



Pexip One-Touch Join

Deployment Guide

Software Version 32

Document Version 32.a

June 2023

] pexip [

Contents

About One-Touch Join	6
Enabling One-Touch Join	6
Supported Google Workspace editions	7
Supported Exchange environments	7
Exchange servers	7
Outlook clients	7
Supported endpoints	7
Cisco OBTP	8
Poly OTD	8
Supported meeting types	9
Supported number of endpoints and Conferencing Nodes	9
Pexip Infinity server requirements	9
One-Touch Join process and deployment overview	10
Process overview	10
Administrator configures OTJ	10
End user sends invitation	10
OTJ provides endpoint with meeting information	10
Frequency and limitations on calendar requests	11
Locations, Conferencing Nodes and redundancy	12
Conferencing Nodes	12
Management Node	12
Network architecture, firewalls and web proxy	12
Conferencing Nodes	12
Management Node	13
Port usage	13
Permitting the service account to access calendars	14
Exchange integrations	14
Google Workspace integrations	15
Using One-Touch Join with personal endpoints and calendars	15
Configuring Exchange on-premises for One-Touch Join	16
Prerequisites	16
Creating a service account	16
Configuring Application Impersonation on the service account	17
Creating a new Distribution Group	17
Configuring application impersonation	18
Enabling authentication	18
NTLMv2 authentication	19
Basic authentication	19
Configuring calendar processing on room resource mailboxes	19
Recommended configuration	20

Optional configuration	20
Checking calendar processing settings	21
Adding a One-Touch Join Exchange integration on Pexip Infinity	23
Next steps	24
Configuring Office 365 using Graph for One-Touch Join	25
Prerequisites	25
Creating and configuring a new App registration in Azure	25
Restricting the scope of the App registration	29
Creating a Distribution Group	30
Restricting access	30
Configuring calendar processing on room resource mailboxes	30
Recommended configuration	30
Optional configuration	31
Checking calendar processing settings	32
Adding a One-Touch Join O365 Graph integration on Pexip Infinity	34
Configuring the O365 Graph integration	34
Next steps	34
Configuring Office 365 using EWS for One-Touch Join	35
Prerequisites	35
Creating a service account	35
Configuring Application Impersonation on the service account	36
Creating a new Distribution Group	37
Configuring application impersonation	37
Configuring calendar processing on room resource mailboxes	38
Recommended configuration	38
Optional configuration	38
Checking calendar processing settings	39
Enabling OAuth authentication	41
Adding a One-Touch Join Exchange integration on Pexip Infinity	46
Configuring the Exchange integration	46
Signing in to the service account	48
Next steps	49
Configuring Google Workspace for One-Touch Join	50
Prerequisites	50
Creating a service account	50
Creating a room resource	54
Configuring the room resource	56
Sharing calendars externally	56
Sharing individual calendars with the service account	57
Auto-accepting invitations	58
Allowing users to book resources	59

Updating the per-user request quota	60
Requesting an increase to API limits	61
Adding a One-Touch Join Google Workspace integration on Pexip Infinity	63
Next steps	64
Configuring Pexip Infinity for One-Touch Join	65
Prerequisites	65
Adding a One-Touch Join profile	65
Hiding or changing the meeting subject	68
Adding One-Touch Join endpoint groups	68
Adding One-Touch Join endpoints	69
Adding endpoints individually	69
Adding OTJ endpoints in bulk	71
Adding One-Touch Join meeting processing rules	73
Testing the rule	75
Next steps	76
Configuring endpoints to support One-Touch Join	77
Prerequisites	77
Configuring Cisco OBTP endpoints for OTJ	77
Prerequisites	78
Creating a Webex Integration	78
Enabling a One-Touch Join profile to use Webex Cloud	78
Adding a Webex endpoint	79
Disabling the calendar	79
Enabling the client API	81
DNS records	81
Poly authentication	82
Deployments with a load balancer	82
Configuring Poly RealPresence Group series	82
Configuring Poly Trio series	83
Configuring Poly HDX series	86
Configuring Poly Studio X series and Poly G7500 series	88
Configuring Poly Debut series	89
One-Touch Join meeting types and transforms	91
Fallback alias matching	91
Supported meeting types	91
Regex meeting type	94
Examples	94
Custom meeting type	95
Examples	96
Deploying a dedicated One-Touch Join platform	98
Minimum hardware requirements	98
On-premises deployments	98

Cloud deployments	98
Minimum Pexip Infinity platform configuration	99
One-Touch Join configuration	99
Scheduling and joining meetings using One-Touch Join	100
Using One-Touch Join in meeting rooms	100
Using One-Touch Join with your personal endpoint	100
Viewing One-Touch Join status	101
Viewing One-Touch Join meetings	101
Viewing One-Touch Join endpoints	102
Configuring Google Workspace for domain user authorization	103
Prerequisites	103
Enabling authorization using OAuth	103
Creating a room resource	107
Configuring the room resource	110
Sharing individual calendars with the authorization user	110
Auto-accepting invitations	111
Allowing users to book resources	112
Updating the per-user request quota	113
Requesting an increase to API limits	114
Adding a One-Touch Join Google Workspace integration on Pexip Infinity	116
Configuring the Google Workspace integration	116
Authorizing calendar access	117
Next steps	119
Troubleshooting One-Touch Join	120

About One-Touch Join

Pexip Infinity's One-Touch Join (OTJ) feature integrates support for existing "click to join" videoconferencing endpoint workflows into your Pexip Infinity deployment. With One-Touch Join, when users schedule a meeting in Microsoft Outlook or Google Calendar and include in the meeting invitation a room that contains a [supported Cisco or Poly endpoint](#), the endpoint will display a Join button just before the meeting is scheduled to begin. Participants can then simply walk into the room and select the button, and the endpoint will automatically dial in to the meeting.

One-Touch Join is available as an optional licensed feature within the Pexip Infinity platform.

In most cases, One-Touch Join will be implemented as a feature within a wider Pexip Infinity deployment, and run on Conferencing Nodes alongside other Pexip Infinity services. However, you can also set up separate OTJ locations within your deployment that contain Conferencing Nodes used solely for One-Touch Join. A third option appropriate in some situations is to implement a separate Pexip Infinity deployment purely for One-Touch Join, for example if you are a [Pexip Service customer wishing to use One-Touch Join](#), or you are a large enterprise wishing to separate the resources used for your One-Touch Join deployment. For more information, see [Deploying a dedicated One-Touch Join platform](#).

Enabling One-Touch Join

All Conferencing Nodes are capable of running One-Touch Join, although the service will only come into active operation on a node when the location the node is in is associated with a One-Touch Join Endpoint Group.

Enabling the Pexip One-Touch Join service within your Pexip Infinity deployment involves the following steps, each described in separate topics:

1. Permitting the One-Touch Join service to access the calendars used for One-Touch Join. How this is achieved depends on which calendar/email service is used in your environment:
 - For **Google Workspace**, you create a service account and share OTJ calendars with the service account. This service account authenticates to Google Workspace with a private key using 2-legged OAuth. For full details, see [Configuring Google Workspace for One-Touch Join](#).
 - ❗ We also offer an alternative means for Google Workspace environments where the recommended method of using a service account is not desirable. This alternative method uses a domain user which authenticates to Google Workspace using 3-legged OAuth. For full details, see [Configuring Google Workspace for domain user authorization](#).
 - For **Exchange on premises**, you create a service account that uses application impersonation to read the OTJ calendars. This service account authenticates to Exchange using basic authentication. For full details, see [Configuring Exchange on-premises for One-Touch Join](#).
 - For **Office 365**, you create an application registration in Azure for OTJ, and grant the application permission to read OTJ calendars using Microsoft's Graph API. OTJ uses 2-legged OAuth to authenticate to Office 365 without the need of a service account. For full details, see [Configuring Office 365 using Graph for One-Touch Join](#).
 - ❗ We continue to support **existing** One-Touch Join deployments for Office 365 that used a service account with application impersonation to read OTJ calendars. This service account authenticated using OAuth and used the EWS API to access mailboxes. However, the EWS API is being [deprecated by Microsoft](#), so for **new** One-Touch Join deployments in Office 365 environments you should instead [use the Graph API](#) to provide access to room resource mailboxes. For information on managing these existing deployments, see [Configuring Office 365 using EWS for One-Touch Join](#).
2. [Configuring Pexip Infinity for One-Touch Join](#)
3. [Configuring endpoints to support One-Touch Join](#)
4. [Viewing One-Touch Join status](#)

For an overview of the process and general deployment and network considerations for One-Touch Join, see [One-Touch Join process and deployment overview](#).

For a guide for end users, see [Scheduling and joining meetings using One-Touch Join](#).

For help with troubleshooting your One-Touch Join deployment, see [Troubleshooting One-Touch Join](#).

Supported Google Workspace editions

Pexip One-Touch Join is supported in the following Google Workspace environments:

- Google Workspace Basic
- Google Workspace Business
- Google Workspace Enterprise

Supported Exchange environments

Pexip One-Touch Join is supported in the following Microsoft Exchange environments:

Exchange servers

- Office 365
- Exchange 2013 (with the latest updates)
- Exchange 2016 (with the latest updates)
- Exchange 2019 (with the latest updates)

Outlook clients

Meetings scheduled in all Outlook clients are supported. Note that different third-party Outlook add-ins for different Outlook versions may format the join details for some meeting types slightly differently.

Supported endpoints

Endpoints used for One-Touch Join **must not** also be registered to the calendaring service on other systems such as the cloud-based Webex Hybrid Calendar Service, or Cisco TMS XE.

Cisco OBTP

Pexip Infinity One-Touch Join is supported on Cisco VTC endpoints that support Cisco One Button to Push (OBTP) and are running TC, CE, or RoomOS software. This includes:

- Cisco Webex Room series (Room, Room Kit)
- Cisco C series (C20, C40, C60, C90)
- Cisco DX series (DX70, DX80)
- Cisco EX series (EX60, EX90)
- Cisco MX series (MX200, MX300, MX700, MX800)
- Cisco SX series (SX10, SX20, SX80)
- Webex Desk Series (Webex Desk, Webex Desk Pro, Webex Desk Mini)

There are two ways in which One-Touch Join can be implemented for these endpoints, depending on whether or not the endpoint is on the same network as the OTJ Conferencing Nodes.

- If the endpoint is on the same network as the OTJ Conferencing Node, the Conferencing Node will connect directly to the endpoint to provide it with the necessary meeting information. When setting up these endpoints in Pexip Infinity, you assign them an **Endpoint type** of *Cisco OBTP*.

For more information on how to configure these endpoints, see [Configuring Cisco OBTP endpoints for OTJ](#).

- If the endpoint is not on the same network as the OTJ Conferencing Node (for example if it is located in a home office) but is registered to Webex or Webex Edge for Devices, the Conferencing Node will connect to Webex Cloud, which will in turn connect to the endpoint to provide it with the necessary meeting information. When setting up these endpoints in Pexip Infinity, you assign them an **Endpoint type** of *Webex Cloud registered*.

For more information on how to configure these endpoints, see [Configuring Cisco Webex Cloud registered endpoints](#).

Poly OTD

Pexip Infinity One-Touch Join is supported on Poly VTC endpoints that support Poly One Touch Dial (OTD). This includes:

- Poly RealPresence Group series v5.0.0 or later
- Poly Trio series
- Poly HDX series (unless Pexip Infinity has been [deployed in a secure mode of operation](#) - for more information, see [Poly authentication](#)); must be running a software version that supports NTLMv2 for calendaring, e.g. 3.1.11 or later
- Poly Studio X series
- Poly G7500 series
- Poly Debut series

For information on how to configure these endpoints to support Pexip One-Touch Join, see [Configuring Poly OTD endpoints for OTJ](#).

Supported meeting types

This release of Pexip One-Touch Join can be used to join the following types of meetings:

- Pexip Infinity meetings (i.e. those scheduled using the [VMR Scheduling for Exchange](#) feature)
- Pexip Service meetings (i.e. those scheduled using the plugin available to [Pexip Service](#) users)
- Google Meet (for Google Workspace integrations only)
- Microsoft Teams
- Skype for Business
- Webex
- Zoom
- BlueJeans
- GoToMeeting

You can also create your own meeting processing rules for meeting types not listed above. For more information, see [One-Touch Join meeting types and transforms](#).

Supported number of endpoints and Conferencing Nodes

The One-Touch Join feature will support up to 4,000 room resource calendars and associated endpoints. This applies to One-Touch Join both when integrated with a Pexip Infinity deployment (i.e. when running on Conferencing Nodes alongside other Pexip Infinity services), and as a [deployment dedicated to One-Touch Join](#).

For **integrated One-Touch Join deployments** (i.e. where OTJ is being implemented as a feature within a wider Pexip Infinity deployment), a Pexip Infinity deployment with a single Conferencing Node per location should also support up to 170 OTJ room resource calendars and associated endpoints (although you may wish to include one or more additional Conferencing Nodes for redundancy). For large or busy deployments, you may need to add additional Conferencing Nodes per location to provide the additional capacity required when One-Touch Join is implemented — we recommend that you consult your Pexip authorized support representative for advice on your particular deployment.

These recommendations apply to Pexip Infinity deployments with one or two One-Touch Join Integrations. For deployments with multiple OTJ Integrations (for example, when implemented by service providers with multiple customers) we recommend a dedicated One-Touch Join deployment.

For **dedicated One-Touch Join deployments** of all sizes (i.e. up to the supported 4,000 room resource calendars and associated endpoints), we recommend one Conferencing Node for every 1,000 endpoints in a location (although you may wish to include one or more additional Conferencing Nodes for redundancy).

Pexip Infinity server requirements

In most cases you will be enabling One-Touch Join within a new or existing Pexip Infinity deployment, and the One-Touch Join service can be run alongside other Pexip Infinity services on each Conferencing Node. Enabling One-Touch Join within most Pexip Infinity deployments (up to 170 endpoints — see [Supported number of endpoints and Conferencing Nodes](#)) will not significantly increase the processing requirements of the Management Node or Conferencing Nodes, therefore our standard [server design guidelines](#) still apply. However, if your deployment is expected to be particularly large or busy, we recommend that you consult your Pexip authorized support representative for advice.

For dedicated One-Touch Join deployments, see [Minimum hardware requirements](#).

In both cases, we recommend that each Conferencing Node runs on a different VM host and uses different storage.

One-Touch Join process and deployment overview

This topic gives an overview of the process used by One-Touch Join to extract calendar information and provide it to endpoints, along with information on general deployment and network considerations.

Process overview

The general process from setting up One-Touch Join through to having the endpoint display a Join button at the start of a meeting is as follows:

Administrator configures OTJ

1. The administrator configures their [Google Workspace](#), [Exchange on-premises](#) or [Office 365](#) deployment to support Pexip Infinity One-Touch Join, and ensures that each physical meeting room that contains an endpoint to be used for One-Touch Join has an associated email address.
2. The administrator then [configures One-Touch Join on the Pexip Infinity Management Node](#). This configuration is automatically replicated to the One-Touch Join service that runs on each Conferencing Node in the Pexip Infinity deployment.
3. Finally, the administrator [configures their endpoints](#) to support One-Touch Join.

End user sends invitation

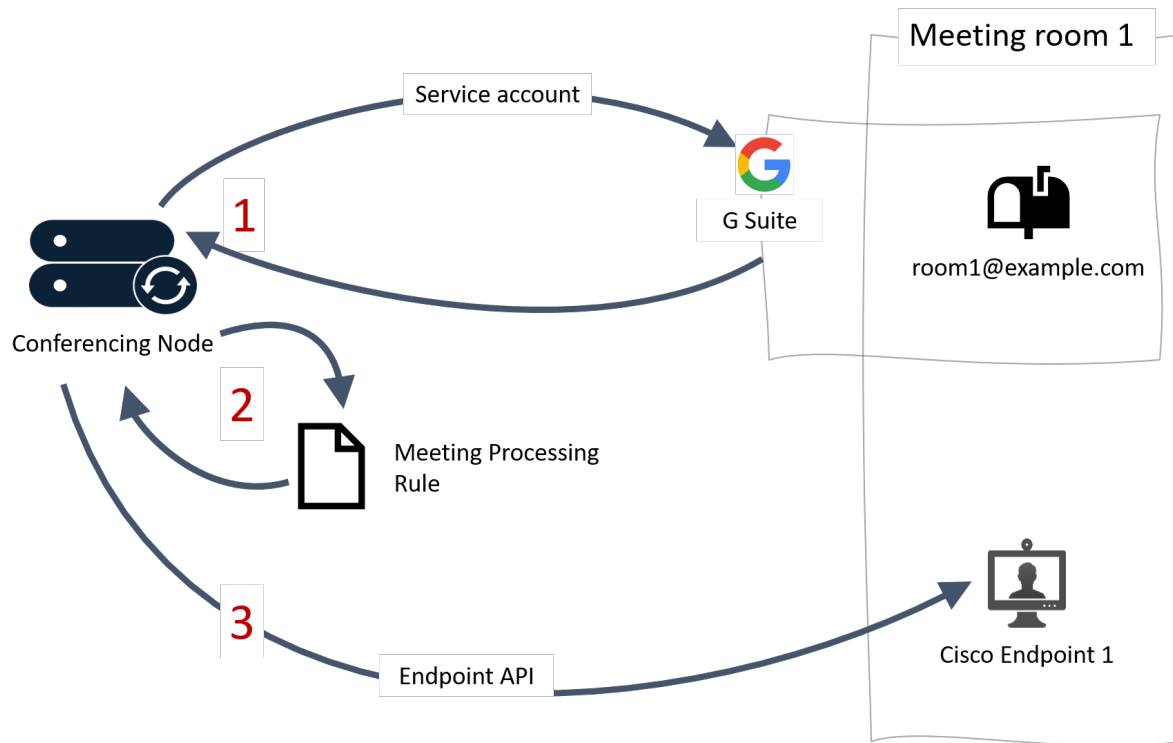
When an end user wants to [use a One-Touch Join room for a meeting](#), they create a meeting invitation in their usual way, using their usual client, ensuring that the room resource is added to the invitation.

- i** Generally, rooms are added to a meeting invitation as a room resource, but One-Touch Join will also work if the room resource's email address is included in the list of invitees, or as a location.

OTJ provides endpoint with meeting information

1. Each meeting room resource has one Conferencing Node which will be its primary node. Periodically, One-Touch Join on the Conferencing Node connects to Google Workspace or Microsoft Exchange and reads the calendars of each room resource for which it is the primary node. For each room resource, One-Touch Join finds all meetings to which the room has been invited. By default, it does this for all meetings with a scheduled start time from one day in the past up to seven days in the future, but this range is configurable.
2. One-Touch Join parses the meeting invitation (in accordance with the relevant [meeting processing rule](#)) to obtain information about the meeting, which it uses to generate the alias that the endpoint will dial in order to join the meeting.
3. One-Touch Join then provides the meeting information to the endpoint that is associated with the room resource:
 - for Cisco endpoints, One-Touch Join pushes the meeting information to the endpoint - either directly (for endpoints on the same network) or via Webex Cloud (for endpoints on a different network)
 - for Poly endpoints, the endpoint registers to the OTJ calendaring service on the Conferencing Node and periodically requests updated meeting information from the Conferencing Node.
 - i** More than one endpoint can be associated with a single room resource; in this case, all the endpoints will receive the same meeting information.
4. When the meeting is about to start, the endpoint will display a Join button; participants in the room simply click the button and the endpoint will dial in to the meeting.

The flow of information between the calendar/email service, One-Touch Join and the endpoint is shown in the following diagram (using Google Workspace and a Cisco endpoint as the example):



Frequency and limitations on calendar requests

The length of time taken for a meeting booked via Exchange or Google calendar to appear on the corresponding room endpoint depends on a number of factors, but is largely due to the number of endpoints in your deployment.

In general, for deployments of around 170 endpoints or fewer, the One-Touch Join service will poll room resource calendars with a maximum frequency of every 30 seconds. (It does not poll any more frequently than this to avoid impacting the performance of Conferencing Nodes.)

- **Cisco endpoints** will be updated after each poll if a meeting change is detected, and meetings are re-pushed to Cisco endpoints once per hour.
- **Poly endpoints** will generally connect to the Conferencing Node to get updates every minute, but this will depend on the Poly configuration.

As you add more endpoints, One-Touch Join will reduce the frequency of requests correspondingly. For a deployment of 4,000 endpoints (the maximum supported number), endpoints will be updated around every 12 minutes. This is because both Microsoft Exchange and Google limit the number of API requests that can be made to their calendar services in a 24-hour period. It is possible to change the 24-hour quota to increase the frequency of endpoint updates in larger deployments, but note that doing so may impact the performance of the Conferencing Nodes, so you may need to consider [deploying a dedicated One-Touch Join platform](#). We recommend you discuss larger deployments with your Pexip authorized support representative first.

- For Google Workspace deployments, you can change the 24-hour quota by [Requesting an increase to API limits](#) and then increasing the [Maximum Google Workspace API requests](#), but this is a paid-for service.
- For Exchange deployments, you can change the 24-hour quota by increasing the [Find Items Request Quota](#).

Locations, Conferencing Nodes and redundancy

Conferencing Nodes

All Conferencing Nodes in your deployment are capable of running One-Touch Join. However, the service will be in active operation on only those nodes that belong to a location that has been associated with a OTJ Endpoint Group (and when that Endpoint Group has been associated with an OTJ profile).

Within each such location, a maximum of five Conferencing Nodes will actively read room resource calendars and process meeting information. Responsibility for each room resource is spread across these nodes in order to balance the workload and provide redundancy. Should one node become unavailable (for example, if it is put into maintenance mode or loses connectivity), the other nodes will take over responsibility for its room resources.

However, if there are one or more Poly endpoints in the location, the One-Touch Join service on **all** nodes within the location will handle requests from Poly endpoints. Therefore [round-robin DNS records](#) are required for all nodes in a location that has Poly endpoints.

You can use existing system locations for One-Touch Join, in which case up to five Conferencing Nodes in that location will be actively operating One-Touch Join in addition to their core functions. Alternatively, you can set up system locations that will be used specifically for One-Touch Join. These can be in the same physical locations as your existing Conferencing Nodes, but their resources will be dedicated to One-Touch Join.

The concept of media [overflow locations](#) does not apply to One-Touch Join (overflow locations relate specifically to the handling of call media). Therefore if you want to provide redundancy, this can only be done by providing additional Conferencing Nodes **within** a given location. For the same reason, if you put **all** Conferencing Nodes in a One-Touch Join location into maintenance mode, then none of the endpoints in the associated Endpoint Groups will receive any updates.

Management Node

As with other Pexip Infinity services, the One-Touch Join service will continue to function if the Management Node goes offline, although you will not be able to make any changes to the configuration of the service during this time.

For deployments using OAuth, the Management Node periodically refreshes OAuth tokens on behalf of Conferencing Nodes, so eventually (after some weeks) these nodes may become unable to authenticate with Exchange / Google Workspace.

Network architecture, firewalls and web proxy

Conferencing Nodes

Each Conferencing Node used for One-Touch Join requires a persistent connection to one of Google Workspace, on-premises Microsoft Exchange server; Office 365; or the Microsoft Graph API (depending on the calendar service you are integrating with), either directly or via a web proxy*.

If you are using OAuth (i.e. you are using an OTJ Google Workspace Integration, an OTJ Graph Integration, or an OTJ Exchange integration with OAuth enabled), each Conferencing Node must be able to reach the OAuth token endpoint, either directly or via a web proxy*.

Each Conferencing Node must be able to access the Cisco One-Touch Join endpoints within its location (using the endpoints' APIs), either directly or via a web proxy*.

If you have Webex-registered endpoints, each Conferencing Node must be able to access the Webex OAuth token endpoint, and Webex cloud.

Poly endpoints must be able to connect directly to the Conferencing Nodes in their location.

* Web proxies are enabled on a system location basis. When enabled, all One-Touch Join-related outbound requests from Conferencing Nodes in that location will use the web proxy. It is possible to bypass use of the web proxy for connections to endpoints on the local network, or for EWS connections to the Exchange server; for further information, please contact your Pexip authorized support representative.

Management Node

As with all Pexip Infinity deployments, the Management Node must be able to contact each Conferencing Node.

In addition, if your One-Touch Join deployment is using OAuth (within an Exchange integration, a Google Workspace integration with domain user authorization, or where your deployment includes Webex-registered endpoints on a different network to your Conferencing Nodes), the Management Node will send requests to the OAuth token endpoint, both during the initial set up, and periodically thereafter in order to refresh the OAuth tokens. These requests will be sent either directly or via the web proxy (if one has been configured for the Management Node).

Port usage

The following table lists the ports/protocols required for communication between the components of Pexip One-Touch Join:

Source address	Source port	Destination address	Dest. port	Protocol
Management Node	55000–65535	Web proxy (if configured for the Management Node)	8080 †	TCP
Management Node	55000–65535	OAuth token endpoint (for Exchange integrations connecting to O365 using OAuth for the service account; or Google Workspace integrations; or Webex-registered endpoints) ◇ <ul style="list-style-type: none"> for Exchange/O365 service account authorization: login.microsoftonline.com for Google Workspace domain user authorization: oauth2.googleapis.com/token for Webex-registered endpoints: webexapis.com 	443 †	TCP (HTTPS)
Conferencing Node	55000–65535	Web proxy (if configured for the system location to which the Conferencing Node belongs)	8080 †	TCP
Conferencing Node	55000–65535	graph.microsoft.com (for O365 Graph Integrations) ◇	443 †	TCP (HTTPS)
Conferencing Node	55000–65535	Exchange on-premises or Office 365 (for Exchange Integrations or O365 EWS Integrations) ◇	443 †‡	TCP (HTTPS)
Conferencing Node	55000–65535	Exchange Server (only required if the O365 Autodiscover URL lookup has otherwise failed) ◇	80†	TCP (HTTP)
Conferencing Node	55000–65535	OAuth token endpoint (for Exchange Integrations connecting to O365, or O365 Graph Integrations, or Google Workspace integrations, or Webex-registered endpoints) ◇ <ul style="list-style-type: none"> for O365: login.microsoftonline.com for Google Workspace service account authorization: googleapis.com/oauth2/v4/token for Google Workspace domain user authorization: oauth2.googleapis.com/token for Webex-registered endpoints: webexapis.com 	443 †	TCP (HTTPS)
Conferencing Node	55000–65535	googleapis.com (for Google Workspace Integrations) ◇	443	TCP (HTTPS)
Conferencing Node	55000–65535	Cisco endpoint API ◇	80/443 †	TCP (HTTP/HTTPS)

Source address	Source port	Destination address	Dest. port	Protocol
Conferencing Node	55000–65535	Webex cloud: webexapis.com †	443	TCP (HTTPS)
Poly endpoint	<any>	Conferencing Node	443	TCP (HTTPS)

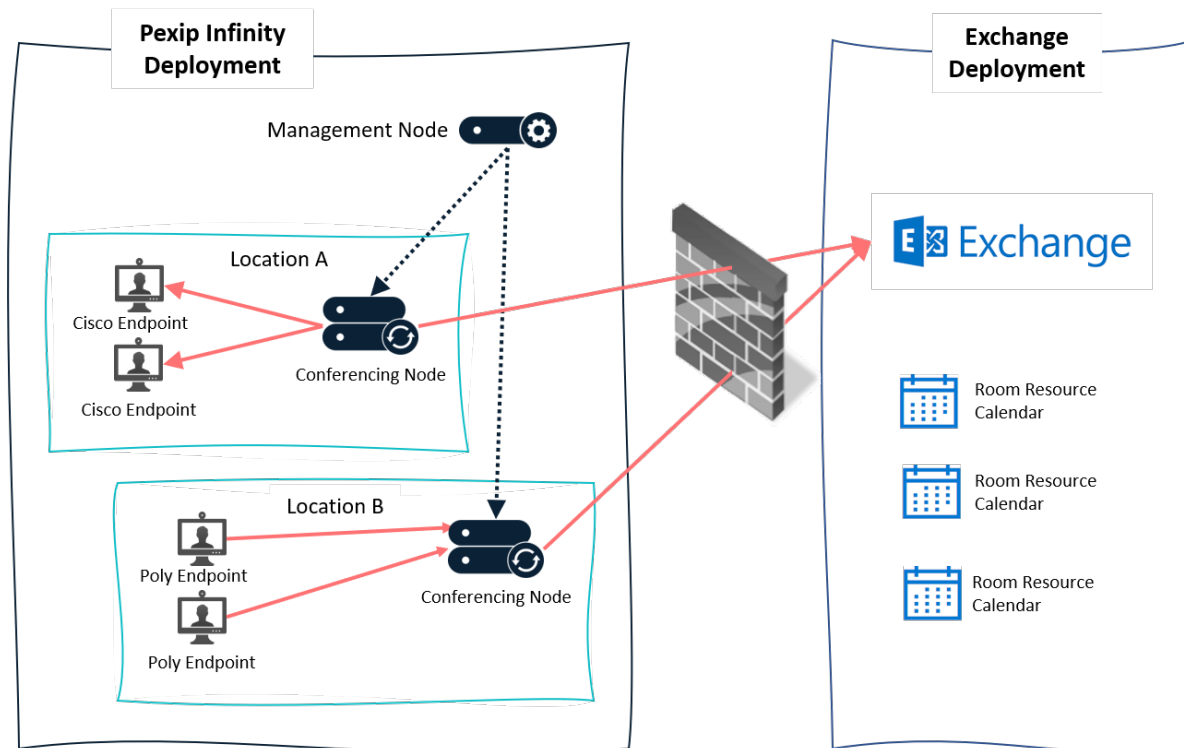
† Configurable by the administrator.

‡ Determined by Exchange.

‡ Does not apply if a [web proxy](#) has been configured.

Note also that the ephemeral port range (55000–65535) is subject to change.

The diagram below summarizes the connectivity required between the components of Pexip One-Touch Join, using Microsoft Exchange as an example.



Note in most cases, and particularly for a [dedicated One-Touch Join](#) deployment, all Conferencing Nodes should remain within the internal network, and not in the DMZ.

Permitting the service account to access calendars

Exchange integrations

For Exchange on-premises integrations, the One-Touch Join service account must be able to impersonate the calendar of each OTJ room resource (or a user's personal calendar, if you wish to [Use OTJ with personal endpoints and calendars](#)). This is achieved by adding the email address to a specific OTJ Distribution Group, and giving the service account application impersonation rights to that group. For instructions on how to do this, see [Configuring Application Impersonation on the service account](#).

- i** Existing Office 365 One-Touch Join deployments that were set up to use the EWS API also use application impersonation; see [Configuring Application Impersonation on the service account](#). However, the EWS API is being **deprecated by Microsoft**, so for **new** One-Touch Join deployments in Office 365 environments you should instead [use the Graph API](#) to provide access to room resource mailboxes.

The use of Exchange impersonation is common in business applications that work with mail, when a single account needs to access many accounts.

The following information from Microsoft provides further background on the use of impersonation in Exchange:

- <https://docs.microsoft.com/en-us/exchange/client-developer/exchange-web-services/impersonation-and-ews-in-exchange> for guidelines on when to use impersonation in your Exchange service applications.
- <https://blogs.msdn.microsoft.com/exchangedev/2009/06/15/exchange-impersonation-vs-delegate-access/> for information on the differences between impersonation and delegate access.

Google Workspace integrations

For Google Workspace integrations, the One-Touch Join service account (or the authentication user, if using 3-legged OAuth) must be able to access the calendar of each room resource. This is achieved by sharing the room resource's calendar (or the user's personal calendar, if you wish to [Use OTJ with personal endpoints and calendars](#)) with the service account. For instructions on how to do this, see [Sharing individual calendars with the service account](#).

Note that the Google calendar API limits the number of calendars that can be shared within a 24 hour period to 750 (for more information, see <https://support.google.com/a/answer/2905486?hl=en>). This means that if you have more than 750 room resources that you wish to use for One-Touch Join, they will need to be set up over a period of days.

Using One-Touch Join with personal endpoints and calendars



Some users in your enterprise may have their own personal endpoints on their desk or in their office, which they want to integrate with their personal calendars so that they can simply use the "Join" button to connect to any video meetings that appear in their calendar.

To achieve this, you use the user's own email address as the [room resource email](#) address when configuring One-Touch Join. You must also ensure that the One-Touch Join service can access the user's calendar. In Exchange environments this is achieved by adding the personal email address to the distribution group used for One-Touch Join; in Google Workspace environments the calendar must be shared with the service account.

Configuring Exchange on-premises for One-Touch Join

This topic describes how to implement Pexip Infinity's One-Touch Join feature in a Microsoft Exchange on-premises environment, by using a service account authenticated using basic authentication to enable the One-Touch Join service to access calendars used for OTJ.

The process involves the following steps, described in detail in the sections that follow:

1. [Creating a service account](#) for One-Touch Join. This service account will be used by One-Touch Join to read each room resource's calendar.
 -  This should be a different service account to that used for VMR Scheduling for Exchange, because the configuration will be different.
2. [Configuring Application Impersonation](#) on the service account.
 -  For more information and guidelines on the use of application impersonation in Exchange, see [Permitting the service account to access calendars](#).
3. [Enabling the authentication method](#) used for the service account — either NTLMv2 or basic authentication.
4. [Configuring calendar processing](#) within Exchange.
5. [Creating an associated Exchange integration](#) on Pexip Infinity.

Prerequisites

Before you begin, ensure that the following configuration is complete:

1. Ensure **each physical room** that will have a One-Touch Join endpoint in it has an associated **room resource with an email address**.
2. **Enable auto calendar processing for each room resource**, so that the room will automatically accept meeting requests if it is available, and automatically decline an invitation if it is already booked.
3. We recommend that if you are using Safe Links, you modify your Safe Links policy so that URLs are not rewritten in any meeting invitations sent to room resources used by One-Touch Join endpoints.
4. Ensure you have access to your Exchange Admin Center (EAC) web interface, and access to Exchange Management PowerShell.
5. If your Exchange server does not use a globally trusted certificate, you must upload a custom CA certificate.

Creating a service account

In this step, you create a service account that will be used to log in to Exchange to access the calendars of the room resources being used for One-Touch Join.

This service account should only be used with One-Touch Join. However, you can use the same Exchange service account for multiple One-Touch Join integrations.

You can create a new service account using **either** EAC or PowerShell, as follows:

EAC	PowerShell
<ol style="list-style-type: none"> Log in to your Exchange Admin Center as an administrator and go to recipients > mailboxes. Add a new mailbox for the service account by selecting the + icon and then User mailbox. Complete the fields as appropriate. Uncheck the Require password change on next logon box. <p>new user mailbox</p> <p>Alias: pexip</p> <p><input type="radio"/> Existing user</p> <p><input type="radio"/> New user</p> <p>First name: Pexip</p> <p>Initials: </p> <p>Last name: Exchange Service</p> <p>*Display name: Pexip Exchange Service</p> <p>*Name: Pexip Exchange Service</p> <p>Organizational unit: browse...</p> <p>*User logon name: pexip @ rd.pexip.com</p> <p>*New password: *****</p> <p>*Confirm password: *****</p> <p><input type="checkbox"/> Require password change on next logon</p> <p>More options...</p> <p>save cancel</p>	<p>i The first command lets the administrator type in a password for the service account as a secure string. This password variable is then used in the second command to create a mailbox for the service account. The third command ensures the password of the service account will not expire.</p> <pre>\$password = Read-Host "Enter password" -AsSecureString</pre> <pre>New-Mailbox -Name "<Account Name>" -UserPrincipalName "<UPN>" -Password \$password -Alias "<Account Alias>" -FirstName "<Account First Name>" -LastName "<Account Last Name>" -DisplayName "<Account Name>"</pre> <pre>Set-ADUser -Identity "<UPN>" -PasswordNeverExpires \$true</pre> <p>For example:</p> <pre>New-Mailbox -Name "Pexip OTJ Service Account" -UserPrincipalName pexip-otj-svc@example.com -Password \$password -Alias pexip-otj-svc -FirstName "Pexip OTJ" -LastName "Service Account" -DisplayName "Pexip OTJ Service Account"</pre> <pre>Set-ADUser -Identity pexip-otj-svc@example.com -PasswordNeverExpires \$true</pre>
<ol style="list-style-type: none"> Select Save. 	

Configuring Application Impersonation on the service account

In this step, you create a new Distribution Group, and add the rooms to be used for One-Touch Join to the group. You then use PowerShell commands to make it so that the service account will only be able to impersonate members of that Group.

Configuring Application Impersonation in this way means that if rooms are added or removed from the group, this automatically updates whether or not the service account can impersonate them.

Creating a new Distribution Group

- Log in to your Exchange Admin Center as an administrator and go to **recipients > groups**.
- Select the + icon and select **add a new Distribution Group**.
- Add the rooms you want to impersonate to the group.

