all functions. For example, making a change to the CollectionID settings under [TeacherUnifiedGroupX] will not affect [StudentUnifiedGroupX]

## CollectionID Basic Examples

### *Example 1*

```
[TeacherDistributionGroup1]
CollectionID=100
..

[TeacherDistributionGroup2]
CollectionID=50
CollectionIDIgnoreThreshold=0
```

In this instance:

- [TeacherDistributionGroup2] will be processed before [TeacherDistributionGroup1], as [TeacherDistributionGroup2] has a lower CollectionID

## Example 2

```
[TeacherDistributionGroup1]
CollectionID=10
RequiredRecursiveGroups=OfficeStaff
DistributionGroupName=OfficeDistributionGroup
```

```
[TeacherDistributionGroup2]
CollectionID=10
RequiredRecursiveGroups=OfficeStaff
DistributionGroupName=OfficeStaffExternal
```

```
[TeacherDistributionGroup3]
CollectionID=20
RequiredRecursiveGroups=AllStaff
DistributionGroupName=TeachingStaffExternal
```

In this situation, a user that is a member of OfficeStaff and AllStaff will be processed against [TeacherDistributionGroup1] and [TeacherDistributionGroup2] as the CollectionID is the same, but not against [TeacherDistributionGroup3] as the CollectionID is higher. When CollectionID20 is processed, matching users that were already processed under a lower collection ID are ignored.

**TIP:** If you wanted to process all users under CollectionID 20, you could:

- Change the Collection ID for [TeacherDistributionGroup3] to 10
- Or, add CollectionIDIgnoreThreshold=10 to [TeacherDistributionGroup3] to 10 – this means that previous CollectionIDs of 10 and below are ignored when considering if the user has already been processed
- Alternatively, add CollectionIDsToExclude=20 to [TeacherDistributionGroup3] to 10

## CollectionIDs: Groups vs. Packages vs. Policies

For Security Groups, Unified Groups and Distribution Groups, and Teams Packages:

- The actual group name being applied, and the other group rules or filters are not examined in CollectionID processing

For Teams Policies:

- The type of policy e.g. TeamsMeetingPolicy is checked in CollectionID processing, and checking to see if processing has already been done for that user, is done at the individual policy level

## Default Processing

For example, setting :

```
[TeacherDistributionGroup1]
...

[TeacherDistributionGroup99]
CollectionID=10000
CollectionIDIgnoreThreshold=0
CollectionIDstoExclude=
```

This ensures that a user that matches the attribute filter rules under [TeacherDistributionGroup99] will be processed only if they haven't been processed before.

You can also do this by using the Default setting:

```
[TeacherDistributionGroup1]
...

[TeacherDistributionGroup99]
Default=True
```

By setting Default=True:

- A CollectionID of 10000 is assumed
- The CollectionIDIgnoreThreshold is set to 0
- CollectionIDstoExclude is empty



## CollectionIDIgnoreThreshold Example

User actions made on Collection IDs of this threshold and below will be ignored when looking at what has already been done to each user.

- If the user has been processed on a collection with an ID above this threshold, the setting or action will not take place
- If the user has not previously been processed, or has been processed on a collection with an ID at or below this threshold, the setting or action will take place

```
[TeacherDistributionGroup1]
CollectionID=10
CollectionIDIgnoreThreshold=0
```

```
[TeacherDistributionGroup2]
CollectionID=50
CollectionIDIgnoreThreshold=0
```

```
[TeacherDistributionGroup3]
CollectionID=100
CollectionIDIgnoreThreshold=50
```

In this instance:

- User A that matches the group filter and attribute filter rules under [TeacherDistributionGroup1] will be processed in CollectionID=10
- The same user that also matches the group filter and attribute filter rules under [TeacherDistributionGroup2] will not be processed in CollectionID=50, because they have already been processed in CollectionID=10, in the collection now being evaluated the CollectionIDIgnoreThreshold is 0, which is less than 10, so does not apply
- The same user that also matches the group filter and attribute filter rules under [TeacherDistributionGroup3] will be processed in CollectionID=100, as they were previously processed in CollectionID=10 which is less than the CollectionIDIgnoreThreshold of 50





## CollectionIDstoExclude Lists

Previous User actions made using these collection IDs will be ignored. If the user has been processed on a collection with an ID **not** in this list, the setting or action will not take place.

If the user has not previously been processed, or has been processed on a collection with an ID on this list, the setting or action will take place.

CollectionIDstoExclude supports a comma separated list of CollectionIDs, and basic numerical ranges, however the list has to be formatted correctly.

Non-Acceptable Examples	Acceptable Examples
30.	30
1 20 25	1,20,25
1 20-25 58	1,20-25,58
1 "20-25" 58	1,"20-25",58
47-45	45-47
- 3	-3
9997 -	9997-
3,15 – 18,21	3,15-18,21 Or 3,"15-18",21

## CollectionIDs Complex Examples

### *Collection ID Group Example 1*

```
[TeacherDistributionGroup1]
CollectionID=10
RequiredRecursiveGroups=Year7Staff
...

[TeacherDistributionGroup2]
CollectionID=20
RequiredRecursiveGroups=OfficeStaff
...

[TeacherDistributionGroup3]
CollectionID=30
CollectionIDstoExclude=10
RequiredRecursiveGroups=OfficeStaff
...
```

In this instance:

User A that meets the group requirements of Year7Staff and OfficeStaff will:

- Be processed under [TeacherDistributionGroup1] (CollectionID=10)
- Won't be processed under [TeacherDistributionGroup2] (CollectionID=20), as they've already been processed under [TeacherDistributionGroup1]
- Will be processed under [TeacherDistributionGroup3], as this has been set to Exclude Collection ID=10

### *Example 2*

```
[TeacherDistributionGroup1]
CollectionID=10
RequiredRecursiveGroups=Year7Staff
...

[TeacherDistributionGroup2]
CollectionID=20
RequiredRecursiveGroups=OfficeStaff
...

[TeacherDistributionGroup3]
CollectionID=30
CollectionIDstoExclude=20
RequiredRecursiveGroups=OfficeStaff
...
```

In this instance:

User A that meets the group requirements of Year7Staff and OfficeStaff will:

- Be processed under [TeacherDistributionGroup1] (CollectionID=10)
- Won't be processed under [TeacherDistributionGroup2] (CollectionID=20), as they've already been processed under [TeacherDistributionGroup1]
- Won't be processed under [TeacherDistributionGroup3] (CollectionID=30), as they've already been processed under [TeacherDistributionGroup1] (CollectionID=10), and the CollectionIDsToExclude=20 doesn't match



## Collection ID Group Example 2

```
[TeacherUnifiedGroup1]
CollectionID=10
EnableUnifiedGroup=True
UnifiedGroupName=Collaboration
RequiredRecursiveGroups=OfficeStaff
Filter=($_.Department -eq "MainSite")
```

```
[TeacherUnifiedGroup2]
CollectionID=10
EnableUnifiedGroup=True
UnifiedGroupName=Collaboration
RequiredRecursiveGroups=LeadershipTeam
Filter=($_.Department -eq "CentralHQ")
```

In this situation, you want:

- Users from the MainSite that are members of Officestaff to be added to the Collaboration Group
- Users from the CentralHQ that are members of LeadershipTeam to be added to the Collaboration Group
- You don't want MainSite users that are members of the Leadership team, or CentralHQ users that are members of Officestaff to be added to the Collaboration Group

Without CollectionID processing, the Collaboration Group members would be altered to match the first rule, and then altered again to only match the second rule. Because the CollectionID is the same, both rules are processed and the resulting membership includes both sets of users.

STEPPER does this by processing all group additions first, and then by processing all remaining pending user removals.

**TIP:** Existing members of groups that don't match any rules or sets of settings for that group are always removed if `RemovedUnMatchedMembers=True`, irrespective of CollectionID. Existing Members aren't matched to CollectionIDs, only new prospective members.

### Collection ID Group Example 3

```
[TeacherUnifiedGroup1]
CollectionID=10
EnableUnifiedGroup=True
UnifiedGroupName=Year1Staff
RequiredOrGroups=Maths, Science, English, Technology
Filter=($_.Department -eq "Year1")
```

```
[TeacherUnifiedGroup2]
CollectionID=20
EnableUnifiedGroup=True
UnifiedGroupName=NonYear1Staff
RequiredOrGroups=Maths, Science, English, Technology
Filter=
```

```
[TeacherUnifiedGroup3]
CollectionID=30
EnableUnifiedGroup=True
UnifiedGroupName=SeniorLeadership-ExcludingYear1
RequiredOrGroups=SLT
CollectionIDsToExclude=20
Filter=
```

```
[TeacherUnifiedGroup4]
Default=True
EnableUnifiedGroup=True
UnifiedGroupName=OtherUsers
RequiredOrGroups=
Filter=
```

#### In this situation

- All users that meet the requirements under [TeacherUnifiedGroup1] / CollectionID 10 are added to the group Year1 Staff
- All users that meet the requirements under [TeacherUnifiedGroup2] / CollectionID 20 are added to the group Year1 Staff, except for the staff that were already added under [TeacherUnifiedGroup1] / CollectionID 10, as this has a lower CollectionID
- All users that meet the requirements under [TeacherUnifiedGroup3] / CollectionID 30 are added to the group Year1 Staff, except for the staff that were already added under [TeacherUnifiedGroup1] / CollectionID 10, as this has a lower CollectionID. Staff that meet the rules of [TeacherUnifiedGroup2] / CollectionID 20 can also be added, as these users have been excluded from consideration as being processed, by the CollectionIDsToExclude=20
- All other users not processed under Unified Group Processing will be added to the OtherUsers Group under [TeacherUnifiedGroup4]



## Domain Filters

From STEPER 1.0.97, domain filters have been made available on most functions.

Function	Supports Attribute Filters	Supports Group Rules	Supports AD OU/CN Rules	Supports Collections	Supports Domain Filters
SourceGroups	No	Yes	No	No	Yes
GlobalAdminTeamsCreation	No	No	No	No	No
Licensing Groups	No	No	No	No	No
AllTeachersGroup / AllStudentsGroup	No	No	No	No	No
Set Student Department	No	No	No	No	No
Information Retrieval	Yes	Yes	Yes	No	Yes
TeacherSecurityGroup / StudentSecurityGroup	Yes	Yes	Yes	Yes	Yes
TeacherUnifiedGroup / StudentUnifiedGroup	Yes	Yes	Yes	Yes	Yes
TeacherDistributionGroup / StudentDistributionGroup	Yes	Yes	Yes	Yes	Yes
TeacherTeamsPackages / StudentTeamsPackages	Yes	Yes	Yes	Yes	Yes
TeacherTeamsPolicies / StudentTeamsPolicies	Yes	Yes	Yes	Yes	Yes

**NOTE:** The DomainFilter settings available for [TeacherSourceGroups] and [StudentSourceGroups] are very similar. See Source Group Domain Filtering [here](#).

The domain filter allows you to filter eligible users based on the domain name portion of their primary username.

When specifying the domain names to be used for filtering, do not include the @ symbol in the filter. The domain filter field allows you to specify a comma separated list of domains.

