



Pexip Connect Apps

Guide for Administrators

Software Version 34

Document Version 34.a

March 2024

]pexip[

Contents

Introduction	5
Connect apps	5
Which Connect apps should I use in my deployment?	6
Connect app guides for end users	6
Making calls from Connect apps	6
Receiving calls to Connect apps	7
Branding the Connect apps	7
Enabling and disabling use of Connect apps	7
Comparison of Connect app and other video endpoints	8
Installing and using Connect apps	9
About the Connect web app	9
Connect web app versions	9
Support for first-time and infrequent users	9
Language support	9
Accessing a conference or making a call	10
Hardware requirements	11
About the Connect desktop app	11
Hardware requirements	11
Installing the Connect desktop app	12
Registering the Connect desktop app	13
Accessing a conference or making a call	13
About the Connect mobile apps	13
Prerequisites	13
Protocols	13
Installing the Connect mobile app for Android	14
Installing the Connect mobile app for iOS	14
Accessing a conference or making a call	14
Registering and provisioning the Connect desktop app	14
Connect app authentication options	14
Setting up appropriate DNS records	15
Provisioning the Connect desktop app with registration and/or branding details	15
Using Connect apps to share content	21
Sharing your screen	21
Sharing images and PDFs	24
Using the Connect app for presentation, chat and conference control only	25
Locking a conference and allowing participants to join a locked conference	26
Locking using the Administrator interface	26
Locking using Connect apps	27
Locking using DTMF	27
Allowing waiting participants to join a locked conference	27
Rejecting a request to join a locked conference	28

Administering Connect app	29
Customizing and branding the Connect apps	29
Creating a branding package using the portals	29
Downloading an existing package	32
Uploading the branding package	33
Updating an existing package	33
Removing a package (reverting to default branding)	34
Applying branding to the desktop clients	34
Creating path-based web app branding	36
Prerequisites	36
Creating a web app path	36
Default paths and default branding	37
Advanced Connect app branding and customization	39
Hosting on the Conferencing Nodes	39
Manually customizing Connect Webapp3	39
Manually customizing Connect Webapp2	51
More information	59
Hosting the web app externally	59
Hosting options overview	60
Copying over the Connect web app	60
Uploading branding files	61
Maintaining customizations when upgrading Pexip Infinity	61
Obtaining diagnostic information from Connect apps	62
Creating preconfigured links to launch conferences via Connect apps	62
Security considerations	62
Links to the web app	62
Links to the desktop and mobile clients	65
Links to a join instructions page	66
Links to the legacy Connect apps	68
Setting up DNS records and firewalls for Connect app connectivity	68
DNS records	68
Firewall configuration	69
Using the Connect app from outside your network	69
Further information and connectivity examples	69
Deploying the Connect desktop app under Citrix	72
Architecture overview	72
Supported Citrix versions	73
Prerequisites	73
Port requirements	73
Installing the Connect desktop app	74
Running as a Citrix Published Desktop	74
Running as a Citrix Published App	74
About Connect web app versions	74
Setting which version is offered by default	75
Troubleshooting Connect app error messages	75
Connect app release notes	79

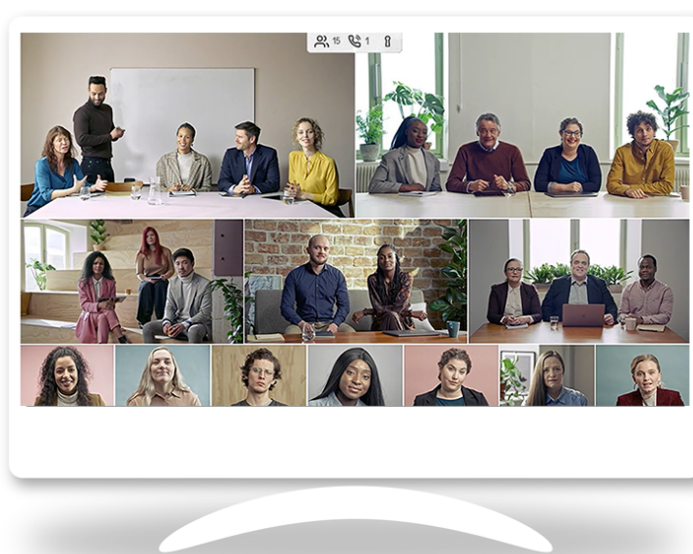
Connect web app release notes	79
What's new?	79
Fixed issues	80
Known limitations	81
Connect desktop app release notes	81
What's new?	81
Fixed issues	82
Known limitations	84
Connect mobile app release notes	84
What's new?	85
Fixed issues	85
Known limitations	85

