

SINGLE MODULE

PowerShell Quick Reference for Microsoft Teams

[Ver. 6.9.0]

Install Module (PS 5+)

Install-Module MicrosoftTeams -Repository PSGallery

Load Teams PS Module

Import-Module MicrosoftTeams -MinimumVersion 4.7.0

Connect to Microsoft Teams with PowerShell

Connect-MicrosoftTeams

List all Commands for Teams

Get-Command -Module MicrosoftTeams

Cmdlets

Current Microsoft Teams count is 577 (2025.03.25)
(Get-Command -Module MicrosoftTeams).Count

List All Teams

Get-Team

Find Single Team

Get-Team -DisplayName 'IT Department'

Create Team

New-Team -MailNickName 'Marketing' -DisplayName 'Marketing' -Visibility 'Private'

Remove a Team

Remove-Team -GroupId '503b7c56-b15d-4b4c-8cca-09103984b2bb'

Modify a Team

Set-Team -GroupId '7a4fab2f-6a22-46d0-826b-a167fdc03892' -Visibility 'Public'

Get-Team

Locate Teams a user is a member of:

Get-Team -User Damian@domain.com

Locate all archived Teams

Get-Team -Archived \$True

Locate all Public Teams

Get-Team -Visibility Public

Locate all Private Teams

Get-Team -Visibility Private

Locate Teams by Mail Nick Name

Get-Team -MailNickName Marketing

Create New Microsoft Team or Channel

New-Team -DisplayName "Marketing Campaign – Big Corp, Inc."

New-Team -DisplayName "Project Rebuild" -Description "First project of 2020 (HR)"

New-Team -DisplayName "New Test Group" -ShowInTeamsSearchAndSuggestions \$False

Creating New TeamChannel (**)

New-TeamChannel -GroupId 933fe926-555a-4832-87d1-8f700736e003 -DisplayName 'Project 007'

New-TeamChannel -GroupId cfdba387-1319-4f6b-a883-8700046f07e7 -DisplayName 'Internal Sales'

** GroupId is from the ID of an existing Team. To see the ID's, run Get-Team

Other Team Settings

Set-TeamsApp -Id bc098240-e8c2-4350-9f90-abe1aa65be4e -Path c:\app.zip

Set-TeamArchivedState -GroupId 8981a765-eecc-4f1b-bd38-54d23bc7bc42 -Archived:\$False

Set-TeamPicture -GroupId c11c57b5-f572-4074-b347-962555999cdb -ImagePath 'c:\images\MarketingLogo.jpg'

** Note that the ImagePath cannot be set to \$Null and there is no Remove-TeamPicture cmdlet. Choose images wisely.

PowerShell Gallery: <https://www.powershellgallery.com/packages/MicrosoftTeams/>

Documentation: <https://docs.microsoft.com/en-us/powershell/module/teams/>

Configuring Team Settings

Configuring Team Settings

Set-Team -GroupId d36f235f-d30e-4460-98a2-5728b906fbfd -GiphyContentRating Strict -AllowStickersAndMemes \$False

Set-Team -GroupId cfdba387-1319-4f6b-a883-8700046f07e7 -AllowCreateUpdateChannels \$True

Set-Team -GroupId cfdba387-1319-4f6b-a883-8700046f07e7 -AllowUserDeleteMessages \$False

Set-Team -GroupId cfdba387-1319-4f6b-a883-8700046f07e7 -ShowInTeamsSearchAndSuggestions \$False

Set-TeamChannel -GroupId cfdba387-1319-4f6b-a883-8700046f07e7 -CurrentDisplayName "Internal Sales" -Description "Sales Channel"

Get-Team -DisplayName 'HR Department' | Set-Team -AllowGuestCreateUpdateChannels \$False

Get-Team -DisplayName 'Marketing' | Set-Team -AllowTeamMentions \$False

Set-Team -GroupId c5f14935-f6b8-4d85-9b5c-2fc61979cec3 -AllowAddRemoveApps \$True

Set-TeamArchivedState -GroupId 9ee34e55-60d6-49cf-8a64-1bc1454d7ee4 -Archived:\$true

Get-Help

Getting Help

Get-Help <command>

Get-Help <command> -Examples

Get-Help <command> -Full

Examples

Get-Help New-Team

Get-Help New-Team -Examples

Get-Help New-Team -Full

Verify Settings

Get-Team

Get-TeamChannel

Get-TeamsApp

Get-TeamUser

Microsoft Teams Cleanup

Remove-Team -GroupId '933fe926-555a-4832-87d1-8f700736e003'

Remove-TeamChannel -GroupId cfdba387-1319-4f6b-a883-8700046f07e7 -DisplayName "Internal Sales"

Remove-TeamUser -GroupId '933fe926-555a-4832-87d1-8f700736e003' -User <UPN>

TIP: You may need this for custom scripts > Set-ExecutionPolicy -ExecutionPolicy Unrestricted

Cmdlet Notes

Get-Team can now use GroupID, DisplayName, MailNickName and Visibility for locating Teams.

Simple Microsoft Teams Scenario

```
# Connect to Teams PowerShell in your O365 tenant
Connect-MicrosoftTeams -AccountId Admin@ThisIsYourDomain.Com
# Create new Microsoft Team
$Team = New-Team -MailNickName "ProjectX" -DisplayName "ProjectX" -Visibility "Private" -Description "Project X: Experimental Project"
# Add Team Channels
New-TeamChannel -GroupId $Team.GroupId -DisplayName "Production"
# Add Team members
Add-TeamUser -GroupId $Team.GroupId -User "JohnSmith@ThisIsYourDomain.Com"
# Change the default fun settings for the team
Set-Team -GroupId $Team.GroupId -GiphyContentRating Strict -AllowStickersAndMemes $False -AllowCustomMemes $False
# Set a Team, if private, is searchable
Set-TeamPicture -GroupId $Team.GroupId -ShowInTeamsSearchAndSuggestions $True
```