Example:

```
[TeacherTeamsPackages1]
```

```
DomainFilter=subdomain.myschool.org.uk,subdomain2.myschool.org.uk
```

- A user that has with the username [joeblogs@myschool.org.uk](#) will not be processed, as their suffix myschool.org.uk does not match subdomain.myschool.org.uk or subdomain2.myschool.org.uk
- A user that has with the username of [johnsmith@subdomain.myschool.org.uk](#) will be processed



## Function Control

You can use [FunctionControl] to turn off some functions in bulk. This saves you making lots of smaller changes to the settings file if you need temporarily disable some parts of STEPER processing.

Setting	Default Setting
TurnOffTeacherFilteredSecurityGroupFunctionality	False
TurnOffStudentFilteredSecurityGroupFunctionality	
TurnOffTeacherFilteredUnifiedGroupFunctionality	
TurnOffStudentFilteredUnifiedGroupFunctionality	
TurnOffTeacherFilteredDistributionGroupFunctionality	
TurnOffStudentFilteredDistributionGroupFunctionality	
TurnOffTeacherFilteredPackageFunctionality	
TurnOffStudentFilteredPackageFunctionality	
TurnOffTeacherFilteredPolicyFunctionality	
TurnOffStudentFilteredPolicyFunctionality	

For example, setting `TurnOffTeacherFilteredUnifiedGroupFunctionality=True` disables all Teacher Unified group processing.

Use these settings instead of individually turning off each section of settings in the configuration file (e.g. `EnableUnifiedGroup=False`)

## Resolving Problems with Users

STEPER may from time-to-time encounter problems processing users, including:

- Assigning users to groups (example: due to a connectivity issue with Microsoft's platform)
- Applying Packages (for example, due to the user not being licensed, or enabled)
- Applying Policies (for example, due to the user not being licensed, or enabled)

Common causes for individual users to experience issues are:

- Legacy users that haven't been properly provisioned
- The user isn't actually licensed (which can happen if you aren't using group-based licensing with your licensing groups)
- Users that are disabled in Microsoft Teams
- Problems with your internet connection or the Microsoft platform during Package or Policy processing, resulting in an assignment error

## Methods of Resolving Problems with Users

- From STEPER 1.0.31 onwards, STEPER supports [Remediating](#) these users by removing them from the licensing group, allowing them to be processed again on the next run. This setting is called "RemoveFailedUsersFromLicensingGroup" and is available on Package and Policy Processing
- Alternatively, from STEPER 1.0.68, STEPER supports [User Re-Processing](#). This is done by recording in the system registry for that STEPER Profile that either All teachers, and/or All Students should be processed again on the next STEPER run. User Re-Processing is enabled by default unless switched off in the settings file.

## Methods of Resolving STEPER Failures

From STEPER 1.0.68, User Re-Processing will also be triggered by default if STEPER encounters a major warning or problem. This feature is not available with User Remediation.

## Choosing User Remediation or User Re-Processing

### Function Support

Function	Supports Individual User Remediation	Supports All Teacher / All Students User Re-Processing
GlobalAdminTeamsCreation	No	No, as this checked each run if the function is enabled
Licensing Groups		No, as Licensing Groups are always checked on each run
AllTeachersGroup / AllStudentsGroup		Yes - OnFailureApplytoAllUsersOnNextRun=[list]
Set Student Department		Yes - OnFailureApplytoAllUsersOnNextRun=True (Students Only)
Information Retrieval	-	-
TeacherSecurityGroup / StudentSecurityGroup	No	Yes - OnFailureApplytoAllUsersOnNextRun=[list]
TeacherUnifiedGroup / StudentUnifiedGroup		
TeacherDistributionGroup / StudentDistributionGroup		
TeacherTeamsPackages / StudentTeamsPackages	Yes – RemoveFailedUsersFromLicensingGroup=[list]	Yes – OnFailureApplytoAllUsersOnNextRun=[list]
TeacherTeamsPolicies / StudentTeamsPolicies		

### Comparison

User Impact	User Remediation	User Re-Processing
Minimum STEPER version	1.0.31	1.0.68
Enabled by Default	No	Yes
Requires Teachers and or Students to be being assigned to licensing groups	Yes	No
Potentially De-licenses the User until STEPER is run again	Yes	No
Increases the amount of STEPER processing time on the next run	Depends on the number of users to be remediated	Yes, as all users will be processed again, even those in the licensing groups
Reliably corrects the user problem	Yes, as long as the original cause is resolved	
If the user failed again	User will be re-submitted for Remediation	All Users will be re-processed
[Remediation] OnlyRemediateNewTeachers / OnlyRemediateNewStudents	Yes, as the individual users are worked on again in the next run to resolve the problem	No
Triggered if STEPER encounters a Major Error	No	Yes, by default
Triggered if STEPER encounters a Warning	No	Supported, but not recommended
Other	-	Can be used as a fall-back to User Remediation

## Recommendations to Avoid the Need for Remediation or Re-Processing

- Check your STEPER results emails or event logs for ongoing problems
- Increase the [amount of time that STEPER pauses](#) after making a licensing group change if you have lots of problems with lots of users
- Do not reboot your computer whilst STEPER is running
- Do not run multiple copies of STEPER against the same tenancy and profile settings at the same time
- Occasionally, consider processing all users anyway to ensure consistency





## User Remediation

STEPER is designed to operate in one of two main modes:

- Processing All Teachers or All Students
- Processing just new Teachers or Students – these are identified by not being members of Licensing Groups

Most functions allow you specify which mode to use using the settings below:

### Processing All Users

Example: `ApplyTeamsPoliciesToAllTeachers=True`

In this situation All Teachers in the AllTeachers source group are processed. This is most suited to an initial first run of STEPER where you want to update all users.

### Processing New Users Only

Example: `ApplyTeamsPoliciesToAllTeachers=False`

In this situation only Teachers in the AllTeachers source group who are **not already** in the Teacher Licensing Group are processed.

This is most suited to a daily quick run STEPER where you want to process new users only.

If you are using STEPER to process new users only, the overview process is:

- Work out which users need to be processed
- Users are added to the licensing group
- Filtered Group processing
- Package and Policy processing

If the user is added to the licensing group they will therefore not be processed by STEPER on the next run. In this situation, if the user cannot be properly processed by Package or Policy processing you need to be able to remediate the user in the next run.

User Remediation makes use of the [Overrides] Apply to all Teacher and Apply to All Students functionality, but uses an additional setting stored in the registry against the STEPER profile in use.

## Enabling Remediation - RemoveFailedUsersFromLicensingGroup

The following functions support User Remediation:

- [TeacherTeamsPackagesX] / [StudentTeamsPackagesX]
- [TeacherTeamsPoliciesX] / [StudentTeamsPoliciesX]

```
[TeacherTeamsPolicies1]
;STEPPER supports up to 100 different sets of Teacher Policies by default
ApplyTeamsPolicies=False
;The description is for reporting purposes only
TeamsPoliciesDescription=Standard Stone DFE Teacher Policies
;Leave the Policy Names below blank if you dont want them changed
TeamsMessagingPolicy=Stone_EduTeacher
OnlyChangeTeamsMessagingPolicyIfBlank=True
...
Filter=
FilterSource=
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=False
RequiredOrOUs=
NotRequiredOUs=
RemoveFailedUsersFromLicensingGroup=
OnFailureApplytoAllUsersOnNextRun=
Default=False
CollectionID=
CollectionIDIgnoreThreshold=
CollectionIDsToExclude=
```

Use the setting RemoveFailedUsersFromLicensingGroup to give a comma separated list of problem types you would like remediated.



### Problem Settings

Setting	(Internal Boolean Value)	Problem or Failure Type
False (default)	0	None / no remediation
AssignmentError	2	Users that had an assignment error (e.g. MS platform issue) will be remediated
Disabled	4	Disabled users will be remediated (only applies to Policy or Package processing)
NotLicensed	8	Unlicensed users will be remediated (only applies to Policy or Package processing)
1, All or True	14	All problem types will be remediated

For example, to remediate Assignment Errors and Disabled users, but to ignore Not Licensed problems:

```
RemoveFailedUsersFromLicensingGroup=AssignmentError,Disabled
```

With this setting, any users that STEPER detects were not assigned the Policy or package properly, or users that were disabled, will then be removed from the licensing group so that they can be processed again on the next STEPER run.

### Remediation – Remediating New Failed Users Only, or All Failed Users

If you select all users for Policy processing, (e.g. `ApplyTeamsPoliciesToAllTeachers=True`) this could include a mixture of new and existing users. If you enable remediation by setting `RemoveFailedUsersFromLicensingGroup=All` under `[TeacherTeamsPoliciesX]`, then any user that fails processing will be remediated, including those existing and new users.

An existing user that is remediated could have their licensing removed if they are removed from the licensing group. You may prefer instead that STEPER only remediates new users only.

From STEPER 1.0.32 onwards, a new global setting is available that prevents previously existing users from being remediated. This setting is enabled by default when remediation is used.

#### [Remediation]

```
OnlyRemediateNewTeachers=True
```

```
OnlyRemediateNewStudents=True
```



## When STEPER Won't Remediate

- If adding Teachers to a licensing group is not enabled, STEPER will not remediate Teachers
- If adding Students to a licensing group is not enabled, STEPER will not remediate Students

## Remediation Settings Summary

Setting Area	Setting	Meaning
[TeacherTeamsPackagesX] / [StudentTeamsPackagesX]	RemoveFailedUsersFromLicensingGroup=[problem type list]	Enable remediation for that function for different problems or failures
[TeacherTeamsPoliciesX] / [StudentTeamsPoliciesX]	RemoveFailedUsersFromLicensingGroup=[problem type list]	Enable remediation for that function for different problems or failures
[Remediation]	OnlyRemediateNewTeachers / OnlyRemediateNewStudents	Global setting to prevent existing users from being de-licensed. Default is True

## User Re-processing

Re-Processing stores two values in the system registry against that STEPER profile, called “ApplytoAllTeachersOnNextRun” and “ApplytoAllStudentsOnNextRun”

When STEPER next runs, it examines these values, and as appropriate internally uses the [Override](#) feature to force re-processing.

**NOTE:** If that run of STEPER does not complete without errors, the “ApplytoAllTeachersOnNextRun” and “ApplytoAllStudentsOnNextRun” settings are left in the registry. STEPER must complete without errors for these values to be removed.

### User Re-Processing Settings Summary

Setting Area	Setting	Meaning
[Override]	GlobalOnFailureApplytoAllUsersOnNextRun=[global problem list]	Global setting for enabling User Re-processing
[AllTeachersGroup] / [AllStudentsGroup]	OnFailureApplytoAllUsersOnNextRun=[function problem list]	If All Teachers Unified Group / All Students Unified Group encounters group assignment errors, User Re-Processing will be triggered
[Department]	OnFailureApplytoAllUsersOnNextRun=[function problem list]	If Set Department encounters assignment errors, User Re-Processing will be triggered
[TeacherSecurityGroupX] / [StudentSecurityGroupX] / [TeacherUnifiedGroupX] / [StudentUnifiedGroupX] / [TeacherDistributionGroupX] / [StudentDistributionGroupX] / [TeacherTeamsPackagesX] / [StudentTeamsPackagesX] / [TeacherTeamsPoliciesX] / [StudentTeamsPoliciesX]	OnFailureApplytoAllUsersOnNextRun=[function problem list]	If that function encounters problems or failures, User Re-Processing will be triggered based on the problems or failure types selected
[Remediation]	OnFailureApplytoAllUsersOnNextRun=[function problem list]	If Remediation encounters problems or failures, User Re-Processing will be triggered

## Enabling User Re-Processing – Global Setting

```
[Override]
GlobalOnFailureApplytoAllUsersOnNextRun=True
;The settings below only take effect if set to True
ApplytoAllTeachers=False
ApplytoAllStudents=False
```

You can enable this feature at a global level in the settings file, or in [individual functions](#).

