Pexip Secure Scheduler for Exchange

Deployment Guide

Software Version 38

Document Version 38.a

July 2025

]pexip[

Contents

Scheduling secure meetings using Microsoft Exchange **6**
Supported Exchange deployments 6
Supported clients 6
Exchange on-premises 6
Microsoft 365 7
Support for delegate access to calendars 7
Network architecture and firewalls 7
Using a load balancer in your Exchange deployment 8
Enabling Exchange Scheduler with personal VMRs **9**
Overview 9
Prerequisites 9
Process flow - personal VMRs 9
User activates add-in 10
Initial user sign-in 11
Conferencing Node obtains VMRs 11
Add-in adds joining details to the meeting invitation 11
Next step 11
Configuring AD FS SSO for personal VMRs **12**
How it works 12
Prerequisites 12
AD FS version 12
Internet accessibility and security 12
Certificates 13
Setting up an OAuth 2.0 Client on Windows Server 13
Windows Server 2016 and later 13
AD FS 3.0 on Windows Server 2012 R2 15
Next steps 17
Configuring Azure SSO for personal VMRs **18**
How it works 18
Creating and configuring a new App Registration in Azure 18
Taking note of configuration 22
Next steps 23
Enabling Exchange Scheduler in single-use VMRs **24**
Overview 24
Process flow - single-use VMRs 25
Add-in requests aliases 27
Management Node generates aliases and join instructions 27
Management Node creates VMRs 27

PXPS:- and TOK:- security tags ... 28
Using application impersonation ... 28
Recurring meetings ... 28
PINs and authentication ... 29
Next step ... 29

Configuring Exchange on-premises for scheduling ... 30
Prerequisites ... 30
Creating a service account ... 30
Configuring Application Impersonation on the service account ... 31
Impersonating all users and resources ... 31
Impersonating resources only ... 32
Creating an equipment resource ... 33
Configuring the equipment resource ... 33
Calendar processing ... 33
Hiding resource availability in scheduling assistant ... 34
More information ... 34
Enabling authentication ... 34
Basic authentication ... 35
NTLMv2 authentication ... 35
Kerberos authentication ... 35
Viewing the equipment resource's mailbox ... 36
Next steps ... 36

Configuring Office 365 for scheduling using app permissions ... 37
Prerequisites ... 37
Creating an equipment resource ... 37
Configuring the equipment resource ... 38
Calendar processing ... 38
Hiding resource availability in scheduling assistant ... 39
More information ... 39
Creating an app registration ... 39
Creating a new app registration ... 40
Creating a client secret ... 44
Creating a management scope ... 45
Copy the app's Enterprise Object ID ... 45
Creating a distribution list ... 46
Creating and applying the management scope ... 47
Taking note of configuration ... 48
Viewing the equipment resource's mailbox ... 49
Next steps ... 49

Configuring Office 365 for scheduling using a service account ... 50

Configuring Pexip Infinity for Exchange Scheduler ... 51

Prerequisites ... 51
Adding a Secure Scheduler for Exchange Integration to Pexip Infinity ... 51
Signing in to the service account ... 66
Saving and checking configuration ... 67
Formatting the email text ... 67
Working with jinja2 templates ... 67
Variables ... 68
Examples ... 69
Deleting and replacing Secure Scheduler for Exchange Integrations ... 69
Using multiple Secure Scheduler for Exchange Integrations ... 70
Different groups of users within the same Microsoft Exchange deployment ... 70
Different Microsoft Exchange deployments ... 70
Next step ... 70

Add-in authentication for O365 ... 71
NAA (Nested App Authentication) token authentication ... 71
Creating a Web API registration ... 72
Enabling Version 2.0 access tokens ... 76
Creating the NAA App ... 77
Pre-authorizing the NAA App to access the NAA Web API ... 82
Configuring Pexip Infinity ... 84
SSO (Single Sign-On) token authentication ... 85
Creating the app registration ... 85
Assigning users and groups to the App registration ... 90
Configuring Pexip Infinity ... 92
Uploading the add-in manifest ... 92

Making the Exchange add-in available to users ... 93
Prerequisites ... 93
Downloading the add-in XML file ... 93
Testing the integration ... 96
Troubleshooting ... 96

Restricting the Exchange add-in to specific users ... 97
Prerequisites ... 97
Active Directory (AD) group ... 97
Exchange distribution group ... 98
Entra ID group ... 98
Verifying the add-in is available as expected ... 98
Changing the users for an existing add-in ... 99

Managing Exchange Scheduler conferences ... 100
Scheduled conference VMRs ... 100
Viewing all single-use VMRs used for scheduled conferences ... 100
Editing the VMR for a scheduled conference ... 100

Viewing all upcoming scheduled meetings 101
Troubleshooting 101

Maintenance and recovery procedures for Secure Scheduler for Exchange 102
Running the scripts 102
Recovering meetings 102
Example 102
Deleting old mailbox items 103
Example 103

Locating the add-in 104
Outlook desktop client (new version) 104
Outlook desktop client 104
Office 365 (new version) 104
Office 365 104
On-prem OWA 105

Creating a new video meeting 106
Using a personal VMR 106
Using a single-use VMR 106
Joining details 107

Changing an existing meeting into a video meeting 108

Editing or canceling an existing video meeting 109

Example joining instructions 110
Including images in the joining instructions 110
Example: dark background with images 110
Example: light background with images 113

Troubleshooting Secure Scheduler for Exchange 116

Scheduling secure meetings using Microsoft Exchange

The Secure Scheduler for Exchange feature (previously known as VMR Scheduling for Exchange) allows you to create an add-in that enables Microsoft Outlook users in Office 365 or Exchange environments to quickly and easily add a Pexip VMR to their meeting invitations, enabling any meeting to be held over video.

Users can host their meeting in a single-use VMR that is created specifically for the meeting and only available for its duration, or they can host their meeting in their own personal VMR. You can either let users decide which type of VMR to use for each meeting, or make just one type of VMR available in your deployment.

You can provide the add-in button as the only option within Outlook for scheduling video meetings, or you can offer it alongside other add-ins. This means that if you already have a primary meeting solution, you can still offer a "two button" approach, with Pexip Infinity as a complementary meeting alternative — for example, when the meeting's content or security classification requires it, or for business continuity purposes. For more information about Pexip Secure Meetings, see our website.

Secure Scheduler for Exchange is an optional licensed feature within the Pexip Infinity platform. When it is enabled, you can create Secure Scheduler for Exchange Integrations to one or more Microsoft Exchange deployments.

In this topic:

- Supported Exchange deployments
- Supported clients
- Support for delegate access to calendars
- Network architecture and firewalls

Supported Exchange deployments

Secure Scheduler for Exchange is supported on the following Microsoft Exchange deployments:

- Microsoft 365
- Exchange 2016 (with the latest updates)
- Exchange 2019 (with the latest updates)

Supported clients

Exchange on-premises

The Secure Scheduler for Exchange add-in is supported on all Outlook clients that support the Microsoft Outlook add-in API. At the time of release, this includes the following clients:

- Outlook Web Application (OWA) when connected to any supported Microsoft Exchange on-premises deployment
- Office on Windows (Office 2016 and later)
- Office on Mac (Office 2016 and later)
- Outlook as part of Microsoft 365 on Windows and Mac.

Outlook apps for iOS and Android are not supported.

Microsoft 365

The Secure Scheduler for Exchange add-in uses APIs that are only supported in Outlook clients from 2021 LTSC and later. At the time of release, this includes the following clients:

- Office on the web (Office 365)
- Office on Windows (Office 2021 LTSC and later)
- Office on Mac (version 16.40 and later)
- Outlook as part of Microsoft 365 on Windows and Mac.

Outlook apps for iOS and Android are not supported.

There are some minor usability issues when using Outlook add-ins under certain circumstances; see Troubleshooting Secure Scheduler for Exchange for more information.

The Pexip add-in is dependent on the Microsoft Outlook add-in API. Any changes to the API should be backwards-compatible, but may impact the functionality of the Pexip add-in.

Support for delegate access to calendars

Pexip Infinity supports the use of the Secure Scheduler for Exchange add-in within delegate calendars to schedule meetings in single-use VMRs. Delegates cannot use the add-in to schedule meetings in the personal VMR of the person for whom they are a delegate.

To use delegate calendars with Secure Scheduler for Exchange:

- Delegate users must be set up in accordance with Microsoft's instructions for granting delegate access.
- Users must be using a version of Outlook that supports add-ins for delegates:
  - Exchange Online users must be using Outlook version 1910 (Build 12130.20272) or later.
  - Exchange on-premises 2016/2019 users must be using Outlook Version 2206 (Build 15330.20000) for the Current Channel and Version 2207 (Build 15427.20000) for the Monthly Enterprise Channel.
  - For more information on supported clients and platforms, see Microsoft's documentation.

Network architecture and firewalls

The diagram below summarizes the connectivity required between the components of the Pexip and Exchange/O365 deployments.

In this example, there are firewalls in place between the Pexip Infinity deployment and the Exchange and Office 365 deployments. Your own deployment may or may not have these, but in all cases you must ensure the following connections are permitted:

- from the Pexip Infinity Management Node to:
  - each Microsoft Exchange server: HTTPS, TCP port 443
  - **login.microsoftonline.com** (if you are using Office 365)
  - the Kerberos Key Distribution Center (KDC) (if Kerberos Authentication is enabled): UDP and TCP port 88
  - the KDC Proxy (if Kerberos Authentication and Kerberos KDC HTTPS proxy is enabled): HTTPS, TCP port 443
  - the load balancer (if you have one): HTTPS, TCP port 443
- from the Pexip Infinity Conferencing Nodes to the User OAuth token URI (if personal VMRs are enabled)
- from the Outlook clients to the hostname specified by the Add-in server FQDN. This must be reachable either directly, or by using split DNS to resolve to a Transcoding Conferencing Node, Proxying Edge Node, or reverse proxy: HTTPS, TCP port 443

- from the Outlook clients to **https://appsforoffice.microsoft.com**. This connection is required in order to use the JavaScript API for Office — for more information, see this Microsoft article.

  You can host these resources locally for deployments that are entirely offline. For more information, see Advanced options.

Using a load balancer in your Exchange deployment

Secure Scheduler for Exchange Integrations support the use of load balancers in front of the Exchange servers. However, NTLM Authentication will not work if a Layer 7 load balancer is in use. To work around this issue we recommend reconfiguring your load balancer to Layer 4.

In all deployments using load balancers, the FQDN of the load balancer must still be configured in the list of Exchange domains, even if the EWS URL uses the address of the load balancer.

Enabling Exchange Scheduler with personal VMRs

This topic gives an overview of the process and prerequisites for using Secure Scheduler for Exchange with personal VMRs. This allows users who already have their own personal VMR in your deployment to schedule meetings in that VMR.

You can also enable Secure Scheduler for Exchange with single-use VMRs, either instead of, or in addition to, personal VMRs.

In this topic:

- Overview
- Prerequisites
- Process flow - personal VMRs
- Next step

See Locating the add-in for a guide for end users on how to use the add-in.

Overview

Enabling the use of personal VMRs within the Secure Scheduler for Exchange service involves the following steps:

1. Either Configuring AD FS SSO for personal VMRs or Configuring Azure SSO for personal VMRs.
2. Configuring Pexip Infinity, including enabling personal VMRs and creating the Secure Scheduler for Exchange add-in.
3. Making the add-in available to users within your Microsoft Exchange deployment.

Prerequisites

- You must be using either AD FS or Azure Single Sign-On (SSO) for Outlook end users.
- There must be a that has an **Email address** configured with the user's Single Sign-on (SSO) email address (or alternatively, you must configure this feature to Allow new users, in which case the required record is created by Pexip Infinity).
- You must ensure that for every user who will be scheduling meetings in their personal VMRs, your deployment includes at least one VMR that has an **Owner's email address** configured with the user's Single Sign-on (SSO) email address. If there is more that one VMR configured with the user's email address, the user is presented with all the VMRs, and can choose which VMR to use for each particular meeting. Each one of these VMRs must also have at least one alias configured.

Process flow - personal VMRs

The diagram below outlines the process that is initiated when users activate the Pexip Secure Scheduler for Exchange add-in from their Outlook client and the **Personal VMR** option is used (either because this is the only option available to users, or because they have elected to use this option instead of a single-use VMR when both are available). This process is described in more detail in the paragraphs that follow.

User activates add-in

When a user activates the Pexip Secure Scheduler for Exchange add-in from within their Outlook client meeting request, the add-in sends a request to the reverse proxy or Conferencing Node identified by the **Add-in server FQDN** field. If a reverse proxy is used, it forwards the request on to an appropriate Conferencing Node. If personal VMRs are enabled, the user is offered the option to sign in (if they have not already done so).

Initial user sign-in

If the user has not previously signed in, they are presented with the sign in panel. Users' sign-in details are remembered and they should not need to sign in again (unless their User record gets deleted from Pexip Infinity) and they will instead be presented with a list of applicable VMRs.

The user must sign in with their SSO email address and password which has already been set up in your selected SSO provider. (For information on how to configure SSO, see Configuring AD FS SSO for personal VMRs or Configuring Azure SSO for personal VMRs.)

The Conferencing Node then checks whether there is already a Pexip Infinity User configured (**Users & Devices > Users**) with the same email address.

- If the user already exists, the Conferencing Node obtains VMRs.
- If the user does not already exist, but **Allow new users** (**System > Secure Scheduler For Exchange > Personal VMR Configuration > Allow New Users**) has been enabled, the Management Node will create a user with the SSO email address.
- If the user does not already exist and **Allow new users** is disabled, the user sees the configured Error getting personal VMRs message.

Conferencing Node obtains VMRs

The Conferencing Node checks every configured VMR and returns a list of all VMRs with an **Owner's email address** that matches the user's SSO email address.

The list of VMRs is presented to the user, who then selects which VMR to use for that particular meeting.

Add-in adds joining details to the meeting invitation

The add-in then applies the **Personal VMR joining instructions template** to generate the joining instructions for the selected VMR, and inserts these into the invitation body. It also uses the **Personal VMR location template** to generate a location, and inserts that into the **Location** field of the meeting invitation.

The user then adds their own content to the meeting invitation and sends it in the usual way.

Next step

Depending on your SSO provider, either Configuring AD FS SSO for personal VMRs or Configuring Azure SSO for personal VMRs.

Configuring AD FS SSO for personal VMRs

You can set up Secure Scheduler for Exchange to offer users the option to use their personal VMRs when scheduling meetings. This requires Single Sign-On (SSO).

This topic explains how to configure AD FS to enable SSO for Secure Scheduler for Exchange when using personal VMRs.

In this topic:

- How it works
- Prerequisites
- Setting up an OAuth 2.0 Client on Windows Server
- Next steps

How it works

The process of Secure Scheduler for Exchange users authenticating to Pexip Infinity via AD FS to obtain their personal VMRs works as follows:

1. When the Outlook add-in launches, it opens up a pop-up window for the user to sign in to AD FS using their SSO credentials.
2. After the user has successfully signed in, they are redirected back to the Outlook add-in, which then requests an access token from AD FS.
3. The AD FS server returns the AD FS access token to the Outlook add-in. This token proves that the user has successfully authenticated with AD FS.
4. Pexip Infinity then updates its record for that user so that the user does not need to log in again when they next activate the add-in.

This process means that:

- users' SSO credentials are not stored by Pexip Infinity
- if you delete and recreate a user's record on Pexip Infinity (**Users & Devices > Users**), they will need to sign in to AD FS again.

Prerequisites

Before you integrate your AD FS deployment with Pexip Infinity, you must make sure your AD FS deployment satisfies the following requirements.

AD FS version

You must be using a version of Windows Server that supports OAuth 2.0 with AD FS, i.e. Windows Server 2012 R2 or later.

Internet accessibility and security

Your Federation Service must be accessible by:

- all Pexip Infinity Conferencing Nodes or, if you are using a reverse proxy for Secure Scheduler for Exchange, all Conferencing Nodes to which the reverse proxy points
- all users who need to sign into AD FS to authenticate with Pexip Infinity.

In practice (and unless all users and Conferencing Nodes in your deployment are entirely internal) this means your Federation Service must be accessible from the internet. This raises security concerns, but Microsoft provide documentation about the recommended deployment of AD FS:

- Top-level AD FS documentation can be found at AD FS Overview.
- Note the network layout recommendations made in the Microsoft documentation, for example in this Legacy AD FS Federation Server Farm Using SQL Server article where all AD FS servers are deployed inside the corporate network and are load-balanced.

- To make the Federation Service accessible from the internet, separate servers in the DMZ can be installed with the Web Application Proxy (WAP) role, which proxy requests to the internal AD FS servers from the internet.
- Your Federation Service Name, e.g. adfs.example.com, must be routable from both inside and outside your corporate network.
  - Inside your corporate network, it should resolve directly to your AD FS server (or the IP address used to load balance multiple AD FS servers).
  - Outside your corporate network, it should resolve to your WAP servers in the DMZ.

  This requires the correct DNS configuration to be setup. Microsoft also provide documentation about this, for example in this Configure Name Resolution for a Federation Server Proxy in a DNS Zone That Serves Only the Perimeter Network article.
- Another very good source of information for creating a highly available AD FS deployment can be found in the series of blog posts at Windows Server 2012 R2 AD FS - Federated Web SSO.

  These posts refer to AD FS 3.0 on Windows Server 2012 R2, but they also apply to AD FS 4.0 on Windows Server 2016.

Certificates

Each AD FS server must be provided with a valid certificate which is trusted by your Pexip Infinity deployment. The subject of this certificate needs to match the Federation Service Name.

Setting up an OAuth 2.0 Client on Windows Server

To set up an OAuth 2.0 Client, use the appropriate set of instructions below for your version of AD FS and Windows Server.

Windows Server 2016 and later

Creating a Native Application

In this step you create a new application group with a new native application for the OAuth client — which is the Outlook add-in in this case.

1. Log on to a computer that can make configuration changes to your Federation Service. If your AD FS deployment uses Windows Internal Database (WID), this must be the Primary AD FS Server. If your AD FS deployment uses SQL Server then any AD FS server can make configuration changes.
2. From the top right of the Server Manager application window, select **Tools > AD FS Management**.
3. From the left panel, select **Application Groups** and then from the right panel select **Add Application Group**.
4. At the **Welcome** screen, enter a **Name** and **Description** for the application group. Then from the **Template** section, under **Standalone applications**, select ***Native application.***
5. At the **Native Application** screen, enter a **Name** for the application. A new **Client Identifier** is randomly generated for you (this will be required later when configuring the User OAuth client ID on Pexip Infinity.

   In the **Redirect URI** field, enter:

   https://<address>/api/client/v2/msexchange_schedulers/oauth_redirect
   where **<address>** is the Add-in server FQDN configured on Pexip Infinity, for example
   https://px01.vc.example.com/api/client/v2/msexchange_schedulers/oauth_redirect
6. At the **Summary** screen, review the settings and select **Next**.

The new application group, along with the new native application, is created.

Creating a Web API Resource

In this step you create a Web API within your application group. The Web API acts as the resource that is accessed when users authenticate to Pexip Infinity using their AD credentials.

1. On the AD FS Management Tool, from the left-hand panel select **Application Groups** and from the middle panel select the application group you have just created. From the right-hand panel, select **Properties**.
2. At the bottom of the **Properties** window, select **Add application...**
3. At the **Welcome** screen, from the **Template** section, select **Web API**.

4. At the **Configure Web API** screen, enter a **Name** and an **Identifier** (which must be in URL format). You must use this URL as the AD FS Resource Identifier on Pexip Infinity when configuring a Secure Scheduler for Exchange Integration.
5. At the **Choose Access Control Policy** screen select an appropriate policy. The default is ***Permit everyone***.
6. At the **Configure Application Permissions** screen, ensure the native application you created above appears in the list of **Client applications**. All the **Permitted scopes** should be deselected because Secure Scheduler for Exchange does not request any scopes.
7. At the **Summary** screen, review the entered settings then select **Next**.

The new Web API should now be created.

Configuring Claim Rules

In this step you configure an Issuance Transform Rule for the Web API. This rule specifies which claims should be sent to the Relying Party (i.e. which claims will be inside the OAuth token that is sent to Pexip Infinity).

Pexip Infinity requires certain claims to be present inside the token in order to establish the user's identity. These are claims that come from the user's Active Directory account.

1. Ensure there is an Attribute Store configured for Active Directory. To do this, from the AD FS Management Tool, in the left-hand panel expand **Service** and select **Attribute Stores**. Select the attribute store called Active Directory. Open its properties and ensure its **Attribute store type** is ***Active Directory***.

   If the Attribute Store is not present, add it by selecting **Add Attribute Store**. In the **Add An Attribute Store** window, enter a **Display name** of ***Active Directory*** and select an **Attribute store type** of ***Active Directory***.
2. Add the Issuance Transform Rule on the Web API you just created. To do this, from the AD FS Management Tool, at the bottom of the left-hand panel select **Application Groups**. Select the Application Group you just created and open its Properties.
3. Select the Web API you created earlier and then select **Edit**.
4. Select the **Issuance Transform Rules** tab and then select **Add Rule**.
5. Select a **Claim rule template** of ***Send Claims Using a Custom Rule*** and then select **Next**.
6. Enter a **Claim rule name** and enter the following as the **Custom rule**.

   ```
   c:[Type ==
   "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
   => issue(store = "Active Directory", types = ("email", "first_name", "last_name", "display_name"), query =
   ";mail,givenName,sn,displayName;{0}", param = c.Value);
   ```

   The above rule queries Active Directory for the attributes: **mail**, **givenName**, **sn** and **displayName**, and then maps them to the claims: **email**, **first_name**, **last_name** and **display_name** which will appear in the token payload that is returned when the user successfully logs in. The **email** claim is required by Pexip Infinity when verifying the token. If it is not present in the token, the user will fail to authenticate to Pexip Infinity.
7. Select **Finish**, and in the next window ensure you select **Apply**.