Note that the service account should **not** be added as a member of this distribution group. Instead, this step allows the service account to impersonate any member of this distribution group (i.e. any of the room resources).

4. Make sure to uncheck the option to make the group owner a group member. Otherwise the service account will be able to impersonate your account.
5. Also make sure to lock the group down so people cannot accidentally add themselves as group members. Do this by selecting *Closed: Members can be added / removed only by the group owners*.

Configuring application impersonation

We recommend that you use combined PowerShell commands to configure application impersonation for the service account. This allows you to use variables, thus reducing possible copy and paste errors.

1. Configure the following variables with the values you actually want to use:
 - `otj_group_id`: the email of the distribution list whose members you want to be impersonated.
 - `otj_service_account`: the email of the service account you want to grant impersonation to.
 - `management_scope_to_create`: the name you want the newly created management scope to have.
 - `impersonation_role_name_to_create`: the name you want the newly created impersonation role to have.

For example:

```
$otj_group_id = "otjrooms@example.com"
$otj_service_account = "pexip-otj-svc@example.com"
$management_scope_to_create = "OTJ Management Scope"
$impersonation_role_name_to_create = "OTJ Impersonation"
```

2. Create the management scope:

```
$otj_group = Get-DistributionGroup -Identity $otj_group_id
$otj_group_dn = $otj_group.DistinguishedName
$restriction_filter = "MemberOfGroup -eq '$otj_group_dn'"
New-ManagementScope -Name $management_scope_to_create -RecipientRestrictionFilter $restriction_filter
```

Example output:

Name	ScopeRestrictionType	Exclusive	RecipientRoot	RecipientFilter
OTJ Management Scope	RecipientScope	False		MemberOfGroup -eq 'CN=OTJ Rooms2111430164340,OU...

3. Set up application impersonation using the previously created management scope:

```
New-ManagementRoleAssignment -Name $impersonation_role_name_to_create -Role ApplicationImpersonation -User $otj_service_account -CustomRecipientWriteScope $management_scope_to_create
```

Example output:

Name	Role	RoleAssigneeName	RoleAssigneeType	AssignmentMethod	EffectiveUserName
OTJ Impersonation	ApplicationImp...	pexip-otj-svc	User	Direct	

4. Verify that the above commands worked as expected. In the following command, replace `<resource_email>` with the email of the room resource mailbox you want to test. If it is a room which is a member of the distribution list, it should show the OTJ Impersonation in the returned roles. If it is anything else outside of the distribution list, it should not have the OTJ Impersonation listed, which means the OTJ service account does not have permission to impersonate that user.

```
Get-ManagementRoleAssignment -Role ApplicationImpersonation -WritableRecipient "<resource_email>" | Format-List Name, Role, RoleAssignee, CustomRecipientWriteScope
```

Expected output:

Name	: OTJ Impersonation
Role	: ApplicationImpersonation
RoleAssignee	: pexip-otj-svc

Enabling authentication

In this step you enable your Exchange on-premises deployment to support your chosen authentication method for the service account. One-Touch Join uses basic authentication by default, but you can [elect to use NTLMv2 authentication](#) instead.

For both forms of authentication, Pexip Infinity stores the credentials in encrypted form and all authentication is carried out over a secure TLS channel.

NTLMv2 authentication

In most on-premises Exchange deployments, NTLMv2 authentication is enabled by default. To confirm that it has been enabled in your environment:

1. Open Server Manager and select the server on which Exchange is installed.
2. From the top right options select **Tools > Local Security Policy**.
3. On the tree on the left, expand **Local Policies** then select **Security Options**.
4. Scroll down to **Network security: Restrict NTLM: Incoming NTLM traffic**.
5. Ensure this is either left to the default value of **Not Defined**, or set to **Allow All**.

Basic authentication

If you are using basic authentication with on-prem Exchange you need to ensure it is enabled for both Autodiscover and Exchange Web Services (EWS).

You can do this using **either** Windows Service Manager or PowerShell, as follows:

Windows Service Manager	PowerShell
<p>To enable basic authentication for Autodiscover:</p> <ol style="list-style-type: none"> 1. Go to the Windows server on which Exchange is installed and open the Service Manager. 2. Select the server on which Exchange is installed, and right-click to select Computer Management. 3. From the panel on the left, select Services and Applications > Internet Information Services (IIS) Manager. 4. Expand the options and select Sites > Default Web Site > Autodiscover. 5. Select the Authentication button in the main pane. 6. Find Basic Authentication in the list and ensure it is Enabled. (If not, right-click and select Enable.) 7. Select Save. 	<p>This command enables basic authentication for Autodiscover on a specific server:</p> <pre>Set-AutodiscoverVirtualDirectory -Identity "<server>\Autodiscover (Default Web Site)" -BasicAuthentication \$true</pre> <p>For example, if your server name is PEXCHANGE then:</p> <pre>Set-AutodiscoverVirtualDirectory -Identity "PEXCHANGE\Autodiscover (Default Web Site)" -BasicAuthentication \$true</pre>
<p>To enable basic authentication for EWS:</p> <ol style="list-style-type: none"> 1. Go to the Windows server on which Exchange is installed and open the Service Manager. 2. Select the server on which Exchange is installed, and right-click to select Computer Management. 3. From the panel on the left, select Services and Applications > Internet Information Services (IIS) Manager. 4. Expand the options and select Sites > Default Web Site > EWS. 5. Select the Authentication button in the main pane. 6. Find Basic Authentication in the list and ensure it is Enabled. (If not, right-click and select Enable.) 7. Select Save. 	<p>This command enables basic authentication for EWS on a specific server:</p> <pre>Set-WebServicesVirtualDirectory -Identity "<server>\EWS (Default Web Site)" -BasicAuthentication \$true</pre> <p>For example, if your server name is PEXCHANGE then:</p> <pre>Set-WebServicesVirtualDirectory -Identity "PEXCHANGE\EWS (Default Web Site)" -BasicAuthentication \$true</pre>

Configuring calendar processing on room resource mailboxes

In this step, you change the calendar processing settings for room resources from the default to those required to support One-Touch Join.

Recommended configuration

To take full advantage of the functionality offered by One-Touch Join, we recommend that, for One-Touch Join room resources, you change the following calendar processing options from the default:

1. The meeting invite body is deleted by default. If you want One-Touch Join to parse meeting details from the body then you must set the `DeleteComments` property to *False*. If you leave this set to *True*, only those rules that process information in the calendar headers can be used (because the body will be deleted).
2. When a meeting invite is received by a resource mailbox, by default the meeting subject is deleted and is replaced with the name of the organizer (for more information, see <https://support.microsoft.com/en-gb/help/2842288/resource-mailbox-s-calendar-shows-the-organizer-s-name-instead-of-the>).
Because One-Touch Join accesses the meeting invites through the resource mailboxes, this default behavior means it won't have access to the original subject. You can choose to leave the default behavior for privacy reasons, or you can modify the calendar processing options for each mailbox so that the meeting subject is available and thus can be displayed on the meeting room endpoints.
3. The private flag is cleared by default. If you want meetings that are marked as private by the organizer to remain marked as private in the room mailbox, you must set the `RemovePrivateProperty` flag to *False*.
4. Room resources created using PowerShell commands may by default have `AutomateProcessing` set to *AutoUpdate*. In these cases it should be changed to *AutoAccept*.

PowerShell command

To modify the calendar processing on a room from the default settings to those we recommend for One-Touch Join, [connect to Exchange Online PowerShell](#) and use the following PowerShell command (replacing `<resource_email>` with the address of the room resource whose processing you want to change):

```
Set-CalendarProcessing -Identity <resource_email> -DeleteComments $False -DeleteSubject $False -AddOrganizerToSubject $False -  
RemovePrivateProperty $False -AutomateProcessing "AutoAccept"
```

Optional configuration

Hiding invitation details from other users

In order for One-Touch Join to function fully, the service account must be able to access the body of the invitation (which is why we recommend that you set the `DeleteComments` property to *False*). However, this means that all other users in your deployment with access to the room resource calendar may also be able to view the body of the invitation (depending on your deployment's other policies). If you want to prevent this, you can use the following PowerShell command to restrict what users can see by default, without restricting what the service account can access.

In the following command, replace `resource_name` with the name of the room resource, and replace `role` with one of the following roles:

- **AvailabilityOnly**: users can view the room's availability, but nothing else.
- **LimitedDetails**: users can view the room's availability and the meeting subject and location, but not the body of the invitation.

```
Set-MailboxFolderPermission "resource_name:\Calendar" -User Default -AccessRights role
```

Allowing forwarding of external invitations

Below is some recommended configuration to enable external invitations to be forwarded to your internal OTJ room resources so that the meetings can be joined from those endpoints. In all cases, we recommend that you consult your Exchange administrator to determine what is appropriate in your environment.

- If you want to enable users to forward invitations from other organizations to your OTJ room resources, you must set the **ProcessExternalMeetingMessages** flag to **True**. Note that this will allow any users external to your organization to invite the resource directly. To prevent this, you can use an Exchange transport rule similar to the example shown below so that only users internal to your organization can forward external invitations to OTJ meeting rooms.

Block external invites to OTJ rooms, allow internal forwarding

Conditions Settings

Name *

Block external invites to OTJ rooms, allow internal forwarding

Apply this rule if *

The recipient is a member of this group +

The recipient is a member of 'otj.rooms@pexample.com'

And

The message properties include the message type

The message type is 'Calendaring'

And

The sender is external/internal

The sender is located 'NothOrganization'

Do the following *

Block the message reject the message and i... +

Reject the message with the explanation
'This room does not accept bookings from an external account.'

And

Block the message reject the message with t... +

Reject the message with the enhanced status code '5.7.1'

Except if

The message headers... includes any of these wo... +

'Sender' message header includes 'pexample.com'

Save Cancel

- If your Microsoft Exchange environment uses a security application (such as Office 365 ATP, or Mimecast) to re-write URLs, this may prevent OTJ from being used to join external Microsoft Teams meetings (for example, when a user inside your organization forwards an external Microsoft Teams meeting invitation to an OTJ room resource in order to join the meeting from that endpoint). To enable users to join these meetings using OTJ, you must ensure that the security application's URL re-write rules include an exception for any URL starting with the domain `https:\\teams.microsoft.com\`

Checking calendar processing settings

The following PowerShell command can be used to check calendar processing settings on all of the rooms in the Distribution Group that was created for One-Touch Join.

We recommend copying and saving this as a file and running it from within PowerShell.

Before running, ensure that you edit `$otj_group_id = "otjrooms@example.com"` to use the email of the Distribution Group used in your own deployment.

```
$deleted_subjects = @()
```

```

$organizer_added = @()
$deleted_bodies = @()
$private_flag_reset = @()
$not_auto_accept = @()
$process_external = @()
$otj_group_id = "otjrooms@example.com"

Get-DistributionGroupMember -Identity $otj_group_id -ResultSize Unlimited | ForEach-Object {
    Write-Host "Checking room '$($_.name)'"
    $processing = Get-CalendarProcessing -Identity $_.name
    $pass = $true
    if ($processing.DeleteSubject) {
        Write-Host "WARNING: The room '$($_.name)' is deleting the meeting subject" -ForegroundColor Red
        $deleted_subjects += $_.name
        $pass = $false
    }
    if ($processing.AddOrganizerToSubject) {
        Write-Host "WARNING: The room '$($_.name)' is adding the organizer to the meeting subject" -ForegroundColor Red
        $organizer_added += $_.name
        $pass = $false
    }
    if ($processing.DeleteComments) {
        Write-Host "WARNING: The room '$($_.name)' is deleting the meeting body" -ForegroundColor Red
        $deleted_bodies += $_.name
        $pass = $false
    }
    if ($processing.RemovePrivateProperty) {
        Write-Host "WARNING: The room '$($_.name)' is clearing the private flag on meetings" -ForegroundColor Red
        $private_flag_reset += $_.name
        $pass = $false
    }
    if ($processing.AutomateProcessing -ne "AutoAccept") {
        Write-Host "WARNING: The room '$($_.name)' is not configured to Auto Accept. Processing='$($processing.AutomateProcessing)'" -
        ForegroundColor Red
        $not_auto_accept += $_.name
        $pass = $false
    }
    # Optional permission for allowing the external invites:
    if ($processing.ProcessExternalMeetingMessages) {
        Write-Host "The room '$($_.name)' is configured to process external (forwarded) meetings"
        $process_external += $_.name
    }
    if ($pass) {
        Write-Host "INFO: All checks passed for room '$($_.name)'" -ForegroundColor Green
    }
}

Write-Host "Summary:"
Write-Host "There are $($deleted_subjects.count) rooms deleting the meeting subject"
if ($deleted_subjects) {
    Write-Host $deleted_subjects -Separator ", "
    Write-Host ""
}
Write-Host "There are $($organizer_added.count) rooms adding the organizer to the meeting subject"
if ($organizer_added) {
    Write-Host $organizer_added -Separator ", "
    Write-Host ""
}
Write-Host "There are $($deleted_bodies.count) rooms deleting the meeting body"
if ($deleted_bodies) {
    Write-Host $deleted_bodies -Separator ", "
    Write-Host ""
}
Write-Host "There are $($private_flag_reset.count) rooms clearing the private flag on meetings"
if ($private_flag_reset) {
    Write-Host $private_flag_reset -Separator ", "
    Write-Host ""
}
Write-Host "There are $($not_auto_accept.count) rooms not configured to Auto Accept"
if ($not_auto_accept) {
    Write-Host $not_auto_accept -Separator ", "
    Write-Host ""
}

```

```



}
Write-Host "There are $($process_external.count) rooms configured to process external (forwarded) meetings"
if ($process_external) {
    Write-Host $process_external -Separator ", "
    Write-Host ""
}

```

Adding a One-Touch Join Exchange integration on Pexip Infinity

In this step you log in to the Pexip Infinity Administrator interface and add details of the Exchange deployment you are integrating with, including details of the service account username and password (based on the configuration you have just set up in Exchange).

From the Pexip Infinity Administrator interface, go to **One-Touch Join > OTJ Exchange Integrations**.

Option	Description
Name	The name of this One-Touch Join Exchange integration.
Description	An optional description of this One-Touch Join Exchange integration.
Service account username	<p>The username of the service account to be used by the One-Touch Join Exchange integration.</p> <p>If you are using NTLMv2, this must be in the format name@domain.</p> <p>Otherwise, the format may be either domain\name or name@domain, depending on your domain.</p>
Enable OAuth	Leave this option disabled. (OAuth 2.0 is supported for Exchange in Office 365 only.)
Enable NTLM	Enable this option to authenticate the service account using NTLMv2. (This option is only supported for Exchange on-premises.) Leave this option disabled to authenticate the service account using basic authentication.
Service account password	<p>(Available if OAuth has not been enabled)</p> <p>The password of the service account to be used by the One-Touch Join Exchange integration.</p>
Advanced options	
Find Items Request Quota	<p>The number of Find Item requests that can be made by OTJ to your Exchange Server in a 24-hour period.</p> <p>The default of 1,000,000 should be sufficient for most deployments — for more information, see Frequency and limitations on calendar requests.</p> <p>We do not recommend increasing this quota unless you have deployed a dedicated One-Touch Join platform, because it will impact the performance of the Conferencing Nodes.</p>
OTJ Exchange Autodiscover URLs	
<p> This section is optional and will generally only be required if the Autodiscover URLs in your deployment do not use a standard location.</p>	
Name	The name of this Exchange Autodiscover URL.
Description	An optional description of this Exchange Autodiscover URL.
Autodiscover URL	<p>The URL used to connect to the Autodiscover service on the Exchange deployment.</p> <p>If you are using Office 365, you may need to enter your autodiscover URL manually, particularly if you are using a hybrid Exchange deployment. If your OTJ room resources and service account are hosted on O365, then you should enter https://autodiscover-s.outlook.com/autodiscover/autodiscover.svc as the Autodiscover URL.</p> <p> The URL must end in .svc; URLs ending in .xml are not supported.</p>

Next steps

You must now configure the remainder of the One-Touch Join components on Pexip Infinity, as described in [Configuring Pexip Infinity for One-Touch Join](#).

Configuring Office 365 using Graph for One-Touch Join

This topic describes how to implement Pexip Infinity's One-Touch Join feature in a Microsoft Office 365 environment, by configuring Microsoft Azure and the Exchange Graph API to enable the One-Touch Join service to access calendars used for OTJ.

- i** We continue to support **existing** One-Touch Join deployments for Office 365 that used a service account with application impersonation to read OTJ calendars. This service account authenticated using OAuth and used the EWS API to access mailboxes. However, the EWS API is being [deprecated by Microsoft](#), so for **new** One-Touch Join deployments in Office 365 environments you should instead [use the Graph API](#) to provide access to room resource mailboxes.

The process involves the following steps, described in detail in the sections that follow:

1. [Creating and configuring a new App registration in Azure.](#)
2. [Restricting the scope of the App registration.](#)
3. [Configuring calendar processing](#) within Exchange.
4. [Creating an associated Exchange integration](#) on Pexip Infinity.

Prerequisites

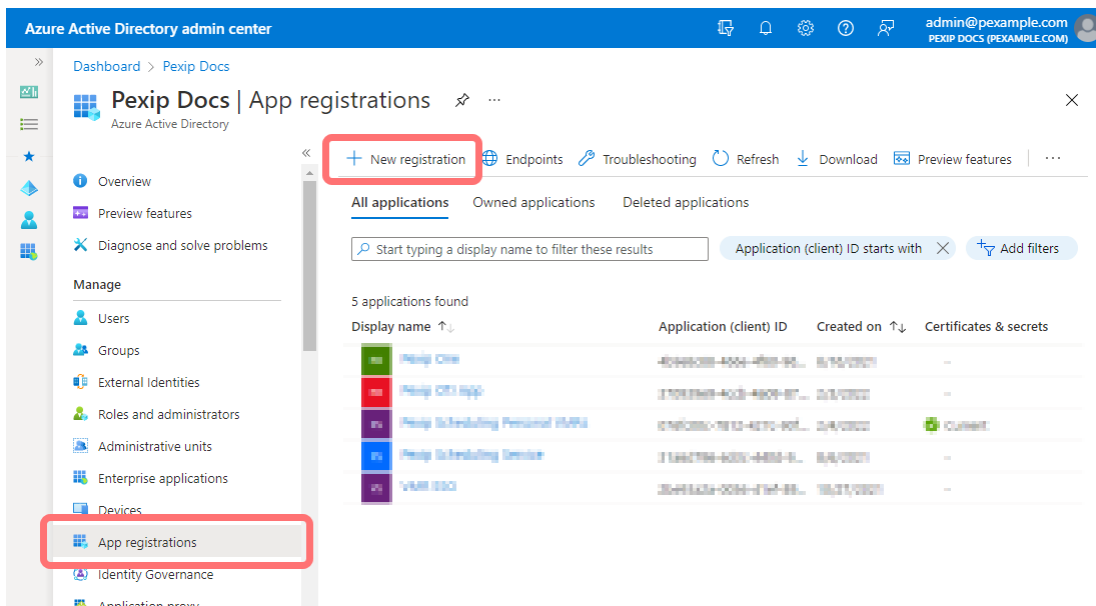
Before you begin, ensure that the following configuration is complete:

1. Ensure **each physical room** that will have a One-Touch Join endpoint in it has an associated **room resource with an email address**.
2. **Enable auto calendar processing for each room resource**, so that the room will automatically accept meeting requests if it is available, and automatically decline an invitation if it is already booked.
3. We recommend that if you are using Safe Links, you modify your Safe Links policy so that URLs are not rewritten in any meeting invitations sent to room resources used by One-Touch Join endpoints.
4. Ensure you have access to the Azure portal, using an account that can grant admin consent.
5. Ensure you have admin access to your Office 365 web interface, and access to the Microsoft Exchange Online and Azure Active Directory Modules for Windows PowerShell. (If you are connecting from your Windows PC for the first time, you may need to install these modules. See these Microsoft articles about connecting to [Exchange online](#) and [Microsoft 365](#) with PowerShell for more information.)

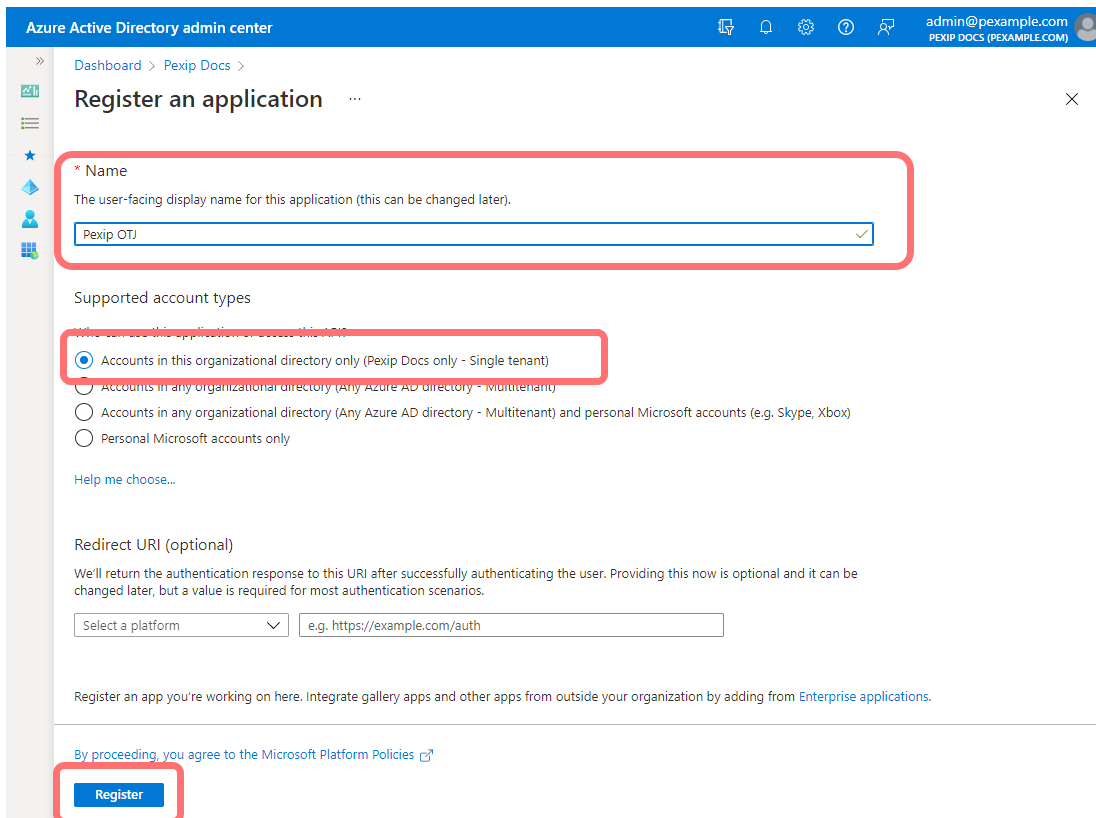
Creating and configuring a new App registration in Azure

In this step, you create an App registration in Azure for the OTJ service, and grant it permission to read calendars. (In a [subsequent step](#) you will restrict the app to read OTJ calendars only.)

1. Log into the Azure portal at aad.portal.azure.com as an admin user.
2. From the main panel on the left, select **Azure Active Directory**.
3. Select **App Registrations** and then **New registration**:



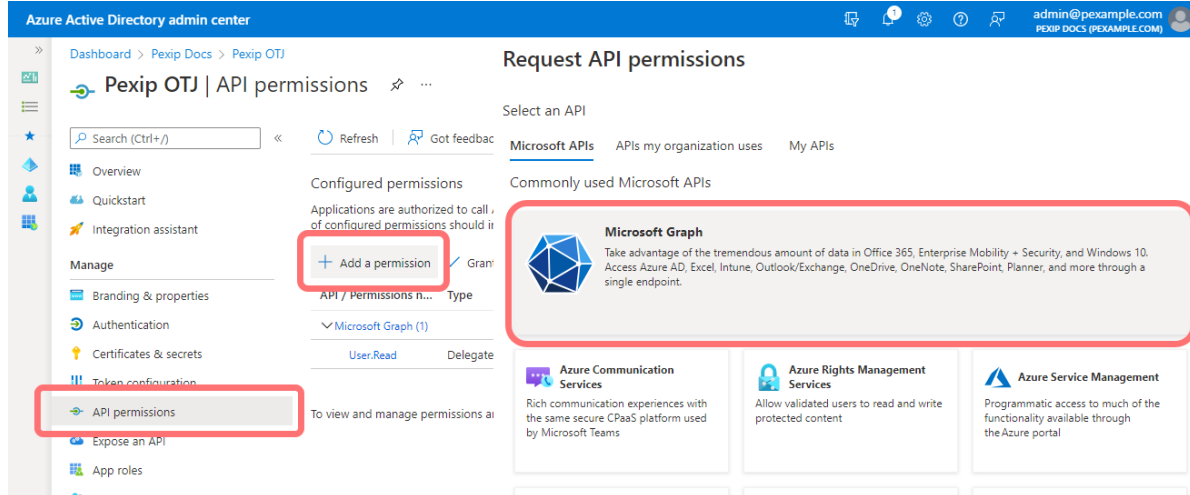
4. In the **Register an application** panel, enter the following options:
- Name:** this can be anything you wish. In our example we have used *Pexip OTJ*.
 - Supported account types:** select the option most appropriate for your environment. In most cases, the default *Accounts in this organizational directory only* can be used.
 - Redirect URI:** leave this blank.



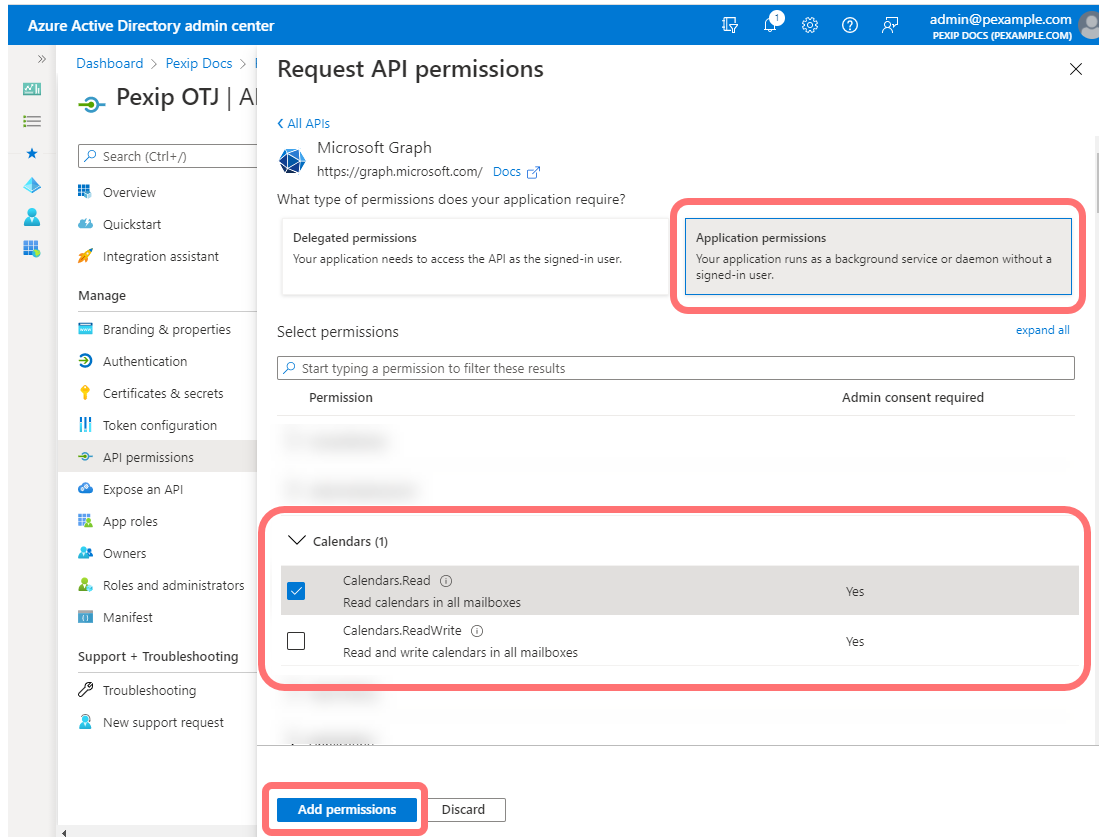
5. Select **Register**.

You can now configure your application.

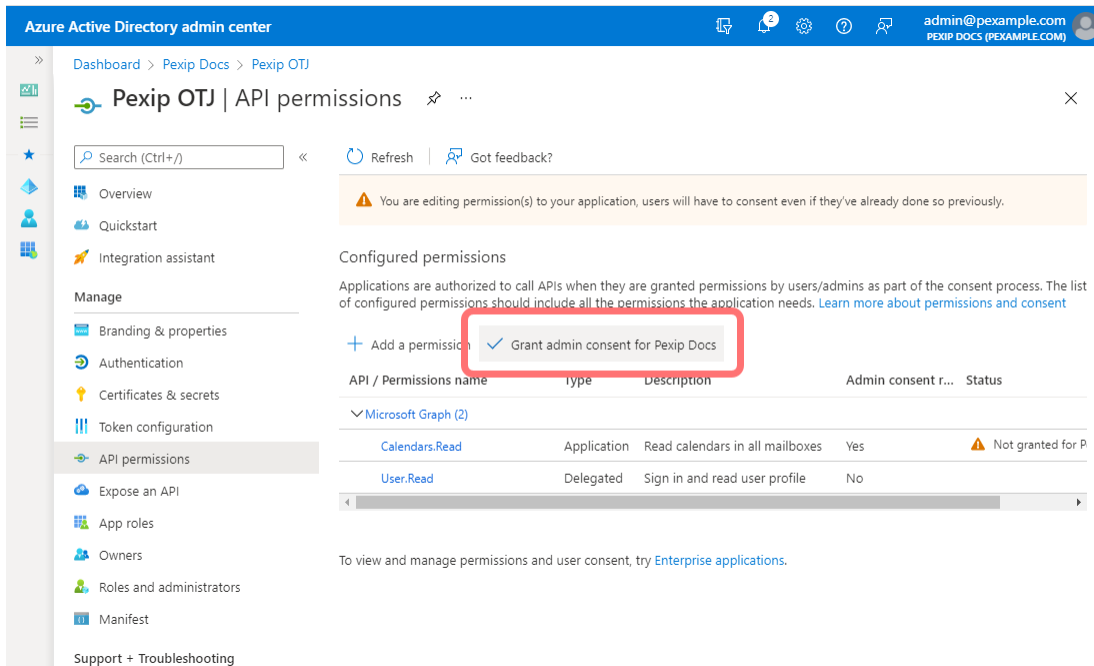
- From the panel on the left, select **API permissions** and then **Add a permission**.
- Select **Microsoft Graph**:



- Select **Application Permissions**. Scroll down to **Calendars**, expand it, and select **Calendars.Read**. Then select **Add Permissions**:

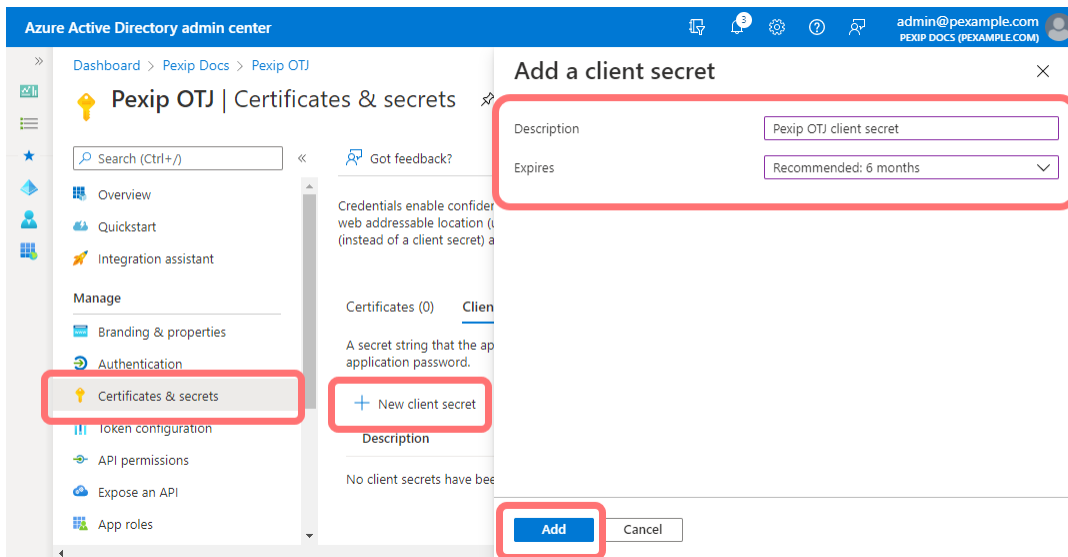


- Select **Grant admin consent for <your tenant>**:

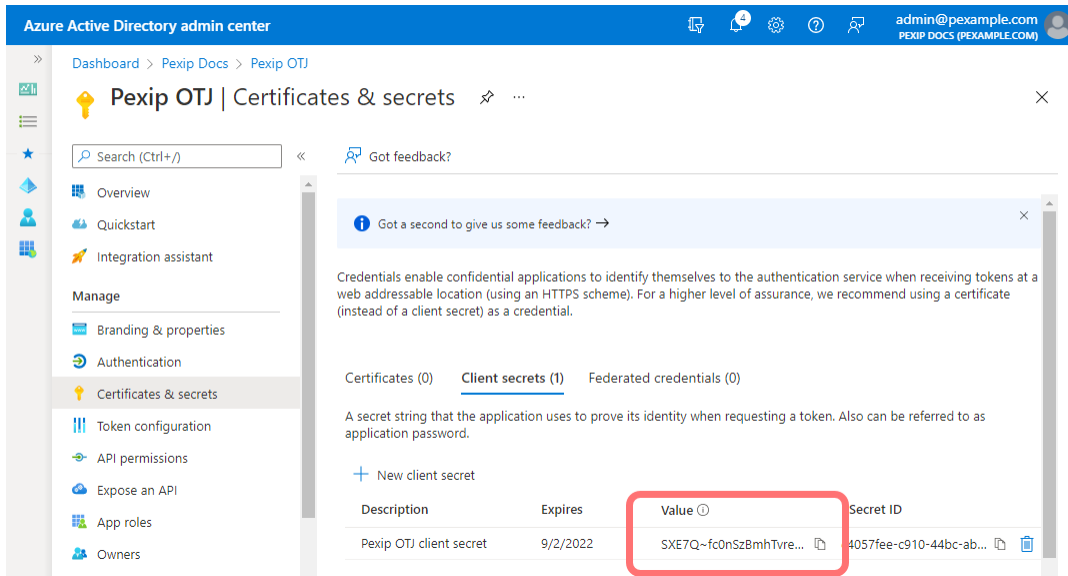


Next you need to obtain the client secret.

- From the panel on the left, select **Certificates & secrets** and then **New client secret**.
- Enter a **Description**. Under **Expires** select a duration in accordance with your organization's security policies, and select **Add**:

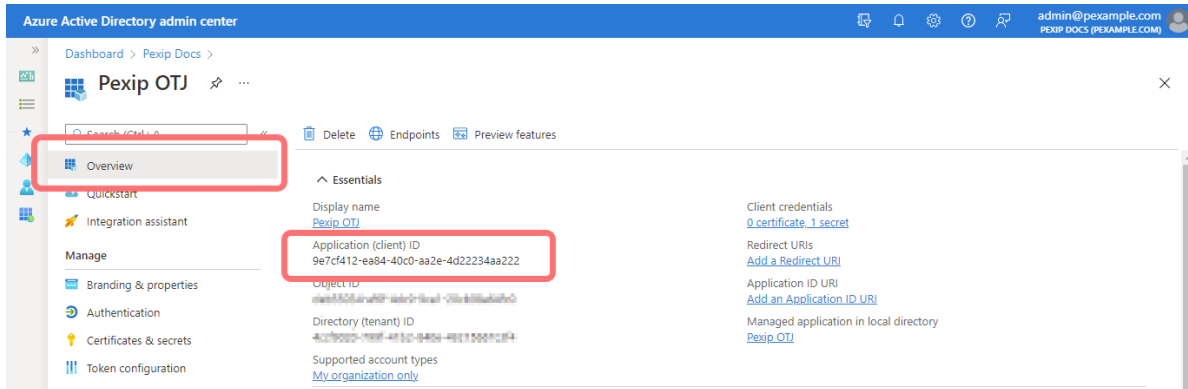


- The new client secret will appear in the list at the bottom of the page. You must copy the **Value** now, before you navigate away from the page:



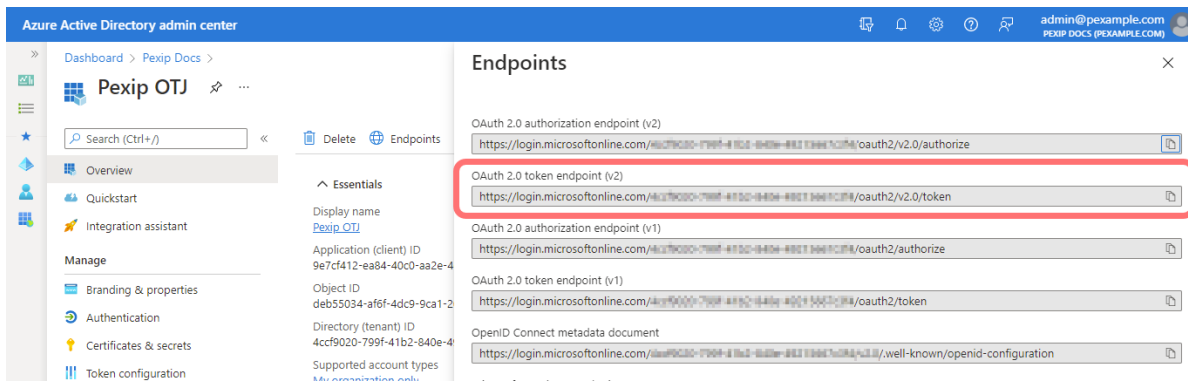
i You must enter this as the Client secret when adding an O365 Graph integration on Pexip Infinity.