### Global Problem Settings List

Setting	(Internal Boolean Value)	Global Problem or Failure Type
False	0	None / no triggered user re-processing on next run
Functions	2	Functions that had an assignment error (e.g. MS platform issue) will cause user teacher and/or student re-processing on next run
Errors	4	Major Errors in STEPER processing will cause teacher and student re-processing on next run.  When STEPER runs, if it finds that the previous run of STEPER did not complete, with GlobalOnFailureApplytoAllUsersOnNextRun=Errors set, then all users will be re-processed on the same run
1 or True (default)	6	Function that had an assignment error or Major Errors in STEPER processing will cause teacher and student re-processing on next run  When STEPER runs, if it finds that the previous run of STEPER did not complete, with GlobalOnFailureApplytoAllUsersOnNextRun=True set, then all users will be re-processed on the same run
Warnings	8	Warnings STEPER processing will cause teacher and student re-processing on next run (this setting is not recommended)
All	14	All problem types cause teacher and student re-processing on next run (this setting is not recommended)

For example, to remediate STEPER Major Errors and Functions:

GlobalOnFailureApplytoAllUsersOnNextRun=Functions,Errors

or

GlobalOnFailureApplytoAllUsersOnNextRun=True

**NOTE:** If the GlobalOnFailureApplytoAllUsersOnNextRun setting is not included in the settings file, it will be added with a value of True. If you want to turn this functionality off, set it to False

## Enabling User Re-Processing – for Individual Functions

Instead of enabling User Processing for Functions at a Global level, you can control the setting at a function level.

The following functions support user-processing on the event of a problem:

- [AllTeachersGroup] / [AllStudentsGroup]
- [Department]
- TeacherSecurityGroup / StudentSecurityGroup / TeacherUnifiedGroup / StudentUnifiedGroup / TeacherDistributionGroup / StudentDistributionGroup
- [TeacherTeamsPackagesX] / [StudentTeamsPackagesX] / [TeacherTeamsPoliciesX] / [StudentTeamsPoliciesX]
- [Remediation]

For example, with the setting below:

OnFailureApplytoAllUsersOnNextRun=True

- Any Teachers that encounter assignment errors, or are disabled, will trigger Teacher re-processing
- Any Students that encounter assignment errors, or are disabled, will trigger Student re-processing

```
[TeacherTeamsPolicies1]
;STEPPER supports up to 100 different sets of Teacher Policies by default
ApplyTeamsPolicies=False
;The description is for reporting purposes only
TeamsPoliciesDescription=Standard Stone DFE Teacher Policies
;Leave the Policy Names below blank if you dont want them changed
TeamsMessagingPolicy=Stone_EduTeacher
OnlyChangeTeamsMessagingPolicyIfBlank=True
...
Filter=
FilterSource=
RequiredGroups=
RequiredRecursiveGroups=
RequiredOrGroups=
RequiredOrRecursiveGroups=
NotRequiredGroups=
NotRequiredRecursiveGroups=
DisableNameorIDCheckingForGroupRules=False
DontAllowMissingNotRequiredGroups=False
IncludeDistributionGroupsInGroupRules=False
RequiredOrOUs=
NotRequiredOUs=
RemoveFailedUsersFromLicensingGroup=
OnFailureApplytoAllUsersOnNextRun=
Default=False
CollectionID=
CollectionIDIgnoreThreshold=
CollectionIDsToExclude=
```

[AllTeachersGroup] / [AllStudentsGroup]

Setting	Boolean Value	Function Problem or Failure Type
False (default)	0	None / no triggered user re-processing on next run
AssignmentError	2	Users that had an assignment error (e.g. user could not be added to the group) will cause user re-processing on next run
GroupProblem	4	The group is missing or could not be created
-	8	(Reserved for future use)
1, All or True	14	All problem types will cause user re-processing on next run

Example:

[AllTeachersGroup]

OnFailureApplytoAllUsersOnNextRun=True

or

OnFailureApplytoAllUsersOnNextRun=2

or

OnFailureApplytoAllUsersOnNextRun=GroupProblem

[Department]

Setting	Boolean Value	Function Problem or Failure Type
False (default)	0	None / no triggered user re-processing on next run
1, AssignmentError, All or True	1	Users that had an assignment error (e.g. user could not be added to the group, or a package or policy assignment problem) will cause user re-processing on next run

Example:

[Department]

OnFailureApplytoAllUsersOnNextRun=True





TeacherSecurityGroup / StudentSecurityGroup / TeacherUnifiedGroup / StudentUnifiedGroup /  
TeacherDistributionGroup / StudentDistributionGroup

Setting	Boolean Value	Function Problem or Failure Type
False (default)	0	None / no triggered user re-processing on next run
AssignmentError	2	Users that had an assignment error (e.g. user could not be added to the group) will cause user re-processing on next run
GroupProblem	4	The group is missing or more than 1 matching group was found
-	8	(Reserved for future use)
1, All or True	14	All problem types will cause user re-processing on next run

Example:

[TeacherSecurityGroup1]

OnFailureApplytoAllUsersOnNextRun=True

or

OnFailureApplytoAllUsersOnNextRun=4

or

OnFailureApplytoAllUsersOnNextRun=AssignmentError



[TeacherTeamsPackagesX] / [StudentTeamsPackagesX] / [TeacherTeamsPoliciesX] /  
[StudentTeamsPoliciesX]

Setting	Boolean Value	Function Problem or Failure Type
False (default)	0	None / no triggered user re-processing on next run
AssignmentError	2	Users that had an assignment error (e.g. a package or policy assignment problem) will cause user re-processing on next run
Disabled	4	Users that are Disabled will cause user re-processing on next run
NotLicensed	8	Users that are Not Licensed will cause user re-processing on next run
1, All or True	14	All problem types will cause user re-processing on next run

Example:

[StudentTeamsPolicies2]

OnFailureApplytoAllUsersOnNextRun=True

or

OnFailureApplytoAllUsersOnNextRun=4

or

OnFailureApplytoAllUsersOnNextRun=AssignmentError,Disabled



[Remediation]

Setting	Boolean Value	Function Problem or Failure Type
False (default)	0	None / no triggered user re-processing on next run
AssignmentError	2	Users that had an assignment error (e.g. the user could not be removed from the licensing group during remediation) will cause user re-processing on next run
1, All or True	14	All problem types will cause user re-processing on next run

Example:

[Remediation]

OnFailureApplytoAllUsersOnNextRun=True

or

OnFailureApplytoAllUsersOnNextRun=2

or

OnFailureApplytoAllUsersOnNextRun=AssignmentError

## Other Settings and Functionality

### [ResultsEmail]

```
[ResultsEmail]
;The FromAddress is usually the same as the GA_UserName
FromAddress=
ToAddress=
UseSSL=
Port=25
Server=smtp.office365.com
DisableEmail=False
OnlySendEmailonWarningorError=False
```

STEPPER will by default send the log file and any other information back to the GA\_User email address. You can use the [ResultsEmail] settings to change the email settings, or disable email completely. Additionally, you can reduce the number of emails sent by setting `OnlySendEmailonWarningorError=True`

### [PowerShell]

```
[PowerShell]
;Note that if any modules below are disabled, the module will not automatically be installed as required
AutoSetTLSecurity=True
DisableNuGetCheck=False
DisablePowerShellGetCheck=False
DisableAzureADPreviewCheck=False
DisableMSOnlineCheck=False
DisableMicrosoftTeamsCheck=False
DisableExchangeConnectorCheck=False
DisableMonthlyModuleUpdateChecks=False
DisableRemoteModuleUpdateRequests=False
ForceUpdateModules=False
```

### AutoSetTLSecurity

By default, STEPPER will force Powershell to use TLS1.2 if STEPPER detects an older version of Windows in use. Set `AutoSetTLSecurity=False` to turn off this functionality.

### DisableMonthlyModuleUpdateChecks

By default, STEPPER will check that the modules and providers it needs are installed. You can disable these checks for the individual modules or providers in this section. In addition, STEPPER then checks once a month for updates to these modules. You can disable this monthly check using `DisableMonthlyModuleUpdateChecks=True`

### ForceUpdateModules

You can force STEPPER to update the modules on each run using the `ForceUpdateModules=True` option. Additionally, STEPPER may attempt to update the Powershell modules when instructed by the STEPPER update engine. You can disable this by specifying `DisableRemoteModuleUpdateRequests=True`

## [Misc]

### [Misc]

```
GetExtraStatistics=True
;The setting below will introduce a pause after applying licensing groups, if any licensing
SleepSecondsAfterApplyingLicensing=600
RefreshTeacherLicensingStatusAfterLicensingGroupChanges=False
RefreshStudentLicensingStatusAfterLicensingGroupChanges=False
;And/or - use the setting below to introduce a pause before Teams Policies are applied
SleepSecondsBeforeApplyingPolicies=
DontPauseForChanges=False
;Set the threshold to 0 to always use Batch Policy Processing
TeacherTeamsPackageBatchProcessingThreshold=99999
StudentTeamsPackageBatchProcessingThreshold=99999
TeacherTeamsPolicyBatchProcessingThreshold=99999
StudentTeamsPolicyBatchProcessingThreshold=99999
MaximumBatchSize=5000
CreateAzureADAccessPolicy=False
;Prevent Modifying Azure AD Policies when Azure AD Token Lifetime Policies already exist
ApplyAzureADAccessPolicyWhenPoliciesAlreadyExist=False
ThrowWarningInsteadofErrorOnAzureADAccessPolicyFailure=False
DontCheckOnMicrosoftCOMPrefixedWhenRemovingAzureADAccessPolicy=False
AssumeAzureADAccessPolicyNotRequiredTimeoutHours=24
;The settings below apply to Attribute filters. If Local attribute filtering is not used, th
UseLocalAttributeFiltering=True
UseAzureADforLocalAttributeFilteringByDefault=True
CheckforUpdates=True
DisableAutoUpdate=False
Verbose=True
;If you do not specify the Logfile folder, it will be chosen automatically
DetailedLogFileFolder=
DisableDetailedLogFile=
DisableSaveSettingsWithLogZIP=
UseCSVforUserDataInLogFile=False
;Use the option below to delete old log files from the Log Folder - must be a minimum of 5 c
CleanUpLogFilesOlderThanDays=60
RepeatLogEventThreshold=20
LongJobUpdateInterval=100
DontRunIfStepperMayBeRunningTimeoutInMinutes=0
MaxSettingsGroups=100
SettingsFileVersion=117
```

## AutoUpdate

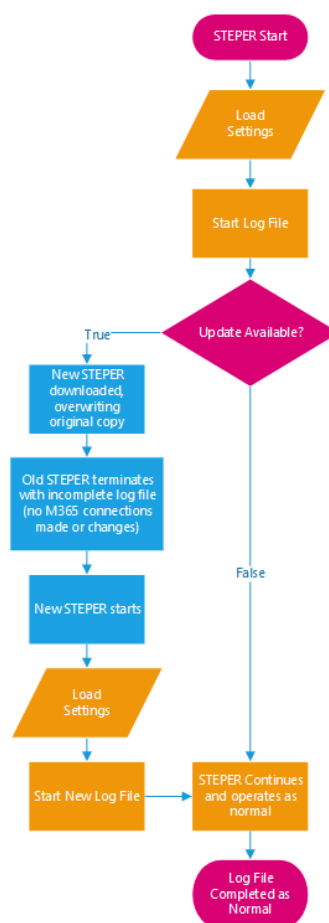
Beginning with version 1.0.20, STEPER now has the ability to update itself to the latest version of the program. This functionality is enabled by default.