Checking and enabling AD FS endpoints

In this step you ensure that the appropriate AD FS endpoints are enabled to support Pexip's requirements. In the context of AD FS, an endpoint is a URL that AD FS is configured to serve.

To find the details of these AD FS endpoints:

1. From the Server Manager application window, select **Tools > AD FS Management**.
2. From the AD FS Management Tool, in the left-hand panel expand **AD FS > Service > Endpoints**.
3. Locate and check the following endpoint:
   - an **OAuth** type endpoint with path **/adfs/oauth2/**
     (this is used by users who sign in to AD FS)

   Ensure that this endpoint is **Enabled**. If you are using a Web Application Proxy (WAP) you must also ensure they are **Proxy Enabled**.

Determining Federation Service Properties

To configure AD FS SSO for personal VMRs on the Pexip Infinity Management Node, you must first determine your **Federation Service Name** (this is the FQDN that clients use to access AD FS). The Federation Service Name is used as the domain part of the User OAuth authorization URI and the User OAuth token URI. To check this:

- From the AD FS Management Tool, from the left-hand panel select the top level **AD FS** folder, and then select **Edit Federation Service Properties**.

AD FS 3.0 on Windows Server 2012 R2

Creating a Relying Party Trust

In this step you create a Relying Party Trust, which acts as the resource that is accessed when users authenticate to Pexip Infinity using their AD credentials.

1. Log on to a computer that can make configuration changes to your Federation Service. If your AD FS deployment uses Windows Internal Database (WID), this must be the Primary AD FS Server. If your AD FS deployment uses SQL Server then any AD FS server can make configuration changes.
2. From the Server Manager application window, select **Tools > AD FS Management**.
3. From the left-hand panel, expand **AD FS > Trust Relationships > Relying Party Trusts**. Then from the right-hand panel select **Add Relying Party Trust...**.
4. At the **Add Relying Party Trust Wizard** welcome screen, select **Start**.
5. At the **Select Data Source** screen, select **Enter data about the relying party manually** and select **Next**.
6. At the **Specify Display Name** screen, enter a **Display name** and **Note**, and select **Next**.
7. At the **Choose Profile** screen, select **AD FS profile** and select **Next**.
8. At the **Configure Certificate** screen, do not configure a certificate. Simply select **Next**.
9. At the **Configure URL** screen, do not enable support for either the WS-Federation Passive or the SAML 2.0 WebSO protocols. Simply select **Next**.
10. At the **Configure Identifiers** screen, enter a Relying Party Trust Identifier. This must be unique against all of your Relying Party Trusts, and must be in URL format.

    You will need to enter this later when adding an AD FS Resource Identifier on Pexip Infinity when configuring a Secure Scheduler for Exchange Integration.
11. At the **Configure Multi-factor Authentication Now?** screen, optionally enable multi-factor authentication. Enabling multi-factor authentication will affect how your users sign in to AD FS.
12. At the **Choose Issuance Authorization Rules** screen, select **Permit all users to access this relying party**.
13. At the **Ready To Add Trust** screen, review the settings and then select **Next**.
14. At the **Finish** screen, verify that the relying party trust was added successfully. Choose to **Open the Edit Claim Rules dialog...** then select **Close**.

Configuring Claim Rules

In this step you configure an Issuance Transform Rule for the Relying Party. This rule specifies which claims should be sent to the Relying Party (i.e. which claims will be inside the OAuth token that is sent to Pexip Infinity).

Pexip Infinity requires certain claims to be present inside the token in order to establish the user's identity. These are claims that come from the user's Active Directory account.

1. Ensure there is an Attribute Store configured for Active Directory. To do this, in the AD FS Management Tool, from the left-hand panel expand **AD FS > Trust Relationships > Attribute Stores**. Look for the attribute store called Active Directory. Open its properties and ensure its **Attribute store type** is ***Active Directory***.

   If the Attribute Store is not present, add it by selecting **Add Attribute Store**. In the **Add An Attribute Store** window, enter a **Display name** of ***Active Directory*** and select an **Attribute store type** of ***Active Directory***.
2. From the left-hand panel expand **AD FS > Trust Relationships > Relying Party Trusts**, select the Relying Party you just created, and from the right-hand panel select **Edit Claim Rules...**.
3. Select the **Issuance Transform Rules** tab, then select **Add Rule...**.
4. At the **Select Rule Template** screen, select a **Claim rule template** of ***Send Claims Using a Custom Rule***.
5. At the **Configure Rule** screen, enter a **Claim rule name**, and in the **Custom rule** section enter the following:

   ```
   c:[Type ==
   "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]
   ```

```
=> issue(store = "Active Directory", types = ("email", "first_name", "last_name", "display_name"), query =
";mail,givenName,sn,displayName;{0}", param = c.Value);
```

The above rule queries Active Directory for the attributes: **mail**, **givenName**, **sn** and **displayName**, and then maps them to the claims: **email**, **first_name**, **last_name** and **display_name** which will appear in the token payload that is returned when the user successfully logs in. The **email** claim is required by Pexip Infinity when verifying the token. If it is not present in the token, the user will fail to authenticate to Pexip Infinity.

6. Select **Finish**.
7. You are returned to the **Issuance Transform Rules** tab. Select **Apply**.

Checking and enabling AD FS endpoints

In this step you ensure that the appropriate AD FS endpoints are enabled to support Pexip's requirements. In the context of AD FS, an endpoint is a URL that AD FS is configured to serve.

To find the details of these AD FS endpoints:

1. From the Server Manager application window, select **Tools > AD FS Management**.
2. From the AD FS Management Tool, in the left-hand panel expand **AD FS > Service > Endpoints**.
3. Locate and check the following endpoint:
   - an **OAuth** type endpoint with path **/adfs/oauth2/**
     (this is used by users who sign in to AD FS)

   Ensure that this endpoint is **Enabled**. If you are using a Web Application Proxy (WAP) you must also ensure they are **Proxy Enabled**.

Determining Federation Service Properties

To configure AD FS SSO for personal VMRs on the Pexip Infinity Management Node, you must first determine your **Federation Service Name** (this is the FQDN that clients use to access AD FS). The Federation Service Name is used as the domain part of the User OAuth authorization URI and the User OAuth token URI. To check this:

- From the AD FS Management Tool, from the left-hand panel select the top level **AD FS** folder, and then select **Edit Federation Service Properties**.

Adding the OAuth 2.0 Client to AD FS

In this final step you add the OAuth 2.0 client to AD FS. This is done using the `Add-AdfsClient` PowerShell command.

A client ID is required to run this command. To obtain an appropriate client ID you can either:

- Use the client ID generated by Pexip Infinity when adding a Secure Scheduler for Exchange Integration. In this case you must now complete the configuration described in Adding a Secure Scheduler for Exchange Integration to Pexip Infinity, take note of the User OAuth client ID that is generated, and then return to this step to complete the AD FS configuration.
- Generate your own version 4 UUID using a tool such as https://www.uuidgenerator.net/. In this case, you should use the generated UUID to complete the AD FS configuration in this step, and then enter that same UUID as the User OAuth client ID when Adding a Secure Scheduler for Exchange Integration to Pexip Infinity.

After you have obtained a client ID, open PowerShell on your AD FS server, then run this command:

```
Add-AdfsClient -Name "<name>" -ClientId "<client id>" -RedirectUri "https://<address>/api/client/v2/msexchange_schedulers/oauth_
redirect" -Description "<description>"
```

where:

- `<name>` is a name of your choice used to identify the AD FS client
- `<client id>` is the the User OAuth client ID obtained in either of the two ways described above
- `<address>` is the Add-in server FQDN configured on Pexip Infinity (i.e. the FQDN of the reverse proxy or Conferencing Node that provides the add-in content)
- `<description>` is an optional description.

For example:

```
Add-AdfsClient -Name "Pexip Scheduling" -ClientId "141c7f59-790b-46f7-add1-7490ee4b489e" -RedirectUri
"https://tobyrp.rd.pexip.com/api/client/v2/msexchange_schedulers/oauth_redirect" -Description "The OAuth Client for Pexip
Scheduling personal VMRs"
```

To verify this worked, run `Get-AdfsClient` and make sure the client you just added is in the results.

Next steps

1. Configuring Pexip Infinity to integrate with your Microsoft Exchange deployment and create the Secure Scheduler for Exchange add-in.
2. Making the add-in available to users within your Microsoft Exchange deployment.

Configuring Azure SSO for personal VMRs

You can set up Secure Scheduler for Exchange to offer users the option to use their personal VMRs when scheduling meetings. This requires Single Sign-On (SSO).

This topic explains how to configure Azure to enable SSO for Secure Scheduler for Exchange when using personal VMRs.

In this topic:

- How it works
- Creating and configuring a new App Registration in Azure
- Taking note of configuration
- Next steps

How it works

The process of Secure Scheduler for Exchange users authenticating to Pexip Infinity via Azure to obtain their personal VMRs works as follows:

1. When the Outlook add-in launches, it opens up a pop-up window for the user to sign in to Azure using their SSO credentials.
2. After the user has successfully signed in, they are redirected back to the Outlook add-in, which then requests an access token from Azure.
3. The Azure server returns the Azure access token to the Outlook add-in. This token proves that the user has successfully authenticated with Azure.
4. Pexip Infinity then updates its record for that user so that the user does not need to log in again when they next activate the add-in.

This process means that:

- users' SSO credentials are not stored by Pexip Infinity
- if you delete and recreate a user's record on Pexip Infinity (**Users & Devices > Users**), they will need to sign in to Azure again.

Creating and configuring a new App Registration in Azure

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

4. In the **Register an application** panel, enter the following options:
   a. **Name**: this can be anything you wish. Our example uses ***Pexip Scheduling Personal VMRs***.
   b. **Supported account types**: select ***Accounts in this organizational directory only***.
   c. **Redirect URI**: from the drop-down menu, select ***Web***. The URI should be in the format:
   https://<address>/api/client/v2/msexchange_schedulers/oauth_redirect
   where **<address>** is the Add-in server FQDN configured on Pexip Infinity, for example
   https://px01.vc.example.com/api/client/v2/msexchange_schedulers/oauth_redirect

   Our example uses **https://pexip.example.com/api/client/v2/msexchange_schedulers/oauth_redirect**

   The **OAuth redirect URI** is the URI to which users are returned after they have successfully signed in to their account. It must use the same domain as the Add-in server FQDN configured on Pexip Infinity in order for Azure to validate the sign-in request.

5. Select **Register**.

   You can now configure your application.
6. From the panel on the left, select **Certificates & secrets**.
7. Select **New client secret**.
8. Enter a **Description**, under **Expires** select a duration in accordance with your organization's security policies, and select **Add**:

9. The new client secret will appear in the list at the bottom of the page. Before you navigate away from the page you must copy the string in the **Value** field (do **not** copy the string in the **Secret ID** field):

You must enter this as the **Azure OAuth client secret** when configuring the Secure Scheduler for Exchange Integration and enabling personal VMRs

10. The ***User.Read*** permission should have been added by default, with a type of ***Delegated***. To confirm this, from the panel on the left, select **API permissions** and ensure the appropriate permission is present.

    If it is not present, add it as follows:

    a. Select **Add a permission**.

    b. From the **Request API permissions** panel, select **Microsoft Graph**:

    c. From the **Request API permissions** page, select **Delegated permissions**, and from the list that appears, scroll down to the

User section and select ***User.Read***. Then select **Add permissions**:

Taking note of configuration

When you Configure the Secure Scheduler for Exchange Integration and enable personal VMRs using Azure as your authentication provider, you'll need to provide the following information from Azure (in addition to the **Client Secret** which you have already noted):

- **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 **User OAuth client ID** when configuring the Secure Scheduler for Exchange Integration and enabling personal VMRs.

- **OAuth 2.0 endpoints**. To find this information:
  a. In the Azure Portal, select **Azure Active Directory > App registrations**, select the app you have just registered, and select the **Endpoints** tab:

b. Copy the URL of the **OAuth 2.0 authorization endpoint (v2)**.

   Ensure that you use the URL for **... endpoint (v2)**, not **... endpoint (v1)**.

   You will need to enter this as the **User OAuth authorization URI** when configuring the Secure Scheduler for Exchange Integration and enabling personal VMRs.

c. Copy the URL of the **OAuth 2.0 token endpoint (v2)**

   Ensure that you use the URL for **... endpoint (v2)**, not **... endpoint (v1)**.

   You will need to enter this as the **User OAuth token URI** when configuring the Secure Scheduler for Exchange Integration and enabling personal VMRs.

Next steps

1. Configuring Pexip Infinity to integrate with your Microsoft Exchange deployment and create the Secure Scheduler for Exchange add-in.
2. Making the add-in available to users within your Microsoft Exchange deployment.

Enabling Exchange Scheduler in single-use VMRs

This topic gives an overview of the process and prerequisites for using Secure Scheduler for Exchange with single-use VMRs. These VMRs are created dynamically for a specific meeting, and are only available for the duration of that meeting (and any scheduled recurrences).

You can also enable Secure Scheduler for Exchange with personal VMRs, either instead of, or in addition to, single-use VMRs.

In this topic:

- Overview
- Process flow - single-use VMRs
- PXPS:- and TOK:- security tags
- Using application impersonation
- Recurring meetings
- PINs and authentication
- Next step

Overview

Enabling the use of single-use VMRs within your Secure Scheduler for Exchange service involves the following steps:

1. Configuring Microsoft Exchange on-premises or Configuring Office 365 with an equipment resource and either a service account or app registration, and enabling the use of application impersonation.

   This step is required if you wish to enable single-use VMRs. It is not required if you are **only** enabling personal VMRs in your deployment.
2. Configuring Pexip Infinity, including enabling single-use VMRs, providing the information required to connect to your Microsoft Exchange deployment, and generating the associated add-in.
3. Making the add-in available to users within your Microsoft Exchange deployment.

See Locating the add-in for a guide for end users on how to use the add-in.

When Secure Scheduler for Exchange has been implemented, you will be able to view scheduled VMRs in the same way as existing VMRs. For more information, see Managing Exchange Scheduler conferences.

Process flow - single-use VMRs

The diagram below outlines the process that is initiated when users activate the Pexip Secure Scheduler for Exchange add-in from their Outlook client and select the **Single-use VMR** option; this process is described in more detail in the paragraphs that follow.

Outlook user activates add-in
Add-in obtains a security token using the configured *Add-In authentication method*
Add-in sends request to Reverse Proxy or Conferencing Node specified by *Server FQDN*
Conferencing Node receives request and contacts Management Node
Management Node online?
Yes
No
Management Node generates alias, join instructions and *PXPS:* ID, sends to Conferencing Node
Conferencing Node replies to add-in with joining instructions and *PXPS:* ID
Conferencing Node replies to add-in with placeholder text
Add-in inserts joining instructions and *PXPS:* ID into body of meeting request
Add-in inserts placeholder text and security token (*TOK:*) into body of meeting request
Add-in adds room resource as required attendee
User sends meeting request
Request sent to all participants including room resource
User sends meeting update/ cancellation
Management Node monitors room resource mailbox
New meeting request?
No
Cancellation or update?
Yes
update
cancellation
Does new request include a *PXPS:* ID?
No
Yes
Management Node checks security token and replaces it with *PXPS:* ID, alias, and joining instructions
Management Node updates VMR
Management Node via Exchange sends email with joining instructions to invitees on behalf of user
Management Node generates VMR with alias
Meeting is available for invitees to join at requested time (plus buffers)
One-off meeting/ last in a series?
No
Yes
Management Node deletes VMR
Key
End user / participant actions
Add-in / Exchange actions
Pexip Infinity Conferencing Node/ Management Node actions

The fields described in the following process are configured on the Management Node via **System > Secure Scheduler For Exchange IntegrationS**.

Add-in requests aliases

When a user activates the Pexip Secure Scheduler for Exchange add-in from within their Outlook client meeting request, the add-in sends a request to the reverse proxy or Conferencing Node identified by the **Add-in server FQDN** field. If a reverse proxy is used, it forwards the request on to an appropriate Conferencing Node. The Conferencing Node then forwards the request to the Management Node and asks for an alias for the VMR to use for the meeting. If no Conferencing Node can be contacted at this stage, the user sees a message in the add-in pane of their client informing them that the add-in is not available.

Management Node generates aliases and join instructions

If the Management Node is online, it generates a unique number for the VMR and then uses this to create two aliases (one with a domain appended and one without), using the information in the following fields:

- Scheduled alias prefix
- Scheduled alias suffix length
- Scheduled alias domain

For example, if we have configured a **prefix** of ***555***, a **suffix length** of ***6*** and a **domain** of ***example.com***, the two aliases created for one particular VMR could be:

- 555123456
- 555123456@example.com

The alias suffix length used must result in aliases of a sufficient length to ensure **security** and **quantity** within your deployment:

- **security**: the longer the alias, the less susceptible your deployment is to external attacks by third parties (such as SIP scanners) attempting to guess meeting aliases
- **quantity**: if your deployment is generating a high volume of scheduled meetings, longer aliases mean you are less likely to run out of numbers.

The Management Node also generates some joining instructions for the VMR, based on the VMR's alias and the **Joining instructions template** field. This information is sent back to the Conferencing Node which forwards it to the add-in as a response to the request. The add-in then inserts the join instructions into the top of the meeting request email. A unique security ID (in the format **PXPS:-<xxx>#**) is also obtained from the Management Node and inserted into the email; this text must not be edited or deleted.

If the Management Node cannot be contacted by the Conferencing Node, the Conferencing Node will respond to the add-in with the placeholder text from the **Placeholder instructions text** field. A multi-line security token in the format **TOK:-<xxx>#** (obtained directly from the Microsoft Exchange server by the add-in) is also inserted into the email; again, this text must not be edited or deleted.

The add-in also adds the equipment resource as a required attendee. This equipment resource must be included in all subsequent requests relating to this meeting.

Management Node creates VMRs

When the user sends the meeting request, a message is sent to the equipment resource's mailbox. The Management Node monitors this mailbox using EWS impersonation. For each new meeting request received, the Management Node generates a VMR with the required aliases (one numeric, the other numeric with domain). For recurring meetings, the same alias is used for all instances of the meeting. For meeting updates and cancellations, the Management Node will update the associated VMR accordingly.

The VMR is created as soon as the Management Node receives the meeting request, but meeting participants cannot use the VMR until they are within the period of time configured by the **Join before buffer** and **Join after buffer**. Participants who have successfully joined the VMR can continue to use it until the last participant has left; the conference will not be terminated automatically by the scheduling service, although it may be automatically terminated for other reasons.

If the Management Node has been offline, when it comes online it connects to the equipment resource's mailbox and reads the requests. At this point it creates and updates VMRs. For meetings that were not assigned a **PXPS:-** ID, the Management Node will generate an alias and create the VMR. It then replaces the **Placeholder instructions text** and security token (**TOK:-<xxx>#**) in the original meeting request with a **PXPS:-** ID, along with details of the aliases and the joining instructions. The Management Node then

uses impersonation to send the updated request to all attendees on behalf of the meeting organizer (the request is shown in the organizer's sent items folder). This updated request is also sent to the equipment resource's mailbox.

PXPS:- and TOK:- security tags

To ensure that the Secure Scheduler for Exchange service processes only those meeting requests created using the add-in, and in order to track each meeting request, each request contains a security tag in the format of either **TOK:-** or **PXPS:-**.

When the add-in is activated, it obtains a **TOK:-** security tag from the user's Microsoft Exchange server. This **TOK:-** is then passed to the Management Node when it is asked to allocate an alias for the VMR. If the **TOK:-** is valid, the Management Node will generate the alias and return a **PXPS:-** ID to the Conferencing Node. The **PXPS:-** ID is passed back to the add-in and inserted into the body of the meeting request.

In normal circumstances, end users will not see the **TOK:-** tag because the request to generate a VMR alias is actioned immediately by the Management Node. However, if the Management Node is offline at the time the request is sent, and therefore a VMR alias is not generated immediately, the **TOK:-** will be inserted into the body of the meeting request. When the Management Node comes back online, it will process the requests; for any with a valid **TOK:-** it generates a VMR alias and **PXPS:-** ID and sends an updated meeting request that includes these.

Note that the **TOK:-** is only valid for a limited amount of time. If the Management Node processes the request after the **TOK:-** expires, it will be considered invalid. In such cases, the meeting organizer must create a new meeting request using the add-in.

Using application impersonation

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

The Secure Scheduler for Exchange feature uses EWS impersonation of **resources** (required) and **users** (optional) for the following purposes:

- To **access the mailbox of the equipment resource** used for Secure Scheduler for Exchange. This impersonation is required in order for the Secure Scheduler for Exchange feature to work.
- To **send emails on behalf of all Secure Scheduler for Exchange users**. This impersonation is only required in the two scenarios described below. If the service account / app is not permitted to impersonate users, the users can still schedule and update meetings as usual, but the following scenarios are not supported:
  - If an invitation was sent when the Management Node was offline. In this case, when the Management Node comes back online it will generate a new alias and joining instructions for the meeting. It then updates the meeting invitation using impersonation, so that it appears as though the meeting organizer is sending out the updated joining instructions to all the attendees.
  - When the scheduling recovery tool is run after the Management Node has been restored from a backup. The recovery tool queries the room resource mailbox and finds all meetings that have previously been accepted, and checks whether they are in the scheduling service's database. If not, the Management Node will generate a new alias and new joining instructions for that meeting. This meeting update is sent using impersonation so that it appears as though the meeting organizer is sending out the updated joining instructions to all the attendees.

