

Teams Direct Routing - 04 - User Provisioning

In This Article

- Overview
- Provision a User with the Teams Admin Center
 - Assign a Direct Routing Phone Number
 - Assign the Voice Routing & Tenant Dial Plan Policies
- Provision Using PowerShell
 - Check a User's Status
 - Provision a Single Teams User
 - Provision from a CSV File

Overview

This article covers provisioning users for Teams direct routing in the Teams Admin Center (TAC) and PowerShell.



IMPORTANT

Make sure your user accounts have been licensed before configuring them for direct routing. They should have a license that includes MS Teams, and the Teams Phone add-on license. Details on licensing requirements can be found in the [Teams Direct Routing - 01 - Planning and Prerequisites](#) article.

Additionally, you should **never remove a Teams Phone add-on license** from a Teams user configured for direct routing until you have removed the direct routing configuration.

Removing the Teams Phone add-on license will not remove the user's direct routing configuration. The phone number assigned to the user will remain associated with the user account, and the phone number cannot be assigned to another user until it is cleared, or you delete the user account from your environment.

If you need to clear the phone number from an unlicensed user account, you will need to re-assign a Teams Phone add-on license, and properly remove (deprovision) the user's direct routing configuration.



TIP

If you are switching a user from a Microsoft calling plan to Teams direct routing, remove the phone number from the user, and then remove the calling plan license. Make sure you keep the Teams Phone add-on license assigned.

Provision a User with the Teams Admin Center

These steps show you how to configure a single user account for Teams direct routing in the Teams Admin Center (TAC). The TAC does not provide any feedback on a user's current direct routing status. If you need to check the status of a user's configuration, or do any troubleshooting, you should use PowerShell.



NOTE

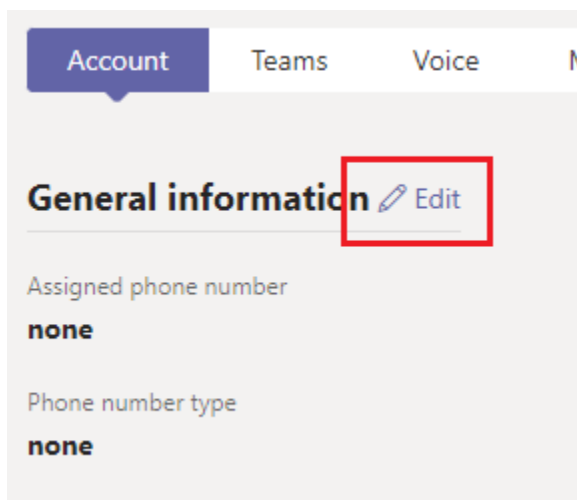
Make sure the user account has been properly licensed before configuring them for direct routing. The user account should have a license that includes MS Teams, and the Teams Phone add-on license. Details on licensing requirements can be found in the [Teams Direct Routing - 01 - Planning and Prerequisites](#) article.

Sign into the Teams Admin Center (TAC): <https://admin.teams.microsoft.com>

Assign a Direct Routing Phone Number

Users > Manage Users > Click on a user's Display Name to view their settings

Next to **General Information**, click **Edit**.



Account Teams Voice

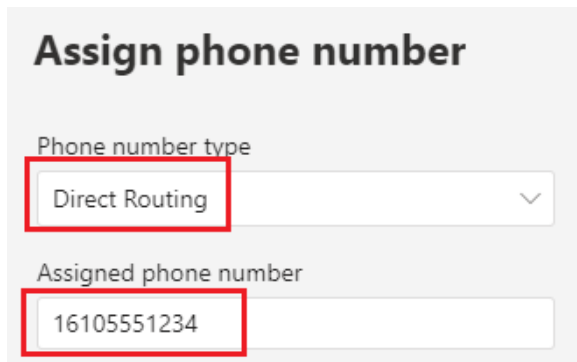
General information [Edit](#)

Assigned phone number
none

Phone number type
none

In the right-hand flyout pane:

- **Phone Number Type:** Direct Routing
- **Assigned Phone Number:** Enter a phone number including the country code



Assign phone number

Phone number type
Direct Routing

Assigned phone number
16105551234

Click the **Apply** button at the bottom of the flyout pane.