- This can be disabled by setting `DisableAutoUpdate=True` under the [Misc] section
- By default, with AutoUpdate enabled, when an update becomes available, STEPER will update itself online and then run again automatically. You do not need to call STEPER again manually as part of the process
- Updates are recorded in the Windows Application Event log
- STEPER will not AutoUpdate if you modify the script

```
ApplySkypeAccessTokenTimeoutRemediationWhenPoliciesAlreadyExist=False
ThrowWarningInsteadofErrorOnRemediationFailure=False
DontCheckOnMicrosoftCOMPrefixedWhenRemovingRemediation=False
CheckForUpdates=True
DisableAutoUpdate=True
Verbose=True
;If you do not specify the Logfile folder, it will be chosen automatically
DetailedLogFileFolder=
DisableDetailedLogFile=
```

If you do not consent to receive updates to STEPER, please either do not use the software, or alternatively use the instructions above to disable AutoUpdate.

- With `DisableAutoUpdate=True`, STEPER will still report on the availability of updates with `CheckForUpdates=True`
- Old versions of STEPER can be disabled by the Update Check (`CheckForUpdates=True`) even when `DisableAutoUpdate=True`



## Azure AD Access Policies

(Previously known as RemediateSkypeAccessTokenTimeout for the Skype For Business Online Powershell Connector)

As of STEPER 1.0.52, the Skype for Business Powershell connector functionality has been removed. The Microsoft Teams Powershell connection only is used.

The Skype connector often required an Azure AD Access Token Policy setting to maintain a working connection during long operations or operations that took more than 1 hour.

Whilst this should no longer be required for the Microsoft Teams Powershell connection, the functionality has been renamed and is disabled by default.

```
[Misc]
GetExtraStatistics=True
;The setting below will introduce a pause after applying licensing groups, if any licensing
SleepSecondsAfterApplyingLicensing=600
RefreshTeacherLicensingStatusAfterLicensingGroupChanges=False
RefreshStudentLicensingStatusAfterLicensingGroupChanges=False
;And/or - use the setting below to introduce a pause before Teams Policies are applied
SleepSecondsBeforeApplyingPolicies=
DontPauseForChanges=False
;Set the threshold to 0 to always use Batch Policy Processing
TeacherTeamsPackageBatchProcessingThreshold=99999
StudentTeamsPackageBatchProcessingThreshold=99999
TeacherTeamsPolicyBatchProcessingThreshold=99999
StudentTeamsPolicyBatchProcessingThreshold=99999
MaximumBatchSize=5000
CreateAzureADAccessPolicy=False
;Prevent Modifying Azure AD Policies when Azure AD Token Lifetime Policies already exist
ApplyAzureADAccessPolicyWhenPoliciesAlreadyExist=False
ThrowWarningInsteadofErrorOnAzureADAccessPolicyFailure=False
DontCheckOnMicrosoftCOMPrefixWhenRemovingAzureADAccessPolicy=False
AssumeAzureADAccessPolicyNotRequiredTimeoutHours=24
```

### *CreateAzureADAccessPolicy*

Disabled by default. Set this to True to force STEPER to make an Azure AD Access Token Policy. This functionality is just for use whilst STEPER is running. When STEPER has finished, it will delete the policy.

To request STEPER to create a temporary Azure AD policy, set CreateAzureADAccessPolicy=True

**NOTE 1:** You must specify the [Tenancy] OnMicrosoftCOMPrefix setting for this functionality to work

**NOTE 2:** If you use CreateAzureADAccessPolicy=True and close STEPER whilst it is running, you must manually delete the Azure AD Policy "STEPPER-AccessTokenLifetime" if it has been created, or run STEPER again with CreateAzureADAccessPolicy=True (and MakeChanges=False if required).



### *ApplyAzureADAccessPolicyWhenPoliciesAlreadyExist*

By default, STEPER will not set its own Azure AD access token policy, when an organisation policy already exists, in case it conflicts with an existing policy. Setting `ApplyAzureADAccessPolicyWhenPoliciesAlreadyExist=True`, STEPER will set its own Azure AD access token policy even when one already exists.

### *ThrowWarningInsteadofErrorOnAzureADAccessPolicyFailure*

If STEPER encounters a problem in setting the Azure AD token access policy, it will by default stop processing, and report an error. No users will be modified. Using `ThrowWarningInsteadofErrorOnRemediationFailure=True` will allow STEPER to continue but no policy will be created.

If STEPER encounters a problem in removing the Azure AD token access policy, it will by default report an error. Users may have already been modified. Using `ThrowWarningInsteadofErrorOnRemediationFailure=True` will tell STEPER to report a warning only.

From version 1.0.1, Azure AD Access Token Lifetime Policy functionality is now aware of multiple instances of STEPER running on the same machine, with different STEPER profiles. For example, running STEPER will a profile of (Teachers) and a profile of (Students) – if they both need to set the Azure AD Access Token Lifetime Policy, the first script to run will set the policy. The second will re-use the existing policy. Whichever script finishes first will leave the policy removal to the second instance. This functionality only works when all instances are run from the same host computer.

**NOTE:** For this functionality to work properly, if you are using multiple runs of STEPER at the same time on the same tenancy, each run must use a different profile name, and all runs should be from the same host computer. `CreateAzureADAccessPolicy=True` should be set on all instances.

### *DontCheckOnMicrosoftCOMPREFIXWhenRemovingAzureADAccessPolicy*

From version 1.0.4, STEPER is also tenancy aware when removing Azure AD Access Token Lifetime Policies. By default, STEPER will only leave a policy in place for another instance of STEPER to use, if both instances are using the same `OnMicrosoftCOMPREFIX`. You change this, so that STEPER ignores checking the `OnMicrosoftCOMPREFIX`, set `DontCheckOnMicrosoftCOMPREFIXWhenRemovingRemediation=True` in all configuration files, however this is more likely to remove Azure AD Access Token Lifetime Policies when they could still be in use.





### *AssumeAzureADAccessPolicyNotRequiredTimeoutHours*

When checking to see if an Azure AD Access Token Lifetime Policy should be removed, STEPER will check to see if other instances of STEPER could be using the policy. STEPER will by default assume that if another instance has been running for 24 hours or more, then it has likely stalled or failed, and the policy should be removed.

If you want to change this timeout time period, set `AssumeAzureADAccessPolicyNotRequiredTimeoutHours` to a different value (minimum = 1, maximum = 168).

### *AzureAD Access Policies – Recommended Settings*

If you need to use this functionality, the following settings are recommended:

`CreateAzureADAccessPolicy=True` (default = False)

`ApplyAzureADAccessPolicyWhenPoliciesAlreadyExist=False` (default = False)

`ThrowWarningInsteadofErrorOnAzureADAccessPolicyFailure=False` (default = False)

`DontCheckOnMicrosoftCOMPREFIXWhenRemovingAzureADAccessPolicy=False` (default = False)

`AssumeAzureADAccessPolicyNotRequiredTimeoutHours=24` (default = 24)

**REMINDER:** For this functionality to work properly, if you are using multiple runs of STEPER at the same or similar times on the same tenancy, each run must use a different profile name, and all runs should be from the same host computer. We recommend that all instances use the same AzureAD Access Policy settings



Batch Processing Thresholds – see [\[TeacherTeamsPackagesX\]](#) / [\[StudentTeamsPackagesX\]](#) and [\[TeacherTeamsPoliciesX\]](#) / [\[StudentTeamsPoliciesX\]](#)

### CheckForUpdates

By default, STEPER will check online to see if there is a new version of the program available and report the outcome in the log file. To obtain a new version, see the Stone Computers Knowledgebase here: <https://kb.stonegroup.co.uk/index.php?View=entry&EntryID=796>

To disable checking for updates, set `CheckForUpdate=False`

**NOTE:** In some circumstances it may be necessary for old versions of STEPER to be disabled by the updater. This usually happens because of a change in M365 or Powershell which makes a new version mandatory. If this happens, this will be recorded in the log file and you will need to update to the newest version.

### Console Warning Suppression

#### *LongJobUpdateInterval*

Some operations can take some considerable time. STEPER can output to the console and log file progress so far based on the number of users processed. For example, with `LongJobUpdateInterval=100`, Steper will update the console screen every 100 users with the estimated time remaining.

Long jobs include:

- `GetAllUserPolicyInformation`
- Teams Package Processing – Get Filtered user existing Package
- Teams Package Processing - Direct user package assignment
- Teams Policy Processing - Direct user Policy assignment

Set this to 99999 to effectively disable it.

#### *RepeatLogEventThreshold*

Use this setting to suppress duplicated messages in the Console. For example, setting `RepeatLogEventThreshold=20` means that console messages about Disabled users or errors applying Policies will be limited to 20. This stops the console filling up with too many error messages.



### DontRunIfSteperMayBeRunningTimeOutInMinutes

You may wish to prevent STEPER from running if the previous run has not finished.

- Runs that use the same data or settings file are identified by Profile name

For example:

- If you run two instances of STEPER at the same time, but using different data and different profile names, your instances wont overlap
- If you run STEPER every hour, using the same data and profile name, but a particular run takes more than an hour, it is possible that your STEPER runs could overlap, which is not ideal

To prevent STEPER runs from overlapping, use the DontRunIfSteperMayBeRunningTimeOutInMinutes option. When STEPER detects that the previous run using that profile name has not completed, it will look to see if this option has been set. If you specify DontRunIfSteperMayBeRunningTimeOutInMinutes=120 but only 90 minutes has passed, STEPER will abort. If 120 minutes has passed, STEPER will run as normal.

### IgnoreMissingNETFramework

STEPPER will warn you if your system does not have .NET Framework 4.x installed and if you need to connect to Microsoft Teams via Powershell. The Teams Powershell module needs .NET Framework 4.x and may fail to connect if it is missing.