Complex Microsoft Teams Scenario

```
# Connect to Teams PowerShell in your O365 tenant
Connect-MicrosoftTeams -AccountId Admin@ThisIsYourDomain.Com
# Create new Microsoft Team
$Marketing = New-Team -MailNickName "ProjectCodeX" -DisplayName "ProjectCodeX" -AccessType "Public" -Description "Marketing Dept. Team"
$SalesTeam = New-Team -MailNickName "Sales" -DisplayName "Sales" -Visibility "Private" -Description "Sales Dept. Team"
$ITTeam = New-Team -MailNickName "ITDept" -DisplayName "IT Dept." -Visibility "Private" -Description "IT Dept. Collaboration"
$ManagementTeam = New-Team -MailNickName "Management" -DisplayName "Management" -Visibility "Private" -Description "Management Team"
# Add Team Channels
New-TeamChannel -GroupId $Marketing.GroupId -DisplayName "2018 Campaign"
New-TeamChannel -GroupId $SalesTeam.GroupId -DisplayName "Midwest US Region"
New-TeamChannel -GroupId $SalesTeam.GroupId -DisplayName "West Coast US Region"
New-TeamChannel -GroupId $SalesTeam.GroupId -DisplayName "East US Region"
New-TeamChannel -GroupId $ITTeam.GroupId -DisplayName "End User Support"
New-TeamChannel -GroupId $ManagementTeam.GroupId -DisplayName "Annual Planning"
# Add Team members
$MsolUsers = Get-MsolUser -all
Foreach ($MsolUser in $MsolUsers) {
    $Department = $MsolUser.Department
    $Email = $MsolUser.PrimarySMTPAddress
    If ($Department -eq 'Marketing') {Add-TeamUser -GroupId $MarketingTeam.GroupId -User $Email}
    If ($Department -eq 'Sales') {Add-TeamUser -GroupId $SalesTeam.GroupId -User $Email}
    If ($Department -eq 'IT') {Add-TeamUser -GroupId $ITTeam.GroupId -User $Email}
    If ($Department -eq 'Management') {Add-TeamUser -GroupId $ManagementTeam.GroupId -User $Email}
}
# Change the default fun settings for the team
Set-Team -GroupId $MarketingTeam.GroupId -AllowStickersAndMemes $True -AllowCustomMemes $True
Set-Team -GroupId $SalesTeam.GroupId -AllowStickersAndMemes $True -AllowCustomMemes $True
Set-Team -GroupId $ITTeam.GroupId -GiphyContentRating Strict -AllowStickersAndMemes $False -AllowCustomMemes $False
Set-Team -GroupId $ManagementTeam.GroupId -GiphyContentRating Strict -AllowStickersAndMemes $False -AllowCustomMemes $False
# Set a Team, if private, is searchable
Set-Team -GroupId d36f235f-d30e-4460-98a2-5728b906fbfd -ShowInTeamsSearchAndSuggestions $True
Set-Team -GroupId cfd8a387-1319-4f6b-a883-8700046f07e7 -ShowInTeamsSearchAndSuggestions $False
Set-Team -GroupId a4d3425f-98b2-420d-a0a0-7c3eff32493f -ShowInTeamsSearchAndSuggestions $True
Set-Team -GroupId 933fe926-555a-4832-87d1-8f700736e003 -ShowInTeamsSearchAndSuggestions $False
```

Created By:**Damian Scoles**

Microsoft MVP

Book Author

www.practicalpowershell.comwww.powershellgeek.com

@PPowerShell

Helpful Tips

Use tab to autocomplete cmdlets
Tab through parameters to see all available
Check for latest module version
Read the latest Microsoft Docs for Teams
Read Teams MVP blogs for more tips

**More On PowerShell****Windows PowerShell Blog**<https://devblogs.microsoft.com/scripting/>**Script Center**technet.microsoft.com/scriptcenter**PowerShell Books**www.practicalpowershell.com**PowerShell Blog**www.powershellgeek.com**PowerShell Team – GitHub**<https://github.com/powershell>

Cortana Policy

Retrieve existing Cortana Policies

Get-CsTeamsCortanaPolicy

Assign Policies to Users [HR Department in this example]:

```
$GroupMembers = (Get-AzureADGroup | Where {$_.DisplayName -eq 'HR Department'} |
    Get-AzureADGroupMember).UserPrincipalName
Foreach ($GroupMember in $GroupMembers) {
    Grant-CsTeamsCortanaPolicy -identity $GroupMember -PolicyName SecuredTeam
}
```

Create new Cortana Policies

```
New-CsTeamsCortanaPolicy -Identity ITTesting -CortanaVoiceInvocationMode PushToTalkUserOverride
New-CsTeamsCortanaPolicy -Identity SecuredTeams -CortanaVoiceInvocationMode Disabled
```

Remove existing Cortana Policies

```
Remove-CsTeamsCortanaPolicy -Identity SecuredTeams
Remove-CsTeamsCortanaPolicy -Identity ITTesting
```

Change existing Cortana Policies

```
Set-CsTeamsCortanaPolicy Global -AllowCortanaVoiceInvocation $False -
AllowCortanaAmbientListening $False -AllowCortanaInContextSuggestions $False
```

Teams Feedback Policy

Disable a Teams Feedback Policy

Get-CsTeamsFeedbackPolicy

Assign a Teams Feedback Policy

```
Grant-CsTeamsFeedbackPolicy -PolicyName 'C-Level Feedback Policy' -Identity Damian
```

List existing Teams Feedback Policy

```
New-CsTeamsFeedbackPolicy -Identity 'C-Level Feedback Policy' -AllowLogCollection $False
-ReceiveSurveysMode Enabled
```

Remove an existing Teams Feedback Policy

```
Remove-CsTeamsFeedbackPolicy 'C-Level Feedback Policy'
```

Change settings on an existing Teams Feedback Policy

```
Set-CsTeamsFeedbackPolicy -Identity 'C-Level Feedback Policy' -AllowEmailCollection $True
```

Teams Holidays

List current Holidays

Get-CsOnlineSchedule

```
Get-CsOnlineSchedule | Where {$_.Type -eq 'Fixed'}
```

Create new Holiday

```
$Date = New-CsOnlineDateTimeRange -Start '4/1/2025 0:00' -End '4/1/2025 23:45'
New-CsOnlineSchedule -Name 'Employee Day' -DateRanges $Date -FixedSchedule
```

Change existing Holiday

```
$Schedule = Get-CsOnlineSchedule 93ce710a-f0b6-4c77-80eb-45c73228aa8b
```

```
$Schedule.Name = 'New Years Party'
```

```
Set-CsOnlineSchedule -Instance $Schedule
```

Remove existing Holiday

```
Remove-CsOnlineSchedule -Id 15f9c478-89a3-4052-9b1b-50eb87e769cf
```

Teams Compliance Recording

Search for Application Instances

```
Find-CsOnlineApplicationInstance -SearchQuery 'Auto'
```

List Application Instances

Get-CsOnlineApplicationInstance

Create a new Application Instance in Azure AD

```
$Guid = (New-Guid).Guid
New-CsOnlineApplicationInstance -UserPrincipalName 'TeamsRecording@domain.com'
-ApplicationID $Guid -DisplayName 'Teams Compliance Recording'
```

Change settings on an existing Online Application Instance

```
Set-CsOnlineApplicationInstance -Identity 'TeamsRecording@domain.com' -DisplayName
'Compliance Recording' -ApplicationId 3b12c192-78c6-4c07-90f7-4be371f7a8f5
```

Sync Application Instances to Agent Provisioning Service

```
Sync-CsOnlineApplicationInstance -ObjectId 23beeda4-9bd8-4384-8521-9bed0ee3bbd7
-ApplicationId 3b12c192-78c6-4c07-90f7-2be371f7a8f5
```