Introduction

Pexip Infinity is a self-hosted, virtualized and distributed multipoint video conferencing platform. It can be deployed in an organization's own datacenter, or in a private or public cloud such as Microsoft Azure, Amazon Web Services (AWS), Google Cloud Platform (GCP) or Oracle Cloud Infrastructure, as well as in any hybrid combination. It enables scaling of video, voice and data collaboration across organizations, regardless of technology platforms, seamlessly allowing everyone to engage in high definition video, web, and audio conferencing.

It provides any number of users with their own personal Virtual Meeting Rooms (VMRs), which they can use to hold conferences, share presentations, and chat. Participants can join over audio or video from any location using the endpoint or client of their choice, including:

- Professional video conferencing room systems (Microsoft Teams Rooms, SIP and H.323 devices)
- Desktop/mobile (with the Pexip Connect app suite of clients)
- Web browsers (WebRTC - no downloads required)
- Traditional audio conferencing (PSTN dialing)



Pexip VMRs maintain the same customized address and are always available for spontaneous 1-to-1 or group meetings.

VMRs can also be accessed through a Virtual Reception IVR service, which allows all participants to dial a single number to access Pexip Infinity, and then use the dial tones on their endpoint or phone to select the conference they want to join.

There is a wide choice of layouts. You can select from a range of classic layouts, Pexip's AI-driven Adaptive Composition layout featuring real-time automatic face detection and framing, or you can even design your own custom layouts.

The platform also includes the Infinity Gateway service, which allows end users to place calls to other endpoints that use different protocols and media formats, or to seamlessly connect into an externally-hosted conference, such as a Microsoft Teams or Skype for Business meeting, or Google Meet.

Pexip's Media Playback Service allows you to play prerecorded video content (such as adverts and informational videos) to consumers. When the media finishes playing, the user can be transferred to another service, such as a VMR conference, or they can be disconnected.

It automatically transcodes all the popular video and audio codecs and supports standard protocols including SIP, H.323, and WebRTC. It supports all standards-based devices including those from Cisco, Poly, Lifesize, Sony, Radvision, Yealink, and Avaya. It also supports software clients such as Microsoft Teams Rooms, Skype for Business and Surface Hub.

Connect apps

The Connect apps allow users to join conferences (Virtual Meeting Rooms, Virtual Auditoriums and so on) within the Pexip Infinity deployment.

In addition to sharing audio and video, Connect app users can also control the conference, view presentations, share content, and exchange chat messages with other conference participants. The Connect app can also be used in conjunction with the Infinity Gateway to make person-to-person calls, or join conferences hosted on other platforms, such as Skype for Business meetings.

All Connect apps can make calls to Pexip Infinity services. The Connect desktop app can also register to Pexip Infinity in order to receive calls and use directory services.

Connect apps are available for almost any device:

- The [Connect web app](#) is included as part of all Pexip Infinity deployments. It is used to access Pexip Infinity services from all of the major web browsers.
- The [Connect desktop app](#) is an installable client, supported on Windows, OS X, and Linux.
- The [Connect mobile apps](#) are available for Android and iOS devices.

All Connect apps are available for free with the Pexip Infinity platform (although, as with any other endpoint, you must still have a license with sufficient call capacity before you can place calls).

Which Connect apps should I use in my deployment?

The Connect apps all offer identical conference join and control features, and have the same high-quality video experience. You can use a combination of some or all Connect apps within your deployment, depending on your requirements. In general, we recommend the following:

- Users connecting from outside your organization and who do not have their own video device should generally use the Connect web app to access VMRs. This means that they won't need to download or install anything in order to access meetings, but will still have the same high-quality user experience and functionality of participants using the Connect desktop app.
You'll need to make sure that at least one Conferencing Node is accessible externally, and you'll also need to [set up appropriate DNS records](#) for connections from both inside and outside your network.
- Users connecting from inside your organization should also use the Connect web app, unless you want them to be able to register to receive incoming calls — in which case they need to use the Connect desktop app.
- The Connect desktop app should be used if you want to take advantage of the additional registration (to receive incoming calls) and internal directory service features. Administrators can also set up [Call Routing Rules](#) that apply to registered devices only, meaning that you can permit registered Connect desktop app users to make calls that Connect web app users cannot.
If you are deploying the Connect desktop app in your environment, we recommend that you make use of [provisioning](#), and you'll also need to [set up appropriate DNS records](#).
- The Connect mobile app is aimed at users who want to be able to control a conference and view presentations while in a meeting using a video device that does not support those features — for example, a video conferencing endpoint in a meeting room.
- The Connect mobile app can also be used to join a meetings as an audio or video participants, but because of the nature of mobile devices this may result in intensive battery use.