Set IgnoreMissingNETFramework=True to turn off this warning

## Logging

### *DetailedLogFileFolder and DisableDetailedLogFile*

- By default, STEPER will save a log file in a “STEPPER Logs” subfolder in the folder that the configuration file was found. Alternatively, specify an alternative folder using DetailedLogFileFolder, for example DetailedLogFileFolder=c:\mylogs
- You can prevent STEPER from saving a log file using DisableDetailedLogFile=True, although this is not recommended

### *DisableSaveSettingsWithLogZIP*

- By default, a ZIP file is generated at the end of each run, containing:
  - The detailed logfile
  - Any GetAllUserPolicyInformation CSV file
  - A copy of the settings file used in the run, with the password redactedYou can turn off the settings file from being saved inside the ZIP, with DisableSaveSettingsWithLogZIP=True, although this is not recommended

### *CleanUpLogFilesOlderThanDays*

- The log folder can be cleaned up of old logs using the CleanUpLogFilesOlderThanDays setting. This will remove log files, CSV files and ZIP files older than the specified number of days (5 days minimum):
  - STEPER v\* Log File.txt
  - STEPER v\* User Export.csv
  - STEPER v\* Log Bundle.zip

### *UseCSVforUserDataInLogFile*

- By default, all user information recorded in the log file is put into easy-to-read tables. If you want to make the user information easier to import into a spreadsheet, use UseCSVforUserDataInLogFile=True in the [Misc] section. This will save the user information in comma separated variables format

### *GetExtraStatistics*

- STEPER can generate some extra statistics for package and policy processing. This includes an additional summary list of actions carried out to all users. This can add some additional process run time when a large number of users have been processed.
- If you need to turn this facility off, set GetExtrastatistics=False



## MaximumBatchSize

Set the maximum number of users that can be submitted for processing for Packages or Policies. If the number of users to be changed is higher than this number, then multiple batches will be used. The default value is 5000 and this is based on the maximum number of users supported by the Teams PowerShell module.

## MaxSettingsGroups

By default, STEPER supports up to 100 groups of settings for the following functions:

- [TeacherSecurityGroupX]
- [StudentSecurityGroupX]
- [TeacherUnifiedGroupX]
- [StudentUnifiedGroupX]
- [TeacherDistributionGroupX]
- [StudentDistributionGroupX]
- [TeacherTeamsPackagesX]
- [StudentTeamsPackagesX]
- [TeacherTeamsPoliciesX]
- [StudentTeamsPoliciesX]

For example, [TeacherTeamsPackages1], [TeacherTeamsPackages2]...[TeacherTeamsPackages100]

You can change the maximum number of groups using MaxSettingsGroups, for example  
MaxSettingsGroups=500

**TIP:** The minimum for MaxSettingsGroups is 10 and the maximum is 1000. Setting this to 1000 unnecessarily will increase program runtime without any benefit, unless you are using huge numbers of settings or actions. Settings are always processed in Collection ID order and then settings numerical order, rather than the order that they appear in the settings file.



## Pauses for Licensing or User Changes

### *SleepSecondsAfterApplyingLicensing*

- By default, STEPER will pause for 600 seconds, after processing licensing groups, if licensing changes were made. You can adjust this value by changing `SleepSecondsAfterApplyingLicensing`. Also see [Refreshing Teacher or Student Licensing Status after Adding Users to Licensing Groups](#)

### *SleepSecondsBeforeApplyingPolicies*

- Alternatively, you can introduce a pause in STEPER before package or policy processing, irrespective of the number of licensing changes, with the `SleepSecondsBeforeApplyingPolicies` option.

### *DontPauseForChanges*

- `DontPauseForChanges` – by default, when simulation mode is turned off (e.g. `MakeChanges=True`) you will get a 20 second console warning from STEPER that changes are going to be made to users. By setting `DontPauseForChanges=True`, this pause is turned off.



## Refreshing Teacher or Student Licensing Status after Adding Users to Licensing Groups

### *RefreshTeacherLicensingStatus*

- By default, when you add a Teacher to the licensing group STEPER does not re-evaluate the users licensing status. This is so that settings such as `OnlyApplytoUnlicensedTeachers=True` take into account the users licensing status when STEPER started, and not the new licensing status. Use `RefreshTeacherLicensingStatusAfterLicensingGroupChanges=True` to get the new licensing status for Teachers that have been added to the Teacher licensing group.

### *RefreshStudentLicensingStatus*

- By default, when you add a Student to the licensing group STEPER does not re-evaluate the users licensing status. This is so that settings such as `OnlyApplytoUnlicensedStudents=True` take into account the users licensing status when STEPER started, and not the new licensing status. Use `RefreshStudentLicensingStatusAfterLicensingGroupChanges=True` to get the new licensing status for Students that have been added to the Student licensing group.

Use either of these settings if you want to work on users that did not get a working license from the licensing group.

**NOTE:** The use of `RefreshTeacherLicensingStatusAfterLicensingGroupChanges=True` or `RefreshStudentLicensingStatusAfterLicensingGroupChanges=True` means that `OnlyApplytoUnlicensedTeachers=True` or `OnlyApplytoUnlicensedStudents=True` will likely not process any users if you have licenses attached to your license group, as the Source List licensing status has been updated

**NOTE 2:** The use of `RefreshTeacherLicensingStatusAfterLicensingGroupChanges=True` or `RefreshStudentLicensingStatusAfterLicensingGroupChanges=True` does not change the effectiveness of `ApplyAllTeachersToGroup` / `ApplyAllStudentsToGroup` / `ApplyTeamsPackagestoAllTeachers` / `ApplyTeamsPackagestoAllStudents` / `ApplyTeamsPoliciestoAllTeachers` / `ApplyTeamsPoliciestoAllStudents`, as these are based on the users membership of the licensing group at the start of STEPER

## UseCSVforUserDataInLogFile

```
[Misc]
GetExtraStatistics=True
;The setting below will introduce a pause after applying licensing groups, if any licensing
SleepSecondsAfterApplyingLicensing=600
RefreshTeacherLicensingStatusAfterLicensingGroupChanges=False
RefreshStudentLicensingStatusAfterLicensingGroupChanges=False
;And/or - use the setting below to introduce a pause before Teams Policies are applied
SleepSecondsBeforeApplyingPolicies=
DontPauseForChanges=False
;Set the threshold to 0 to always use Batch Policy Processing
TeacherTeamsPackageBatchProcessingThreshold=99999
StudentTeamsPackageBatchProcessingThreshold=99999
TeacherTeamsPolicyBatchProcessingThreshold=99999
StudentTeamsPolicyBatchProcessingThreshold=99999
MaximumBatchSize=5000
CreateAzureADAccessPolicy=False
;Prevent Modifying Azure AD Policies when Azure AD Token Lifetime Policies already exist
ApplyAzureADAccessPolicyWhenPoliciesAlreadyExist=False
ThrowWarningInsteadofErrorOnAzureADAccessPolicyFailure=False
DontCheckOnMicrosoftCOMPrefixedWhenRemovingAzureADAccessPolicy=False
AssumeAzureADAccessPolicyNotRequiredTimeoutHours=24
;The settings below apply to Attribute filters. If Local attribute filtering is not used, tr
UseLocalAttributeFiltering=True
UseAzureADforLocalAttributeFilteringByDefault=True
CheckforUpdates=True
DisableAutoUpdate=False
Verbose=True
;If you do not specify the Logfile folder, it will be chosen automatically
DetailedLogFileFolder=
DisableDetailedLogFile=
DisableSaveSettingsWithLogZIP=
UseCSVforUserDataInLogFile=False
;Use the option below to delete old log files from the Log Folder - must be a minimum of 5 c
CleanUpLogFilesOlderThanDays=60
RepeatLogEventThreshold=20
LongJobUpdateInterval=100
DontRunIfStepperMayBeRunningTimeoutInMinutes=0
MaxSettingsGroups=100
SettingsFileVersion=117
```

Set this to True in to tell STEPER to record all user information in the log file in CSV format. This makes it easier to import and manipulate the user data in a spreadsheet such as Excel.

User Attribute Filters (Local or Remote, AzureAD or CSONline)

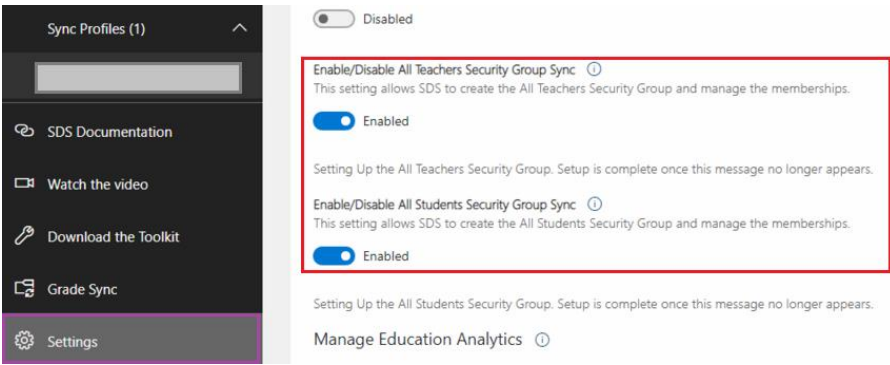
*UseLocalAttributeFiltering*

*UseAzureADforLocalAttributeFilteringByDefault*

See [Powershell Attribute Filters](#).



## Troubleshooting

Problem	Possible Causes
STEPPER says that there is a problem with the settings provided	<ul style="list-style-type: none"> <li>Check the settings file, starting the basic tenancy information</li> <li>Check the log file and look for the “Errors and Warnings” section – example below</li> </ul> <pre> Errors and Warnings: ===== 1 Errors Encountered and 0 Warnings  Errors: - Missing required OnMicrosoftCOMPREFIX from the Settings File           </pre>
STEPPER cannot connect to any online services	<ul style="list-style-type: none"> <li>If you use federated access on your tenancy, try changing the global admin username in the settings file to the onmicrosoft.com address – e.g. <a href="mailto:globaladmin@myschool.onmicrosoft.com">globaladmin@myschool.onmicrosoft.com</a></li> </ul>
STEPPER does not find the All Students or All Teachers Groups, or the groups are empty	<ul style="list-style-type: none"> <li>Ensure your SDS settings have the options ticked to add Students and Teachers to these groups</li> </ul>  <ul style="list-style-type: none"> <li>Make sure the account used has the Global Admin role, and is licensed</li> <li>Check the domain filters under [TeacherSourceGroups]/ [StudentSourceGroups]</li> </ul>
STEPPER does not find all of my users in All Students or All Teachers	<ul style="list-style-type: none"> <li>Manually check the members of these groups in Azure AD</li> <li>If you have nested group, or implied membership (for example, UserX is a member of Group1, which is a member of All Students) then you need to make sure that <a href="#">recursion is turned on</a></li> <li>Use the -Verbose option on the command line, or turn on <a href="#">Log Available Teachers / Log Available Students</a>.</li> </ul>
STEPPER reports that the previous run did not finish	<ul style="list-style-type: none"> <li>Check the log files. STEPPER may still be running in another session</li> <li>Check the Windows Application event log for ID 5201, which is recorded by STEPPER on completion</li> </ul>
STEPPER immediately closes	<ul style="list-style-type: none"> <li>Make sure it is run with Administrative rights</li> <li>Check the Application Event Log</li> <li>Try running it from an existing Administrative Powershell Window</li> </ul>
STEPPER was working but has stopped	<ul style="list-style-type: none"> <li>Check the log files</li> <li>Try updating the Powershell modules (see [PowerShell])</li> <li>Some Microsoft M365 operations take additional time. Please be patient. By default, STEPPER will update you on progress after every 100 operations</li> </ul>
STEPPER reports: This version of STEPPER has been disabled - please update this program	<ul style="list-style-type: none"> <li>Your version of STEPPER has been disabled because of a compatibility issue, likely a change by Microsoft to the M365 platform</li> <li>Contact your account manager for access to the latest version of STEPPER, if available</li> </ul>
STEPPER reports: Could not connect to Microsoft Teams.	<ul style="list-style-type: none"> <li>Check that your Global Admin account is licensed</li> <li>Log onto Microsoft M365 with the account and check that the password works, and that no password change is required</li> </ul>

