



Pexip One-Touch Join

Deployment Guide

Software Version 23

Document Version 23.a

March 2020

] pexip [

Contents

About One-Touch Join	5
Supported G Suite editions	5
Supported Exchange environments	5
Exchange servers	5
Outlook clients	5
Supported endpoints	5
Supported number of endpoints and Conferencing Nodes	6
Supported meeting types	6
Pexip Infinity server requirements	6
Enabling One-Touch Join	7
Process overview	7
Frequency and limitations on calendar requests	8
Locations, Conferencing Nodes and One-Touch Join	8
Network architecture and firewalls	8
Permitting the service account to access calendars	9
Exchange integrations	9
G Suite integrations	9
Configuring Exchange on-premises for One-Touch Join	11
Prerequisites	11
Creating a service account	11
Configuring Application Impersonation on the service account	13
Creating a new Distribution Group	13
Configuring application impersonation	13
Enabling basic authentication	14
Configuring calendar processing on room resource mailboxes	14
Recommended configuration	14
Optional configuration	15
Checking calendar processing settings	15
Adding a One-Touch Join Exchange integration on Pexip Infinity	16
Next steps	17
Configuring Office 365 for One-Touch Join	18
Prerequisites	18
Creating a service account	18
Configuring Application Impersonation on the service account	19
Creating a new Distribution Group	19

Configuring application impersonation	20
Enabling OAuth authentication	20
Configuring calendar processing on room resource mailboxes	24
Recommended configuration	24
Optional configuration	25
Checking calendar processing settings	25
Adding a One-Touch Join Exchange integration on Pexip Infinity	26
Configuring the Exchange integration	26
Signing in to the service account	28
Next steps	29
Configuring G Suite for One-Touch Join	30
Creating a Service Account	30
Creating a room resource	33
Configuring the room resource	34
Sharing individual calendars with the Service Account	34
Auto-accepting invitations	35
Allowing users to book resources	36
Requesting an increase to API limits	37
Adding a One-Touch Join G Suite integration on Pexip Infinity	38
Next steps	38
Configuring Pexip Infinity for One-Touch Join	39
Prerequisites	39
Adding a One-Touch Join profile	39
Adding One-Touch Join endpoint groups	41
Adding One-Touch Join endpoints	41
Adding endpoints individually	42
Adding OTJ endpoints in bulk	42
Adding One-Touch Join meeting processing rules	44
Testing the rule	46
One-Touch Join meeting types and transform rules	47
Supported meeting types	47
Custom meeting type	49
Examples	50
Deploying a dedicated One-Touch Join platform	51
Minimum hardware requirements	51
Minimum Pexip Infinity platform configuration	51
One-Touch Join configuration	52

Scheduling and joining meetings using One-Touch Join	53
Viewing One-Touch Join meetings	54
Troubleshooting One-Touch Join	55
Adding a temporary custom rule for debugging purposes	56

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 endpoint (such as Cisco's One Button to Push - OBTP), 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. However, in some situations you may wish 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](#).

Supported G Suite editions

Pexip One-Touch Join is supported in the following G Suite environments:

- G Suite Basic
- G Suite Business
- G Suite 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

Pexip Infinity One-Touch Join is supported on the Cisco VTC endpoints that support Cisco One Button to Push (OBTP) and are running either TC or CE* 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 VTC systems (SX10, SX20, SX80)

* Endpoints registered to WebEx cloud must be using Webex Edge for Devices.

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 deployments** of up to 170 room resource calendars and associated endpoints, a single Conferencing Node per location should suffice (although you may wish to include one or more additional Conferencing Nodes for redundancy). For more than 170 endpoints, to minimize the impact of One-Touch Join on other Pexip Infinity services, we recommend that each location that contains One-Touch Join endpoints contains at least 5 Conferencing Nodes (plus one or more nodes to provide redundancy). These recommendations apply to deployments with one or two OTJ Integrations. For deployments with multiple One-Touch Join Integrations (for example, when implemented by service providers with multiple customers) we recommend a dedicated One-Touch Join deployment.

For **dedicated 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).

We recommend that you consult with your Pexip authorized support representative for advice regarding large or busy deployments.

 For information on the hardware requirements for OTJ, see [Pexip Infinity server requirements](#).

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 Hangouts Meet (for G Suite integrations only)
- Microsoft Teams
- Skype for Business
- Webex
- Zoom
- BlueJeans
- GoToMeeting

Pexip Infinity server requirements

In most cases you will be enabling One-Touch Join within a new or existing Pexip Infinity deployment, and the 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.

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.

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 it 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 two steps, described in separate topics:

1. Depending on which calendar/email service is used in your environment, do one of the following:
 - [Configuring G Suite for One-Touch Join](#)
 - [Configuring Exchange on-premises for One-Touch Join](#)
 - [Configuring Office 365 for One-Touch Join](#)
2. [Configuring Pexip Infinity for One-Touch Join](#)

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

Process overview

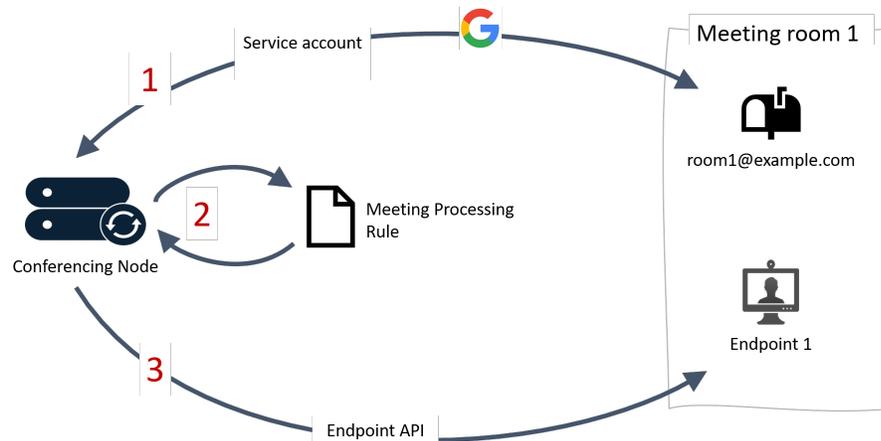
The administrator configures their [G Suite](#), [Exchange on-premises](#) or [Office 365](#) deployment to support Pexip Infinity One-Touch Join, and ensures that each physical meeting room that contains an endpoint to be used for One-Touch Join has an associated email address.

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.

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.

- ⓘ 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.

The flow of information is shown in the following diagram (using G Suite as the calendar/email service):



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 G Suite or Microsoft Exchange and uses the configured service account details to impersonate 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 logs into the endpoint that is associated with the room resource, using the endpoint's API, and updates it with the meeting information.
 - 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.

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.

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 and update endpoints 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.)

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 G Suite deployments, you can change the 24-hour quota by [Requesting an increase to API limits](#) and then increasing the [Maximum G Suite 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 One-Touch Join

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 nodes will actively handle One-Touch Join; the One-Touch Join service on the remainder will remain inactive unless one of the first five nodes becomes unavailable.

Each room resource has one Conferencing Node which will be its primary node, and another that is its secondary node. The primary and secondary nodes both connect to Exchange or G Suite to get the meetings for that room resource. This is done so that should the primary node become unavailable, the secondary node will become the primary node for that room resource and the transition will be smooth because the secondary node already knows about the meeting rooms it is now responsible for.

When maintenance mode is enabled on a Conferencing Node, or the node goes offline, it will no longer act as the primary node for its room resources. Each of the remaining Conferencing Nodes in that location will be aware that the original node is no longer available, and will become primary node for any of the room resources for which it was the secondary node, thus reallocating the node's room resources between them. Note that if you put all Conferencing Nodes in a One-Touch Join location into maintenance mode, then none of the endpoints in the associated Endpoint Group will receive any updates (overflow locations are not used by One-Touch Join).

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.

