

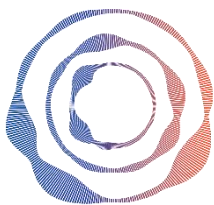
PeterConnects

Receptionist for Teams

Preview version

Deployment Guide

V2021.03.23



PeterConnects
• enabling • meaningful • connections

PeterConnects Receptionist Preview version for Teams Deployment Guide

Welcome to the PeterConnects Receptionist **Preview version** Deployment Guide!

Peterconnects reserves the right to make changes to the GA realease as this preview version does not reflect the latest improvements. Another deployment guide will be available with GA.

This documentation explains how to configure a Peterconnects receptionist preview in your Teams tenant:

- Configuration of our calling bot using the script provided
- Distribute the application to your Teams clients

Contents

Important note on using the deployment guide	3
Configuring PeterConnects Receptionist Callingbot	4
Architecture.....	4
How does this work?	4
Teams with Microsoft Calling Plan	5
Prerequisites.....	5
Step 1. Consent for the Applications.....	7
Step 2.Launch the Script.....	8
Step 3. Function: Prepare-PcBot	9
Step 4. Function: Update-PcBotOnline.....	10
Last Step : Dial Outside and checks.....	11
Teams with Direct Routing	13
Prerequisites.....	13
Step 1. Consent for the Applications.....	14
Step 2.Launch the Script.....	15
Step 3. Function: Prepare-PcBot	16
Step 4. Function: Update-PcBotDirect	17
Last Step : Dial Outside and Checks.....	18
Teams Admin Center 3 rd party apps.....	20
Permission Policy	20
Manage the application.....	21
Setup Policies	21
Peterconnects Receptionist in the Teams client	22
Access the Peterconnects Receptionist via a webbrowser outside the Teams Client.....	23
Microsoft App Store	24
PeterConnects Receptionist from the Microsoft App Store	24
References.....	25
Q&A	26

Important note on using the deployment guide

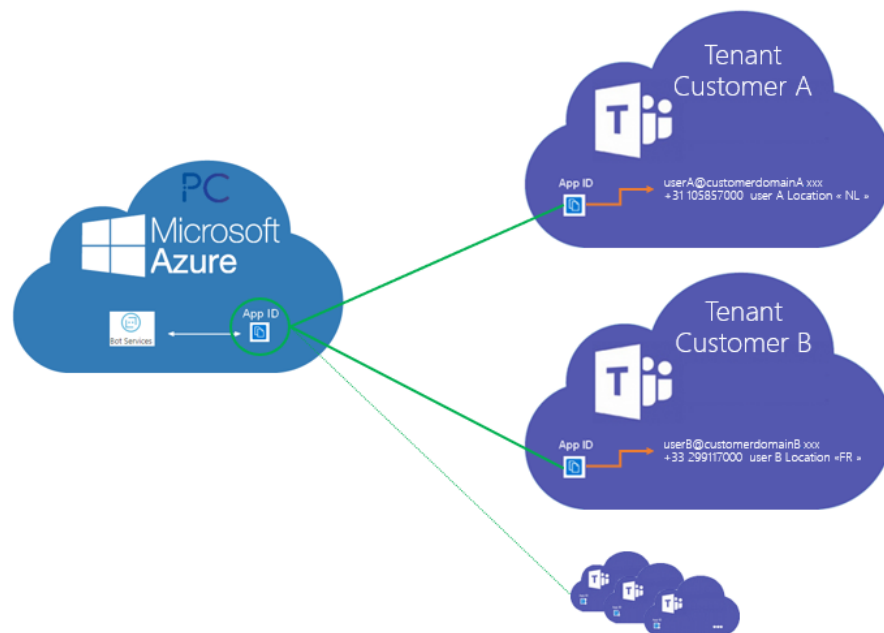
3 components are used to run Peterconnects Receptionist for Teams:

Peterconnects Preview Receptionist Teams App	msteams.peterconnects.com.zip
Deployment Script	Add-PcBotAppInstance.ps1
Preview CallingBot ID (used when creating endpoint)	227ce992-e7a1-4695-a857-54b23670af35

Configuring PeterConnects Receptionist Callingbot

Architecture

This following diagram explains how customers tenant connect Peterconnects Calling Bot service. After consent, a virtual user is defined and associated to a telephone number (from the Direct Routing plan or Microsoft Service Number)



How does this work?

Tenant A wants to call to our PeterConnects Calling bot and must then know our Application IDs. An administrator of Tenant A needs to **consent** so that Peterconnects applications ID can call in tenant A: this is done via our web site (cf. page 4.).

On tenant A, a subset is created that is related to our Calling Bot: Tenant A can use our Calling Bot

To be able to call, we need an endpoint. This uses a virtual end-user in the customer directory of Tenant A, associated with a phone number, and a free Virtual User add-on License (page 6.)

This User is a "1- to-1" relation with our Application ID and is assigned to a service phone number to be dialed, either from your Direct Routing plan OR Microsoft Calling Pplans.

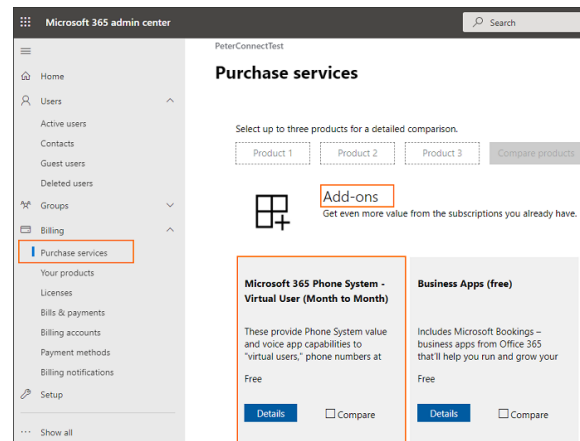
To create this virtual user, we simplified Teams Powershell commands in our [Add-PcBotAppInstance.ps1](#) script located in [PeterConnects-Teams-Receptionist-CallingBot-Script.zip](#).

Teams with Microsoft Calling Plan

Prerequisites

1. Microsoft Calling Plan ordered for in/out calls
2. Microsoft 365 phone system – Virtual User add-on License (free)

👉 One virtual license is required per Calling Queue number ([more info](#))



3. Microsoft Service (tool) free numbers

You can order Service (toll) **free** numbers either from the *Microsoft Teams Admin Center – Voice – Phone numbers menu* or with using the Microsoft [PDF LOA forms](#) :

New telephone number request



Thank you for choosing Microsoft as your service provider. This form must be completed and supplied to Microsoft PSTN Service Desk to get new telephone numbers. We will keep you posted on the progress.

IMPORTANT

1. Please submit requests for different type of numbers in different form submissions. You can only request one number type per request/form. [More info](#) (<https://go.microsoft.com/fwlink/?linkid=851394>)
2. A full valid office address is required to order the numbers. You can list only one "emergency calling address" per request/form.

Number type requested (1 above)

☐ User number

☒ Service (toll) number

Office address for emergency calling (2 above)

VAT ID

Amount of numbers required

NOTE: Numbers are ordered in block size (1/10/100)

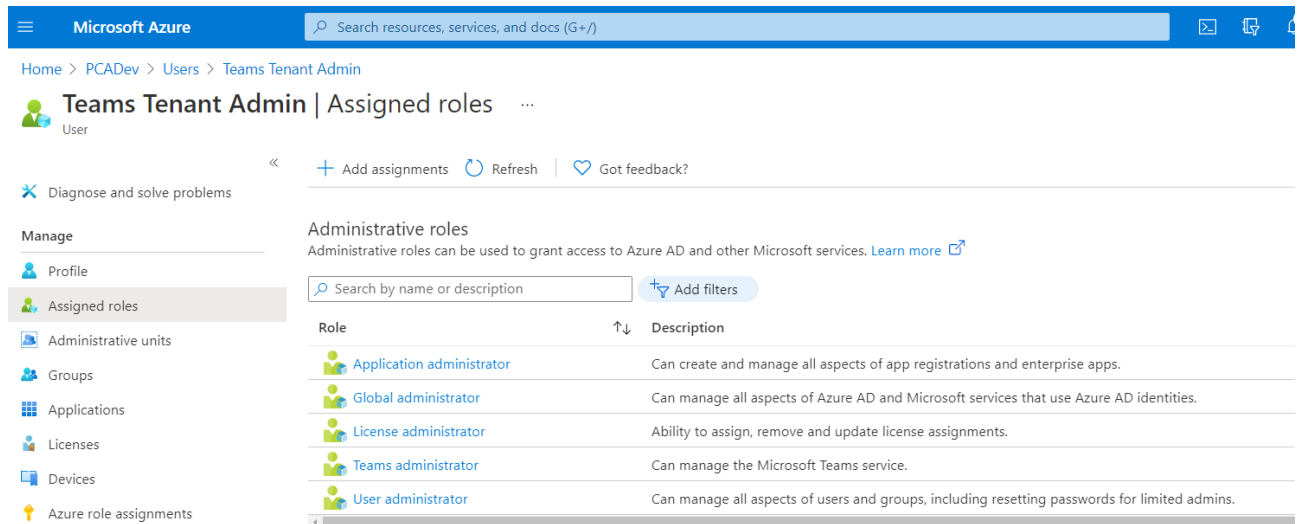
Directory Listing (optional)

☐ No, do not publish my company information.

☐ Yes, I would like to publish my company information with the numbers. When you choose this option,

👉 The Microsoft 365 Phone Service number must belong to the **same Location** as the user created and associated to the queue Service number. This is related to your Emergency Code Area.

1. An Azure AD user with the following roles on your tenant (@yourteamtenant.com):
 - [Global Administrator](#) (need for the consent steps, can be removed after)
 - [Application Administrator](#)
 - [Teams Administrator](#)
 - [License Administrator](#)
 - [User Administrator](#)



The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure' and a search bar. Below it, the breadcrumb trail reads 'Home > PCADev > Users > Teams Tenant Admin'. The main heading is 'Teams Tenant Admin | Assigned roles'. On the left, there's a 'Manage' sidebar with options like Profile, Assigned roles (selected), Administrative units, Groups, Applications, Licenses, Devices, and Azure role assignments. The main content area is titled 'Administrative roles' and includes a search bar and a table of roles.