List any existing Compliance Recording Policies

Get-CsTeamsComplianceRecordingPolicy

Assign any existing Compliance Recording Policies

```
Grant-CsTeamsComplianceRecordingPolicy -Identity 'Damian' -PolicyName 'CorpRecordingPolicy'
```

Remove Assignment of Compliance Recording Policy From a User

```
Grant-CsTeamsComplianceRecordingPolicy -Identity 'Damian' -PolicyName $Null
```

Create a new Compliance Recording Policies

```
New-CsTeamsComplianceRecordingPolicy -Identity 'CorpRecordingPolicy' -Enabled
$True -ComplianceRecordingApplications @(New-CsTeamsComplianceRecordingApplication
-Parent 'TeamsRecording@domain.com' -Id '$Guid')
```

Remove an existing Compliance Recording Policies

```
Remove-CsTeamsComplianceRecordingPolicy 'CorpRecordingPolicy'
```

Change settings on an existing Compliance Recording Policies

```
Set-CsTeamsComplianceRecordingPolicy -Identity 'CorpRecordingPolicy' -Enabled $False
```

List any existing Compliance Recording Application

Get-CsTeamsComplianceRecordingApplication

Create a new Compliance Recording Application

```
New-CsTeamsComplianceRecordingApplication -Identity "Tag:CorpRecordingPolicy/$Guid"
```

Remove an existing Compliance Recording Application

```
Remove-CsTeamsComplianceRecordingApplication -Identity
"Tag:BigBoxComplianceRecordingPolicy/$Guid"
```

Change settings on an existing Compliance Recording Application

```
Set-CsTeamsComplianceRecordingApplication -Identity "Tag:BigBoxComplianceRecordingPolicy/
$Guid" -RequiredBeforeMeetingJoin $True -RequiredDuringMeeting $False
```

Create a new pairing of Compliance Recording Applications

```
New-CsTeamsComplianceRecordingPairedApplication -Id $Guid
```

Teams Client Configuration

List current Client Configuration settings

Get-CsTeamsClientConfiguration

Change Client Configuration Settings

```
Set-CsTeamsClientConfiguration -Identity Global -AllowBox $False -AllowGoogleDrive $False -
AllowEgnyte $False -AllowShareFile $False
```

Teams Meeting Policy

List any existing Meeting Policies

```
Get-CsTeamsMeetingPolicy
Get-CsTeamsMeetingPolicy | where {$_.Description -NotLike 'Do not assign*'} | ft
```

Assign Meeting Policies

```
Grant-CsTeamsMeetingPolicy -identity 'Jsmith' -PolicyName 'ITDeptMeetings'
```

Create a new Teams Meeting Policy

```
New-CsTeamsMeetingPolicy -Identity RnDPolicy -AllowTranscription $True
New-CsTeamsMeetingPolicy -Identity Marketing Policy -AutoAdmittedUsers "Everyone"
-AllowBreakoutRooms $False
```

Remove an existing Teams Meeting Policy

```
Remove-CsTeamsMeetingPolicy 'RnDPolicy'
```

Change settings on an existing Teams Meeting Policy

```
Set-CsTeamsMeetingPolicy -Identity 'ITPolicy' -AllowBreakoutRooms $False
Set-CsTeamsMeetingPolicy -Identity 'SalesPolicy' -AllowMeetNow $True
```

Teams Meeting Configuration

List current Meeting Configuration Settings

```
Get-CsTeamsMeetingConfiguration
```

Change Meeting Configuration settings

```
Set-CsTeamsMeetingConfiguration -LogoURL www.bixbox.com/BBLogo.jpg
Set-CsTeamsMeetingConfiguration -LegalURL www.bixbox.com/legal.html
```

Teams Guest Meeting Configuration

List current Guest Meeting Configuration Settings

```
Get-CsTeamsGuestMeetingConfiguration
Get-CsTeamsGuestMeetingConfiguration | Ft
```

Change Guest Meeting Configuration settings

```
Set-CsTeamsGuestMeetingConfiguration -ScreenSharingMode 'Single Application'
Set-CsTeamsGuestMeetingConfiguration -AllowMeetNow $Trues
```

Teams Meeting Policy

List current Teams Meeting Broadcast policies

```
Get-CsTeamsMeetingBroadcastPolicy
```

Assign Teams Meeting Broadcast policy to user

```
Grant-CsTeamsMeetingBroadcastPolicy -Global -Identity 'Grant Merks'
```

Create a new Teams Meeting Broadcast policy

```
New-CsTeamsMeetingBroadcastPolicy -Identity Management -AllowBroadcastTranscription $True
```

Remove current Teams Meeting Broadcast policies

```
Remove-CsTeamsMeetingBroadcastPolicy 'Old Corp Policy'
```

Change settings on current Teams Meeting Broadcast policies

```
Set-CsTeamsMeetingBroadcastPolicy -Global -BroadcastRecordingMode 'AlwaysDisabled'
```

List global settings on current Teams Meeting Broadcast

```
Get-CsTeamsMeetingBroadcastConfiguration
```

Change global settings for Teams Meeting Broadcasts

```
Set-CsTeamsMeetingBroadcastConfiguration -Identity Global -EnableAnonymousBroadcastMeeting
$True -EnableBroadcastMeetingRecording $True
```

Teams VDI Policy

List any existing VDI policies

```
Get-CsTeamsVdiPolicy
```

Assign a VDI Policy to a user

```
Grant-CsTeamsVdiPolicy -Identity 'Remote Users VDI' -Identity Damian
```

Create a new VDI Policy

```
New-CsTeamsVdiPolicy -PolicyName 'Remote Users VDI' -DisableCallsAndMeetings $True
```

Remove an existing VDI Policy

```
Remove-CsTeamsVdiPolicy 'Remote Users VDI'
```

Change settings on an existing VDI Policy

```
Set-CsTeamsVdiPolicy -Identity 'Remote Users VDI' -DisableAudioVideoInCallsAndMeetings $True
```

Teams Application Settings

App Permission Policies

"The existence of this cmdlet is being documented for completeness, but do not use this cmdlet. We require that all creation and modification of app permission polices (not including the assignment or removal of policies from users) happens in the Microsoft Teams & Skype for Business Admin Center to ensure that the policy matches your expectations for the end user experience."

Cmdlets for app setup policies:

```
Get-CsTeamsAppPermissionPolicy
Grant-CsTeamsAppPermissionPolicy
New-CsTeamsAppPermissionPolicy
Remove-CsTeamsAppPermissionPolicy
Set-CsTeamsAppPermissionPolicy
```

App Setup Policies

"The existence of this cmdlet is being documented for completeness, but do not use this cmdlet. We require that all creation and modification of app setup polices (not including the assignment or removal of policies from users) happens in the Microsoft Teams & Skype for Business Admin Center to ensure that the policy matches your expectations for the end user experience."