Apply Cancel

The user account's properties will update to show the applied changes.

Account Teams Voice

General information [Edit](#)

Assigned phone number
+1 610555123

Phone number type
Direct Routing

Assign the Voice Routing & Tenant Dial Plan Policies

Select the user account's **Policies** tab

Account Teams Voice Meetings & calls **Policies** Usage

Edit the Assigned Policies

Assigned policies [Edit](#)

In the right-hand flyout pane, towards the bottom of the policy list, select one of the Evolve IP policy configurations for the following:

- **Dial Plan:** EvolveIP-TenantDialPlan
- **Voice Routing Policy:** EvolveIP-<PolicyName>

Dial plan
EvolveIP-TenantDialPlan

Voice routing policy
EvolveIP-East

Click the **Apply** button at the bottom of the flyout pane.

Apply Cancel

The user's policy assignments will update to show the applied changes.

Dial plan

EvolveIP-TenantDialPlan (Direct)

Details

Voice routing policy

EvolveIP-East (Direct)

Details



NOTE

It can take up to 24 hours for Microsoft's services to apply the changes. Normally it takes 10-30 minutes.

Provision Using PowerShell

These steps show you how to configure one or more user accounts for Teams direct routing using PowerShell. Before performing these steps make sure you are connected to your Microsoft Teams services using PowerShell. If needed, review the [Teams Direct Routing - 01 - Planning and Prerequisites](#) article for details.



IMPORTANT

The commands in this article require the latest **Microsoft Teams PowerShell module**. If needed, use the `Update-Module -Name MicrosoftTeams` command to make sure you're on the latest version.



NOTE

Make sure all user accounts have been properly licensed before configuring them for direct routing. The user accounts should have a license that includes MS Teams, and the Teams Phone add-on license. Details on licensing requirements can be found in the [Teams Direct Routing - 01 - Planning and Prerequisites](#) article.

Check a User's Status

The below commands can be used to view the voice properties of one or more users. This can be useful for checking the status of a user before and after configuring them for direct routing.

- [Get-CsOnlineUser](#)

```
# View a list of the user's properties related to Direct Routing
Get-CsOnlineUser user@domain.com | FL Displ*, UserPri*, SipA*,
IsSipEnabled, Inter*, TeamsUpgradeE*, Enterprise*, OnlineVoiceR*,
TeamsCall*, TenantD*, OnlineDial*, LineURI

# View all of the user's properties
Get-CsOnlineUser user@domain.com | FL *

# Export a list of all voice users and their properties to a CSV file
$CsvFilePath = "C:\Path\to\file.csv"

Get-CsOnlineUser -ResultSize 2147483647 | Select Displ*, UserPri*, SipA*,
IsSipEnabled, Inter*, TeamsUpgradeE*, Enterprise*, OnlineVoiceR*,
TeamsCall*, TenantD*, OnlineDial*, LineURI | Export-Csv $CsvFilePath -nti
```

Here's a description of the more important properties outputted with the `Get-CsOnlineUser` command:

- **IsSipEnabled** - Indicates whether the user is enabled for MS Teams. If this is showing False, then the user account cannot be configured for direct routing.
- **EnterpriseVoiceEnabled** - Indicates whether the user has been enabled for Enterprise Voice features (aka. Teams Phone). Enterprise voice has to be enabled for Teams Direct Routing to work.
- **LineURI** - Shows the provisioned phone number assigned to the user.
- **InterpretedUserType** - This can help with troubleshooting users that are not working as expected. Unfortunately, there's a lot of "UserTypes", and Microsoft does not have them documented. People in the Teams community have tried creating some documentation of their own, and doing a Google search for InterpretedUserType will turn up some results, which may help. If not, you'll need to open a support case with Microsoft.
- **TeamsUpgradeEffectiveMode** - For Teams direct routing to work, this should show **TeamsOnly**.
- **OnlineVoiceRoutingPolicy** - If this is blank, the user is not assigned a voice routing policy. When a user is configured for Teams direct routing, this should be set to an **Evolve IP** voice routing policy. To view a list of voice routing policies in your tenant use the Get-CsOnlineVoiceRoutingPolicy command.
- **TenantDialPlan** - If this is blank, the user is assigned to the **Global** tenant dial plan. To view a list of dial plans in your tenant use the Get-CsTenantDialPlan command.
- **TeamsCallingPolicy** - If this is blank, the user is assigned to the **Global** calling policy. To view a list of calling policies use the Get-CsTeamsCallingPolicy command.
- **TeamsCallParkPolicy** - If this is blank, the user is assigned to the **Global** call park policy. To view a list of call park policies use the Get-CsTeamsCallParkPolicy command.
- **OnlineDialinConferencingPolicy** - If this shows **ServiceAllowed**, the user is enabled for Microsoft's Audio Conferencing. If this is blank, the user is not enabled for Audio Conferencing.
- **OnlineDialOutPolicy** - This policy defines the dial out restrictions from an audio conference. A table of the different policies and their restrictions can be found in this [Microsoft article](#).