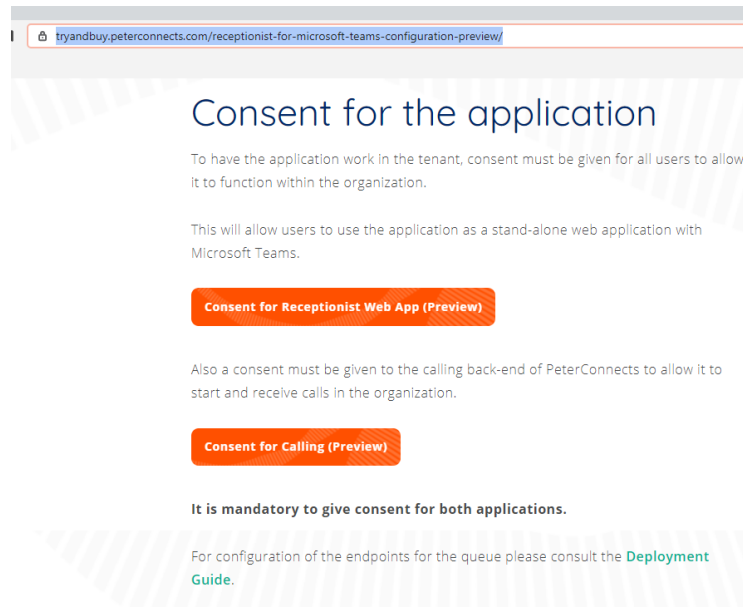
Role	Description
Application administrator	Can create and manage all aspects of app registrations and enterprise apps.
Global administrator	Can manage all aspects of Azure AD and Microsoft services that use Azure AD identities.
License administrator	Ability to assign, remove and update license assignments.
Teams administrator	Can manage the Microsoft Teams service.
User administrator	Can manage all aspects of users and groups, including resetting passwords for limited admins.

Those [rights](#) are subject to change for the GA Release.

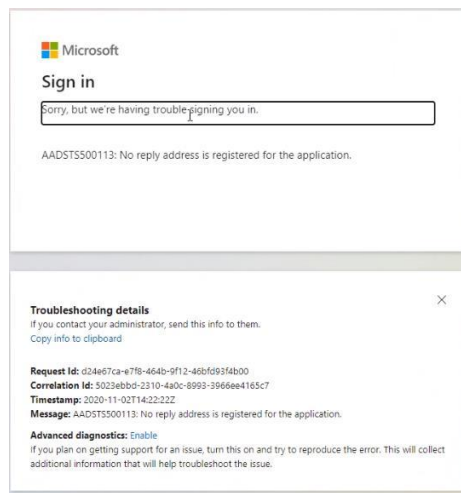
Step 1. Consent for the Applications

Consent for **both Apps** must have been done via the following webpage **before** configuring the endpoint:

<https://tryandbuy.peterconnects.com/receptionist-for-microsoft-teams-configuration-preview/>



After consenting (Agree) you can get the following page. **This is intended** as we do not have supplied a reply URL for the preview App.



Once this is done, you can continue.

Step 2. Launch the Script

1. Install Teams Powershell and AzureAD module

You must be able to call the powershell commands [Connect-MicrosoftTeams](#) and [Connect-AzureAD](#) <https://docs.microsoft.com/en-us/microsoftteams/teams-powershell-overview>

PS C:\ > Install-Module MicrosoftTeams

```
Administrator: Windows PowerShell

PS C:\Users\salesdemoadmin> Install-Module MicrosoftTeams

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): a
```

PS C:\ > Install-Module AzureAD

```
Administrator: Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Installing package 'AzureAD'
Unzipping
[oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo]

PS C:\Users\admin> Install-Module AzureAD

NuGet provider is required to continue
PowerShellGet requires NuGet provider version '2.8.5.201' or newer to interact with NuGet-based repositories. The NuGet
provider must be available in 'C:\Program Files\PackageManagement\ProviderAssemblies' or
'C:\Users\salesdemoadmin\AppData\Local\PackageManagement\ProviderAssemblies'. You can also install the NuGet provider
by running 'Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force'. Do you want PowerShellGet to install
and import the NuGet provider now?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): a

PS C:\Users\admin>
```

2. Load the functions from the script

Enter the **exact** following syntax (**point space point antislash**) to load the script

. .\Add-PcBotAppInstance.ps1

```
Select Administrator: Windows PowerShell

PS C:\Scripts> . .\Add-PcBotAppInstance.ps1
PS C:\> Prepare-PcBot
WARNING: You will be asked to login twice. One time for Teams and One time for AzureAD

Account          Environment Tenant          TenantId
-----
sebastien@pcadev.net AzureCloud bf0dfb98-fb04-42b8-984f-34283f7089a6 bf0dfb98-fb04-42b8-984f-34283f7089a6
sebastien@pcadev.net AzureCloud bf0dfb98-fb04-42b8-984f-34283f7089a6 bf0dfb98-fb04-42b8-984f-34283f7089a6
```

👉 The loading will not give any feedback on the console.

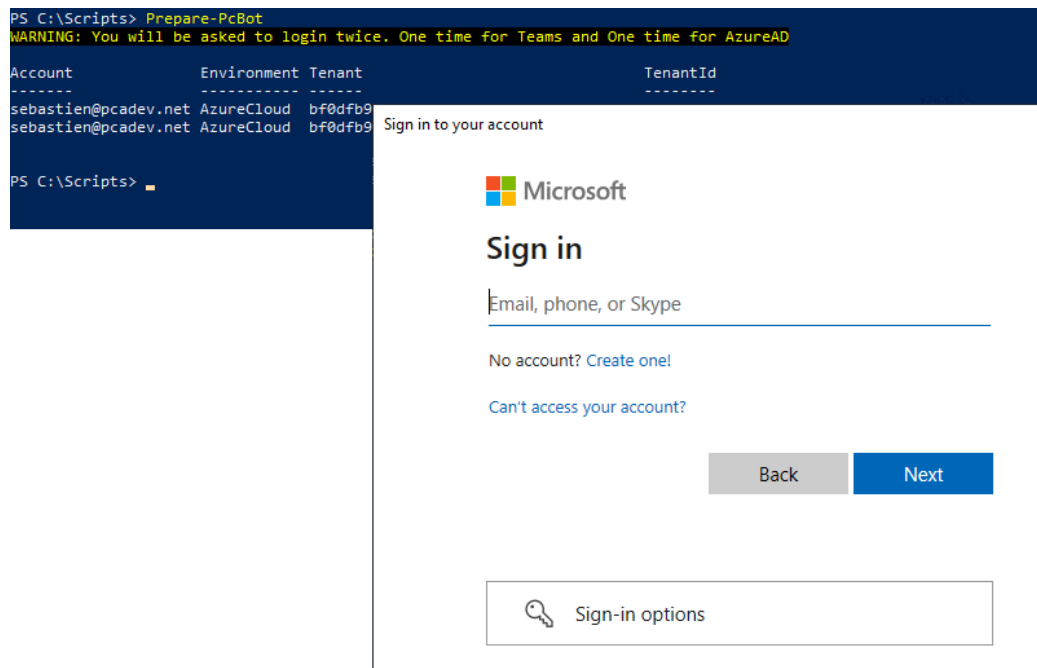
Step 3. Function: Prepare-PcBot

This function requires the user to login **twice**, in Teams Powershell and in the Azure ID Connector. It will merge the session into the current powershell session so that the required functions can be executed.

PowerShell
PS C:\> Prepare-PcBot

This function will then ask the user to **sign-in twice to connect Microsoft Teams and AzureAD**.

Use the Administrator account of your tenant (cf. [prerequisites](#))



Step 4. Function: Update-PcBotOnline

PowerShell

```
PS C:\> Update-PcBotOnline -UserPrincipalName receptionistqueue@yourdomain.com -ApplicationId 227ce992-e7a1-4695-a857-54b23670af35 -DisplayName "Receptionist Queue" -TelephoneNumber +3123456789 -UsageLocation NL
```

This command will create/update the endpoint for receptionist@yourdomain.com for the ApplicationID. It will use the Display Name when calling users.

⚠ Warning! Some steps can take time to sync inside Teams infra (min to hrs).
The script is only executing Teams Powershell commands. Execute the script again after waiting few minutes.