Connect app guides for end users

This guide covers topics that are only relevant to an administrator.

We publish a series of quick guides aimed at end users of the Connect desktop app, the Connect web app, and the Connect mobile app. These guides are available in PDF format from https://docs.pexip.com/admin/download_pdf.htm#enduser.

Making calls from Connect apps

For a Connect app to make a call, it must be able to connect to a Conferencing Node that can route that call on its behalf.

The **Connect web app** connects directly to a Conferencing Node or Reverse Proxy (via the host's FQDN or IP address). When a call is placed from the Connect web app, it is treated as an **incoming call request** by the Conferencing Node, and routed accordingly. For more information, see [Service precedence](#). All other Connect apps typically use DNS SRV records to find a Conferencing Node to connect to.

You must ensure that your deployment has appropriate internal and external DNS configured to allow Connect apps located inside and outside your internal network to resolve the Conferencing Node address successfully. The actual address Connect apps use when attempting to locate a host Conferencing Node depends on the domain being called and the app's own configuration. For more information, see [Setting up DNS records and firewalls for Connect app connectivity](#).

Receiving calls to Connect apps

For a Connect app to receive a call, it must register with a Conferencing Node. The client's **Registration Host** setting specifies the domain, FQDN or IP address of the Conferencing Node that it should register to; therefore, you must ensure that the address used is reachable from the client from the internal or external network as appropriate, and that any FQDNs can be resolved via DNS lookups. For more information, see [Registering and provisioning the Connect desktop app](#).

Currently, only the Connect desktop app can register to a Conferencing Node.

Branding the Connect apps

The branding and styling of the Connect web apps and Connect desktop app can be customized. This changes the look and feel of the Connect app regardless of which service is being accessed. See [Customizing and branding the Connect apps](#) for more information.

Enabling and disabling use of Connect apps

Access to conferences from all Connect apps is enabled by default. If you do not want users to access conferences within your deployment from Connect apps, you can disable this functionality.

To disable or re-enable this functionality:


1. Go to **Platform > Global Settings**.
2. From within the **Connectivity** section:
 - a. Deselect or select **Enable support for Pexip Infinity Connect and Mobile App**. This controls access from all Connect apps and third-party clients using the client APIs.
 - b. When **Enable support for Pexip Infinity Connect and Mobile App** is selected, you must also ensure that **Enable WebRTC** is selected.

When access is disabled, users attempting to use Connect apps to access a conference or make a call are presented with the message **Call Failed: Disabled** (you can customize the Connect apps to change the wording of this message if required).

Comparison of Connect app and other video endpoints

The Connect apps are developed directly by Pexip and use Pexip's client APIs to integrate with the Pexip Infinity platform. This means that there are some differences in the experience of joining and participating in a Pexip Infinity conference via a Connect app, when compared with Skype for Business clients and other types of software and hardware endpoints.

The table below summarizes these behavioral differences.