Cmdlets for app setup policies:

```
Get-CsTeamsAppSetupPolicy
Grant-CsTeamsAppSetupPolicy
New-CsTeamsAppSetupPolicy
Remove-CsTeamsAppSetupPolicy
Set-CsTeamsAppSetupPolicy
```

Link to some Help URLs for Teams Apps cmdlets:

<https://docs.microsoft.com/en-us/powershell/module/skype/Set-CsTeamsAppSetupPolicy>
<https://docs.microsoft.com/en-us/powershell/module/skype/new-csteamsappsetuppolicy>

Teams Call Policies

Import audio file for hold music

```
$AudioFileForHold = Get-Content 'C:\Media\ElevatorMusak.wav' -Encoding byte -ReadCount 0
$NewAudioFile = Import-CsOnlineAudioFile -ApplicationId 'CorpAutoAttendant' -FileName
'ElevatorMusak.wav' -Content $AudioFileForHold
```

Import audio file for hold music

```
$AudioFileForHold = Get-Content 'C:\Source\ElevatorMusak.wav' -Encoding byte -ReadCount 0
$NewAudioFile = New-CsOnlineAudioFile -FileName 'ElevatorMusak.wav' -Content
$AudioFileForHold
```

List existing Call Hold Policies

```
Get-CsTeamsCallHoldPolicy
```

Assign existing Call Hold Policies

```
Grant-CsTeamsCallHoldPolicy -Identity 'damian@practicalpowershell.com' -PolicyName 'BigBox-
HoldPolicy'
```

Create new Call Hold Policy

```
New-CsTeamsCallHoldPolicy -Identity 'BigBox-HoldPolicy' -Description 'Elevator Musak' -AudioFileID
"522caf-c807a7-4b43a6-d72c11"
```

*** AudioFileID from Import-CsOnlineAudioFile or New-CsOnlineAudioFile cmdlets ***

Remove existing Call Hold Policies

```
Remove-CsTeamsCallHoldPolicy 'BigBox-HoldPolicy'
```

Change settings on existing Call Hold Policies

```
Set-CsTeamsCallHoldPolicy -Identity 'BigBox-HoldPolicy' -Description 'New Music 2021'
```

List existing Calling Policies

```
Get-CsTeamsCallingPolicy
```

Assign an existing Calling Policies

```
Grant-CsTeamsCallingPolicy -identity "Damian" -PolicyName 'ITCallingPolicy'
```

Create a new Calling Policies

```
New-CsTeamsCallingPolicy -Identity 'ITCallingPolicy' -AllowPrivateCalling $False -AllowVoicemail
AlwaysEnabled -AllowDelegation $False
```

Remove an existing Calling Policies

```
Remove-CsTeamsCallingPolicy 'ITCallingPolicy'
```

Change settings on existing Calling Policies

```
Set-CsTeamsCallingPolicy -Identity 'ITCallingPolicy' -AllowCallForwardingToUser $False
```

List existing Call Park Policies

```
Get-CsTeamsCallParkPolicy
```

List existing Call Park Policies

```
Grant-CsTeamsCallParkPolicy -Identity Damian
```

List existing Call Park Policies

```
New-CsTeamsCallParkPolicy -Identity 'MarketingHoldPolicy' -AllowCallPark $True
```

List existing Call Park Policies

```
Remove-CsTeamsCallParkPolicy 'MarketingHoldPolicy'
```

List existing Call Park Policies

```
Set-CsTeamsCallParkPolicy -Identity 'MarketingHoldPolicy' -Description 'Marketing Hold Policy'
```

Emergency Calls

List Existing Emergency Calling Policies

```
Get-CsTeamsEmergencyCallingPolicy
```

Assign Existing Emergency Calling Policies

```
Grant-CsTeamsEmergencyCallingPolicy -Identity Dave -PolicyName CorpECRP
```

Create a New Emergency Calling Policies

```
New-CsTeamsEmergencyCallingPolicy -Identity CorpECRP -Description "Corporate ECRP"
-NotificationGroup "emergence@practicalpowershell.com" -NotificationDialOutNumber
"3125551212" -NotificationMode NotificationOnly -ExternalLocationLookupMode $True
```

Create a New Emergency Calling Policies

```
Remove-CsTeamsEmergencyCallingPolicy -PolicyName CorpECRP
```

Change Settings on Existing Emergency Calling Policies

```
Set-CsTeamsEmergencyCallingPolicy -Identity CorpECRP -NotificationDialOutNumber 8475551212
```

Create new Teams Emergency Number(s)

```
New-CsTeamsEmergencyNumber -EmergencyDialString 811 -EmergencyDialMask 456
-OnlinePSTNUsage "US911"
```

List Existing Emergency Calling Policies

```
Get-CsTeamsEmergencyCallRoutingPolicy
```

Assign Existing Emergency Calling Policies

```
Grant-CsTeamsEmergencyCallRoutingPolicy
```

Create a New Emergency Calling Policies

```
$TeamsENumber = New-CsTeamsEmergencyNumber -EmergencyDialString 811
-EmergencyDialMask 456 -OnlinePSTNUsage "US911"
```

```
New-CsTeamsEmergencyCallRoutingPolicy -Identity "CorpECallRouting" -EmergencyNumbers
@{add=$TeamsENumber} -AllowEnhancedEmergencyServices:$true -Description "Teams Number 1"
```

Remove an Existing Emergency Calling Policies

```
Remove-CsTeamsEmergencyCallRoutingPolicy "CorpECallRouting"
```

Change Settings for an Existing Emergency Calling Policies

```
Set-CsTeamsEmergencyCallRoutingPolicy -Identity "CorpECallRouting" -Description "Corp
Emergency Number Call Routing"
```

Guest Calling Policies

List existing Guest Calling Policies

```
Get-CsTeamsGuestCallingConfiguration
```

Change settings on existing Guest Calling Policies

```
Set-CsTeamsGuestCallingConfiguration -AllowPrivateCalling $True
```

Education Assignment App Policies

List any existing Education Assignment App Policies

```
Get-CsTeamsEducationAssignmentsAppPolicy
```

Change settings on an existing Education Assignment App Policies

```
Set-CsTeamsEducationAssignmentsAppPolicy -Identity Global -ParentDigestEnabledType $True
```

TIP

-WhatIf

Use what if to see what action(s) the PowerShell cmdlet would have performed without the switch.