Network architecture and firewalls

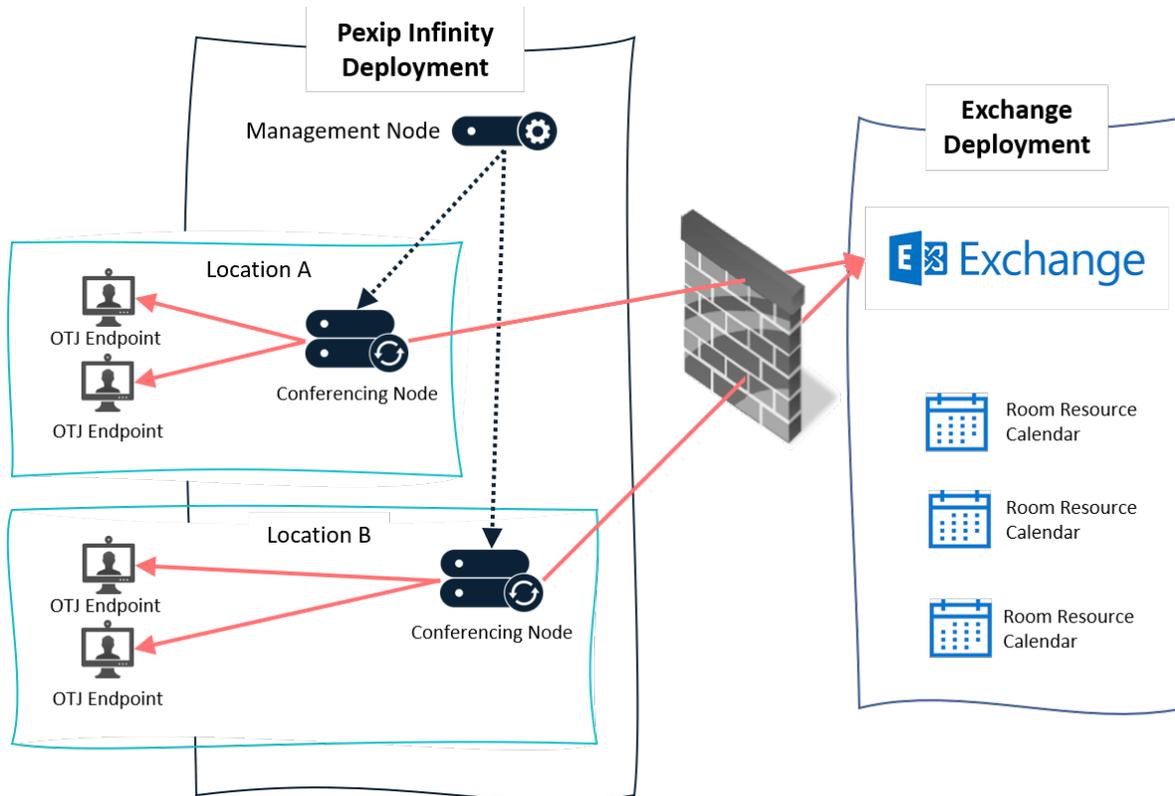
Each Conferencing Node used for One-Touch Join requires a persistent connection to either G Suite or the Microsoft Exchange server (direct access is required; web proxies are not currently supported), and must be able to sign in to it as the service account.

Each Conferencing Node must be able to access the One-Touch Join endpoints within its location, via the endpoints' APIs.

You must therefore ensure the following access is enabled **from each Conferencing Node** to:

- G Suite or Exchange server: 443/TCP
- each OTJ endpoint in the same location: 443/TCP for HTTPS (otherwise 80/TCP for HTTP)

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



Permitting the service account to access calendars

Exchange integrations

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

The Pexip Infinity One-Touch Join feature requires that the service account is configured to use impersonation to access the room resources' calendars.

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.

G Suite integrations

When integrating with G Suite, the One-Touch Join service account must be able to read the calendars of each room resource. This is achieved by [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.

Configuring Exchange on-premises for One-Touch Join

This topic describes how to configure Microsoft Exchange in order to implement Pexip Infinity's One-Touch Join feature in a Microsoft Exchange on-premises environment.

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 that 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. [Enabling basic authentication](#) for the service account.
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. Ensure that you have a license available for the service account; this is required for the service account to access Exchange.
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 not be used for any other purpose other than One-Touch Join. You can however use the same service account for multiple One-Touch Join Exchange integrations.

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

EAC

1. Log in to your Exchange Admin Center as an administrator and go to recipients > mailboxes.
2. Add a new mailbox for the service account by selecting the + icon and then User mailbox.
3. Complete the fields as appropriate.
4. Uncheck the Require password change on next logon box.

new user mailbox

Alias:

pexip

 Existing user

 New user

First name:

Pexip

Initials:

Last name:

Exchange Service

*Display name:

Pexip Exchange Service

*Name:

Pexip Exchange Service

Organizational unit:

*User logon name:

 @

*New password:

*Confirm password:

 Require password change on next logon

[More options...](#)

5. Select Save.

PowerShell

- 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.

```
$password = Read-Host "Enter password"
-AsSecureString
```

```
New-Mailbox -Name "<Account Name>"
-UserPrincipalName "<UPN>" -Password
$password -Alias "<Account Alias>"
-FirstName "<Account First Name>"
-LastName "<Account Last Name>" -
DisplayName "<Account Name>"
```

```
Set-ADUser -Identity "<UPN>" -
PasswordNeverExpires $true
```

For example:

```
New-Mailbox -Name "Service Account" -
UserPrincipalName pexip@rd.pexip.com -
Password $password -Alias pexip -
FirstName Service -LastName Account -
DisplayName "Service Account"
```

```
Set-ADUser -Identity
pexip@rd.pexip.com -
PasswordNeverExpires $true
```

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

1. Log in to your **Exchange Admin Center** as an administrator and go to **recipients > groups**.
2. Select the + icon and select **add a new Distribution Group**.
3. Add the rooms you want to impersonate to the group.
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 be 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@pexip.no"
$otj_service_account = "otjsvc@pexip.no"
$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 Rooms20190430164340,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...	otjsvc	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   : otjsvc
```

Enabling basic authentication

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

If you are using on-prem Exchange you need to ensure basic authentication is enabled for Exchange Web Services (EWS). When basic authentication is enabled, Pexip Infinity stores the credentials in encrypted form and all authentication is carried out over a secure TLS channel.

You can enable basic authentication using **either** Windows Service Manager or PowerShell, as follows:

Windows Service Manager	PowerShell
<ol style="list-style-type: none"> Go to the Windows server on which Exchange is installed and open the Service Manager. Select the server on which Exchange is installed, and right-click to select Computer Management. From the panel on the left, select Services and Applications > Internet Information Services (IIS) Manager. Expand the options and select Sites > Default Web Site > EWS. Select the Authentication button in the main pane. Find Basic Authentication in the list and ensure it is Enabled. (If not, right-click and select Enable.) Select Save. 	<p>This command enables basic authentication 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

In order to take full advantage of the functionality offered by Pexip Infinity One-Touch Join, we recommend that, for One-Touch Join room resources, you change three calendar processing options from the default.

- Firstly, 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 One-Touch Join 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 to One-Touch Join and can be displayed on the meeting room endpoints.

- Secondly, by default the meeting invite body is deleted. If you wish 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).
- Thirdly, by default the private flag is cleared. If you wish 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**.

PowerShell command

To modify the calendar processing on a room from the default settings to those we recommend for One-Touch Join, use the following PowerShell command (replacing **resource_email** with the address of the room resource whose processing you wish to change):

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

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 does mean 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 wish 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 will be able to view the room's availability, but nothing else
- **LimitedDetails**: users will be able to 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

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**. This also allows users external to your organization to invite the resource directly; you should therefore consult your Exchange administrator to determine whether this is appropriate in your environment.

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:

```
$deleted_subjects = @()
$organiser_added = @()
$deleted_bodies = @()
$private_flag_reset = @()
$not_auto_accept = @()
$process_external = @()
$otj_group_id = "otjrooms@pexip.no"

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 organiser to the meeting subject" -
ForegroundColor Red
        $organiser_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 $($organiser_added.count) rooms adding the organiser to the meeting subject"
    if ($organiser_added) {
        Write-Host $organiser_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	The username of the service account to be used by the One-Touch Join Exchange integration . This must be in the format <code>name@example.com</code> .
Enable OAuth	Leave this option disabled to continue using Basic Auth. (OAuth 2.0 is supported for Exchange in Office 365 only.)
Service account password	(Available if OAuth has not been enabled) The password of the service account to be used by the One-Touch Join Exchange integration.