We recommend permitting the service account / app to access all users (in addition to the equipment resource) to enable the full functionality of the Secure Scheduler for Exchange feature, but we also provide instructions for permitting the service account / app to access the equipment resource only if this is more appropriate for your environment.

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

- Impersonation and EWS in Exchange for guidelines on when to use impersonation in your Exchange service applications.
- Exchange Impersonation vs. Delegate Access for information on the differences between impersonation and delegate access.

Recurring meetings

For recurring meetings in single-use VMRs, the Secure Scheduler for Exchange service creates a single VMR which exists until after the last meeting in the series has taken place. However, participants can only join this VMR during any of the scheduled meeting times.

When editing a recurring meeting, the option to edit **This and all following events** is not supported. You must instead choose to edit **All events in the series**.

PINs and authentication

By default, single-use VMRs are created without PINs, meaning participants do not need to enter a PIN and all participants join with Host privileges.

You can optionally require that all participants must authenticate with an Identity Provider in order to join a single-use VMR; this is configured on a per-Secure Scheduler for Exchange Integration basis, using the Identity Provider group and Other participants settings.

When enabled for single-use VMRs, the requirement for participant authentication applies to all meetings scheduled using the add-in associated with that integration. To allow users to choose between scheduling authenticated and unauthenticated meetings, create two integrations and associated add-ins, one with authentication enabled and one without.

In addition, after an individual single-use VMR has been created, you can edit the configuration for that VMR to add Host PINs and Guest PINs, and/or require participant authentication.

Next step

- Configuring Microsoft Exchange on-premises or Configuring Office 365 .

Configuring Exchange on-premises for scheduling

To enable users to schedule meetings in single-use VMRs in an Exchange on-premises environment using a service account, you must first configure Exchange on-premises appropriately. The steps are as follows, described in more detail in the sections that follow:

1. Creating a service account. The Pexip Infinity Secure Scheduler for Exchange feature uses a unique service account to log into Exchange.
   - If you are also using the One-Touch Join feature you cannot use the same service account, because the configuration is different.
2. Configuring Application Impersonation on the service account. This allows the service account to impersonate all users and resources in Exchange.
   - As an alternative, you can allow the service account to impersonate the Secure Scheduler for Exchange equipment resource only, but this limits the functionality of the feature. For more information, see Using application impersonation.
3. Create an equipment resource. This resource is added as an attendee to all meetings scheduled using the Pexip add-in.
4. Configure the equipment resource.
5. Enable the authentication method used for the service account — basic authentication, NTLM authentication, or Kerberos authentication.

PowerShell commands are provided for all steps; parameters to be replaced with your own relevant information are shown in the format **<your info here>**.

Prerequisites

Before you start, ensure you have access to your Exchange Admin Center (EAC) web interface, and access to Exchange Management PowerShell.

Creating a service account

In this step, you create the service account used by the Secure Scheduler for Exchange integration.

Secure Scheduler for Exchange uses a service account to log into Exchange. You can use an existing service account (the same service account can be used by more than one integration), or create a new account.

You need to provide the authentication credentials of this service account when configuring Pexip Infinity to integrate with a Microsoft Exchange deployment.

- A single service account can be used by more than one integration.
- Changes to the service account may take some time to take effect, depending on the cache lifetime configured on the Microsoft Exchange server.

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.

5. Select **Save**.

PowerShell

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@example.com -Password $password -Alias pexip -FirstName Service -LastName Account -DisplayName "Service Account"
```

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

Configuring Application Impersonation on the service account

In this step you configure the service account with a **Role** of ***Application Impersonation***. This allows the service account to impersonate the equipment resource that is used for Secure Scheduler for Exchange, and all users who will be using this service. Alternatively, you can allow impersonation of the equipment resource only, but this limits the functionality of the Secure Scheduler for Exchange feature. For more information, see Using application impersonation.

Impersonating all users and resources

The Secure Scheduler for Exchange service impersonates users when sending email updates containing joining instructions (in cases where these instructions could not be added at the time of scheduling). This makes it appear to the recipients that the joining instructions were sent from the meeting organizer, instead of the service account. The service account also impersonates the Secure Scheduler for Exchange equipment resource to read the calendar.

To check if your service account has Application Impersonation already configured, use the PowerShell command:

```
Get-ManagementRoleAssignment -RoleAssignee "<email_of_service_account>" -Role ApplicationImpersonation | Format-List
```

Below is an example of the output of the command when the service account already has Application Impersonation configured:

```
[PS] C:\Windows\system32>Get-ManagementRoleAssignment -RoleAssignee pexip@rd.pexip.com -Role ApplicationImpersonation |
Format-List

RunspaceId                    : 439dc5ce-7895-4341-ad2e-8182741cd596
DataObject                    : PexipSchedulingService
User                          : rd.pexip.com/Users/Pexip Exchange Service
AssignmentMethod              : Direct
Identity                      : PexipSchedulingService
EffectiveUserName             : Pexip Exchange Service
AssignmentChain               :
RoleAssigneeType              : User
RoleAssignee                  : rd.pexip.com/Users/Pexip Exchange Service
Role                          : ApplicationImpersonation
RoleAssignmentDelegationType  : Regular
CustomRecipientWriteScope     :
CustomConfigWriteScope        :
RecipientReadScope            : Organization
ConfigReadScope               : None
RecipientWriteScope           : Organization
ConfigWriteScope              : None
Enabled                       : True
RoleAssigneeName              : Pexip Exchange Service
IsValid                       : True
ExchangeVersion               : 0.11 (14.0.550.0)
Name                          : PexipSchedulingService
DistinguishedName             : CN=PexipSchedulingService,CN=Role Assignments,CN=RBAC,CN=Pexip Dev,CN=Microsoft
                                Exchange,CN=Services,CN=Configuration,DC=rd,DC=pexip,DC=com
Guid                          : 4bf0f9ac-94cf-4056-94c1-1403cc3ba630
ObjectCategory                : rd.pexip.com/Configuration/Schema/ms-Exch-Role-Assignment
ObjectClass                   : {top, msExchRoleAssignment}
WhenChanged                   : 11/05/2017 14:51:17
WhenCreated                   : 11/05/2017 14:51:17
WhenChangedUTC                : 11/05/2017 13:51:17
WhenCreatedUTC                : 11/05/2017 13:51:17
OrganizationId                :
OriginatingServer             : tobyad.rd.pexip.com
ObjectState                   : Unchanged


[PS] C:\Windows\system32>_
```

If the service account does not have Application Impersonation configured, then the above command will not return anything at all. If this is the case, enable Application Impersonation as follows:

```
New-ManagementRoleAssignment -name:"<role_name>" -Role:ApplicationImpersonation -User:"<email_of_service_account>"
```

For example:

```
New-ManagementRoleAssignment -name:PexipSchedulingService -Role:ApplicationImpersonation -User:pexip@exchange.example.com
```

This enables the service account to impersonate all users and resources in the organization.

For more information on these commands, see Microsoft help:

- Configuring Application Impersonation: https://msdn.microsoft.com/en-us/library/office/dn722376%28v=exchg.150%29.aspx
- `Get-ManagementRoleAssignment` command: https://technet.microsoft.com/en-us/library/dd351024(v=exchg.150).aspx
- `New-ManagementRoleAssignment` command: https://technet.microsoft.com/en-us/library/dd335193(v=exchg.150).aspx
- `Set-ManagementRoleAssignment` command (used if you need to edit the role assignment): https://technet.microsoft.com/en-us/library/dd335173(v=exchg.150).aspx

Impersonating resources only

If you do not want to permit the service account to access all users and resources in your organization, you can instead permit it to access the Secure Scheduler for Exchange equipment resource only, but this limits the functionality of the feature. For more information, see Using application impersonation.

To do this, you must first create the equipment resource and assign it an email address.

Then use the following PowerShell command:

```
$restriction_filter = "UserPrincipalName -eq ""<email_of_equipment_resource>"""
$scheduling_service_account = "<email_of_service_account>"
$management_scope_to_create = "Pexip Scheduling Management Scope"
$impersonation_role_name_to_create = "Pexip Scheduling Impersonation"

New-ManagementScope -Name $management_scope_to_create -RecipientRestrictionFilter $restriction_filter

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

Creating an equipment resource

In this step you create an equipment resource used by Secure Scheduler for Exchange. This resource is added as an attendee to all meetings scheduled using the Pexip add-in. The scheduling service monitors the equipment resource's mailbox, processes all meeting requests sent to it, and schedules the meetings as appropriate.

The equipment resource is added as a resource attendee to all Secure Scheduler for Exchange meetings. Users will see replies from this resource when it accepts or rejects a meeting request.

Each equipment resource can be used by only one Secure Scheduler for Exchange Integration. If you have separate test and development environments, you must use a different equipment resource for each.

Previously we recommended using a room resource, but this may cause issues when using the Room Finder tool. Thus, we now recommend using an equipment resource.

You can create an equipment resource using either EAC or PowerShell, as follows:

EAC	PowerShell

1. Log in to your **Exchange Admin Center**.
2. Go to **Recipients > Resources**.
3. Select **Add an equipment resource**.
4. Give the equipment resource a **Name** and **email address**.
The equipment name will appear as the location of any meeting requests, and as a recipient.
 	
5. Select **Save**.
	This command creates an equipment resource with the specified **Name**, **Alias** and **Display Name**. **Name** and **Display Name** should be the same, and will appear as the location of any meeting requests, and as a recipient. The **Alias** (also known as the mail nickname) is used as the email address.
	`New-Mailbox -Equipment -Name "<Equipment Name>" -Alias "<Equipment Alias>" -DisplayName "<Equipment Name>"`
	For example:
	`New-Mailbox -Equipment -Name "Pexip Scheduling Service" -Alias pexip.scheduling -DisplayName "Pexip Scheduling Service"`

Configuring the equipment resource

In this step you configure calendar processing to disable automatic processing for the equipment resource, so that the processing can be done by the scheduling service. You must also configure it to permit conflicts, because meetings may be scheduled at the same time by different users. You can optionally hide the resource mailbox's availability to improve the usability of the Scheduling Assistant.

Calendar processing

This configuration is done using the following PowerShell command:

```
Set-CalendarProcessing -Identity "<email_of_equipment_resource>" -AutomateProcessing None -AllowConflicts $true -
BookingWindowInDays 1080 -MaximumDurationInMinutes 0 -AllowRecurringMeetings $true -EnforceSchedulingHorizon $false -
ScheduleOnlyDuringWorkHours $false -ConflictPercentageAllowed 100 -MaximumConflictInstances 2147483647
```

To verify that the above command has configured everything correctly, use the PowerShell command:

```
Get-CalendarProcessing -Identity "<email_of_equipment_resource>" | Format-List
```

The output should look something like this:

```
[PS] C:\Windows\system32>Get-CalendarProcessing -Identity pexipvmrschedulingservice@rd.pexip.com | Format-List

RunspaceId                          : 439dc5ce-7895-4341-ad2e-8182741cd596
AutomateProcessing                  : None
AllowConflicts                      : True
BookingWindowInDays                 : 1080
MaximumDurationInMinutes            : 0
AllowRecurringMeetings              : True
EnforceSchedulingHorizon            : False
ScheduleOnlyDuringWorkHours         : False
ConflictPercentageAllowed           : 100
MaximumConflictInstances            : 2147483647
ForwardRequestsToDelegates          : True
DeleteAttachments                   : True
DeleteComments                      : True
RemovePrivateProperty               : True
DeleteSubject                       : True
AddOrganizerToSubject               : True
DeleteNonCalendarItems              : True
TentativePendingApproval            : True
EnableResponseDetails               : True
OrganizerInfo                       : True
ResourceDelegates                   : {}
RequestOutOfPolicy                  : {}
AllRequestOutOfPolicy               : False
BookInPolicy                        : {}
AllBookInPolicy                     : True
RequestInPolicy                     : {}
AllRequestInPolicy                  : False
AddAdditionalResponse               : False
AdditionalResponse                  :
RemoveOldMeetingMessages            : True
AddNewRequestsTentatively           : True
ProcessExternalMeetingMessages      : False
RemoveForwardedMeetingNotifications : False
MailboxOwnerId                      : rd.pexip.com/Users/Pexip VMR Scheduling Service
Identity                            : rd.pexip.com/Users/Pexip VMR Scheduling Service
IsValid                             : True
ObjectState                         : Changed


[PS] C:\Windows\system32>_
```

Hiding resource availability in scheduling assistant

When viewing meetings in Outlook's Scheduling Assistant, the equipment resource and its availability will appear in the list of attendees. Because the same equipment resource is used for all scheduled meetings, it may show as unavailable if there are other meetings scheduled at the same time, which may be confusing to users.

To avoid this, you can hide the availability of the equipment resource's mailbox, so that it is always shown as "unknown". This is done using the PowerShell command:

```
Set-MailboxFolderPermission -Identity "<email_of_equipment_resource>:\Calendar" -User Default -AccessRights None
```

More information

For more information on these commands, see Microsoft help:

- `Set-CalendarProcessing` command: https://learn.microsoft.com/en-us/powershell/module/exchange/set-calendarprocessing
- `Get-CalendarProcessing` command: https://learn.microsoft.com/en-us/powershell/module/exchange/get-calendarprocessing
- `Set-MailboxFolderPermission` command: https://learn.microsoft.com/en-us/powershell/module/exchange/set-mailboxfolderpermission

Enabling authentication

In this step you enable your Exchange on-premises deployment to support your chosen authentication method for the service account. Secure Scheduler for Exchange uses basic authentication by default, but you can elect to use NTLMv2 authentication or Kerberos authentication instead.

For all forms of authentication, Pexip Infinity stores the credentials in encrypted form. We recommend that your service account user has a strong and unique password.

For Basic and NTLM authentication, all authentication is carried out over a secure TLS channel. For Kerberos, authentication messages are sent in plain text but Kerberos tickets are encrypted using the service account password. Depending on the security requirements of your organization, you may wish to enable the Kerberos KDC HTTPS proxy, in which case all Kerberos Authentication messages are sent over a secure TLS channel.

Basic authentication

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

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

Windows Service Manager	PowerShell

1. Go to the Windows server on which Exchange is installed and open the Service Manager.
2. Select the server on which Exchange is installed, and right-click to select **Computer Management**.
3. From the panel on the left, select **Services and Applications > Internet Information Services (IIS) Manager**.
4. Expand the options and select **Sites > Default Web Site > EWS**.
5. Select the **Authentication** button in the main pane.
6. Find **Basic Authentication** in the list and ensure it is ***Enabled***. (If not, right-click and select ***Enable***.)
7. Select **Save**.
	This command enables basic authentication for EWS on a specific server:
	`Set-WebServicesVirtualDirectory -Identity "<server>\EWS (Default Web Site)" -BasicAuthentication $true`
	For example, if your server name is PEXCHANGE then:
	`Set-WebServicesVirtualDirectory -Identity "PEXCHANGE\EWS (Default Web Site)" -BasicAuthentication $true`

NTLMv2 authentication

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

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

Kerberos authentication

Kerberos authentication is enabled by default in Windows and you should not need to do anything specific to use it.

Pexip Infinity supports Kerberos KDC in Active Directory on Windows server 2012 R2 and newer.

We strongly recommend that Kerberos Pre-Authentication is enabled on the service account. On most Windows systems this should already be enabled by default.

When configuring Pexip Infinity to use Kerberos as an authentication method for your Secure Scheduler for Exchange Integration, note that:

- the **Kerberos realm** is usually your Exchange domain, in upper case
- the **Kerberos KDC** (key distribution center) is usually your Active Directory server.

You can optionally use Kerberos over TLS; to do so you must enable and configure the KDC Proxy on your Windows server and then enable and configure use of the proxy on your Secure Scheduler for Exchange Integration.

Pexip Infinity supports the following encryption types for Kerberos:

- aes256-cts-hmac-sha384-192
- aes128-cts-hmac-sha256-128
- aes256-cts-hmac-sha1-96
- aes128-cts-hmac-sha1-96

Viewing the equipment resource's mailbox

There may be occasions, such as when troubleshooting, that you want to view the equipment resource's mailbox or calendar. To do this, you first need to assign full access to the equipment resource's mailbox to a delegate account, and then view the mailbox or calendar using the delegate account. (The delegate account could be the service account (if you are using one), or it could be, for example, an administrator's account.)

Next steps

1. Configuring Pexip Infinity to integrate with your Microsoft Exchange deployment and create the Secure Scheduler for Exchange add-in.
2. Making the add-in available to users within your Microsoft Exchange deployment.

Configuring Office 365 for scheduling using app permissions

To enable users to schedule meetings in single-use VMRs in an Office 365 environment you must first complete the following steps, described in more detail in the sections that follow:

1. Creating an equipment resource. This resource is added as an attendee to all meetings scheduled using the Pexip add-in.
2. Configuring the equipment resource.
3. Creating an app registration, used by the scheduling service.
4. Creating a management scope, which permits the app to impersonate the members of the scheduling distribution group.

PowerShell commands are provided for some steps; parameters to be replaced with your own relevant information are shown in the format **<your info here>**.

Prerequisites

Before you start, ensure you can access:

- your Microsoft 365 Admin Center: https://admin.cloud.microsoft.com/.
- Either:
  - the Azure portal at https://portal.azure.com, or
  - the Microsoft Entra admin centre at https://entra.microsoft.com/.

You also need to have a remote PowerShell session to your Exchange server. See these Microsoft articles about connecting to Exchange online and Microsoft 365 with PowerShell for more information.

Creating an equipment resource

In this step you create an equipment resource used by Secure Scheduler for Exchange. This resource is added as an attendee to all meetings scheduled using the Pexip add-in. The scheduling service monitors the equipment resource's mailbox, processes all meeting requests sent to it, and schedules the meetings as appropriate.

The equipment resource is added as a resource attendee to all Secure Scheduler for Exchange meetings. Users will see replies from this resource when it accepts or rejects a meeting request.

Each equipment resource can be used by only one Secure Scheduler for Exchange Integration. If you have separate test and development environments, you must use a different equipment resource for each.

Previously we recommended using a room resource, but this may cause issues when using the Room Finder tool. Thus, we now recommend using an equipment resource.

You can create an equipment resource using either the Office 365 admin portal or PowerShell, as follows:

Microsoft 365	PowerShell

To create the equipment resource via the Microsoft 365 admin portal:
1. Go to https://www.microsoft365.com/ and log in as the administrator.
2. Go to the Microsoft 365 admin centre by selecting the **Admin** tile from the left-hand menu (this takes you to https://admin.microsoft.com/#/homepage).
3. From the left-hand menu, select **Resources > Rooms & Equipment** and then **Add resource**.
4. Select a **Resource type** of ***Equipment***.
5. Assign an appropriate **Name** and **Email** address and select **Add**.
The equipment name will appear as the location of any meeting requests, and as a recipient.
 	
The equipment resource now appears in the list of resources.
	This command creates an equipment resource with the specified **Name**, **Alias** and **Display Name**. **Name** and **Display Name** should be the same, and will appear as the location of any meeting requests, and as a recipient. The **Alias** (also known as the mail nickname) is used as the email address.
	`New-Mailbox -Equipment -Name "<Equipment Name>" -Alias "<Equipment Alias>" -DisplayName "<Equipment Name>"`
	For example:
	`New-Mailbox -Equipment -Name "Pexip Scheduling Service" -Alias pexip.scheduling -DisplayName "Pexip Scheduling Service"`

Configuring the equipment resource

In this step you configure calendar processing to disable automatic processing for the equipment resource, so that the processing can be done by the scheduling service. You must also configure it to permit conflicts, because meetings may be scheduled at the same time by different users. You can optionally hide the resource mailbox's availability to improve the usability of the Scheduling Assistant.

Calendar processing

This configuration is done using the following PowerShell command:

```
Set-CalendarProcessing -Identity "<email_of_equipment_resource>" -AutomateProcessing None -AllowConflicts $true -
BookingWindowInDays 1080 -MaximumDurationInMinutes 0 -AllowRecurringMeetings $true -EnforceSchedulingHorizon $false -
ScheduleOnlyDuringWorkHours $false -ConflictPercentageAllowed 100 -MaximumConflictInstances 2147483647
```

To verify that the above command has configured everything correctly, use the PowerShell command:

```
Get-CalendarProcessing -Identity "<email_of_equipment_resource>" | Format-List
```

The output should look something like this:

```
PS C:\WINDOWS\system32> Get-CalendarProcessing -Identity pexipvmrschedulingservice@pexip1.onmicrosoft.com | Format-List

RunspaceId                        : bb53f66c-015d-414a-a11f-057381dd8398
AutomateProcessing                : None
AllowConflicts                    : True
BookingWindowInDays               : 1080
MaximumDurationInMinutes          : 0
AllowRecurringMeetings            : True
EnforceSchedulingHorizon          : False
ScheduleOnlyDuringWorkHours       : False
ConflictPercentageAllowed         : 100
MaximumConflictInstances          : 2147483647
ForwardRequestsToDelegates        : True
DeleteAttachments                 : True
DeleteComments                    : True
RemovePrivateProperty             : True
DeleteSubject                     : True
AddOrganizerToSubject             : True
DeleteNonCalendarItems            : True
TentativePendingApproval          : True
EnableResponseDetails             : True
OrganizerInfo                     : True
ResourceDelegates                 : {}
RequestOutOfPolicy                : {}
AllRequestOutOfPolicy             : False
BookInPolicy                      : {}
AllBookInPolicy                   : True
RequestInPolicy                   : {}
AllRequestInPolicy                : False
AddAdditionalResponse             : False
AdditionalResponse                :
RemoveOldMeetingMessages          : True
AddNewRequestsTentatively         : True
ProcessExternalMeetingMessages    : False
RemoveForwardedMeetingNotifications : False
MailboxOwnerId                    : Pexip VMR Scheduling Service
Identity                          : Pexip VMR Scheduling Service
IsValid                           : True
ObjectState                       : Changed


PS C:\WINDOWS\system32>
```

Hiding resource availability in scheduling assistant

When viewing meetings in Outlook's Scheduling Assistant, the equipment resource and its availability will appear in the list of attendees. Because the same equipment resource is used for all scheduled meetings, it may show as unavailable if there are other meetings scheduled at the same time, which may be confusing to users.