Teams Messaging Policies

Get-CsTeamsMessagingPolicy
Assign use an existing Teams Messaging Policy
Grant-CsTeamsMessagingPolicy
Create a new Teams Messaging Policy
New-CsTeamsMessagingPolicy -Identity 'ITDepartment' -AllowOwnerDeleteMessage \$True
Remove an existing Teams Messaging Policy
Remove-CsTeamsMessagingPolicy 'ITDepartment'
Change settings for any Teams Messaging Policy
Set-CsTeamsMessagingPolicy ItDepartment -ReadReceiptsEnabledType Everyone
Get-CsTeamsGuestMessagingConfiguration
Change settings for the Teams Guest Messaging Policy
Set-CsTeamsGuestMessagingConfiguration -AllowGiphy \$True -AllowStickers \$True

Teams Mobility Policies

Get-CsTeamsMobilityPolicy
Assign any existing Mobility Policy
Grant-CsTeamsMobilityPolicy -Identity Damian -PolicyName 'Marketing'
Create new Teams Mobility Policy
New-CsTeamsMobilityPolicy -Identity 'Marketing' -IPVideoMobileMode WiFiOnly
Remove existing Teams Mobility Policy
Remove-CsTeamsMobilityPolicy 'Marketing'
Change settings on any existing Teams Mobility Policy
Set-CsTeamsMobilityPolicy 'Marketing' -IPAudioMobileMode WifiOnly

Teams Shifts Policies

Get-CsTeamsShiftsPolicy
Assign an existing Teams Shifts Policy to a user
Grant-CsTeamsShiftsPolicy -Identity Ezeikel -PolicyName 'Warehouse1'
Create new Teams Shift Policy
New-CsTeamsShiftsPolicy -Identity 'Warehouse1' -EnableShiftPresence \$True
Remove an existing Teams Shift Policy
Remove-CsTeamsShiftsPolicy
Change Settings on an existing Teams Shift Policy
Set-CsTeamsShiftsPolicy 'Warehouse2' -AccessType OffShift_TeamsAppBlocked

List Teams Shift App Policy
Get-CsTeamsShiftsAppPolicy
Change Settings on a Teams Shift App Policy (only works on Global policy)
Set-CsTeamsShiftsAppPolicy 'Default' -AllowTimeClockLocationDetection \$False

Teams Targeting Policies

Get-CsTeamsTargetingPolicy
Remove an existing Policy – Default policy is a Global policy, do not use. Also cannot create new.
Remove-CsTeamsTargetingPolicy
Change settings on existing policy
Set-CsTeamsTargetingPolicy Global -CustomTagsMode Enabled

Teams Call Policies

Configure workloads like meeting, messaging, calling that are enabled and/or pinned for the user:

List all existing Workload Policies
Get-CsTeamsWorkLoadPolicy
Assign existing Workload Policy to a user
Grant-CsTeamsWorkLoadPolicy -PolicyName Corporate -Identity damian@practicalpowershell.com
Create new Workload Policies
New-CsTeamsWorkLoadPolicy -Identity Corporate
Remove existing Workload Policies
Remove-CsTeamsWorkLoadPolicy -Identity Corporate
Change settings on existing Workload Policies
Set-CsTeamsWorkLoadPolicy -Identity Corporate -AllowCallingPinned \$False

Teams IP Phone Policies

List existing IP Phone Policies
Get-CsTeamsIPPhonePolicy
Assign an IP Phone Policy to a user
Grant-CsTeamsIPPhonePolicy -Identity Damian -PolicyName ResearchAndDev
Create a new IP Phone Policy
New-CsTeamsIPPhonePolicy -Identity 'ResearchAndDev' -AllowHotDesking \$False
Remove an existing IP Phone Policy
Remove-CsTeamsIPPhonePolicy ResearchAndDev
Change settings on an existing IP Phone Policy
Set-CsTeamsIPPhonePolicy -Identity 'ResearchAndDev' -AllowBetterTogether Disabled

Teams Notification and Feeds Policies

List all existing Teams Notifications and Feeds Policies
Get-CsTeamsNotificationAndFeedsPolicy
Remove an existing Notifications and Feeds Policy [Only Default and Global exist, no New cmdlet]
Remove-CsTeamsNotificationAndFeedsPolicy
Change settings on an existing Notifications and Feeds Policy
Set-CsTeamsNotificationAndFeedsPolicy Global -SuggestedFeedsEnabledType EnabledUserOverride

Teams Translation Rules

List all existing Teams Translation Rules
Get-CsTeamsTranslationRule
Create new Teams Translation Rule
New-CsTeamsTranslationRule -Identity 'USTRanslation' -Pattern '^\d{10}\$' -Translation '+\$1'
Remove an existing Teams Translation Rule
Remove-CsTeamsTranslationRule 'USTRanslation'
Change settings of an existing Teams Translation Rule
Set-CsTeamsTranslationRule 'USTRanslation' -Description 'Add a +1 to a US Phone number.'
Test an existing Teams Translation Rule
Test-CsTeamsTranslationRule -PhoneNumber 2365558989

Location Information Service

List any existing LIS Civic Addresses

Get-CsOnlineLisCivicAddress

Create new LIS Civic Address

New-CsOnlineLisCivicAddress -HouseNumber 123 -StreetName Wacker -StreetSuffix Drive -
PostDirectional W -City Chicago -StateorProvince Illinois -Country US -PostalCode 60606 -Description
"Chicago Headquarters" -CompanyName 'BigBox Corp'

Remove an existing LIS Civic Address

Remove-CsOnlineLisCivicAddress -CivicAddressId aa12bcc9-e681-48f1-b1f7-6688f90c2615

Change settings on an existing LIS Civic Address

Set-CsOnlineLisCivicAddress -CivicAddressId aa12bcc9-e681-48f1-b1f7-6688f90c2615 -Elin 3125551212

Test an existing LIS Civic Address

Test-CsOnlineLisCivicAddress -CivicAddressId aa12bcc9-e681-48f1-b1f7-6688f90c2615

List any existing Lis Location

Get-CsOnlineLisLocation

Create a new Lis Location – add to an existing Civic Address

New-CsOnlineLisLocation -CivicAddressId aa12bcc9-e681-48f1-b1f7-6688f90c2615 -Location "6th Floor"

Remove an existing Lis Location

Remove-CsOnlineLisLocation -Location b80a11dd-022c-47ce-8849-fed8b06215c7

Change settings on an existing Lis Location

Set-CsOnlineLisLocation -LocationId 290acf5b-d275-4c39-8069-fd6c8891371a -Location '7th floor'

List any existing Lis Port

Get-CsOnlineLisPort

Remove an existing Lis Port

Remove-CsOnlineLisPort -PortID 10500

Create an association between Lis Port and Lis Location (Creates new location if none)

Set-CsOnlineLisPort -PortID 10500 -ChassisID 0D-67-CD-16-EE-CC -LocationId 290acf5b-d275-4c39-8069-
fd6c8891371a

List any existing Lis Subnet

Get-CsOnlineLisSubnet

Remove an existing Lis Subnet

Remove-CsOnlineLisSubnet -Subnet 192.168.0.1

Create an association between Lis Subnet and Lis Location (Creates new location if none)

Set-CsOnlineLisSubnet -Subnet 192.168.0.1 -LocationId f037a9ad-4334-455a-a1c5-3838ec0f5d02

List any existing Lis Switch

Get-CsOnlineLisSwitch

Remove an existing Lis Switch

Remove-CsOnlineLisSwitch -ChassisID 0D-67-CD-16-EE-CC

Change settings on an existing Lis Switch

Set-CsOnlineLisSwitch -ChassisID 0D-67-CD-16-EE-CC -LocationId 5e675b9f-c622-4cfb-bd7b-
91d0b1d0929b

List any existing Wireless Access Point

Get-CsOnlineLisWirelessAccessPoint

Remove an existing Lis Wireless Access Point

Remove-CsOnlineLisWirelessAccessPoint -BSSID 0B-57-CD-16-AA-CC

Change settings on an existing Lis Wireless Access Point

Set-CsOnlineLisWirelessAccessPoint -BSSID 0B-57-CD-16-AA-CC -LocationId 5e675b9f-c622-4cfb-bd7b-
91d0b1d0929b

Get-CSTenant* Cmdlets

Get 'information about the tenants that have been configured for use in your organization.'