Parameters

-UserPrincipalName

A User Principal Name that you want for your endpoint [user@domain.xxx](#) (if you use ADFS with federation you can use the microsoft domain)

-ApplicationID

Enables you to specify the application ID for the Receptionist Preview Calling bot service ie 227ce992-e7a1-4695-a857-54b23670af35

-DisplayName

Enables you to specify the DisplayName (visible when it calls to users in your tenant)

-TelephoneNumber

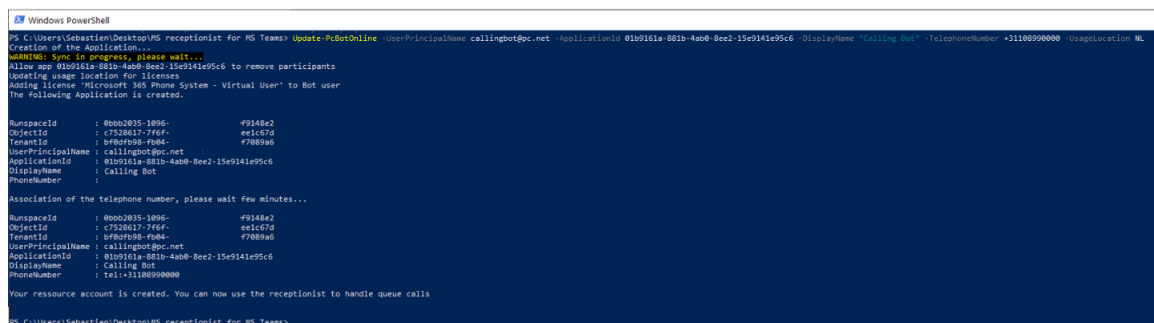
Enables you to assign a free service phone number previously ordered

-UsageLocation

Users 's location (country code used for the licensing) where is this phone located. The location **must match** the location of the phone number (cf. emergency location area)

The script will run the following actions:

1. Creation of the virtual user (*New-CsOnlineApplicationInstance*)
2. Sync it (**this can take several minutes**) (*Sync-CsOnlineApplicationInstance*)
3. Assign the virtual user license (*Set-AzureADUserLicense*)
4. Assign the system phone number (*Set-CsOnlineVoiceApplicationInstance*)
5. Additional rights to also remove participants from meetings as an app (*Set-CsApplicationMeetingConfiguration*)



```
PS C:\Users\Sebastian\Desktop\MS receptionist for MS Teams> Update-PcBotOnline userPrincipalName callingbot@pc.net ApplicationId 01b9161a-8b1b-4ab0-8ec1-15e9141e95c6 DisplayName "Calling Bot" TelephoneNumber +31180990000 UsageLocation NL
Creation of the Application...
WARNING: Sync is progress, please wait...
Allow app 01b9161a-8b1b-4ab0-8ec1-15e9141e95c6 to remove participants
Updating usage location for licenses
Adding license "Microsoft 365 Phone System - Virtual User" to Bot user
The following Application is created.
NameSpaceId      : 00000000-0000-0000-0000-000000000000
ObjectId         : c7528617-7f69-4e1c-b167-77089a5
TenantId        : b70cf988-f0d4-4b26-8000-000000000000
UserPrincipalName : callingbot@pc.net
ApplicationId     : 01b9161a-8b1b-4ab0-8ec1-15e9141e95c6
DisplayName      : Calling Bot
TelephoneNumber  :
Association of the telephone number, please wait few minutes...
NameSpaceId      : 00000000-0000-0000-0000-000000000000
ObjectId         : c7528617-7f69-4e1c-b167-77089a5
TenantId        : b70cf988-f0d4-4b26-8000-000000000000
UserPrincipalName : callingbot@pc.net
ApplicationId     : 01b9161a-8b1b-4ab0-8ec1-15e9141e95c6
DisplayName      : Calling Bot
TelephoneNumber  : tel:+31180990000
Your resource account is created. You can now use the receptionist to handle queue calls
PS C:\Users\Sebastian\Desktop\MS receptionist for MS Teams>
```

Figure 1. Endpoint creation example

Last Step : Dial Outside and checks

In Microsoft 365 Admin Center search for your Virtual User (queue).

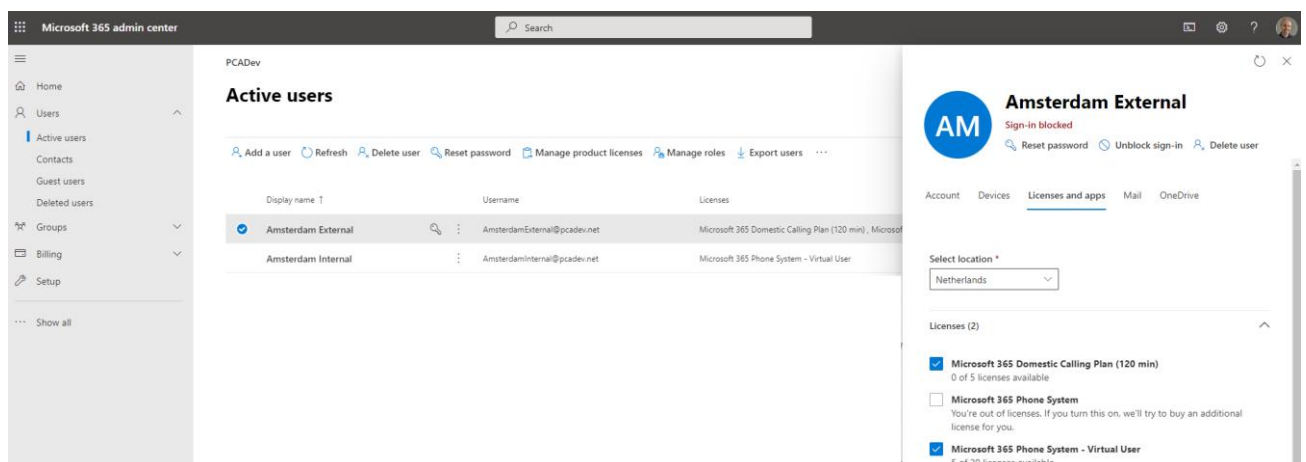
This user must have the ☒ **Microsoft 365 Phone System - Virtual User** assigned by the script.

PSTN OUT to call outside :

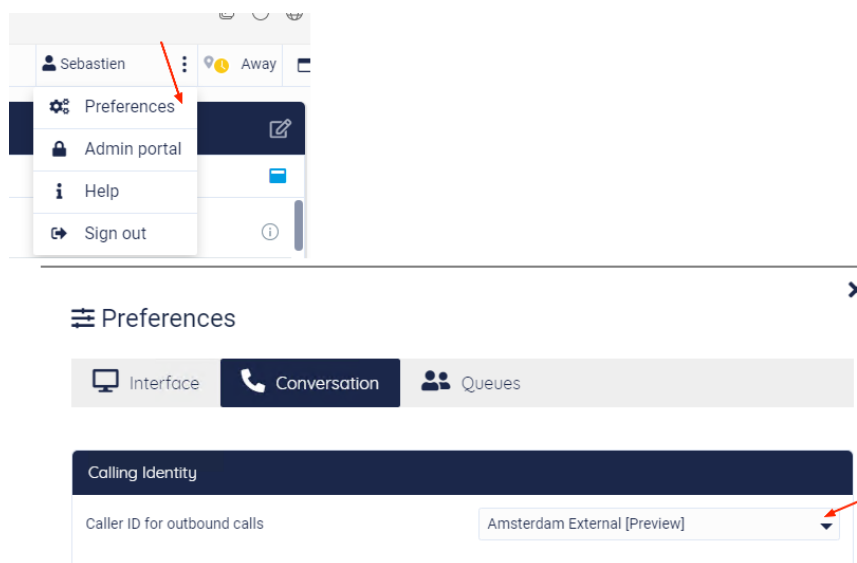
To call outside, you must assign the corresponding Calling License to the Virtual User (queue)

In Microsoft 365 Admin Center, search for the virtual user and assign the corresponding license :

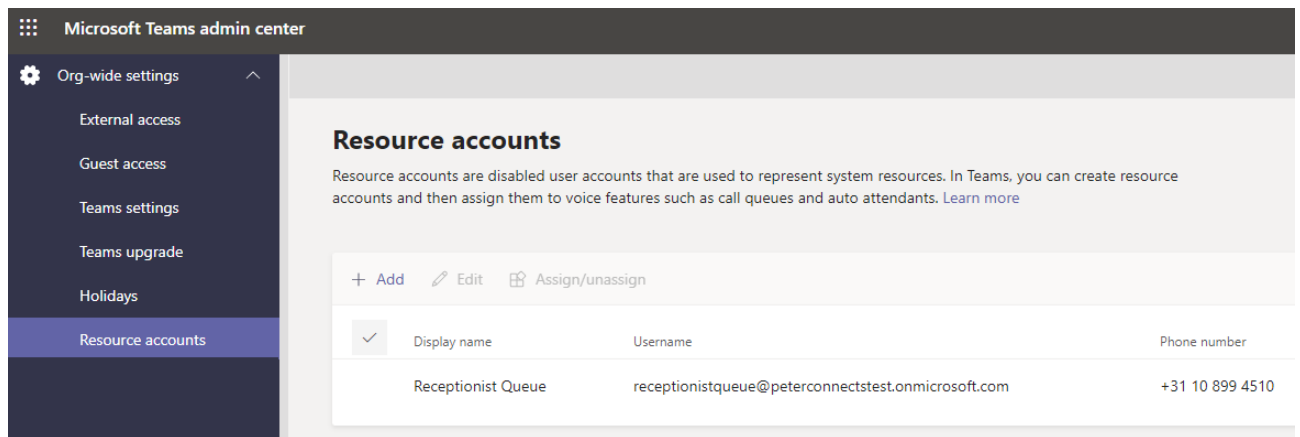
- Microsoft 365 **Domestic Calling Plan**,
- OR
- Microsoft 365 **Domestic and International Calling Plan...**



In the Attendant Console client (later), you have to specify the Queue used to dial out:



In Microsoft Teams admin center, under **Org-Wide Settings**, you can verify the resource account with the phone number assigned:



The screenshot shows the Microsoft Teams admin center interface. On the left is a dark sidebar with a list of settings categories: Org-wide settings, External access, Guest access, Teams settings, Teams upgrade, Holidays, and Resource accounts (which is highlighted). The main content area is titled 'Resource accounts' and includes a descriptive paragraph about resource accounts and a 'Learn more' link. Below this is a table with columns for a selection checkbox, Display name, Username, and Phone number. One resource account is listed: 'Receptionist Queue' with the username 'receptionistqueue@peterconnectstest.onmicrosoft.com' and the phone number '+31 10 899 4510'.