Advanced options

Find Items Request Quota	The number of Find Item requests that can be made by OTJ to your Exchange Server in a 24-hour period. The default of 1,000,000 should be sufficient for most deployments — for more information, see Frequency and limitations on calendar requests . 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.
--------------------------	--

OTJ Exchange Autodiscover URLs

 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.
Autodiscover URL	The URL used to connect to the Autodiscover service on the Exchange deployment.

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 for One-Touch Join

This topic describes how to configure Microsoft Exchange in order to implement Pexip Infinity's One-Touch Join feature in a Microsoft Office 365 environment.

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. [Enabling OAuth authentication](#) for the service account.
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. Ensure that you have a license available for the service account; this is required for the service account to access Exchange.
4. Ensure you have access to your Office 365 web interface, and access to the Microsoft Azure Active Directory Module for Windows PowerShell. (If you are connecting from your Windows PC for the first time, you may need to run the `Install-Module MSOnline` PowerShell command; for more information, see <https://docs.microsoft.com/en-us/office365/enterprise/powershell/connect-to-office-365-powershell>.)

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 not be used for any other purpose other than One-Touch Join. You can however use the same service account for multiple One-Touch Join Exchange integrations.

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

O365

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

- Select **Add** to create the user.

PowerShell

Establishing a remote connection

To use PowerShell for Office 365 you first need to connect remotely. Full instructions are given at <https://technet.microsoft.com/en-gb/library/dn568015.aspx> but the commands are:

```
Set-ExecutionPolicy RemoteSigned
$UserCredential = Get-Credential
$Session = New-PSSession -ConfigurationName Microsoft.Exchange -
ConnectionUri https://outlook.office365.com/powershell-liveid/ -Credential
$UserCredential -Authentication Basic -AllowRedirection
Import-PSSession $Session
Import-Module MsOnline
Connect-MsolService -Credential $UserCredential
```

Creating the service account

- 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. The final command terminates the remote session.

```
$password = Read-Host "Enter password" -AsSecureString
New-Mailbox -Name "<Account Name>" -MicrosoftOnlineServicesID "<UPN>"
-Password $password -Alias "<Account Alias>" -FirstName "<Account
First Name>" -LastName "<Account Last Name>" -DisplayName
"<Account Name>"
Set-MsolUser -UserPrincipalName "<UPN>" -PasswordNeverExpires $true
Remove-PSSession $Session
```

For example:

```
New-Mailbox -Name "Service Account" -MicrosoftOnlineServicesID
pexip@pexip.no -Password $password -Alias pexip -FirstName Service -
LastName Account -DisplayName "Service Account"
Set-MsolUser -UserPrincipalName pexip@pexip.no -PasswordNeverExpires $true
```

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

- Go to admin.microsoft.com and log in as the administrator.
- From the menu on the left hand side, select **Groups > Add a group**.
- For the **Group Type**, select **Distribution List**. Enter a name, email address and description and select **Add**.
- 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.

5. Open up a remote PowerShell connection to Office 365 and import an Exchange session. For example see <https://docs.microsoft.com/en-us/office365/enterprise/powershell/connect-to-all-office-365-services-in-a-single-windows-powershell-window>

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@pexip.no"
$otj_service_account = "otjsvc@pexip.no"
$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 Rooms20190430164340,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...	otjsvc	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 : otjsvc
```

Enabling OAuth authentication

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

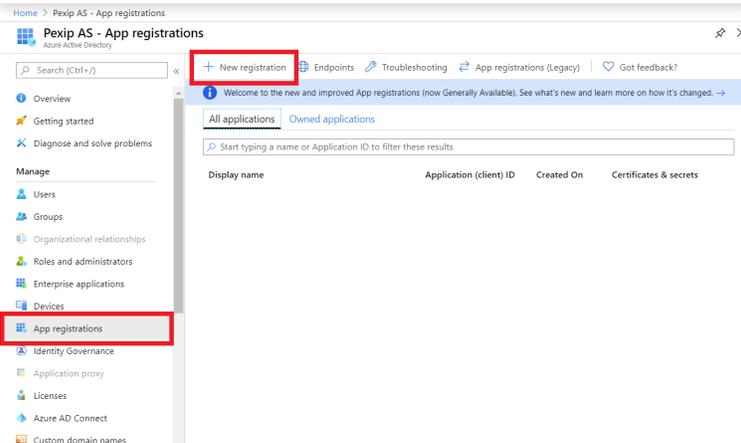
As of October 2020, Microsoft will stop supporting and fully decommission Basic Authentication for EWS to access Exchange Online (for more information, see [Microsoft's announcement](#)). We therefore strongly recommend that for Office 365, all new

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 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 should be 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/otjexchangedeployment/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 URI to which you will be returned after you have successfully signed in to the service account. It 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/otjexchangedeployment/>

[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, scroll down to the bottom and select **Exchange**:

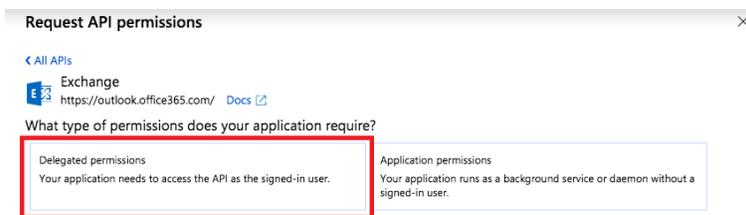
Request API permissions

Azure Key Vault Manage your key vaults as well as the keys, secrets, and certificates within your Key Vaults	Azure Rights Management Services Allow validated users to read and write protected content	Customer Insights Create profile and interaction models for your products
Data Export Service for Microsoft Dynamics 365 Export data from Microsoft Dynamics CRM organization to an external destination	Dynamics CRM Access the capabilities of CRM business software and ERP systems	Dynamics ERP Programmatic access to Dynamics ERP data
OneNote Create and manage notes, lists, pictures, files, and more in OneNote notebooks	PowerApps Runtime Service Powerful data storage, modeling, security and integration capabilities	Speech Create powerful speech-enabled features using speech to text and text to speech conversion
Visual Studio Team Services Integrate with Visual Studio Team Services (VSTS) and Team Foundation Server (TFS) accounts	Yammer Access resources in the Yammer web interface (e.g. messages, users, groups etc.)	

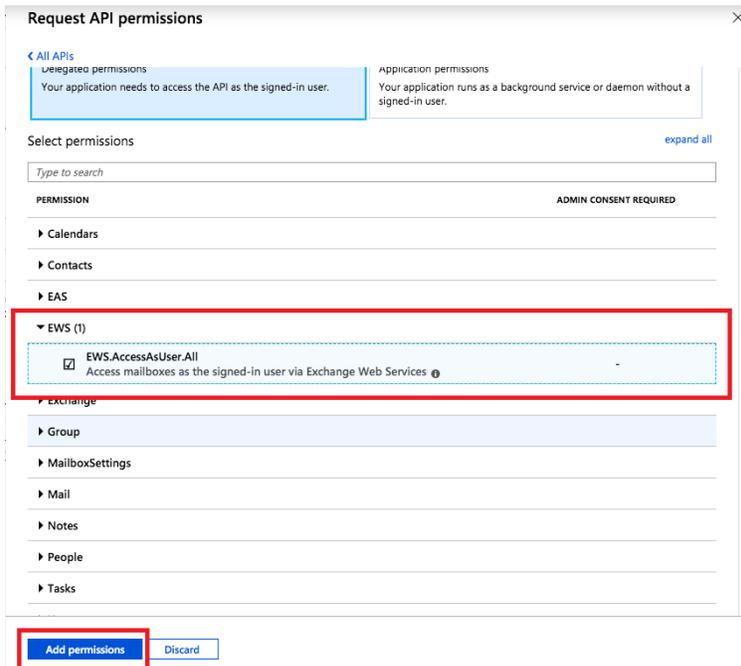
Supported legacy APIs

Azure Active Directory Graph Programmatic access to directory data and objects	Exchange A powerful, easy-to-use way to access and manipulate Exchange data
--	---

9. Select **Delegated permissions**:



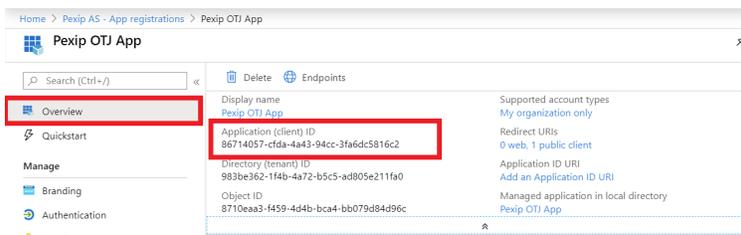
10. From the **Select permissions** list, expand **EWS** and select **Access mailboxes as the signed-in user via Exchange Web Services**, and then select **Add permissions**:



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:



- 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.
 - 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:

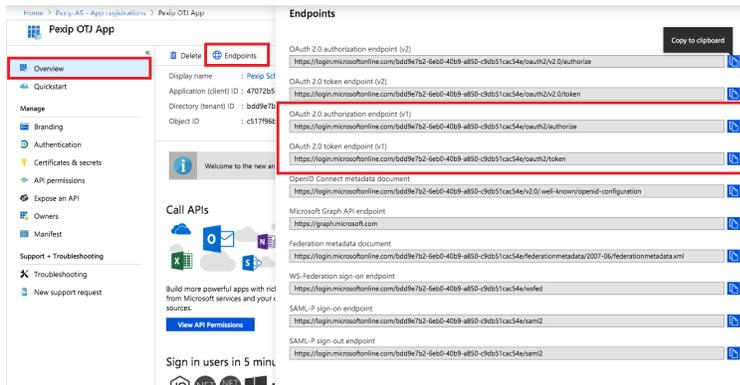
1. In the Azure Portal, select **Overview > Endpoints**.

2. Copy the URL of the **OAuth 2.0 token endpoint (v1)**.

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

3. Copy the URL of the **OAuth 2.0 authorization endpoint (v1)**.

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



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

In order to take full advantage of the functionality offered by Pexip Infinity One-Touch Join, we recommend that, for One-Touch Join room resources, you change three calendar processing options from the default.

- Firstly, 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 One-Touch Join 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 to One-Touch Join and can be displayed on the meeting room endpoints.

- Secondly, by default the meeting invite body is deleted. If you wish 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).
- Thirdly, by default the private flag is cleared. If you wish 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**.

PowerShell command

To modify the calendar processing on a room from the default settings to those we recommend for One-Touch Join, use the following PowerShell command (replacing **resource_email** with the address of the room resource whose processing you wish to change):