13. Go to the overview page for the App registration you have just created and copy the **Application (client) ID**:



i You must enter this as the Client ID when adding an O365 Graph integration on Pexip Infinity.

14. Select the **Endpoints** tab and copy the **OAuth 2.0 token endpoint (v2)** value:



i You must enter this as the OAuth 2.0 token endpoint URL when adding an O365 Graph integration on Pexip Infinity.

Restricting the scope of the App registration

In this step, you create a group for the room resources to be used for One-Touch Join, and then restrict the App to only read these calendars.

Creating a Distribution Group

1. Go to admin.microsoft.com and log in as the administrator.
2. From the menu on the left hand side, select **Active teams & groups** and then **Add a group**.
3. For the **Group Type**, select **Mail-enabled security**. Select **Next**.
4. Enter a **Name** and **Description**. Select **Add**.
5. Enter a **Group email address**. Leave the **Communication** checkbox clear. Select **Next**.
6. Select **Create Group**.
7. Navigate back to **Active teams & groups**, select the **Mail-enabled security** tab, and then select the group you have just created. From the panel on the right, select the **Members** tab and then **View all and manage members**.
8. Add as members of the group the resources to be used for One-Touch Join. These will be the only calendars that the OTJ App will be able to read.

Restricting access

Open up a remote PowerShell connection to Office 365 and import an Exchange session. For example see <https://docs.microsoft.com/en-us/powershell/exchange/connect-to-exchange-online-powershell?view=exchange-ps>

Run the following command, using the following values:

- **AppId:** the **Application (client) ID** that was generated by Azure when you created the OTJ Graph API application.
- **PolicyScopeGroupId:** the email of the [distribution group](#) containing the One-Touch Join resources.
- **Description:** a description of the access policy.

For example:

```
New-ApplicationAccessPolicy -AppId e7e4dbfc-046f-4074-9b3b-2ae8f144f59b -PolicyScopeGroupId otjrooms@pexample.com -AccessRight RestrictAccess -Description "Restrict this app to members of distribution group otjrooms."
```

Configuring calendar processing on room resource mailboxes

In this step, you change the calendar processing settings for room resources from the default to those required to support One-Touch Join.

Recommended configuration

To take full advantage of the functionality offered by One-Touch Join, we recommend that, for One-Touch Join room resources, you change the following calendar processing options from the default:

1. The meeting invite body is deleted by default. If you want One-Touch Join to parse meeting details from the body then you must set the **DeleteComments** property to **False**. If you leave this set to **True**, only those rules that process information in the calendar headers can be used (because the body will be deleted).
2. When a meeting invite is received by a resource mailbox, by default the meeting subject is deleted and is replaced with the name of the organizer (for more information, see <https://support.microsoft.com/en-gb/help/2842288/resource-mailbox-s-calendar-shows-the-organizer-s-name-instead-of-the>).
Because One-Touch Join accesses the meeting invites through the resource mailboxes, this default behavior means it won't have access to the original subject. You can choose to leave the default behavior for privacy reasons, or you can modify the calendar processing options for each mailbox so that the meeting subject is available and thus can be displayed on the meeting room endpoints.
3. The private flag is cleared by default. If you want meetings that are marked as private by the organizer to remain marked as private in the room mailbox, you must set the **RemovePrivateProperty** flag to **False**.

- Room resources created using PowerShell commands may by default have **AutomateProcessing** set to **AutoUpdate**. In these cases it should be changed to **AutoAccept**.

PowerShell command

To modify the calendar processing on a room from the default settings to those we recommend for One-Touch Join, [connect to Exchange Online PowerShell](#) and use the following PowerShell command (replacing **<resource_email>** with the address of the room resource whose processing you want to change):

```
Set-CalendarProcessing -Identity <resource_email> -DeleteComments $False -DeleteSubject $False -AddOrganizerToSubject $False -RemovePrivateProperty $False -AutomateProcessing "AutoAccept"
```

Optional configuration

Hiding invitation details from other users

In order for One-Touch Join to function fully, the service account must be able to access the body of the invitation (which is why we recommend that you set the **DeleteComments** property to **False**). However, this means that all other users in your deployment with access to the room resource calendar may also be able to view the body of the invitation (depending on your deployment's other policies). If you want to prevent this, you can use the following PowerShell command to restrict what users can see by default, without restricting what the service account can access.

In the following command, replace **resource_name** with the name of the room resource, and replace **role** with one of the following roles:

- AvailabilityOnly**: users can view the room's availability, but nothing else.
- LimitedDetails**: users can view the room's availability and the meeting subject and location, but not the body of the invitation.

```
Set-MailboxFolderPermission "resource_name:\Calendar" -User Default -AccessRights role
```

Allowing forwarding of external invitations

Below is some recommended configuration to enable external invitations to be forwarded to your internal OTJ room resources so that the meetings can be joined from those endpoints. In all cases, we recommend that you consult your Exchange administrator to determine what is appropriate in your environment.

- If you want to enable users to forward invitations from other organizations to your OTJ room resources, you must set the **ProcessExternalMeetingMessages** flag to **True**. Note that this will allow any users external to your organization to invite the resource directly. To prevent this, you can use an Exchange transport rule similar to the example shown below so that only users internal to your organization can forward external invitations to OTJ meeting rooms.

Block external invites to OTJ rooms, allow internal forwarding

Conditions Settings

Name *

Block external invites to OTJ rooms, allow internal forwarding

Apply this rule if *

The recipient is a member of this group +

The recipient is a member of 'otj.rooms@pexample.com'

And

The message properties include the message type

The message type is 'Calendar'

And

The sender is external/internal

The sender is located 'NothOrganization'

Do the following *

Block the message reject the message and i... +

Reject the message with the explanation
'This room does not accept bookings from an external account.'

And

Block the message reject the message with t...

Reject the message with the enhanced status code '5.7.1'

Except if

The message headers... includes any of these wo... +

'Sender' message header includes 'pexample.com'

Save

Cancel

- If your Microsoft Exchange environment uses a security application (such as Office 365 ATP, or Mimecast) to re-write URLs, this may prevent OTJ from being used to join external Microsoft Teams meetings (for example, when a user inside your organization forwards an external Microsoft Teams meeting invitation to an OTJ room resource in order to join the meeting from that endpoint). To enable users to join these meetings using OTJ, you must ensure that the security application's URL re-write rules include an exception for any URL starting with the domain `https://teams.microsoft.com\`

Checking calendar processing settings

The following PowerShell command can be used to check calendar processing settings on all of the rooms in the Distribution Group that was created for One-Touch Join.

We recommend copying and saving this as a file and running it from within PowerShell.

Before running, ensure that you edit `$otj_group_id = "otjrooms@example.com"` to use the email of the Distribution Group used in your own deployment.

```
$deleted_subjects = @()
$organizer_added = @()
$deleted_bodies = @()
$private_flag_reset = @()
$not_auto_accept = @()
```



```

$process_external = @()
$otj_group_id = "otjrooms@example.com"

Get-DistributionGroupMember -Identity $otj_group_id -ResultSize Unlimited | ForEach-Object {
    Write-Host "Checking room '$($_.name)'"
    $processing = Get-CalendarProcessing -Identity $_.name
    $pass = $true
    if ($processing.DeleteSubject) {
        Write-Host "WARNING: The room '$($_.name)' is deleting the meeting subject" -ForegroundColor Red
        $deleted_subjects += $_.name
        $pass = $false
    }
    if ($processing.AddOrganizerToSubject) {
        Write-Host "WARNING: The room '$($_.name)' is adding the organizer to the meeting subject" -ForegroundColor Red
        $organizer_added += $_.name
        $pass = $false
    }
    if ($processing.DeleteComments) {
        Write-Host "WARNING: The room '$($_.name)' is deleting the meeting body" -ForegroundColor Red
        $deleted_bodies += $_.name
        $pass = $false
    }
    if ($processing.RemovePrivateProperty) {
        Write-Host "WARNING: The room '$($_.name)' is clearing the private flag on meetings" -ForegroundColor Red
        $private_flag_reset += $_.name
        $pass = $false
    }
    if ($processing.AutomateProcessing -ne "AutoAccept") {
        Write-Host "WARNING: The room '$($_.name)' is not configured to Auto Accept. Processing='$($processing.AutomateProcessing)'" -
        ForegroundColor Red
        $not_auto_accept += $_.name
        $pass = $false
    }
    # Optional permission for allowing the external invites:
    if ($processing.ProcessExternalMeetingMessages) {
        Write-Host "The room '$($_.name)' is configured to process external (forwarded) meetings"
        $process_external += $_.name
    }
    if ($pass) {
        Write-Host "INFO: All checks passed for room '$($_.name)'" -ForegroundColor Green
    }
}

Write-Host "Summary:"
Write-Host "There are $($deleted_subjects.count) rooms deleting the meeting subject"
    if ($deleted_subjects) {
        Write-Host $deleted_subjects -Separator ", "
        Write-Host ""
    }
Write-Host "There are $($organizer_added.count) rooms adding the organizer to the meeting subject"
    if ($organizer_added) {
        Write-Host $organizer_added -Separator ", "
        Write-Host ""
    }
Write-Host "There are $($deleted_bodies.count) rooms deleting the meeting body"
    if ($deleted_bodies) {
        Write-Host $deleted_bodies -Separator ", "
        Write-Host ""
    }
Write-Host "There are $($private_flag_reset.count) rooms clearing the private flag on meetings"
    if ($private_flag_reset) {
        Write-Host $private_flag_reset -Separator ", "
        Write-Host ""
    }
Write-Host "There are $($not_auto_accept.count) rooms not configured to Auto Accept"
    if ($not_auto_accept) {
        Write-Host $not_auto_accept -Separator ", "
        Write-Host ""
    }
Write-Host "There are $($process_external.count) rooms configured to process external (forwarded) meetings"
    if ($process_external) {
        Write-Host $process_external -Separator ", "
        Write-Host ""
    }
}




```

Adding a One-Touch Join O365 Graph integration on Pexip Infinity

In this step you log in to the Pexip Infinity Administrator interface and add details of the Graph API application you have just configured.

Configuring the O365 Graph integration

From the Pexip Infinity Administrator interface, go to **One-Touch Join > OTJ Graph Integrations**.

Option	Description
Name	The name of this One-Touch Join O365 Graph integration.
Description	An optional description of this One-Touch Join O365 Graph integration.
Client ID	<p>The Application (client) ID which was generated by Azure when you created the OTJ Graph API application (see Creating and configuring a new App registration in Azure).</p> <p> This is available in Azure under App Registrations, by selecting the application and viewing the Essentials section.</p>
Client secret	<p>The client secret of the OTJ Graph API application.</p> <p> If you didn't copy this at the time the registration was created, you'll need to generate a new one.</p>
OAuth 2.0 token endpoint URL	<p>The URL of the OAuth 2.0 (v2) token endpoint for this OTJ Graph API application.</p> <p> This is available in Azure under App Registrations, by selecting the application and then selecting the Endpoints tab.</p>
Advanced options	
Maximum Graph API requests	<p>The maximum number of API requests that can be made by OTJ to the Microsoft Graph API in a 24-hour period.</p> <p>The default of 1,000,000 should be sufficient for most deployments — for more information, see Frequency and limitations on calendar requests.</p> <p>We do not recommend increasing this quota unless you have deployed a dedicated One-Touch Join platform, because it will impact the performance of the Conferencing Nodes.</p>
Graph API FQDN	The FQDN to use when connecting to the Graph API.

Next steps

You must now configure the remainder of the One-Touch Join components on Pexip Infinity, as described in [Configuring Pexip Infinity for One-Touch Join](#).

Configuring Office 365 using EWS for One-Touch Join

This topic describes how to implement Pexip Infinity's One-Touch Join feature in a Microsoft Office 365 environment, by using a service account authenticated using OAuth and the EWS API to enable the One-Touch Join service to access calendars used for OTJ.

- i** The EWS API is being **deprecated by Microsoft**, so for **new** One-Touch Join deployments in Office 365 environments you should instead **use the Graph API** to provide access to room resource mailboxes. This topic is intended as a reference for **existing** deployments.

The process involves the following steps, described in detail in the sections that follow:

1. **Creating a service account** for One-Touch Join. This service account will be used by One-Touch Join to read each room resource's calendar.
 - i** This should be a different service account to any used for VMR Scheduling for Exchange, because the configuration will be different.
2. **Configuring Application Impersonation** on the service account.
 - i** For more information and guidelines on the use of application impersonation in Exchange, see **Permitting the service account to access calendars**.
3. **Configuring calendar processing** within Exchange.
4. **Enabling OAuth authentication** for the service account.
5. **Creating an associated Exchange integration** on Pexip Infinity.

Prerequisites

Before you begin, ensure that the following configuration is complete:

1. Ensure **each physical room** that will have a One-Touch Join endpoint in it has an associated **room resource with an email address**.
2. **Enable auto calendar processing for each room resource**, so that the room will automatically accept meeting requests if it is available, and automatically decline an invitation if it is already booked.
3. We recommend that if you are using Safe Links, you modify your Safe Links policy so that URLs are not rewritten in any meeting invitations sent to room resources used by One-Touch Join endpoints.
4. Ensure that you have a Microsoft license available for the service account; this is required for the service account to access Exchange.
5. Ensure you have admin access to your Office 365 web interface, and access to the Microsoft Exchange Online and Azure Active Directory Modules for Windows PowerShell. (If you are connecting from your Windows PC for the first time, you may need to install these modules. See these Microsoft articles about connecting to **Exchange online** and **Microsoft 365** with PowerShell for more information.)
6. Ensure you have access to your Exchange Admin Center (EAC) web interface, and access to Exchange Management PowerShell.
7. If your Exchange server does not use a globally trusted certificate, you must upload a custom CA certificate.

Creating a service account

In this step, you create a dedicated service account to use to log in to Exchange to access the calendars of the room resources being used for One-Touch Join. After creating the service account, you must assign it an appropriate Exchange license, such as Office 365 Enterprise E1, Office 365 Business Basic (formerly Essentials) or one of the Exchange Online plans.

This service account should only be used with One-Touch Join. However, you can use the same Exchange service account for multiple One-Touch Join integrations.

If the service account is subject to a password rotation policy or uses multi-factor authentication (MFA), then each time the password changes or the MFA is refreshed, you must **sign in to the service account again** via the Pexip Infinity Administrator interface.

You can create a new service account using **either** the Office 365 admin portal or PowerShell, as follows:

O365

1. Go to portal.office.com and log in as the administrator.
2. Go to the admin portal by selecting the Admin tile (this takes you to <https://portal.office.com/adminportal/home#/homepage>).
3. From the Users section, select Add a user and complete the necessary fields:
 - a. In the Password section:
 - Select Let me create the password.
 - Uncheck Make this user change their password when they first sign in.
 - b. In the Product licenses section, assign an appropriate product license from the available list.

4. Select Add to create the user.

PowerShell

You must run Powershell as administrator.

Establishing a remote connection

To use PowerShell for Office 365 you first need to connect remotely. These commands install the required PowerShell modules (if they are not already installed) and then connects to Exchange Online:

```
#If not installed, install Exchange Online Module
Install-Module ExchangeOnlineManagement

#If not installed, install Azure AD Module
Install-Module -Name AzureAD

#Connect to Exchange Online and AzureAD, works also with a MFA
enabled account
Connect-ExchangeOnline
```

Creating the service account

The first command lets the administrator type in a password for the service account as a secure string. This password variable is then used in the second command to create a mailbox for the service account. The remaining commands log you into Azure AD and then set the password of the service account to never expire.

```
#Capture password for service account
$password = Read-Host "Enter password" -AsSecureString

# Create service account and mailbox
New-Mailbox -Name "<Account Name>" -MicrosoftOnlineServicesID
"<UPN>" -Password $password -Alias "<Account Alias>" -FirstName
"<Account First Name>" -LastName "<Account Last Name>" -DisplayName
"<Account Name>"

#Connect to AzureAD
Connect-AzureAD

#Set password policy
Set-AzureADUser -ObjectId "<UPN>" -PasswordPolicies
DisablePasswordExpiration
```

Example New-Mailbox command:

```
New-Mailbox -Name "Pexip OTJ Service Account" -
MicrosoftOnlineServicesID pexip-otj-svc@example.com -Password
$password -Alias pexip-otj-svc -FirstName "Pexip OTJ" -LastName
"Service Account" -DisplayName "Pexip OTJ Service Account"
```

Example Set-AzureADUser command:

```
Set-AzureADUser -ObjectId pexip-otj-svc@example.com -
PasswordPolicies DisablePasswordExpiration
```

Assigning a license to the service account

You must now assign an appropriate license to the service account.

See <https://docs.microsoft.com/en-us/powershell/azure/active-directory/enabling-licenses-sample> for information on how to do this.

Configuring Application Impersonation on the service account

In this step, you create a new Distribution Group, and add the rooms to be used for One-Touch Join to the group. You then use PowerShell commands to make it so that the service account will only be able to impersonate members of that Group.

Configuring Application Impersonation in this way means that when a room is added to the group, the service account will automatically be able to impersonate it. Likewise, when a room is removed, the service account will no longer be able to impersonate it.

Creating a new Distribution Group

1. Go to admin.microsoft.com and log in as the administrator.
2. From the menu on the left hand side, select **Groups > Add a group**.
3. For the **Group Type**, select **Distribution List**. Enter a name, email address and description and select **Add**.
4. Add as members of the Group the rooms to be used for One-Touch Join. These will be the rooms that the service account will impersonate.

Note that the service account should **not** be added as a member of this distribution group. Instead, this step allows the service account to impersonate any member of this distribution group (i.e. any of the room resources).

5. Open up a remote PowerShell connection to Office 365 and import an Exchange session. For example see <https://docs.microsoft.com/en-us/powershell/exchange/connect-to-exchange-online-powershell?view=exchange-ps>

Configuring application impersonation

We recommend that you use combined PowerShell commands to configure application impersonation for the service account. This allows you to use variables, thus reducing possible copy and paste errors.

1. You may need to enable customization, if this has not already been done within your organization:

```
Enable-OrganizationCustomization
```

2. Configure the following variables with the values you actually want to use:

- `otj_group_id`: the email of the distribution list whose members you want to be impersonated.
- `otj_service_account`: the email of the service account you want to grant impersonation to.
- `management_scope_to_create`: the name you want the newly created management scope to have.
- `impersonation_role_name_to_create`: the name you want the newly created impersonation role to have.

For example:

```
$otj_group_id = "otjrooms@example.com"
$otj_service_account = "pexip-otj-svc@example.com"
$management_scope_to_create = "OTJ Management Scope"
$impersonation_role_name_to_create = "OTJ Impersonation"
```

3. Create the management scope:

```
$otj_group = Get-DistributionGroup -Identity $otj_group_id
$otj_group_dn = $otj_group.DistinguishedName
$restriction_filter = "MemberOfGroup -eq '$otj_group_dn'"
New-ManagementScope -Name $management_scope_to_create -RecipientRestrictionFilter $restriction_filter
```

Example output:

Name	ScopeRestrictionType	Exclusive	RecipientRoot	RecipientFilter
OTJ Management Scope	RecipientScope	False	MemberOfGroup	-eq 'CN=OTJ Rooms2111430164340,OU...

4. Set up application impersonation using the previously created management scope:

```
New-ManagementRoleAssignment -Name $impersonation_role_name_to_create -Role ApplicationImpersonation -User $otj_service_account -CustomRecipientWriteScope $management_scope_to_create
```

Example output:

Name	Role	RoleAssigneeName	RoleAssigneeType	AssignmentMethod	EffectiveUserName
OTJ Impersonation	ApplicationImp...	pexip-otj-svc	User	Direct	

5. Verify that the above commands worked as expected. In the following command, replace `<resource_email>` with the email of the room resource mailbox you want to test. If it is a room which is a member of the distribution list, it should show the OTJ Impersonation in the returned roles. If it is anything else outside of the distribution list, it should not have the OTJ Impersonation listed, which means the OTJ service account does not have permission to impersonate that user.

```
Get-ManagementRoleAssignment -Role ApplicationImpersonation -WritableRecipient "<resource_email>" | Format-List Name, Role, RoleAssignee, CustomRecipientWriteScope
```

Expected output:

Name	: OTJ Impersonation
Role	: ApplicationImpersonation
RoleAssignee	: pexip-otj-svc

Configuring calendar processing on room resource mailboxes

In this step, you change the calendar processing settings for room resources from the default to those required to support One-Touch Join.

Recommended configuration

To take full advantage of the functionality offered by One-Touch Join, we recommend that, for One-Touch Join room resources, you change the following calendar processing options from the default:

1. The meeting invite body is deleted by default. If you want One-Touch Join to parse meeting details from the body then you must set the `DeleteComments` property to *False*. If you leave this set to *True*, only those rules that process information in the calendar headers can be used (because the body will be deleted).
2. When a meeting invite is received by a resource mailbox, by default the meeting subject is deleted and is replaced with the name of the organizer (for more information, see <https://support.microsoft.com/en-gb/help/2842288/resource-mailbox-s-calendar-shows-the-organizer-s-name-instead-of-the>).
Because One-Touch Join accesses the meeting invites through the resource mailboxes, this default behavior means it won't have access to the original subject. You can choose to leave the default behavior for privacy reasons, or you can modify the calendar processing options for each mailbox so that the meeting subject is available and thus can be displayed on the meeting room endpoints.
3. The private flag is cleared by default. If you want meetings that are marked as private by the organizer to remain marked as private in the room mailbox, you must set the `RemovePrivateProperty` flag to *False*.
4. Room resources created using PowerShell commands may by default have `AutomateProcessing` set to *AutoUpdate*. In these cases it should be changed to *AutoAccept*.

PowerShell command

To modify the calendar processing on a room from the default settings to those we recommend for One-Touch Join, [connect to Exchange Online PowerShell](#) and use the following PowerShell command (replacing `<resource_email>` with the address of the room resource whose processing you want to change):

```
Set-CalendarProcessing -Identity <resource_email> -DeleteComments $False -DeleteSubject $False -AddOrganizerToSubject $False -  
RemovePrivateProperty $False -AutomateProcessing "AutoAccept"
```

Optional configuration

Hiding invitation details from other users

In order for One-Touch Join to function fully, the service account must be able to access the body of the invitation (which is why we recommend that you set the `DeleteComments` property to *False*). However, this means that all other users in your deployment with access to the room resource calendar may also be able to view the body of the invitation (depending on your deployment's other policies). If you want to prevent this, you can use the following PowerShell command to restrict what users can see by default, without restricting what the service account can access.

In the following command, replace `resource_name` with the name of the room resource, and replace `role` with one of the following roles:

- **AvailabilityOnly**: users can view the room's availability, but nothing else.
- **LimitedDetails**: users can view the room's availability and the meeting subject and location, but not the body of the invitation.

```
Set-MailboxFolderPermission "resource_name:\Calendar" -User Default -AccessRights role
```

Allowing forwarding of external invitations

Below is some recommended configuration to enable external invitations to be forwarded to your internal OTJ room resources so that the meetings can be joined from those endpoints. In all cases, we recommend that you consult your Exchange administrator to determine what is appropriate in your environment.

- If you want to enable users to forward invitations from other organizations to your OTJ room resources, you must set the **ProcessExternalMeetingMessages** flag to **True**. Note that this will allow any users external to your organization to invite the resource directly. To prevent this, you can use an Exchange transport rule similar to the example shown below so that only users internal to your organization can forward external invitations to OTJ meeting rooms.

Block external invites to OTJ rooms, allow internal forwarding

Conditions Settings

Name *

Block external invites to OTJ rooms, allow internal forwarding

Apply this rule if *

The recipient is a member of this group +

The recipient is a member of 'otj.rooms@pexample.com'

And

The message properties include the message type

The message type is 'Calendaring'

And

The sender is external/internal

The sender is located 'NotInOrganization'

Do the following *

Block the message reject the message and i... +

Reject the message with the explanation 'This room does not accept bookings from an external account.'

And

Block the message reject the message with t... +

Reject the message with the enhanced status code '5.7.1'

Except if

The message headers... includes any of these wo... +

'Sender' message header includes 'pexample.com'

Save Cancel

- If your Microsoft Exchange environment uses a security application (such as Office 365 ATP, or Mimecast) to re-write URLs, this may prevent OTJ from being used to join external Microsoft Teams meetings (for example, when a user inside your organization forwards an external Microsoft Teams meeting invitation to an OTJ room resource in order to join the meeting from that endpoint). To enable users to join these meetings using OTJ, you must ensure that the security application's URL re-write rules include an exception for any URL starting with the domain `https:\\teams.microsoft.com\\`

Checking calendar processing settings

The following PowerShell command can be used to check calendar processing settings on all of the rooms in the Distribution Group that was created for One-Touch Join.

We recommend copying and saving this as a file and running it from within PowerShell.

Before running, ensure that you edit `$otj_group_id = "otjrooms@example.com"` to use the email of the Distribution Group used in your own deployment.

```
$deleted_subjects = @()
$organizer_added = @()
$deleted_bodies = @()
$private_flag_reset = @()
$not_auto_accept = @()
$process_external = @()
$otj_group_id = "otjrooms@example.com"

Get-DistributionGroupMember -Identity $otj_group_id -ResultSize Unlimited | ForEach-Object {
    Write-Host "Checking room '$($_.name)'"
    $processing = Get-CalendarProcessing -Identity $_.name
    $pass = $true
    if ($processing.DeleteSubject) {
        Write-Host "WARNING: The room '$($_.name)' is deleting the meeting subject" -ForegroundColor Red
        $deleted_subjects += $_.name
        $pass = $false
    }
    if ($processing.AddOrganizerToSubject) {
        Write-Host "WARNING: The room '$($_.name)' is adding the organizer to the meeting subject" -ForegroundColor Red
        $organizer_added += $_.name
        $pass = $false
    }
    if ($processing.DeleteComments) {
        Write-Host "WARNING: The room '$($_.name)' is deleting the meeting body" -ForegroundColor Red
        $deleted_bodies += $_.name
        $pass = $false
    }
    if ($processing.RemovePrivateProperty) {
        Write-Host "WARNING: The room '$($_.name)' is clearing the private flag on meetings" -ForegroundColor Red
        $private_flag_reset += $_.name
        $pass = $false
    }
    if ($processing.AutomateProcessing -ne "AutoAccept") {
        Write-Host "WARNING: The room '$($_.name)' is not configured to Auto Accept. Processing='$(($processing.AutomateProcessing))'" -
        ForegroundColor Red
        $not_auto_accept += $_.name
        $pass = $false
    }
    # Optional permission for allowing the external invites:
    if ($processing.ProcessExternalMeetingMessages) {
        Write-Host "The room '$($_.name)' is configured to process external (forwarded) meetings"
        $process_external += $_.name
    }
    if ($pass) {
        Write-Host "INFO: All checks passed for room '$($_.name)'" -ForegroundColor Green
    }
}

Write-Host "Summary:"
Write-Host "There are $($deleted_subjects.count) rooms deleting the meeting subject"
if ($deleted_subjects) {
    Write-Host $deleted_subjects -Separator ", "
    Write-Host ""
}
Write-Host "There are $($organizer_added.count) rooms adding the organizer to the meeting subject"
if ($organizer_added) {
    Write-Host $organizer_added -Separator ", "
    Write-Host ""
}
Write-Host "There are $($deleted_bodies.count) rooms deleting the meeting body"
if ($deleted_bodies) {
    Write-Host $deleted_bodies -Separator ", "
    Write-Host ""
}
Write-Host "There are $($private_flag_reset.count) rooms clearing the private flag on meetings"
if ($private_flag_reset) {
    Write-Host $private_flag_reset -Separator ", "
    Write-Host ""
}
}
```



```
Write-Host "There are $($not_auto_accept.count) rooms not configured to Auto Accept"
if ($not_auto_accept) {
    Write-Host $not_auto_accept -Separator ", "
    Write-Host ""
}
Write-Host "There are $($process_external.count) rooms configured to process external (forwarded) meetings"
if ($process_external) {
    Write-Host $process_external -Separator ", "
    Write-Host ""
}
```

Enabling OAuth authentication

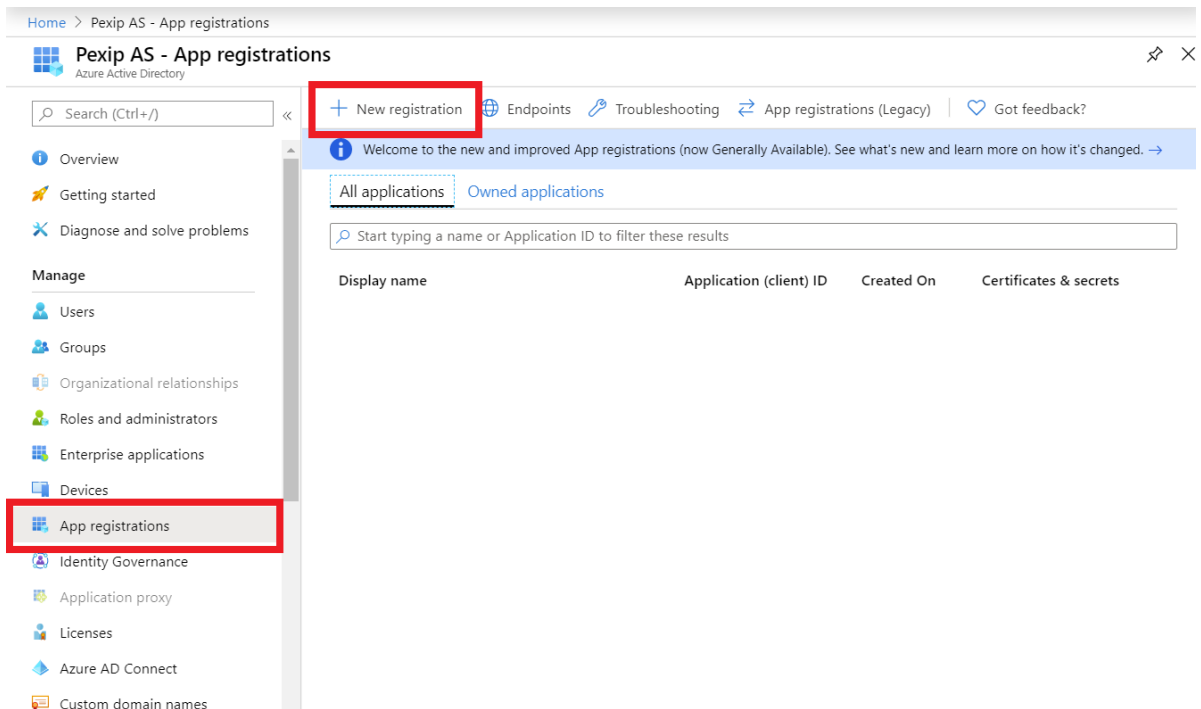
In this step, you enable OAuth authentication for the service account that One-Touch Join uses to log in to Exchange.

As of October 2022, Microsoft will stop supporting and fully decommission basic authentication for EWS to access Exchange Online (for more information, see [Microsoft's announcement](#)). We therefore strongly recommend that for Office 365, all new deployments authenticate the service account using OAuth 2.0, and all existing deployments are updated to enable this option as soon as possible.

To use OAuth for the service account, you must create an app registration in Azure and then use the settings from this app registration when enabling and configuring the OAuth options within the One-Touch Join Exchange integration.

Create a new App Registration in Azure

1. Log into the Azure portal at aad.portal.azure.com.
2. From the main panel on the left, select **Azure Active Directory**.
3. Select **App Registrations** and then **New registration**:



4. In the **Register an application** panel, enter the following options:
 - a. **Name:** this can be anything you wish. In our example we have used *Pexip OTJ App*.
 - b. **Supported account types:** select *Accounts in this organizational directory only*.
 - c. **Redirect URI:** from the drop-down menu, select *Public client/native (mobile and desktop)*. The URI must use the IP address or FQDN of the Management Node, in the format
`https://<Management Node Address>/admin/platform/mjxexchangedeployment/oauth_redirect/`
In our example we have used `https://infinity.example.com/admin/platform/mjxexchangedeployment/oauth_redirect/`
You will need to enter this as the **OAuth redirect URI** when configuring a One-Touch Join Exchange integration.
- i** The **OAuth redirect URI** is the page on the Administrator interface to which the Pexip Infinity administrator will be returned after they have successfully signed in to the service account. Because it is a page on the Management Node, this URI is internal to your deployment and only needs to be accessible from the administrator's web browser; you do not need to make it externally accessible. This URI must be the same on Azure and Pexip Infinity in order for Azure to validate the sign-in request.

Home > Pexip AS - App registrations > Register an application

Register an application

*** Name**
The user-facing display name for this application (this can be changed later).

Pexip OTJ app ✓

Supported account types
Who can use this application or access this API?

☒ Accounts in this organizational directory only (Pexip AS only - Single tenant)

☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

[Help me choose...](#)

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Public client/native (mobile ... ▼ https://infinity.example.com/admin/platform/mjxexchangedeployment ✓

By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

5. Select **Register**.
A new panel will open where you can configure your application.
6. From the panel on the left, select **API permissions**.
7. Select **Add a permission**.
8. From the **Request API permissions** panel, select **APIs my organization uses**, search for **Office 365 Exchange Online** and select it:

Request API permissions



Select an API

Microsoft APIs

APIs my organization uses

My APIs

Apps in your directory that expose APIs are shown below

<input type="text" value="office"/>	
Name	Application (client) ID
Office 365 Exchange Online	00000002-0000-0ff1-ce00-000000000000
Office 365 Information Protection	2f3f02c9-5679-4a5c-a605-0de55b07d135
Office 365 Management APIs	c5393580-f805-4401-95e8-94b7a6ef2fc2
Office 365 Search Service	66a88757-258c-4c72-893c-3e8bed4d6899
Office 365 SharePoint Online	00000003-0000-0ff1-ce00-000000000000
Office Agent Service	5225545c-3ebd-400f-b668-c8d78550d776
Office Delve	94c63fef-13a3-47bc-8074-75af8c65887a
Office Hive	166f1b03-5b19-416f-a94b-1d7aa2d247dc
Office Personal Assistant at Work Service	28ec9756-deaf-48b2-84d5-a623b99af263
Office Scripts Service	62fd1447-0ef3-4ab7-a956-7dd05232ecc1
Office Shredding Service	b97b6bd4-a49f-4a0c-af18-af507d1da76c
Office365 Zoom	0d38933a-0bbd-41ca-9ebd-28c4b5ba7cb7
OfficeServicesManager	9e4a5442-a5c9-4f6f-b03f-5b9fcaaf24b1

9. Select **Delegated permissions**, and from the **Select permissions** list, expand **EWS** and select **Access mailboxes** as the signed-in user via **Exchange Web Services**, and then select **Add permissions**:

Request API permissions

[← All APIs](#)

Office 365 Exchange Online

<https://outlook-tdf-2.office.com/>

What type of permissions does your application require?

Delegated permissions

Your application needs to access the API as the signed-in user.

Application permissions

Your application runs as a background service or daemon without a signed-in user.