Check the credentials and version of the Teams Powershell Module (or similar)	<ul style="list-style-type: none"> <li>• Make sure that the system you are using has .NET Framework 4.x installed</li> </ul>
STEPER hasn't set the Student Department field	<ul style="list-style-type: none"> <li>• You may need a Premium AzureAD subscription to make changes to cloud users that are reflected on premises</li> </ul>
STEPER reports: (Required for other functions) Check the Students Licensing settings to make sure the Licensing Group Name is configured	<ul style="list-style-type: none"> <li>• This means that you have specified a function with ApplyAllStudentsXXXX=False. This means that only Students that were not in the licensing group are to be processed. For this to work, STEPER needs to know the name of the Student licensing group.</li> </ul>
STEPER reports: (Required for other functions) Check the Teachers Licensing settings to make sure the Licensing Group Name is configured	<ul style="list-style-type: none"> <li>• This means that you have specified a function with ApplyAllTeachersXXXX =False. This means that only Teachers that were not in the licensing group are to be processed. For this to work, STEPER needs to know the name of the Teacher licensing group.</li> </ul>
STEPER does not update group memberships (for example, licensing groups)	<ul style="list-style-type: none"> <li>• STEPER may not be able to add users to groups where the cloud group is controlled by M365 membership rules</li> <li>• Most of the groups that STEPER uses are Cloud based groups. STEPER will not be able to update groups that originate and are synced from on-prem, unless you have the relevant AzureAD Subscription</li> </ul>
I need to know what changes STEPER has made	<ul style="list-style-type: none"> <li>• Always use Simulation mode first, when deploying STEPER, by setting MakeChanges=False</li> <li>• All STEPER changes are recorded in the detailed log file</li> <li>• In the case of Teams Policies, check for ongoing batch jobs (see GetPolicyJobs)</li> <li>• In the case of needing to use STEPER to correct some manual changes (for example, re-applying new Teams policies to users) you may wish to use the GetAllUserPolicyInformation function first to record the existing settings</li> </ul>
STEPER hasn't processed policies or packages based on group changes	<ul style="list-style-type: none"> <li>• If you are in Simulation mode (MakeChanges = False), STEPER will not actually make changes to group memberships</li> <li>• This means that any further processing that relies on those updated memberships, for example during package or policy processing, will not be aware of the groups that will be updated when MakeChanges is set to True</li> <li>• In this situation, if you do need to simulate the changes, test doing the group changes first, and when you are happy with these, make the changes by setting MakeChanges=True. Then test the policy works after adding the policy settings to the configuration, with MakeChanges=False</li> </ul>

## Settings Reference

Section	Setting Name	Notes	Default *
[General]	<a href="#">STEPPERProfileName</a>		(Untitled)
	<a href="#">MakeChanges</a>		False
	GA_Username	Required	
	<a href="#">GA_Password</a>	Required if no ProtectedGA_Password present	
	ProtectedGA_Password		
[Tenancy]	Protect_Password		True
	<a href="#">OnMicrosoftCOMPrefix</a>	Required if CreateAzureADAccessPolicy is being used to create an Azure AD Access Policy to prevent MS Teams Powershell from being disconnected	
[GlobalAdminTeamsCreation]	PrivateDomain	Not Required	
	<a href="#">AddGlobalAdminToCreateTeamsGroup</a> <i>(previously AddGlobalAdminToCreateTeamsUnifiedGroup)</i>	Can be name or Object ID	False
	<a href="#">DisableNameOrIDChecking</a>		False
[TeacherSourceGroups]	CreateTeamsGroupName <i>(previously CreateTeamsUnifiedGroupName)</i>	Required if AddGlobalAdminToCreateTeamsGroup=True	
	RequiredGroups	Supports List <i>(Previously AllTeachersSourceGroupName)</i>	
	RequiredRecursiveGroups	Supports List <i>(Previously related to EnableAllTeachersSourceGroupRecursion)</i>	
	<a href="#">RequiredOrGroups</a>	Supports List	
	<a href="#">RequiredOrRecursiveGroups</a>	Supports List	* if NotRequiredGroups or NotRequiredRecursiveGroups provided
	<a href="#">NotRequiredGroups</a>	Supports List <i>(Previously ExcludedTeachersGroupName)</i>	
	<a href="#">NotRequiredRecursiveGroups</a>	Supports List <i>(Previously related to EnableExcludedTeachersGroupRecursion)</i>	
	<a href="#">DisableNameOrIDCheckingForGroupRules</a>		False
	<a href="#">DontAllowMissingNotRequiredGroups</a>		False
	<a href="#">IncludeDistributionGroupsInGroupRules</a>		True
	<a href="#">DomainFilter</a>	Supports List	
	<a href="#">DontProcessStudentsFoundinTeacherSourceGroups</a>		False
	<a href="#">LogAvailableUsers</a>		False (unless Verbose specified)
[StudentSourceGroups]	RequiredGroups	Supports List <i>(Previously AllStudentsSourceGroupName)</i>	
	RequiredRecursiveGroups	Supports List <i>(Previously related to EnableAllStudentsSourceGroupRecursion)</i>	
	<a href="#">RequiredOrGroups</a>	Supports List	
	<a href="#">RequiredOrRecursiveGroups</a>	Supports List	* if NotRequiredGroups or NotRequiredRecursiveGroups provided
	<a href="#">NotRequiredGroups</a>	Supports List <i>(Previously ExcludedStudentsGroupName)</i>	
	<a href="#">NotRequiredRecursiveGroups</a>	Supports List <i>(Previously related to EnableExcludedStudentsGroupRecursion)</i>	
	<a href="#">DisableNameOrIDCheckingForGroupRules</a>		False
	<a href="#">DontAllowMissingNotRequiredGroups</a>		False
	<a href="#">IncludeDistributionGroupsInGroupRules</a>		True
	<a href="#">DomainFilter</a>	Supports List	
	<a href="#">DontProcessTeachersFoundinStudentSourceGroups</a>		False
	<a href="#">LogAvailableUsers</a>		False (unless Verbose specified)
	[Override]	ApplytoAllTeachers	
ApplytoAllStudents			False
<a href="#">GlobalOnFailureApplytoAllUsersOnNextRun</a>		Supports List	True
[LicensingGroups]	<a href="#">AssignTeacherstoLicensingGroup</a>		False
	AllTeachersLicensingGroupName	Required if AssignTeacherstoLicensingGroup=True Or any other function where ApplytoAllTeachers=False  Can be name or Object ID	
	<a href="#">DisableTeacherGroupNameOrIDChecking</a>		False
	<a href="#">OnlyApplytoUnlicensedTeachers</a>		False
	<a href="#">CreateAllTeachersLicensingGroupIfNotExists</a>		False
	<a href="#">AssignStudentstoLicensingGroup</a>		False
	AllStudentsLicensingGroupName	Required if AssignStudentstoLicensingGroup=True Or any other function where ApplytoAllStudents=False  Can be name or Object ID	
	<a href="#">DisableStudentGroupNameOrIDChecking</a>		False
	<a href="#">OnlyApplytoUnlicensedStudents</a>		False
	<a href="#">CreateAllStudentsLicensingGroupIfNotExists</a>		False
[AllTeachersGroup]	<a href="#">AddAllTeacherstoUnifiedGroup</a>		False
	<a href="#">OnlyApplytoUnlicensedTeachers</a>		False
	AllTeachersUnifiedGroup	Required if AddAllTeacherstoUnifiedGroup=True	
	<a href="#">DisableNameOrIDChecking</a>		False

	CreateAllTeachersUnifiedGroupIfNotExists		False
	OnFailureApplytoAllUsersOnNextRun	Supports List	False
[AllStudentsGroup]	AddAllStudentsToUnifiedGroup		False
	OnlyApplytoUnlicensedTeachers		False
	AllStudentsUnifiedGroup	Required if AddAllStudentsToUnifiedGroup=True	
	DisableNameOrIDChecking		False
	CreateAllStudentsUnifiedGroupIfNotExists		False
	OnFailureApplytoAllUsersOnNextRun	Supports List	False
[Department]	SetStudentDepartment		False
	UseGraduationYear		False
	OnlyApplytoUnlicensedStudents		False
	SetStudentDepartmentToAllStudents		False
	SetStudentDepartmentOnlyIfDepartmentBlank		False
	OnFailureApplytoAllUsersOnNextRun	Supports List	False
[TeacherSecurityGroup1] to [TeacherSecurityGroup100]	EnableSecurityGroup		False
	SecurityGroupDescription		
	SecurityGroupName	Can be name or Object ID	
	DisableNameOrIDChecking		False
	ApplyAllTeachersToGroup		False
	OnlyApplytoUnlicensedTeachers		False
	Filter	Supports Expression	
	FilterSource		See <a href="#">Filter Sources</a>
	RemoveUnMatchedMembers	Supports List	False
	OnlyRemoveUnMatchedTeachers (option only available if RemoveUnMatchedMembers=True)		True
	RequiredGroups	Supports List	
	RequiredRecursiveGroups	Supports List	
	RequiredOrGroups	Supports List	
	RequiredOrRecursiveGroups	Supports List	
	NotRequiredGroups	Supports List	
	NotRequiredRecursiveGroups	Supports List	
	DisableNameOrIDCheckingForGroupRules		False
	DontAllowMissingNotRequiredGroups		False
	IncludeDistributionGroupsInGroupRules		True
	DomainFilter	Supports List	
	RequiredOrOUs	Supports List	*, if NotRequiredOUs provided
	NotRequiredOUs	Supports List	
	OnFailureApplytoAllUsersOnNextRun	Supports List	False
	Default		False
	CollectionID		100 (or 10000 if Default is True)
	CollectionIDIgnoreThreshold		0
	CollectionIDsToExclude	Supports Lists and Ranges	
[StudentSecurityGroup1] to [StudentSecurityGroup100]	EnableSecurityGroup		False
	SecurityGroupDescription		
	SecurityGroupName	Can be name or Object ID	
	DisableNameOrIDChecking		False
	ApplyAllStudentsToGroup		False
	OnlyApplytoUnlicensedStudents		False
	Filter	Supports Expression	
	FilterSource		See <a href="#">Filter Sources</a>
	RemoveUnMatchedMembers	Supports List	False
	OnlyRemoveUnMatchedStudents (option only available if RemoveUnMatchedMembers=True)		True
	RequiredGroups	Supports List	
	RequiredRecursiveGroup	Supports List	
	RequiredOrGroups	Supports List	
	RequiredOrRecursiveGroups	Supports List	
	NotRequiredGroups	Supports List	
	NotRequiredRecursiveGroups	Supports List	
	DisableNameOrIDCheckingForGroupRules		False
	DontAllowMissingNotRequiredGroups		False
	IncludeDistributionGroupsInGroupRules		True
	DomainFilter	Supports List	
	RequiredOrOUs	Supports List	*, if NotRequiredOUs provided
	NotRequiredOUs	Supports List	
	OnFailureApplytoAllUsersOnNextRun	Supports List	False
	Default		False
	CollectionID		100 (or 10000 if Default is True)
	CollectionIDIgnoreThreshold		0
	CollectionIDsToExclude	Supports Lists and Ranges	
[TeacherUnifiedGroup1] to [TeacherUnifiedGroup100]	EnableUnifiedGroup		False
	UnifiedGroupDescription		
	UnifiedGroupName	Can be name or Object ID	
	ApplyAllTeachersToGroup		False
	OnlyApplytoUnlicensedTeachers		False
	Filter	Supports Expression	