Provision a Single Teams User

- [Get-CsOnlineUser](#)
- [Get-CsOnlineVoiceRoutingPolicy](#)
- [Get-CsTenantDialPlan](#)
- [Set-CsPhoneNumberAssignment](#)
- [Grant-CsTeamsUpgradePolicy](#)
- [Grant-CsOnlineVoiceRoutingPolicy](#)
- [Grant-CsTenantDialPlan](#)

Get the name of the EIP Voice Routing Policy, which has to be entered into the below block of PowerShell code.

```
Get-CsOnlineVoiceRoutingPolicy | Where { $_.Identity -like "**Evolve*" }
```

Get the name of the EIP Tenant Dial Plan, which has to be entered in the below block of PowerShell code.

```
Get-CsTenantDialPlan | Where { $_.Identity -like "**Evolve*" }
```

Provision a User for Direct Routing

NOTE

Microsoft requires that all phone numbers be in the E.164 format. This means the following:

- Must include the country code
- Include a "+" before the country code
- No special characters or spaces
- Example: +12154567890



IMPORTANT

In the below PowerShell code, be sure to change the relevant data on **lines 2, 5, 8, & 11** to match your configuration and specific user.

```
# Enter the name of the Evolve IP voice routing policy
$VrPolicyName = "EvolveIP-USEast"

# Enter the name of the Evolve IP tenant dial plan
$TdpPolicyName = "EvolveIP-TenantDialPlan"

# Enter the user's user principal name
$UPN = "user@domain.com"

# Enter the user's telephone number (no spaces, dashes, or parentheses)
$Phone = "+16105551234"

### Begin User Provisioning ###

# This will check if the user's coexistence mode is set to TeamsOnly for
Direct
# Routing. If it is not set to TeamsOnly, it will be set.

If ((Get-CsOnlineUser $UPN).TeamsUpgradeEffectiveMode -ne "TeamsOnly") {
    Grant-CsTeamsUpgradePolicy -Identity $UPN -PolicyName UpgradeToTeams
}

# This will configure the user's account for direct routing and add the
account
# to the voice routing policy & tenant dial plan

Set-CsPhoneNumberAssignment -Identity $UPN -PhoneNumber $Phone -
PhoneNumberType DirectRouting
Grant-CsOnlineVoiceRoutingPolicy -Identity $UPN -PolicyName $VrPolicyName
Grant-CsTenantDialPlan -Identity $UPN -PolicyName $TdpPolicyName

### End User Provisioning ###
```

Provision from a CSV File

- [Get-CsOnlineVoiceRoutingPolicy](#)
- [Get-CsTenantDialPlan](#)
- [Grant-CsTeamsUpgradePolicy](#)
- [Grant-CsOnlineVoiceRoutingPolicy](#)
- [Grant-CsTenantDialPlan](#)
- [Set-CsPhoneNumberAssignment](#)
- [Import-Csv](#)

CSV File Requirements

Create or use an existing CSV file that contains the following required fields:

- **DisplayName** - User account display Name
- **UserUPN** - User account's User Principal Name (Office 365 sign in address)
- **PhoneNumber** - Assigned phone number including the country code (no special characters or spaces)
- **Location** - Determines which location a user will be assigned

Here's an example for the US region:

DisplayName	UserUPN	PhoneNumber	Location
George Kastanza	jkastanza@seineild.com	16105551234	east

Elaine Benes	ebenes@seinfeld.com	16105551235	west
--------------	---------------------	-------------	------

Here's an example for the EU region:

DisplayName	UserUPN	PhoneNumber	Location
Cosmo Kramer	ckramer@seinfeld.com	31884281234	NNpeer01
Newman	newman@seinfeld.com	31884221234	NNpeer02

Provision Users from a CSV File



TIP

Paste the below PowerShell code into an editor like the PowerShell ISE or VSCode, and run it from there.