```
Set-CalendarProcessing -Identity <resource_email> `
-DeleteSubject $False `
-AddOrganizerToSubject $False `
```

```
-DeleteComments $False`  
-RemovePrivateProperty $False
```

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 does mean 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 wish 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 will be able to view the room's availability, but nothing else
- **LimitedDetails**: users will be able to 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

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`. This also allows users external to your organization to invite the resource directly; you should therefore consult your Exchange administrator to determine whether this is appropriate in your environment.

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:

```
$deleted_subjects = @()  
$organiser_added = @()  
$deleted_bodies = @()  
$private_flag_reset = @()  
$not_auto_accept = @()  
$process_external = @()  
$otj_group_id = "otjrooms@pexip.no"  
  
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 organiser to the meeting subject" -  
ForegroundColor Red  
        $organiser_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 $($organiser_added.count) rooms adding the organiser to the meeting subject"
    if ($organiser_added) {
        Write-Host $organiser_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 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 service account to be used by the One-Touch Join Exchange integration . This must be in the format <code>name@example.com</code> .
Enable OAuth	Enable this option to authenticate the service account using OAuth 2.0. (This option is only supported for Exchange in Office 365.)  As of October 2020, 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.
OAuth client ID	(Available if OAuth has been enabled) The Application ID which was generated by Azure when creating an App Registration in Azure Active Directory.
OAuth redirect URI	(Available if OAuth has been enabled) The redirect URI you entered when creating an App Registration in Azure Active Directory. This should be in the format <code>https://<Management Node Address>/admin/platform/mjxexchangedeployment/oauth_redirect/</code>
OAuth authorization endpoint	(Available if OAuth has been enabled) The URI of the OAuth authorization endpoint. This should be copied from the 'Endpoints' section in Azure Active Directory App Registrations.
OAuth token endpoint	(Available if OAuth has been enabled) The URI of the OAuth token endpoint. This should be copied from the 'Endpoints' section in Azure Active Directory App Registrations.

Advanced options

Find Items Request Quota	The number of Find Item requests that can be made by OTJ to your Exchange Server in a 24-hour period. The default of 1,000,000 should be sufficient for most deployments — for more information, see Frequency and limitations on calendar requests . 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.
--------------------------	--

OTJ Exchange Autodiscover URLs

 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.
Autodiscover URL	The URL used to connect to the Autodiscover service on the Exchange deployment.

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:

- 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**:

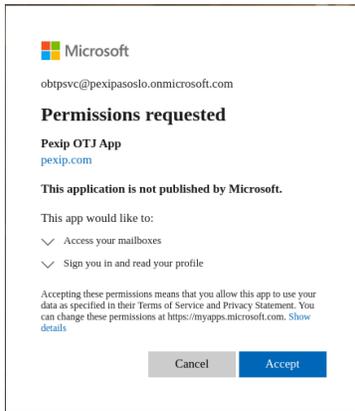
The screenshot shows the 'Change OTJ Exchange Integration' page. It has two main sections for OAuth configuration. The first section is for the 'OAuth authorization endpoint'. It shows a 'Currently' field with the URL 'https://infinity.example.com/admin/platform/otj/oauth_redirect/' and a 'Change' field with the same URL. Below this is a note: 'The URI of the OAuth authorization endpoint. This should be copied from the 'Endpoints' section in Azure /'. The second section is for the 'OAuth token endpoint'. It also shows 'Currently' and 'Change' fields with the same URL, and a note: 'The URI of the OAuth token endpoint. This should be copied from the 'Endpoints' section in Azure Active D'. At the bottom of the page, there are three buttons: 'Save', 'Save and add another', and 'Sign in to service account'. The 'Sign in to service account' button is highlighted with a red box.

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

The screenshot shows the 'Sign in to service account' page. At the top, it says ']pexip[Infinity Conferencing Platform'. Below that is a navigation menu with items: Status, History & Logs, System, Platform, Call Control, Services, Users & Devices, One-Touch Join, and Utilities. The main heading is 'Sign in to service account'. Below this is a paragraph: '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.' There is a 'Warning' section: '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.' Below the warning is a 'Sign-in link' section. A red box highlights a text input field containing the sign-in link: '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 will be asked to permit the Application registration to access the service account:



5. Select **Accept**.

You should be 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 G Suite for One-Touch Join

This topic describes how to configure G Suite in order to implement Pexip Infinity's One-Touch Join feature in a G Suite 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. For larger deployments, [Requesting an increase to API limits](#).
 5. [Adding a One-Touch Join G Suite integration](#) on Pexip Infinity.
- i** If you have already set up a One-Touch Join G Suite integration and simply wish to add an existing room to it, you need only [configure the room resource](#) in G Suite and then [add the endpoint to the G Suite integration](#) in Pexip Infinity.

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 G Suite 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 G Suite integrations.

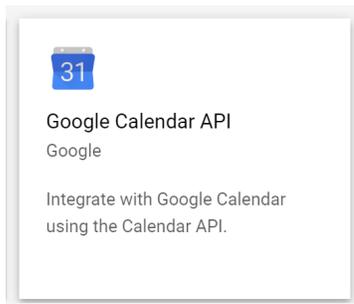
1. Creating a new project:
 - a. Go to <https://console.developers.google.com> (logged in as a G Suite administrator).
 - b. From the top left of the page, select the down arrow:

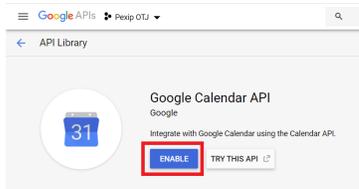


- c. Select **New Project**.
 - d. Enter a **Project name** (e.g. One-Touch Join) and select **Create**.
2. Enabling the Calendar API for the project:
 - a. Go to <https://console.developers.google.com>
 - b. From the top left of the page, select the down arrow, select your newly-created project, and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