Select permissions

[expand all](#)

Permission	Admin consent required
> Calendars	
> Contacts	
> EAS	
✓ EWS (1)	
<input checked="" type="checkbox"/> EWS.AccessAsUser.All ⓘ Access mailboxes as the signed-in user via Exchange Web Services	
> Exchange	

Add permissions

Discard

Taking note of configuration

When you [Configure the One-Touch Join Exchange integration](#) and enable OAuth authentication for the service account, you'll need to provide the following information from Azure:

- **Application (client) ID:** this was generated for you by Azure when you saved the App Registration:

Home > Pexip AS - App registrations > Pexip OTJ App

Pexip OTJ App

Search (Ctrl+ /) << Delete Endpoints

Overview	Display name Pexip OTJ App	Supported account types My organization only
Quickstart	Application (client) ID 86714057-cfda-4a43-94cc-3fa6dc5816c2	Redirect URIs 0 web, 1 public client
Manage	Directory (tenant) ID 983be362-1f4b-4a72-b5c5-ad805e211fa0	Application ID URI Add an Application ID URI
Branding	Object ID 8710eaa3-f459-4d4b-bca4-bb079d84d96c	Managed application in local directory Pexip OTJ App
Authentication		
Certificates & secrets		

i You can find this again in Azure under **Azure Active Directory > App Registrations**, under the **Application (client) ID** column. You will need to enter this as the **OAuth client ID** when configuring the One-Touch Join Exchange integration.

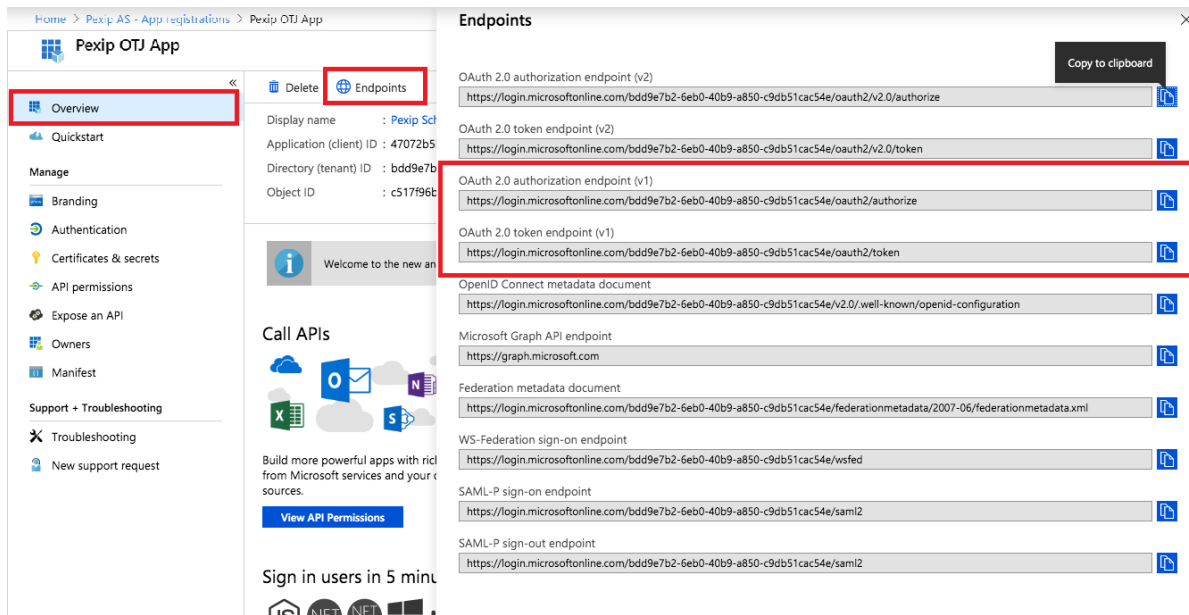
- **Redirect URI:** this is the URI you entered when creating the App Registration.

- i** You can find this again in Azure under **Azure Active Directory > App Registrations**, clicking on the app registration, and then clicking **Redirect URIs**.

You will need to enter this as the **OAuth redirect URI** when configuring the One-Touch Join Exchange integration.

You will also need to know the OAuth Endpoints to use. To find this information:

1. In the Azure Portal, select **Overview > Endpoints**.
2. Copy the URL of the **OAuth 2.0 authorization endpoint (v1)**.
i Ensure that you use the URL for **... endpoint (v1)**, not **... endpoint (v2)**.
You will need to enter this as the **OAuth authorization endpoint** when configuring the One-Touch Join Exchange integration.
3. Copy the URL of the **OAuth 2.0 token endpoint (v1)**.
i Ensure that you use the URL for **... endpoint (v1)**, not **... endpoint (v2)**.
You will need to enter this as the **OAuth token endpoint** when configuring the One-Touch Join Exchange integration.



Adding a One-Touch Join Exchange integration on Pexip Infinity


In this step you log in to the Pexip Infinity Administrator interface and add details of the Exchange deployment you are integrating with, including details of the service account and OAuth access (based on the configuration you have just set up in Exchange). You must then sign in to Exchange using the service account.

Configuring the Exchange integration

From the Pexip Infinity Administrator interface, go to **One-Touch Join > OTJ Exchange Integrations**.

Option	Description
Name	The name of this One-Touch Join Exchange integration.
Description	An optional description of this One-Touch Join Exchange integration.
Service account username	The username of the <u>service account to be used by the One-Touch Join Exchange integration</u> . This is usually in the format <code>name@domain</code> .

Option	Description
Enable OAuth	<p>Enable this option to authenticate the service account using OAuth 2.0. (This option is only supported for Exchange in Office 365.)</p> <p>i As of October 2022, Microsoft will stop supporting and fully decommission basic authentication for EWS to access Exchange Online (for more information, see Microsoft's announcement). We therefore strongly recommend that for Office 365, all new deployments authenticate the service account using OAuth 2.0, and all existing deployments are updated to enable this option as soon as possible.</p>
Enable NTLM	Leave this option disabled. (NTLM is supported for Exchange on-premises only.)
OAuth client ID	<p>(Available if OAuth has been enabled)</p> <p>The Application (client) ID which was generated by Azure when creating an App Registration in Azure Active Directory (see Taking note of configuration).</p>
OAuth redirect URI	<p>(Available if OAuth has been enabled)</p> <p>The redirect URI you entered when creating an App Registration in Azure Active Directory.</p> <p>This must be in the format <code>https://<Management Node Address>/admin/platform/mjxexchangedeployment/oauth_redirect/</code></p> <p>The OAuth redirect URI is the page on the Administrator interface to which the Pexip Infinity administrator will be returned after they have successfully signed in to the service account. Because it is a page on the Management Node, this URI is internal to your deployment and only needs to be accessible from the administrator's web browser; you do not need to make it externally accessible. This URI must be the same on Azure and Pexip Infinity in order for Azure to validate the sign-in request.</p>
OAuth authorization endpoint	<p>(Available if OAuth has been enabled)</p> <p>The URL of the OAuth authorization endpoint (see Taking note of configuration).</p> <p>Ensure that you use the URL for ... endpoint (v1), not ... endpoint (v2).</p>
OAuth token endpoint	<p>(Available if OAuth has been enabled)</p> <p>The URL of the OAuth token endpoint (see Taking note of configuration).</p> <p>i Ensure that you use the URL for ... endpoint (v1), not ... endpoint (v2).</p>
Advanced options	
Find Items Request Quota	<p>The number of Find Item requests that can be made by OTJ to your Exchange Server in a 24-hour period.</p> <p>The default of 1,000,000 should be sufficient for most deployments — for more information, see Frequency and limitations on calendar requests.</p> <p>We do not recommend increasing this quota unless you have deployed a dedicated One-Touch Join platform, because it will impact the performance of the Conferencing Nodes.</p>
OTJ Exchange Autodiscover URLs	
i This section is optional and will generally only be required if the Autodiscover URLs in your deployment do not use a standard location.	
Name	The name of this Exchange Autodiscover URL.
Description	An optional description of this Exchange Autodiscover URL.

Option	Description
Autodiscover URL	<p>The URL used to connect to the Autodiscover service on the Exchange deployment.</p> <p>If you are using Office 365, you may need to enter your autodiscover URL manually, particularly if you are using a hybrid Exchange deployment. If your OTJ room resources and service account are hosted on O365, then you should enter https://autodiscover-s.outlook.com/autodiscover/autodiscover.svc as the Autodiscover URL.</p> <p> The URL must end in .svc; URLs ending in .xml are not supported.</p>

When you have completed the above fields, select **Save**. You will be returned to the main OTJ Exchange Integration page. You must now sign in to the Exchange integration using the service account details you have just created.

Signing in to the service account

If you have enabled OAuth for the first time, you must sign in to the service account after saving the configuration of the One-Touch Join Exchange integration.

You may also need to re-sign in to the service account if:

- the service account password has changed
- the service account uses multi-factor authentication (MFA) and the MFA is refreshed
- you disable and then subsequently re-enable OAuth
- you update any of the following configuration for the One-Touch Join Exchange integration:
 - Service account username
 - OAuth client ID
 - OAuth token endpoint
- the Management Node has been offline for more than 90 days.

To sign in to the service account:

1. Ensure you have signed out of **all** Microsoft accounts on your device, including the Microsoft Azure portal.
2. From the Management Node, go to **One-touch Join > OTJ Exchange Integrations**, select the Exchange integration you have just created. At the bottom of the **Change OTJ Exchange Integration** page, select **Sign in to service account**:

The redirect URI you entered when creating an App Registration in Azure Active Directory. It should be in the form of `https://<domain>/<path>`.

OAuth authorization endpoint

Currently: https://infinity.example.com/admin/platform/otj/oauth_redirect/

Change:

The URI of the OAuth authorization endpoint. This should be copied from the 'Endpoints' section in Azure Active Directory.

OAuth token endpoint

Currently: https://infinity.example.com/admin/platform/otj/oauth_redirect/

Change:

The URI of the OAuth token endpoint. This should be copied from the 'Endpoints' section in Azure Active Directory.

Save **Save and add another** **Sign in to service account**

You will be taken to the **Sign in to service account** page.:

]pexip[Infinity Conferencing Platform

Status ▾ History & Logs ▾ System ▾ Platform ▾ Call Control ▾ Services ▾ Users & Devices ▾ **One-Touch Join ▾** Utilities ▾

Sign in to service account

Please open the link below. It will take you to a Microsoft sign-in page where you must sign in as the service account with username **test**.

Warning: If you are already signed into a Microsoft account, you may not be prompted to enter a username. Please make sure you are not signed in to a Microsoft account before opening the link.

Sign-in link

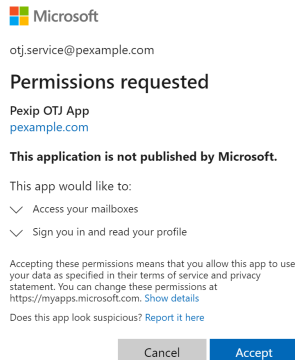
Sign in

https://infinity.example.com/admin/platform/otj/oauth_redirect/?state=2e546d1d-cb4f-46d1-b6c6-f7caf1d97f44&redirect_uri=

3. Copy the **Sign in** link and paste it into a new browser tab.

4. Sign in as the service account.

You are asked to permit the OTJ application to sign in as the service account, and to access the mailboxes that the service account has been granted access to. (The service account will only have access to the mailboxes of the OTJ room resources, if you completed the steps in [Configuring Application Impersonation on the service account.](#))



- i** If there is an option to **Consent on behalf of your organization**, do not select this — consent only needs to be given to the service account.

5. Select **Accept**.

You are returned to the Management Node.

6. You may be asked to sign in to the Management Node again. If so, you must sign in to the Management Node (using your Management Node credentials) to complete the process of signing in to the service account.

When complete, you are returned to the **Sign in to service account** page and see the message **Successfully signed in**.

Next steps

You must now configure the remainder of the One-Touch Join components on Pexip Infinity, as described in [Configuring Pexip Infinity for One-Touch Join](#).

Configuring Google Workspace for One-Touch Join

This topic describes how to configure Google Workspace in order to implement Pexip Infinity's One-Touch Join feature in a Google Workspace environment.

The process involves the following steps, described in more detail in the sections that follow:

1. [Creating a Service Account](#) to use for One-Touch Join.
 2. [Creating a room resource](#) for each physical room that will have a One-Touch Join endpoint in it.
 3. [Configuring the room resource](#) with the necessary permissions and settings to support One-Touch Join.
 4. [Updating the quota](#) for the number of user requests per 100 seconds.
 5. For larger deployments, [Requesting an increase to API limits](#).
 6. [Adding a One-Touch Join Google Workspace integration](#) on Pexip Infinity.
- i** If you have already set up a One-Touch Join Google Workspace integration and simply wish to add an existing room to it, you need only [configure the room resource](#) in Google Workspace and then [add the endpoint to the Google Workspace integration](#) in Pexip Infinity.

We recommend that you authorize One-Touch Join to access calendar information using a service account, as described in the following steps. This method (sometimes referred to as two-legged OAuth) offers the easiest setup for One-Touch Join, and is recommended by Google because it is designed for server-to-server applications (for more information, see <https://developers.google.com/identity/protocols/oauth2/service-account>). Alternatively, you may need to use a Google Workspace domain user for authorization (sometimes referred to as three-legged OAuth); for instructions on how to do this, see [Configuring Google Workspace for domain user authorization](#).

Prerequisites


In the deployment model described below, the service account will require access to the endpoints' calendars. Google Workspace service accounts always use the `iam.gserviceaccount.com` domain rather than your own domain, so you will need to configure Google Workspace to [allow endpoint calendars to be shared externally](#). This does not in itself allow any external accounts to access the calendars — each calendar must then also explicitly nominate the accounts with whom it is to be shared.

Some enterprises will require internal approval for this configuration, so you should confirm that it will be permitted within your deployment. If not, you can consider [Configuring Google Workspace for domain user authorization](#) as an alternative.

Creating a service account

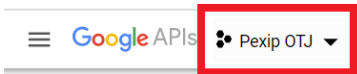
In this step, you create a project to use for One-Touch Join. You then create the service account that One-Touch Join will use to access the room resources' calendars, and generate a private key that One-Touch Join will use to authenticate when signing in to Google Workspace as the service account.

The service account belongs to the project you have created for OTJ. It can be used for multiple One-Touch Join Google Workspace integrations.

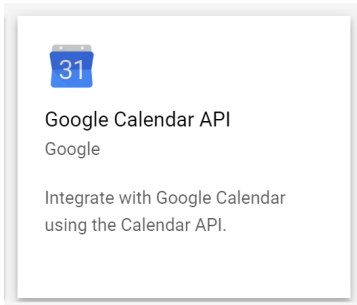
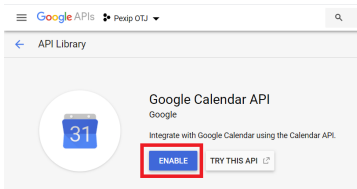
1. Creating a new project:
 - a. Go to <https://console.developers.google.com> (logged in as a Google Workspace administrator).
 - b. From the top left of the page, select the down arrow:

 - c. Select **New Project**.
 - d. Enter a **Project name** (e.g. One-Touch Join) and select **Create**.

2. Enabling the Calendar API for the project:

- Go to <https://console.developers.google.com>
- From the top left of the page, select the down arrow, select your newly-created project, and select **Open**. Your new project should now be showing at the top left of the page:

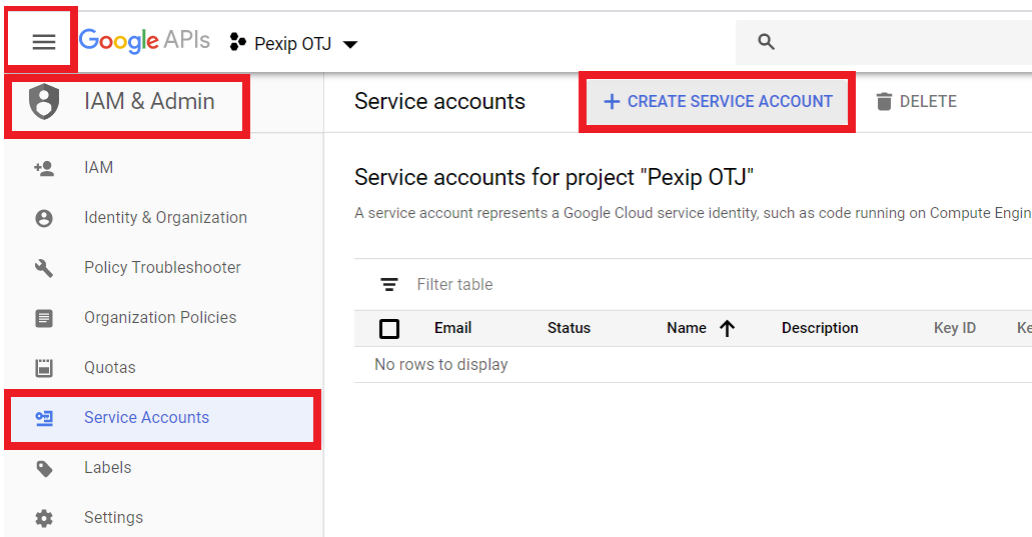


- From the navigation menu on the left of the screen, select **APIs & Services > Library**, then scroll down and select the **Google Calendar API** tile:

d. Select **Enable**:

3. Creating the service account:

- Go to <https://console.developers.google.com>
- From the navigation menu on the left of the screen, select **IAM & Admin > Service Accounts**.
- Select **Create Service Account**:



- Enter a name (e.g. One-Touch Join Calendar Reader) and select **Create**:

Google APIs Pexip OTJ

IAM & Admin

Create service account

1 Service account details — 2 Grant this service account access to project (optional)

Service account details

Service account name
OTJ calendar reader

Display name for this service account

Service account ID
otj-calendar-reader @pexip-otj-270111.iam.gserviceaccount.com

Service account description
Describe what this service account will do

CREATE CANCEL

- e. On the next page, which asks about permissions, select **Cancel** (the account does not need any of these permissions):

Google APIs Pexip OTJ

IAM & Admin

Create service account

✓ Service account details — 2 Grant this service account access to project (optional)

Service account permissions (optional)

Grant this service account access to Pexip OTJ so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Select a role

Condition
[Add condition](#)

+ ADD ANOTHER ROLE

CONTINUE CANCEL

4. Generating a key file:

- a. From the **Service accounts** page, select the service account.

Take note of the service account's Email address here - you will need it in later steps:

The screenshot shows the Google Cloud IAM & Admin console. On the left is a navigation menu with options: IAM, Identity & Organization, Policy Troubleshooter, Organization Policies, Quotas, Service Accounts (selected), Labels, and Settings. The main content area is titled 'OTJ calendar reader' and includes 'EDIT' and 'DELETE' buttons. Under 'Service account details', the 'Name' is 'OTJ calendar reader', 'Description' is empty, 'Email' is 'otj-calendar-reader@pexip-otj-270111.iam.gserviceaccount.com' (highlighted with a red box), and 'Unique ID' is '104368448542965592436'.

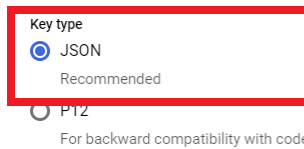
- b. From the Service account details page, select Edit, then Create Key:

This screenshot shows the same 'OTJ calendar reader' service account details page. The 'EDIT' button is now highlighted with a red box. Below the 'Service account details' section is the 'Service account status' section, which indicates the account is 'currently active' and provides buttons for 'DISABLE SERVICE ACCOUNT' and 'SHOW DOMAIN-WIDE DELEGATION'. The '+ CREATE KEY' button is also highlighted with a red box. At the bottom are 'SAVE' and 'CANCEL' buttons.

- c. Select a Key type of *JSON* and select Create:

Create private key for "OTJ calendar reader"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.



Key type

☒ JSON
Recommended

☐ P12
For backward compatibility with code using the P12 format

CANCEL

CREATE

This will download a JSON file containing the private key. This key will be required when [Adding a One-Touch JoinGoogle Workspace integration](#).

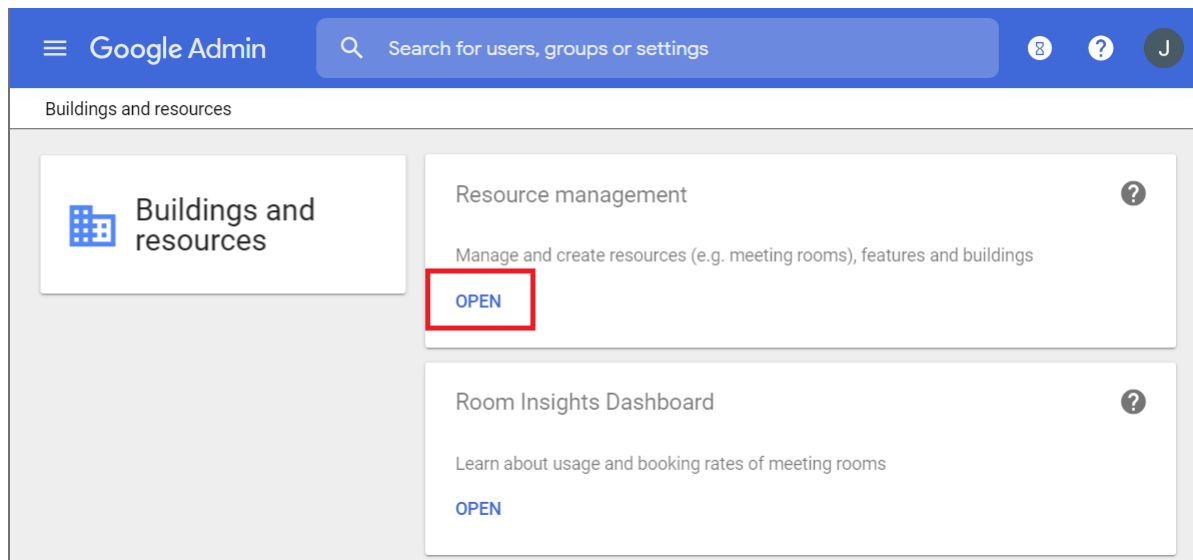
For more information on using OAuth 2.0 to authenticate the service account, see <https://developers.google.com/identity/protocols/OAuth2ServiceAccount>.

Creating a room resource

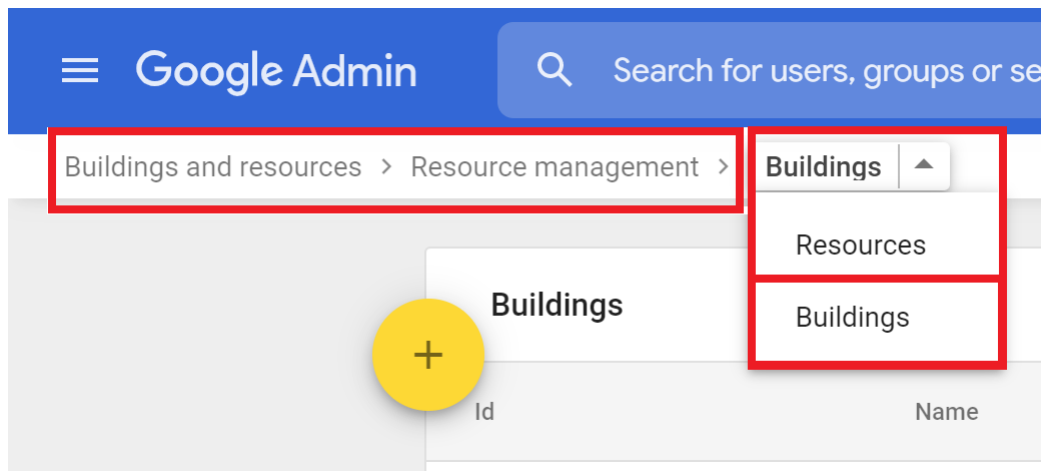
(Required only if your room resources do not already exist - otherwise you can skip this step.)

In this step, you create a room resource in Google Workspace for each physical room that is to be used for One-Touch Join. Google Workspace will automatically assign an email address to the room.

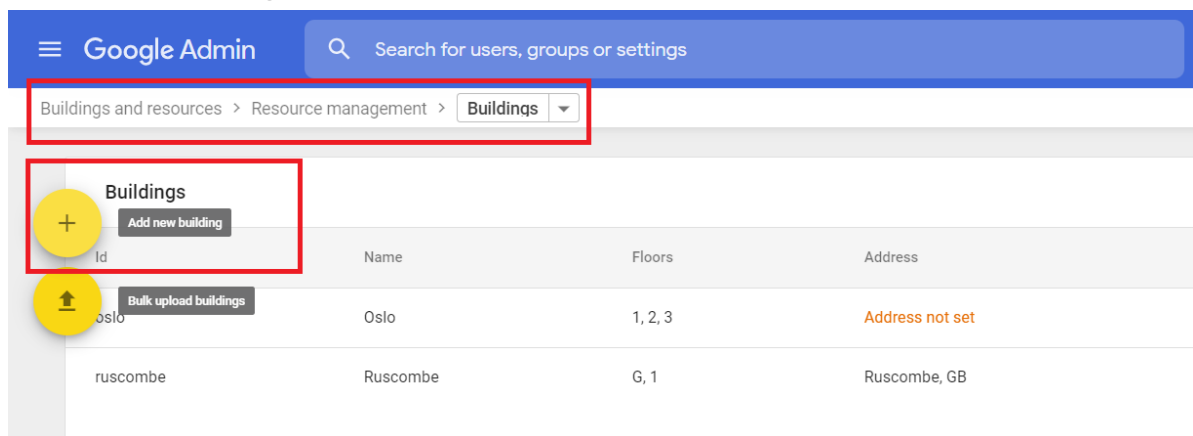
1. If a **building** for the room resource does not already exist, create one as follows:
 - a. Go to <https://admin.google.com> (logged in as a Google Workspace administrator).
 - b. Select the **Buildings and resources** tile, and then from the **Resource management** section select **Open**:



From the drop-down along the top left of the screen, select **Buildings**:



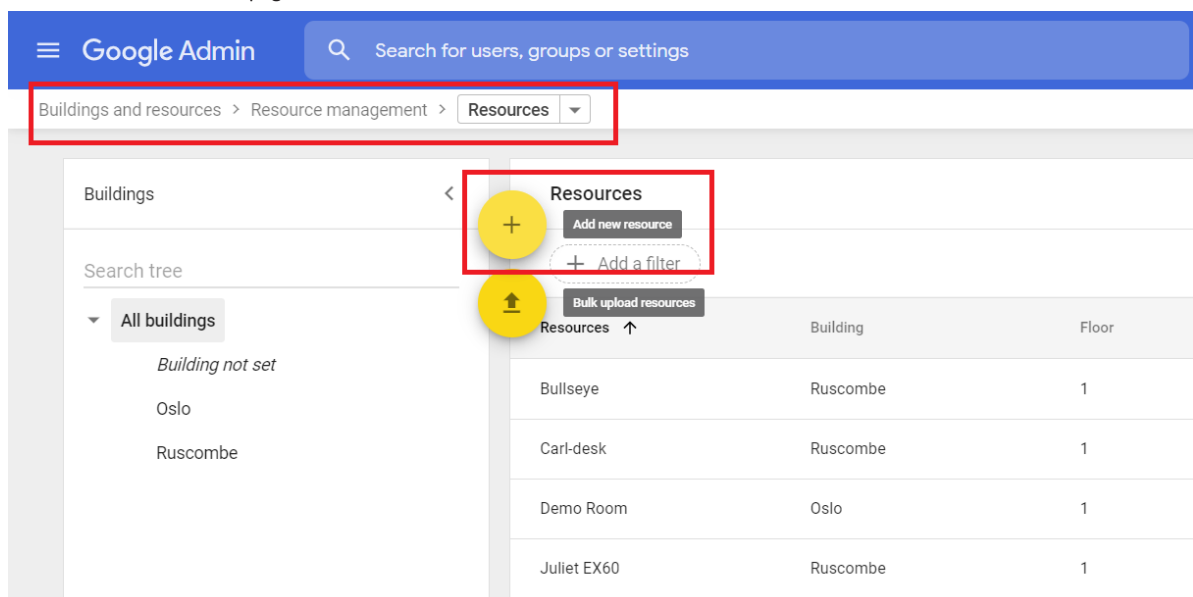
- c. Select + to Add new building:



- d. Enter a Name and the list of Floors, and select Add Building.

2. Create the room resource:

- a. Go back to the Resources page and Select + to Add new resource:




- b. For the **Category**, select *Meeting space (room, phone booth,...)*.
- c. Select the **Building** and **Floor** in which the room is located, enter a **Name** and the room's **Capacity**, then select **Add Resource**:

Add resource

Auto-generated resource name

Oslo-2-Board room (20)

Category * 

Meeting space (room, phone booth,...)

Type

meeting room

Building *

Oslo

Floor *

2

Floor section

Resource name *

Board room

Capacity *

20


Features

Define features first to add them.

User visible description

Add description (internal)


* indicates a required field

 Setting resource information, such as building, floor or capacity, affects the auto-generated resource names and room search. [Learn more.](#)

CANCEL

ADD RESOURCE

The resource will be created and added to the list. You can click on the new resource to view information about it, such as the email address it was automatically assigned.

-  For more information on setting up buildings and other resources in Google Workspace, including how to add buildings and resource in bulk and using CSV imports, see <https://support.google.com/a/answer/1033925>.

Configuring the room resource

In these steps, you permit the One-Touch Join service account to access the calendar of each room resource that you want to use for One-Touch Join, and then set the calendar to auto-accept invitations. We also recommend that you make the calendar available to all users in your domain in such a way that allows them to book meetings using the resource, without being able to view the details of any other meetings in the resource's calendar.

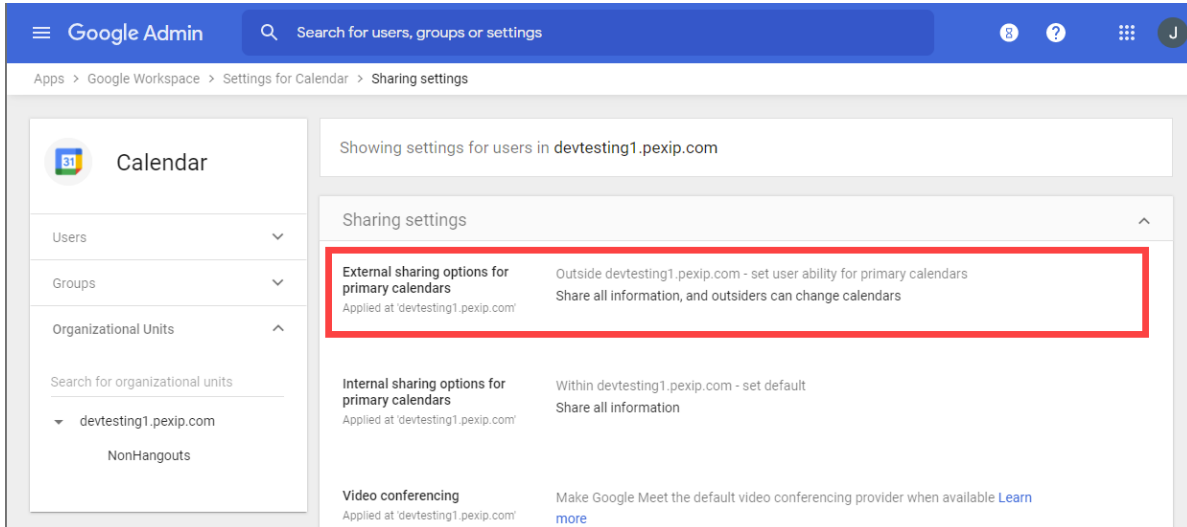
Sharing calendars externally

In this step, you configure Google Workspace to permit endpoint calendars within your domain to be shared externally. This permission is required because the service account uses the external `iam.gserviceaccount.com` domain and is therefore considered an "outsider". Granting this permission does not in itself allow any external accounts to access the calendars — each calendar must then be shared with the service account. For more information, see [Prerequisites](#).

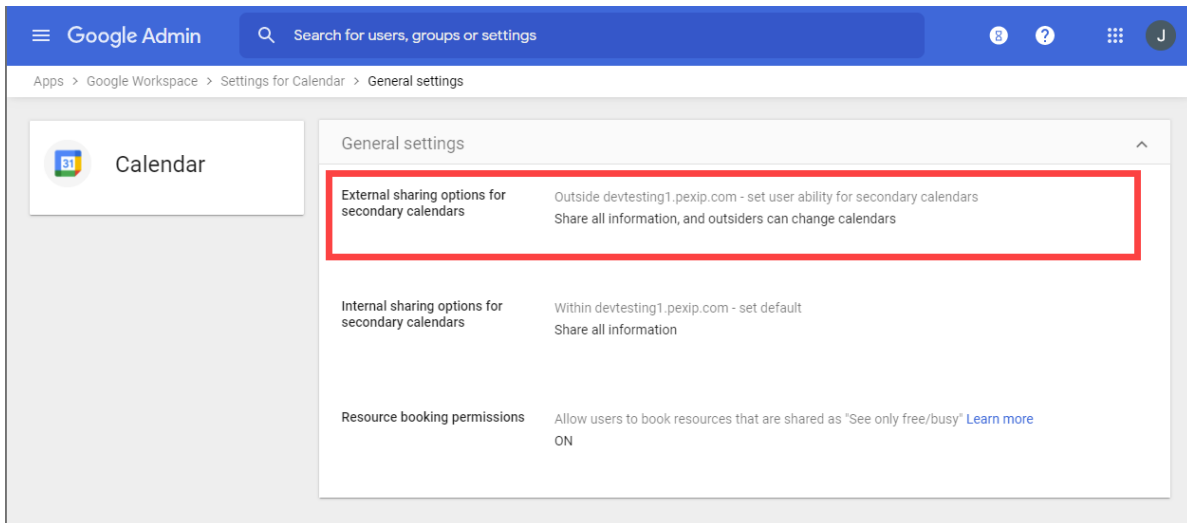
To enable calendars to be shared externally:

1. Go to <https://admin.google.com/> (logged in as a Google Workspace administrator) and select **Apps > Google Workspace > Calendar**.

- In the **Sharing settings** section, ensure that **External sharing options for primary calendars** is set to *Share all information, and outsiders can change calendars*:



- In the **General Settings** section, under **External sharing options for secondary calendars**, select *Share all information, and outsiders can change calendars*:



- i** Selecting these options to ... *and outsiders can change calendars* will enable users to use One-Touch Join to join all meetings, including private meetings. If you will not be using One-Touch Join with private meetings in your deployment, both these options can be set to ... *but outsiders cannot change calendars*.

Sharing individual calendars with the service account

Note that the Google calendar API limits the number of calendars that can be shared within a 24 hour period to 750 (for more information, see <https://support.google.com/a/answer/2905486?hl=en>). This means that if you have more than 750 room resources that you wish to use for One-Touch Join, they will need to be set up over a period of days.

For deployments with more than around 50 rooms, we have developed a Python script that can be used to share your room resource calendars with the service account, and create a CSV that can be used to import endpoint configuration to One-Touch Join. You must be familiar with Python in order to use this script; contact your Pexip authorized support representative for more information.

To share calendars with the service account:

1. Go to <https://calendar.google.com> (logged in as a Google Workspace administrator so that you have permission to share the calendars).
2. From the left-hand panel, select the + next to **Other calendars** and then select **Browse resources**.
3. Expand the sections if necessary, and tick the boxes of all the room resources whose calendars you want to share with the service account.

This will add the room resources to the **Settings for other calendars** section in the left-hand panel.

4. For each of the rooms:
 - a. From the **Settings for my calendars** section, select the room resource and then select **Share with specific people**.
 - b. Select **Add people**.
 - c. In the **Share with specific people** dialog, enter the email address of the One-Touch Join service account. Ensure the **Permissions** are set to either:
 - **Make changes to events** (if you want users to be able to use OTJ to join all meetings, including private meetings, from this endpoint)
 - **See all event details** (if you don't want to offer OTJ for private meetings on this endpoint).

If the option to **Make changes to events** is grayed out, then check that you have selected the options to **Share all information, and outsiders can change calendars** when [Sharing calendars externally](#).

- i** If your deployment includes personal endpoints that are associated with a user's personal calendar, then either you or the end user will need to ensure that their calendar allows the One-Touch Join service account to **Make changes to events** if they wish to use OTJ to join their own private meetings from their endpoint.