First, **get the name of your Voice Routing Policies** and **Tenant Dial Plan**, which has to be entered into the below block of US or EU PowerShell code on **lines 8, 9, & 12**.

```
Get-CsOnlineVoiceRoutingPolicy | Where { $_.Identity -like "*Evolve*" }
Get-CsTenantDialPlan | Where { $_.Identity -like "*Evolve*" }
```

Next, use these blocks of code, depending on your region, with the output from the previous query on **lines 8, 9, & 12**. Also define the path to your CSV file on **line 5**.

```
[CmdletBinding()]
Param ()

# Path to the CSV file
$UserData = Import-Csv "C:\Path\to\file.csv"

# Enter the name of your voice routing policies here
$VrPolicyNameEast = "EvolveIP-USEast"
$VrPolicyNameWest = "EvolveIP-USWest"

# Enter the name of your tenant dial plan here
$TdpPolicyName = "EvolveIP-TenantDialPlan"

# The following will loop through each entry in the CSV file and output
the provisioning status to the console

ForEach ($User in $UserData) {

    Write-Host "`nPROVISIONING: $($User.DisplayName) ($($User.UserUPN))"

    # This will check the user's location and associate it with the
    correct routing policy
    If ($User.Location.trim() -eq "east") { $Vrp = $VrPolicyNameEast }
    ElseIf ($User.Location.trim() -eq "west") { $Vrp = $VrPolicyNameWest }
    Else { Write-Host "No location defined. Continuing to the next user." -
    ForegroundColor Yellow; Continue }

    # This will check the user's phone number, and set it to the E.164
    format
    $PhoneNumber = ""
    $PhoneNumber = $User.PhoneNumber.trim()

    # Check the length of the number, and assume "+" is not included in
    the number
    If ($PhoneNumber -notmatch "^\d+$" -or $PhoneNumber.Length -lt 10 -or
    $PhoneNumber.Length -gt 11) {
        Write-Host "User phone number is not valid: $($PhoneNumber)" -
        ForegroundColor Yellow
    }
}
```

```

        Write-Host "Continuing to the next user." -ForegroundColor Yellow
        Continue
    }

    # Check if the phone number starts with the US country code, and add
    the "+" to the front of the number
    If (-not($PhoneNumber.StartsWith('1')) -and $PhoneNumber.Length -eq
    10) { $PhoneNumber = "+1$($PhoneNumber)" }
    ElseIf ($PhoneNumber.StartsWith('1') -and $PhoneNumber.Length -eq 11)
    { $PhoneNumber = "+$($PhoneNumber)" }
    Else {
        Write-Host "User phone number is not valid: $($PhoneNumber)" -
        ForegroundColor Yellow
        Write-Host "Continuing to the next user." -ForegroundColor Yellow
        Continue
    }

    # This will define the user's direct routing properties
    $CsUserProp = @{
        Identity      = $User.UserUpn
        PhoneNumber   = $PhoneNumber
        PhoneNumberType = "DirectRouting"
    }

    # This will check if the user's coexistence mode is set to TeamsOnly.
    If not, it will be set.
    If ((Get-CsOnlineUser $CsUserProp.Identity).TeamsUpgradeEffectiveMode -
    ne "TeamsOnly") {
        Try {
            Grant-CsTeamsUpgradePolicy -Identity $CsUserProp.Identity -
            PolicyName UpgradeToTeams -ErrorAction Stop
            Write-Host "User set to TeamsOnly coexistence mode."
        } Catch { Write-Host "Failed to set user to TeamsOnly coexistence
        mode. Continuing to the next user." -ForegroundColor Yellow; Continue }
    }

    # This will configure the user's account for direct routing & add the
    account to your voice routing policy
    Try {
        Set-CsPhoneNumberAssignment @CsUserProp -ErrorAction Stop
        Write-Host "User enabled for direct routing."
    } Catch { Write-Host "Failed to enable user for direct routing.
    Continuing to the next user." -ForegroundColor Yellow; Continue }

    Try {
        Grant-CsOnlineVoiceRoutingPolicy -Identity $User.UserUPN -
        PolicyName $Vrp -ErrorAction Stop
        Write-Host "User assigned a VRP: $($Vrp)"
    } Catch { Write-Host "Failed to assign the VRP: $($Vrp)" -
    ForegroundColor Yellow }

    Try {
        Grant-CsTenantDialPlan -Identity $User.UserUPN -PolicyName
        $TdpPolicyName -ErrorAction Stop
        Write-Host "User assigned a Tenant Dial Plan: $($TdpPolicyName)"
    } Catch { Write-Host "Failed to assign a Tenant Dial Plan:
    $($TdpPolicyName)" -ForegroundColor Yellow }
} # End ForEach