Get-CsTenant

'Retrieve tenant blocked calling numbers setting'

Get-CsTenantBlockedCallingNumbers

Replaced by Get-CsInboundExemptNumberPattern (see next cmdlet)

Get-CsInboundExemptNumberPattern

'Returns a specific or the full list of all number patterns exempt from call blocking.'

Get-CsInboundExemptNumberPattern

Replaced by Get-CsInboundExemptNumberPattern

Get-CsTenantBlockedNumberExceptionPattern

'Returns a specific or the full list of all number patterns exempt from call blocking.'

Get-CsInboundExemptNumberPattern

'Retrieve a tenant dial plan.'

Get-CsTenantDialPlan

'Returns information about the federation configuration settings.'

Get-CsTenantFederationConfiguration

List 'licensing information for the specified tenant is available in the admin center.'

Get-CsTenantLicensingConfiguration

'Check if Meeting Migration Service (MMS) is enabled in your organization.'

Get-CsTenantMigrationConfiguration

List network configuration settings for a tenant

Get-CsTenantNetworkConfiguration

List ' network region setting in the tenant ... used for Location Based Routing.'

Get-CsTenantNetworkRegion

List ' network site setting in the tenant ... used for Location Based Routing.'

Get-CsTenantNetworkSite

Get ' network subnet setting in the tenant ... used for Location Based Routing.'

Get-CsTenantNetworkSubnet

List external trusted IP addresses a tenant

Get-CsTenantTrustedIPAddress

Mass Teams Creation

Create teams (up to 500) based on CSV file, populate with users (up to 25/team), and notify the ITAdmins status/errors.
New-CsBatchTeamsDeployment -TeamsFilePath "C:\Teams\TeamsGroup01.csv" -UsersFilePath "C:\Teams\UsersGroup01.csv"
-UsersToNotify "ITAdmins@practicalpowershell.com"
Check Status of the Creation of Teams
Get-CsBatchTeamsDeploymentStatus -OrchestrationId

Telephone Number Order

Create a new order report
\$OrderID = New-CsOnlineTelephoneNumberOrder -Name "Chi-312" -Description "Chicago Region" -Country "US" -NumberType "AutoAttendantToll" -Quantity 1 -AreaCode 312 -CivicAddressId 8d3eb3c3-dc64-4e48-b988-2572280bba33
Retrieve an existing order report of a specific telephone number search order:
Get-CsOnlineTelephoneNumberOrder -OrderId \$OrderID
Complete the order for the telephon search order and new phone numbers that come with it
Complete-CsOnlineTelephoneNumberOrder -OrderID \$OrderID
Remove an order:
Clear-CsOnlineTelephoneNumberOrder -OrderID \$OrderID

Teams Shifts

List any available Teams Shifts App Policies:
Get-CsTeamsShiftsAppPolicy
Change settings on existing Teams Shifts App Policy:
Set-CsTeamsShiftsAppPolicy -Identity Global -AllowTimeClockLocationDetection \$False
List available Shifts Connectors
Get-CsTeamsShiftsConnectionConnector
List any Team mapping error reports
Get-CsTeamsShiftsConnectionErrorReport
List existing Shifts Connections:
Get-CsTeamsShiftsConnectionInstance
Create a new Shifts Connection Instance:
\$ShiftConnectionInstance = New-CsTeamsShiftsConnectionInstance -ConnectorId "b735f8d6-f024-4f37-9725-6344a2b311ce" -ConnectorAdminEmail "admin@practicalpowershell.com", "superadmin@practicalpowershell.com" -DesignatedActorId "5d5bb63f7-b554-4b3e-84de-513a48a1dca4" -EnabledConnectorScenario "shift", "swapRequest", "openShift", "openShiftRequest", "timeOff", "timeOffRequest", "timeCard" -EnabledWifiScenario "swapRequest", "openShiftRequest", "timeOffRequest", "UserShiftPreferences" -Name "My Connector Instance" -SyncFrequencyInMin 10 -ConnectorSpecificSettings (New-Object Microsoft.Teams.ConfigAPI.Cmdlets.Generated.Models.ConnectorSpecificBlueYonderSettingsRequest -Property @{ AdminApiUrl = "https://practicalpowershell.com/retail/data/wfmadm/api/v1-beta3"; SiteManagerUrl = "https://practicalpowershell.com/retail/data/wfmsm/api/v1-beta4"; EssApiUrl = "https://practicalpowershell.com/retail/data/wfmess/api/v1-beta2"; RetailWebApiUrl = "https://practicalpowershell.com/retail/data/retailwebapi/api/v1"; CookieAuthUrl = "https://practicalpowershell.com/retail/data/login"; FederatedAuthUrl = "https://practicalpowershell.com/retail/data/login"; LoginUserName = "PlaceholderForUsername"; LoginPwd = "PlaceholderForPassword" })
Remove an existing Shifts Connection Instance:
Remove-CsTeamsShiftsConnectionInstance -ConnectorId "b735f8d6-f024-4f37-9725-6344a2b311ce"
Change settings on a Shifts Connection Instance:
Set-CsTeamsShiftsConnectionInstance -ConnectorId "b735f8d6-f024-4f37-9725-6344a2b311ce" -ConnectorAdminEmail "damian@practicalpowershell.com"
Get list of Team mappings:
Get-CsTeamsShiftsConnectionTeamMap -ConnectorInstanceId WCI-95BF2848-2DDA-4425-B0FE-D62AFED4C0A0

Teams Channel Users

Add a user to a Private Group only:
Add-TeamChannelUser -GroupId 5130aa62-aa13-406f-887c-e40ec8a88d88 -DisplayName Corporate -User damian@practicalpowershell.com
List users in a Teams Channel:
Get-TeamChannelUser -GroupId b0bb1829-77ba-43e6-9a47-95652c0b28f0 -DisplayName General
Remove a user from a Private Teams Channel Only
Remove-TeamChannelUser -GroupId 5130aa62-aa13-406f-887c-e40ec8a88d88 -DisplayName Corporate -User damian@practicalpowershell.com