	FilterSource		See <a href="#">Filter Sources</a>
	<a href="#">RemoveUnMatchedMembers</a>	Supports List	False
	DisableNameOrIDChecking		False
	OnlyRemoveUnMatchedTeachers (option only available if RemoveUnMatchedMembers=True)	Supports List	True
	<a href="#">RequiredGroups</a>	Supports List	
	RequiredRecursiveGroups	Supports List	
	RequiredOrGroups	Supports List	
	RequiredOrRecursiveGroups	Supports List	
	NotRequiredGroups	Supports List	
	NotRequiredRecursiveGroups	Supports List	
	DisableNameorIDCheckingForGroupRules		False
	DontAllowMissingNotRequiredGroups		False
	IncludeDistributionGroupsInGroupRules		True
	<a href="#">DomainFilter</a>	Supports List	
	<a href="#">RequiredOrOUs</a>	Supports List	*, if NotRequiredOUs provided
	NotRequiredOUs	Supports List	
	<a href="#">OnFailureApplytoAllUsersOnNextRun</a>	Supports List	False
	Default		False
	<a href="#">CollectionID</a>		100 (or 10000 if Default is True)
	CollectionIDIgnoreThreshold		0
	CollectionIDsToExclude	Supports Lists and Ranges	
[StudentUnifiedGroup1] to [StudentUnifiedGroup100]	<a href="#">EnableUnifiedGroup</a>		False
	UnifiedGroupDescription		
	UnifiedGroupName	Can be name or Object ID	
	ApplyAllTeachersToGroup		False
	OnlyApplytoUnlicensedStudents		False
	<a href="#">Filter</a>	Supports Expression	
	FilterSource		See <a href="#">Filter Sources</a>
	<a href="#">RemoveUnMatchedMembers</a>	Supports List	False
	DisableNameOrIDChecking		False
	OnlyRemoveUnMatchedStudents (option only available if RemoveUnMatchedMembers=True)	Supports List	True
	<a href="#">RequiredGroups</a>	Supports List	
	RequiredRecursiveGroups	Supports List	
	RequiredOrGroups	Supports List	
	RequiredOrRecursiveGroups	Supports List	
	NotRequiredGroups	Supports List	
	NotRequiredRecursiveGroups	Supports List	
	DisableNameorIDCheckingForGroupRules		False
	DontAllowMissingNotRequiredGroups		False
	IncludeDistributionGroupsInGroupRules		True
	<a href="#">DomainFilter</a>	Supports List	
	<a href="#">RequiredOrOUs</a>	Supports List	*, if NotRequiredOUs provided
	NotRequiredOUs	Supports List	
	<a href="#">OnFailureApplytoAllUsersOnNextRun</a>	Supports List	False
	Default		False
	<a href="#">CollectionID</a>		100 (or 10000 if Default is True)
	CollectionIDIgnoreThreshold		0
	CollectionIDsToExclude	Supports Lists and Ranges	
[TeacherDistributionGroup1] to [TeacherDistributionGroup100]	<a href="#">EnableDistributionGroup</a>		False
	DistributionGroupDescription		
	DistributionGroupName	Can be name or Object ID	
	ApplyAllStudentsToGroup		False
	OnlyApplytoUnlicensedTeachers		False
	<a href="#">Filter</a>	Supports Expression	
	FilterSource		See <a href="#">Filter Sources</a>
	<a href="#">RemoveUnMatchedMembers</a>	Supports List	False
	DisableNameOrIDChecking		False
	OnlyRemoveUnMatchedTeachers (option only available if RemoveUnMatchedMembers=True)	Supports List	True
	<a href="#">RequiredGroups</a>	Supports List	
	RequiredRecursiveGroups	Supports List	
	RequiredOrGroups	Supports List	
	RequiredOrRecursiveGroups	Supports List	
	NotRequiredGroups	Supports List	
	NotRequiredRecursiveGroups	Supports List	
	DisableNameorIDCheckingForGroupRules		False
	DontAllowMissingNotRequiredGroups		False
	IncludeDistributionGroupsInGroupRules		True
	<a href="#">DomainFilter</a>	Supports List	
	<a href="#">RequiredOrOUs</a>	Supports List	*, if NotRequiredOUs provided
	NotRequiredOUs	Supports List	
	<a href="#">OnFailureApplytoAllUsersOnNextRun</a>	Supports List	False
	Default		False
	<a href="#">CollectionID</a>		100

	CollectionIDIgnoreThreshold		(or 10000 if Default is True)
	CollectionIDsToExclude	Supports Lists and Ranges	0
[StudentDistributionGroup1] to [StudentDistributionGroup100]	<a href="#">EnableDistributionGroup</a>		False
	DistributionGroupDescription		
	DistributionGroupName	Can be name or Object ID	
	ApplyAllStudentsToGroup		False
	OnlyApplyToUnlicensedStudents		False
	<a href="#">Filter</a>	Supports Expression	
	FilterSource		See <a href="#">Filter Sources</a>
	<a href="#">RemoveUnMatchedMembers</a>	Supports List	False
	DisableNameOrIDChecking		False
	OnlyRemoveUnMatchedStudents (option only available if RemoveUnMatchedMembers=True)	Supports List	True
	<a href="#">RequiredGroups</a>	Supports List	
	RequiredRecursiveGroups	Supports List	
	RequiredOrGroups	Supports List	
	RequiredOrRecursiveGroups	Supports List	
	NotRequiredGroups	Supports List	
	NotRequiredRecursiveGroups	Supports List	
	DisableNameOrIDCheckingForGroupRules		False
	DontAllowMissingNotRequiredGroups		False
	IncludeDistributionGroupsInGroupRules		True
	<a href="#">DomainFilter</a>	Supports List	
	<a href="#">RequiredOrOUs</a>	Supports List	*, if NotRequiredOUs provided
	NotRequiredOUs	Supports List	
	<a href="#">OnFailureApplytoAllUsersOnNextRun</a>	Supports List	False
	<a href="#">Default</a>		False
	<a href="#">CollectionID</a>		100 (or 10000 if Default is True)
	CollectionIDIgnoreThreshold		0
	CollectionIDsToExclude	Supports Lists and Ranges	
[TeacherTeamsPackages1] to [TeacherTeamsPackages100]	<a href="#">ApplyTeamsPackage</a>		False
	TeamsPackageDescription		
	TeamsPackage	Required if ApplyTeamsPackage=True	
	OnlyChangeTeamsPackageIfBlank		True
	ApplyTeamsPaketestoAllTeachers		False
	OnlyApplytoUnlicensedTeachers		False
	ProcessDisabledUsers		False
	SkipUpdatedLicensingCheck		False
	<a href="#">Filter</a>	Supports Expression	
	FilterSource		See <a href="#">Filter Sources</a>
	<a href="#">RequiredGroups</a>	Supports List	
	RequiredRecursiveGroups	Supports List	
	RequiredOrGroups	Supports List	
	RequiredOrRecursiveGroups	Supports List	
	NotRequiredGroups	Supports List	
	NotRequiredRecursiveGroups	Supports List	
	DisableNameOrIDCheckingForGroupRules		False
	DontAllowMissingNotRequiredGroups		False
	IncludeDistributionGroupsInGroupRules		True
	<a href="#">DomainFilter</a>	Supports List	
	<a href="#">RequiredOrOUs</a>	Supports List	*, if NotRequiredOUs provided
	NotRequiredOUs	Supports List	
	<a href="#">RemoveFailedUsersFromLicensingGroup</a>	Supports List	
	<a href="#">OnFailureApplytoAllUsersOnNextRun</a>	Supports List	False
	<a href="#">Default</a>		False
	<a href="#">CollectionID</a>		100 (or 10000 if Default is True)
	CollectionIDIgnoreThreshold		0
	CollectionIDsToExclude	Supports Lists and Ranges	
[StudentTeamsPackages1] to [StudentTeamsPackages100]	<a href="#">ApplyTeamsPackage</a>		False
	TeamsPackageDescription		
	TeamsPackage	Required if ApplyTeamsPackage=True	
	OnlyChangeTeamsPackageIfBlank		True
	ApplyTeamsPaketestoAllStudents		False
	OnlyApplytoUnlicensedStudents		False
	ProcessDisabledUsers		False
	SkipUpdatedLicensingCheck		False
	<a href="#">Filter</a>	Supports Expression	
	FilterSource		See <a href="#">Filter Sources</a>
	<a href="#">RequiredGroups</a>	Supports List	
	RequiredRecursiveGroups	Supports List	
	RequiredOrGroups	Supports List	
	RequiredOrRecursiveGroups	Supports List	
	NotRequiredGroups	Supports List	
	NotRequiredRecursiveGroups	Supports List	
	DisableNameOrIDCheckingForGroupRules		False
	DontAllowMissingNotRequiredGroups		False