```

```

[CmdletBinding()]
Param ()

# Path to the CSV file
$UserData = Import-Csv "C:\Path\to\file.csv"

```

```

# Enter the name of your voice routing policies here
$VrPolicyNameNNpeer01 = "EvolveIP-NLNNpeer01"
$VrPolicyNameNNpeer02 = "EvolveIP-NLNNpeer02"

# Enter the name of your tenant dial plan here
$TdpPolicyName = "EvolveIP-TenantDialPlan"

# The following will loop through each entry in the CSV file and output
the provisioning status to the console

ForEach ($User in $UserData) {

    Write-Host "`nPROVISIONING: $($User.DisplayName) ($($User.UserUPN))"

    # This will check the user's location and associate it with the
    correct routing policy
    If ($User.Location.trim() -eq "NNpeer01") { $Vrp =
$VrPolicyNameNNpeer01 }
    ElseIf ($User.Location.trim() -eq "NNpeer02") { $Vrp =
$VrPolicyNameNNpeer02 }
    Else { Write-Host "No location defined. Continuing to next user." -
ForegroundColor Yellow; Continue }

    # This will define the user's direct routing properties
    $CsUserProp = @{
        Identity          = $User.UserUpn
        PhoneNumber       = "+$($User.PhoneNumber)"
        PhoneNumberType = "DirectRouting"
    }

    # This will check if the user's coexistence mode is set to TeamsOnly.
    If not, it will be set.
    If ((Get-CsOnlineUser $CsUserProp.Identity).TeamsUpgradeEffectiveMode -
ne "TeamsOnly") {
        Try {
            Grant-CsTeamsUpgradePolicy -Identity $CsUserProp.Identity -
PolicyName UpgradeToTeams -ErrorAction Stop
            Write-Host "User set to TeamsOnly coexistence mode."
        } Catch { Write-Host "Failed to set user to TeamsOnly coexistence
mode. Continuing to next user." -ForegroundColor Yellow; Continue }
    }

    # This will configure the user's account for direct routing & add the
    account to your voice routing policy
    Try {
        Set-CsPhoneNumberAssignment @CsUserProp -ErrorAction Stop
        Write-Host "User enabled for direct routing."
    } Catch { Write-Host "Failed to enable user for direct routing.
Continuing to next user." -ForegroundColor Yellow; Continue }

    Try {
        Grant-CsOnlineVoiceRoutingPolicy -Identity $User.UserUPN -
PolicyName $Vrp -ErrorAction Stop
        Write-Host "User assigned a VRP: $($Vrp)"
        Write-Host "User successfully provisioned for direct routing" -
ForegroundColor Green
    } Catch { Write-Host "Failed to assign the VRP: $($Vrp)" -
ForegroundColor Yellow }

    Try {
        Grant-CsTenantDialPlan -Identity $User.UserUPN -PolicyName
$TdpPolicyName -ErrorAction Stop
        Write-Host "User assigned a Tenant Dial Plan: $($TdpPolicyName)"
    } Catch { Write-Host "Failed to assign a Tenant Dial Plan:
$($TdpPolicyName)" -ForegroundColor Yellow }

} # End ForEach

```