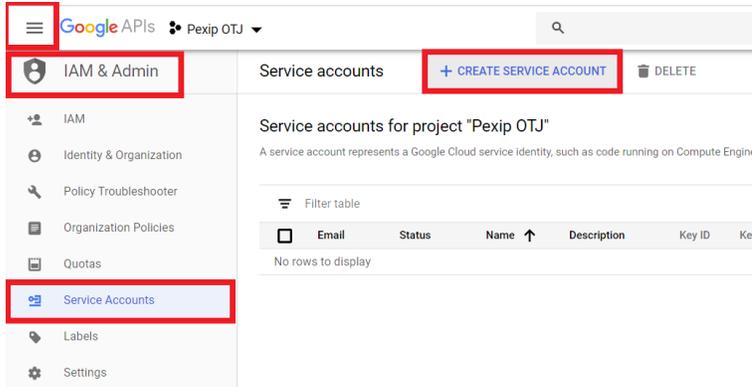
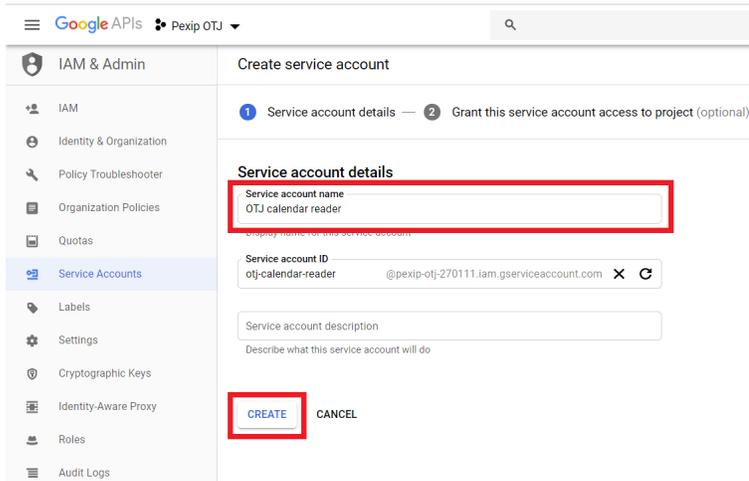


- c. From the navigation menu on the left of the screen, select **APIs & Services > Library**, then scroll down and select the **Google Calendar API** tile:

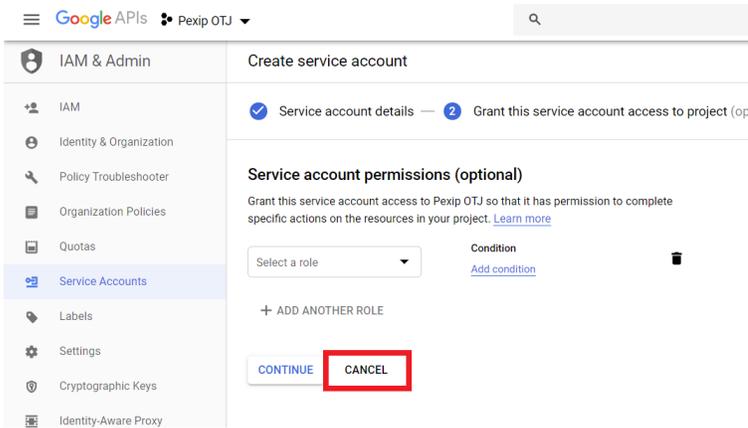


d. Select **Enable**:

3. Creating the Service Account:

a. Go to <https://console.developers.google.com>b. From the navigation menu on the left of the screen, select **IAM & Admin > Service Accounts**.c. Select **Create Service Account**:d. Enter a name (e.g. One-Touch Join Calendar Reader) and select **Create**:

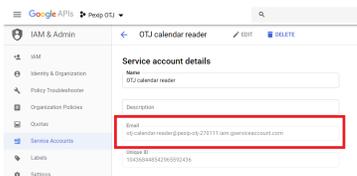
- e. On the next page, which asks about permissions, select **Cancel** (the account does not need any of these permissions):



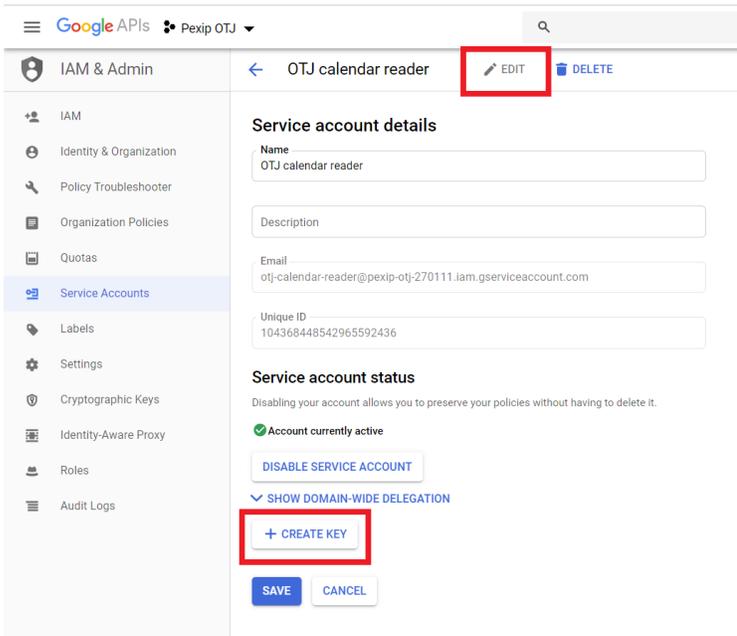
4. Generating a key file:

- 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:



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



- 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.



CANCEL CREATE

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

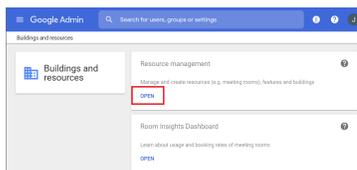
For more information on using OAuth 2.0 to authenticate the Service Account, see <https://developers.google.com/identity/protocols/OAuth2ServiceAccount>.

Creating a room resource

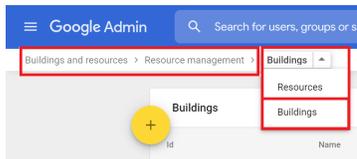
(Required only if your room resources do not already exist - otherwise you can skip this step.)

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

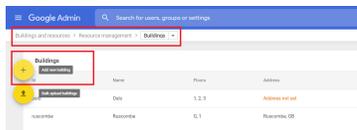
1. If a **building** for the room resource does not already exist, create one as follows:
 - a. Go to <https://admin.google.com> (logged in as a G Suite administrator).
 - b. Select the **Buildings and resources** tile, and then from the **Resource management** section select **Open**:



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

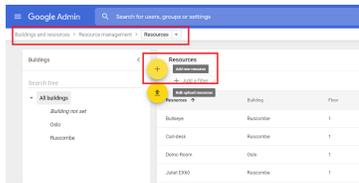


- c. Select **+** to **Add new building**:



- 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: Dalo-2 Board room (20)
 Category: Meeting space (room, phone booth,...) | Name: meeting room
 Building: Dalo | Floor: 2 | Floor section: [blank]
 Capacity: 20
 Board room
 Features: Define features first to add them.
 User visible description: [blank]
 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.

- i** For more information on setting up buildings and other resources in G Suite, 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 One-Touch Join Service Account to access the 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 Service Account

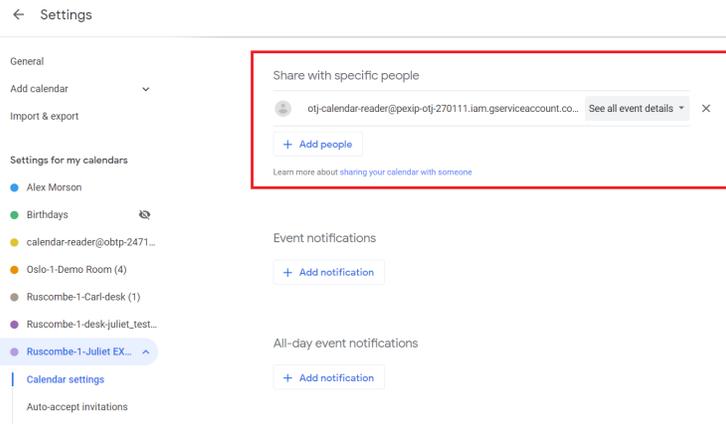
Note that the Google calendar API limits the number of calendars that can be shared within a 24 hour period to 750 (for more information, see <https://support.google.com/a/answer/2905486?hl=en>). This means that if you have more than 750 room resources that you wish to use for One-Touch Join, they will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