← Settings

General

Add calendar

Import & export

Settings for my calendars

- Alex Morson
- Birthdays
- calendar-reader@obtp-2471...
- Oslo-1-Demo Room (4)
- Ruscombe-1-Carl-desk (1)
- Ruscombe-1-desk-juliet_test...
- Ruscombe-1-Juliet EX...**

Calendar settings

Auto-accept invitations

Share with specific people

otj-calendar-reader@pexip-otj-270111.iam.gserviceaccount. Make changes to events

+ Add people

Learn more about sharing your calendar with someone

Event notifications

+ Add notification

All-day event notifications

+ Add notification

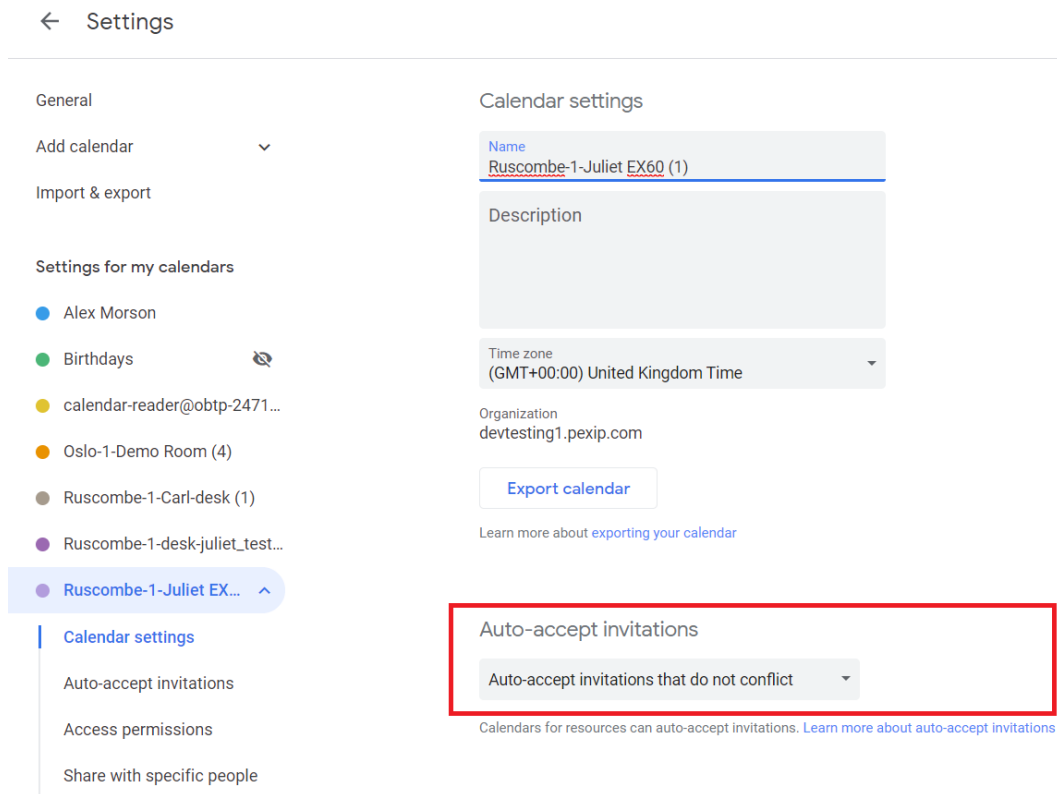
- i** For more information on sharing room and resource calendars in Google Workspace, see <https://support.google.com/a/answer/1034381>.

Auto-accepting invitations

By default, when creating room resources in Google Workspace, calendar processing is set to **Auto-accept invitations that do not conflict**. You must ensure you keep this setting for all room resources, so that the room will automatically accept meeting requests if it is available, and automatically decline an invitation if it is already booked.

To check this setting:

1. Go to <https://calendar.google.com> (logged in as a Google Workspace administrator so that you have permission to share the calendars).
2. From the left-hand panel, select the room resource and select **Settings and sharing**.
3. In the **Auto-accept invitations** section, ensure that **Auto-accept invitations that do not conflict** is selected:



Allowing users to book resources

We recommend that you configure your Google Workspace calendar settings to allow end users to book a room resource without seeing details of the room's other bookings. To do this, you configure the room resource's calendar so that all users in your domain have permission to see its free/busy status, without being able to see the invitation details. You then on a global basis permit users to book resources to which they have free/busy access.

To do this:

1. Go to <https://calendar.google.com> (logged in as a Google Workspace administrator so that you have permission to share the calendars).
2. From the left-hand panel, select the room resource and select **Settings and sharing**.
3. In the **Access permissions** section, select **Make available for <your domain>**, and ensure that **See only free/busy (hide details)** is selected:

Auto-accept invitations

Auto-accept invitations that do not conflict ▼

Calendars for resources can auto-accept invitations. [Learn more about auto-accept invitations](#)

Access permissions

☐ Make available to public

See all event details ▼

☒ Make available for devtesting1.pexip.com

See only free/busy (hide details) ▼

[Get shareable link](#)Learn more about [sharing your calendar](#)

Share with specific people



otj-calendar-reader@pexip-otj-270111.iam.gserviceaccount.co...

See all event details ▼

[+ Add people](#)Learn more about [sharing your calendar with someone](#)

4. Go to admin.google.com (logged in as a Google Workspace administrator).
5. From the left-hand menu, select **Apps** > **Google Workspace** > **Calendar**.
6. Scroll down to **General Settings** and select **Resource Booking Permissions**.
7. Ensure that **Allow users to book resources that are shared as See only free/busy** is set to **ON**:

The screenshot shows the Google Admin console interface. At the top is the Google Admin header with a search bar. Below it is a breadcrumb trail: Apps > Google Workspace > Settings for Calendar > General settings. On the left is a sidebar with the 'Calendar' icon and label. The main content area is titled 'General settings' and contains three sections: 'External sharing options for secondary calendars', 'Internal sharing options for secondary calendars', and 'Resource booking permissions'. The 'Resource booking permissions' section is highlighted with a red rectangular box. It contains the text 'Allow users to book resources that are shared as "See only free/busy" [Learn more](#)' and the status 'ON'.

Updating the per-user request quota

In this step you increase the limit on the number of queries per 100 seconds per user to the Google Calendar API.

The default number of queries per 100 seconds per user is 500. In this context, the "user" is the service account. In deployments with fewer than around 180 rooms, each room resource calendar is queried every 30 seconds by two conferencing nodes (both using the same service account), resulting in 5,760 queries per room per day. (In larger deployments, room resource calendars are queried less frequently.)

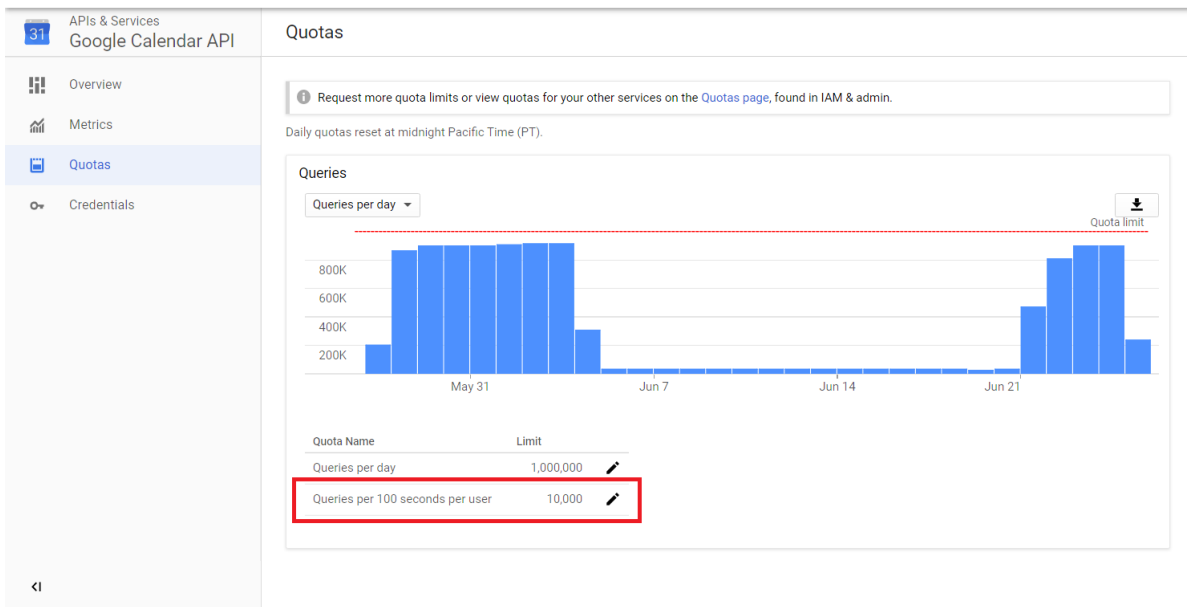
We recommend that you increase the number of queries per 100 seconds per user to 10,000 to provide sufficient processing overhead and room for expansion (there is currently no additional cost to this increase).

To increase this quota to 10,000:

1. Go to <https://console.developers.google.com> (logged in as a Google Workspace administrator).
2. From the top left of the page, select the project you created for One-Touch Join:



3. From the navigation menu at the top left of the page, select **IAM & Admin > Quotas**.
4. From the Quotas page, select **Edit Quotas** and then select **Google Calendar API - Queries per 100 seconds per user**. You will be taken to the **Google Calendar API > Quotas** page.
5. Change **Queries per 100 seconds per user** to **10,000**:



- i** You may also need to request an increase to the number of **Queries per day** for larger deployments - for more information, see [Requesting an increase to API limits](#).

Requesting an increase to API limits

This optional step applies to larger deployments only (more than around 170 room resources), and should be performed if you wish to reduce the amount of time taken for endpoints to be updated with additions or changes to their corresponding room resource calendar.

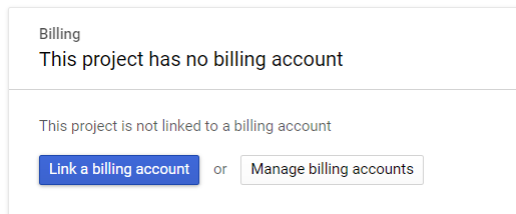
The maximum frequency with which an endpoint will be updated with meeting information is every 30 seconds. For deployments with more than around 170 endpoints, this frequency will decrease in line with the number of endpoints (up to around 20 minutes for deployments with around 6,000 endpoints). This is due to a limit on the number of Calendar API requests permitted by Google in a 24-hour period — for more information, see <https://developers.google.com/calendar/pricing>.

To reduce the time taken to update endpoints in these larger deployments, you can request an increase to the number of Calendar API requests One-Touch Join can make.

- i** When your request has been implemented by Google, you must then increase the [Maximum Google Workspace API requests](#) on Pexip Infinity in order to take advantage of the increase.

To request an increase to the API limits:

1. If you do not already have one, create a Cloud Billing Account (note that this is different from a Google Workspace billing account). Full instructions are available via https://cloud.google.com/billing/docs/how-to/manage-billing-account#create_a_new_billing_account.
2. Link the Cloud Billing Account to the project you created when [Creating a service account](#):
 - a. Go to <https://console.developers.google.com> (logged in as a Google Workspace administrator).
 - b. Ensure that the project shown in the top left corner is the one you created for One-Touch Join when [Creating a service account](#).
 - c. Select the burger menu from the top left of the page and select **Billing**. When the following message appears, select **Link a billing account**:



- d. Select the account to link to:

Set the billing account for project “Quickstart”

Billing account ?

There is only one billing account currently available to link this project to

My Billing Account

CANCEL

SET ACCOUNT

3. Request an increase to your quota:
 - a. From the navigation menu at the top left of the page, select **IAM & admin > Quotas**.
 - b. From the Quotas page, select **Edit Quotas** and then select **Google Calendar API**.
In the panel that appears on the right, enter the **New quota limit** that you wish to request, and in the **Request description** field, enter the reason for requesting the increase:

The screenshot shows the Google Cloud IAM & Admin console. On the left, the 'IAM & admin' menu is open, and 'Quotas' is selected. The main panel shows the 'Quotas' page for the 'Google Calendar API'. The 'Quota type' is 'All quotas', 'Service' is 'All services', 'Metric' is 'All metrics', and 'Location' is 'All locations'. The 'Google Calendar API' quota is selected, and the 'New quota limit' is being set to 100,000. The 'Request description' field is also visible.

c. Select **Submit request**.

Quota increase requests typically take two business days to process.

Adding a One-Touch Join Google Workspace integration on Pexip Infinity

In this step you configure Pexip Infinity with details of the Google Workspace deployment configured above, including details of the service account used to access calendars.

From the Pexip Infinity Administrator interface, go to **One-Touch Join > OTJ Google Workspace Integrations**.

Option	Description
Name	The name of this One-Touch JoinGoogle Workspace integration.
Description	An optional description of this One-Touch JoinGoogle Workspace integration.
Account email	<p>If you are authorizing using a service account, enter the email address of the service account that One-Touch Join will use to log in to Google Workspace.</p> <p>If you are authorizing using a Google Workspace domain user, enter the email address of the user.</p>
Enable user authorization	<p>If you are authorizing using a service account — the recommended method — this should be left blank.</p> <p>Select this option only if you will be authorizing using a Google Workspace domain user.</p>
Private key	<p>(Available when authorizing using a service account, i.e. user consent authorization has not been enabled)</p> <p>The private key used by One-Touch Join to authenticate the service account when logging in to Google Workspace. For instructions on how to obtain this, see Generating a key file.</p> <p>This must include all the text in the file between (and including) -----BEGIN PRIVATE KEY----- and -----END PRIVATE KEY-----</p>

Advanced options

Maximum Google Workspace API requests	<p>The maximum number of API requests that can be made by One-Touch Join to your Google Workspace Domain in a 24-hour period.</p> <p>We recommend you set this value to 90% of your total permitted requests. Google's default is 1,000,000 so by default this is set to 900,000 on Pexip Infinity. If you increase the number of API requests, you should also increase this setting to 90% of that number.</p> <p>For more information, see Frequency and limitations on calendar requests.</p>
Google OAuth 2.0 endpoint	The URI of the Google OAuth 2.0 endpoint.

Option	Description
Google authorization server	The URI of the Google authorization server.

Next steps

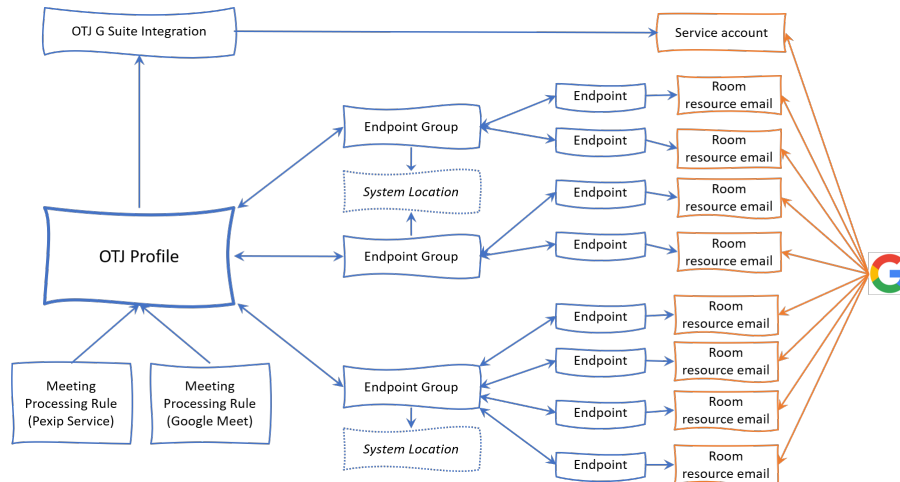
You must now configure the remainder of the One-Touch Join components on Pexip Infinity, as described in [Configuring Pexip Infinity for One-Touch Join](#).

Configuring Pexip Infinity for One-Touch Join

This topic describes how to configure Pexip Infinity when enabling the One-Touch Join feature. It covers configuration of the various Pexip Infinity components, each described in detail in the sections that follow:

1. [Adding a One-Touch Join profile](#)
2. [Adding One-Touch Join endpoint groups](#)
3. [Adding One-Touch Join endpoints](#)
4. [Adding One-Touch Join meeting processing rules](#)

The diagram below shows (in blue) the components that are configured on Pexip Infinity and how they are related to each other. It also shows (in orange) how the Pexip Infinity components are associated with your calendar/email service — in this example we have used Google Workspace, with support for Google Meet and Pexip Service meeting types:



Prerequisites

Before you start configuring Pexip Infinity, you must first do one of the following, depending on your calendar/email service:

- [Configure Google Workspace for One-Touch Join](#), including [Adding a One-Touch Join Google Workspace integration on Pexip Infinity](#), or
- [Configure Exchange on-premises for One-Touch Join](#), including [Adding a One-Touch Join Exchange integration on Pexip Infinity](#)
- [Configure Office 365 for One-Touch Join](#), including [Adding a One-Touch Join Graph integration on Pexip Infinity](#).

i Existing customers may have previously implemented One-Touch Join in a Microsoft Office 365 environment using a service account authenticated using OAuth and the EWS API. The EWS API is being deprecated by Microsoft, so we do not recommend its use for new deployments; for existing customers, these deployments are described in [Adding a One-Touch Join Exchange integration on Pexip Infinity](#).

Adding a One-Touch Join profile

In this step you create a profile that you will use to link together all the components for this particular deployment: the Exchange or Google Workspace integration, the endpoint groups (and therefore endpoints), and the rules to be used to process meeting invitations.

A single Pexip Infinity One-Touch Join profile is associated with a single integration type — Exchange, Google Workspace, or O365 Graph. However, a One-Touch Join profile can contain a mixture of Cisco and Poly endpoints.

An endpoint group, and therefore an endpoint (and its room resource), can belong to only one One-Touch Join profile. If you do not assign an endpoint group to a One-Touch Join profile, the endpoints in that group will not be used for One-Touch Join.

To add a One-Touch Join profile, from the Pexip Infinity Administrator interface, go to **One-Touch Join > OTJ Profiles**.

Option	Description
Name	The name of this One-Touch Join profile.
Description	An optional description of this One-Touch Join profile.
No. of upcoming days	The number of days of upcoming One-Touch Join meetings to be shown on endpoints. This will also be the number of days of future meetings shown on the One-Touch Join Status page.
Enable non-video meetings	<p>Enabled: If One-Touch Join has not been able to obtain a video address from the meeting, then the meeting will still appear on the endpoint as a scheduled meeting, showing the information that was able to be parsed, but the Join button will not appear.</p> <p>Disabled: If there is no video address, the available meeting information will not appear on the endpoint. Note that the meeting will still exist in the room resource's calendar, so conflicting meetings cannot occur.</p>
Enable private meetings	<p>Determines whether or not meetings flagged as private are processed by the One-Touch Join service.</p> <p>Enabled: Private meetings will be processed in the same way as any other meeting.</p> <p>Disabled: Private meetings are not processed by One-Touch Join, and therefore the meeting information will not appear on the endpoint. Note that the meeting will still exist in the room resource's calendar, so conflicting meetings cannot occur.</p> <p>Note that if this is set to Enabled, you can still prevent private meeting details from being displayed on individual Poly endpoints by disabling the endpoint's Show Private Meeting Information setting.</p>
Process alias for private meetings	<p>(Applies if Enable private meetings has been selected)</p> <p>Enabled: For private meetings, the meeting alias will be extracted from the invitation in the usual way.</p> <p>Disabled: For private meetings, the available meeting information — apart from the meeting alias — will appear on the endpoint and therefore the Join button will be disabled.</p>
Replace subject	<p>Always: For all meetings, the endpoint will display either the text in the Replace subject string (if present) or the organizer's name, in place of the meeting subject.</p> <p>Never: For all meetings, including private meetings, the endpoint will display the meeting subject in the usual way.</p> <p>Private meetings only: For private meetings, the endpoint will display either the text in the Replace subject string (if present) or the organizer's name, in place of the meeting subject. For all other meetings, the endpoint will display the meeting subject in the usual way.</p> <p>For more information and examples, see Hiding or changing the meeting subject.</p>
Replace subject string	<p>(Applies if Replace subject is set to Always or Private meetings only)</p> <p>A Jinja2 snippet that defines how the subject should be replaced (when this has been enabled). If this field is left blank, the subject will be replaced with the name of the organizer.</p> <p>For more information and examples, see Hiding or changing the meeting subject.</p>
Replace empty subject	<p>Enabled: For meetings that do not have a subject, the endpoint will display the organizer's name in place of the subject.</p> <p>Disabled: For meetings that do not have a subject, the endpoint will display a blank field in place of the subject.</p>

Option	Description
Exchange integration	<p>(Applies if this OTJ profile is for an Exchange on premises integration, or an Office 365 integration that uses the EWS API)</p> <p>The Exchange integration used by this One-Touch Join profile.</p> <p>You should already have created this as part of either Configuring Exchange on-premises for One-Touch Join or Configuring Office 365 using EWS for One-Touch Join, but you can configure it now by selecting the green plus symbol + to the right of the field.</p>
Google Workspace integration	<p>(Applies if this OTJ profile is for a Google Workspace integration)</p> <p>The Google Workspace integration used by this One-Touch Join profile.</p> <p>You should have already created this as part of Configuring Google Workspace for One-Touch Join, but you can configure it now by selecting the green plus symbol + to the right of the field.</p>
O365 Graph integration	<p>(Applies if this OTJ profile is for an Office 365 integration that uses the Graph API)</p> <p>The Exchange integration used by this One-Touch Join profile.</p> <p>You should already have created this as part of Configuring Office 365 using Graph for One-Touch Join, but you can configure it now by selecting the green plus symbol + to the right of the field.</p>
Endpoint Groups	The Endpoint Groups used by this One-Touch Join profile.
Cisco OBTP endpoint configuration	
Start buffer	<p>The number of minutes before a meeting's scheduled start time that the "Join" button on the endpoint will become enabled for that meeting.</p> <p>An endpoint can offer more than one "Join" button if there is an overlap between different meetings' start and end buffers.</p>
End buffer	The number of minutes after a meeting's scheduled end time that the "Join" button on the endpoint will become disabled for that meeting.
Default API username	<p>The user name and password used by One-Touch Join to access a Cisco OBTP endpoint's API. The API is used by the Conferencing Node to configure the endpoint with meeting information. The account being used must have a role of either <i>User</i> or <i>Admin</i>.</p> <p>The Default API username and password is only used if the configuration for the Cisco OBTP endpoint in within One-Touch Join does not include an API username and password. A default is offered because some deployments will have the same username and password for all endpoints.</p>
Default API password	
Verify endpoint certificates by default	<p>Whether or not to verify the TLS certificate of a Cisco OBTP endpoint by default when accessing its API. Can be overridden per endpoint using the endpoint's Verify endpoint API TLS certificate setting.</p> <p>For more information, see Managing trusted CA certificates.</p>
Use HTTPS for endpoint API	<p>Whether or not to use HTTPS by default when accessing a Cisco OBTP endpoint's API. Can be overridden per endpoint using the endpoint's Use HTTPS setting.</p> <p>Enabled: Use HTTPS to access an endpoint's API.</p> <p>Disabled: Use HTTP to access an endpoint's API.</p>
Cisco Webex Cloud configuration	
Enable Webex Cloud	Select this option to if you have endpoints that are registered to Webex or Webex Edge for devices, and you want to enable them to use One-Touch Join. Note that you must first create a Webex Integration .
Client ID	The Client ID that was generated when you created a Webex Integration .

Option	Description
Client secret	The Client Secret that was generated when you created a Webex Integration .
Redirect URI	The Redirect URI you entered when you created a Webex Integration . This must point to the IP address or FQDN of the Management Node, and be in the format <code>https://<Management Node Address>/admin/platform/mjxintegration/oauth_redirect/</code>

Hiding or changing the meeting subject

In some cases, you may not want the subject of upcoming meetings to be displayed on an endpoint. One-Touch Join allows you to [replace the subject](#) for either all meetings, or only those meetings flagged as private. When replacing the subject, you can elect to replace it with the name of the meeting organizer, or you can [use a Jinja2 snippet](#) to re-write the meeting subject. Note that this method does not affect the meeting invitation, just what is displayed on the endpoint for that meeting.

Where a meeting subject has been changed or hidden, the original subject is not shown on the Management Node Administrator interface — it will only display the altered subject.

The Jinja2 snippet used in the **Replace subject string** has the same access to `calendar_event` information as the [Custom meeting type](#), including the meeting subject and organizer's name. For more information on using Jinja2 with Pexip Infinity, see [Jinja2 templates and filters](#).

i Hiding meeting subjects on endpoints is done on a per-profile basis. Therefore, if you want to hide the subject for just some of the endpoints in your deployment, you should create a separate profile for these endpoints with **Replace subject** enabled, and then add the endpoints to an Endpoint Group that you associate with that profile.

Example: Add text after organizer's name for private meetings

In this example we want to replace the meeting subject for all private meetings with the name of the organizer followed by the text 's meeting

To do this, we set **Replace subject** to *Private meetings only*, and enter the following Jinja2 snippet in the **Replace subject string** field:

```
{{ calendar_event.organizer_full_name }}'s meeting
```

Example: add text to meetings with an external organizer

In this example, we want to check the email address of the meeting organizer and, if the organizer is not from our own `example.com` domain, prepend the subject of the meeting with the text **External**:

To do this, we set **Replace subject** to *Always*, and enter the following Jinja2 snippet in the **Replace subject string** field:

```
{% set domain = pex_regex_search("@([a-z0-9.-]+.com)", calendar_event.organizer_email) %}
{% if domain[0] != "example.com" %}
  External: {{ calendar_event.subject }}
{% endif %}
```

Adding One-Touch Join endpoint groups

In this step you create endpoint groups, and optionally add endpoints to each group. Each endpoint can belong to only one endpoint group; an endpoint group can contain a mix of Cisco OBTP and Poly OTD endpoints. In general, we recommend that all endpoints in the same physical location are assigned to one endpoint group.

Each endpoint group is associated with a system location; if there are more than 5 Conferencing Node in one location, only 5 will be actively running One-Touch Join. This is because each Conferencing Node will be connecting to Exchange, and the messaging overhead needs to be limited.

From the Pexip Infinity Administrator interface, go to **One-Touch Join > OTJ Endpoint Groups**.

Option	Description
Name	The name of this One-Touch Join endpoint group.
Description	An optional description of this One-Touch Join endpoint group.
System location	The system location of the Conferencing Nodes which will provide One-Touch Join services for this endpoint group.
OTJ profile	The One-Touch Join profile to which this endpoint group belongs.
Disable web proxy	Select this option to bypass the web proxy (where configured for this system location) when sending requests to Cisco OBTP endpoints in this One-Touch Join endpoint group.
Endpoints	The endpoints that belong to this One-Touch Join endpoint group.

Adding One-Touch Join endpoints

In this step you add details of the endpoints that will be used for One-Touch Join, and the room resource that each endpoint is associated with. You can [add endpoints individually](#), or [in bulk using a CSV import](#).

After you have added details of your One-Touch Join endpoints to Pexip Infinity, you will also need to [configure the settings on each endpoint](#) to support One-Touch Join. We recommend that you do this after you have completed the following configuration.

If there are multiple endpoints in a single room, you should associate each endpoint with the same room resource, so that each endpoint will receive the same meeting details.

Adding endpoints individually

From the Pexip Infinity Administrator interface, go to **One-Touch Join > OTJ Endpoints**.

Option	Description
Endpoint name	The name of this One-Touch Join endpoint.
Description	An optional description of this One-Touch Join endpoint.
Endpoint type	<p>The type of "click to join" feature supported by this endpoint.</p> <p>Cisco OBTP: an endpoint that supports Cisco's One Button to Push (OBTP) and is located on the same network as the OTJ Conferencing Nodes. You should ensure that this endpoint has already been set up in accordance with Configuring OBTP endpoints on the same network.</p> <p>Webex Cloud registered: an endpoint that supports Cisco's One Button to Push (OBTP) and is located on a different network as the OTJ Conferencing Nodes. The endpoint must be registered to Webex or Webex Edge for Devices. You must also complete the steps described in Configuring Cisco Webex Cloud registered endpoints.</p> <p>Poly OTD: an endpoint that supports Poly's One Touch Dial (OTD). You must complete the steps in this Adding One-Touch Join Endpoints section before you set up your Poly endpoints in accordance with Configuring Poly OTD endpoints for OTJ.</p>

Configuration options for Cisco OBTP endpoints

Endpoint address	The IP address or FQDN of the endpoint's API.
Endpoint API port	The port of the endpoint's API. Default: 443 if HTTPS is used, otherwise 80 for HTTP.

Option	Description
Endpoint API username	<p>The user name and password used by One-Touch Join to access a Cisco OBTP endpoint's API. The API is used by the Conferencing Node to configure the endpoint with meeting information. The account being used must have a role of either <i>User</i> or <i>Admin</i>.</p> <p>Either both these fields must be configured, or both these fields must be left blank.</p> <p>If both these fields are left blank, the One-Touch Join profile's Default API username and password will be used.</p>
Endpoint API password	
Verify endpoint API TLS certificate	
Use HTTPS	<p>Whether to enable TLS verification when accessing this endpoint's API. Only applicable if using HTTPS to access this endpoint's API.</p> <p><i>Use OTJ profile default:</i> Use the Verify endpoint certificates by default setting configured for the One-Touch Join profile that this endpoint is associated with.</p> <p><i>On:</i> Enable TLS verification.</p> <p><i>Off:</i> Do not use TLS verification.</p> <p>For more information, see Managing trusted CA certificates.</p>
Configuration options for Poly OTD endpoints	
Poly Calendaring Username	<p>The username the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.</p> <p>This must be the same as the User Name or User (the field name will vary) configured on the Poly endpoint, and must be unique.</p> <p>This field is case-sensitive.</p>
Poly Calendaring password	<p>The password the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.</p> <p>This must be the same as the Password configured on the Poly endpoint.</p> <p>This field is case-sensitive.</p>
Configuration options for Webex Cloud registered endpoints	
Webex Device ID	<p>The Webex endpoint's unique identifier. You can find the IDs for all devices in your Webex deployment by going to https://developer.webex.com/docs/api/v1/devices/list-devices and from the right-hand panel selecting Run.</p>
Configuration options for all endpoints	

Option	Description
Raise alarms	<p>When enabled, an alarm will be raised:</p> <ul style="list-style-type: none"> for Poly endpoints: when this endpoint has not contacted the calendaring service on the Conferencing Node in the last 10 minutes for Cisco OBTP endpoints: when an attempt to push calendar updates to the endpoint was unsuccessful for Webex Cloud registered endpoints: when an attempt to push calendar updates from the Conferencing Node to Webex Cloud, or from Webex Cloud to the endpoint, was unsuccessful. <p>Default: enabled.</p>
Room resource email	<p>The email address of the room resource associated with this endpoint. This must match an email address that has been configured in Exchange or Google Workspace.</p> <p>For Poly endpoints, this must be the same as the Email or Mailbox (where this setting is available) configured on the Poly endpoint.</p>
Endpoint Group	The Endpoint Group to which this endpoint belongs.

Adding OTJ endpoints in bulk

You can add multiple One-Touch Join endpoints by importing a CSV file.

When formatting your import file:

- A header row in the CSV file is optional. If included, it must use the same field names as shown in the following sections, but you may change the order of the fields. If a header row is not used, fields must be in the same order as shown.
- All non-blank fields must contain valid data.
- If non-ASCII characters are used, the file must be encoded as UTF-8 text.
- All fields are case-sensitive.
- Values may optionally be enclosed in double quotation marks; any strings containing commas must be enclosed in double quotation marks e.g. "description for x, y and z".


Note that you can perform an export of existing data to produce an example file in the correct format.

To add multiple endpoints by importing a CSV file:


- Create the CSV file, using the following format:

```
name,description,endpoint_type,api_address,api_port,api_username,api_password,poly_username,poly_password,poly_raise_alarms_for_this_endpoint,webex_device_id,verify_cert,use_https,room_resource_email,mjx_endpoint_group_name
```

where

Field name	Content	Required field for...
name	 This field cannot be blank.	Cisco
	The name of this One-Touch Join endpoint.	Poly
	You should ensure there are no duplicate names, either within the CSV file, or between the CSV file and the existing endpoints (unless you wish the existing configuration to be overwritten).	Webex
description	An optional description of this One-Touch Join endpoint.	

Field name	Content	Required field for...
endpoint_type	<p>The type of "click to join" feature supported by this endpoint.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◦ <i>CISCO</i> ◦ <i>POLY</i> ◦ <i>WEBEX</i> 	<p>Cisco</p> <p>Poly</p> <p>Webex</p>
api_address	The IP address or FQDN of the Cisco OBTP endpoint's API.	Cisco
api_port	<p>The port of the Cisco OBTP endpoint's API.</p> <p>If this is left blank, the defaults (443 if HTTPS is used, otherwise 80 for HTTP) will be used.</p>	
api_username	The username used by OTJ to access the Cisco OBTP endpoint's API.	
api_password	The password used by OTJ to access the Cisco OBTP endpoint's API.	
poly_username	The username the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	Poly
poly_password	The password the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	Poly
poly_raise_alarms_for_this_endpoint	<p>Whether to raise an alarm if OTJ is unable to provide this endpoint with meeting information. (Note that despite the field name, this applies to all endpoint types.)</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◦ <i>TRUE</i> ◦ <i>YES</i> ◦ <i>FALSE</i> ◦ <i>NO</i> <p>If this is left blank, the default TRUE (enabled) will be used.</p>	
webex_device_id	The Webex endpoint's unique identifier.	Webex
verify_cert	<p>Whether to enable TLS verification when accessing the Cisco OBTP endpoint's API. Only applicable if using HTTPS to access this endpoint's API.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◦ <i>GLOBAL</i>: Use the Verify endpoint certificates by default setting configured for the One-Touch Join profile that this endpoint is associated with. ◦ <i>YES</i>: Enable TLS verification. ◦ <i>NO</i>: Do not use TLS verification. 	
use_https	<p>Whether to use HTTPS to access the Cisco OBTP endpoint's API.</p> <ul style="list-style-type: none"> ◦ <i>GLOBAL</i>: Use the Use HTTPS for endpoint API setting configured for the One-Touch Join profile that this endpoint is associated with. ◦ <i>YES</i>: Use HTTPS to access this endpoint's API. ◦ <i>NO</i>: Use HTTP to access this endpoint's API. 	

Field name	Content	Required field for...
room_resource_email	<p> This field cannot be blank.</p> <p>The email address of the room resource associated with this endpoint. This must match the email address that has been configured in Exchange or Google Workspace.</p>	<p>Cisco</p> <p>Poly</p> <p>Webex</p>
mjx_endpoint_group_name	<p>The endpoint group to which this endpoint belongs.</p> <p>If this field is set, it must contain the name of an existing endpoint group.</p>	

- From the Pexip Infinity Administrator interface, go to **One-Touch Join > OTJ Endpoints** and from the bottom right of the screen, select **Import**.
- From the **Import OTJ Endpoint Configuration** page, select **Choose file** and then navigate to the CSV file you have created.
- Select **Save**.

The imported endpoints will be added to your One-Touch Join configuration.

Duplicates

If any records in the CSV file have the same **name** field (regardless of whether or not any of the other fields are different), only one endpoint with that name will be created. This endpoint will use the last record that was imported.

If any records in the CSV file have the same **name** as an existing endpoint, the existing configuration will be overwritten by the imported endpoint's configuration.

Adding One-Touch Join meeting processing rules

In this step you create a prioritized set of rules that specifies each of the meeting types you expect users in your deployment to encounter, and how the invitations for these meetings should be processed in order to obtain the alias that the endpoint must dial in order to join the meeting.

One-Touch Join supports meetings from a number of different providers. For each of these supported meeting types, One-Touch Join knows what information to look for in the meeting invitation, and how to use what it finds to derive an alias that the endpoint can dial in order to join that meeting. In most cases, you can simply use the default processing for each supported meeting type. However, you also have the option to override the default processing with your own transform pattern to change how the alias is constructed. You can also write your own [regex](#) and [custom](#) rules if you wish to enable One-Touch Join for other meeting types or conferencing providers not currently supported.

A single One-Touch Join profile will normally have multiple meeting processing rules associated with it — we recommend that you create one rule for each [Meeting type](#) you expect users in your environment to encounter, including any invitations received from external contacts where users may wish to use an internal meeting room to join the meeting. The **Priority** option should be used to ensure that all rules for supported meeting types are processed before any **Domain**, **Regex** or **Custom** rules. (Note that the order in which the supported meeting types are prioritized between themselves is not important.)

When One-Touch Join processes a meeting invitation, it goes through each meeting rule in order of priority to find a match.

- If a match is found, it uses the information in the invitation, processed in accordance with the rule's settings, to derive an alias to use to join the meeting.
- If none of the meeting processing rules match (or there are no meeting processing rules configured or enabled), One-Touch Join will search the invitation for a URI or address with a **sip:**, **sips:** or **h323:** prefix, and use that as the alias.

One-Touch Join then provides the endpoint with the alias, along with other meeting information such as the start time, end time, subject, and organizer's name.