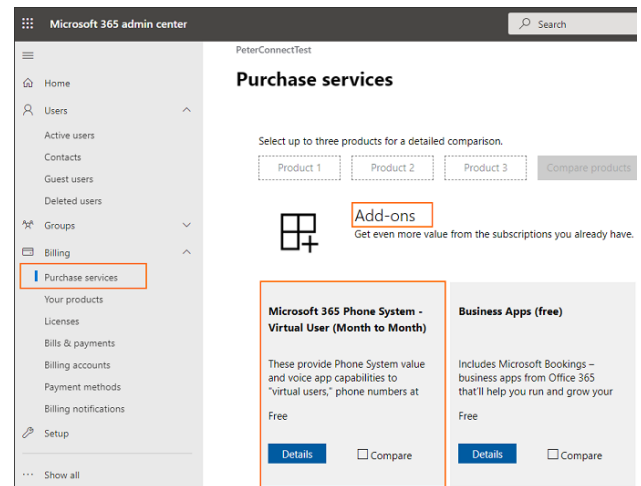
	Display name	Username	Phone number
<input checked="" type="checkbox"/>	Receptionist Queue	receptionistqueue@peterconnectstest.onmicrosoft.com	+31 10 899 4510

Teams with Direct Routing

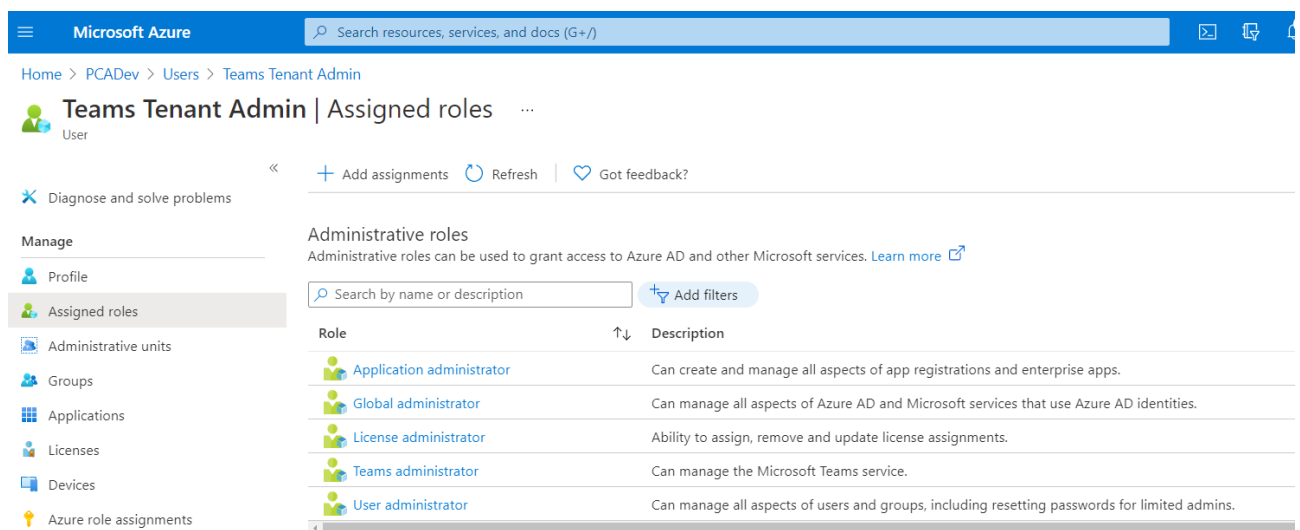
Prerequisites

1. Direct Routing must be configured for in/out calls
2. Microsoft 365 phone system – Virtual User add-on License (free)

One virtual license is required per Calling Queue number ([more info](#))



3. An Azure AD user with the following roles on your tenant (@yourteamtenant.com):
 - *Global Administrator* (only used for the consent steps, **can be removed after**)
 - *Application Administrator*
 - *Teams administrator*
 - *License administrator*
 - *User Administrator*

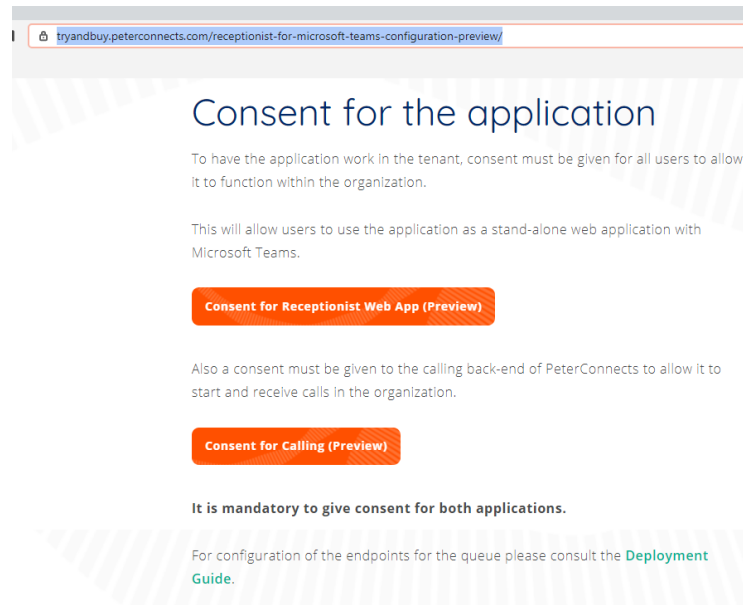


Those *rights* are subject to change for the GA Release.

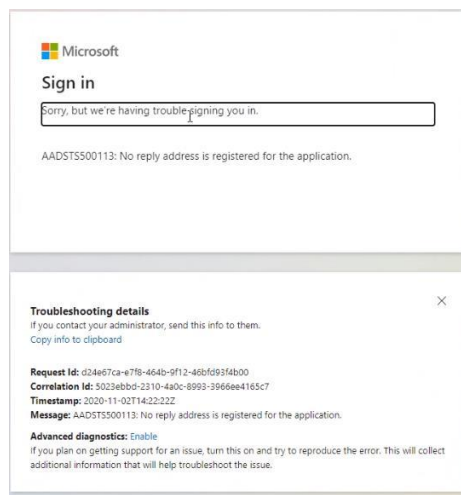
Step 1. Consent for the Applications

Consent for **both Apps** must have been done via the following webpage **before** going to step2:

<https://tryandbuy.peterconnects.com/receptionist-for-microsoft-teams-configuration-preview/>



After consenting (Agree) you can get the following page. **This is intended** as we do not have supplied a reply URL for the preview App.



Once this is done, you can continue.

1. Install Teams Powershell and AzureAD module

```
PS C:\ > Install-Module MicrosoftTeams
```

```
Administrator: Windows PowerShell
PS C:\Users\salesdemoadmin> Install-Module MicrosoftTeams

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): a
```

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Installing package 'AzureAD'
  Unzipping
  [oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo]

PS C:\Users\admin> Install-Module AzureAD

NuGet provider is required to continue
PowerShellGet requires NuGet provider version '2.8.5.201' or newer to interact with NuGet-based repositories. The NuGet
provider must be available in 'C:\Program Files\Package Management\ProviderAssemblies' or
'C:\Users\salesdemoadmin\AppData\Local\PackageManagement\ProviderAssemblies'. You can also install the NuGet provider
by running 'Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force'. Do you want PowerShellGet to install
and import the NuGet provider now?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): a

PS C:\Users\admin>
```

Enter the **exact** following syntax (**point space point antislash**) to load the script

```
. .\Add-PcBotAppInstance.ps1
```

```

PS C:\Scripts> . .\Add-PcBotAppInstance.ps1
PS C:\> Prepare-PcBot
WARNING: You will be asked to login twice. One time for Teams and One time for AzureAD

Account Environment Tenant TenantId
-----
sebastien@pcadev.net AzureCloud bf0dfb98-fb04-42b8-984f-34283f7089a6 bf0dfb98-fb04-42b8-984f-34283f7089a6
sebastien@pcadev.net AzureCloud bf0dfb98-fb04-42b8-984f-34283f7089a6 bf0dfb98-fb04-42b8-984f-34283f7089a6

```

15

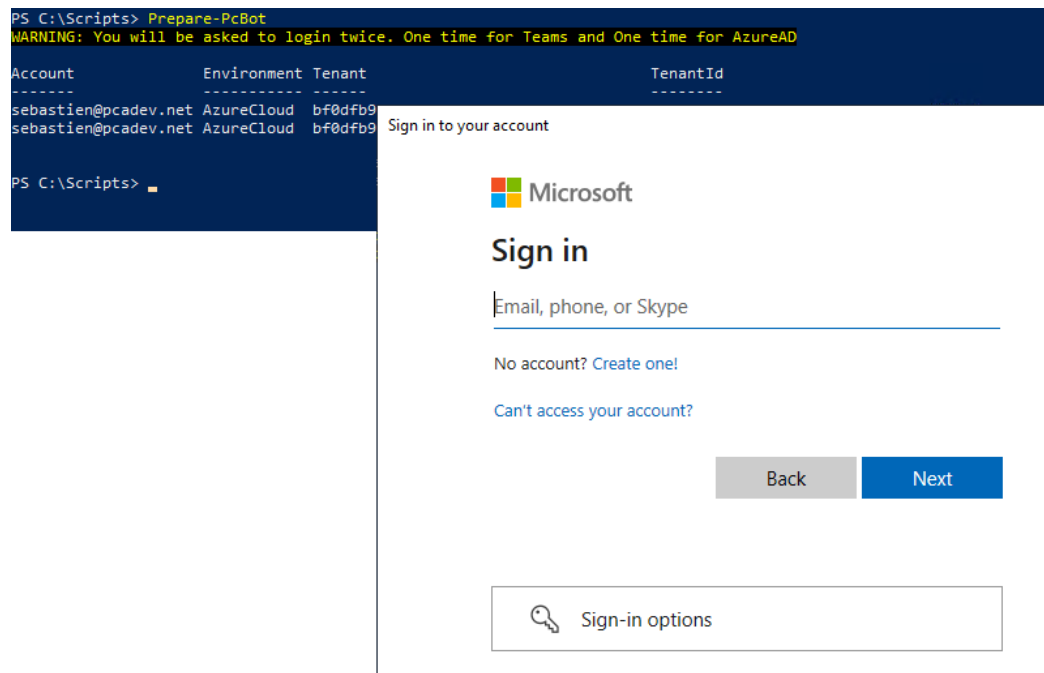
Step 3. Function: Prepare-PcBot

This function requires the user to login **twice**, in Teams Powershell and in the Azure ID Connector. It will merge the session into the current powershell session so that the required functions can be executed.

PowerShell
PS C:\> Prepare-PcBot

This function will then ask the user to **sign-in twice to connect Microsoft Teams and AzureAD**.

Use the Administrator account of your tenant (cf. [prerequisites](#))



The screenshot shows a PowerShell terminal window on the left and a Microsoft Sign in page on the right. The terminal window displays the command `PS C:\Scripts> Prepare-PcBot` and a warning message: `WARNING: You will be asked to login twice. One time for Teams and One time for AzureAD`. Below the warning, a table lists account information:

Account	Environment	Tenant	TenantId
sebastien@pcadev.net	AzureCloud	bf0dfb9	
sebastien@pcadev.net	AzureCloud	bf0dfb9	

The terminal window also shows the prompt `PS C:\Scripts>` . The Microsoft Sign in page on the right features the Microsoft logo, the text "Sign in", a text input field for "Email, phone, or Skype", and links for "No account? Create one!" and "Can't access your account?". There are "Back" and "Next" buttons, and a "Sign-in options" link at the bottom.

Step 4. Function: Update-PcBotDirect

PowerShell