1. Go to <https://calendar.google.com> (logged in as a G Suite 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 (this was automatically generated when you [created the Service Account](#), and can be found at <https://console.developers.google.com/iam-admin/serviceaccounts>). Ensure the **Permissions** are set to **See all event details**.
 - d. Select **Send**:



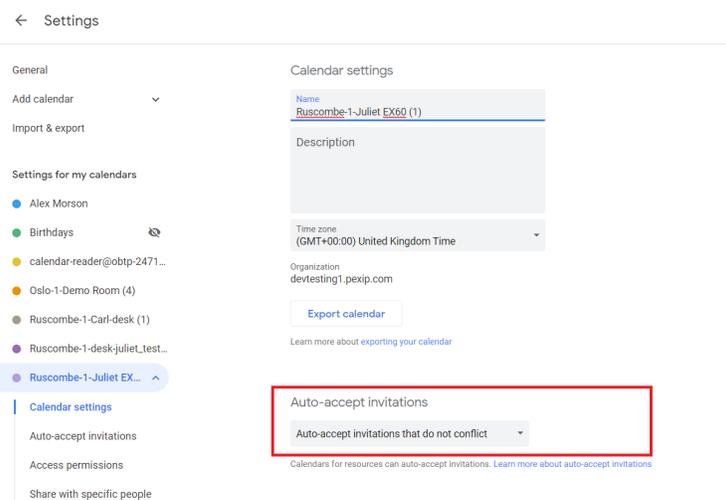
- i** For more information on sharing room and resource calendars in G Suite, see <https://support.google.com/a/answer/1034381>.

Auto-accepting invitations

By default, when creating room resources in G Suite, 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 G Suite 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:

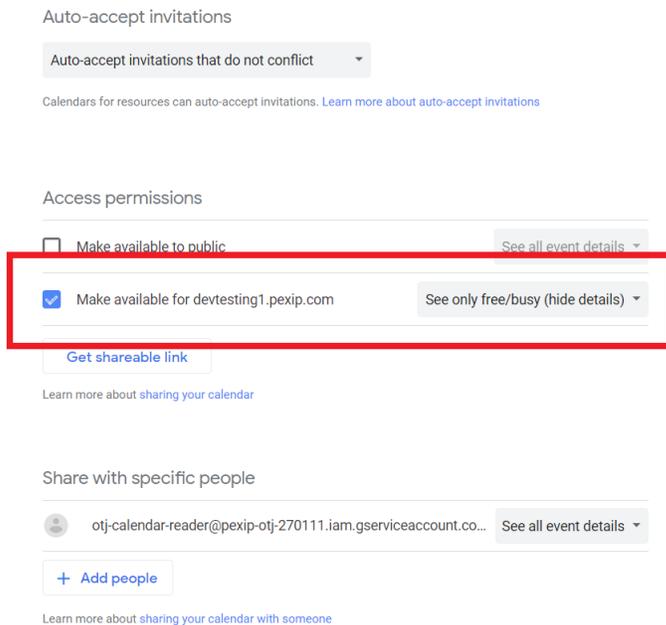


Allowing users to book resources

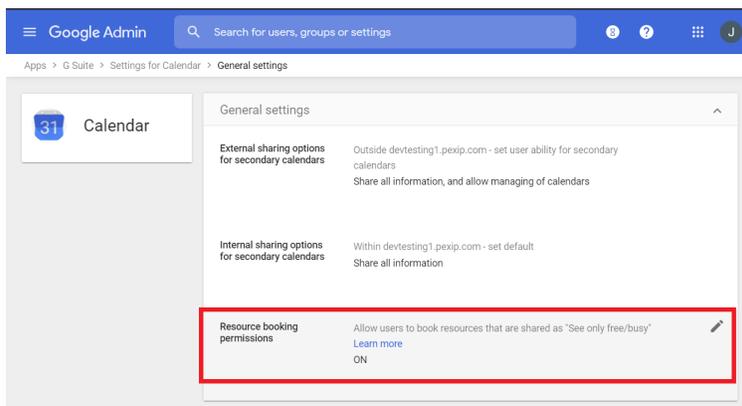
We recommend that you configure your G Suite calendar settings to allow end users to book a room resource without seeing details of the room's other bookings. To do this, you configure the room resource's calendar so that all users in your domain have permission to see its free/busy status, without being 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 G Suite 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:



4. Go to admin.google.com (logged in as a G Suite administrator).
5. From the left-hand menu, select **Apps > G Suite > 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**:



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.

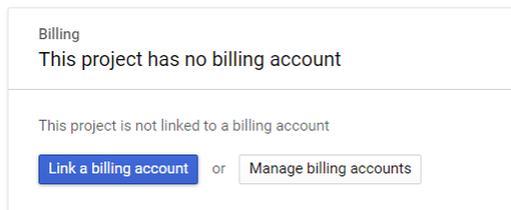
The maximum frequency with which an endpoint will be updated with meeting information is every 30 seconds. For deployments with more than around 170 endpoints, this frequency will decrease in line with the number of endpoints (up to around 20 minutes for deployments with around 6,000 endpoints). This 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.

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

To request an increase to the API limits:

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



- Select the account to link to:

Set the billing account for project "Quickstart"

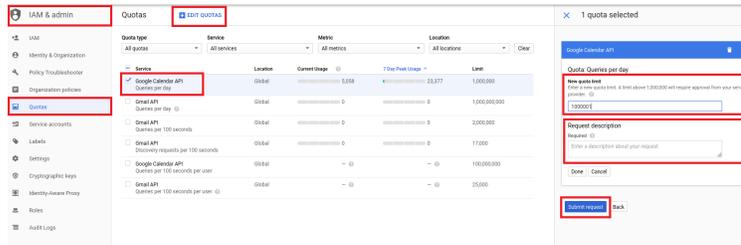
Billing account

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

My Billing Account

CANCEL SET ACCOUNT

- Request an increase to your quota:
 - Select the burger menu from the top left of the page and select **IAM & admin > Quotas**.
 - 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:



c. Select **Submit request**.

Quota increase requests typically take two business days to process.

Adding a One-Touch Join G Suite integration on Pexip Infinity

In this step you configure Pexip Infinity with details of the G Suite 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 G Suite Integrations**.

Option	Description
Name	The name of this One-Touch Join G Suite integration.
Description	An optional description of this One-Touch Join G Suite integration.
Service account email	The email address of the service account that One-Touch Join will use to log in to G Suite.
Private key	The private key used by One-Touch Join to authenticate the service account when logging in to G Suite. For instructions on how to obtain this, see Generating a key file .
Advanced options	
Maximum G Suite API requests	The maximum number of API requests that can be made by One-Touch Join to your G Suite Domain in a 24-hour period. 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. For more information, see Frequency and limitations on calendar requests .

Next steps

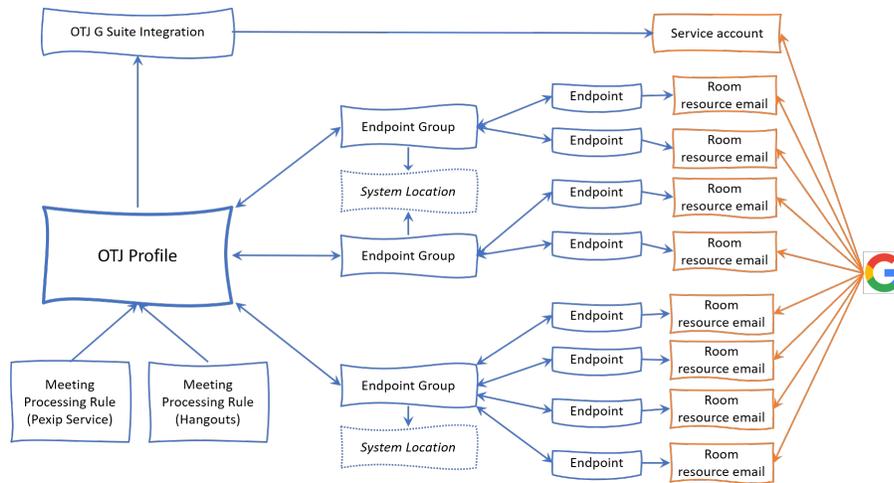
You must now configure the remainder of the One-Touch Join components on Pexip Infinity, as described in [Configuring Pexip Infinity for One-Touch Join](#).