Teams Voice Apps

List current Voice App Policies:
Get-CsTeamsVoiceApplicationsPolicy
Assign the policies to users:
Grant-CsTeamsVoiceApplicationsPolicy -Identity damian@practicalpowershell.com -PolicyName Global
Create a new Voice App Policy:
New-CsTeamsVoiceApplicationsPolicy -Identity Corp -AllowCallQueueMusicOnHoldChange \$True -AllowCallQueueWelcomeGreetingChange \$True
Remove an existing Voice App Policy:
Remove-CsTeamsVoiceApplicationsPolicy -Identity Corp
Change settings on an existing Voice App Policy
Set-CsTeamsVoiceApplicationsPolicy -Identity Corp -AllowCallQueueWelcomeGreetingChange \$False

Teams Shifts

Retrieve details of a mapped teams last sync:
Get-CsTeamsShiftsConnectionSyncResult -ConnectorInstanceId 6A51B888-FF44-4FEA-82E1-839401E9CD74 -TeamId b0bb1829-77ba-43e6-9a47-95652c0b28f0
Remove an existing Teams Mappings:
Remove-CsTeamsShiftsConnectionTeamMap -ConnectorInstanceId 6A51B888-FF44-4FEA-82E1-839401E9CD74 -TeamId b0bb1829-77ba-43e6-9a47-95652c0b28f0
Lists batch mapping operations created by New-CsTeamsShiftsConnectionBatchTeamMap:
Get-CsTeamsShiftsConnectionOperation -OperationId de69f349-a97b-4e71-91c3-d30b5732c3ad

Teams Shifts

Find teams not mapped to WFM teams:

```
Get-CsTeamsShiftsConnectionWfmTeam -ConnectorInstanceId WCI-6A51B888-FF44-4FEA-82E1-839401E9CD74
```

List Workforce management users in a specific Team:

```
Get-CsTeamsShiftsConnectionWfmUser -ConnectorInstanceId WCI-6A51B888-FF44-4FEA-82E1-839401E9CD74 -WfmTeamId '2000214'
```

List existing Shifts Policies:

```
Get-CsTeamsShiftsPolicy
```

Assign existing Shifts Policy to users

```
Grant-CsTeamsShiftsPolicy -Identity damian@practicalpowershell.com -PolicyName Corp
```

Create new Teams Shifts Policy

```
New-CsTeamsShiftsPolicy -Identity RemoteSite1 -EnableShiftPresence $True -ShiftNoticeMessageType CustomMessage -ShiftNoticeMessageCustom 'Unauthorized access detected. Off hours access to Teams is not allowed.'
```

Remove existing Teams Shifts Policy

```
Remove-CsTeamsShiftsPolicy RemoteSite1
```

Change settings on a Teams Shifts Policy

```
Set-CsTeamsShiftsPolicy RemoteSite1 -AccessGracePeriodMinutes 30
```

Connect Teams and Workforce management Teams:

```
$TeamMap = @{
    TeamId = '00a8358f-aae6-426b-a486-ba8b72f3dd7a'
    WfmTeamId = 1004593
    TimeZone = "America/Chicago"
}
```

```
New-CsTeamsShiftsConnectionBatchTeamMap -ConnectorInstanceId WCI-2afeb8ec-a0f6-4580-8f1e-85fd4a343e01 -TeamMapping @($TeamMap)
```

Removes a Shifts scheule in a specified time range:

```
Remove-CsTeamsShiftsScheduleRecord -TeamId "5c8be873-a38a-4fbd-801a-f4c442244233" -DateRangeStartDate "2022-09-01 00:00:00" -DateRangeEndDate "2022-09-08 00:00:00" -ClearSchedulingGroup:$false -EntityType "swapRequest", "openShiftRequest", "timeOffRequest" -DesignatedActorId "5fc8abf1-2178-4043-b601-4802e0a703cd"
```

Unassigned Number Treatment

List any existing Unassigned Number Treatments:

```
Get-CsTeamsUnassignedNumberTreatment
```

Create a new Unassigned Number Treatment:

```
$UserId = (Get-CsOnlineUser -Identity damian@practicalPowerShell.com).Identity
New-CsTeamsUnassignedNumberTreatment -Identity UNT1 -Pattern "\^+13127774431$" -TargetType User -Target $UserId -TreatmentPriority 2
```

Remove an existing Unassigned Number Treatment:

```
Remove-CsTeamsUnassignedNumberTreatment UNT1
```

Change settings of any existing Unassigned Number Treatments:

```
Set-CsTeamsUnassignedNumberTreatment UNT1 -TreatmentPriority 5
```

Test any existing Unassigned Number Treatments:

```
Test-CsTeamsUnassignedNumberTreatment -PhoneNumber 3127774431
```

Misc New Cmdlets

Test Workforce Management connector:

```
$InstanceName = "Teams Dev Conn Test";$UserName = "Dev_Svc_Test";
$Password = "@#SWDFDT$"
Test-CsTeamsShiftsConnectionValidate -ConnectorId "c374fa9f-62fa-45e2-8600-831097447e2c" -ConnectorSpecificSettingAdminApiUrl "https://dev01.bigboxcorp.com/dev/main/testing/apitest1"
```

Download an existing Audio file, like an AutoAttendant:

```
Export-CsOnlineAudioFile -ApplicationId "Corp-Hunting" -Identity 22fe00dc46ef4f148cb44b9026a91b05 [System.IO.File]::WriteAllBytes('C:\CorpAAMsg.wav', $Content)
```

List existing Teams Files Policies:

```
Get-CsTeamsFilesPolicy
```

List existing Teams Media Logging policies:

```
Get-CsTeamsMediaLoggingPolicy
```

List existing Teams Room Video TeleConferencing Policies:

```
Get-CsTeamsRoomVideoTeleConferencingPolicy
```

All switches not available, but should assign policy to user:

```
Grant-CsTeamsRoomVideoTeleConferencingPolicy
```

Create a new delegate for callig in Teams:

```
New-CsUserCallingDelegate -Identity damian@practicalpowershell.com -Delegate dave@practicalpowershell.com -MakeCalls $True -ReceiveCalls $True -ManageSettings $False
```

Remove calling delegate:

```
Remove-CsUserCallingDelegate -Identity damian@practicalpowershell.com -Delegate dave@practicalpowershell.com
```

Change settings on an existing calling delegate:

```
Set-CsUserCallingDelegate -Identity damian@practicalpowershell.com -Delegate dave@practicalpowershell.com -ManageSettings $True
```

Teams Enhanced Encryption Policy

List any existing Teams Enhanced Encryption Policies:

```
Get-CsTeamsEnhancedEncryptionPolicy
```