If no alias has been obtained, One-Touch Join may still provide the meeting information to the endpoint, depending on the [Enable non-video meetings](#) and [Enable private meetings](#) settings for the profile being used.

Each meeting processing rule is associated with a single One-Touch Join profile, and therefore will apply to either an Exchange integration or a Google Workspace integration, but not both.

To view, edit and create meeting processing rules, from the Pexip Infinity Administrator interface, go to **One-Touch Join > OTJ Meeting Processing Rules**.

Option	Description
Name	The name of this One-Touch Join meeting processing rule.
Description	An optional description of this meeting processing rule.
OTJ profile	The One-Touch Join profile associated with this meeting processing rule.
Priority	<p>The priority of this rule. Rules are checked in ascending priority order (starting at 1) until the first matching rule is found, and it is then applied.</p> <p>We recommend that meeting types other than <i>Domain</i>, <i>Regex</i> or <i>Custom</i> are given highest priority. You can then use lower <i>Priority</i> options to determine the order in which any <i>Domain</i>, <i>Regex</i> and <i>Custom</i> rules are applied, particularly if you are using more than one of these meeting types.</p>
Meeting type	<p>The type of meeting invitation to which this rule applies. You can select one of the supported meeting types from the drop-down list, or select <i>Regex</i> or <i>Custom</i> if you wish to define your own meeting processing rule.</p> <p>For a full list of available meeting types, and guidance on which to use in your deployment, particularly when joining Teams or Skype for Business meetings, see Supported meeting types.</p>
Include password	<p>(Available when a Meeting type of <i>Zoom</i> has been selected)</p> <p>Enable this option to search the meeting information for the meeting password, and if found, include the password to the alias used to join the meeting, so that users do not need to enter the password themselves.</p>
Default processing enabled	<p>(Does not apply to <i>Custom</i> meeting types)</p> <ul style="list-style-type: none"> For meeting types other than <i>Regex</i>: <ul style="list-style-type: none"> check this box to use the default transform pattern for the selected meeting type (for a list of the default transform patterns for each meeting type, see Supported meeting types), or clear this box to write your own Transform pattern for this meeting type. For <i>Regex</i> meeting type: <ul style="list-style-type: none"> check this box to use the matched string, unchanged, as the alias that the endpoint will dial to join the meeting, or clear this box to use a regex Replace string to transform the matched string into the alias to dial. <p>For more information, see Regex meeting type.</p>
Transform pattern	<p>(Available and required when Default processing is disabled and any Meeting type option other than <i>Custom</i> or <i>Regex</i> has been selected.)</p> <p>A Jinja2 snippet that is used to process the meeting information from calendar events of the selected Meeting type in order to derive the meeting alias.</p> <p>If you disable Default processing after creating and saving the rule, this field will show the default transform pattern, which you can then edit.</p> <p>For a list of the valid variables for each meeting type, see Supported meeting types.</p>
Match string	<p>(Available and required when a Meeting type of <i>Regex</i> has been selected.)</p> <p>The regular expression that defines the string to search for in the invitation.</p>

Option	Description
Replace string	<p>(Available and required when Default processing is disabled and a Meeting type of <i>Regex</i> has been selected.)</p> <p>A regular expression that defines how to transform the matched string into the alias to dial.</p>
Domain	<p>(Available and required when a Meeting type of <i>Domain</i>, <i>Microsoft Teams Meeting Properties</i> or <i>Google Meet SIP Guest Join</i> has been selected.)</p> <ul style="list-style-type: none"> For a Meeting type of <i>Domain</i>, this is the domain that OTJ will search for in the meeting body, in order to match this rule. For a Meeting type of <i>Microsoft Teams Meeting Properties</i>, this is the domain that OTJ will append to the meeting ID after the rule has been matched, in order to create the alias that the endpoint will dial to join the meeting. For a Meeting type of <i>Google Meet SIP Guest Join</i>, this is the domain that OTJ will append to the meeting ID after the rule has been matched, in order to create the alias that the endpoint will dial to join the meeting. In this case it should be the domain of the service providing the Pexip Google interop — for the Pexip Service this is <code>google.pexip.me</code>.
Custom template	<p>(Available and required when a Meeting type of <i>Custom</i> has been selected.)</p> <p>A Jinja2 script which is used to process the meeting information from calendar events in order to extract the meeting alias.</p> <p>For more information, see Custom meeting type.</p>
Enabled	<p>Determines whether or not the rule is enabled. Any disabled rules still appear in the rules list but are ignored. Use this setting to test configuration changes, or to temporarily disable specific rules.</p>


Testing the rule

When you have created and saved a meeting processing rule, a **Test OTJ Meeting Processing Rule** button will appear at the bottom of the page. This will take you to the **Test Meeting Processing** page, which allows you to test that the rule works as expected for the selected deployment and meeting type, and also allows you to edit the configuration for that rule until you get the desired results.

When searching a meeting invitation for the text to transform into an alias, OTJ will search either the invitation's properties, or the invitation's body (depending on the selected **Meeting type**) — and so when testing a rule, you will see either a **Calendar event properties** field or a **Calendar event body** field as appropriate. These fields will in most cases contain some example text in the format expected by OTJ, but you can enter other text here to help you test the rule, for example if you know that the format will be different in your deployment. However, since these two fields are there purely to assist you when testing the rule, and do not make up part of the rule itself, any changes to these fields will not be saved.

To test the rule:

1. Review and complete the following fields:

Option	Description
Read-only fields	
Integration type	This read-only field shows whether the rule will be applied to a Google Workspace or Exchange integration. This is based on the integration option selected in the OTJ profile associated with the rule.
Meeting type	This read-only field shows the meeting type associated with this rule.
Configuration that can be edited and saved	
<p> The available fields will depend on the selected meeting type.</p> <p>You can edit these fields and re-test the rule until you get the desired results.</p>	
Domain	The Domain currently configured for this rule.
Match string	The Match string (and Replace string, where applicable) currently configured for this rule.
Replace string	
Transform pattern	<p>The pattern that will be used to transform specific text in the meeting invitation into an alias to dial.</p> <ul style="list-style-type: none"> ◦ If you selected Default processing enabled, this will be the default transform pattern for this meeting type. ◦ If you did not select Default processing enabled, this will be the Transform pattern you entered.
Custom template	The Custom template currently configured for this rule.
Example text used when testing the rule	
Calendar event properties	<p>(Available for some meeting types)</p> <p>A JSON field representing the event properties that OTJ expects to find for the selected Meeting type (for Google Workspace integrations, this will contain a subset of the Google Event Properties; for Exchange integrations, this will be the Exchange MAPI Properties). This data will be used to generate the meeting alias.</p> <p>In most cases this field will be populated automatically, but you can edit it if you know that the format used in your deployment will be different.</p>
Calendar event body	<p>(Available for some meeting types)</p> <p>An example of the text that OTJ expects to find in the body of the invitation for the selected Meeting type, and which will be used to generate the meeting alias. In most cases this will be populated automatically, but you can paste in the full text from an actual meeting invitation used in your deployment and test the rule against this.</p>

2. Select **Test OTJ Meeting Processing Rule**.

The **Result** field shows the meeting alias that would be extracted based on the rule's current configuration and the example calendar event properties or body.

- If this is blank, the example calendar event properties / body did not contain any text that could be matched and transformed according to the rule as currently configured.
- If the result is not as expected, edit the fields above as appropriate.

3. When the configuration is producing the desired result, to save the changes you have made, select **Save changes and return**.

Next steps

You should now complete the steps in [Configuring endpoints to support One-Touch Join](#) for each endpoint.

Configuring endpoints to support One-Touch Join

This topic describes how to configure each of the supported endpoint types — [Cisco OBTP](#) (either when on the same network as the One-Touch Join Conferencing Nodes, or when on a different network), or [Poly OTD](#) — so they can be used with Pexip Infinity One-Touch Join.

Prerequisites

We recommend that you have already completed the steps in [Configuring Pexip Infinity for One-Touch Join](#). In particular, you will need some of the information that you previously entered when [Adding One-Touch Join endpoints](#) to Pexip Infinity, in order to complete the configuration on each endpoint.

Configuring Cisco OBTP endpoints for OTJ

For endpoints on the same network as the One-Touch Join Conferencing Node, we recommend you configure the endpoint to allow the Conferencing Node to connect directly to it, as per the instructions in [Configuring OBTP endpoints on the same network](#).

For endpoints on a different network to the One-Touch Join Conferencing Node, these endpoints must be registered to Webex Cloud, and One-Touch Join must be configured to connect to the endpoint via Webex Cloud. For full instructions, see [Configuring Cisco Webex Cloud registered endpoints](#).

Configuring OBTP endpoints on the same network

In order for Cisco OBTP endpoints to obtain One-Touch Join meeting information, the Conferencing Node associated with the endpoint uses the endpoint's API to push the information out to the endpoint.

The endpoint must have an account set up with a role of either *Admin* or *User* that can be used by One-Touch Join to access the endpoint's API.

Configuring Cisco Webex Cloud registered endpoints

In order for endpoints registered to Cisco Webex Cloud to obtain One-Touch Join meeting information, the Conferencing Node connects to the Webex Cloud, which then uses the endpoint's API to push the meeting information to the endpoint. To enable this, you must set up a Webex integration to be used for OTJ, and then configure Pexip Infinity OTJ with details of the integration.


Prerequisites

Webex Cloud registered endpoints must be:

- registered to either Webex or Webex Edge for Devices
- running one of the following software versions:
 - CE (9.14 or later)
 - TC (any version supported by Webex Cloud)
 - RoomOS
- have Cloud Calendar disabled
- be running in Room mode (not Personal mode).

Creating a Webex Integration

In this step, you create a new Webex integration to be used for One-Touch Join.

1. Go to <https://developer.webex.com/my-apps/new/integration> and sign in with your account.
2. Configure the following fields as follows (all other fields can be configured as appropriate for your environment):
 - **Redirect URI(s)**: this should point to the IP address or FQDN of your Management Node, and be in the format `https://<Management Node_address>/admin/platform/mjxintegration/oauth_redirect/`
 -  The **OAuth Redirect URI** is the page on the Pexip Infinity Administrator interface to which the administrator will be returned after they have successfully signed in to the Webex Integration. Because it is a page on the Management Node, this URI is internal to your deployment and only needs to be accessible from the administrator's web browser; you do not need to make it externally accessible.
 - **Scopes**: select `spark:xapi_commands`
3. Select **Add Integration**.
Your integration will be created.
4. Take note of the **Client ID** and **Client Secret** (which will be generated for you after the integration has been created) and the **Redirect URI** (which you entered in step 2 above). These will be required when you are [Enabling a One-Touch Join profile to use Webex Cloud](#).

Enabling a One-Touch Join profile to use Webex Cloud

1. On the Management Node, go to **One-Touch Join > OTJ Profiles** and either select an existing profile, or create a new profile.
2. In the **Cisco Webex Endpoint Config Options** section, select **Enable Webex Cloud** and in the fields that then appear, enter the **Client ID**, **Client Secret** and **Redirect URI** from the previous steps.
3. Select **Save**.
You will be taken back to the main **OTJ Profiles** page.
4. Select the profile you have just edited/created, and at the bottom of the page select **Authorize Webex Cloud API access**.
5. From the **Authorize Login to Webex Integration** page, select **Authorize**:

]pexip[Infinity Conferencing Platform

Status ▾ History & Logs ▾ System ▾ Platform ▾ Call Control ▾ Services ▾ Users & Devices ▾ **One-Touch Join ▾** Utilities ▾

Authorize Login to Webex Integration

Webex Username

Authorize link

Authorize

https://webexapis.com/v1/authorize?scope=spark%3Axapi_commands&state=oauth_state%3D02a

6. Sign in to Webex, and accept the permissions being requested.
You are redirected back to the Management Node.
7. You may be asked to sign in to the Management Node again. If so, you must sign in to the Management Node (using your Management Node credentials) to complete the authorization process.

When complete, you see the message **Successful sign in to Webex Cloud**.

Adding a Webex endpoint

1. On the Management Node, go to **One-Touch Join > OTJ Endpoints** and select **Add OTJ Endpoint**.
2. Select an **Endpoint type** of *Webex Cloud registered*.
3. Enter the **Webex Device ID**.
You can find the IDs for all devices in your Webex deployment by going to <https://developer.webex.com/docs/api/v1/devices/list-devices> and from the right-hand panel selecting **Run**.
4. Enter the **Room resource email** associated with this endpoint.

Disabling the calendar

1. Sign in to <https://admin.webex.com/>
2. From the panel on the left, select **Workspaces**.
3. Ensure that for each workspace containing an endpoint to be used for OTJ, the **Calendar** column shows *Not configured*.

- If a calendar is listed, select **Actions > Edit Calendar** and from the **Calendar** drop-down menu select *Off*:

Add Calendar ✕

Calendar service enables One Button To Push for the Workspace.

Calendar

Off

Off

Microsoft Exchange/Office 365

Google Calendar

Cancel

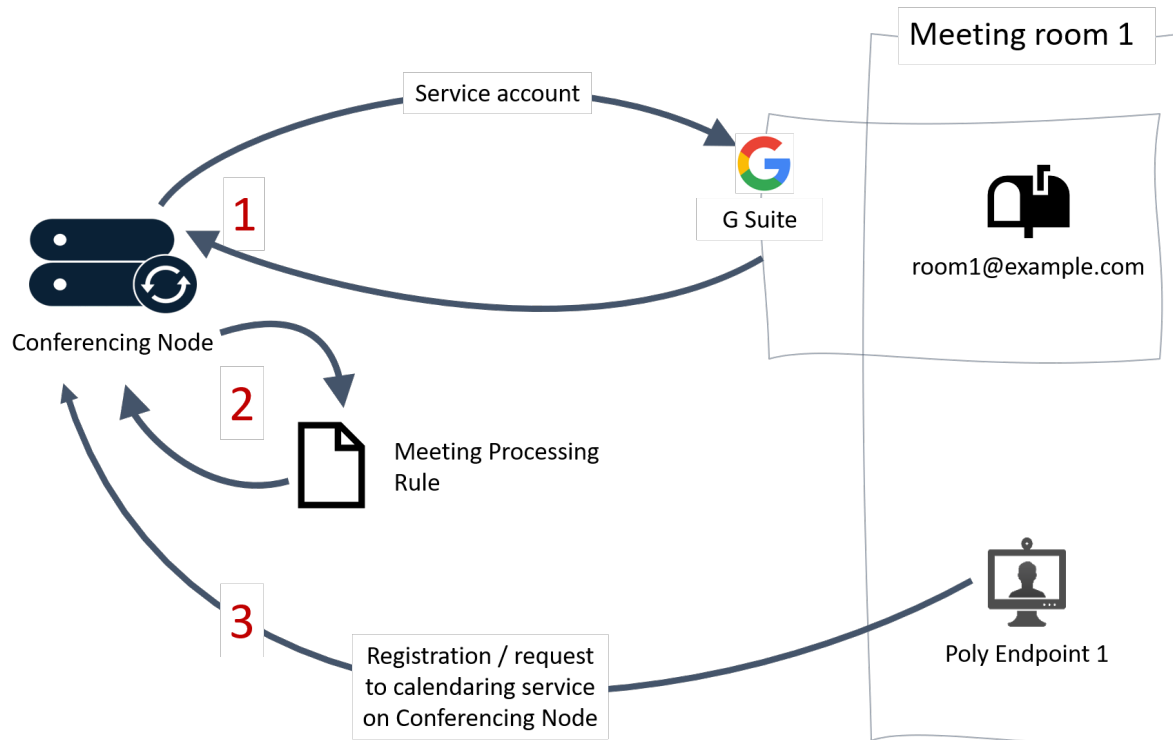
Save

Configuring Poly OTD endpoints for OTJ

In order for Poly OTD endpoints to obtain One-Touch Join meeting information, each One-Touch Join Conferencing Node emulates a Microsoft Exchange server. The Poly endpoint then connects to the Conferencing Node and registers to the calendaring service on the node in order to pull meeting information, as shown in the diagram below.

Note that this emulation of an Exchange calendaring service on the Conferencing Node is purely to provide the Poly endpoint with its meeting information. It is completely separate to the process by which the Conferencing Node initially obtains the meeting information from the calendar/email service being used for One-Touch Join — which can be either Exchange or Google Workspace.

It is important that you do not set up your Poly endpoints until after you have completed the steps to [add the endpoint details to Pexip Infinity](#).



Enabling the client API

In order to use Poly endpoints with One-Touch Join, you must ensure that the Client API is enabled on Pexip Infinity, via the global setting [Enable support for Pexip Infinity Connect clients and Client API](#).

- i** If you are deploying One-Touch Join as a [dedicated stand-alone platform](#) that includes Poly endpoints, you must still enable the client API. However, for added security you can disable some or all call protocols (i.e. SIP, SIP UDP, H.323, WebRTC and RTMP). For more information, see [Enabling and disabling SIP, H.323, WebRTC and RTMP](#).

DNS records

If you have a One-Touch Join deployment that includes Poly endpoints in a location with more than one Conferencing Node, you should spread the Poly endpoint registrations across all nodes in the location to maximize performance and provide redundancy. To achieve this, we recommend that all Poly endpoints in a location register to a single FQDN which uses round-robin DNS to resolve to each Conferencing Node in turn. This will require you to set up appropriate DNS records for all Conferencing Nodes in the location, and ensure that your DNS server is configured to round-robin between these records.

For more information and examples, see [Enabling Poly endpoints to register to One-Touch Join](#).

Poly authentication

In normal Pexip Infinity usage Poly endpoints authenticate to One-Touch Join using digest authentication, with the exception of HDX endpoints which require NTLMv2.

When Pexip Infinity has been [deployed in a secure mode of operation](#) (and therefore FIPS compliance has been enabled), NTLMv2 and digest authentication are disabled and basic authentication is used. As a result, when in this mode:

- HDX endpoints are not supported
- Trio endpoints must be configured to allow basic authentication.

Deployments with a load balancer

If there is a **load balancer** between the Poly endpoints and the One-Touch Join Conferencing Node, the load balancer should be configured to set the `X-Forwarded-For` header. This preserves the endpoint's IP address in communications, allowing requests and subsequent responses to be directed to the correct endpoint.

Configuring Poly RealPresence Group series

One-Touch Join supports Poly RealPresence Group Series endpoints running v5.0.0 or later.

To configure a Poly RealPresence Group Series for One-Touch Join, use the following settings (which can be found on the endpoint under **Admin settings > Servers > Calendaring service**):

Field	Poly configuration	Matching Infinity configuration	Additional info
Email	The email address of the room resource configured in Exchange or Google Workspace that is associated with this endpoint.	This must be the same as the Room resource email configured on Pexip Infinity for this endpoint.	
Domain	Leave blank.		This is the Exchange domain, and is not required for One-Touch Join.
User Name	The username the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	This must be the same as the Poly Calendaring Username configured on Pexip Infinity for this endpoint.	This field is case-sensitive. Each Poly endpoint must have a unique User Name .
Password	The password the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	This must be the same as the Poly Calendaring password configured on Pexip Infinity for this endpoint.	This field is case-sensitive.
Auto Discover Using	Do not select this button.		Auto Discovery is not supported. Instead, you should manually configure the Microsoft Exchange Server settings.

Field	Poly configuration	Matching Infinity configuration	Additional info
Microsoft Exchange Server	<ul style="list-style-type: none"> If you have a single Conferencing Node in this location, enter the IP Address or FQDN of the node (in the format 192.168.0.0 or host.example.com). If you have multiple Conferencing Nodes in this location, you should use DNS round robin; therefore this will be the FQDN of the DNS record for this location (in the format host.example.com). <p>In both cases, the location is the Pexip Infinity location associated with the Endpoint Group to which this endpoint belongs.</p>		
Secure Connection Protocol	Select Automatic .		
Meeting Reminder Time in Minutes	Optional		Can still be used in conjunction with One-Touch Join.
Play Reminder Tone When Not in a Call	Optional		Can still be used in conjunction with One-Touch Join.
Show Information for Meetings Set to Private	Optional		<p>Enable private meetings must be enabled on the One-Touch Join Profile associated with this endpoint in order for this setting to apply.</p> <p>If Enable private meetings has been disabled on the One-Touch Join Profile, this setting will have no effect.</p>

To confirm that the Poly RealPresence Group Series endpoint has registered successfully with the calendaring service:

1. On the endpoint, go to **Admin Settings > Servers > Calendaring Service**.
2. Confirm that the **Registration Status** is showing as **Registered**.

Configuring Poly Trio series

When configuring Poly Trio series endpoints for One-Touch Join, you should use a [Generic base profile](#) unless your deployment specifically requires you to use a [Skype for Business base profile](#). Configuration instructions for each are given below.

Configuring Poly Trio using a generic base profile

1. Open the endpoint's web UI at <https://<ipaddress>>, select **Admin**, and log in using the admin password.
2. From **Simple Setup > Base Profile** select **Generic** and then select **Save**.
3. Edit the config file as follows (this can be done via **Utilities > Import & Export Configuration**):
 - add the following:
`feature.contactPhotoIntegration.enabled="0"`
 - if the Trio is running software version 5.9.1.11135 or later and Pexip Infinity has been [deployed in a secure mode of operation](#) (and therefore FIPS compliance has been enabled), you must allow the Trio to use basic authentication by adding:
`feature.exchange.allowBasicAuth="1"`
4. From **Settings > Applications**, configure the Poly trio as follows:

Field	Poly configuration	Matching Infinity configuration	Additional info
Exchange Applications			
Exchange Calendar	Select Enable .		
Auto Discover Using	Select Disable .		Auto Discover is not supported. Instead, you should manually configure the Exchange Server URL settings.
Exchange Server URL	Enter <code>https://<address>/EWS/Exchange.asmx</code> For <address> : <ul style="list-style-type: none"> If you have a single Conferencing Node in this location, enter the IP Address or FQDN of the node. If you have multiple Conferencing Nodes in this location, you should use DNS round robin; therefore this will be the FQDN of the DNS record for this location. In both cases, the location is the Pexip Infinity location associated with the Endpoint Group to which this endpoint belongs.		Note that the URL is case-sensitive; in particular, ensure that EWS and Exchange are capitalised as shown.
Exchange Sign-In *			
Exchange Email	Leave blank.		
Domain	Leave blank.		This is the Exchange domain, and is not required for One-Touch Join.
User	The username the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	This must be the same as the Poly Calendaring Username configured on Pexip Infinity for this endpoint.	This field is case-sensitive. Each Poly endpoint must have a unique User name.
Password	The password the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	This must be the same as the Poly Calendaring password configured on Pexip Infinity for this endpoint.	This field is case-sensitive.
* Available for endpoints running version 5.9.2.7727 or later. For earlier versions, you must have physical access to the endpoint's touch screen; use this to log in using the User and Password credentials as described above.			

To confirm that the Poly Trio endpoint has registered successfully with the calendaring service:

1. On the endpoint, go to **Diagnostics > Exchange Status**.
2. Confirm that **Exchange Calendar** is showing as **Synchronized**.

Configuring Poly Trio using Skype for Business base profile

You should only use the Skype for Business base profile if specifically required in your deployment (for example, if you wish to place PSTN calls via Skype for Business server); otherwise use the [generic base profile](#).

When the Trio is configured as described below, it will still register with the calendaring service on the Conferencing Node to obtain meeting information, but it will also register with Skype for Business and use that to place outbound calls.

1. Open the endpoint's web UI at <https://<ipaddress>>, select **Admin**, and log in using the admin password.
2. From **Simple Setup > Base Profile** select **Skype for Business** and then select **Save**.
3. Edit the config file (this can be done via **Utilities > Import & Export Configuration**) by adding:


```
feature.exchangeVoiceMail.enabled="0"
exchange.showSeparateAuth="1"
feature.exchangeContacts.enabled="0"
```
4. From **Settings > Skype For Business SignIn**, configure the Poly trio as follows:

Field	Poly configuration	Matching Infinity configuration	Additional info
Skype for Business			
Use User Credentials	Select Enable .		
Authentication Type	Select User Credentials .		
Sign-in Address	The endpoint's Skype for Business address.		
Domain	The endpoint's Skype for Business domain.		
User	The name the endpoint will use to authenticate with Skype for Business.		
Password	The password the endpoint will use to authenticate with Skype for Business.		
Microsoft Exchange Server Configuration			
Exchange Email	The email address of the room resource that is associated with this endpoint.	This must be the same as the Exchange target mailbox configured on the endpoint, and Room resource email configured on Pexip Infinity for this endpoint.	
Exchange Domain	Leave blank.		
Exchange User	The username the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	This must be the same as the Poly Calendaring Username configured on Pexip Infinity for this endpoint.	This field is case-sensitive. Each Poly endpoint must have a unique User name.
Exchange Password	The password the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	This must be the same as the Poly Calendaring password configured on Pexip Infinity for this endpoint.	This field is case-sensitive.

Field	Poly configuration	Matching Infinity configuration	Additional info
Exchange Target Mailbox	The email address of the room resource that is associated with this endpoint.	This must be the same as the Exchange Email configured on the endpoint, and Room resource email configured on Pexip Infinity for this endpoint.	

5. From **Settings > Applications**, configure the Poly trio as follows:

Field	Poly configuration	Matching Infinity configuration	Additional info
Exchange Applications			
Exchange Calendar	Select Enable .		
Auto Discover	Select Disable .		Auto Discover is not supported. Instead, you should manually configure the Exchange Server URL settings.
Exchange Server URL	Enter https://<address>/EWS/Exchange.asmx For <address>: <ul style="list-style-type: none"> If you have a single Conferencing Node in this location, enter the IP Address or FQDN of the node. If you have multiple Conferencing Nodes in this location, you should use DNS round robin; therefore this will be the FQDN of the DNS record for this location. In both cases, the location is the Pexip Infinity location associated with the Endpoint Group to which this endpoint belongs.		Note that the URL is case-sensitive; in particular, ensure that EWS and Exchange are capitalised as shown.

To confirm that the Poly Trio endpoint has registered successfully with the calendaring service:

- On the endpoint, go to **Diagnostics > Exchange Status**.
- Confirm that **Exchange Calendar** is showing as *Synchronized*.

Configuring Poly HDX series

To configure the Poly HDX for One-Touch Join, go to the endpoint's IP address.

From **Admin Settings > Global Services > Calendaring Service**, enter the following:

Field	Poly configuration	Matching Infinity configuration	Additional info
Enable Calendaring Service	Select this option.		

Field	Poly configuration	Matching Infinity configuration	Additional info
Microsoft Exchange Server Address	<ul style="list-style-type: none"> If you have a single Conferencing Node in this location, enter the IP Address or FQDN of the node (in the format 192.168.0.0 or host.example.com). If you have multiple Conferencing Nodes in this location, you should use DNS round robin; therefore this will be the FQDN of the DNS record for this location (in the format host.example.com). <p>In both cases, the location is the Pexip Infinity location associated with the Endpoint Group to which this endpoint belongs.</p>		
Domain	Leave blank.		This is the Exchange domain, and is not required for One-Touch Join.
User Name	The username the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	This must be the same as the Poly Calendaring Username configured on Pexip Infinity for this endpoint.	<p>This field is case-sensitive.</p> <p>Each Poly endpoint must have a unique User Name.</p>
Password	Select this option. The following two fields will appear:		
New Password Confirm Password	The password the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	This must be the same as the Poly Calendaring password configured on Pexip Infinity for this endpoint.	This field is case-sensitive.
Mailbox	<p>For Exchange integrations: the email address of the room resource configured in Exchange that is associated with this endpoint.</p> <p>For Google Workspace integrations: the User Name entered above.</p>	For Exchange integrations, this must be the same as the Room resource email configured on Pexip Infinity for this endpoint.	
Reminder Time in Minutes	Optional		Can still be used in conjunction with One-Touch Join.
Play Reminder Tone	Optional		Can still be used in conjunction with One-Touch Join.
Show Private Meeting Information	Optional		<p>Enable private meetings must be enabled on the One-Touch Join Profile associated with this endpoint in order for this setting to apply.</p> <p>If Enable private meetings has been disabled on the One-Touch Join Profile, this setting will have no effect.</p>

To confirm that the Poly HDX endpoint has registered successfully with the calendaring service:

1. On the endpoint, go to **Admin Settings > Global Services > Calendaring Service**.
2. Confirm that there is a green tick next to **Enable Calendaring Service**.

Configuring Poly Studio X series and Poly G7500 series

To configure the Poly Studio or Poly G7500 for One-Touch Join, go to the endpoint's IP address and sign in to the endpoint if required.

From **Servers > Calendaring Service**, enter the following:

Field	Poly configuration	Matching Infinity configuration	Additional info
Enable Calendaring Service	Select this option.		
Email	The email address of the room resource configured in Exchange or Google Workspace that is associated with this endpoint.	This must be the same as the Room resource email configured on Pexip Infinity for this endpoint.	
Domain	Leave blank.		This is the Exchange domain, and is not required for One-Touch Join.
User Name	The username the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	This must be the same as the Poly Calendaring Username configured on Pexip Infinity for this endpoint.	This field is case-sensitive. Each Poly endpoint must have a unique User Name .
Password	The password the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	This must be the same as the Poly Calendaring password configured on Pexip Infinity for this endpoint.	This field is case-sensitive.
Microsoft Exchange Server	<ul style="list-style-type: none"> • If you have a single Conferencing Node in this location, enter the IP Address or FQDN of the node (in the format <code>192.168.0.0</code> or <code>host.example.com</code>). • If you have multiple Conferencing Nodes in this location, you should use DNS round robin; therefore this will be the FQDN of the DNS record for this location (in the format <code>host.example.com</code>). <p>In both cases, the location is the Pexip Infinity location associated with the Endpoint Group to which this endpoint belongs.</p>		
Meeting Reminder Time in Minutes	Optional		Can still be used in conjunction with One-Touch Join.
Play Reminder Tone When Not in a Call	Optional		Can still be used in conjunction with One-Touch Join.

Field	Poly configuration	Matching Infinity configuration	Additional info
Show Information for Meetings set to Private	Optional		<p>Enable private meetings must be enabled on the One-Touch Join Profile associated with this endpoint in order for this setting to apply.</p> <p>If Enable private meetings has been disabled on the One-Touch Join Profile, this setting will have no effect.</p>

To confirm that the Poly Studio / Poly G7500 endpoint has registered successfully with the calendaring service:

1. On the endpoint, go to **Servers > Calendaring Service**.
2. Confirm that the **Registration Status** is showing as **Registered**.

Configuring Poly Debut series

To configure the Poly Debut for One-Touch Join, from **Server Settings > Calendar**, enter the following:

Field	Poly configuration	Matching Infinity configuration	Additional info
Enable Calendar	Select <i>Enable</i> .		
Microsoft Exchange Server	<ul style="list-style-type: none"> If you have a single Conferencing Node in this location, enter the IP Address or FQDN of the node (in the format 192.168.0.0 or host.example.com). If you have multiple Conferencing Nodes in this location, you should use <u>DNS round robin</u>; therefore this will be the FQDN of the DNS record for this location (in the format host.example.com). <p>In both cases, the location is the Pexip Infinity location associated with the Endpoint Group to which this endpoint belongs.</p>		
Domain	Leave blank.		This is the Exchange domain, and is not required for One-Touch Join.
User Name	The username the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	This must be the same as the <u>Poly Calendaring Username</u> configured on Pexip Infinity for this endpoint.	<p>This field is case-sensitive.</p> <p>Each Poly endpoint must have a unique User Name.</p>
Password	The password the endpoint will use when connecting and authenticating to the calendaring service on the Conferencing Node, to obtain meeting information.	This must be the same as the <u>Poly Calendaring password</u> configured on Pexip Infinity for this endpoint.	This field is case-sensitive.

To confirm that the Poly Debut endpoint has registered successfully with the calendaring service:

1. On the endpoint, go to the **Device Status** page.
2. In the **Calendar** row of the table, check that the **Status** is showing as *Registered*.

One-Touch Join meeting types and transforms

This topic details the meeting types, transform patterns and variables that are supported when [Adding One-Touch Join meeting processing rules](#).

You must configure One-Touch Join with information about all the different types of meeting invitations you expect to encounter in your deployment, and rules for how the information in each of these invitations should be used to derive the alias that the endpoint will dial to join the meeting.

You can select from the currently [supported meeting types](#) (which you can edit if necessary), or create your own [regex](#) or [custom](#) rules if you wish to enable One-Touch Join for other meeting types or conferencing providers not already supported. There are also some non-configurable [fallback](#) settings that are used when no other rules match.

You must also ensure that your deployment has appropriate [Call Routing Rules](#) to enable the OTJ endpoint to dial the meeting aliases that are derived for each meeting type.

Fallback alias matching

If One-Touch Join cannot find a valid meeting alias because none of the meeting processing rules match, or because there are no meeting processing rules configured or enabled, as a fallback it will **always** search the body and the location of the invitation for one of the following patterns to use as the alias to dial:

- sip:<uri>
- sips:<uri>
- h323:<address>

Supported meeting types

The table below lists the currently supported configurable **Meeting types**. For each type, the **Default transform pattern** shows how, when [default processing](#) is enabled, One-Touch Join uses the information it finds in the meeting invitation to derive the alias that the endpoint will dial to join the meeting. The table also lists the **Valid variables** that can be used when creating a [custom transform pattern](#) for this meeting type.

i These meeting types are supported by the current version of Pexip Infinity at the time of its release, but sometimes conferencing providers change the format of their meeting invitations. Until these changes are incorporated into a subsequent release of Pexip Infinity, you may need to use a custom rule in order to continue to use One-Touch Join for these meeting types. Where possible we will provide these for you in the [Custom meeting type](#) section of the Pexip documentation, so we suggest that you check this content regularly between upgrades.