To avoid this, you can hide the availability of the equipment resource's mailbox, so that it is always shown as "unknown". This is done using the PowerShell command:

```
Set-MailboxFolderPermission -Identity "<email_of_equipment_resource>:\Calendar" -User Default -AccessRights None
```

More information

For more information on these commands, see Microsoft help:

- `Set-CalendarProcessing` command: https://learn.microsoft.com/en-us/powershell/module/exchange/set-calendarprocessing
- `Get-CalendarProcessing` command: https://learn.microsoft.com/en-us/powershell/module/exchange/get-calendarprocessing
- `Set-MailboxFolderPermission` command: https://learn.microsoft.com/en-us/powershell/module/exchange/set-mailboxfolderpermission

Creating an app registration

In this step, you create a Microsoft Entra ID app registration used by Secure Scheduler for Exchange.

You must use the settings from this app registration when configuring the integration with the Exchange server OAuth options.

Creating a new app registration

1. Log in to either:
   - the Azure portal at https://portal.azure.com, and under **Azure services** select **Microsoft Entra ID**. From the side panel select **App registrations**.
   - the Microsoft Entra admin centre at https://entra.microsoft.com/, and from the side panel select **Applications > App registrations**.
2. Select **New registration**.
3. 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 Scheduling app permissions***.
   b. **Supported account types**: for most customers, ***Accounts in this organizational directory only (... only - Single tenant)*** will be appropriate.
   c. **Redirect URI**: leave blank.

4. Select **Register**.

   A new panel opens where you can configure your application.
5. From the **Overview** page, note the following:
   - **Display name**. You will need this later to search for the app.
   - **Application (client) ID**: used when configuring the management scope and when configuring Pexip Infinity.

6. Select the **Endpoints** tab and copy the **OAuth 2.0 token endpoint (v2)** value. You will need to enter this as the OAuth token endpoint when configuring Pexip Infinity:

Ensure that you use the URL for **... endpoint (v2)**, not **... endpoint (v1)**.

7. From the panel on the left, select **API permissions**.
8. Select **Add a permission**:

9. From the **Request API permissions** panel, select **APIs my organization uses**, search for **Office 365 Exchange Online** and select it:

10. Select **Application permissions**, and from the **Select permissions** list, expand **Other permissions** and select **full_access_as_app**:

11. Select **Add permissions**.

The **full_access_as_app** permission now shows in the list of API permissions. **Admin consent required** is **Yes**, and **Status** is **Not granted for <tenant name>**:

We recommend that you **do not** grant admin consent. Doing so gives the app permission to access every mailbox in your organization. Instead, we recommend you create a Management Scope to restrict the app's access to the equipment resource and scheduling users' mailboxes only.

Creating a client secret

1. From the side panel, select **Certificates & secrets** and select the **Client secrets** tab. Select **New client secret**:

2. Enter a **Description** and select an **Expires** option, and select **Add**.

   The maximum duration before expiry is 24 months. When this secret expires you will need to add a new one, and update the Pexip Infinity config accordingly.

3. The new client secret appears in the list.Before you navigate away from the page you must copy the string in the **Value** field (do

not copy the string in the **Secret ID** field). You will need to enter this as the OAuth client secret when configuring Pexip Infinity:

Creating a management scope

In this step you configure a management scope which you use to specify which mailboxes the app can impersonate.

These steps are required if you have not granted admin consent when adding permissions for the app. If you have granted admin consent, which we do not recommend, the app will already have access to every mailbox in your organization.

As a minimum, the app requires permission to access the equipment resource mailbox. We also recommend that the app is given permission to access all users who will be using Secure Scheduler for Exchange as this will allow meeting instructions to be added to meetings which were created while the Management Node was offline. For more information about how Secure Scheduler for Exchange uses EWS Impersonation, see Using application impersonation.

Copy the app's Enterprise Object ID

First you must obtain the Enterprise Object ID of your app. To do this:

1. Log in to either:
   - the Azure portal at https://portal.azure.com, and under **Azure services** select **Microsoft Entra ID**. From the side panel select **Enterprise applications**.
   - the Microsoft Entra admin centre at https://entra.microsoft.com/, and from the side panel select **Identity > Applications > Enterprise applications**.
2. Search for and select the app.

3. Copy and save the **Object ID**. When creating the management scope, use this as the app's Enterprise Object ID:

Creating a distribution list

The easiest way to apply the management scope to all Secure Scheduler for Exchange users is to use a group in the form of a distribution list. This group should include the equipment resource mailbox as well as all users whose mailboxes should be accessed by the scheduling app.

To create a distribution list:

1. Log in to the Microsoft 365 admin center at https://admin.cloud.microsoft.com/.
2. From the side panel, select **Teams & Groups > Active teams & groups** and select the **Distribution list** tab.
3. Select **Add a distribution list**:

4. Follow the steps to give the list a name and assign owners and members. Ensure that you add all users, as well as the equipment resource mailbox, to the list.
5. Assign an email address to the new group. You'll use this in the next step, creating a management scope.

   We recommend that you do not allow people outside your organization to send email to the Distribution group:

Creating and applying the management scope

We recommend that you use combined PowerShell commands to create and configure the management scope. This allows you to use variables, thus reducing possible copy and paste errors.

For information on connecting to Exchange Online using PowerShell, see https://learn.microsoft.com/en-us/powershell/exchange/connect-to-exchange-online-powershell.

For the following PowerShell command, you must assign values specific to your deployment to the following variables:

- $app_id: the app's **Application (client) ID** noted down when creating the app registration.
- $enterprise_object_id: the app's Enterprise Object ID.
- $scheduling_dist_list_email: the email of the distribution list created earlier, whose members you want the scheduling service to be able to impersonate.

You also assign names for the following:

- $service_principal_to_create
- $management_scope_to_create
- $management_role_assignment_to_create

Run the following PowerShell, replacing our examples with your own:

```
$app_id = "<Application (client) ID>"
$enterprise_object_id = "<Enterprise Object ID>"
$scheduling_dist_list_email = "<distribution list email address>"

$service_principal_to_create = "Pexip VMR Scheduling SP"
$management_scope_to_create = "Pexip VMR Scheduling MS"
$management_role_assignment_to_create = "Pexip VMR Scheduling MRA"

New-ServicePrincipal -AppId $app_id -ObjectId $enterprise_object_id -DisplayName $service_principal_to_create

$pexip_scheduling_group = Get-DistributionGroup -Identity $scheduling_dist_list_email
$pexip_scheduling_group_dn = $pexip_scheduling_group.DistinguishedName
$restriction_filter = "MemberOfGroup -eq ""$pexip_scheduling_group_dn"""

New-ManagementScope -Name $management_scope_to_create -RecipientRestrictionFilter $restriction_filter

New-ManagementRoleAssignment -Name $management_role_assignment_to_create -App $service_principal_to_create -Role "Application
EWS.AccessAsApp" -CustomResourceScope $management_scope_to_create
```

Taking note of configuration

When you configure the associated Secure Scheduler for Exchange Integration, you'll need to provide the following information from Azure:

- **Application (client) ID**: this was generated for you by Azure when you created the app registration:

You can find this again by going to **Applications > App Registrations** and selecting the app registration. It is shown on the overview page under **Application (client) ID**.

You must enter this as the OAuth client ID when configuring the integration.

- **OAuth client secret**: this was generated for you by Azure when you created the app registration.

  You can only view the value immediately after creation.

  You must enter the string from the **Value** field as the OAuth client secret when configuring the integration.

- **OAuth 2.0 token endpoint (v2)**: this was generated for you by Azure when you created the app registration:

You can find this again by going to to **Applications > App Registrations** and selecting the app registration. Select the **Endpoints** tab and copy the value under **OAuth 2.0 token endpoint (v2)**.

You must enter this value as the OAuth token endpoint when configuring the integration.

Viewing the equipment resource's mailbox

There may be occasions, such as when troubleshooting, that you want to view the equipment resource's mailbox or calendar. To do this, you first need to assign full access to the equipment resource's mailbox to a delegate account, and then view the mailbox or calendar using the delegate account. (The delegate account could be, for example, an administrator's account.)

Next steps

1. Configuring Pexip Infinity to integrate with your Microsoft Exchange deployment and create the Secure Scheduler for Exchange add-in.
2. Making the add-in available to users within your Microsoft Exchange deployment.

Configuring Office 365 for scheduling using a service account

The Secure Scheduler for Exchange feature previously supported the use of a service account with the Application Impersonation role. However, Application Impersonation role assignment to service accounts has been deprecated by Microsoft. From May 2024, Microsoft prevented new Application Impersonation role assignments. From February 2025, this role was removed completely (for more information, see Microsoft's announcement). For Office 365, all new Secure Scheduler for Exchange deployments must use the alternative option using application permissions, and all existing deployments must be updated to enable this option as soon as possible.

Configuring Pexip Infinity for Exchange Scheduler

This topic describes the configuration required on your Pexip Infinity deployment to support the Secure Scheduler for Exchange feature.

Prerequisites

- To enable **single-use VMRs**: you must first complete the steps in either Configuring Exchange on-premises for scheduling (for Exchange on-premises environments), or Configuring Office 365 for scheduling using app permissions (for Office 365 environments).
- To enable **personal VMRs**: you must first complete the steps in either Configuring AD FS SSO for personal VMRs or Configuring Azure SSO for personal VMRs.
- For **Microsoft 365** environments, you must first configure an authentication option for the add-in, as per Add-in authentication for O365.

Adding a Secure Scheduler for Exchange Integration to Pexip Infinity

A Secure Scheduler for Exchange Integration defines a specific connection between your Pexip Infinity deployment and a Microsoft Exchange deployment. In some cases a single Pexip Infinity deployment requires more than one Secure Scheduler for Exchange Integration.

Adding a new Secure Scheduler for Exchange Integration involves the following steps:

- deciding whether VMRs used by the Secure Scheduler for Exchange service will be the users' personal VMR(s), single-use (dynamic) VMRs, or both
- (for single-use VMRs) adding details of your Microsoft Exchange deployment and accounts, defining how the single-use VMRs are configured, and configuring the template to use for joining instructions
- (for personal VMRs) adding details of your authentication provider (ADFS or Azure), and defining how the join instructions and meeting location are generated from the existing personal VMR details
- configuring the authentication method, wording and icon used by the add-in.

From the Pexip Infinity Management Node go to **System > Secure Scheduler For Exchange** and select **Add Secure Scheduler for Exchange Integration**. Complete the following fields:

Option	Description	Notes
Name	The name used to refer to this integration.	
Description	An optional description of this integration.	
Enable single-use VMRs	Enable this option to allow Outlook users to schedule meetings in single-use (randomly generated) VMRs.	
Enable personal VMRs	Enable this option to allow Outlook users to schedule meetings in their personal VMRs.	
Exchange server information (available and required when single-use VMRs are enabled)		

EWS URL	The URL used to connect to Exchange Web Services (EWS) on the Microsoft Exchange server.
	• For Office 365 deployments this is **https://outlook.office365.com/EWS/Exchange.asmx**
	• For Exchange on-prem this is usually **https://<exchange_server>/EWS/Exchange.asmx**
	  ○ If you have multiple Exchange servers in your deployment, this can be the URL of EWS on any one of the servers.
	  ○ If you are using a load balancer, this can use the FQDN of the load balancer within the URL (e.g. **https://<load_balancer>/EWS/Exchange.asmx**).	

Option	Description	Notes
Authentication method	The method used to authenticate to Exchange. For more information, see Enabling authentication. • ***Basic Authentication***: only supported in on-premises Exchange environments. Uses a service account which has been granted the Application Impersonation role, with a username and password authenticated using Basic authentication. • ***NTLM Authentication***: only supported in on-premises Exchange environments. Uses a service account which has been granted the Application Impersonation role, with a username and password authenticated using NTLMv2. ⓘ NTLM Authentication will not work if a Layer 7 load balancer is in use. To work around this issue we recommend reconfiguring your load balancer to Layer 4. • ***Kerberos Authentication***: only supported in on-premises Exchange environments. Uses a service account which has been granted the Application Impersonation role, with a username and password authenticated using Kerberos. • ***OAuth (Delegate access with a service account)***: only supported in Office 365 Exchange Online. Uses a service account which has been granted the Application Impersonation role, and is authenticated using OAuth 2.0. ⓘ The Application Impersonation role assignment to service accounts has been deprecated by Microsoft. From May 2024, Microsoft prevented new Application Impersonation role assignments. From February 2025, this role was removed completely (for more information, see Microsoft's announcement). For Office 365, all new Secure Scheduler for Exchange deployments must use the ***OAuth (Application permission without a service account) option***, and all existing deployments must be updated to enable this option as soon as possible. • ***OAuth (Application permission without a service account)***: only supported in Office 365 Exchange Online. Uses an App Registration with relevant Application permissions to allow use of EWS Impersonation. If this is a new Secure Scheduler for Exchange deployment, you must select this option. For more information, see Configuring Office 365 for scheduling using app permissions.	
Basic authentication **and** ***NTLM authentication*** **options**		
Service account username †	The username of the service account used by this integration. This is the username you assigned when Creating a service account for Exchange on-premises. • If you are using NTLMv2, this must be in the format **name@domain.com** or **name@domain** (i.e. without the TLD). • For other on-premises deployments, the format may be either **domain\name** or **name@domain**, depending on your domain. ⓘ A single service account can be used by more than one Secure Scheduler for Exchange Integration.	
Service account password	The password of the service account used by this integration. This is the password you assigned when Creating a service account.	
Kerberos authentication **options**		

Option	Description	Notes
Service account username †	The username of the service account used by this integration. This is the username you assigned when Creating a service account for Exchange on-premises. For Kerberos, this is normally in the format **name@REALM** where **REALM** is your upper-case Kerberos realm. A single service account can be used by more than one Secure Scheduler for Exchange Integration.	
Service account password	The password of the service account used by this integration. This is the password you assigned when Creating a service account.	
Kerberos realm	The Kerberos Realm, which is usually your Exchange domain in upper case. In most cases this should match the REALM part of the service account username.	
Kerberos Exchange SPN	The Exchange Service Principal Name (SPN). If left blank, the hostname of the URL used to connect to Exchange is used as the SPN. You may need to configure this setting if the SPN(s) in your Exchange environment do not match the URL hostname used by Pexip Infinity to connect to Exchange. The format of the SPN is normally the protocol and hostname of the service separated by the @ character, for example **HTTP@my-hostname.example.com**.	
Enable Kerberos KDC HTTPS proxy	If enabled, all communication to the Kerberos Key Distribution Center (KDC) will go through an HTTPS proxy and all traffic to the KDC will be encrypted using TLS.	
Kerberos KDC address	(Available when **Kerberos KDC HTTPS proxy** has **not** been enabled) The address of the Kerberos Key Distribution Center (KDC). This is normally the FQDN of the KDC server, which is normally the Active Directory server.	
Kerberos KDC HTTPS proxy URL	(Available when **Kerberos KDC HTTPS proxy** has been enabled) The URL of the Kerberos Key Distribution Center (KDC) HTTPS proxy. This must begin with **https://** and is typically in the format **https://<KDC Proxy Server FQDN>/KdcProxy**.	
Verify KDC HTTPS proxy SSL certificate using custom CA	(Available when **Kerberos KDC HTTPS proxy** has been enabled) Determines how the KDC HTTPS proxy SSL certificate is verified. Enabled: the certificate is verified using the configured Root Trust CA Certificates. Disabled: the certificate is verified using the system-wide default set of trusted certificates.	
***OAuth (Delegate access with a service account* options**		
Service account username †	The username of the service account used by this integration. This is the username you assigned when Creating a service account for O365. For Office365 deployments, the format is usually **name@domain**. A single service account can be used by more than one Secure Scheduler for Exchange Integration.	
OAuth client ID †	Enter the **Application (client) ID** that was generated for you by Azure when you saved the App Registration. For more information, see Configuring Office 365 for scheduling using a service account.	

Option	Description	Notes
OAuth redirect URI	Enter the **Redirect URI** that you used when Enabling OAuth authentication for Office 365. This must be in the format **https://<Management Node Address>/admin/platform/msexchangeconnector/oauth_redirect/**. The **OAuth redirect URI** is the page on the Administrator interface to which the Pexip Infinity administrator will be returned after they have successfully signed in to the service account. Because it is a page on the Management Node, this URI is internal to your deployment and only needs to be accessible from the administrator's web browser; you do not need to make it externally accessible. This URI must be the same on Azure and Pexip Infinity in order for Azure to validate the sign-in request.	
OAuth authorization endpoint	Enter the URL of the **OAuth 2.0 authorization endpoint (v1)** (see Taking note of configuration). Ensure that you use the URL for **... endpoint (v1)**, not **... endpoint (v2)**.	
OAuth token endpoint †	Enter the URL of the **OAuth 2.0 token endpoint (v1)** (see Configuring Office 365 for scheduling using a service account) Ensure that you use the URL for **... endpoint (v1)**, not **... endpoint (v2)**.	
***OAuth (Application permission without a service account)* options**		
OAuth client ID †	Enter the **Application (client) ID** that was generated for you by Azure when you saved the App Registration. For more information, see Taking note of configuration.	
OAuth client secret	The string from the **Value** field of the client secret which you generated when creating a Microsoft Entra ID App Registration.	
OAuth token endpoint †	Enter the URL of the **OAuth 2.0 token endpoint (v2)** (see Taking note of configuration) Ensure that you use the URL for **... endpoint (v2)**, not **... endpoint (v1)**.	
All authentication options		
Mailbox name	The name of the equipment resource used by this integration. This is the name you assigned when Creating an equipment resource in Exchange on-premises or Creating an equipment resource in O365. An equipment mailbox must only be used by a single Secure Scheduler for Exchange Integration.	
Mailbox email address	The email address of the equipment resource used by this integration. This is the email address you assigned when Creating an equipment resource in Exchange on-premises or Creating an equipment resource in O365.	
Single-use VMR configuration (available and required when single-use VMRs are enabled)		
Conference name template	A jinja2 template that is used to generate the name of scheduled conferences. The name is used by the Management Node to identify the conference, and may also appear to conference participants (depending on the endpoint being used). Note that conference names must be unique, so a random number may be appended if the name that is generated is already in use by another service (Virtual Meeting Room, Virtual Auditorium, Virtual Reception, scheduled conference, Media Playback Service, or Test Call Service). Default: ***{{subject}} ({{organizer_name}})***	Accepted variables: • subject • organizer_email • organizer_name • alias • numeric_alias

Option	Description	Notes
Conference description template	A jinja2 template that is used to generate the description of scheduled conferences. Default: ***Scheduled Conference booked by {{organizer_email}}***	Accepted variables: • subject • organizer_email • organizer_name • alias • numeric_alias
Conference subject template	A jinja2 template that is used to generate the subject field of scheduled conferences. This is shown on the Management Node when viewing information about the conference, and by default uses the subject line of the meeting invitation so the default should be deleted or amended if you do not want administrators to be able to view the meeting subject. Default: ***{{subject}}***	Accepted variables: • subject • organizer_email • organizer_name • alias • numeric_alias
Scheduled alias prefix	The prefix to use when generating aliases for scheduled conferences. Note: this must be between 1 and 8 characters long.	
Scheduled alias suffix length	The length of the random number suffix part of aliases generated for scheduled conferences. This must be a number between 5 and 15. Default: ***6***. We recommend you use a long suffix for security purposes, and to ensure you have sufficient aliases for your deployment. For more information, see Management Node generates aliases and join instructions.	
Scheduled alias domain	The domain to use when generating aliases for scheduled conferences.	
Conference alias description template	A Jinja2 template that is used to produce the description of the scheduled conference alias. Note that this description may be displayed to users of registered Pexip clients and apps who are performing a directory search. Default: ***Scheduled Conference booked by {{organizer_email}}***	Accepted variables: • subject • organizer_email • organizer_name • alias • numeric_alias
Identity Provider group	The set of Identity Providers used to authenticate participants attempting to join scheduled conferences. If this is blank, participants are not required to authenticate. For more information, see About participant authentication.	