```
PS C:\> Update-PcBotDirect -UserPrincipalName receptionist@yourdomain.com -ApplicationId 227ce992-e7a1-4695-a857-54b23670af35 -DisplayName "Receptionist Queue" -TelephoneNumber +3123456789 -UsageLocation NL
```

This command will create/update the endpoint for receptionist@yourdomain.com for the ApplicationID. It will use the Display Name when calling users. You must provide a phone number from your Direct Routing plan.

👉 Warning! Some steps can take time to sync inside Teams infra (min to hrs).

The script is only executing Teams Powershell commands. Execute the script again after waiting few minutes.

Parameters

-UserPrincipalName

A User Principal Name that you want for your endpoint [user@domain.xxx](#) (if you use ADFS with federation you can use the Microsoft domain)

-ApplicationID

Enables you to specify the application ID for the Receptionist Preview Callingbot service ie **227ce992-e7a1-4695-a857-54b23670af35**

-DisplayName

Enables you to specify the DisplayName (visible when it calls to users in your tenant)

-TelephoneNumber

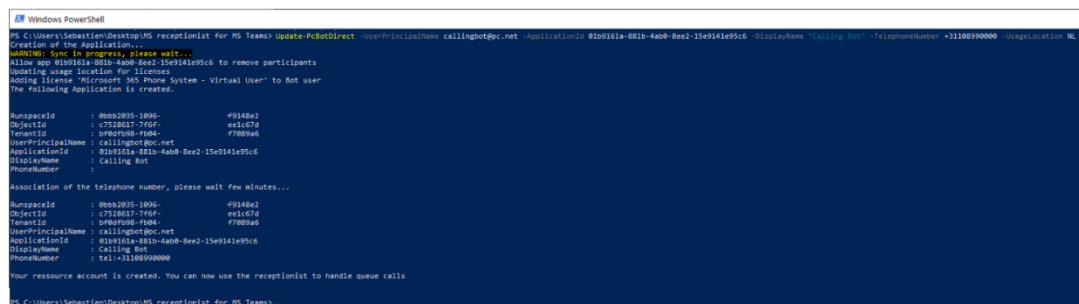
Enables you to assign a phone number from your direct routing plan

-UsageLocation

Users 's location (country code used for the licensing) where is this phone located. The location **must match** the location of the phone number (cf. emergency location area)

The script will run the following actions:

4. Creation of the virtual user (*New-CsOnlineApplicationInstance*)
5. Sync it (**this can take several minutes**) (*Sync-CsOnlineApplicationInstance*)
6. Assign the virtual user license (*Set-AzureADUserLicense*)
7. Assign the system phone number (*Set-CsOnlineApplicationInstance*)
8. Additional rights to also remove participants from meetings as an app (*Set-CsApplicationMeetingConfiguration*)



```
PS C:\Users\Sebastian\Desktop\MS receptionist for MS Teams> Update-PcBotDirect -UserPrincipalName callingbot@pc.net -ApplicationId 227ce992-e7a1-4695-a857-54b23670af35 -DisplayName "Calling Bot" -TelephoneNumber +31100990000 -UsageLocation NL

Creation of the Application...
Warning: Sync is progress, please wait!
Allow app 227ce992-e7a1-4695-a857-54b23670af35 to remove participants
Updating usage location for licenses
Adding license 'Microsoft 365 Phone System - Virtual User' to Bot user
The following Application is created.

RunspaceId      : 00b02035-1096-4914-8a22-79206217766f
ObjectId        : c7920621-766f-4e1c-87d8-7f009a6
TenantId        : 97609f06-f0b4-4780-b060-7f009a6
UserPrincipalName : callingbot@pc.net
ApplicationId    : 227ce992-e7a1-4695-a857-54b23670af35
DisplayName     : Calling Bot
TelephoneNumber : Tel:+31100990000

Association of the telephone number, please wait few minutes...

RunspaceId      : 00b02035-1096-4914-8a22-79206217766f
ObjectId        : c7920621-766f-4e1c-87d8-7f009a6
TenantId        : 97609f06-f0b4-4780-b060-7f009a6
UserPrincipalName : callingbot@pc.net
ApplicationId    : 227ce992-e7a1-4695-a857-54b23670af35
DisplayName     : Calling Bot
TelephoneNumber : Tel:+31100990000

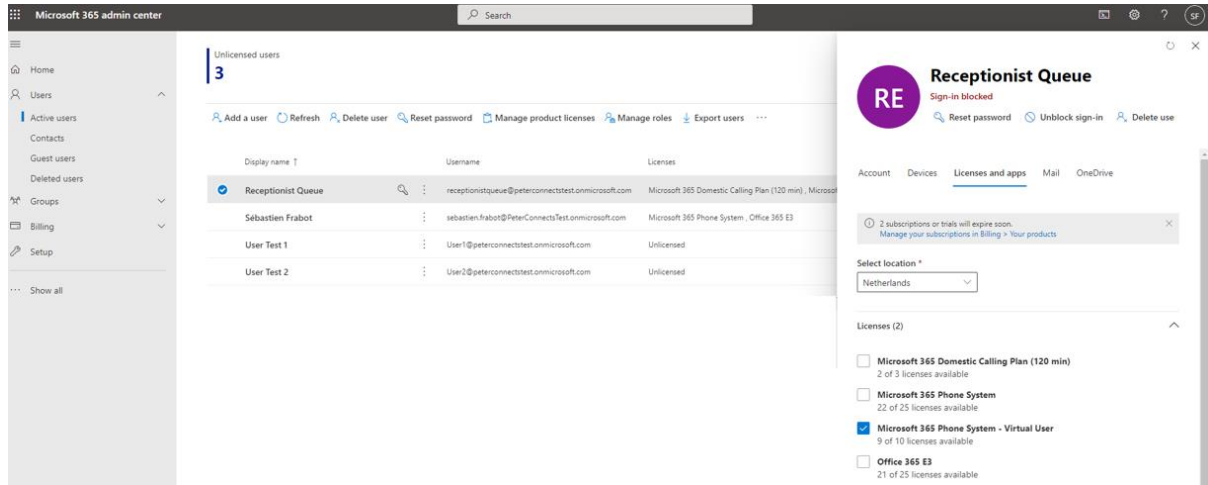
Your resource account is created. You can now use the receptionist to handle queue calls