Meeting type	Usage and notes	Default transform pattern	Valid variables (if not using default transform)
Pexip Infinity	<p>For meetings scheduled using Pexip's VMR Scheduling for Exchange feature, and which use the default Joining instructions template. These meetings typically include a join link in the format pexip://<meeting_id>@<domain>.</p> <p>If your VMR Scheduling for Exchange deployment does not use the default template, or uses an alias in a different format, you should select a Meeting type of Domain or Custom instead.</p>	{{meeting_id}}@{{domain}}	<ul style="list-style-type: none"> • meeting_id • domain

Meeting type	Usage and notes	Default transform pattern	Valid variables (if not using default transform)
Pexip Service	<p>For meetings held in Pexip Service VMRs.</p> <p>By default, the resulting alias will use the domain <code>pexip.me</code></p>	<code>{{meeting_id}}@pexip.me</code>	<ul style="list-style-type: none"> meeting_id domain
Microsoft Teams Meeting Properties	<p>(Not currently supported for Google Workspace integrations)</p> <p>For meetings hosted in Microsoft Teams. This rule should be sufficient if all your Teams meetings are internal; otherwise we recommend that you also add any relevant <i>Microsoft Teams Meeting Body for ...</i> rules.</p> <p>You must provide the Domain that will be used when deriving the alias — this should be the domain from which the meeting invitation was sent.</p>	<code>{{meeting_id}}@{{domain}}</code>	<ul style="list-style-type: none"> meeting_id domain
Microsoft Teams SIP Guest Join	<p>For Microsoft Teams meetings hosted on another domain where the hosting domain does not have Pexip Teams interop but your organization does have access to Pexip Teams interop through the Pexip Service.</p> <p>Leave Default processing enabled checked.</p>	<code>{{base32_encoded_blob}}.{{tenant_id}}@pex.ms</code>	
Microsoft Teams Meeting Body for Pexip Infinity	If you expect users in your deployment to receive invitations to Microsoft Teams meetings sent from domains other than your own, where the meeting organizer is using a Pexip Infinity — Teams integration.	<code>{{prefix}}{{meeting_id}}@{{domain}}</code>	<ul style="list-style-type: none"> meeting_id domain prefix
Microsoft Teams Meeting Body for Pexip Service	If you expect users in your deployment to receive invitations to Microsoft Teams meetings sent from domains other than your own, where the meeting organizer is using a Pexip Service — Teams integration.	<code>{{meeting_id}}@{{domain}}</code>	<ul style="list-style-type: none"> meeting_id domain
Microsoft Teams Meeting Body for Poly	<p>If you expect users in your deployment to receive invitations to Microsoft Teams meetings sent from domains other than your own, where the meeting organizer is using a Poly — Teams integration.</p> <p>By default, the resulting alias will use the domain <code>t.plcm.vc</code></p>	<code>{{tenant_id}}.{{meeting_id}}@t.plcm.vc</code>	<ul style="list-style-type: none"> meeting_id domain tenant_id
Microsoft Teams Meeting Body for BlueJeans	<p>If you expect users in your deployment to receive invitations to Microsoft Teams meetings sent from domains other than your own, where the meeting organizer is using a BlueJeans — Teams integration.</p> <p>By default, the resulting alias will use the domain <code>teams.bjn.vc</code></p>	<code>{{tenant_id}}.{{meeting_id}}@teams.bjn.vc</code>	<ul style="list-style-type: none"> meeting_id domain tenant_id
Microsoft Teams Meeting Body for Cisco	If you expect users in your deployment to receive invitations to Microsoft Teams meetings sent from domains other than your own, where the meeting organizer is using a Cisco — Teams integration.	<code>{{conf_id}}.{{tenant_key}}@{{domain}}</code>	<ul style="list-style-type: none"> conf_id tenant_key domain

Meeting type	Usage and notes	Default transform pattern	Valid variables (if not using default transform)
Google Meet	<p>For meetings scheduled using Google Meet.</p> <p>This is supported in Google Workspace integrations, and in Exchange integrations (on-premises or O365) if the Google Meet invitation uses <i>Long meeting IDs</i> (which automatically include a SIP dial-in address).</p> <p>This option is not supported in Exchange integrations (on-premises or O365) if the Google Meet invitation uses <i>Short meeting IDs</i> only, because the SIP dial-in address is not automatically included in these invitations.</p>	{{meeting_id}}@{{domain}}	<ul style="list-style-type: none"> meeting_id domain
Google Meet SIP Guest Join	<p>For Google Meet meetings hosted on another domain where the hosting domain does not have Pexip Google interop but your organization does have access to Pexip Google interop through the Pexip Service.</p> <p>Leave Default processing enabled checked, and enter the Domain of the service providing the Pexip Google interop — for the Pexip Service this is <code>google.pexip.me</code>.</p>	{{meeting_id}}@{{domain}}	<ul style="list-style-type: none"> meeting_id domain
Skype for Business	<p>For Skype for Business meetings.</p> <p>The domain used is the domain of the organizer's email address.</p> <p>You must also ensure you have a Call Routing Rule configured that includes the following settings (replacing <code>example\com</code> in the example below with the domain of the organizer's email address):</p> <ul style="list-style-type: none"> Destination alias regex match: <pre>__sfb__[a-z0-9]+\.[a-z\.-]+\.(example\com)</pre> Regex replace string: <pre>sip:\2@3;gruu;opaque=app:conf:focus:id:\1</pre> Call target: <p><i>Lync / Skype for Business clients, or meetings via a Virtual Reception</i></p> 	__sfb__{{focus_id}}.{{user}}@{{domain}}	<ul style="list-style-type: none"> focus_id domain
Skype for Business Meeting Body for Poly	<p>For Skype for Business meetings, where the meeting organizer is using a Sfb — Poly integration.</p> <p>By default, the resulting alias will use the domain <code>v.plcm.vc</code></p>	{{tenant_id}}.{{meeting_id}}@v.plcm.vc	<ul style="list-style-type: none"> meeting_id domain tenant_id
Webex	For Webex meetings.	{{meeting_id}}@{{domain}}	<ul style="list-style-type: none"> meeting_id domain
Zoom	<p>For Zoom meetings.</p> <p>By default, the resulting alias will use the domain <code>zoomcrc.com</code></p> <p>Optionally, the meeting password (PIN) can be included in the alias (not supported for Google Workspace integrations).</p>	{{meeting_id}}@zoomcrc.com {{meeting_id}}. {{pin}}@zoomcrc.com	<ul style="list-style-type: none"> meeting_id domain pin

Meeting type	Usage and notes	Default transform pattern	Valid variables (if not using default transform)
BlueJeans	For BlueJeans meetings. By default, the resulting alias will use the domain <code>bjn.vc</code>	<code>{{meeting_id}}@bjn.vc</code>	<ul style="list-style-type: none"> meeting_id domain
GoToMeeting	For GoToMeeting meetings.	<code>{{meeting_id}}@{{domain}}</code>	<ul style="list-style-type: none"> meeting_id domain
Domain	<p>If you expect users in your deployment to receive invitations for meetings that do not fall into any of the above categories, you can use this rule to enable meetings where the alias is from a known domain.</p> <p>We recommend that <i>Domain</i> rules are given a lower priority than any of the other rules.</p> <p>You must provide the Domain that will be searched for in order to match this rule.</p> <p>This rule will search the body and the location for a match.</p> <p>The search will result in a match even if the URI includes one or more subdomains of the domain being searched for. The domain can also include subdomains. When there is a match, the full URI will be used as the meeting alias. For example, if the domain is <code>sales.example.com</code>, that will match <code>alice@sales.example.com</code> and <code>alice@us.sales.example.com</code> but not <code>alice@example.com</code>.</p>	<code>{{meeting_id}}@{{domain}}</code>	<ul style="list-style-type: none"> meeting_id domain
Regex	See Regex meeting type		
Custom	See Custom meeting type		

Regex meeting type

A **Meeting type** of *Regex* enables you to use a regular expression to search for a particular **Match string** in the body and location of the invitation. You can then either:

- select **Default processing enabled** to use the matched string as the alias that the endpoint will dial to join the meeting, or
- disable **Default processing enabled** to use a regex **Replace string** to transform the matched string into the alias to dial.

For more information on using regular expressions with Pexip Infinity, see [Regular expression reference](#).

Examples

Matching without a transform

This example searches the invitation for any alias in the format of `<name>.vmr@example.com`, and uses that as the alias to dial:

Meeting type	Regex
Default processing enabled	Yes
Match string	<code>[w+].vmr@example.com</code>

In this example, if the meeting body contains the following text:

```
From a video system (SIP/H.323): alice.vmr@example.com
```

then the alias that will be dialed to join the meeting will be `alice.vmr@example.com`

Transforming a URL into an alias

This example searches the invitation for a URL in the format `https://<domain>/meet/<name>` and transforms that into an alias in the format `<name>@<domain>`:

Meeting type	Regex
Default processing enabled	No
Match string	<code>https://([^\s]+)/meet/(\d+)</code>
Replace string	<code>\2@\1</code>

In this example, if the meeting body contains the following text:

```
From web browser & other ways to join:  
https://pexip.me/meet/123456
```

then the alias that will be dialed to join the meeting will be `123456@pexip.me`

Custom meeting type

A Meeting type of *Custom* enables more advanced processing by allowing you to use a Jinja2 template with access to all `calendar_event` information, which you can then use to generate the alias that the endpoint will dial to join the meeting. For more information on using Jinja2 with Pexip Infinity, see [Jinja2 templates and filters](#).

A custom meeting type can be used to enable meeting types or conferencing providers not listed above, or to provide a workaround if any supported providers change their current implementations. (Any known workarounds will be given in the [Examples](#) section below.)

You can use the following calendar event dictionary items, in conjunction with any other literal values if required (e.g. if the domain is always a known quantity), to create the Jinja script:

Item	Type	Additional information
subject	string	
organizer_full_name	string	
organizer_first_name	string	
organizer_last_name	string	
organizer_email	string	
start_time	dictionary	Properties:
end_time		<ul style="list-style-type: none">yearmonthdayhourminutesecond
is_private	boolean	
body	string	

Item	Type	Additional information
location	string	
properties	dictionary	<p>Google Workspace</p> <p>A Google Workspace <code>calendar_event</code> will contain a Google Calendar Event resource. For more information, see https://developers.google.com/calendar/v3/reference/events.</p> <p>Exchange</p> <p>An Exchange <code>calendar_event</code> may contain any EWS MAPI properties from the following list:</p> <ul style="list-style-type: none"> <code>item_class</code> (string): for options, see https://docs.microsoft.com/en-gb/office/vba/outlook/Concepts/Forms/item-types-and-message-classes <code>sensitivity</code> (string): for options, see https://docs.microsoft.com/en-us/dotnet/api/microsoft.exchange.webservices.data.sensitivity?view=exchange-ews-api <code>is_recurring</code> (boolean): <i>True</i> if the meeting is part of a recurring series, otherwise <i>False</i>. <code>calendar_item_type</code> (string): for options, see https://docs.microsoft.com/en-us/exchange/client-developer/web-service-reference/calendaritemtype#text-value <code>teams_vtc_conference_id</code>: available for Teams meetings only. <code>online_meeting_conf_link</code>: available for Skype for Business meetings only. <code>uc_capabilities</code>: available for Webex meetings only.

Examples

The following examples show basic jinja templates that can be used in the Custom template field.

Searching by partial alias

This first example searches the `calendar_event.body` (i.e. the text in the body of the meeting invitation) for an alias that includes `.vmr@example.com`. It then uses the full alias as the meeting alias to dial:

```
{% set matches = pex_regex_search("([\\w.-]+\\.vmr@example\\.com)", calendar_event.body) %}
{% if matches %}
  {{ matches[0] }}
{% endif %}
```

In the above example, if the meeting body contains `alice.vmr@example.com`, this will be used as the alias for the meeting.

Searching by top-level domain

This next example searches the `calendar_event.body` (i.e. the text in the body of the meeting invitation) for an alias that includes a domain ending in `.com`. It then uses the full alias as the meeting alias to dial:

```
{% set groups = pex_regex_search("([a-z0-9.-]+)@([a-z0-9.-]+.com)", calendar_event.body) %}
{% if groups %}
  {{ groups[0] }}@{{ groups[1] }}
{% endif %}
```

In the above example, if the meeting body contains `alice.vmr@example.com`, this will be used as the alias for the meeting.

Searching the location for a partial alias

This example searches the `calendar_event.location` (i.e. the text in the location field of the meeting invitation) for an alias that includes `.vmr@example.com`. It then uses the full alias as the meeting alias to dial:

```
{% set matches = pex_regex_search("([\\w.-]+\\.vmr@example\\.com)", calendar_event.location) %}
{% if matches %}
  {{ matches[0] }}
{% endif %}
```

In the above example, if the meeting location contains `alice.vmr@example.com`, this will be used as the alias for the meeting.

Lifesize Cloud example

This example searches a standard Lifesize Cloud meeting invitation and converts the URL into a meeting alias:

```
{% set matches = pex_regex_search("https://call.lifesizecloud.com/([0-9.-]+)", calendar_event.body) %}
{% if matches %}
  {{matches[0]}}@lifesizecloud.com
{% endif %}
```

In the above example, if the meeting body contains <https://call.lifesizecloud.com/123456>, the alias that will be used to join the meeting will be 123456@lifesizecloud.com.

Skype for Business example: different organizer and endpoint domains

This example can be used if you have Skype for Business meeting invitations where the domain of the organizer's email address is not the same as the domain of the alias of the SIP endpoint to be used for the meeting.

```
{% set matches = pex_regex_search("https://meet.pajusa.com/([a-z0-9-]+)\\/([a-z0-9]+)\\/([A-Z0-9]+)", calendar_event.body) %}
{% if matches %}
  __sfb__{{matches[2]}}.{{matches[1]}}@{{matches[0]}}
{% endif %}
```

In the above example, if the meeting body contains <https://meet.pajusa.com/pexample.com/alice/ABC123>, the alias that will be used to join the meeting will be __sfb__ABC123.alice@pexample.com.

Extracting an alias from a URL that has been rewritten for security

This example can be used if you expect that some meeting invitations sent to One-Touch Join room resources may include URLs that have been rewritten for security reasons. This rule matches URLs for Microsoft Teams meetings on the Pexip Service (which are in the format https://pexip.me/teams/<domain>/<meeting_id>) that have been rewritten by either:

- Safe Links in Microsoft Defender for Office 365, which start with something like
<https://nam06.safelinks.protection.outlook.com/?url=https>
- Proofpoint's URL Defense, which start with something like
<https://urldefense.proofpoint.com/v2/url?u=https>

```
{% set matches = pex_regex_search("https(-3A_|%3A%2F%2F)pexip\\.me(\\_|%2F)teams(\\_|%2F)(([a-zA-Z0-9-]+)(\\.[a-zA-Z0-9-]+){0,}\\.[a-zA-Z]{2,})\\(\\_|%2F)\\(\\d+)", calendar_event.body) %}
{% if matches %}
  {{matches[0]}}@{{matches[3]}}
{% endif %}
```

In the above example, If the original meeting invitation contained a URL in the format <https://pexip.me/teams/pexample.com/123456> and that URL was subsequently rewritten by either Safe Links or URL Defense, the alias that will be used to join the meeting will be 123456@pexample.com.

Deploying a dedicated One-Touch Join platform

In most cases, One-Touch Join will be implemented as a feature within a wider Pexip Infinity deployment, and run on Conferencing Nodes alongside other Pexip Infinity services. However, you can also set up separate OTJ locations within your deployment that contain Conferencing Nodes used solely for One-Touch Join. A third option appropriate in some situations is to implement a separate Pexip Infinity deployment purely for One-Touch Join, for example if you are a [Pexip Service customer wishing to use One-Touch Join](#), or you are a large enterprise wishing to separate the resources used for your One-Touch Join deployment.

If you are implementing a dedicated One-Touch Join deployment alongside but separate from a Pexip Infinity deployment, they do not need to be running the same software version, as there is no interaction between the two deployments. This means that existing Pexip Infinity environments can implement a dedicated One-Touch Join deployment without having to upgrade their existing software.

Minimum hardware requirements

A dedicated One-Touch Join deployment consists of one [Management Node](#) and at least one [Conferencing Node](#). Further Conferencing Nodes can be deployed for redundancy.

For dedicated One-Touch Join-only deployments, the resource requirements are minimal, therefore you may use the **minimum** server specifications outlined below. However, if you expect to broaden your deployment to implement some of the wider Pexip Infinity features in the future, you will need to increase the specifications of your hardware.

On-premises deployments

When setting up a dedicated One-Touch Join deployment using servers in your own datacenters, we recommend the following as a minimum:

- Management Node:
 - 4 cores
 - 4 GB RAM
 - AVX or later processor
 - 100 GB SSD storage
 - The Pexip Infinity VMs are delivered as VM images (.ova etc.) to be run directly on the hypervisor. No OS should be installed.
- Conferencing Nodes:
 - 4 cores
 - 4 GB RAM
 - AVX or later processor
 - 50 GB SSD storage per Conferencing Node, 500 GB total per server (to allow for snapshots etc.)
 - The Pexip Infinity VMs are delivered as VM images (.ova etc.) to be run directly on the hypervisor. No OS should be installed.

For more information, see [server design guidelines](#).

Cloud deployments

When setting up a dedicated One-Touch Join deployment using a cloud service, you can generally use the same sized server for the Conferencing Node(s) as you do for the Management Node. We therefore recommend the following as a minimum:

GCP

- Management Node: a machine type with 4 vCPUs (n1-standard-4) or larger
- Conferencing Node: a machine type with 4 vCPUs (n1-standard-4) or larger

AWS

- Management Node: an **m5.xlarge** instance
- Conferencing Node: an **m5.xlarge** instance

Azure

- Management Node: an **F4s v2** instance
- Conferencing Node: an **F4s v2** instance

Minimum Pexip Infinity platform configuration

You must ensure the following components of the Pexip Infinity platform are configured and working appropriately:

- [DNS servers](#)
- [NTP servers](#)
- [Locations](#) (note that you do not need to configure any media overflow locations, as this concept is not used by One-Touch Join).
- [Licenses](#): you will need an **OTJ** license for each endpoint that will use the One-Touch Join feature.
- [Custom CA certificates](#): only required if you are using One-Touch Join with Exchange on-premises, and your Exchange server does not use a globally trusted certificate.

Call Routing Rules are not required on the dedicated One-Touch Join deployment, because these deployments do not handle any calls. However, you must ensure that your call control system is configured so that calls being placed by the endpoints to each of the supported meeting types can be routed appropriately.

One-Touch Join configuration

The process of configuring One-Touch Join in a dedicated environment is the same as when configuring it as part of a wider Pexip Infinity deployment, namely:

1. Configuring your calendar/email service:
 - [Configure Google Workspace for One-Touch Join](#), including [Adding a One-Touch Join Google Workspace integration on Pexip Infinity](#), or
 - [Configure Exchange on-premises for One-Touch Join](#), including [Adding a One-Touch Join Exchange integration on Pexip Infinity](#), or
 - [Configure Office 365 for One-Touch Join](#), including [Adding a One-Touch Join Exchange integration on Pexip Infinity](#)
2. [Adding a One-Touch Join profile](#)
3. [Adding One-Touch Join endpoint groups](#)
4. [Adding One-Touch Join endpoints](#)
5. [Adding One-Touch Join meeting processing rules](#)

For more information, see [Configuring Pexip Infinity for One-Touch Join](#)

Scheduling and joining meetings using One-Touch Join

You can use One-Touch Join to join meetings via the videoconferencing endpoints in your meeting rooms, or via your own personal endpoints.

Using One-Touch Join in meeting rooms

When the One-Touch Join feature has been enabled for meeting rooms in your environment, you don't need to do anything special in order to use it — everything will happen automatically:

1. You or the meeting organizer create a meeting invitation in Outlook, Google calendar, or via the Teams client in your usual way. This includes any invitations that are created by using add-in buttons, for example for Pexip scheduled meetings or for Webex.
2. Add the meeting room to the invitation as a room resource.
3. Each endpoint in each meeting room will display a list of scheduled meetings for that room. When a meeting is due to start, the endpoint in the meeting room will show a **Join** or **Join meeting** button.
4. When you are ready to join the meeting, just press the **Join** button. The endpoint will dial in to the meeting.

Using One-Touch Join with your personal endpoint

If you have a personal videoconferencing endpoint that supports One-Touch Join, you may be able to link this with your own calendar, so that you can use OTJ on the endpoint to join any meetings to which you are invited. This depends on your organization's policies and network, so you'll need to contact your system administrator to see if this is possible.

Viewing One-Touch Join status

You can check the status of your One-Touch Join deployment by viewing a list of all currently scheduled One-Touch Join [meetings](#), and by viewing a list of all [endpoints](#) enabled for One-Touch Join.

Viewing One-Touch Join meetings

To view a list of all currently scheduled meetings that use Pexip Infinity's One-Touch Join feature in your deployment, go to **Status > One-touch Join Meetings**.

This page lists all One-Touch Join meetings with a start time from one day in the past up to the number of days in the future specified by the associated One-Touch Join profile's [No. of upcoming days](#) setting. For recurring meetings, this page will list only those recurrences of the meeting that fall within this timeframe.

This information is updated each time the OTJ process runs. The OTJ process obtains meeting information by reading the room resources' calendars, and then processing the information based on the currently configured OTJ profile settings and meeting processing rules. This means that any changes to room resources' calendars (e.g. adding meetings, canceling meetings, or changing the meeting information), or any changes to the way the meeting information is processed (e.g. changes to the OTJ profile settings, or to meeting processing rules) will be reflected in the status after the OTJ process next runs. This could be between 30 seconds and many minutes, depending on the number of OTJ rooms in your deployment.

To view full details about a meeting, click on the meeting subject. The following information is available for each meeting:

Field	Description
Meeting subject	<p>The text that appears in the subject line of the meeting invitation.</p> <p>This field will show the organizer's name instead of the meeting subject if:</p> <ul style="list-style-type: none"> Replace subject is set to <i>Always</i>, and the Replace subject string is empty, or Replace subject is set to <i>Private</i>, the meeting was flagged as private, and the Replace subject string is empty, or Replace empty subject has been enabled and there was no subject.
Organizer name *	The name of the person who created the meeting invitation.
Organizer email	The email address of the person who created the meeting invitation.
Start time	The scheduled start time of the meeting. This does not include the Start buffer .
End time	The scheduled end time of the meeting. This does not include the End buffer .
Endpoint name	The name of the endpoint, as configured in Pexip Infinity.
OTJ Profile name	The name of the OTJ profile used when processing this meeting.
Meeting alias	<p>The alias that the endpoint will use to dial in to the meeting.</p> <p>This will be blank if either:</p> <ul style="list-style-type: none"> Process alias for private meetings has been disabled and the meeting was flagged as private, or Enable non-video meetings has been enabled, but OTJ was not able to obtain a valid alias for the meeting.
Meeting room email *	The email address of the room resource in whose calendar the meeting has been scheduled.
Matched meeting processing rule *	<p>The name of the meeting processing rule that was matched and used to process this meeting.</p> <p>This will be blank if the meeting information did not match any meeting processing rules, and Enable non-video meetings has been enabled.</p>

* Only displayed when you have selected an individual OTJ meeting to view.

Viewing One-Touch Join endpoints

To view a list of all endpoints in your deployment that are actively available for use by Pexip Infinity's One-Touch Join feature, go to **Status > One-touch Join Endpoints**.

This page lists all Cisco/Webex endpoints that One-Touch Join has successfully contacted, and all Poly endpoints that have successfully contacted One-Touch Join. For both, it lists the date and time of the most recent contact.

Cisco and Webex endpoints are contacted once an hour, or sooner if the OTJ process detects a change in the room resource's calendar when it next runs. Poly endpoints make contact at least every 5 minutes.

To view full details about an endpoint, click on the endpoint name. The following information is available for each endpoint:

Field	Description
Endpoint name	The name of the endpoint, as configured in Pexip Infinity.
Endpoint type	The type of "click to join" feature supported by this endpoint.
Endpoint address	The IP address of the endpoint.
Meeting room email	The email address of the room resource associated with this endpoint.
OTJ Profile name	The name of the OTJ profile used when processing this meeting.
Current node *	The IP address and name of the Conferencing Node that last had contact with the endpoint.
Last contacted	The date and time that contact was last made with the endpoint.
Number of meetings *	The number of currently scheduled One-Touch Join meetings that will use this endpoint.
* Only displayed when you have selected an individual OTJ meeting to view.	

Configuring Google Workspace for domain user authorization

This topic describes an alternative method to configuring Google Workspace for One-Touch Join in environments where the recommended method of using a service account for authorization is not desirable. This alternative method uses a domain user for authorization (referred to as the "authorization user"), which authenticates to Google Workspace using 3-legged OAuth.

The process involves the following steps, described in more detail in the sections that follow:

1. [Setting up OAuth authentication](#) for One-Touch Join.
 2. [Creating a room resource](#) for each physical room that will have a One-Touch Join endpoint in it.
 3. [Configuring the room resource](#) with the necessary permissions and settings to support One-Touch Join.
 4. [Updating the quota](#) for the number of user requests per 100 seconds.
 5. For larger deployments, [Requesting an increase to API limits](#).
 6. [Adding a One-Touch Join Google Workspace integration](#) on Pexip Infinity.
- i** If you have already set up a One-Touch Join Google Workspace integration and simply wish to add an existing room to it, you need only [configure the room resource](#) in Google Workspace and then [add the endpoint to the Google Workspace integration](#) in Pexip Infinity.

Prerequisites

You must have already created a user account specifically to be used as the Google Workspace authorization user. This user account does not need to have any special privileges; as part of the configuration described below you will grant this user access to all the One-Touch Join room resource calendars.

Enabling authorization using OAuth

In this step you create a project to use for One-Touch Join. You then enable the Calendar API for this project, and create the OAuth credentials to be used when One-Touch Join accesses the API as the authorization user.

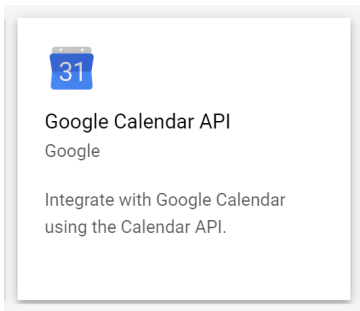
1. Creating a new project:
 - a. Go to <https://console.developers.google.com> (logged in as a Google Workspace administrator).
 - b. From the top left of the page, select the down arrow:



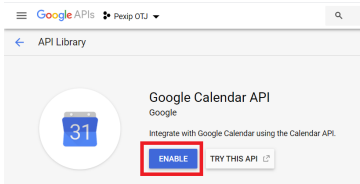
- c. Select **New Project**.
 - d. Enter a **Project name** (e.g. One-Touch Join) and select **Create**.
2. Enabling the Calendar API for the project:
 - a. Go to <https://console.developers.google.com>
 - b. From the top left of the page, select the down arrow, select your newly-created project, and select **Open**. Your new project should now be showing at the top left of the page:



- c. From the navigation menu on the left of the screen, select **APIs & Services > Library**, then scroll down and select the **Google Calendar API** tile:

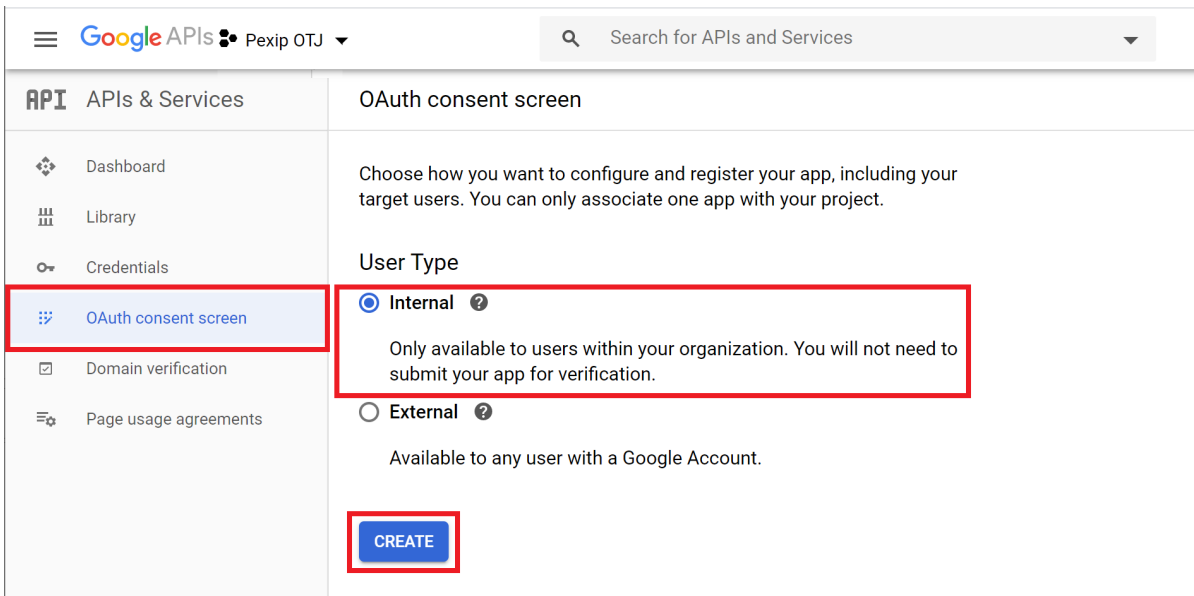


d. Select Enable:



3. Creating an OAuth consent screen:

- a. From <https://console.developers.google.com>, from the left-hand panel select **OAuth consent screen**. Select a **User Type** of **Internal** and then select **Create**:



b. From the OAuth consent screen page:

- under **Application name**, enter a name for your OTJ application
- under **Authorized domains**, enter the domain of the Management Node.

Select Save:

APIs & Services

- Dashboard
- Library
- Credentials
- OAuth consent screen**
- Domain verification
- Page usage agreements

OAuth consent screen

Before your users authenticate, this consent screen will allow them to choose whether they want to grant access to their private data, as well as give them a link to your terms of service and privacy policy. This page configures the consent screen for all applications in this project.

Application type

☐ Public
Any Google Account can grant access to the scopes required by this app.
[Learn more about scopes](#)

☒ Internal
Only users with a Google Account in your organization can grant access to the scopes requested by this app.

Application name ⓘ
The name of the app asking for consent
Pexip OTJ

Application logo ⓘ
An image on the consent screen that will help users recognize your app
Local file for upload

Support email ⓘ
Shown on the consent screen for user support
juliet@pexip.com

Scopes for Google APIs
Scopes allow your application to access your user's private data. [Learn more](#)
If you add a sensitive scope, such as scopes that give you full access to Calendar or Drive, Google will verify your consent screen before it's published.

email
profile
openid

Authorized domains ⓘ
To protect you and your users, Google only allows applications that authenticate using OAuth to use Authorized Domains. Your applications' links must be hosted on Authorized Domains. [Learn more](#)
pexip.com
Type in the domain and press Enter to add it

Application Homepage link
Shown on the consent screen. Must be hosted on an Authorized Domain.
https:// or http://

Application Privacy Policy link
Shown on the consent screen. Must be hosted on an Authorized Domain.
https:// or http://

Application Terms of Service link (Optional)
Shown on the consent screen. Must be hosted on an Authorized Domain.
https:// or http://

About the consent screen

The consent screen tells your users who is requesting access to their data and what kind of data you're asking to access.

OAuth verification

To protect you and your users, your consent screen and application may need to be verified by Google. Verification is required if your app is marked as **Public** and at least one of the following is true:

- Your app uses a sensitive and/or restricted scope
- Your app displays an icon on its OAuth consent screen
- Your app has a large number of authorized domains
- You have made changes to a previously-verified OAuth consent screen

The verification process may take up to several weeks, and you will receive email updates as it progresses. [Learn more](#) about verification.

Before your consent screen and application are verified by Google, you can still test your application with limitations. [Learn more](#) about how your app will behave before it's verified.

[Let us know what you think about our OAuth experience.](#)

OAuth grant limits

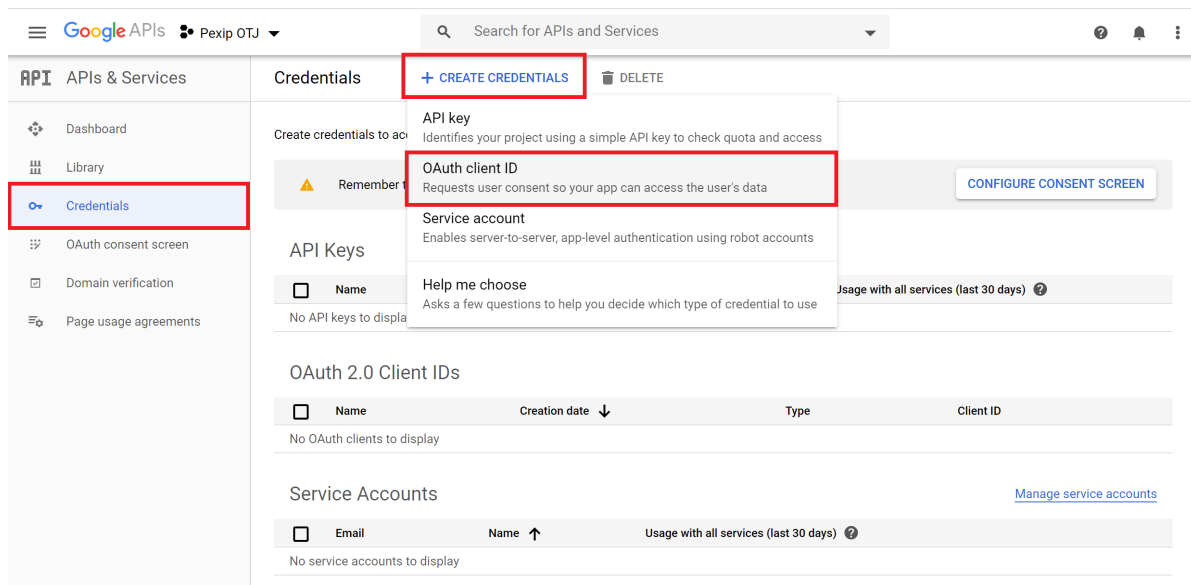
Token grant rate
Your current per minute token grant rate limit is 100 grants per minute. The per minute token grant rate resets every minute. Your current per day token grant rate limit is 10,000 grants per day. The per day token grant rate resets every day.

[Raise limit](#)

1h 6h 1d 7d 30d
Apr 22, 2020 2:22 PM
No data for this time interval

4. Creating the OAuth credentials:

- From <https://console.developers.google.com>, from the left-hand panel select **Credentials** and then select **Create Credentials > OAuth client ID**:



b. From the **Create OAuth client ID** page:, select an **Application type** of **Web application**.

- Enter a **Name** for the application
- under **Authorized redirect URIs**, enter **https://<Management Node FQDN>/admin/platform/mjxgoogledeployment/oauth_redirect/**
 - ❗ This must use the Management Node's FQDN; it cannot use its IP address. You must therefore ensure you have appropriate internal DNS records set up for the Management Node.
 - ❗ The **OAuth Redirect URI** is the page on the Pexip Infinity Administrator interface to which the administrator will be returned after they have successfully signed in to the Google Workspace integration. Because it is a page on the Management Node, this URI is internal to your deployment and only needs to be accessible from the administrator's web browser; you do not need to make it externally accessible.

Select **Create**:

Google APIs docs-otj

Create OAuth client ID

For applications that use the OAuth 2.0 protocol to call Google APIs, you can use an OAuth 2.0 client ID to generate an access token. The token contains a unique identifier. See [Setting up OAuth 2.0](#) for more information.

Application type

- ☒ Web application
- ☐ Android [Learn more](#)
- ☐ Chrome App [Learn more](#)
- ☐ iOS [Learn more](#)
- ☐ Other

Name

Web client 1

Restrictions

Enter JavaScript origins, redirect URIs, or both [Learn More](#)

Origins and redirect domains must be added to the list of Authorized Domains in the [OAuth consent settings](#).

Authorized JavaScript origins

For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (https://*.example.com) or a path (https://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

https://www.example.com

Type in the domain and press Enter to add it

Authorized redirect URIs

For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

https://pexip.com/admin/platform/mjxgoogledeployment/oauth_redirect/

Type in the domain and press Enter to add it

Create Cancel

- c. The following OAuth client created screen will appear. Take note of the **Your Client ID** and **Your Client secret**; you will need these when [Adding a One-Touch Join Google Workspace integration on Pexip Infinity](#) on the Management Node:

OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services

i OAuth access is restricted to users within your organization unless the [OAuth consent screen](#) is published and verified.

Your Client ID

953250980346-3girlk9isqcp2157391g29m286km3ris.apps.gc

Your Client Secret

SErHm3girlk9HTwyEk6gNJq

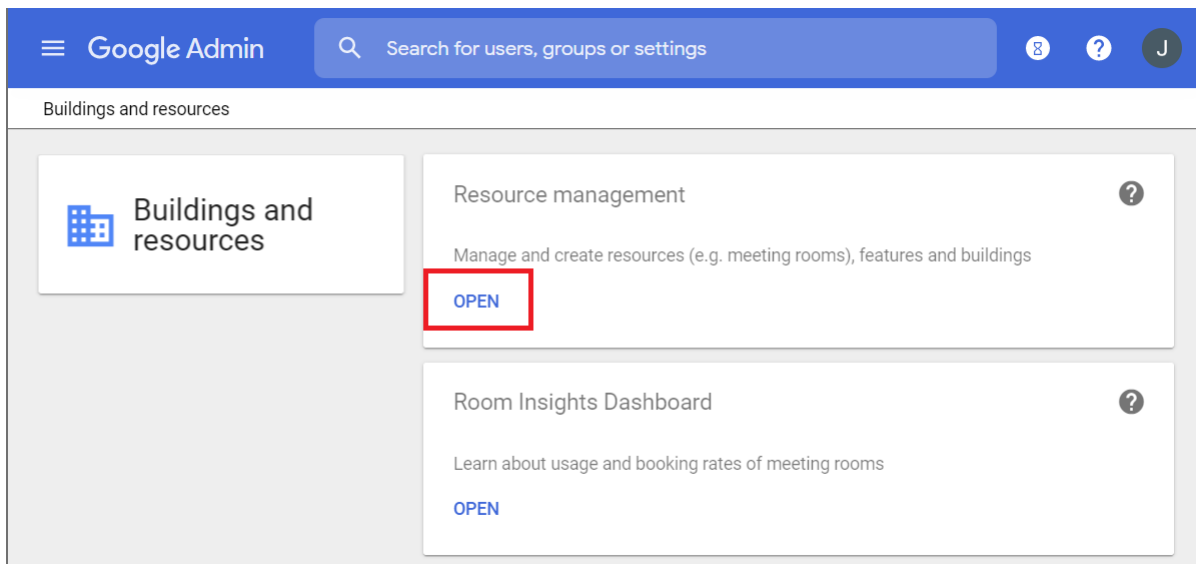
OK

Creating a room resource

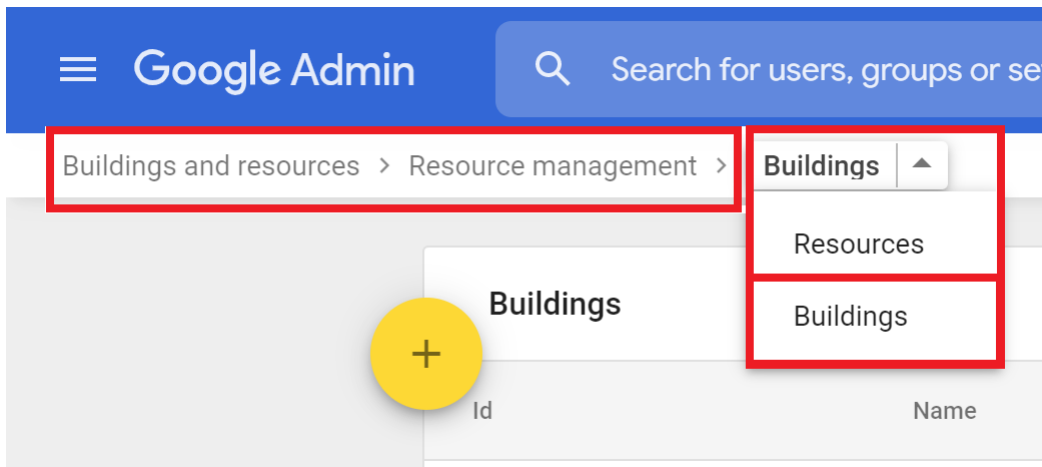
(Required only if your room resources do not already exist - otherwise you can skip this step.)