Option	Description	Notes
Other participants	(Available when an **Identity Provider Group** is selected) Determines whether participants joining a SSO-protected service from devices other than the web app (for example SIP or H.323 endpoints) are allowed to dial in to the service. • ***Disallow all***: these devices are placed in a waiting room where they must wait to be admitted by a Host. • ***Allow if trusted***: these devices may join the service if they are locally registered. They must still enter a Host PIN or Guest PIN if either is required. All other devices are placed in a waiting room where they must wait to be admitted by a Host. Default: ***Disallow all***	
Join before buffer	The number of minutes before the meeting's scheduled start time that participants can join the VMR. Range: ***0*** to ***180***. Default: ***30***.	
Join after buffer	The number of minutes after the meeting's scheduled end time that participants can join the VMR. Range: ***0*** to ***180***. Default: ***60***.	
Scheduled conference theme	The theme used by all conferences scheduled using the add-in. For more information, see Customizing images and voice prompts using themes. Default: ***<use Default theme>*** (the global default theme is used).	
Personal VMR configuration (available and required when personal VMRs are enabled)		
Allow new users	Enable this option to allow users who do not have an existing Pexip Infinity User record (**Users & Devices > Users**) to access the Outlook add-in. When these users sign in, Pexip Infinity will create a new user record for them. Disable this option to allow only those users with an existing User record to access the Outlook add-in.	
Authentication provider	The method by which users sign into the Outlook add-in.	
User OAuth authorization URI	The authorization URI of the OAuth application used to authenticate users when signing in to the Outlook add-in. **Azure** This is in the format **https://login.microsoftonline.com/<UUID>/oauth2/v2.0/authorize** • To find this in Azure, see Taking note of configuration. **AD FS** This is in the format **https://<Federation Service Name>/adfs/oauth2/authorize** • To find this in AD FS with Windows Server 2016 or later, see Determining Federation Service Properties • To find this in AD FS with Windows Server 2012, see Determining Federation Service Properties	

Option	Description	Notes
User OAuth token URI	The token URI of the OAuth application used to authenticate users when signing in to the Outlook add-in. **Azure** This is in the format **https://login.microsoftonline.com/<UUID>/oauth2/v2.0/token** • To find this in Azure, see Taking note of configuration. **AD FS** This is in the format **https://<Federation Service Name>/adfs/oauth2/token** • To find this in AD FS with Windows Server 2016 or later, see Determining Federation Service Properties • To find this in AD FS with Windows Server 2012, see Determining Federation Service Properties	
User OAuth client ID	The client ID of the OAuth application used to authenticate users when signing in to the Outlook add-in. • To find this in Azure, see Taking note of configuration. • To find this in AD FS with Windows Server 2016 or later, see Creating a Native Application • For AD FS with Windows Server 2012, you can either use the default client ID shown here, or enter the one you generated earlier. For more information, see Adding the OAuth 2.0 Client to AD FS.	
AD FS Resource Identifier	(Applies when an **Auth Provider** of ***AD FS*** is selected) The URL which identifies the OAuth 2.0 resource in AD FS. • For AD FS with Windows Server 2016 or later, this should be the identifier you gave when Creating a Web API Resource. • For AD FS with Windows Server 2012, this should be the identifier you gave when Creating a Relying Party Trust.	
Azure OAuth client secret	(Applies when an **Auth Provider** of ***Azure*** is selected) The string from the **Value** field of the client secret of the OAuth application used to authenticate users when signing in to the Outlook add-in. For more information, see Creating and configuring a new App Registration in Azure.	

Option	Description	Notes
Personal VMR joining instructions template	A jinja2 template that is used to generate the joining instructions that are added by Secure Scheduler for Exchange to the body of the meeting request when a personal VMR is used. For more details on constructing the URLs used for joining from a web browser, see Creating preconfigured links to launch conferences via Connect apps. Default: `{% if domain_aliases %}` `  {% set alias = domain_aliases[0] %}` `{% elif other_aliases %}` `  {% set alias = other_aliases[0] %}` `{% else %}` `  {% set alias = numeric_aliases[0] %}` `{% endif %}` `{% if (not allow_guests) and pin %}` `  {% set meeting_pin = pin %}` `{% elif allow_guests and guest_pin %}` `  {% set meeting_pin = guest_pin %}` `{% else %}` `  {% set meeting_pin = "" %}` `{% endif %}` ` ` `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `<b>Please join my Pexip Virtual Meeting Room in one of the following ways:</b> ` `From a VC endpoint or a Skype/Lync client: ` `{{alias}} ` ` `From a web browser: ` `<a href="https://{{addin_server_domain}}/webapp/#/?conference={{alias}}">https://{{addin_server_domain}}/webapp/#/?conference={{alias}}</a> ` ` `From a Pexip Infinity Connect client: ` `pexip://{{alias}} ` ` `{% if numeric_aliases %}` `From a telephone: ` `[Your number], then {{numeric_aliases[0]}} # ` ` `{% endif %}` `{% if meeting_pin %}` `Please join using the PIN <b>{{meeting_pin}}</b> ` ` ` `{% endif %}` `</div>`	Accepted variables: • aliases • domain_aliases • numeric_aliases • other_aliases • addin_server_domain • name • description • service_type • pin • allow_guests • guest_pin • owners_email

Option	Description	Notes

Personal VMR location template	A jinja2 template that is used to generate the text that is inserted into the **Location** field of the meeting request when a personal VMR is used. The output of this should be a single line of text.
	
	Default:
	
	`{% if domain_aliases %}`
	`  {% set alias = domain_aliases[0] %}`
	`{% elif other_aliases %}`
	`  {% set alias = other_aliases[0] %}`
	`{% else %}`
	`  {% set alias = numeric_aliases[0] %}`
	`{% endif %}`
	`https://{{addin_server_domain}}/webapp/#/?conference={{alias}}`
		Accepted variables:
		• aliases
		• domain_aliases
		• numeric_aliases
		• other_aliases
		• addin_server_domain
		• name
		• description
		• service_type
		• pin
		• allow_guests
		• guest_pin
		• owners_email

Personal VMR name template	A jinja2 template that is used to generate the name of the personal VMR, as it appears on the button offered to users when selecting which VMR to use. By default, this uses the **Name** configured for the user's personal VMR. We recommend that the output is a single line. Default: `{{name}}`	Accepted variables: • aliases • domain_aliases • numeric_aliases • other_aliases • addin_server_domain • name • description • service_type • pin • allow_guests • guest_pin • owners_email

Option	Description	Notes
Personal VMR description template	A jinja2 template that is used to generate the text that appears when hovering over the button offered to users when selecting which VMR to use. By default this uses the **Description** configured for the user's personal VMR. Default: `{{description}}`	Accepted variables: • aliases • domain_aliases • numeric_aliases • other_aliases • addin_server_domain • name • description • service_type • pin • allow_guests • guest_pin • owners_email
Add-in configuration		
Add-in authentication token type *	The type of token the Outlook add-in uses to authenticate to Pexip. ***Exchange Identity Token***: required for (and only supported in) Exchange on-premises environments. ***SSO Token***: required for Microsoft 365 Exchange Online environments using Microsoft Office 2021 LTSC. Not supported in Exchange on-premises environments. ***NAA (Nested App Authentication) Token***: recommended for all other Microsoft 365 Exchange Online environments. Not supported in Exchange on-premises environments. For full instructions on configuring add-in authentication in Microsoft 365 environments, see Add-in authentication for O365.	
***SSO Token* additional options**		
Add-in auth application ID *	The Application (client) ID which was generated when creating the App Registration in Microsoft Entra for add-in authentication.	For details about where to find this configuration, see Configuring Pexip Infinity
Add-in auth OIDC Metadata URL	The OpenID Connect metadata document copied from the App Registration created in Microsoft Entra for add-in authentication.	
***NAA (Nested App Authentication) Token* additional options**		
Add-in auth application ID	The Application (client) ID which was generated when creating the App Registration in Microsoft Entra for add-in authentication.	For details about where to find this configuration, see Configuring Pexip Infinity
Add-in auth NAA Web API application ID	The Application (client) ID for the NAA Web API which was generated when creating the App Registration in Microsoft Entra.	
Add-in auth authority URL	The Authority URL copied from the App Registration created in Microsoft Entra for add-in authentication.	
Add-in auth OIDC Metadata URL	The OpenID Connect metadata document copied from the App Registration created in Microsoft Entra for add-in authentication.	
All add-in configuration options		

Option	Description	Notes
Add-in server FQDN *	The FQDN of the reverse proxy or Conferencing Node (which can be either a Proxying Edge Node or Transcoding Conferencing Node) that provides the add-in content. This reverse proxy or Conferencing Node must: • be reachable by all Outlook clients, whether they are located on an internal network or on the public internet • have installed a valid, trusted certificate; in particular, there must be a server certificate with a subject name that matches the FQDN, and which is signed by a trusted root CA. Pexip's Reverse Proxy and TURN Server v3 and later supports the Secure Scheduler for Exchange feature; we recommend v5 for additional security. If you do not have a reverse proxy in your deployment, you can choose any Conferencing Node that meets the above criteria.	
Add-in provider name *	The name of the organization which provides the add-in. Default: ***Pexip***.	
Add-in display name *	The display name of the add-in. Default: ***Pexip Scheduling Service***.	
Add-in description *	The description of the add-in. Maximum length: 250 characters. Default: ***Turns meetings into Pexip meetings***.	
Add-in group label *	The name of the group in which to place the add-in button on desktop clients. Default: ***Pexip Meeting***.	
Add-in button label *	The label for the add-in button on desktop clients. Default: ***Create a Pexip Meeting***.	
Add-in supertip title *	The title of the supertip help text for the add-in button on desktop clients. Default: ***Makes this a Pexip Meeting***.	
Add-in supertip text *	The text of the supertip for the add-in button on desktop clients. Default: ***Turns this meeting into an audio or video conference hosted in a Pexip VMR. The meeting is not scheduled until you select Send.***	
Add-in pane title	The title of the add-in on the side pane. Default: ***Add a VMR***.	
Add-in pane description	The description of the add-in on the side pane. Default: ***This assigns a Virtual Meeting Room for your meeting***.	
Add-in pane single-use VMR button label	The label of the button on the side pane. Default: ***Add a Single-use VMR***.	
Add-in pane success heading	The message that appears on the side pane when an alias has been obtained successfully from the Management Node. Default: ***Success***.	
Already video meeting heading	The heading that appears on the side pane when the add-in is activated after an alias has already been obtained for the meeting. Default: ***VMR already assigned***.	

Option	Description	Notes
Unable to add joining instructions heading	The heading that appears on the side pane when the Management Node cannot be contacted to obtain an alias. Default: ***Cannot assign a VMR right now***.	
General error heading	The heading that appears on the side pane when an error occurs trying to add the joining instructions. Default: ***Error***.	
Success message	The message that appears on the side pane when an alias has been obtained successfully from the Management Node. Default: ***This meeting is now set up to be hosted as an audio or video conference in a Virtual Meeting Room. Please note this conference is not scheduled until you select Send.***	
Already video meeting message	The message that appears on the side pane when the add-in is activated after an alias has already been obtained for the meeting. Default: ***It looks like this meeting has already been set up to be hosted in a Virtual Meeting Room. If this is a new meeting, select Send to schedule the conference.***	
Unable to add joining instructions message	The message that appears on the side pane when the Management Node cannot be contacted to obtain an alias. Maximum length: 250 characters. Default: ***Sorry, we are unable to assign a Virtual Meeting Room at this time. Select Send to schedule the meeting, and all attendees will be sent joining instructions later.***	
Error inserting single-use VMR message	The message that appears on the side pane when an error occurs trying to add the joining instructions of a single-use VMR. Default: ***There was a problem adding the joining instructions. Please try again.***	
Add-in pane personal VMR button label	The label of the button on the side pane used to add a personal VMR. Default: ***Add a Personal VMR***	
Add-in pane sign in button label	The label of the button on the side pane requesting users to sign in to obtain the list of their personal VMRs. Default: ***Sign In***	
Select personal VMR message	The message that appears on the side pane requesting users to select a personal VMR to use for the meeting. Default: ***Select the VMR you want to add to the meeting***	
No personal VMR message	The message that appears on the side pane when the user has no personal VMRs. Default: ***You do not have any personal VMRs***	
Error getting personal VMRs message	The message that appears on the side pane when an error occurs trying to obtain a list of the user's personal VMRs. Default: ***There was a problem getting your personal VMRs. Please try again.***	
Error signing in message	The message that appears on the side pane when an error occurs trying to sign the user in. Default: ***There was a problem signing you in. Please try again.***	
Error inserting personal VMR meeting message	The message that appears on the side pane when an error occurs trying to add the personal VMR details to the meeting. Default: ***There was a problem adding the joining instructions. Please try again.***	

Option	Description	Notes
Add-in image icon	Select the image file to use as the add-in icon. Images must be in PNG file format and 80 x 80 pixels in size. Note that Outlook clients may cache the add-in icon, so it may be some time after uploading a new icon that it appears to end users. You can resolve this by deleting the cache. Default:	
Single-use VMR email text (available and required when single-use VMRs are enabled; for more information, see Formatting the email text)		
Single-use VMR joining instructions template	A jinja2 template that is used to generate the joining instructions that are added by Secure Scheduler for Exchange to the body of the meeting request when a single-use VMR is used. Note that the `{{alias_uuid}}` variable, which inserts the **PXPS:** token, **must** be included. For examples of templates that use images and other formatting, see Example joining instructions. Default: ` ` `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `<b>Please join my Pexip Virtual Meeting Room in one of the following ways:</b> ` ` `From a VC endpoint or a Skype/Lync client: ` `{{alias}} ` ` ` `From a web browser: ` `<a href="https://{{addin_server_domain}}/webapp/#/?conference={{alias}}">https://{{addin_server_domain}}/webapp/#/?conference={{alias}}</a> ` ` ` `From a Pexip Infinity Connect client: ` `pexip://{{alias}} ` ` ` `From a telephone: ` `[Your number], then {{numeric_alias}} # ` ` ` `{{alias_uuid}} ` ` `	Accepted variables: • alias • addin_server_domain • numeric_alias • alias_uuid
Placeholder instructions text	The text that is added by Secure Scheduler for Exchange to email messages when the actual joining instructions cannot be obtained. Default: `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `This meeting will be hosted in a Virtual Meeting Room. Joining instructions will be ` `sent to you soon in a separate email. ` `</div>`	
Accept new single meeting template	A jinja2 template that is used to produce the message sent to meeting organizers when Secure Scheduler for Exchange has successfully scheduled a new single meeting. Default: `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `This meeting has been successfully scheduled using the aliases: {{alias}} and {{numeric_alias}}. ` `</div>`	Accepted variables: • start_time • end_time • alias • numeric_alias

Option	Description	Notes
Accept edited single meeting template	A jinja2 template that is used to produce the message sent to meeting organizers when Secure Scheduler for Exchange has successfully scheduled an edited single meeting. Default: `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `This meeting has been successfully rescheduled using the aliases: {{alias}} and {{numeric_alias}}. ` `</div>`	Accepted variables: • start_time • end_time • alias • numeric_alias
Accept new recurring meeting template	A jinja2 template that is used to produce the message sent to meeting organizers when Secure Scheduler for Exchange has successfully scheduled a new recurring meeting. Default: `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `This recurring meeting series has been successfully scheduled. ` `All meetings in this series will use the aliases: {{alias}} and {{numeric_alias}}. ` `</div>`	Accepted variables: • start_time • end_time • alias • numeric_alias
Accept edited occurrence template	A jinja2 template that is used to produce the message sent to meeting organizers when Secure Scheduler for Exchange has successfully scheduled an edited occurrence in a recurring series. Default: `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `This meeting occurrence in a recurring series has been successfully rescheduled using the aliases: {{alias}} and {{numeric_alias}}. ` `</div>`	Accepted variables: • start_time • end_time • alias • numeric_alias
Accept edited recurring meeting template	A jinja2 template that is used to produce the message sent to meeting organizers when Secure Scheduler for Exchange has successfully scheduled an edited recurring meeting. Default: `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `This recurring meeting series has been successfully rescheduled. ` `All meetings in this series will use the aliases: {{alias}} and {{numeric_alias}}. ` `</div>`	Accepted variables: • start_time • end_time • alias • numeric_alias
Reject invalid alias ID text	The text that is sent to meeting organizers when Secure Scheduler for Exchange has failed to schedule a meeting because the alias ID in the meeting email is invalid. Default: `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `This meeting request does not contain currently valid scheduling data, and therefore cannot be processed. ` `Please use the add-in to create a new meeting request, without editing any of the content that is inserted by the add-in. ` `If this issue continues, please contact your system administrator. ` `</div>`	
Reject alias conflict template	A jinja2 template that is used to produce the message sent to meeting organizers when Secure Scheduler for Exchange has failed to schedule a meeting because the alias conflicts with an existing alias. Default: `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `We are unable to schedule this meeting because the alias: {{alias}} is already  ` `in use by another Pexip Virtual Meeting Room. Please try creating a new meeting. ` `</div>`	Accepted variable: • alias

Option	Description	Notes
Reject alias deleted text	The text that is sent to meeting organizers when Secure Scheduler for Exchange has failed to schedule a meeting because the alias for this meeting has been deleted. Default: `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `We are unable to schedule this meeting because its alias has been deleted. ` `Please try creating a new meeting. ` `</div>`	
Reject recurring series in past text	The text that is sent to meeting organizers when Secure Scheduler for Exchange has failed to schedule a recurring meeting because all instances occurred in the past. Default: `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `This recurring series cannot be scheduled because all ` `occurrences happen in the past. ` `</div>`	
Reject single meeting in past text	The text that is sent to meeting organizers when Secure Scheduler for Exchange has failed to schedule a meeting because it occurred in the past. Default: `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `This meeting cannot be scheduled because it occurs in the past. ` `</div>`	
Reject general error template	A jinja2 template that is used to produce the message sent to meeting organizers when Secure Scheduler for Exchange has failed to schedule a meeting because a general error has occurred. The `{{correlation_id}}` variable is a UUID which can be used to find more information from the administrator log. Default: `<div style="font-size:11.0pt; color:#000000; font-family:Calibri,Arial,Helvetica,sans-serif;">` `We are unable to schedule this meeting. Please try creating a new meeting. ` `If this issue continues, please forward this message to your system administrator,` `including the following ID: ` `CorrelationID="{{correlation_id}}". ` `</div>`	Accepted variable: • correlation_id
Advanced options		
Disable web proxy	Select this option to bypass the web proxy (where configured for the Management Node) for outbound requests sent from this integration.	
Enable add-in debug logs	Enable this option to view debug logs within the add-in side pane in the desktop and web Outlook client. Only do this as a temporary measure if you are experiencing issues deploying the add-in. Note that these logs will appear for all users of this add-in.	
Send Kerberos authorization in every request	When ***Kerberos authentication*** is used, enable this option to send a Kerberos Authorization header in every request to the Exchange server. You may need to do this if you use a Layer 7 load balancer in your Exchange environment. If this option is disabled, Persistent-Auth is used which means a Kerberos Authorization header is sent once per connection.	
Use custom add-in sources	Enable this option to override the default locations from which the add-in JavaScript and CSS are served. This option is intended for use in fully air-gapped environments, where Outlook users will not have internet access and therefore these resources must be available locally. If you enable this option, you must host these files on your own internal server, and enter the URLs for each in the relevant fields below. For full instructions, see Configuring Exchange Scheduler in an air-gapped environment.	

Option	Description	Notes
Office.js URL	(Available if **Use custom add-in sources** is enabled) The URL of the Office.js JavaScript library on the local web server.	
Microsoft Fabric CSS URL	(Available if **Use custom add-in sources** is enabled) The URL of the Microsoft Fabric CSS on the local web server.	
Microsoft Fabric Components CSS URL	(Available if **Use custom add-in sources** is enabled) The URL of the Microsoft Fabric Components CSS on the local web server.	
Additional add-in script sources	(Available if **Use custom add-in sources** is enabled) Optionally specify additional URLs on the local web server from which to download JavaScript script files. These URLs are used in preference to any other URLs specified elsewhere for the same resource. In particular, we recommend that you specify here the URL for **MicrosoftAjax.js**. Each URL must be entered on a separate line.	
Exchange Metadata Domains and URLs		
Domain or URL	(Required for single-use VMRs and personal VMRs when using an Add-in authentication token type of ***Exchange Identity Token***. Can be left blank if using ***SSO Token*** or ***NAA (Nested App Authentication) Token***.) An FQDN or URL which can be used to access a page containing the Exchange Metadata for your Exchange deployment. This page provides the public key of the Microsoft Exchange Server Auth Certificate used by the add-in to verify user identities. If a FQDN is supplied, the default URL path **https://<FQDN>/autodiscover/metadata/json/1** is used. This URL must be reachable by the management node. If you have a hybrid Exchange and Office 365 deployment, you must include **outlook.office365.com** in the list of domains as well as the domain of one of your Exchange on-premises servers. If your Exchange deployment uses more than one domain or URL (for example, if you have an on-premises Microsoft Exchange deployment with more than one Microsoft Exchange server, or your Exchange server has more than one FQDN), **and** they use separate signing certificates, you must include all the FQDNs of all the Exchange servers in your deployment. To do this, select **Add another Exchange Metadata Domain or URL** and add the FQDN of each.	

* If you change this setting, you must re-generate and re-install the add-in XML file.

† If you change this configuration when OAuth is enabled you must also sign in to the service account again.

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 Secure Scheduler for Exchange Integration.

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

- the service account password has changed
- the service account uses multi-factor authentication (MFA) and the MFA is refreshed
- you disable and then subsequently re-enable OAuth
- you update any of the following configuration for the Secure Scheduler for 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 **System > Secure Scheduler For Exchange** and select the integration. At the bottom of the **Change VMR Exchange Integration** page, select **Sign in to service account**.

   You are taken to the **Sign in to service account** page.
3. Copy the **Sign in** link and paste it into a new browser tab.
4. Sign in as the service account.

   You are asked to permit the scheduling application registration to sign in as the service account:

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

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

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

Saving and checking configuration

When you have finished, select **Save**. You are taken back to the main **Secure Scheduler For Exchange Integration** page. The Pexip Infinity platform will attempt to contact the Microsoft Exchange deployment, and if there are any issues, it will raise an alarm on the Management Node.

Formatting the email text

All the templates and text specified in the **Email text** section can be entered as HTML. This allows you to customize the text (for example, the font, size, and color). When using HTML, you must ensure all HTML tags are closed properly, otherwise you may affect the format of any existing text in the email body.

The add-in pane headings and text can also be formatted using HTML, although some formatting may be overridden by the base HTML. We recommend that you check that any formatting applied to add-ins appears as expected in all clients used in your environment.

Working with jinja2 templates

Secure Scheduler for Exchange uses a subset of the jinja2 templating language to create the text used in emails.

This section lists the variables that can be used when creating templates, followed by some examples.

Variables

The following variables can be used when creating the jinja2 templates used for Secure Scheduler for Exchange. Note that not all variables can be used in all templates; see the descriptions of each template in Single-use VMR configuration, Personal VMR configuration and Single-use VMR email text for a list of which variables can be used in each template.