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 G Suite, with support for Google Hangouts 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 G Suite for One-Touch Join](#), including [Adding a One-Touch Join G Suite 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 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 G Suite 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 can apply to **either** an Exchange integration or a G Suite integration, but not both.

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.
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 Default API password	<p>The user name and password used by One-Touch Join to access the 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 a One-Touch Join endpoint configuration 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>
Verify endpoint certificates by default	<p>Whether or not to verify the TLS certificate of an 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 an 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>
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>
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>

Option	Description
Replace subject for private meetings	(Applies if Enable private meetings has been selected) <i>Enabled:</i> For private meetings, the endpoint will display the organizer's name in place of the meeting subject. <i>Disabled:</i> For private meetings, the endpoint will display the meeting subject in the usual way.
Replace empty subject	<i>Enabled:</i> For meetings that do not have a subject, the endpoint will display the organizer's name in place of the subject. <i>Disabled:</i> For meetings that do not have a subject, the endpoint will display a blank field in place of the subject.
Exchange integration	(Applies if this OTJ profile is for an Exchange or Office 365 integration) The Exchange integration used by this One-Touch Join profile. You should already have created this as part of either Configuring Exchange on-premises for One-Touch Join or Configuring Office 365 for One-Touch Join , but you can configure it now by selecting the green plus symbol + to the right of the field.
G Suite integration	(Applies if this OTJ profile is for a G Suite integration) The G Suite integration used by this One-Touch Join profile. You should have already created this as part of Configuring G Suite for One-Touch Join , but you can configure it now by selecting the green plus symbol + to the right of the field.
Endpoint groups	The endpoint groups used by this One-Touch Join profile.

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.

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.
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](#).

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
Name	The name of this One-Touch Join endpoint.
Description	An optional description of this One-Touch Join endpoint.
Endpoint type	The type of "click to join" feature supported by this endpoint. For this release, the only available option is <i>Cisco OBTP</i> .
Endpoint API 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.
Endpoint API username	The user name and password used by One-Touch Join to access the 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> . Either both these fields must be configured, or both these fields must be left blank. If both these fields are left blank, the One-Touch Join profile's Default API username and password will be used.
Endpoint API password	
Verify endpoint API TLS certificate	Whether to enable TLS verification when accessing this endpoint's API. Only applicable if using HTTPS to access this endpoint's API. <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. <i>On:</i> Enable TLS verification. <i>Off:</i> Do not use TLS verification. For more information, see Managing trusted CA certificates .
Use HTTPS	Whether to use HTTPS to access this endpoint's API. <i>Use OTJ profile default:</i> Use the Use HTTPS for endpoint API setting configured for the One-Touch Join profile that this endpoint is associated with. <i>On:</i> Use HTTPS to access this endpoint's API. <i>Off:</i> Use HTTP to access this endpoint's API.
Room resource email	The email address of the room resource associated with this endpoint. This must match the email address that has been configured in Exchange or G Suite.
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 for each import file type.

- 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:

1. Create the CSV file, using the following format:

```
name,description,endpoint_type,api_address,api_port,api_username,api_password,verify_cert,use_https,room_resource_email,mjx_endpoint_group_name
```

where

Field name	Content
name	<p> This field cannot be blank.</p> <p>The name of this One-Touch Join endpoint.</p> <p>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).</p>
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>Valid values are:</p> <ul style="list-style-type: none"> ◦ CISCO
api_address	<p> This field cannot be blank.</p> <p>The IP address or FQDN of the endpoint's API.</p>
api_port	<p>The port of the 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 endpoint's API.
api_password	The password used by OTJ to access the endpoint's API.
verify_cert	<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>Valid values are:</p> <p>GLOBAL: Use the Verify endpoint certificates by default setting configured for the One-Touch Join profile that this endpoint is associated with.</p> <p>On: Enable TLS verification.</p> <p>Off: Do not use TLS verification.</p>
use_https	<p>Whether to use HTTPS to access this endpoint's API.</p> <p>GLOBAL: Use the Use HTTPS for endpoint API setting configured for the One-Touch Join profile that this endpoint is associated with.</p> <p>On: Use HTTPS to access this endpoint's API.</p> <p>Off: Use HTTP to access this endpoint's API.</p>

Field name	Content
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 G Suite.</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 meeting type. However, you also have the option to override the default processing with your own transform rule to change how the alias is constructed. You can also write your own [custom rules](#) if you wish to enable One-Touch Join for other meeting types or conferencing providers not already 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** 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 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 pushes the alias, along with other meeting information such as the start time, end time, subject, and organizer's name, to the endpoint.

If no alias has been obtained, One-Touch Join may still push 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 G Suite 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> 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> or <i>Custom</i> rules are applied, particularly if you are using more than one of either.</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>Custom</i> if you wish to write 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>
Default processing enabled	<p>Check this box to use the default transform rule for this meeting type. For a list of the default transform rules for each meeting type, see Supported meeting types.</p> <p>Clear this box to write your own Transform rule for this meeting type.</p>
Transform rule	<p>(Available and required when Default processing is disabled and any Meeting type option other than <i>Custom</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 rule, which you can then edit.</p> <p>For a list of the valid variables for each meeting type, see Supported meeting types.</p>
Domain	<p>(Available and required when a Meeting type of either <i>Domain</i> or <i>Microsoft Teams Meeting Properties</i> has been selected.)</p> <p>The domain from which the meeting invitation was sent.</p> <p>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.</p> <p>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.</p>
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	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.

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, allow you to test that the rule works as expected for the selected deployment and meeting types, and also allows you to edit the template until you get the desired results.

To test the rule:

1. Review and complete the following fields:

Option	Description
Integration type	This read-only field shows whether the rule will be applied to a G Suite 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.
Transform rule	This shows the transform rule being used. <ul style="list-style-type: none">○ If you selected Default processing enabled, this will be the default rule for this meeting type.○ If you did not select Default processing enabled, this will be the Transform rule you entered. You can edit the rule here and re-test it until you get the desired results.
Calendar event properties	(Available for some meeting types) The expected format of the calendar events for the selected Meeting type . In some cases this will be populated automatically.
Calendar event body	(Available for some meeting types) This field allows you to paste in the raw text from an email body. This will be used if the meeting alias cannot be obtained from the Calendar event properties , for example if an invitation has been forwarded. In some cases this will be populated automatically.

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

The **Result** field shows the meeting alias that would be extracted based on the rule's current settings.

- If this is blank, there was an error running the script.
 - If the result is not as expected, edit the fields above as appropriate.
3. When the template is working as expected, to save the changes you have made, select **Save changes and return**.

One-Touch Join meeting types and transform rules

This topic details the meeting types, transform rules and variables that are supported when [Adding One-Touch Join meeting processing rules](#). For each of the [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. If necessary, you can change how the alias is constructed, or write your own [custom rules](#) if you wish to enable One-Touch Join for other meeting types or conferencing providers not already supported.

Note that 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), it will always search the body 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 **Meeting types**. For each type, the **Default transform rule** 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 rule](#) for this meeting type.