	IncludeDistributionGroupsInGroupRules		True
	<a href="#">DomainFilter</a>	Supports List	
	<a href="#">RequiredOrOUs</a>	Supports List	*, if NotRequiredOUs provided
	NotRequiredOUs	Supports List	
	<a href="#">RemoveFailedUsersFromLicensingGroup</a>	Supports List	
	<a href="#">OnFailureApplytoAllUsersOnNextRun</a>	Supports List	False
	Default		False
	<a href="#">CollectionID</a>		100 (or 10000 if Default is True)
	CollectionIDIgnoreThreshold		0
	CollectionIDsToExclude	Supports Lists and Ranges	
[TeacherTeamsPolicies1] to [TeacherTeamsPolicies100]	<a href="#">ApplyTeamsPolicies</a>		False
	TeamsPoliciesDescription		
	TeamsMessagingPolicy		
	OnlyChangeTeamsMessagingPolicyIfBlank		True
	TeamsMeetingPolicy		
	OnlyChangeTeamsMeetingPolicyIfBlank		True
	TeamsAppSetupPolicy		
	OnlyChangeTeamsAppSetupPolicyIfBlank		True
	TeamsCallingPolicy		
	OnlyChangeTeamsCallingPolicyIfBlank		True
	TeamsMeetingBroadcastPolicy		
	OnlyChangeTeamsMeetingBroadcastPolicyIfBlank		True
	TeamsChannelsPolicy		
	OnlyChangeTeamsChannelsPolicyIfBlank		True
	TeamsAppPermissionPolicy		
	OnlyChangeTeamsAppPermissionPolicyIfBlank		True
	TeamsFeedbackPolicy		
	OnlyChangeTeamsFeedbackPolicyIfBlank		True
	ApplyTeamsPoliciesToAllTeachers		False
	OnlyApplytoUnlicensedTeachers		False
	<a href="#">ProcessDisabledUsers</a>		False
	<a href="#">SkipUpdatedLicensingCheck</a>		False
	<a href="#">Filter</a>	Supports Expression	
	FilterSource		See <a href="#">Filter Sources</a>
	<a href="#">RequiredGroups</a>	Supports List	
	RequiredRecursiveGroups	Supports List	
	RequiredOrGroups	Supports List	
	RequiredOrRecursiveGroups	Supports List	
	NotRequiredGroups	Supports List	
	NotRequiredRecursiveGroups	Supports List	
	DisableNameorIDCheckingForGroupRules		False
	DontAllowMissingNotRequiredGroups		False
	IncludeDistributionGroupsInGroupRules		True
	<a href="#">DomainFilter</a>	Supports List	
	<a href="#">RequiredOrOUs</a>	Supports List	*, if NotRequiredOUs provided
	NotRequiredOUs	Supports List	
	<a href="#">RemoveFailedUsersFromLicensingGroup</a>	Supports List	
	<a href="#">OnFailureApplytoAllUsersOnNextRun</a>	Supports List	False
	Default		False
	<a href="#">CollectionID</a>		100 (or 10000 if Default is True)
	CollectionIDIgnoreThreshold		0
	CollectionIDsToExclude	Supports Lists and Ranges	
[StudentTeamsPolicies1] to [StudentTeamsPolicies100]	<a href="#">ApplyTeamsPolicies</a>		False
	TeamsPoliciesDescription		
	TeamsMessagingPolicy		
	OnlyChangeTeamsMessagingPolicyIfBlank		True
	TeamsMeetingPolicy		
	OnlyChangeTeamsMeetingPolicyIfBlank		True
	TeamsAppSetupPolicy		
	OnlyChangeTeamsAppSetupPolicyIfBlank		True
	TeamsCallingPolicy		
	OnlyChangeTeamsCallingPolicyIfBlank		True
	TeamsMeetingBroadcastPolicy		
	OnlyChangeTeamsMeetingBroadcastPolicyIfBlank		True
	TeamsChannelsPolicy		
	OnlyChangeTeamsChannelsPolicyIfBlank		True
	TeamsAppPermissionPolicy		
	OnlyChangeTeamsAppPermissionPolicyIfBlank		True
	TeamsFeedbackPolicy		
	OnlyChangeTeamsFeedbackPolicyIfBlank		True
	<a href="#">ProcessDisabledUsers</a>		False
	<a href="#">SkipUpdatedLicensingCheck</a>		False
	ApplyTeamsPoliciesToAllStudents		False
	OnlyApplytoUnlicensedStudents		False
	<a href="#">Filter</a>	Supports Expression	
	FilterSource		See <a href="#">Filter Sources</a>
	<a href="#">RequiredGroups</a>	Supports List	



	RequiredRecursiveGroups	Supports List	
	RequiredOrGroups	Supports List	
	RequiredOrRecursiveGroups	Supports List	
	NotRequiredGroups	Supports List	
	NotRequiredRecursiveGroups	Supports List	
	DisableNameorIDCheckingForGroupRules		False
	DontAllowMissingNotRequiredGroups		False
	IncludeDistributionGroupsInGroupRules		True
	<a href="#">DomainFilter</a>	Supports List	
	<a href="#">RequiredOrOUs</a>	Supports List	*, if NotRequiredOUs provided
	NotRequiredOUs	Supports List	
	<a href="#">RemoveFailedUsersFromLicensingGroup</a>	Supports List	
	<a href="#">OnFailureApplytoAllUsersOnNextRun</a>	Supports List	False
	Default		False
	<a href="#">CollectionID</a>		100 (or 10000 if Default is True)
	CollectionIDIgnoreThreshold		0
	CollectionIDsToExclude	Supports Lists and Ranges	
[InformationRetrieval]	<a href="#">GetAllUserPolicyInformation</a>		False
	<a href="#">Filter</a>	Supports Expression	
	FilterSource		See <a href="#">Filter Sources</a>
	<a href="#">GetUserPackageInformation</a>		
	<a href="#">GetUniqueUserDepartmentNames</a>		False
	<a href="#">GetPolicyJobs</a>		False
	<a href="#">RequiredGroups</a>	Supports List	
	RequiredRecursiveGroups	Supports List	
	RequiredOrGroups	Supports List	
	RequiredOrRecursiveGroups	Supports List	
	NotRequiredGroups	Supports List	
	NotRequiredRecursiveGroups	Supports List	
	DisableNameorIDCheckingForGroupRules		False
	DontAllowMissingNotRequiredGroups		False
	IncludeDistributionGroupsInGroupRules		False
	<a href="#">DomainFilter</a>	Supports List	
	<a href="#">RequiredOrOUs</a>	Supports List	*, if NotRequiredOUs provided
	NotRequiredOUs	Supports List	
[ResultsEmail]	FromAddress		<GA_Username>
	ToAddress		<GA_Username>
	UseSSL		True
	Port		587
	Server		smtp.office365.com
	DisableEmail		False
	OnlySendEmailonWarningorError		False
[PowerShell]	<a href="#">AutoSetTLSecurity</a>		True
	DisableNuGetCheck		False
	DisablePowerShellGetCheck		False
	DisableAzureADPreviewCheck		False
	DisableMSOnlineCheck		False
	DisableMicrosoftTeamsCheck		False
	DisableExchangeConnectorCheck		False
	<a href="#">DisableMonthlyModuleUpdateChecks</a>		False
	DisableRemoteModuleUpdateRequests		False
	<a href="#">ForceUpdateModules</a>		False
[Remediation]	<a href="#">OnlyRemediateNewTeachers</a>		True
<small>(settings files that don't contain this section will have it automatically added)</small>	<a href="#">OnlyRemediateNewStudents</a>		True
	<a href="#">OnFailureApplytoAllUsersOnNextRun</a>	Supports List	False
[Collections]	<a href="#">IgnoreCollectionIDUserType</a>		False
[FunctionControl]	<a href="#">TurnOffTeacherFilteredSecurityGroupFunctionality</a>		False
<small>(settings files that don't contain this section will have it automatically added)</small>	<a href="#">TurnOffStudentFilteredSecurityGroupFunctionality</a>		False
	<a href="#">TurnOffTeacherFilteredUnifiedGroupFunctionality</a>		False
	<a href="#">TurnOffStudentFilteredUnifiedGroupFunctionality</a>		False
	<a href="#">TurnOffTeacherFilteredDistributionGroupFunctionality</a>		False
	<a href="#">TurnOffStudentFilteredDistributionGroupFunctionality</a>		False
	<a href="#">TurnOffTeacherFilteredPackageFunctionality</a>		False
	<a href="#">TurnOffStudentFilteredPackageFunctionality</a>		False
	<a href="#">TurnOffTeacherFilteredPolicyFunctionality</a>		False
	<a href="#">TurnOffStudentFilteredPolicyFunctionality</a>		False
[Misc]	<a href="#">SleepSecondsAfterApplyingLicensing</a>		600 (max 82800)
	<a href="#">SleepSecondsBeforeApplyingPolicies</a>		0
	<a href="#">DontPauseForChanges</a>		False
	<a href="#">RefreshTeacherLicensingStatusAfterLicensingGroupChanges</a>		False
	<a href="#">RefreshStudentLicensingStatusAfterLicensingGroupChanges</a>		False
	<a href="#">TeacherTeamsPackageBatchProcessingThreshold</a>		99999
	<a href="#">StudentTeamsPackageBatchProcessingThreshold</a>		99999
	<a href="#">TeacherTeamsPolicyBatchProcessingThreshold</a>		99999
	<a href="#">StudentTeamsPolicyBatchProcessingThreshold</a>		99999
	<a href="#">UseLocalAttributeFiltering</a>		False
	<a href="#">UseAzureADforLocalAttributeFilteringByDefault</a>		False
	<a href="#">MaximumBatchSize</a>		5000 (min 20)



	<a href="#">CreateAzureADAccessPolicy</a>		False
	<a href="#">ApplyAzureADAccessPolicyWhenPoliciesAlreadyExist</a>		False
	<a href="#">ThrowWarningInsteadofErrorOnAzureADAccessPolicyFailure</a>		False
	<a href="#">DontCheckOnMicrosoftCOMPrefixWhenRemovingAzureADAccessPolicy</a>		False
	<a href="#">AssumeAzureADAccessPolicyNotRequiredTimeoutHours</a>		24.0 (1-168 decimal)
	<a href="#">CheckforUpdates</a>		True
	<a href="#">DisableAutoUpdate</a>		False
	<a href="#">IgnoreAutoUpdateHash</a>		False
	<a href="#">IgnoreMissingNETFramework</a>		False
	<a href="#">Verbose</a>		False
	<a href="#">DetailedLogFileFolder</a>		STEPPER script location\STEPPER Logs
	<a href="#">DisableDetailedLogFile</a>		False
	<a href="#">GetExtraStatistics</a>		True
	<a href="#">UseCSVforUserDataInLogFile</a>		False (unless Verbose specified)
	<a href="#">DisableSaveSettingsWithLogZIP</a>		False
	<a href="#">CleanUpLogFileOlderThanDays</a>		(disabled)
	<a href="#">RepeatLogEventThreshold</a>		20
	<a href="#">LongJobUpdateInterval</a>		100
	<a href="#">DontRunIfStepperMayBeRunningTimeOutInMinutes</a>		Disabled if not specified; if invalidly specified, the default is 240 minutes (4 hours)
	<a href="#">MaxSettingsGroups</a>		100 (min 10; max 1000)
	<a href="#">SettingsFileVersion</a>		100 (up to v1.0.20) 101 (v1.0.21 to 1.0.25) 102 (v1.0.26 to 1.0.31) 103 (v1.0.32 to 1.0.35) 104 (v1.0.36 to 1.0.43) 105 (v1.0.44 to 1.0.46) 106 (v1.0.47 to 1.0.51) 107 (v1.0.52) 108 (v1.0.53 to 1.0.60) 109 (v1.0.61 to 1.0.63) 110 (v1.0.64 to 1.0.67) 111 (v1.0.68) 112 (v1.0.69 to 1.0.72) 113 (v1.0.73 to 1.0.84) 114 (v1.0.85) 115 (v1.0.86 to 1.0.90) 116 (v1.0.91 to 1.0.93) 117 (v1.0.94 or newer) 118 (v1.0.97 or newer)
(Command Line)	<a href="#">ApplytoAll</a>		False
	<a href="#">ApplytoAllTeachers</a>		False
	<a href="#">ApplytoAllStudents</a>		False
	<a href="#">MakeChanges</a>	Set to False if -Whatif Specified on the command line	False
	<a href="#">SettingsFile</a>		<STEPPERScriptDirectory>\STEPPER.INI
	<a href="#">UpdateModules</a>		False
	<a href="#">Verbose</a>		False
	<a href="#">Whatif</a>		False

\* This indicates the default value or setting if the setting is not specified in the settings file.