Variable	Description
Valid for Single-use VMRs	
{{addin_server_domain}} *	Inserts the FQDN configured in the Add-in server FQDN field. This FQDN is used by Pexip apps as part of the address to use when connecting to the meeting.
{{alias}}	Inserts the full alias that was generated for the VMR used for this meeting. This is in the format: **<prefix><random_number>@domain**.
{{alias_uuid}} *	Inserts the **PXPS:-** ID. For more information, see PXPS:- and TOK:- security tags. The Single-use VMR joining instructions template **must** contain this variable.
{{correlation_id}}	Used in the Reject general error template only Inserts a UUID which can be used to find more information about the error from the administrator log.
{{end_time}} **	Inserts the end time of the meeting, as per the meeting request. (This time does not include the Join after buffer.) Note that this uses the format **hh:mm on DD/MM/YYYY**, e.g. **15:30 on 31/07/2017**; this format cannot be changed.
{{numeric_alias}}	Inserts the numeric part of the alias that was generated for the VMR used for this meeting. This is in the format: **<prefix><random_number>**.
{{organizer_email}} †	Inserts the email address of the meeting organizer.
{{organizer_name}} †	Inserts the name of the meeting organizer, as it appears in the meeting invitation.
{{start_time}} **	Inserts the start time of the meeting, as per the meeting request. (This time does not include the Join before buffer.) Note that this uses the format **hh:mm on DD/MM/YYYY**, e.g. **15:30 on 31/07/2017**; this format cannot be changed.
{{subject}} †	Inserts the subject line of the meeting invitation sent by the meeting organizer. Note that this information is visible to anyone with access to the Management Node, so do not use this variable if privacy is an issue.
Valid for Personal VMRs	
{{addin_server_domain}}	Inserts the FQDN configured in the Add-in server FQDN field. This FQDN is used by Pexip apps as part of the address to use when connecting to the meeting.
{{aliases}}	Provides access to a jinja2 list object which contains all aliases configured for the personal VMR. You can then access any particular alias by using a list index. See the Personal VMR joining instructions template for example usage.
{{allow_guests}}	Returns `true` or `false`, which can then be used in an `if` statement. See the Personal VMR joining instructions template for example usage.
{{description}}	Inserts the configured **Description** of the personal VMR.
{{domain_aliases}}	Provides access to a jinja2 list object which contains all aliases configured for this personal VMR that include the **@** character followed by a domain. You can then access any one of these aliases by using a list index. See the Personal VMR joining instructions template for example usage.

Variable	Description
{{guest_pin}}	Inserts the configured **Guest PIN** of the personal VMR.
{{name}}	Inserts the configured **Name** of the personal VMR.
{{numeric_aliases}}	Provides access to a jinja2 list object which contains all aliases configured for this personal VMR that include only digits. You can then access any one of these aliases by using a list index. See the Personal VMR joining instructions template for example usage.
{{other_aliases}}	Provides access to a jinja2 list object which contains all aliases configured for this personal VMR that do not include either the **@** character followed by a domain, or only digits. You can then access any one of these aliases by using a list index. See the Personal VMR joining instructions template for example usage.
{{owners_email}}	Inserts the configured **Owner's email address** of the personal VMR.
{{pin}}	Inserts the configured **Host PIN** of the personal VMR.
{{service_type}}	Inserts a string to indicate whether the personal VMR is a "conference" (Virtual Meeting Room), "lecture" (Virtual Auditorium), or "two_stage_dialing" (Virtual Reception).

* Can only be used in joining instructions.

** Can only be used in acceptance messages.

† Can only be used in the name, description or subject fields of the VMR to be created for this meeting.

Examples

Hiding the meeting subject

By default, the meeting subject is saved to the corresponding scheduled VMR's **Name** and **Subject** field. These will be visible to administrators from the Management Node Administrator interface when Managing Exchange Scheduler conferences. The meeting name is also shown to participants accessing the scheduled conference from a Virtual Reception.

If you do not want the subject to be visible, you must change the Conference name template and Conference subject template fields on the associated Secure Scheduler for Exchange Integration.

For example, to replace the meeting subject with the organizer's name for any meetings that have the word "Confidential" in the subject, use the following values:

Conference name template:

```
{% if "confidential" in subject.lower() %}
Confidential Meeting - ({{organizer_name}})
{% else %}
{{subject}} - ({{organizer_name}})
{% endif %}
```

Conference subject template:

```
{% if "confidential" in subject.lower() %}
Confidential Meeting - ({{organizer_name}})
{% else %}
{{subject}}
{% endif %}
```

In the above example, if Alice Jones scheduled a meeting with the subject **Merger discussions - Confidential**, this would appear in the Management Node Administrator interface as **Confidential Meeting - Alice Jones**.

Deleting and replacing Secure Scheduler for Exchange Integrations

If you delete an existing Secure Scheduler for Exchange Integration and replace it with another, you must also re-generate and re-install the add-in XML file, even if the configuration of the new integration is identical to that of the old one.

Using multiple Secure Scheduler for Exchange Integrations

Different groups of users within the same Microsoft Exchange deployment

You can provide different groups of users within your Microsoft Exchange deployment with different options when using the Secure Scheduler for Exchange feature. For example, you may wish to vary the prefix used as part of the VMR alias, or use different text for the joining instructions. To do this, create multiple Secure Scheduler for Exchange Integrations that connect to the same Exchange environment. (Note however that each integration must have a separate equipment resource.)

Each integration that you create has an associated add-in which you can then make available to specific users by using Exchange PowerShell commands.

The diagram below shows a single Pexip Infinity deployment with two Secure Scheduler for Exchange Integrations to the same Microsoft Exchange deployment. Each integration uses the same EWS URL and is configured with the FQDNs of all the Exchange servers in the Exchange deployment.

The first connection provides an add-in for sales staff; the second provides an add-in for development staff. Both add-ins are uploaded to Microsoft Exchange, but each user only sees the add-in relevant to their group.

Different Microsoft Exchange deployments

If you are a service provider, you can configure one or more Secure Scheduler for Exchange Integrations for each of your customers.

The diagram below shows a single Pexip Infinity deployment with two integrations to two different Microsoft Exchange deployments. The first connection provides an add-in for everyone at Example Corp; the second provides an add-in everyone at Acme Corp.

Next step

- Making the add-in available to users within your Exchange deployment.

Add-in authentication for O365

The Outlook add-in must authenticate its requests to a Conferencing Node. This topic explains how to configure authentication for these requests in Microsoft 365 environments. Add-in authentication in these environments can use either an SSO (Single Sign-On) token (required for Microsoft Office 2021 LTSC Outlook clients), or an NAA (Nested App Authentication) token (recommended for all other Microsoft 365 environments).

If your environment includes **any** Microsoft Office 2021 LTSC Outlook clients, you must use SSO (Single Sign-On) token authentication, even if you also use other clients.

To check whether an Outlook client is part of Microsoft Office 2021 LTSC, go to **File > Office Account** and select **About Outlook**. For LTSC 2021 clients, text similar to **Microsoft® Outlook®LTSC** and with a version that includes **14332** is shown:

For Exchange on-premises environments, an Exchange Identity token is used, and no additional configuration is required for add-in authentication.

NAA (Nested App Authentication) token authentication

This process involves creating an application in Microsoft Entra with two separate registrations — a Web API registration and an App registration. We recommend that you also pre-authorize the App to access the Web API, to avoid users having to grant consents the first time they use the add-in. When you have completed this process, you must configure Pexip Infinity with details of the app registration.

We recommend that you use this authentication method if your Microsoft 365 environment does not include any Microsoft Office 2021 LTSC clients.

Creating a Web API registration

1. Go to https://entra.microsoft.com/ and log in as an administrator.
2. From the left hand panel, select **Identity** > **Applications** > **App registrations** and from the main window select **New registration**:

3. Enter details for the new registration, as follows:
   - **Name**: in our example we have used **Pexip Scheduling NAA Web API**.
   - **Supported account types**: you must select **Accounts in this organizational directory only...**
   - **Redirect URI**: leave blank.
4. Select **Register**:

5. Select **Manage** > **Owners** > **Add owners**:

Adding an owner is optional. We recommend adding yourself as an owner because it makes it easier to find the app registration in later steps. You can add additional users as owners at any point.

6. **Search** for your own user. Tick the box next to your user, and then select **Select**:

7. Select **Manage > Expose an API**. Next to **Application ID URI**, select **Add**:

8. **Save** the automatically generated **Application ID URI**. Do not edit it:

9. Select **Add a scope**:

10. Enter the following:
    - **Scope name**: enter **PexipScheduling.Access** — you must use this value, do not modify it.
    - **Who can consent?**: select ***Admins and users***.
    - **Admin consent display name**: we recommend a descriptive name such as **Access Pexip VMR Scheduling**.
    - **Admin consent description**: we recommend something descriptive such as **Allows the add-in to access Pexip VMR Scheduling for the signed in user**.
    - **User consent display name**: we recommend you use a descriptive name such as **Access Pexip VMR Scheduling**.
    - **User consent description**: we recommend something descriptive such as **Allows you to use the Pexip VMR Scheduling Outlook add-in**.
    - **State**: must be ***Enabled***.

    Select **Add scope**:

Enabling Version 2.0 access tokens

Version 1.0 access tokens are issued by default. However, Pexip requires Version 2.0. To enable Version 2.0 tokens:

1. Select **Manage > Manifest**. In the Microsoft Graph app manifest JSON, find the string `requestedAccessTokenVersion` (you can use your web browser's search function to locate it).

   By default, it is set to a value of `null`:

2. Replace `null` with the value `2`, so it looks like this:

   `"requestedAccessTokenVersion": 2,`

Select **Save**:

Creating the NAA App

1. Go to https://entra.microsoft.com/ and log in as an administrator.
2. From the left hand panel, select **Identity > Applications > App registrations** and from the main window select **New registration**.
3. Enter details for the new registration, as follows:
   - **Name**: in our example we have used **Pexip Scheduling NAA App**.
   - **Supported account types**: you must select **Accounts in this organizational directory only...**
   - **Redirect URI**: leave blank.
4. Select **Register**:

5. Select **Manage > Owners > Add owners**:

Adding an owner is optional. We recommend adding yourself as an owner because it makes it easier to find the app registration in later steps. You can add additional users as owners at any point.

6. **Search** for your own user. Tick the box next to your user, and then select **Select**:

7. From **Manage > Authentication** select **Add a platform**:

8. Select the **Single-page application** tile:

9. In the **Redirect URIs** field, enter a URI in the form: **brk-multihub://<add-in server FQDN>**, for example **brk-multihub://pexample.westeurope.cloudapp.azure.com**

   The **<add-in server FQDN>** must be the FQDN of the server on which your add-in is hosted (i.e. the Add-in server FQDN field configured on the Management Node).

   The redirect URI must be in this form; do not modify it.

10. Leave all the other options blank and unchecked, and select **Configure**:

11. Select **Manage > API permissions > Add a permission**:

12. Select the **My APIs** tab. Find and select the Web API you created in the previous step (in our example, this is named **Pexip Scheduling NAA Web API**):

13. Select **Delegated permissions** and tick the checkbox next to the **PexipScheduling.Access** permission. Select **Add permission**:

Pre-authorizing the NAA App to access the NAA Web API

This step is optional, but we recommend you do this to make the sign in flow for your users easier. Without this step, by default when users first use the add-in, they will be presented with two pop-ups requesting consent — one for the **Pexip Scheduling NAA App** (which also refers to the **Access Pexip VMR Scheduling** custom scope we created in the previous step), and one for the **Pexip Scheduling NAA Web API**.

To avoid users having to grant consent, pre-authorize the NAA app by telling it to trust the NAA Web API, as follows:

1. Go to https://entra.microsoft.com/ and log in as an administrator.
2. From the left hand panel, select **Identity > Applications > App registrations**.
3. From the **App registrations** page, select the **Owned applications** tab. From the list of Apps, find the NAA App you created earlier. (In our example, this was named **Pexip Scheduling NAA App**):

   Do not select the NAA Web API.

4. Copy the **Application (client) ID**:

5. Go back to the list of **Owned applications** and find the NAA Web API you created earlier.

   Do not select the NAA App.

   Select **Manage > Expose an API** and select **Add a client application**:

6. In the Add a client application pane, enter the following:

- **Client ID**: paste the client ID of the NAA App you have just copied.
- **Authorized scopes**: only one scope should be listed; this should end in **PexipScheduling.Access**; tick the checkbox next to this scope.

Select **Add application**:

The configuration is now complete and you can leave the Microsoft Entra admin center.

Configuring Pexip Infinity

When configuring a Secure Scheduler for Exchange Integration on Pexip Infinity, select an Add-in authentication token type of ***NAA (Nested App Authentication) Token***. Additional fields appear; enter the configuration from your new app registration in Microsoft Entra, as follows:

Field	Description	Entra configuration
Add-in auth application ID	The **Application (client) ID** which was generated when creating the App Registration in Microsoft Entra for add-in authentication.	From **Identity > Applications > App registrations**, select the **Pexip Scheduling NAA App** and copy the **Application (client) ID**.
Add-in auth NAA Web API application ID	The **Application (client) ID** for the NAA Web API which was generated when creating the App Registration in Microsoft Entra.	From **Identity > Applications > App registrations**, select the **Pexip Scheduling NAA Web API** and copy the **Application (client) ID**.
Add-in auth authority URL	The **Authority URL** copied from the App Registration created in Microsoft Entra for add-in authentication.	From **Identity > Applications > App registrations**, select the **Pexip Scheduling NAA App** and select the **Endpoints** tab. Copy the contents of the **Authority URL (Accounts in this organizational directory only)** field.

Field	Description	Entra configuration
Add-in auth OIDC Metadata URL	The **OpenID Connect metadata document** copied from the App Registration created in Microsoft Entra for add-in authentication.	From **Identity > Applications > App registrations**, select the **Pexip Scheduling NAA App** and select the **Endpoints** tab. Copy the contents of the **OpenID Connect metadata document** field.

SSO (Single Sign-On) token authentication

This process involves creating an app registration in Microsoft Entra, assigning add-in users and groups to the registration, and then configuring Pexip Infinity with details of the app registration. You must then distribute the add-in to your Outlook users.

You must use this authentication method if your Microsoft 365 environment includes **any** Microsoft Office 2021 LTSC clients.

Creating the app registration

1. Ensure you have enabled Modern Authentication. For more information, see Microsoft's article.
2. Go to https://entra.microsoft.com/ and log in as an administrator.
3. From the left hand panel, select **Identity > Applications > App registrations** and from the main window select **New registration**.
4. Enter details for the new registration, as follows:
   - **Name**: in our example we have used **Pexip Scheduling SSO App**.
   - **Supported account types**: you must select **Accounts in this organizational directory only...**
   - **Redirect URI**: leave blank.

   Select Register:

5. From the **App registrations** page, select the new application.
6. Select **Manage > Expose an API** and next to **Application ID URI** select **Add**:

7. An **Application ID URI** with a unique app UUID is generated automatically. Change the URI so that it is in the format **api://<add-in server FQDN>/<app-id>**, where
   - **<add-in server FQDN>** is the FQDN of the server on which your add-in is hosted (i.e. the Add-in server FQDN field configured on the Management Node), and
   - **<app-id>** is the UUID that was generated.

   For example, if the value that was generated was **api://ffe83558-b6bd-45eb-b3ac-7c34b08be297**, you should change this to e.g. **api://pexample.westeurope.cloudapp.azure.com/ffe83558-b6bd-45eb-b3ac-7c34b08be297**.

   Select **Save**:

8. Select **Add a scope** and enter the following:
   - **Scope name**: enter **PexipScheduling.Access** — you must use this value, do not modify it.
   - **Who can consent?**: select ***Admins and users***.
   - **Admin consent display name**: we recommend a descriptive name such as **Access Pexip VMR Scheduling**.
   - **Admin consent description**: we recommend something descriptive such as **Allows the add-in to access Pexip VMR Scheduling for the signed in user**.
   - **User consent display name**: we recommend you use a descriptive name such as **Access Pexip VMR Scheduling**.
   - **User consent description**: we recommend something descriptive such as **Allows you to use the Pexip VMR Scheduling Outlook add-in**.
   - **State**: must be ***Enabled***.

   Select **Add scope**:

9. Select **Add a client application** and enter the following:
   - **Client ID**: enter `ea5a67f6-b6f3-4338-b240-c655ddc3cc8e` — this corresponds to all Office platforms.
   - **Authorized scopes**: only one scope should be listed; this should end in **PexipScheduling.Access**; tick the checkbox next to this scope.

   Select **Add application**:

10. From **Manage > API permissions**, select and remove the Microsoft Graph **User.Read** permission (this is configured by default for all new app registrations, but is not required):

11. Select **Manage > Manifest**. In the Microsoft Graph app manifest JSON, find the string `requestedAccessTokenVersion` (you can use your web browser's search (ctrl + f) to help locate it).

By default, it is set to a value of `null`.

Replace `null` with the value `2`, i.e.:

`"requestedAccessTokenVersion": 2,`

12. Select **Save**:

Assigning users and groups to the App registration

1. Go to https://entra.microsoft.com/ and log in as an administrator.
2. From the left hand panel, select **Identity > Applications > Enterprise applications** and select the application you have just created (in our example this was called **Pexip Scheduling SSO App**):

3. Select **Manage > Users and groups** and select **Add user/group**:

4. Follow the prompts to add the users/groups that will be using the add-in.

Configuring Pexip Infinity

When configuring a Secure Scheduler for Exchange Integration on Pexip Infinity, select an Add-in authentication token type of ***SSO Token***. Additional fields appear; enter the configuration from your new app registration in Microsoft Entra as follows:

Field	Description	Entra configuration
Add-in auth application ID	The **Application (client) ID** which was generated when creating the App Registration in Microsoft Entra for add-in authentication.	From **Identity > Applications > App registrations**, select the **Pexip Scheduling SSO App** and copy the **Application (client) ID**.
Add-in auth OIDC Metadata URL	The **OpenID Connect metadata document** copied from the App Registration created in Microsoft Entra for add-in authentication.	From **Identity > Applications > App registrations**, select the **Pexip Scheduling SSO App** and select the **Endpoints** tab. Copy the contents of the **OpenID Connect metadata document** field.

Uploading the add-in manifest

If you have modified an existing Secure Scheduler for Exchange Integration to use SSO for add-in authentication, you must re-distribute the add-in XML manifest file to your Outlook users. For full instructions, see Making the Exchange add-in available to users.

Making the Exchange add-in available to users

This topic explains how to make the Pexip Secure Scheduler for Exchange add-in available to Outlook users from their Outlook web apps and desktop clients. This involves uploading an XML manifest file to your Microsoft Exchange deployment. There are two ways to do this:

- For Office 365 users, our recommended method is to use Centralized Deployment. Using this method allows you to deploy the add-in either to all users, or to selected groups of users; when group members are added or removed, the add-in is added or removed accordingly.
- You can also upload the file via Exchange Admin Center (EAC). Using this method allows you to deploy the same add-in to all users in your Exchange deployment. It is possible to restrict the add-in to a particular group of users using PowerShell, but this is a manual process that must be re-run every time group members are added or removed, and is limited to groups of 1,000 or fewer. For more information, see Restricting the Exchange add-in to specific users.

Prerequisites

Before you start you must have completed steps described in Configuring Pexip Infinity for Exchange Scheduler.

Downloading the add-in XML file

The add-in XML manifest file contains all the add-in configuration and is generated by the Pexip Infinity Management Node based on the information you provided when Configuring Pexip Infinity for Exchange Scheduler.

To download the file:

1. From the Management Node, go to **System > Secure Scheduler For Exchange**.
2. Select the Secure Scheduler for Exchange Integration you have configured.
3. From the bottom of the page, select **Download Exchange add-in manifest**.

You must now upload the add-in XML manifest file to your Microsoft Exchange deployment.

Uploading the add-in XML file to Office 365 using Centralized Deployment

Centralized Deployment is Microsoft's recommended way for an Office 365 admin to deploy Office add-ins, provided that the organization meets all requirements for using Centralized Deployment. For information about these requirements, see https://docs.microsoft.com/en-gb/office365/admin/manage/centralized-deployment-of-add-ins?view=o365-worldwide.

There are a different ways to upload an add-in XML manifest file using Centralized Deployment. The instructions below are one way; if you do not see the same options when logged in to the Microsoft Office 365 admin center then see https://docs.microsoft.com/en-us/microsoft-365/admin/manage/manage-deployment-of-add-ins for alternatives.

1. Log in to the Microsoft Office 365 admin center at https://admin.microsoft.com.
2. From the left-hand panel, select **Settings > Integrated apps**.
3. At the top of the **Integrated apps** page, select **Upload custom apps**:

4. From the **Upload Apps to deploy** page, under **Choose how to upload app**, select **Upload manifest file (.xml) from device** and select the file.

   When it has been successfully uploaded you'll see a **Manifest file validated** notification. Select **Next**:

5. From the **Add users** page, specify whether you want to give access to your **Entire organization**, or to **Specific users/groups**. Select **Next**:

6. From the **Accept permissions requests** page, review the permissions and select **Next**:

7. From the **Review and finish deployment** page, select **Finish deployment**:

8. After a short delay you see the **Deployment completed** page. Select **Done**:

9. You are returned to the **Integrated apps** page, which now lists the add-in:

It can take up to 24 hours for the add-in icon to appear to all Outlook users, and users might need to relaunch Office.

When the add-in appears, users will be able to click on it to quickly schedule meetings in Pexip Infinity VMRs.