Meeting type	Usage and notes	Default transform rule	Valid variables
Pexip Infinity	<p>For meetings scheduled using Pexip's VMR Scheduling for Exchange feature.</p> <p>If your VMR Scheduling for Exchange deployment does not use the default scheduling template, you should select a Meeting type of Domain or Custom instead.</p>	{{meeting_id}}@{{domain}}	<ul style="list-style-type: none"> • meeting_id • domain
Pexip Service	For meetings held in Pexip Service VMRs.	{{meeting_id}}@pexip.me	<ul style="list-style-type: none"> • meeting_id • domain
Microsoft Teams Meeting Properties	<p>(Not currently supported for G Suite 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...</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>	{{meeting_id}}@{{domain}}	<ul style="list-style-type: none"> • meeting_id • domain
Microsoft Teams Meeting Body for Poly	If you expect users in your deployment to receive invitations to Microsoft Teams meetings where the meeting organizer is using a Poly / Teams integration.	{{tenant_id}}.{{meeting_id}}@t.plcm.vc	<ul style="list-style-type: none"> • meeting_id • domain • tenant_id

Meeting type	Usage and notes	Default transform rule	Valid variables
Microsoft Teams Meeting Body for BlueJeans	If you expect users in your deployment to receive invitations to Microsoft Teams meetings where the meeting organizer is using a BlueJeans / Teams integration.	{{tenant_id}}.{{meeting_id}}@teams.bjn.vc	<ul style="list-style-type: none"> meeting_id domain tenant_id
Microsoft Teams Meeting Body for Pexip Service	If you expect users in your deployment to receive invitations to Microsoft Teams meetings where the meeting organizer is using a Pexip Service / Teams integration.	{{meeting_id}}@{{domain}}	<ul style="list-style-type: none"> meeting_id domain
Microsoft Teams Meeting Body for Pexip Infinity	If you expect users in your deployment to receive invitations to Microsoft Teams meetings where the meeting organizer is using a Pexip Infinity / Teams integration.	{{prefix}}{{meeting_id}}@{{domain}}	<ul style="list-style-type: none"> meeting_id domain prefix
Google Hangouts Meet	(Not currently supported for Exchange integrations) For meetings scheduled using Google Hangouts Meet.	{{meeting_id}}@{{domain}}	<ul style="list-style-type: none"> meeting_id domain
Skype for Business	For Skype for Business meetings. The domain used is the domain of the organizer's email address.	__sfb__{{focus_id}}.{{user}}@{{domain}}	<ul style="list-style-type: none"> focus_id domain
Skype for Business Meeting Body for Poly	For Skype for Business meetings, where the meeting organizer is using a Sfb / Poly integration. The domain is hard-coded to v.plcm.vc.	{{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	For Zoom meetings. In all cases the domain is hard coded to zoomcrc.com.	{{meeting_id}}@zoomcrc.com	<ul style="list-style-type: none"> meeting_id domain
BlueJeans	For BlueJeans meetings. In all cases the domain is hard coded to bjn.vc.	{{meeting_id}}@bjn.vc	<ul style="list-style-type: none"> meeting_id domain
GoToMeeting	For GoToMeeting meetings.	{{meeting_id}}@{{domain}}	<ul style="list-style-type: none"> meeting_id domain

Meeting type	Usage and notes	Default transform rule	Valid variables
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>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

Custom meeting type

You can also select a **Meeting type** of *Custom*, which 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.

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	
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"> year month day hour minute second
is_private	boolean	

Item	Type	
body	string	
properties	dictionary	<p>G Suite</p> <p>A G Suite <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.

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.

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.

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. However, in some situations you may wish 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, or for additional capacity for larger deployments (more than around 170 endpoints).

For small dedicated One-Touch Join-only deployments — for example, those with fewer than 50 endpoints — the resource requirements are minimal, therefore the following recommendations apply:

- 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.

Note that if you expect to broaden your deployment to implement some of the wider Pexip Infinity features in the future, you may need to review your choice of hardware.

For full details of Pexip Infinity server requirements, see [server design guidelines](#).

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 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 G Suite for One-Touch Join, including Adding a One-Touch Join G Suite 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

When Pexip Infinity's 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 by using add-in buttons, for example for Pexip scheduled meetings or Webex. Just ensure you have added the meeting room to the invitation as a room resource.
2. Each endpoint in each meeting room will display a list of upcoming meetings for that room. When a meeting is due to start, the endpoint in the meeting room will show a **Join** button.
3. When you are ready to join the meeting, just press the **Join** button. The endpoint will dial in to the meeting.

Viewing One-Touch Join meetings

To view a list of currently scheduled meetings in your deployment that use Pexip Infinity's One-Touch Join feature, 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.

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	The text that appears in the subject line of the meeting invitation. This field will show the organizer's name instead of the meeting subject if either: <ul style="list-style-type: none"> • Replace subject for private meetings has been enabled and the meeting was flagged as private, 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	The alias that the endpoint will use to dial in to the meeting. This will be blank if either: <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 *	The name of the meeting processing rule that was matched and used to process this meeting. This will be blank if the meeting information did not match any meeting processing rules, and Enable non-video meetings has been enabled.

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

Troubleshooting One-Touch Join

This section provides guidance on troubleshooting issues with Pexip Infinity's One-Touch Join feature.

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

Symptom	Possible cause	Resolution
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 sip:, sips: or h323:.	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, if 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 G Suite 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.
	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. You can confirm that this is the case by Adding a temporary custom rule for debugging purposes .	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).
There is a delay between a meeting invitation being sent and it appearing on the room endpoint.	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.
	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. For more information, see Frequency and limitations on calendar requests .	For larger G Suite 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 . You could also consider Deploying a dedicated One-Touch Join platform .

Symptom	Possible cause	Resolution
On the status page and logs, the Alias field is blank.	<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>
On the status page and logs, the Subject field is showing the organizer's name.	<p>Replace subject for private meetings has been enabled and the meeting was flagged as private, or</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>
<p>When configuring Exchange you are getting the following errors or warnings:</p> <pre>ErrorCode="InvalidUser" ErrorMessage="Invalid user"</pre>	<p>The service account being used for One-Touch Join does not exist, or does not have a license.</p>	<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 a valid license.
<p>One-Touch Join cannot contact an endpoint via its API. The following appears in the alarms and logs:</p> <pre>Non-200 status code returned when trying to upload OBTP bookings to endpoint and StatusCode="307"</pre>	<p>One-Touch Join is configured to communicate with the endpoint via HTTP and the endpoint redirects to HTTPS.</p>	<p>Configure One-Touch Join to use HTTPS to communicate with the endpoint.</p>

Adding a temporary custom rule for debugging purposes

In some situations, the information in the body of the original meeting invitation may not be available for One-Touch Join to process. This will result in a meeting that appears in the calendar of the room resource associated with the endpoint, and in some cases on the endpoint itself, but there is no Join button.

The following steps explain how to create a temporary custom meeting processing rule to determine whether the information in the original meeting invitation is being received by One-Touch Join.

- From **One-Touch Join > OTJ Meeting Processing Rules**, select **Add OTJ Meeting Processing Rule**.
- Configure the rule as follows:

Name	<p>Enter a name for this rule.</p> <p>In our example we have used OTJ Troubleshooting.</p>
Description	<p>Although a description is optional, we recommend that you enter one for this rule.</p> <p>In our example we have used Temporary troubleshooting of MS Exchange misconfiguration.</p>

OTJ Profile	Select the OTJ Profile associated with the Exchange Integration that you wish to troubleshoot. You should also double-check the selected OTJ Profile to ensure that the Endpoint Group to which the endpoint in question belongs has been added to the profile's Chosen Endpoint Groups section.
Priority	We recommend that you give this rule the lowest priority, i.e. 200 .
Meeting type	Select Custom .
Custom template	Enter the following Jinja2 snippet. You may wish to replace <code>OTJTemporaryDiagnostics</code> with a string of your choice, but it should be easy to search for and distinct: <pre>{{pex_debug_log("OTJTemporaryDiagnostics: Subject:", calendar_event.subject, "; BodyLength:", calendar_event.body pex_strlen, "Organizer:", calendar_event.organizer_email)}}</pre>
Enabled	Select Enabled prior to testing the rule.  Ensure that you de-select this option as soon as you have finished testing.

3. Create a test meeting in the usual way, and invite the room resource. Ensure you give the meeting a distinctive **Subject**, so you can find it easily when searching the logs.

In our example we have used **Testing OTJ troubleshooting**.

4. Go to **History & Logs > Support Log**. In the **Search** field at the top right of the page, enter the string that you used at the start of the Jinja2 snippet for the **Custom template**.

In our example this was **OTJTemporaryDiagnostics**.

5. Check the relevant log entries for those that contain `pex_debug_log` and the subject you used for your test meeting.

In our example, we found:

```
2020-02-21T11:22:01.168+00:00 mcu-workervm-1 2020-02-21 11:22:01,168 Level="INFO" Name="support.jinja2" pex_debug_log
Detail="OTJTemporaryDiagnostics: Subject:,Testing OTJ troubleshooting,;
BodyLength:,0,Organizer:,pete@pexipasoslo.onmicrosoft.com"
```

We can see from the above entry that the `BodyLength` was 0, therefore we can assume that the body of the message is being removed. We should therefore 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).

6. Remember to disable or delete the custom rule to prevent generating further log output.