In this step, you create a room resource in Google Workspace for each physical room that is to be used for One-Touch Join. Google Workspace will automatically assign an email address to the room.

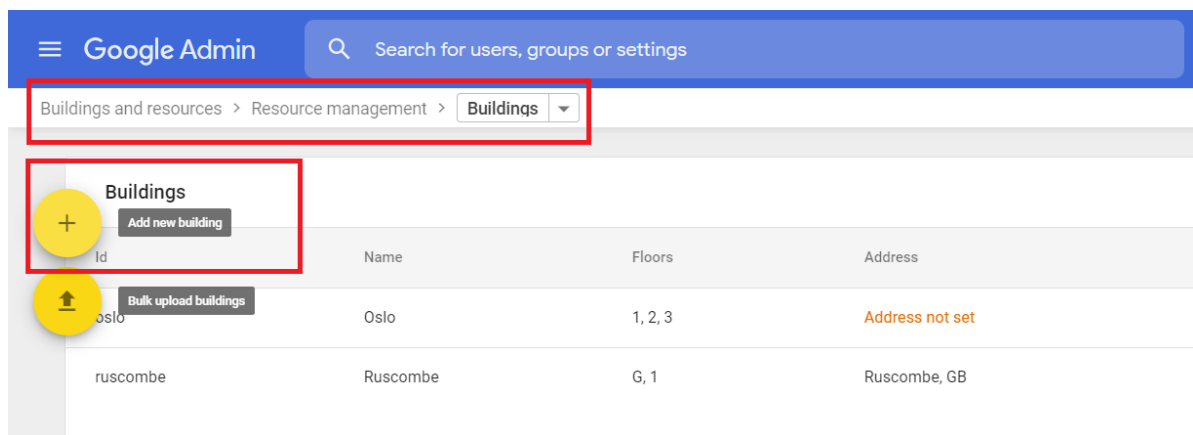
1. If a **building** for the room resource does not already exist, create one as follows:
 - a. Go to <https://admin.google.com> (logged in as a Google Workspace administrator).
 - b. Select the **Buildings and resources** tile, and then from the **Resource management** section select **Open**:



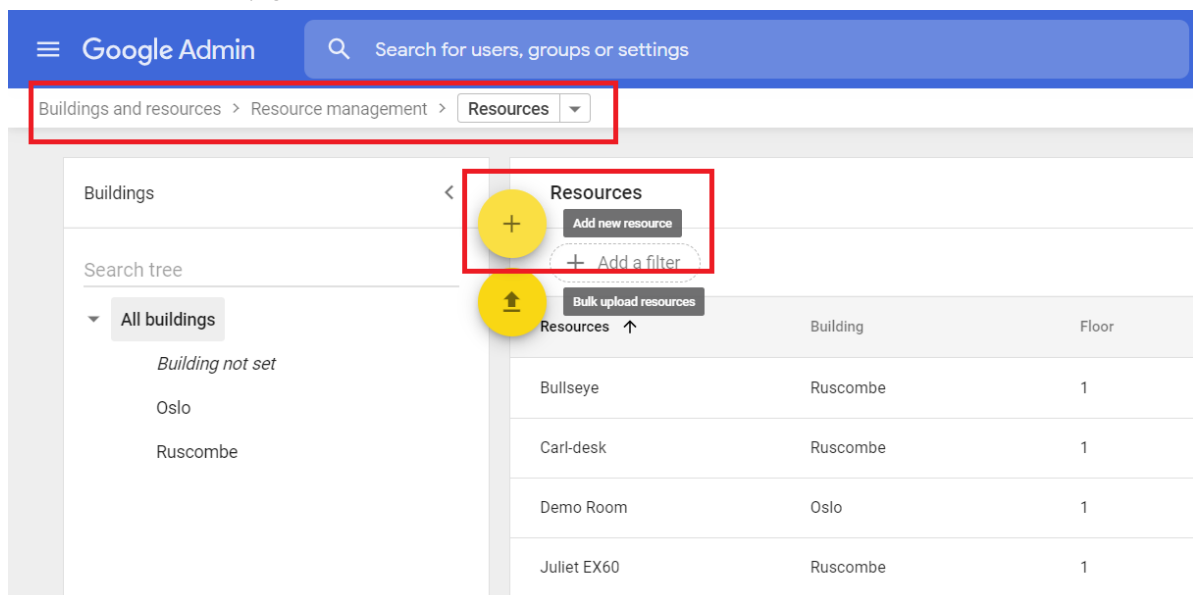
From the drop-down along the top left of the screen, select **Buildings**:



- c. Select + to Add new building:



- d. Enter a Name and the list of Floors, and select **Add Building**.
2. Create the room resource:
 - a. Go back to the **Resources** page and Select + to **Add new resource**:



- b. For the Category, select *Meeting space (room, phone booth,...)*.

- c. Select the **Building** and **Floor** in which the room is located, enter a **Name** and the room's **Capacity**, then select **Add Resource**:

Add resource

Auto-generated resource name
Oslo-2-Board room (20)

Category * ⓘ
Meeting space (room, phone booth,...)

Type
meeting room

Building *
Oslo

Floor *
2

Floor section

Resource name *
Board room


Capacity *
20

Features
Define features first to add them.

User visible description

[Add description \(internal\)](#)

* indicates a required field

 Setting resource information, such as building, floor or capacity, affects the auto-generated resource names and room search. [Learn more.](#)

CANCEL

ADD RESOURCE

The resource will be created and added to the list. You can click on the new resource to view information about it, such as the email address it was automatically assigned.

-  For more information on setting up buildings and other resources in Google Workspace, including how to add buildings and resource in bulk and using CSV imports, see <https://support.google.com/a/answer/1033925>.

Configuring the room resource

In these steps, you allow the authorization user to access each calendar of each room resource that you want to use for One-Touch Join, and set the calendar to auto-accept invitations. We also recommend that you make the calendar available to all users in your domain in such a way that allows them to book meetings using the resource, without being able to view the details of any other meetings in the resource's calendar.

Sharing individual calendars with the authorization user

Note that the Google calendar API limits the number of calendars that can be shared within a 24 hour period to 750 (for more information, see <https://support.google.com/a/answer/2905486?hl=en>). This means that if you have more than 750 room resources that you wish to use for One-Touch Join, they will need to be set up over a period of days.

To share calendars with the authorization user:

1. Go to <https://calendar.google.com> (logged in as a Google Workspace administrator so that you have permission to share the calendars).
2. From the left-hand panel, select the + next to **Other calendars** and then select **Browse resources**.

- Expand the sections if necessary, and tick the boxes of all the room resources whose calendars you want to share with the authorization user.

This will add the room resources to the **Settings for other calendars** section in the left-hand panel.

- For each of the rooms:
 - From the **Settings for my calendars** section, select the room resource and then select **Share with specific people**.
 - Select **Add people**.
 - In the **Share with specific people** dialog, enter the email address of the One-Touch Join authorization user. Ensure the **Permissions** are set to either:
 - Make changes to events** (if you want users to be able to use OTJ to join all meetings, including private meetings, from this endpoint)
 - See all event details** (if you don't want to offer OTJ for private meetings on this endpoint).
- i** If your deployment includes personal endpoints that are associated with a user's personal calendar, then either you or the end user will need to ensure that their calendar allows the One-Touch Join authorization user to **Make changes to events** if they wish to use OTJ to join their own private meetings from their endpoint.

← Settings

- i** For more information on sharing room and resource calendars in Google Workspace, see <https://support.google.com/a/answer/1034381>.

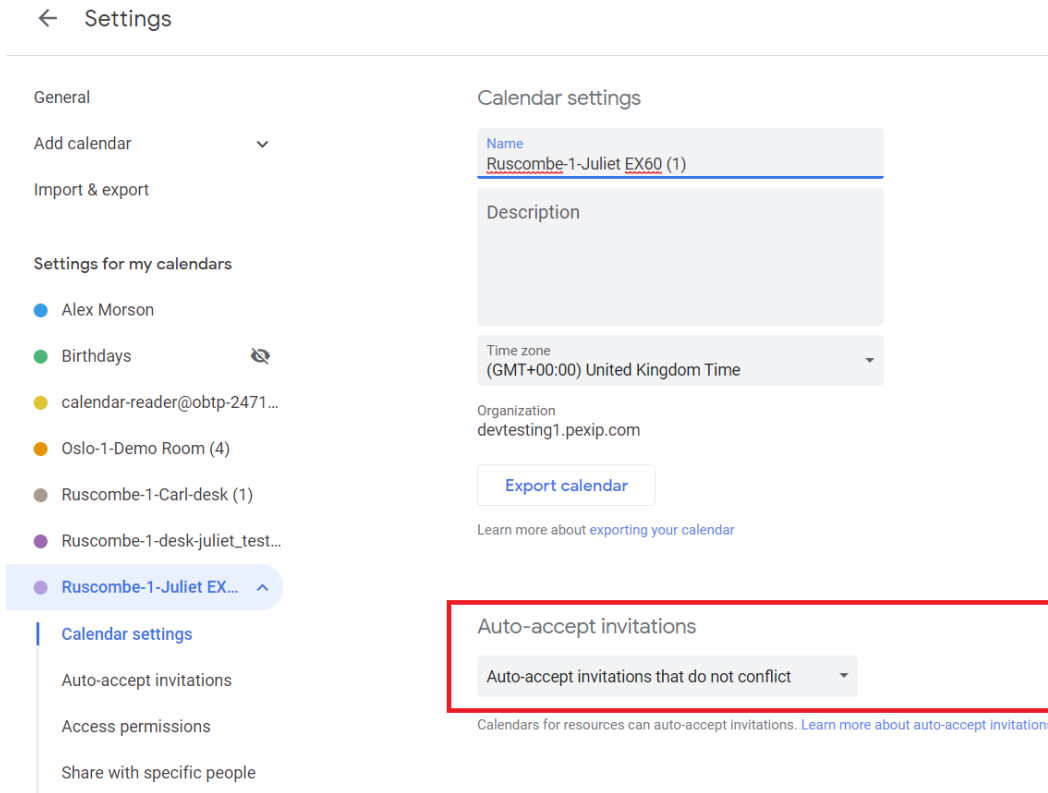
Auto-accepting invitations

By default, when creating room resources in Google Workspace, calendar processing is set to **Auto-accept invitations that do not conflict**. You must ensure you keep this setting for all room resources, so that the room will automatically accept meeting requests if it is available, and automatically decline an invitation if it is already booked.

To check this setting:

- Go to <https://calendar.google.com> (logged in as a Google Workspace administrator so that you have permission to share the calendars).
- From the left-hand panel, select the room resource and select **Settings and sharing**.

3. In the **Auto-accept invitations** section, ensure that *Auto-accept invitations that do not conflict* is selected:



Allowing users to book resources

We recommend that you configure your Google Workspace calendar settings to allow end users to book a room resource without seeing details of the room's other bookings. To do this, you configure the room resource's calendar so that all users in your domain have permission to see its free/busy status, without being able to see the invitation details. You then on a global basis permit users to book resources to which they have free/busy access.

To do this:

1. Go to <https://calendar.google.com> (logged in as a Google Workspace administrator so that you have permission to share the calendars).
2. From the left-hand panel, select the room resource and select **Settings and sharing**.
3. In the **Access permissions** section, select **Make available for <your domain>**, and ensure that *See only free/busy (hide details)* is selected:

Auto-accept invitations

Auto-accept invitations that do not conflict ▼

Calendars for resources can auto-accept invitations. [Learn more about auto-accept invitations](#)

Access permissions

☐ Make available to public

See all event details ▼

☒ Make available for devtesting1.pexip.com

See only free/busy (hide details) ▼

[Get shareable link](#)Learn more about [sharing your calendar](#)

Share with specific people



otj-calendar-reader@pexip-otj-270111.iam.gserviceaccount.co...

See all event details ▼

[+ Add people](#)Learn more about [sharing your calendar with someone](#)

4. Go to admin.google.com (logged in as a Google Workspace administrator).
5. From the left-hand menu, select **Apps** > **Google Workspace** > **Calendar**.
6. Scroll down to **General Settings** and select **Resource Booking Permissions**.
7. Ensure that **Allow users to book resources that are shared as See only free/busy** is set to **ON**:

The screenshot shows the Google Admin console interface. At the top, there's a blue header with 'Google Admin' and a search bar. Below the header, a breadcrumb trail reads 'Apps > Google Workspace > Settings for Calendar > General settings'. On the left, there's a sidebar with a 'Calendar' icon and label. The main content area is titled 'General settings' and contains three sections: 'External sharing options for secondary calendars', 'Internal sharing options for secondary calendars', and 'Resource booking permissions'. The 'Resource booking permissions' section is highlighted with a red rectangular box. It shows the setting 'Allow users to book resources that are shared as "See only free/busy"' is set to 'ON'. A link 'Learn more' is provided next to the setting.


Updating the per-user request quota

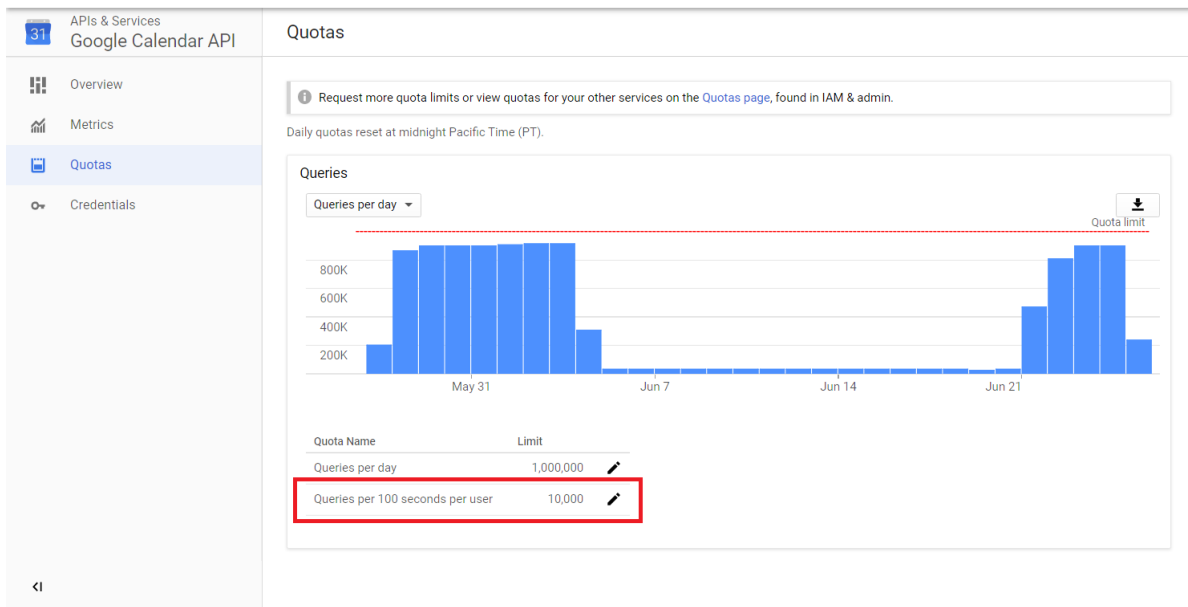
In this step you increase the limit on the number of queries per 100 seconds per user to the Google Calendar API.

The default number of queries per 100 seconds per user is 500. In this context, the "user" is the authorization user. In deployments with fewer than around 180 rooms, each room resource calendar is queried every 30 seconds by two conferencing nodes (both using the same authorization user account), resulting in 5,760 queries per room per day. (In larger deployments, room resource calendars are queried less frequently.)

We recommend that you increase the number of queries per 100 seconds per user to 10,000 to provide sufficient processing overhead and room for expansion (there is currently no additional cost to this increase).

To increase this quota to 10,000:

1. Go to <https://console.developers.google.com> (logged in as a Google Workspace administrator).
2. From the top left of the page, select the project you created for One-Touch Join:

3. From the navigation menu at the top left of the page, select **IAM & Admin > Quotas**.
4. From the Quotas page, select **Edit Quotas** and then select **Google Calendar API - Queries per 100 seconds per user**. You will be taken to the **Google Calendar API > Quotas** page.
5. Change **Queries per 100 seconds per user** to **10,000**:



- i** You may also need to request an increase to the number of **Queries per day** for larger deployments - for more information, see [Requesting an increase to API limits](#).

Requesting an increase to API limits

This optional step applies to larger deployments only (more than around 170 room resources), and should be performed if you wish to reduce the amount of time taken for endpoints to be updated with additions or changes to their corresponding room resource calendar.

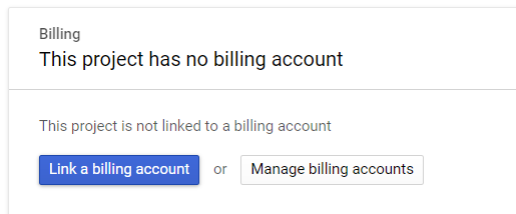
The maximum frequency with which an endpoint will be updated with meeting information is every 30 seconds. For deployments with more than around 170 endpoints, this frequency will decrease in line with the number of endpoints (up to around 20 minutes for deployments with around 6,000 endpoints). This is due to a limit on the number of Calendar API requests permitted by Google in a 24-hour period — for more information, see <https://developers.google.com/calendar/pricing>.

To reduce the time taken to update endpoints in these larger deployments, you can request an increase to the number of Calendar API requests One-Touch Join can make.

- i** When your request has been implemented by Google, you must then increase the [Maximum Google Workspace API requests](#) on Pexip Infinity in order to take advantage of the increase.

To request an increase to the API limits:

1. If you do not already have one, create a Cloud Billing Account (note that this is different from a Google Workspace billing account). Full instructions are available via https://cloud.google.com/billing/docs/how-to/manage-billing-account#create_a_new_billing_account.
2. Link the Cloud Billing Account to the project you created when [Creating a service account](#):
 - a. Go to <https://console.developers.google.com> (logged in as a Google Workspace administrator).
 - b. Ensure that the project shown in the top left corner is the one you created for One-Touch Join when [Creating a service account](#).
 - c. Select the burger menu from the top left of the page and select **Billing**. When the following message appears, select **Link a billing account**:



- d. Select the account to link to:

Set the billing account for project “Quickstart”

Billing account ?

There is only one billing account currently available to link this project to

My Billing Account

CANCEL

SET ACCOUNT

3. Request an increase to your quota:
 - a. From the navigation menu at the top left of the page, select **IAM & admin > Quotas**.
 - b. From the Quotas page, select **Edit Quotas** and then select **Google Calendar API**.
In the panel that appears on the right, enter the **New quota limit** that you wish to request, and in the **Request description** field, enter the reason for requesting the increase:

The screenshot shows the Google Cloud IAM & Admin console. On the left, the 'IAM & admin' menu is open, and 'Quotas' is selected. The main panel shows the 'Quotas' page with a table of quotas. The 'Google Calendar API' quota is selected, and the 'New quota limit' is being set to 100,000. The 'Request description' field is also visible.

Quota type	Service	Metric	Location	Current Usage	7 Day Peak Usage	Limit
<input checked="" type="checkbox"/> Google Calendar API	Queries per day	Global	5,058	23,377	1,000,000	
<input type="checkbox"/> Gmail API	Queries per day	Global	0	0	1,000,000,000	
<input type="checkbox"/> Gmail API	Queries per 100 seconds	Global	0	0	2,000,000	
<input type="checkbox"/> Gmail API	Discovery requests per 100 seconds	Global	0	0	17,000	
<input type="checkbox"/> Google Calendar API	Queries per 100 seconds per user	Global	-	-	100,000,000	
<input type="checkbox"/> Gmail API	Queries per 100 seconds per user	Global	-	-	25,000	

Quota: Queries per day

New quota limit
Enter a new quota limit. A limit above 1,000,000 will require approval from your service provider.

100000

Request description
Required
Enter a description about your request

Done Cancel

Submit request Back

c. Select **Submit request**.

Quota increase requests typically take two business days to process.

Adding a One-Touch Join Google Workspace integration on Pexip Infinity

In this step you configure Pexip Infinity with details of the Google Workspace deployment configured above. You must then log in to Google Workspace as the authorization user and grant the One-Touch Join app access to the room resource calendars.

Configuring the Google Workspace integration

From the Pexip Infinity Administrator interface, go to **One-Touch Join > OTJ Google Workspace Integrations**.

Option	Description
Name	The name of this One-Touch JoinGoogle Workspace integration.
Description	An optional description of this One-Touch JoinGoogle Workspace integration.
Account email	<p>If you are authorizing using a service account, enter the email address of the service account that One-Touch Join will use to log in to Google Workspace.</p> <p>If you are authorizing using a Google Workspace domain user, enter the email address of the user.</p>
Enable user authorization	<p>If you are authorizing using a service account — the recommended method — this should be left blank.</p> <p>Select this option only if you will be authorizing using a Google Workspace domain user.</p>
Client ID	<p>(Available when user consent authorization has been enabled)</p> <p>The client ID of the application you created in the Google API Console, for use by OTJ.</p>
Client secret	<p>(Available when user consent authorization has been enabled)</p> <p>The client secret of the application you created in the Google API Console, for use by OTJ.</p>
Redirect URI	<p>(Available when user consent authorization has been enabled)</p> <p>The redirect URI you configured in the Google API Console. It must be in the format: <a href="https://<Management Node FQDN>/admin/platform/mjxgoogledeployment/oauth_redirect/">https://<Management Node FQDN>/admin/platform/mjxgoogledeployment/oauth_redirect/</p> <p>i This must use the Management Node's FQDN; it cannot use its IP address. You must therefore ensure you have appropriate internal DNS records set up for the Management Node.</p>

Advanced options

Option	Description
Maximum Google Workspace API requests	<p>The maximum number of API requests that can be made by One-Touch Join to your Google Workspace Domain in a 24-hour period.</p> <p>We recommend you set this value to 90% of your total permitted requests. Google's default is 1,000,000 so by default this is set to 900,000 on Pexip Infinity. If you increase the number of API requests, you should also increase this setting to 90% of that number.</p> <p>For more information, see Frequency and limitations on calendar requests.</p>
Google OAuth 2.0 endpoint	The URI of the Google OAuth 2.0 endpoint.
Google authorization server	The URI of the Google authorization server.

When you have completed the above fields, select **Save**. You will be returned to the main OTJ Google Workspace Integration page. You must now authorize calendar API access to the Google Workspace Integration using the account details you have just created, using the following steps.

Authorizing calendar access

If you have enabled OAuth for the first time, after saving the configuration of the One-Touch Join Google Workspace integration you must sign in to Google Workspace as the authorization user.

You may also need to re-sign in to the authorization user account if:

- you disable and then subsequently re-enable OAuth
- you update any of the following configuration for the One-Touch Join Google Workspace integration:
 - Account email
 - Client ID
 - Client secret
 - Google OAuth 2.0 endpoint
 - Google authorization server
- the refresh token has expired (for more information about when this might happen, see <https://developers.google.com/identity/protocols/oauth2#expiration>).

To sign in to Google Workspace as the authorization user:

1. Ensure you have signed out of all Google accounts on your device.
2. From the Management Node, go to **One-touch Join > OTJ Google Workspace Integrations** and select the Google Workspace integration you have just created. At the bottom of the **Change OTJ Google Workspace Integration** page, select **Authorize calendar API access**:

Enable user consent authorization ☒

Enable this option to authorize google api access through user consent OAuth 2.0. Leave this option disabled to continue using service account to access google api.

Client ID

The client ID for the application you created in the Google API Console Credentials page. Maximum length: 250 characters.

Client secret

The client secret for the application you created in the Google API Console Credentials page.

Redirect URI

Currently: https://pexip.com/admin/platform/mjxgoogledeployment/oauth_redirect/

Change:

The redirect URI you configured in your client's API Console Credentials page. It should be in the format 'https://[Management Node Address]/admin/platform/mjxgoo

Advanced options (Show)

[Save](#) [Save and add another](#) [Authorize calendar API access](#)

- You will be taken to the **Authorize Calendar API access** page. Select **Authorize**:

]pexip[Infinity Conferencing Platform

Status ▾ History & Logs ▾ System ▾ Platform ▾ Call Control ▾ Services ▾ Users & Devices ▾ **One-Touch Join ▾** Utilities ▾

Authorize Calendar API access

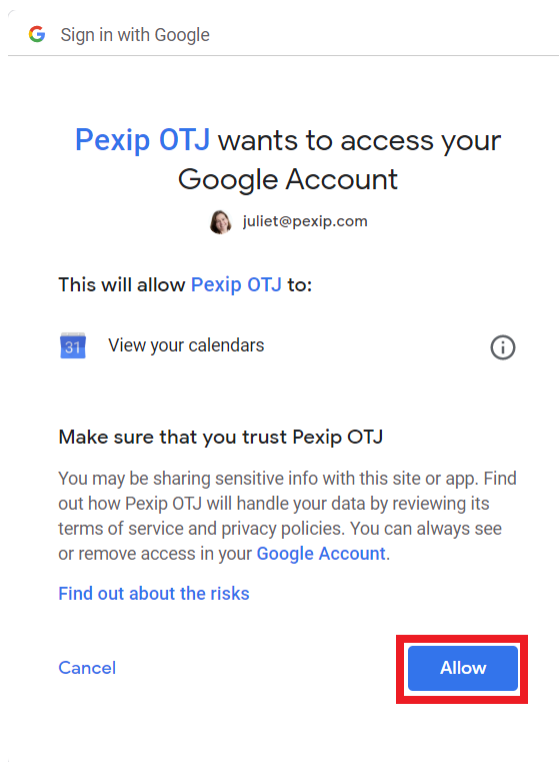
Please open the link below. It will take you to a Google consent page where you must sign in as the google account with username **juliet@pexip.com**.
You can then consent to grant calendar access requested by the G Suite Integration.

Authorize link

Authorize

https://accounts.google.com/o/oauth2/v2/auth?prompt=consent&access_type=offline&state=oauth2

- Enter the email address of the authorization user (which you previously entered as the **Account email**) and sign in.
- At the consent screen, **Allow** the Pexip OTJ app to **View your calendars**:



6. You may be asked to sign in to the Management Node again. If so, you must sign in to the Management Node (using your Management Node credentials) to complete the process of signing in as the authorization user.

When complete, you are returned to the **Authorize Calendar API access** page and see the message **Successfully authorized**.

Next steps

You must now configure the remainder of the One-Touch Join components on Pexip Infinity, as described in [Configuring Pexip Infinity for One-Touch Join](#).

Troubleshooting One-Touch Join

This section provides guidance on troubleshooting issues with Pexip Infinity's One-Touch Join feature, including issues specific to [Cisco](#) or [Poly](#) endpoints.

For guidance on the troubleshooting of general issues, see [Troubleshooting the Pexip Infinity platform](#).

Symptom	Possible cause	Resolution
One-Touch Join issues		
A meeting has been scheduled and is showing on the room endpoint, but there is no Join button.	One-Touch Join has not been able to obtain a meeting room alias from the invitation because it does not match a meeting processing rule and does not contain a URI or address prefixed with <code>sip:</code> , <code>sips:</code> or <code>h323:</code> .	Review the meeting processing rules.
	The meeting processing rule that you expected to match is associated with a different OTJ profile than the endpoint. For example, the endpoint has an Exchange email address and is associated with an Exchange integration, but the rule that the meeting matches is associated with a Google Workspace integration, or vice versa.	<ol style="list-style-type: none"> 1. Check that the OTJ Endpoint has been associated with an OTJ Endpoint Group. 2. Check that the OTJ Endpoint Group is associated with the same OTJ Profile as the Meeting Processing Rule that you expected to match.
	The meeting is not a video meeting.	If you do not want non-video meetings to appear on the room endpoint, you can disable the Enable non-video meetings option.
	One-Touch Join has not been able to obtain a meeting room alias because the URL in the invitation has been rewritten for security reasons (for example, by Safe Links in Microsoft Defender for Office 365, or Proofpoint's URL Defense) and therefore does not match the default rule.	<ul style="list-style-type: none"> • Modify your Safe Links policy using the "Do not rewrite the following URLs" list so that URLs in meeting invitations sent to OTJ room resources are not rewritten. • Add a custom rule to match the rewritten URLs. See Extracting an alias from a URL that has been rewritten for security for an example.
	The meeting was scheduled using the Microsoft Teams plugin for Google Workspace.	This is due to a known issue with the Microsoft Teams plugin for Google Workspace whereby it does not include the required CVI information in the meeting body.
A meeting has been scheduled and is showing on the room endpoint, but either there is no Join button, or the Join button appeared and then disappeared.	The endpoint is being managed by Webex Cloud Calendar or TMS XE, and these systems are overriding the meeting information from One-Touch Join.	Ensure that any endpoints used for One-Touch Join are not also registered to the calendaring service on other systems such as the cloud-based Webex Hybrid Calendar Service, or Cisco TMS XE.
Meetings are being deleted from an endpoint that is managed by TMS, without TMS XE.	There is a known bug (CSCvv93408) with TMS version 15.9 and later whereby TMS will erroneously replace meetings that have been pushed to the endpoint using the endpoint's API.	<p>Ensure that the following configuration for the endpoint has been made in TMS:</p> <ul style="list-style-type: none"> • Disable Allow booking for the endpoint • Change Meeting Type to Reservation. <p>If the problem persists, we recommend removing the endpoint from TMS until this bug is fixed by Cisco.</p>

Symptom	Possible cause	Resolution
<p>A meeting has been scheduled and is showing on the room endpoint, but there is no Join button. The support log shows the message:</p> <pre>Could not find an alias for this meeting which had no body. This could be a meeting room configuration issue.</pre>	<p>One-Touch Join has not been able to obtain a meeting room alias from the invitation because the meeting information supplied in the body ("description") of the invitation has been stripped by Exchange prior to One-Touch Join processing the meeting.</p>	<p>Change the calendar processing rules for the room to ensure that the meeting body is not deleted. For instructions on how to do this, see either Configuring calendar processing (for Exchange on-premises) or Configuring calendar processing (for O365).</p>
<p>An external Microsoft Teams meeting has been scheduled but there is no Join button.</p>	<p>Your Microsoft Exchange environment uses a security application (such as Office 365 ATP, or Mimecast) to re-write URLs, meaning that One-Touch Join has not been able to obtain the join URL. For more information, see Allowing forwarding of external invitations (for Exchange on-premises) or Allowing forwarding of external invitations (for O365).</p>	<p>Ensure that the security application's URL re-write rules include an exception for any URL starting with the domain <code>https://teams.microsoft.com/</code></p>
<p>There is a delay between a meeting invitation being sent and it appearing on the room endpoint.</p>	<p>A short delay is expected due to internal processing, and the actual time taken will depend on the number of endpoints in your One-Touch Join deployment, and the number of daily API requests you are allowed to make to your calendar service. Limits are also imposed so that Conferencing Nodes do not become overloaded with One-Touch Join requests.</p> <p>For more information, see Frequency and limitations on calendar requests.</p>	<p>For larger Google Workspace integrations you can ask for an increase to the number of calendar API requests you can make in a 24-hour period, thus allowing you to update endpoints more frequently. For more information, see Requesting an increase to API limits.</p> <p>You could also consider Deploying a dedicated One-Touch Join platform.</p>
<p>On the status page and logs, the Alias field is blank.</p>	<p>Process alias for private meetings has been disabled and the meeting was flagged as private.</p> <p>Enable non-video meetings has been enabled, but OTJ was not able to obtain a valid alias for the meeting.</p> <p>This may be because Exchange is using default calendar processing, which removes the header and body of the invitation, and replaces the subject with the organizer's name.</p>	<p>Review whether these settings are appropriate for your deployment.</p> <p>Ensure that Exchange calendar processing properties are changed from the default, as per the instructions in Configuring calendar processing on room resource mailboxes.</p>
<p>On the status page and logs, the Subject field is showing the organizer's name.</p>	<p>Replace subject has been set to either:</p> <ul style="list-style-type: none"> • <i>Private meetings only</i> (and the meeting was flagged as private), or • <i>Always</i> <p>and the Replace subject string was empty.</p> <p>Replace empty subject has been enabled and there was no subject.</p> <p>This may be because Exchange is using default calendar processing, which removes the header and body of the invitation, and replaces the subject with the organizer's name.</p>	<p>Review whether these settings are appropriate for your deployment.</p> <p>Ensure that Exchange calendar processing properties are changed from the default, as per the instructions in Configuring calendar processing on room resource mailboxes.</p>

Symptom	Possible cause	Resolution
An endpoint has been deleted from the Pexip Infinity configuration but its details are still appearing on the OTJ Endpoints status page.	The status page is refreshed once an hour.	Wait up to one hour for the endpoint's details to be removed.
A meeting that has been canceled is still appearing on the OTJ Meetings status page.	The status page is refreshed once an hour.	Wait up to one hour for the meeting's details to be removed.
When configuring Exchange you are getting the following errors or warnings: <code>ErrorCode="InvalidUser"</code> <code>ErrorMessage="Invalid user"</code>	The service account being used for One-Touch Join does not exist, or does not have a valid license.	<ul style="list-style-type: none"> Ensure that the service account has been added correctly, with the correct username and password/authentication information. Ensure that the service account has an appropriate Exchange license, such as Office 365 Enterprise E1, Office 365 Business Basic (formerly Essentials) or one of the Exchange Online plans.
An endpoint is not appearing on the OTJ Endpoints status page.	Endpoints appear on this page after the initial contact between the endpoint and One-Touch Join has been made. For Cisco endpoints, this happens when a meeting has successfully been pushed to the endpoint. For Poly endpoints, this happens when the Poly endpoint has successfully polled the Conferencing Node.	Wait until a meeting has been created to which the endpoint is invited. Then check again to confirm that the endpoint is appearing on the status page.
Cisco endpoint issues		
One-Touch Join cannot contact an endpoint via its API. The following appears in the alarms and logs: <code>Non-200 status code returned when trying to upload OBTP bookings to endpoint and</code> <code>StatusCode="307"</code>	One-Touch Join is configured to communicate with the endpoint via HTTP and the endpoint redirects to HTTPS.	Configure One-Touch Join to use HTTPS to communicate with the endpoint .
A Cisco SX series endpoint running TC software may display the "Meeting will automatically connect" message if there is no URI in the meeting invitation.	This is a known issue with the Cisco endpoint when running this software.	
Webex endpoint issues		
Meetings are not appearing on the Webex endpoint.	The endpoint is configured to use a Webex Calendar service. This will prevent Pexip One-Touch Join from being able to push meetings to the endpoint.	Disable the calendar .
Poly endpoint issues		

Symptom	Possible cause	Resolution
Meetings are not appearing on the Poly endpoint.	The configuration for the endpoint on Pexip Infinity or on the endpoint itself is incorrect.	Ensure that the configuration for endpoint on Pexip Infinity and on the endpoint itself is correct, in particular that the username and password configured on both match. Ensure that the endpoint is showing as registered to the calendaring service.
	The Poly endpoint is registered to the calendaring service but One-Touch Join hasn't found any meetings.	View the Meeting status page to see if any meetings have been found for this endpoint. Check for any Google Gatherer/Exchange Gatherer alarms, which would indicate issues with reading specific calendars.
	<p>The Poly endpoint has lost connection with the OTJ calendaring service and has become unregistered, meaning it is no longer receiving updated meeting information.</p> <p>To check if there is still contact with the endpoint:</p> <ul style="list-style-type: none">• If Raise alarms is enabled for this endpoint, an OTJ Poly Endpoint Error alarm will appear on the Pexip Infinity Administrator interface if it has been more than 10 minutes since there was contact with the endpoint.• If this option is not enabled, view the Endpoint status and check the last contact time. If this is more than 10 minutes ago the endpoint may have lost connection.	On the Poly endpoint, disable and re-enable the calendaring service.