Uploading the add-in XML file to Microsoft Exchange

To upload the add-in XML manifest file to your Microsoft Exchange deployment:

1. Log in to the Exchange Admin Center (EAC) and select **organization** > **apps** (for Exchange 2013) or **organization** > **add-ins** (for Exchange 2016).
2. Select the add icon (+) and then **Add from file**.
3. Browse to the manifest XML file and then select **Next**.

   The **Pexip Scheduling Service** add-in will appear in the list.
4. Double click the Pexip add-in to edit it.
5. Select **Make this add-in available to users in your organization**.
6. Select either **Optionally, enabled by default** or **Mandatory, always enabled**.

7. Select **Save**.

Now, when users access Outlook, the Pexip Secure Scheduler for Exchange add-in will be available for them to use to schedule meetings in Pexip Infinity VMRs. The add-in is available to all Outlook users in your deployment unless you choose to restrict it to certain users.

Testing the integration

You can test that the add-in is working as expected by logging in to an Outlook client and creating a test meeting, and then joining that meeting using the links that were generated. Note that you should ensure that the test meeting is scheduled to start within the buffer time, otherwise it won't be available to join immediately.

Troubleshooting

If you are having issues installing the add-in, see Troubleshooting Secure Scheduler for Exchange.

Restricting the Exchange add-in to specific users

You can use Windows PowerShell to enable the Secure Scheduler for Exchange scheduling add-in for all users from a specific group (rather than all users in your deployment). You should only use this method if you cannot deploy the add-in using Centralized Deployment.

Note that:

- These instructions extract a list of users in a particular group, and then make the add-in available to each of those users. If the members of a group change, you have to use the `Set-App` command to make the add-in available to the updated list of users.
- Using Windows PowerShell to enable the scheduling add-in for specific users is limited to a maximum size of 1000 users. This means the commands below only work if the AD group contains 1000 users or fewer.

Prerequisites

Before you start you must have completed the following steps:

1. Configuring Pexip Infinity for Exchange Scheduler
2. Downloading the add-in XML file

Exchange 2013 on-prem

If you have an Exchange 2013 on-premises deployment, you can enable the scheduling add-in for users from either a specific Active Directory (AD) group or a specific Exchange distribution group. For full information on the add-in PowerShell commands for Exchange 2013 used in these examples, see the following Microsoft documentation:

- Get-App: https://technet.microsoft.com/en-us/library/jj218673(v=exchg.150).aspx
- New-App: https://technet.microsoft.com/en-us/library/jj218722(v=exchg.150).aspx
- Set-App: https://technet.microsoft.com/en-us/library/jj218630(v=exchg.150).aspx
- Managing user access to add-ins for Outlook: https://technet.microsoft.com/en-us/library/jj943757(v=exchg.150).aspx

Active Directory (AD) group

To restrict the add-in to all members of an AD group (for example, a distribution group or a security group) on your AD server:

1. Get all the users in the group, and then get each user's UPN:

```
$users = Get-AdGroupMember -Identity "<Group Name>" | % { Get-AdUser $_.SamAccountName | select UserPrincipalName }
```

2. Put each UPN in an array:

```
$user_list = @()
for ($i=0; $i -lt $users.length; $i++) {
    $user_list += $users[$i].UserPrincipalName
}
```

3. Read the scheduling add-in manifest from file:

```
$Data=Get-Content -Path "<Scheduling Manifest File Path>" -Encoding Byte -ReadCount 0
```

4. Execute the `New-App` command:

```
New-App -OrganizationApp -FileData $Data -ProvidedTo SpecificUsers -UserList $user_list -DefaultStateForUser Enabled
```

Exchange distribution group

To restrict the add-in to all members of a specific Exchange distribution group:

1. Get all the users in the distribution group:

```
$user_list = Get-DistributionGroupMember "<Group Name>"
```

2. Read the scheduling add-in manifest from file:

```
$Data=Get-Content -Path "<Scheduling Manifest File Path>" -Encoding Byte -ReadCount 0
```

3. Execute the `New-App` command:

```
New-App -OrganizationApp -FileData $Data -ProvidedTo SpecificUsers -UserList $user_list -DefaultStateForUser Enabled
```

Office 365 with Microsoft Entra ID (formerly known as Azure AD)

For full information on the add-in PowerShell commands for Office 365 used in these examples, see the following Microsoft documentation:

- Get-App: https://technet.microsoft.com/en-us/library/jj218673(v=exchg.160).aspx
- New-App: https://technet.microsoft.com/en-us/library/jj218722(v=exchg.160).aspx
- Set-App: https://technet.microsoft.com/en-us/library/jj218630(v=exchg.160).aspx
- Managing user access to add-ins for Outlook: https://technet.microsoft.com/en-us/library/jj943757(v=exchg.160).aspx

Entra ID group

To restrict the add-in to all members of an Entra ID group:

1. Set up a PowerShell session with access to both Azure and Exchange. To do this, follow the instructions at https://technet.microsoft.com/en-gb/library/dn568015.aspx.
2. Get a list of your Groups and their corresponding IDs:

```
Get-MsolGroup | Format-Table -Auto ObjectID,DisplayName
```

Example output:

```
ObjectId                             DisplayName
--------                             -----------
977e8519-c71e-4670-bc7b-2aef408189da Pexip List
145f2550-82ef-419d-830b-ca018a21cea4 Pexip Group
```

3. Get the users from the targeted group:

```
$users = Get-MsolGroupMember -GroupObjectID "<ObjectID>"
```

4. Get each user's email address and put in an array:

```
$user_list = @()
for ($i=0; $i -lt $users.length; $i++) {
    $user_list += $users[$i].EmailAddress
}
```

5. Read the scheduling add-in manifest from file:

```
$Data=Get-Content -Path "<Scheduling Manifest File Path>" -Encoding Byte -ReadCount 0
```

6. Execute the `New-App` command:

```
New-App -OrganizationApp -FileData $Data -ProvidedTo SpecificUsers -UserList $user_list -DefaultStateForUser Enabled
```

Verifying the add-in is available as expected

After you have enabled the add-in for a group, you should confirm it is provided to the users you expect. To do this:

1. Get the App ID of the add-in you want to check:

```
Get-App -OrganizationApp | Format-Table -Auto DisplayName,AppId
```

Example output:

```
DisplayName              AppId
-----------              -----
Pexip Scheduling Service 79623c8b-9daf-42a0-b051-b130b8c3aef4
Bing Maps                7a774f0c-7a6f-11e0-85ad-07fb4824019b
My Templates             a216ceed-7791-4635-a752-5a4ac0a5eb93
Suggested Meetings       bc13b9d0-5ba2-446a-956b-c583bdc94d5e
```

```
Unsubscribe              d39dee0e-fdc3-4015-af8d-94d4d49294b3
Action Items             f60b8ac7-c3e3-4e42-8dad-e4e1fea59ff7
```

2. Execute the `Get-App` command:

```
Get-app -OrganizationApp -Identity "<AppId>" | Format-List DisplayName,AppId,Enabled,Default*,ProvidedTo,UserList
```

Example output:

```
DisplayName         : Pexip Scheduling Service
AppId               : 79623c8b-9daf-42a0-b051-b130b8c3aef4
Enabled             : True
DefaultStateForUser : Enabled
ProvidedTo          : SpecificUsers
UserList            : {rd.pexip.com/Users/Administrator, rd.pexip.com/Users/Toby Finch, rd.pexip.com/Users/Ben Hockley,
rd.pexip.com/Users/Bob Test, rd.pexip.com/Users/Alice Test}
```

Changing the users for an existing add-in

You can edit the set of users to which an existing add-in is available. You may wish to do this if, for example, users have been added or removed from the group, or the add-in was added using EAC and you now wish to restrict it to a particular set of users.

1. Get the App ID of the add-in:

```
Get-App -OrganizationApp | Format-Table -Auto DisplayName,AppId
```

Example output:

```
DisplayName              AppId
-----------              -----
Pexip Scheduling Service 79623c8b-9daf-42a0-b051-b130b8c3aef4
Bing Maps                7a774f0c-7a6f-11e0-85ad-07fb4824019b
My Templates             a216ceed-7791-4635-a752-5a4ac0a5eb93
Suggested Meetings       bc13b9d0-5ba2-446a-956b-c583bdc94d5e
Unsubscribe              d39dee0e-fdc3-4015-af8d-94d4d49294b3
Action Items             f60b8ac7-c3e3-4e42-8dad-e4e1fea59ff7
```

2. Get a list of users using the method appropriate for your version of Exchange:
   - Exchange 2013 AD group
   - Exchange 2013 distribution group
   - O365 Entra ID group
3. Execute the `Set-App` command:

```
Set-App -OrganizationApp -Identity "<AppId>" -ProvidedTo SpecificUsers -UserList $user_list
```

For example:

```
Set-App -OrganizationApp -Identity "79623c8b-9daf-42a0-b051-b130b8c3aef4" -ProvidedTo SpecificUsers -UserList $user_list
```

Managing Exchange Scheduler conferences

This topic describes how to view and modify any scheduled single-use conference VMRs that have been created by the Secure Scheduler for Exchange service.

Scheduled conference VMRs

When a user creates a scheduled single-use conference using Secure Scheduler for Exchange, a unique Virtual Meeting Room is created for that conference. This VMR is just like any other VMR, except it was created by the Secure Scheduler for Exchange service rather than by an administrator. For this reason, on the Pexip Infinity Administrator interface, scheduled single-use conferences have a **Service type** of **VMR**, and a **VMR origin** of **Secure Scheduler for Exchange:** followed by the name of the Secure Scheduler for Exchange Integration being used.

Viewing all single-use VMRs used for scheduled conferences

All single-use VMRs created by Secure Scheduler for Exchange are listed on the **Scheduled Conferences** page (**Services > Scheduled Conferences**).

Depending on when Secure Scheduler for Exchange has last synced with the Microsoft Exchange server, this may include meetings with a start time of up to 24 hours in the past.

Meetings that are scheduled in personal VMRs do not appear here, because they take place in existing, permanent VMRs.

Editing the VMR for a scheduled conference

To view or edit information about a particular single-use scheduled conference's Virtual Meeting Room, select it from the list.

Note that any changes made to a recurring conference's VMR apply to all instances of that meeting, as the same VMR is used for all of them.

You can add additional aliases to a scheduled conference's VMR but you should not change either of the aliases that were assigned by Secure Scheduler for Exchange.

When editing scheduled conference VMRs, all of the standard VMR settings may be modified. However, the options that specifically have scheduling-related content are:

Option	Description
Name	The name used to refer to this Virtual Meeting Room. By default this is the subject line of the meeting invitation followed by the organizer's name, but you can change the default by editing the content of the Conference name template field. If you can access this VMR via a Virtual Reception then the VMR **Name** entered here is shown to conference participants as they are transferred into the VMR (it is overlaid onto the **virtual_reception_connecting** splash screen of the theme associated with the Virtual Reception that is transferring the call).
Description	A description of the Virtual Meeting Room. By default this is ***Scheduled Conference booked by*** followed by the organizer's email address, but you can change the default by editing the content of the Conference description template field.
Owner's email address	The email address of the owner of the VMR. For scheduled conferences, this is the email address of the person who sent the meeting request.
Scheduling details	

Option	Description
Start time*	This read-only field shows the time at which the scheduled conference is available for participants to join. This is the scheduled start time offset by the time specified by the Join before buffer. For recurring meetings, this shows the start time of the current recurrence. Depending on when Secure Scheduler for Exchange has last synced with the Microsoft Exchange server, this could be up to 24 hours in the past or up to 48 hours into the future.
End time*	This read-only field shows the time at which the scheduled conference is no longer available for participants to join. This is the scheduled end time plus the time specified by the Join after buffer. For recurring meetings, this shows the end time of the current recurrence. Depending on when Secure Scheduler for Exchange has last synced with the Microsoft Exchange server, this could be up to 24 hours in the past or up to 48 hours into the future.
Next recurrence*	(For recurring meetings only) If the next recurrence is due within the next two days, this read-only field shows the time at which that conference will be available for participants to join. This is the scheduled start time offset by the time specified by the Join before buffer.
Subject	This read-only field shows the content generated by the Conference subject template field. By default, this is the subject line of the meeting invitation sent by the meeting organizer. Note that if a user updates the meeting subject and re-sends the invitation, the updated subject is not reflected in the VMR configuration unless the user also updates the meeting's start or end time.
* For recurring meetings, information is available up to 48 hours in the future. If there are no instances of the recurring meeting scheduled within this time, these fields show **There are no further scheduled recurrences of this conference in the next two days.**	
Advanced options	
VMR origin	This read-only field shows the name of the service used to create this VMR. This is **Secure Scheduler for Exchange:** followed by the name of the Secure Scheduler for Exchange Integration being used.
Aliases	
Two aliases are automatically generated for all VMRs created by the Secure Scheduler for Exchange service (for more information, see Scheduling secure meetings using Microsoft Exchange). You **must not delete** either of these aliases, but you can add additional aliases.	

Viewing all upcoming scheduled meetings

To view all instances of scheduled meetings in single-use VMRs, log into Microsoft Exchange and view the calendar for the equipment resource. To do this, you first need to assign full access to the equipment resource's mailbox to a delegate account, and then view the mailbox or calendar using the delegate account. (The delegate account could be the service account (if you are using one), or it could be, for example, an administrator's account.)

Troubleshooting

For help with troubleshooting Secure Scheduler for Exchange issues, see Troubleshooting Secure Scheduler for Exchange.

Maintenance and recovery procedures for Secure Scheduler for Exchange

The Secure Scheduler for Exchange feature includes two scripts that can be run from the Management Node to allow you to restore meetings and to delete old calendar and mail items from the equipment resource's mailbox.

Running the scripts

To run the scripts, you must log into the Management Node over SSH. To allow this, SSH access must be enabled (**Platform > Global Settings > Connectivity > Enable SSH** and **Platform > Management Node > Enable SSH**).

Log in as **admin**, using the password that was set during initial installation.

Recovering meetings

The scheduling recovery script allows you to reinstate meetings that may have been lost after Pexip Infinity was reinstalled or restored from a backup.

This tool finds **Accepted** scheduled meetings in the Exchange equipment resource's mailbox and checks whether the corresponding VMR exists on the Management Node. If not, it creates the VMR. In doing so it assigns new aliases to the reinstated VMR (as the Management Node has no record of the previous alias), and also sends out updated joining instructions to attendees.

To run this script, enter:

```
/usr/bin/schedulingrecovery
```

using the following arguments:

Argument	Description
--config	The file path of the **scheduling_config.json** file. Unless this has been changed on the advice of your Pexip authorized support representative, it will be the default so does not need to be specified. **Default:** **/etc/pexip/scheduling/scheduling_config.json**
--exchange-connector-id	The ID of the Secure Scheduler for Exchange Integration you want to process. To process all integrations in the database, do not include this argument. (To find the ID of a particular integration, go to **System > Secure Scheduler For Exchange IntegrationS** and select the integration. The ID is be the number that appears between the slashes at the end of the URL.)
--time-limit	The date and time after which meetings must start in order to be reinstated. You must use the format **YYYY-MM-DDTHH:MM:SS** in UTC. If you do not specify a time then the current time is used, meaning that only meetings happening in the future will be recovered.
--update-message	The message inserted at the top of the email update sent to attendees, which is followed by the text in the Single-use VMR joining instructions template. This argument supports HTML. **Default:** "This meeting has updated joining instructions."

Example

To restore all meetings starting after noon on 1 May 2024, enter:

```
/usr/bin/schedulingrecovery --time-limit 2024-05-01T12:00:00
```

Deleting old mailbox items

The scheduling room maintenance script deletes old inbox and calendar items from the equipment resource's mailbox. This may be useful if mailbox space needs to be freed up.

To run this script, enter:

```
/usr/bin/schedulingmaintenance
```

using the following arguments:

Argument	Description
--config	The file path of the **scheduling_config.json** file. Unless this has been changed on the advice of your Pexip authorized support representative, it will be the default so does not need to be specified. **Default: /etc/pexip/scheduling/scheduling_config.json**
--exchange-connector-id	The ID of the Secure Scheduler for Exchange Integration you want to process. To process all integrations in the database, leave this blank. (To find the ID of a particular integration, go to **System > Secure Scheduler For Exchange IntegrationS** and select the integration. The ID is the number that appears between the slashes at the end of the URL.)
--inbox-time-limit	The date and time by which inbox items must have been received, in order to be deleted. You must use the format **YYYY-MM-DDTHH:MM:SS** in UTC. If you do not specify a time then the current time is used, meaning that all inbox items will be deleted.
--calendar-time-limit	The date and time by which calendar items must finish, in order to be deleted. You must use the format **YYYY-MM-DDTHH:MM:SS** in UTC. Recurring meeting items are finished only when the last instance in the series has finished. If you do not specify a time then the current time is used, meaning that all calendar items that have already finished will be deleted.

Example

To delete all emails received before noon on 1 May 2024, and all meeting items that finished by this date, enter:

```
/usr/bin/schedulingmaintenance --inbox-time-limit 2024-05-01T12:00:00 --calendar-time-limit 2024-05-01T12:00:00
```

You can use the Pexip Secure Scheduler for Exchange add-in in Outlook to turn any of your new or existing meetings into a meeting that can be held over video or audio in a Pexip Virtual Meeting Room (VMR).

The Pexip add-in may be the only meeting option available to you within Outlook, or it may be in addition to your primary meeting solution — with the complementary Pexip meeting alternative used when the meeting's content or security classification requires it. Depending on your environment, meetings may be created in either single-use VMRs (created specifically for the meeting and only available for its duration) or your own existing personal VMR — or you may be given both options.

When creating a meeting using the Pexip add-in, information about how participants can join the VMR from a variety of video and audio clients is inserted automatically into the meeting request, **above** any text you have already entered.

Locating the add-in

The location of the add-in depends on the Outlook client that you are using — some examples are given below. These examples use the default Pexip names and icons for the add-in, but these may be different in your deployment.

Outlook desktop client (new version)

When creating a new meeting request, select the **Create a Pexip meeting** button in the toolbar:

Outlook desktop client

When creating a new meeting request, select the **Create a Pexip meeting** button in the toolbar:

Office 365 (new version)

When creating a new meeting request, select the **More** button in the toolbar and then **Pexip Scheduling Service**:

Office 365

When creating a new meeting request, select the **PEXIP Scheduling Service** button in the toolbar:

On-prem OWA

When creating a new meeting request, select the **APPS** button in the toolbar to open the **Apps For Outlook** panel, and then select **Pexip Scheduling Service**:

Creating a new video meeting

To create a new meeting to be held over video:

1. From your Outlook client, create a new meeting request in your usual way, adding the date, time, rooms and participants as required.
2. Select the Pexip Scheduling Service add-in.
3. If offered a choice, select whether you want to hold the meeting in a **Personal VMR** or a **Single-use VMR**:

Using a personal VMR

The first time you use a personal VMR, you are asked to sign in using your SSO username and password. You may need to grant the requested permissions:

After you have signed in for the first time, whenever you select a personal VMR (or if that is the only option available to you), details of how to join the meeting are inserted at the top of the meeting request (above any text you have already added). Simply add any further information to the meeting request and select **Send**.

If you wish to reschedule or cancel the meeting, you can do this in the usual way by editing the invitation and re-sending it.

Using a single-use VMR

All meetings created in a single-use VMR are assigned their own unique VMR which is created specifically for that meeting.

If you have selected a single-use VMR (or if that is the only option available to you), note that the scheduling service uses a special equipment resource (the name of which depends on your deployment), which is added to the attendee list, and details of how to join the meeting are inserted at the top of the meeting request (above any existing text you have already added).

Do not delete the scheduling service equipment resource as an attendee, or edit any of the details in the joining instructions, particularly the text between **PXPS:-** or **TOK:-** and **#**.

Simply add any further information to the meeting request and select **Send**.

You will soon receive an email from the scheduling service equipment resource accepting the meeting.

You and all other participants can join the VMR at any time between the scheduled start and end times of the meeting. For recurring meetings, you can join the same VMR each time, but it is not available for use outside of any of the scheduled meeting times.

On the rare occasion that the system is unable to assign a VMR at the point at which you select the **Pexip Scheduling Service** add-in button, the text inserted into the meeting request states that joining details will be sent in a subsequent email. You should send this request to all attendees as usual. When the system is then able to assign a VMR, a meeting update is sent on your behalf to all attendees which includes the joining instructions. You will know when this email has been sent because you will receive an updated acceptance email from the scheduling service equipment resource.

Joining details

Details of how participants can join the meeting from a range of devices are automatically added to the meeting request. Below is an example:

Please join my Pexip Meeting
in one of the following ways...

]pexip[

From a VC endpoint :
[redacted]@pexip.vc

From a web browser:
https://pexip.vc/[redacted]

From a Pexip App:
pexip://[redacted]@pexip.vc

From a telephone:
Australia: +61 2 8607 8695, then [redacted]#
Netherlands: +31 308 080 483, then [redacted]#
Norway: +47 2396 1729, then [redacted]#
United Kingdom: +44 11 8315 1787, then [redacted]#
US: +1 571 250 5783, then [redacted]#
Belgium: +32 7848 6584, then [redacted]#
Singapore: +65 3174 0758, then [redacted]#
Sweden: +46 101 992 406, then [redacted]#
Finland: +35 8753 251 459, then [redacted]#
Denmark: +45 8987 9078, then [redacted]#
Switzerland: +41 225 181 111, then [redacted]#
Brazil: +55 1131 978 035 , then [redacted]#

Changing an existing meeting into a video meeting

If you want to turn a meeting you have already scheduled into a video meeting hosted in a VMR:

1. From your Outlook client, open the meeting request.
2. Select the Pexip Scheduling Service add-in.

   Details of how to join the meeting are inserted at the top of the meeting request (above any existing information) and for single-use VMRs, the scheduling service equipment resource is added to the attendee list.