Assign any existing Teams Enhanced Encryption Policies:

```
Grant-CsTeamsEnhancedEncryptionPolicy -Identity damian@bigboxcorp.net -PolicyName Global
```

Create a new Teams Enhanced Encryption Policy:

```
New-CsTeamsEnhancedEncryptionPolicy -Identity Corp-EEP -MeetingEndToEndEncryption DisabledUserOverride
```

Remove any existing Teams Enhanced Encryption Policy:

```
Remove-CsTeamsEnhancedEncryptionPolicy Corp-EEP
```

Change settings of any existing Teams Enhanced Encryption Policies:

```
Set-CsTeamsEnhancedEncryptionPolicy Corp-EEP -MeetingEndToEndEncryption Disabled
```

Teams and Graph PowerShell

```
$Perms = 'TeamSettings.ReadWrite.All','User.Read.All','Group.ReadWrite.All'
Connect-MgGraph -Scopes $Perm -TenantId 33ff8b74-9880-455b-b26f-f3285fd21099
Get-Command | Where Source -eq Microsoft.Graph.Teams
Get-MgTeam
Get-MgTeamChannel
```

New Cmdlets

Clear-TeamsEnvironmentConfig
 Export-CsAcquiredPhoneNumber
 Get-AllM365TeamsApps
 Get-CsExportAcquiredPhoneNumberStatus
 Get-CsSdgBulkSignInRequestsSummary
 Get-CsSdgBulkSignInRequestStatus
 Get-CsTeamsAIPolicy
 Get-CsTeamsAudioConferencingCustomPromptsConfiguration
 Get-CsTeamsBYODAndDesksPolicy
 Get-CsTeamsCustomBannerText
 Get-CsTeamsEducationConfiguration
 Get-CsTeamsExternalAccessConfiguration
 Get-CsTeamsFirstPartyMeetingTemplateConfiguration
 Get-CsTeamsMediaConnectivityPolicy
 Get-CsTeamsMeetingBrandingPolicy
 Get-CsTeamsMeetingTemplateConfiguration
 Get-CsTeamsMeetingTemplatePermissionPolicy
 Get-CsTeamsMessagingConfiguration
 Get-CsTeamsMultiTenantOrganizationConfiguration
 Get-CsTeamsRecordingRollOutPolicy
 Get-CsTeamsSettingsCustomApp
 Get-CsTeamsSharedCallingRoutingPolicy
 Get-CsTeamsShiftsConnection
 Get-CsTeamsSipDevicesConfiguration
 Get-CsTeamsTemplatePermissionPolicy
 Get-CsTeamsVirtualAppointmentsPolicy
 Get-CsTeamsWorkLocationDetectionPolicy
 Get-M365TeamsApp
 Get-M365UnifiedCustomPendingApps
 Get-M365UnifiedTenantSettings
 Grant-CsTeamsAIPolicy
 Grant-CsTeamsBYODAndDesksPolicy
 Grant-CsTeamsMediaConnectivityPolicy
 Grant-CsTeamsMeetingBrandingPolicy
 Grant-CsTeamsMeetingTemplatePermissionPolicy
 Grant-CsTeamsRecordingRollOutPolicy
 Grant-CsTeamsSharedCallingRoutingPolicy
 Grant-CsTeamsVirtualAppointmentsPolicy
 Grant-CsTeamsWorkLocationDetectionPolicy
 New-CsCustomPrompt
 New-CsCustomPromptPackage
 New-CsOnlineDirectRoutingTelephoneNumberUploadOrder
 New-CsOnlineTelephoneNumberReleaseOrder
 New-CsSdgBulkSignInRequest
 New-CsTeamsAIPolicy
 New-CsTeamsAppPermissionPolicy
 New-CsTeamsAppSetupPolicy

New-CsTeamsBYODAndDesksPolicy
 New-CsTeamsCustomBannerText
 New-CsTeamsEmergencyCallingExtendedNotification
 New-CsTeamsHiddenMeetingTemplate
 New-CsTeamsHiddenTemplate
 New-CsTeamsMediaConnectivityPolicy
 New-CsTeamsMeetingBackgroundImage
 New-CsTeamsMeetingBrandingPolicy
 New-CsTeamsMeetingBrandingTheme
 New-CsTeamsMeetingTemplatePermissionPolicy
 New-CsTeamsNdiAssuranceSlate
 New-CsTeamsRecordingRollOutPolicy
 New-CsTeamsSharedCallingRoutingPolicy
 New-CsTeamsShiftsConnection
 New-CsTeamsTemplatePermissionPolicy
 New-CsTeamsVirtualAppointmentsPolicy
 New-CsTeamsWorkLocationDetectionPolicy
 Remove-CsTeamsAIPolicy
 Remove-CsTeamsBYODAndDesksPolicy
 Remove-CsTeamsCustomBannerText
 Remove-CsTeamsMediaConnectivityPolicy
 Remove-CsTeamsMeetingBrandingPolicy
 Remove-CsTeamsMeetingTemplatePermissionPolicy
 Remove-CsTeamsRecordingRollOutPolicy
 Remove-CsTeamsSharedCallingRoutingPolicy
 Remove-CsTeamsShiftsConnection
 Remove-CsTeamsTemplatePermissionPolicy
 Remove-CsTeamsVirtualAppointmentsPolicy
 Remove-CsTeamsWorkLocationDetectionPolicy
 Remove-CsUserLicenseGracePeriod
 Set-CsTeamsAIPolicy
 Set-CsTeamsAppPermissionPolicy
 Set-CsTeamsAppSetupPolicy
 Set-CsTeamsAudioConferencingCustomPromptsConfiguration
 Set-CsTeamsBYODAndDesksPolicy
 Set-CsTeamsCustomBannerText
 Set-CsTeamsEducationConfiguration
 Set-CsTeamsExternalAccessConfiguration
 Set-CsTeamsMediaConnectivityPolicy
 Set-CsTeamsMeetingBrandingPolicy
 Set-CsTeamsMeetingTemplatePermissionPolicy
 Set-CsTeamsMessagingConfiguration
 Set-CsTeamsMultiTenantOrganizationConfiguration
 Set-CsTeamsRecordingRollOutPolicy
 Set-CsTeamsSettingsCustomApp
 Set-CsTeamsSharedCallingRoutingPolicy
 Set-CsTeamsShiftsConnection

Set-CsTeamsSipDevicesConfiguration
 Set-CsTeamsTemplatePermissionPolicy
 Set-CsTeamsVirtualAppointmentsPolicy
 Set-CsTeamsWorkLocationDetectionPolicy
 Set-TeamsEnvironmentConfig
 Set-TeamTargetingHierarchy
 Update-CsTeamsShiftsConnection
 Update-CsTeamsShiftsConnectionInstance
 Update-M365TeamsApp
 Update-M365UnifiedCustomPendingApp
 Update-M365UnifiedTenantSettings