PS C:\Users\Sebastian\Desktop\MS receptionist for MS Teams>
```

Figure 2. Example of script done

Last Step : Dial Outside and Checks

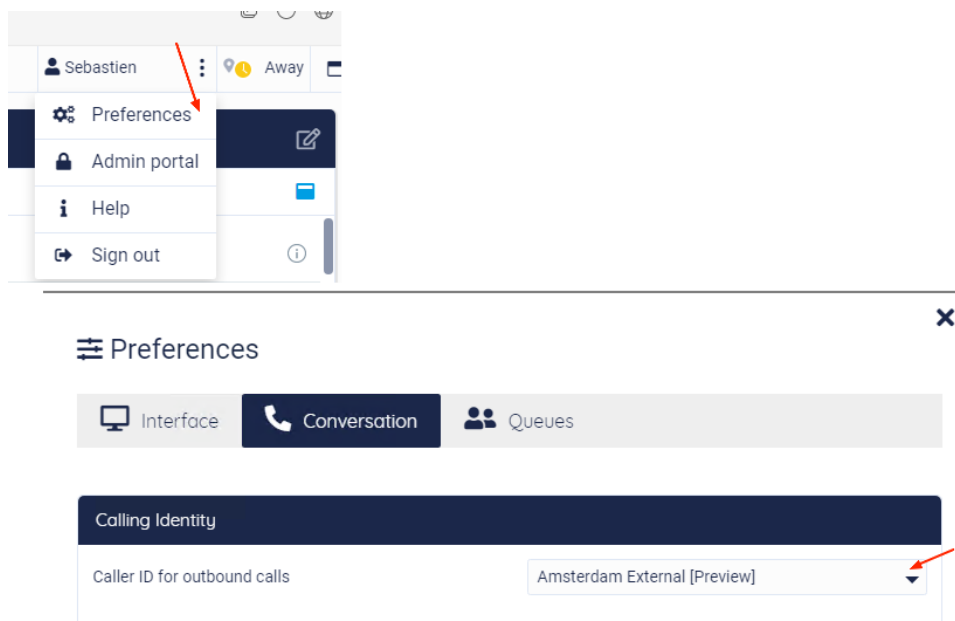
You can check that has ☒ **Microsoft 365 Phone System - Virtual User** already been assigned by the script to the resource account created.



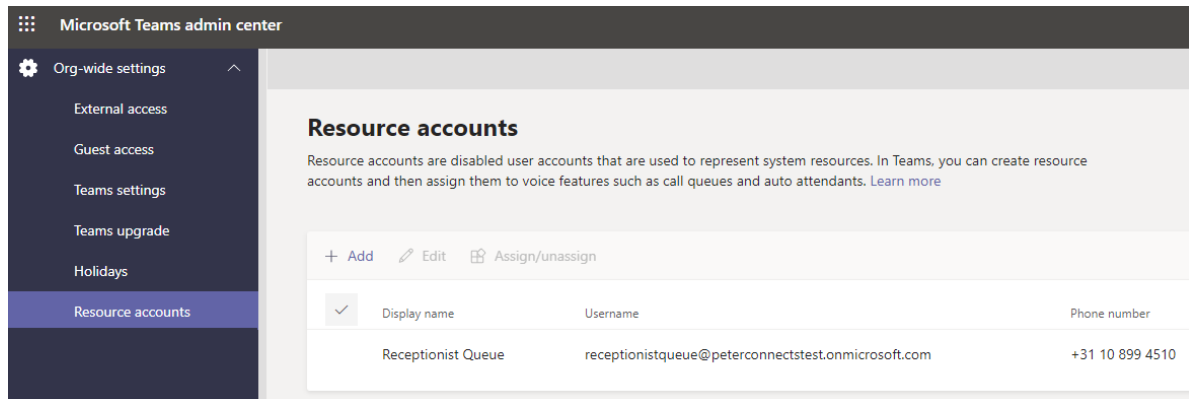
Dial Outside:

To be able to call out, you must validate that your Direct Routing Policy and route allows the Virtual User (queue) call to be routed outside in a National or International dialing plan.

In the Attendant Console client, **Preferences**, you will specify which Queue is used to dial outside:



In Teams admin center, you can list the Resource accounts created with the DID assigned.



The screenshot shows the Microsoft Teams admin center interface. On the left is a dark sidebar with a gear icon and the text "Microsoft Teams admin center". Below this are several menu items: "Org-wide settings" (with an upward arrow), "External access", "Guest access", "Teams settings", "Teams upgrade", "Holidays", and "Resource accounts" (which is highlighted in blue). The main content area has a light gray header with the title "Resource accounts". Below the title is a paragraph explaining that resource accounts are disabled user accounts used for system resources, and that they can be assigned to voice features like call queues and auto attendants, with a "Learn more" link. Below this text is a toolbar with three icons: a plus sign for "Add", a pencil for "Edit", and a box with a plus sign for "Assign/unassign". Below the toolbar is a table with three columns: "Display name", "Username", and "Phone number". The table contains one row with the following data: "Receptionist Queue" for the display name, "receptionistqueue@peterconnectstest.onmicrosoft.com" for the username, and "+31 10 899 4510" for the phone number. A small checkmark icon is visible in the first column of the table header.

✓	Display name	Username	Phone number
	Receptionist Queue	receptionistqueue@peterconnectstest.onmicrosoft.com	+31 10 899 4510

Teams Admin Center 3rd party apps

“Peterconnects receptionist from Teams” is a third-party application, please refer to Microsoft documentation to manage your apps:

<https://docs.microsoft.com/en-us/MicrosoftTeams/manage-apps#:~:text=%20Manage%20your%20apps%20in%20the%20Microsoft%20Teams,where%20you%20allow%20or%20block%20individual...%20More%20>

Permission Policy

You can edit or create a Policy to restrict or authorize the user to add Third-party apps, and apply it to specific users:

The screenshot shows the 'Microsoft Teams admin center' interface. The left sidebar contains navigation links: Dashboard, Teams, Devices, Locations, Users, Meetings, Messaging policies, Teams apps, Manage apps, Permission policies (selected), Setup policies, Customize store, Voice, Policy packages, Analytics & reports, and Org-wide settings. The main content area is titled 'App permission policies \ Receptionist Policy'. It features a 'Description' field, three sections for app permissions: 'Microsoft apps' (set to 'Allow all apps'), 'Third-party apps' (set to 'Allow all apps'), and 'Custom apps' (set to 'Allow all apps'). At the bottom are 'Save' and 'Cancel' buttons.

This screenshot shows the 'App permission policies' page in the Microsoft Teams admin center. The left sidebar is identical to the previous screenshot. The main content area has a title 'App permission policies' and a brief description. Below the description is a toolbar with buttons: '+ Add', 'Edit', 'Duplicate', 'Delete', 'Reset Global policy', and 'Manage users'. A table lists the policies:

✓	Name ↑	Description
✓	Receptionist Policy	
	Global (Org-wide default)	

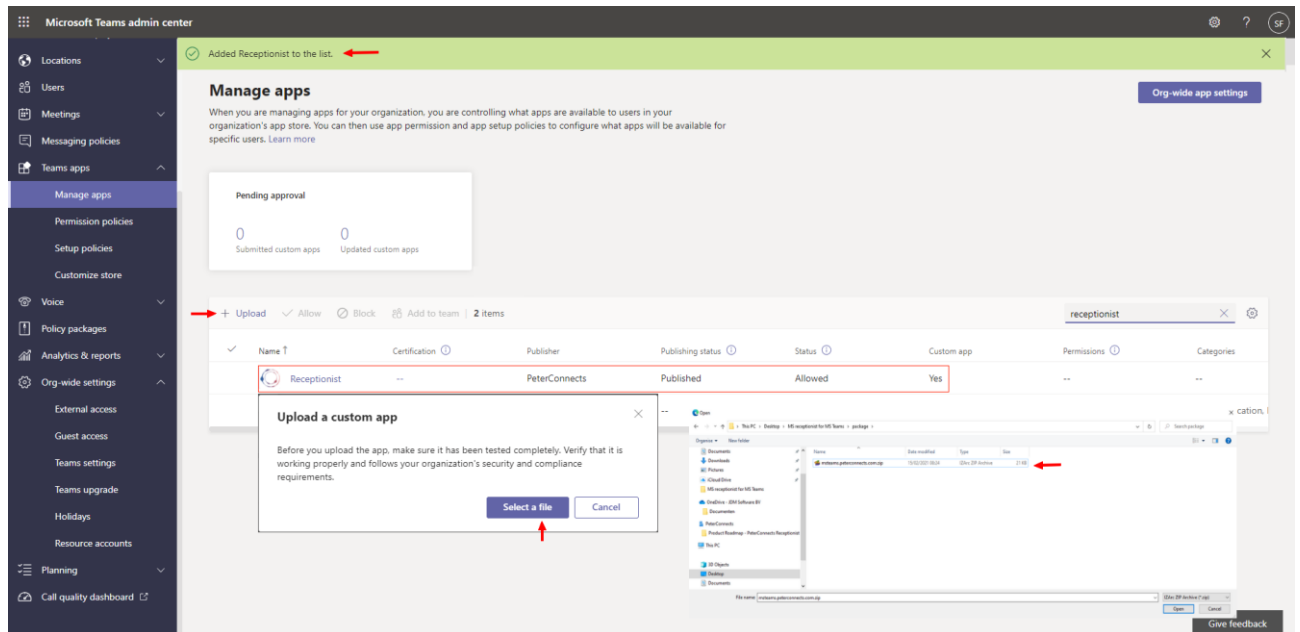
A red arrow points from the 'Manage users' button in the toolbar to the 'Manage users' dialog box shown in the next screenshot.

The 'Manage users' dialog box is shown, titled 'Manage users' with a subtitle 'Receptionist Policy'. It features a search bar with the text 'reception' and a list of results: 'RM Receptionist Maassluis' and 'RP Receptionist Paris'. Below the list is an information icon and the text 'To add users, search by their display or username.' At the bottom are 'Apply' and 'Cancel' buttons.