   Do not delete the scheduling service equipment resource as an attendee, or edit any of the details in the joining instructions, particularly the text between **PXPS:-** or **TOK:-** and **#**.
3. Make any further changes to the meeting and select **Send**.

The updated meeting request is sent to all attendees.

For single-use VMRs, you will soon receive an acceptance email from the equipment resource to confirm that the meeting has been successfully scheduled as a video meeting.

Editing or canceling an existing video meeting

If you want to change the details of an existing video meeting, such as changing its date or time, adding or removing participants, or canceling it completely, you can do this as you would any other Outlook meeting. However, for single-use VMRs you must not remove from the attendee list the equipment resource (the name of which will depend on your system's configuration).

If you want to change your video meeting back into a normal meeting, you must cancel the meeting and then re-create it without activating the Pexip add-in.

Example joining instructions

The joining instructions that are added by Secure Scheduler for Exchange to the body of the meeting request are generated using jinja2 templates. Below are some examples using images and colors that can be used as a basis for your own templates. These examples use the default branding path for the web app URLs, but you can use alternatives.

These examples and others using HTML formatting may not be compatible with Outlook 2013 because of the way it renders HTML. You should ensure that the joining instructions you use display appropriately on all clients in your deployment.

The `{{alias_uuid}}` variable, which inserts the **PXPS:** token, must always be included in the joining instructions.

For detailed guidance on adding content to the body of calendar invitations in Outlook, see Microsoft's documentation.

Including images in the joining instructions

If you wish to use images in the joining instructions, you must upload them to a location that will be accessible from the Outlook clients. You can then specify the image by giving the full path and filename in the HTML.

Images can be in JPG or PNG format. SVG images are not supported.

The images used in these examples are available in the **v22_scheduling.zip** file at https://dl.pexip.com/resources/icons/.

Example: dark background with images

To generate the joining instructions shown above, use the following HTML. You must substitute the following:

- **[your link]** with the FQDN of the server on which the images are stored, followed by the directory path to their location
- all **.png** image file names with the names of the images you have uploaded
- **[your number]** with the telephone number used by audio participants.
  - This can be a direct dial number, or, if the telephone number directs callers to a Virtual Reception, you can include commas (,) after the direct dial number, followed by the conference ID. This means that users of devices that support DTMF can click on the link and their device will automatically enter the conference ID after calling into the g Virtual Reception.

```
<br>
<style>
    .pexip-cell {
        padding: 20px;
    }
    .main-title {
```

```
        font-size: 22px;
        color: #ffffff;
    }
    .pexip-heading {
        font-weight: bold;
        font-size: 14px;
        color: #ffffff;
    }
    .pexip-info {
        font-size: 12px;
        color: #02a8ae;
    }
    a:link {
        color: #02a8ae;
    }
    .pexip-pxps {
        font-size: 12px;
        color: #808080;
    }
    .left-column {
        padding: 20px;
        text-align: right;
    }
    .center-column {
        width: 100%;
        padding: 20px 20px 20px 0px;
    }
</style>
<table style="width: 100%; border-collapse: collapse; background: #0a2136; border-radius: 5px; font-family: IBMPlexSans, Calibri, sans-serif,
serif;">
    <tbody>
        <tr>
            <td colspan="2" class="pexip-cell" style="vertical-align: top;">
                <p>
                    <span class="main-title">
                        Please join my Pexip virtual meeting<br>
                        room in one of the following ways...
                    </span>
                </p>
            </td>
            <td class="pexip-cell" style="text-align: right; vertical-align: top;">
                <img src="https://[your link]/Pexip_logo_white.png" width="111" height="30" alt="Pexip">
            </td>
        </tr>
        <tr>
            <td class="left-column">
                <img src="https://[your link]/Endpoint_icon_white.png" width="42" height="57" alt="Endpoint">
            </td>
            <td class="center-column">
                <p>
                    <span class="pexip-heading">
                        From a VC endpoint:
                    </span>
                    <br>
                    <a href="sip:{{alias}}">
                        <span class="pexip-info">
                            {{alias}}
                        </span>
                    </a>
                </p>
            </td>
            <td class="pexip-cell"/>
        </tr>
        <tr>
            <td class="left-column">
                <img src="https://[your link]/Web_icon_white.png" width="44" height="46" alt="Web">
            </td>
            <td class="center-column">
                <p>
                    <span class="pexip-heading">
                        From a web browser:
                    </span>
                    <br>
                    <a href="https://{{addin_server_domain}}/webapp/m/?conference={{alias}}">
                        <span class="pexip-info">
```

```
                        https://{{addin_server_domain}}/webapp/m/?conference={{alias}}
                    </span>
                </a>
            </p>
        </td>
        <td class="pexip-cell"/>
    </tr>
    <tr>
        <td class="left-column">
            <img src="https://[your link]/App_icon_white.png" width="44" height="44" alt="App">
        </td>
        <td class="center-column">
            <p>
                <span class="pexip-heading">
                    From a Pexip App:
                </span>
                <br>
                <a href="pexip://{{alias}}">
                    <span class="pexip-info">
                        pexip://{{alias}}
                    </span>
                </a>
            </p>
        </td>
        <td class="pexip-cell"/>
    </tr>
    <tr>
        <td class="left-column">
            <img src="https://[your link]/Dial_icon_white.png" width="32" height="35" alt="Phone">
        </td>
        <td class="center-column">
            <p>
                <span class="pexip-heading">
                    From a telephone:
                </span>
                <br>
                <a href="tel:[your number]">
                    <span class="pexip-info">
                        [your number], then {{ numeric_alias|pex_regex_replace('(\d{3})(\d{2})(\d{3})', '\\1 \\2 \\3') }} #
                    </span>
                </a>
            </p>
        </td>
        <td class="pexip-cell"/>
    </tr>
    <tr>
        <td class="left-column"/>
        <td colspan="2" class="pexip-cell" style="text-align: right;">
            <p>
                <span class="pexip-pxps">
                    {{alias_uuid}}
                </span>
            </p>
        </td>
    </tr>
  </tbody>
</table>
```

Example: light background with images

To generate the joining instructions shown above, use the following HTML. You must substitute the following:

- **[your link]** with the FQDN of the server on which the images are stored, followed by the directory path to their location
- all **.png** image file names with the names of the images you have uploaded
- **[your number]** with the telephone number used by audio participants.

  This can be a direct dial number, or, if the telephone number directs callers to a Virtual Reception, you can include commas (,) after the direct dial number, followed by the conference ID. This means that users of devices that support DTMF can click on the link and their device will automatically enter the conference ID after calling into the g Virtual Reception.

```
<br>
<style>
    .pexip-cell {
        padding: 20px;
    }
    .main-title {
        font-size: 22px;
        color: #4a4a4a;
    }
    .pexip-heading {
        font-weight: bold;
        font-size: 14px;
        color: #4a4a4a;
    }
    .pexip-info {
        font-size: 12px;
        color: #02a8ae;
    }
    a:link {
        color: #02a8ae;
    }
    .pexip-pxps {
        font-size: 12px;
        color: #808080;
    }
    .left-column {
        padding: 20px;
        text-align: right;
    }
    .center-column {
        width: 100%;
        padding: 20px 20px 20px 0px;
    }
</style>
<table style="width: 100%; border-collapse: collapse; border: 1px solid #0a2136; font-family: IBMPlexSans, Calibri, sans-serif, serif;">
    <tbody>
```

```
<tr>
    <td colspan="2" class="pexip-cell" style="vertical-align: top;">
        <p>
            <span class="main-title">
                Please join my Pexip virtual meeting<br>
                room in one of the following ways...
            </span>
        </p>
    </td>
    <td class="pexip-cell" style="text-align: right; vertical-align: top;">
        <img src="https://[your link]/Pexip_logo_blue.png" width="112" height="30" alt="Pexip">
    </td>
</tr>
<tr>
    <td class="left-column">
        <img src="https://[your link]/Endpoint_icon_dark.png" width="42" height="57" alt="Endpoint">
    </td>
    <td class="center-column">
        <p>
            <span class="pexip-heading">
                From a VC endpoint:
            </span>
            <br>
            <a href="sip:{{alias}}">
                <span class="pexip-info">
                    {{alias}}
                </span>
            </a>
        </p>
    </td>
    <td class="pexip-cell"/>
</tr>
<tr>
    <td class="left-column">
        <img src="https://[your link]/Web_icon_dark.png" width="42" height="44" alt="Web">
    </td>
    <td class="center-column">
        <p>
            <span class="pexip-heading">
                From a web browser:
            </span>
            <br>
            <a href="https://{{addin_server_domain}}/webapp/m/?conference={{alias}}">
                <span class="pexip-info">
                    https://{{addin_server_domain}}/webapp/m/?conference={{alias}}
                </span>
            </a>
        </p>
    </td>
    <td class="pexip-cell"/>
</tr>
<tr>
    <td class="left-column">
        <img src="https://[your link]/App_icon_dark.png" width="42" height="44" alt="App">
    </td>
    <td class="center-column">
        <p>
            <span class="pexip-heading">
                From a Pexip App:
            </span>
            <br>
            <a href="pexip://{{alias}}">
                <span class="pexip-info">
                    pexip://{{alias}}
                </span>
            </a>
        </p>
    </td>
    <td class="pexip-cell"/>
</tr>
<tr>
    <td class="left-column">
        <img src="https://[your link]/Dial_icon_dark.png" width="31" height="33" alt="Phone">
    </td>
    <td class="center-column">
```

```
                <p>
                    <span class="pexip-heading">
                        From a telephone:
                    </span>
                    <br>
                    <a href="tel:[your number]">
                        <span class="pexip-info">
                            [your number], then {{ numeric_alias|pex_regex_replace('(\d{3})(\d{2})(\d{3})', '\\1 \\2 \\3') }} #
                        </span>
                    </a>
                </p>
            </td>
            <td class="pexip-cell"/>
        </tr>
        <tr>
            <td class="left-column"/>
            <td colspan="2" class="pexip-cell" style="text-align: right;">
                <p>
                    <span class="pexip-pxps">
                        {{alias_uuid}}
                    </span>
                </p>
            </td>
        </tr>
    </tbody>
</table>
```

Troubleshooting Secure Scheduler for Exchange

This section provides guidance on the troubleshooting of issues with Secure Scheduler for Exchange.

For guidance on the troubleshooting of general issues, see Troubleshooting the Pexip Infinity platform.

When troubleshooting Secure Scheduler for Exchange issues via the Administrator log and the Support log, search for `User="schedulingservice"`. This is an internal system user responsible for all configuration changes relating to the scheduling service (such as creating and deleting scheduled VMRs).

Symptom	Possible cause	Resolution
General issues		
Users can use the add-in to generate meeting aliases, but when they send the meeting request they get an email response from the equipment resource with the message configured in the Reject invalid alias ID text field. By default this begins with ***This meeting request does not contain valid data, and cannot be processed.***	Secure Scheduler for Exchange is enabled on more than one Management Node and both have Secure Scheduler for Exchange Integrations that are configured with the same equipment resource. This may occur if you have separate test and development environments, each with their own Management Node; you will be prevented from using the same equipment resource for different integrations on the same Management Node.	Ensure that each equipment resource is being used by no more than one integration and no more than one Management Node.
	A user has changed, removed, or added an invalid meeting ID before the meeting has been accepted by the equipment resource. This ID appears in the format **PXPS:-<xxx>#** or **TOK:-<xxx>#**.	Remind users that they should not edit or delete this text, and use the add-in to generate it.
	A user has added the equipment resource as an attendee, but without activating the add-in.	Remind users that they must always use the add-in to schedule meetings.
	The Management Node was offline and did not come back online before the security token expired (usually around 8 hours).	Create a new meeting request.
	A user has tried to reschedule a meeting that occurred in the past, but the meeting no longer exists in the database because the scheduling service regularly deletes expired meetings.	Users should create new meetings rather than attempt to reschedule meetings that have occurred in the past.
There is a long delay between a meeting being scheduled and a response being received from the meeting resource.	In deployments where Secure Scheduler for Exchange is receiving consistently high levels of usage (i.e. more than 30 meeting requests per minute), the processing of emails by the equipment resource may be delayed.	The workaround is to: 1. Split users into several smaller groups. 2. Create an individual Exchange Integration (with a different equipment resource mailbox) for each group of users. 3. For each Exchange Integration, deploy the add-in manifest file to that specific group of users only.
Users are able to use the add-in to generate meeting aliases, but when they send the meeting request no response is received from the equipment resource.	The scheduling license has expired.	Renew the scheduling license or delete the add-in from the Microsoft Exchange server.
	The equipment resource has been deleted from the meeting request.	Remind users that they must not delete the equipment resource as an attendee.

Symptom	Possible cause	Resolution
When a user attempts to activate the add-in, they get message stating that there was an **Invalid Token** or **Token error**.	The FQDN in the security token does not match the FQDN of any of the configured Microsoft Exchange servers. This occurs when not all Microsoft Exchange server FQDNs have been configured under Exchange Metadata Domains and URLs.	Ensure that the FQDN of all Microsoft Exchange servers in your deployment are listed under Exchange Metadata Domains and URLs. This is required even if you are using just one Microsoft Exchange server.
	This was due to the recent deprecation of Microsoft's **getUserIdentityTokenAsync** API. (For more information, see https://learn.microsoft.com/en-us/office/dev/add-ins/outlook/faq-nested-app-auth-outlook-legacy-tokens)	• Upgrade to v37.1, which includes options to configure authentication for the Outlook add-in. For full instructions, see Add-in authentication for O365. • As a temporary fix, use PowerShell commands to temporarily re-enable the API. For full instructions, see https://learn.microsoft.com/en-us/office/dev/add-ins/outlook/turn-exchange-tokens-on-off.
Outlook for Mac users who activate the add-in before they have added any participants to the invitation get the configured Error inserting single-use VMR message. By default this is ***There was a problem adding the joining instructions. Please try again***.	This is a known Microsoft issue and has been resolved in Outlook for Mac version 16.27 (19062615).	As a workaround, until users have upgraded their clients to Outlook for Mac version 16.27 (19062615) and later, users should add at least one participant to the meeting invitation before they activate the add-in.
Users get the configured Error inserting single-use VMR message. By default this is ***There was a problem adding the joining instructions. Please try again*** along with ***The token is not valid until <time>*** and/or ***The token is not valid after <time>***.	(Mac users) There is a known issue with Outlook for Mac where time does not synchronize properly after a user's laptop has been "asleep" for a period of time.	Restart the Outlook for Mac client.
	The security token has expired, or is not yet valid.	• Restart the Outlook client. • Check that both the Management Node and the Exchange server are synchronizing their time with an NTP server, and that both systems are showing the same time.
Users who attempt to connect to a VMR get the message "Cannot connect to <alias>. Check this address and try again" or "Invalid conference <alias>".	The VMR was created using Secure Scheduler for Exchange but the user is attempting to connect to it outside of the allowed time (which is the scheduled meeting time plus the configured Join before buffer and Join after buffer).	Remind users that VMRs created using the Secure Scheduler for Exchange feature are only available for use during the scheduled meeting time.
The equipment resource has accepted the meeting but the scheduled conference is not listed on the Management Node (**Services > Scheduled Conferences**).	The equipment resource has automatically accepted the meeting invitation. This is the default behavior for an equipment resource, but in order for Secure Scheduler for Exchange to process meeting invitations, automatic processing must be disabled.	Disable automatic processing for the equipment resource - see Configuring Exchange on-premises for scheduling.

Symptom	Possible cause	Resolution
When Microsoft's OWA is used to connect to an Office 365 account and an add-in is activated, the absence of a horizontal scroll bar in the add-in panel may mean that not all text is visible.	This is a known Microsoft issue.	To view all text, Secure Scheduler for Exchange users should either widen the window or pop-out the meeting request.
Users who attempt to join a scheduled meeting by clicking the link under "From a Pexip App" from within the meeting invite get a Microsoft Outlook Security Notice warning that "This location may be unsafe".	Links to meetings that are to be opened using the Connect desktop app begin with `pexip:` However, Outlook does not recognize the `pexip:` protocol, so it will bring up a warning when a user attempts to open such links.	You can modify the registry to disable warnings for specific protocols. Consult Microsoft support documentation for the appropriate way to do this for your version of Outlook.
When configuring Kerberos authentication, an alarm "MS Exchange Connection Failure" is shown, and "Cannot find KDC for requested realm" appears in the error log.	On Pexip Infinity, the domain in the configured Kerberos Service account username and / or Kerberos realm are not in upper case.	Change the domain to use upper case.

Symptom	Possible cause	Resolution
Add-in issues		
The add-in button appears but is grayed out.	Outlook is not connected to the Exchange Server or is running in Offline or Cached mode.	Ensure that Outlook is connected to the Exchange server and can send and receive email.
	(Desktop client users) The user is attempting to use the add-in in a calendar to which they have delegate access.	Enable delegate access - for information, see Support for delegate access to calendars Alternatively, users can use the OWA client.
The add-in icon has been changed but the old icon is still showing in Outlook clients, even after deleting the Outlook profile.	The image icons are being cached on the client device.	1. Close the Outlook client. 2. Delete the "Wef" folder which stores the add-in manifests, from the following location: For Windows users: **..\Users\<user>\AppData\Local\Microsoft\Office\16.0\Wef** For Mac users: **../Users/<user>/Library/Containers/com.microsoft.Outlook/Data/Library/Application Support/Microsoft/Office/16.0/Wef** 3. Clear the cache in Internet Explorer, ensuring you delete **temporary internet files** and **website files**.

The add-in appears for OWA users, but not for desktop client users	Desktop client users' registry settings may be preventing the download and display of the add-in.	Ensure that the users' registry settings include the following settings (for more information, see Microsoft's documentation):
		`Windows Registry Editor Version 5.00`
		`[HKEY_CURRENT_USER\Software\Policies\Microsoft\office\16.0\common\privacy]`
		`"disconnectedstate"=dword:00000001`
		`"usercontentdisabled"=dword:00000001`
		`"downloadcontentdisabled"=dword:00000001`
		`"controllerconnectedservicesenabled"=dword:00000001`

Symptom	Possible cause	Resolution
The add-in button does not appear, does not show the correct image and/or there is an error loading the add-in.	The Conferencing Node or reverse proxy specified by the **Add-in server FQDN** does not have a valid, trusted certificate. To check this, enter the FQDN in a web browser. If the certificate is not valid, a message to that effect will appear.	Ensure that the Conferencing Node or reverse proxy has a trusted, valid certificate.
	The user's device cannot connect to the Conferencing Node or reverse proxy specified by the **Add-in server FQDN**. To check this, from a web browser on the same device, attempt to connect to: **https://<Add-in server FQDN>/api/client/v2/msexchange_schedulers/<connector_id>/images/addin_icon_80.png** where **<connector_id>** is the ID of the integration on the Management Node. To find the ID, select the integration; the ID is the number that appears between the slashes at the end of the URL. For example, if the URL is **https://testmgr.example.com/admin/platform/msexchangeconnector/1/** the ID is **1**.	Resolve the connection issue between the user's device and the Conferencing Node or reverse proxy.

Symptom	Possible cause	Resolution
	The add-in XML manifest file being used by the Outlook client is out of date. This may occur if an Administrator has changed the server FQDN or has re-added an integration without then downloading a new XML file and uploading it to Microsoft Exchange.	Ensure that after making any changes to the configuration of an integration on the Management Node, you download a new add-in XML manifest file and then upload the file to Microsoft Exchange.
	(Desktop client users) The add-in XML manifest file has not been received by the Outlook client from the Microsoft Exchange server. To confirm that it has been received: 1. From the **Exchange admin center**, select the Pexip add-in and note the version number: 2. On the user's device, go to the folder for the Outlook manifest files, such as: **..\AppData\Local\Microsoft\Office\16.0\Wef\{uuid}\random_string\Manifests** and check whether there is a file that ends with the same number: 	Resolve the connection issue between the Outlook client and the Microsoft Exchange server.
	The add-in was specified as ***Optional...***, and the user has disabled it. To check this for Office 365 users: 1. Log into OWA as the user, and then select **Settings**. 2. Search for **Manage Integrations**. 3. Select **My Add-ins** to view the Add-ins page. The Pexip Scheduling Service should be listed here, and shown as ***On***.	• Re-enable the add-in. • Specify the add-in as ***Mandatory***.
	(Desktop client users) The Outlook email account being used was added manually (rather than via the auto account setup option).	Delete the Outlook email profile and re-add it automatically. Note that DNS must be set up properly to allow the auto-discovery service to work.
	(Desktop client users) The "apps for Office" button is not enabled.	Ensure that the "apps for Office" button is enabled. Refer to Microsoft's Outlook add-in troubleshooting guide.
	(Desktop client users) The version of Outlook is out of date.	Update Outlook to use the latest available version. From Outlook 2016 this can be done via **File > Office Account > Office Updates**.

Symptom	Possible cause	Resolution
The pop-up window that opens to allow users to authenticate when using personal VMRs does not automatically close after the user has signed in, and instead appears blank.	(OWA users connecting to Exchange 2016 or 2019) This is a known issue with OWA (for more information, see https://github.com/OfficeDev/office-js/issues/1286).	Users can manually close the blank pop-up window and click the sign-in button again. They will not be asked to sign-in again (if they signed in successfully the first time); instead, they are presented with a list of VMRs to select from, as expected.
OWA users in on-premises Exchange deployments are presented with a message saying that the add-in has been disabled.	Following a recent update to Exchange on-premises, when activating add-ins OWA users are now required to agree that they trust the hosted domain (for more information, see https://github.com/OfficeDev/office-js/issues/1441).	Ensure that your on-premises Exchange servers are updated to the latest Cumulative Update (CU) and run Windows update.
	Other issues not listed above.	Refer to Microsoft's Outlook add-in troubleshooting guide.