Manage the application

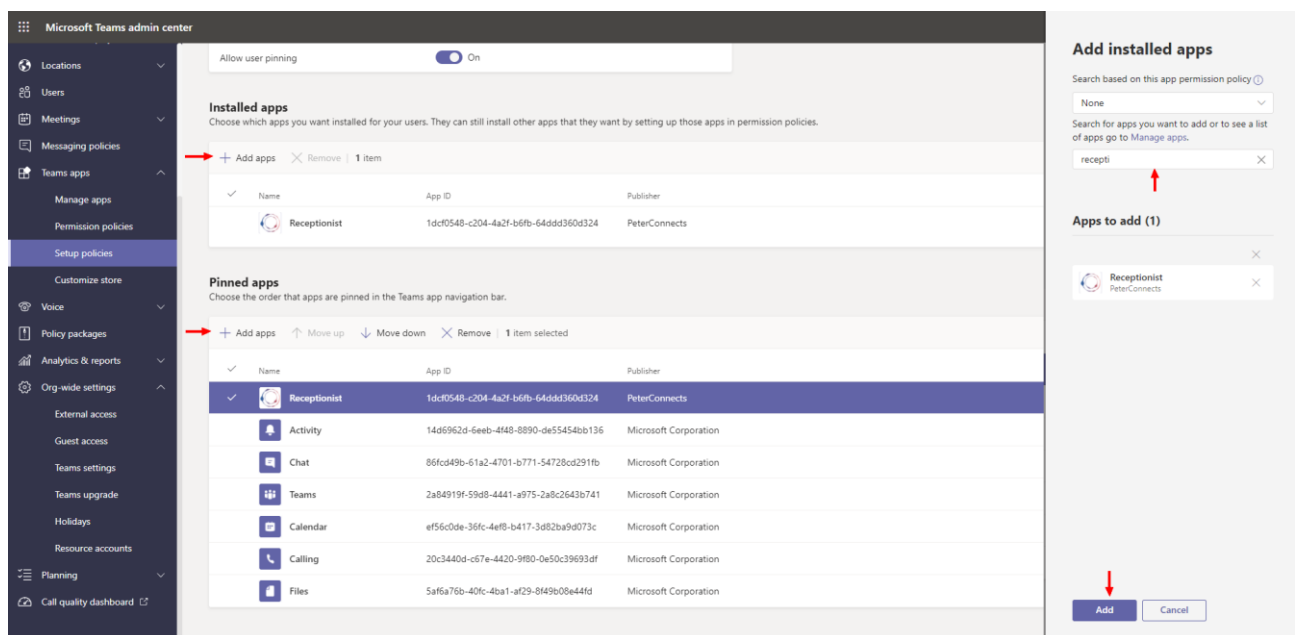
The Teams administrator can upload the app from the “Teams apps – Manage Apps” menu. Click on “Upload” and select the PeterConnects Receptionist Application then “Open”.

Upload the Receptionist App zip file: msteams.peterconnects.com.zip



Setup Policies

You can then choose the application to be installed and Pinned it at the right position

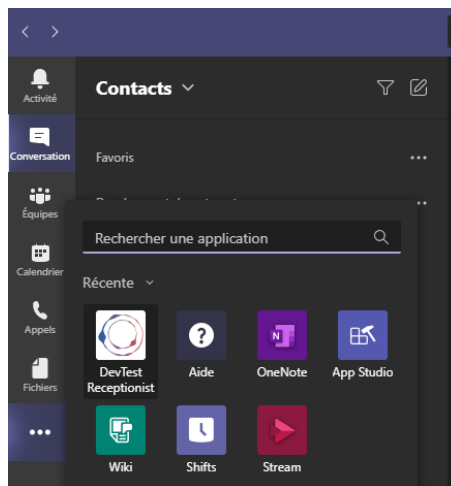


Peterconnects Receptionist in the Teams client

Click on the 3 horizontal dots and select the Peterconnects Receptionist.

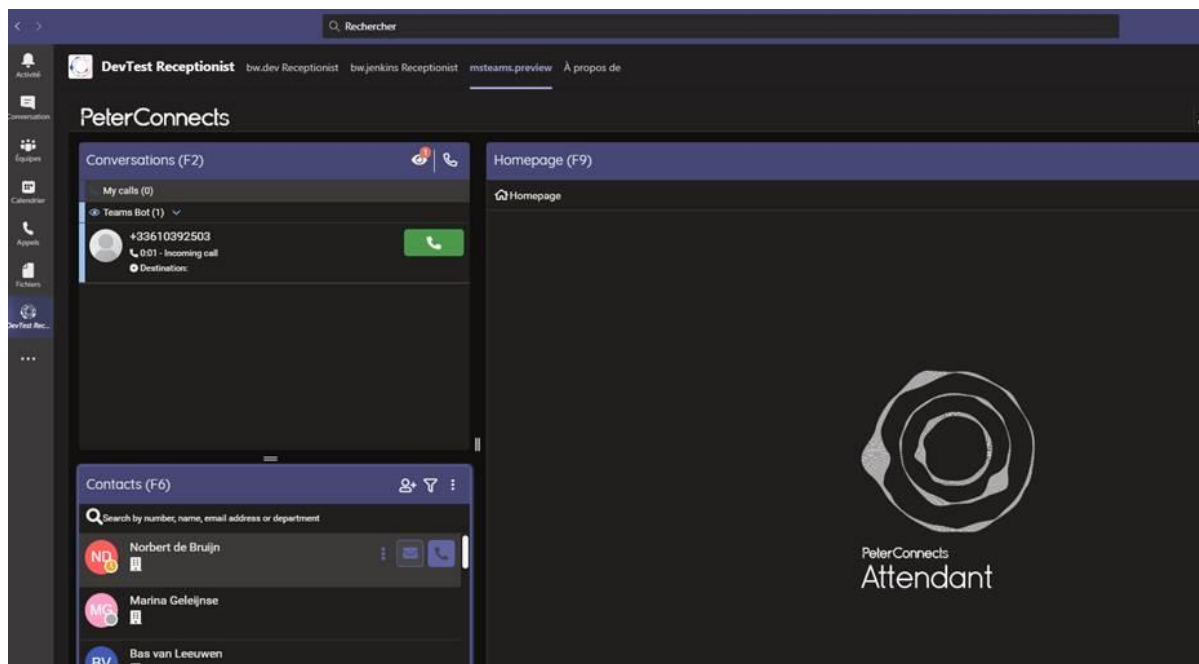
- click on the Apps “Receptionist Preview” icon in Teams

Install it as a New application :



The Receptionist is then available and you can open it.

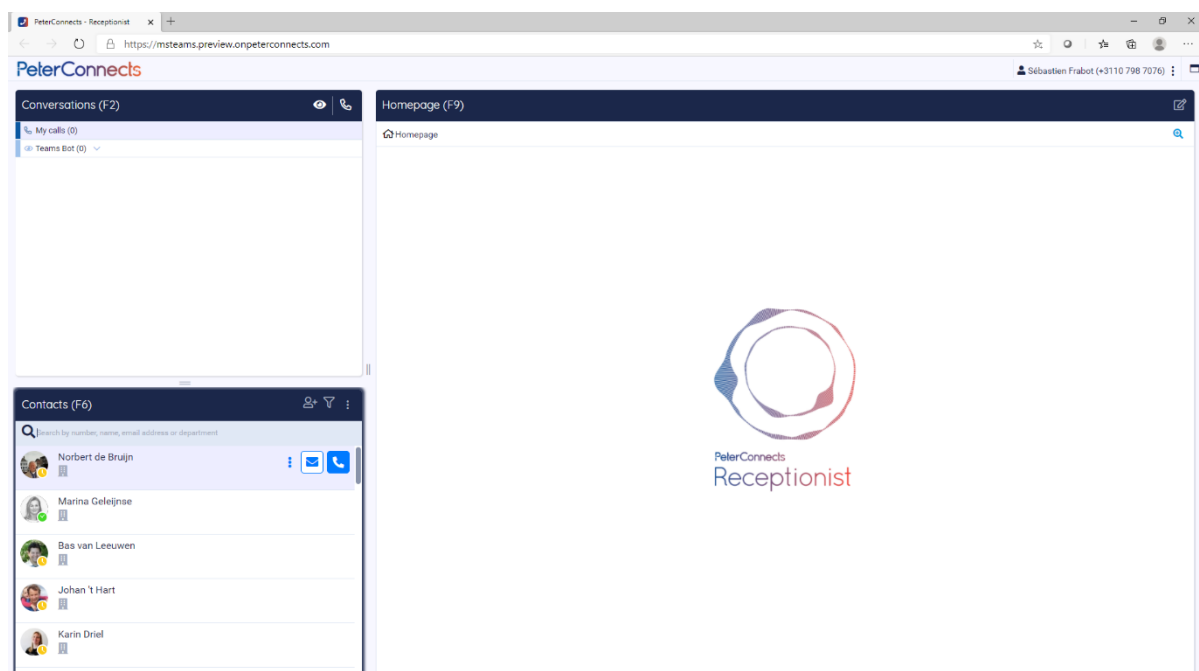
Once opened, pin it to Teams with right-click on the icon in the left bar and select "Pin"



Access the Peterconnects Receptionist via a webbrowser outside the Teams Client

You can launch the following URL to access PeterConnects Receptionist for Teams:

<https://msteams.preview.onpeterconnects.com/>

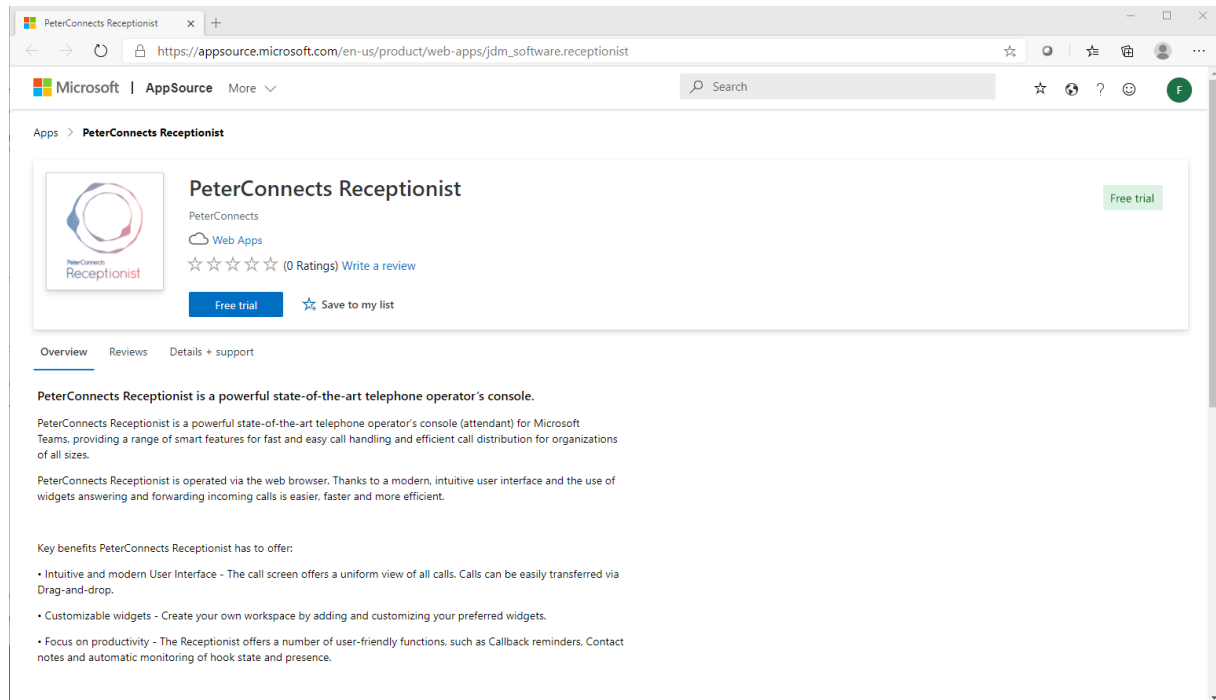


Microsoft App Store

PeterConnects Receptionist from the Microsoft App Store

You can now download the application from Microsoft App Store at this URL:

https://appsource.microsoft.com/en-us/product/web-apps/jdm_software.receptionist



The screenshot shows a web browser window displaying the Microsoft App Store page for the 'PeterConnects Receptionist' application. The browser's address bar shows the URL: https://appsource.microsoft.com/en-us/product/web-apps/jdm_software.receptionist. The page header includes the Microsoft logo and 'AppSource' branding. The main content area features the application's icon, name, and a 'Free trial' button. Below this, there are tabs for 'Overview', 'Reviews', and 'Details + support'. The 'Overview' tab is selected, showing a description of the application as a powerful state-of-the-art telephone operator's console for Microsoft Teams. It highlights key benefits such as an intuitive user interface, customizable widgets, and a focus on productivity. The page also includes a 'Write a review' link and a 'Save to my list' button.

PeterConnects Receptionist
PeterConnects
Web Apps
☆ ☆ ☆ ☆ ☆ (0 Ratings) [Write a review](#)
[Free trial](#) ☆ [Save to my list](#)

Overview | [Reviews](#) | [Details + support](#)

PeterConnects Receptionist is a powerful state-of-the-art telephone operator's console.

PeterConnects Receptionist is a powerful state-of-the-art telephone operator's console (attendant) for Microsoft Teams, providing a range of smart features for fast and easy call handling and efficient call distribution for organizations of all sizes.

PeterConnects Receptionist is operated via the web browser. Thanks to a modern, intuitive user interface and the use of widgets answering and forwarding incoming calls is easier, faster and more efficient.

Key benefits PeterConnects Receptionist has to offer:

- Intuitive and modern User Interface - The call screen offers a uniform view of all calls. Calls can be easily transferred via Drag-and-drop.
- Customizable widgets - Create your own workspace by adding and customizing your preferred widgets.
- Focus on productivity - The Receptionist offers a number of user-friendly functions, such as Callback reminders, Contact notes and automatic monitoring of hook state and presence.

References

- <https://docs.microsoft.com/en-us/microsoftteams/teams-powershell-overview>
- <https://docs.microsoft.com/en-us/microsoftteams/teams-add-on-licensing/virtual-user>
- <https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/deploy-and-publish/apps-upload>
- <https://tryandbuy.peterconnects.com/microsoft-teams-receptionist-buy>
- https://appsource.microsoft.com/en-us/product/web-apps/jdm_software.receptionist

Q&A

Scripts Execution error “script cannot be loaded because running scripts is disabled on this system”

```
PS C:\Scripts> .\Add-PcBotAppInstance.ps1
. : File C:\Scripts\Add-PcBotAppInstance.ps1 cannot be loaded because running scripts is disabled on this system. For
more information, see about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:3
+ .\Add-PcBotAppInstance.ps1
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Scripts>
```

Answer: you execution policy is restricted. Please checkit with `Get-ExecutionPolicy` and `Set-Execution policy` to enabled script load.

Scripts Execution error “The term ‘function’ is not recognized as the name of a cmdlet”

Running a function returns such an error:

```
Prepare-PcBot : The term 'Prepare-PcBot' is not recognized as the name of a cmdlet, function, script file,
or operable program. Check the spelling of the name, or if a path was included, verify that the path is
correct and try again.
At line:1 char:1
+ Prepare-PcBot
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Prepare-PcBot:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

Answer: check that you run the ps1 script with the right path syntax (point space point\)

```
PS C:> .\Add-PcBotAppInstance.ps1
```

Function Prepare-PcBot Execution error “Connect-MicrosoftTeams” term is not recognized

```
PS C:\script> Prepare-PcBot
WARNING: You will be asked to login twice. One time for Teams and One time for AzureAD
Connect-MicrosoftTeams : The term 'Connect-MicrosoftTeams' is not recognized as the name of a cmdlet, function, script
file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct
and try again.
At C:\script\Add-PcBotAppInstance.ps1:22 char:5
+ Connect-MicrosoftTeams
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Connect-MicrosoftTeams:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

Connect-AzureAD : The term 'Connect-AzureAD' is not recognized as the name of a cmdlet, function, script file, or
operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try
again.
At C:\script\Add-PcBotAppInstance.ps1:23 char:5
+ Connect-AzureAD
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Connect-AzureAD:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
PS C:\script>
```

Answer: install or update to the latest MicrosoftTeams powershell module to be able to use this command.
You can check this command is listed and then available via `Get-Command -module MicrosoftTeams`

Function Prepare-PcBot Execution error “Connect-AzureAD” is not recognized

```
Connect-AzureAD : The term 'Connect-AzureAD' is not recognized as the name of a cmdlet, function, script file, or
operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try
again.
At C:\script\Add-PcBotAppInstance.ps1:23 char:5
+ Connect-AzureAD
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Connect-AzureAD:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\script>
```

Answer: install the AzureAD powershell module to be able to use this command

You can check this command is listed and then available via `Get-Command -module AzureAD`

Function Update-PcBotOnline or Update-PcBotDirect execution error

“The term ‘Get-CsOnlineApplicationInstance’ is not recognized as the name of a cmdlet”.

Running Update-PcBotDirect or Update-PcBotOnline returns the following error:

```
Get-CsOnlineApplicationInstance : The term 'Get-CsOnlineApplicationInstance' is not recognized as the name of a
cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify
that the path is correct and try again.
At C:\Users\Sebastien\Desktop\MS receptionist for MS Teams\Add-PcBotAppInstance.ps1:87 char:19
+ $CurrentApp = Get-CsOnlineApplicationInstance -Identity $UserPrin ...
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Get-CsOnlineApplicationInstance:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

Answer: Please verify Teams PowerShell is correctly installed and that Prepare-PcBot has been done before running any Update-PcBotxxx commands

Function Update-PcBotOnline or Update-PcBotDirect execution error

Execution error “Set-AzureADUser : Error occurred while executing SetUser”.

Running Update-PcBotDirect or Update-PcBotOnline returns the following error:

```
Set-AzureADUser : Error occurred while executing SetUser
Code: Authorization_RequestDenied
Message: Insufficient privileges to complete the operation.
RequestId: 21de8203-3b2e-4ad7-aa6e-a5f422d7e103
DateTimeStamp: Thu, 04 Feb 2021 10:13:27 GMT
HttpStatusCode: Forbidden
HttpStatusDescription: Forbidden
HttpResponseStatus: Completed
At C:\Users\Sebastien\Desktop\MS receptionist for MS Teams\Add-PcBotAppInstance.ps1:316 char:9
+ Set-AzureADUser -ObjectId $BotUser.UserPrincipalName -UsageLo ...
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [Set-AzureADUser], ApiException
+ FullyQualifiedErrorId : Microsoft.Open.AzureAD16.Client.ApiException,Microsoft.Open.AzureAD16.PowerShell.SetUser
```

Answer: the user you used to start Teams PowerShell session has not enough right to create an endpoint in Teams. Verify your account role assignment (User Administrator role) according to our prerequisites.

Where can I find the Teams Anywhere 355 Attendant Console App?

Answer: The Receptionist application can be downloaded directly from Microsoft AppSource:

https://appsource.microsoft.com/en-us/product/web-apps/jdm_software.receptionist

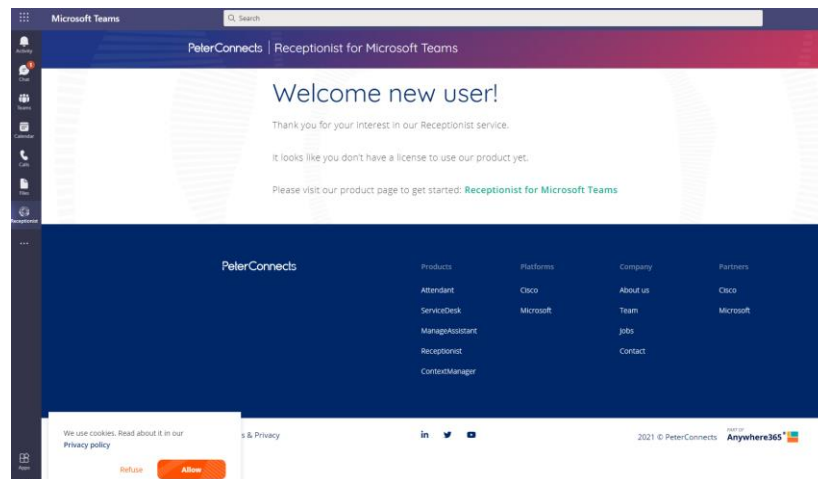
The Exchange agenda is indicated “Agenda not available”.

Answer1: your administrator must have consented the Calling App to be able to read Exchange Agenda.

Answer2: You can as well check on your Attendant Admin portal that the correct method configuration to query the agenda is selected. (Preferences – Admin Portal)

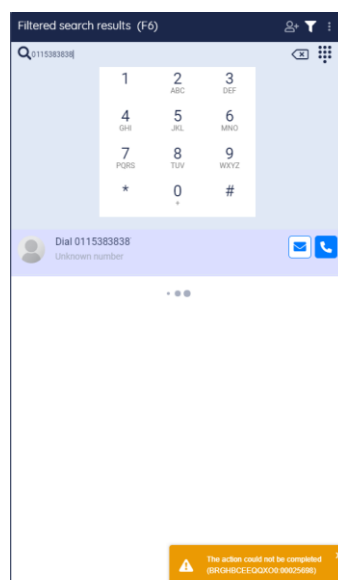
Answer3: The Teams Attendant Console user must have a Microsoft License (E3/E5) to use a mailbox. Test with logging on outlook.microsoft.com

When launching the receptionist App in Teams, I got a “welcome new user!” webpage.



Answer: the user you logged on Teams with doesn't have a Receptionist License.

Error message when dialing out “The action could not be completed”



Answer1: you must have selected a queue to dial-out in Preferences-Conversation tab of the receptionist

Answer2: When using Microsoft Calling Plan, you must assign a domestic-international license to the queue used to dial out