Feature	Connect app	Skype for Business and other video clients
Joining a Host+Guest conference that has a Host PIN but no Guest PIN *	<p>Whether or not a Host has already joined, participants have the option to enter the Host PIN (to join as a Host), or simply select "Join" (to join as a Guest).</p> <p>If they choose to join as a Guest:</p> <ul style="list-style-type: none"> If a Host has not yet joined, they are taken to the "Waiting for Host" screen. <p>While waiting for a Host to join, a keypad option is available on their toolbar which they can use to enter the Host PIN and join as a Host.</p> <ul style="list-style-type: none"> if a Host has already joined, they are taken straight into the conference. 	<ul style="list-style-type: none"> If a Host has not already joined, participants are taken to the "Waiting for Host" screen, where they have the opportunity to enter the Host PIN. If a Host has already joined, participants automatically join as a Guest, unless they have included the Host PIN as part of the dial string.
Joining a Host+Guest conference that has a Host PIN and Guest PIN *	All clients have the same behavior: participants are asked to enter the conference PIN and if they enter the Host PIN, they join the conference. If they enter the Guest PIN and a Host has already joined then they are taken straight into the conference, otherwise they are taken to the "Waiting for Host" screen.	
Conference PINs with a trailing #	When entering PINs, any trailing # is optional.	Participants hear the "please enter the # key" prompts, and must enter the # after the PIN.
Joining a VMR via a Virtual Reception	Participants must dial into the Virtual Reception first, and then at the prompt enter the numeric alias of the target Virtual Meeting Room.	<p>Participants using other clients can join a VMR via a Virtual Reception in a single step. They do this by dialing <code><reception_alias>*<destination_alias>@<domain></code>.</p> <p>H.323 devices can also use the dial format <code><reception_alias>#<destination_alias>@<domain></code>.</p>
Viewing roster	Participants can view the roster.	The roster is not available.
Appearing in the Connect app roster	Other Connect app participants appear in the roster only after they have successfully joined the conference.	<p>Participants using other clients appear in the roster while they are waiting to join the conference, for example while they are being held at the PIN entry screen or waiting for a Host* to join. At this point, they do not have a role assigned.</p> <p> A Host using a Connect app (including Hosts who have joined in presentation and control-only mode) can let these participants into the conference without them having to enter a PIN.</p>
Conference control	Host participants can control the conference (add, mute, and disconnect participants; change a participant's role; lock and unlock the conference).	Participants do not have access to conference control, apart from a limited set of controls available to endpoints that support DTMF.
Chat	Participants using the Connect app and Skype for Business clients can send and receive chat messages, but other video clients cannot.	

* At least one Host must join with media (video and/or audio) before Guests are able to join. Alternatively, Connect app users who have joined as a Host in presentation and control-only mode (and who therefore do not act as a trigger for starting the conference) can elect to Start the meeting.

Installing and using Connect apps


About the Connect web app

The Connect web app is automatically available as part of all Pexip Infinity deployments. It provides a WebRTC interface to Pexip Infinity conferencing services.

The web app is supported in the following browser versions, although we strongly recommend using the latest publicly-released version (i.e. "stable version" or "supported release") of a browser:

- Google Chrome version 87 and later (64-bit only) on Windows, Linux, macOS, iOS*, and Android*
- Mozilla Firefox version 78 and later (but v80 or later is recommended for improved network resilience) on Windows, Linux, macOS, and iOS*
- Microsoft Edge version 88 and later (64-bit only) on Windows and iOS*
- Apple Safari:
 - Webapp3: version 15.4 and later on macOS 12 (Monterrey) and later
 - Webapp2: version 15.4 and later on macOS 11 (Big Sur) and later
- Apple Safari on iOS 12.2 and later

* For the best experience on mobile devices, we recommend using the Connect mobile apps.

 Connect web app is not supported on devices running on a Windows Phone OS.

Connect app users can share their screen, images and PDFs from any browser.

Connect web app versions

There are two versions of the Connect web app currently available:

- Connect Webapp3 (available from version 30 onwards)
- Connect Webapp2

Both versions are available from your Pexip Infinity deployment. To determine which version you want to offer web app users by default, see [Setting which version is offered by default](#).

Support for first-time and infrequent users

Connect Webapp3 offers two levels of assistance for participants prior to joining a meeting.

The default is the "express" flow, designed for frequent users who are familiar with setting up and using Connect Webapp3 and who want to join a meeting quickly and simply. When using this flow, the participant's name will be remembered, so all they need to do is enter the meeting name, check that their devices are working and enabled or disabled as expected, and join the meeting.

The second "step-by-step" flow is designed for participants who might be using Connect Webapp3 for the first time or infrequently, and who might require some guidance. This flow takes users through the process of entering their name and then setting up their camera, speaker, and microphone, allowing them to select each device and test that it is working before moving on to the next. This flow also provides users who can't see, speak or hear with the opportunity to enable further support.

The same meeting can be joined by participants using either flow. To enable the "step-by-step" flow for a participant, include `role=guest` within the URL that you provide them with to join the meeting. All other participants who join the meeting using a URL that does not include this parameter will join using the express flow. For more information, see [Links to a specific meeting with additional parameters included](#).

Language support

Connect Webapp3 supports over 20 of the most popular languages. If the user's browser is set to use any one of these supported languages, Connect Webapp3 will use that automatically instead of the default English. Alternatively, users can view Connect Webapp3 in any of the supported languages by appending the appropriate language code to the end of the URL.

Administrators can add additional languages using branding and customization.

Administrators can restrict which languages are supported via the `availableLanguages` setting in the manifest file. When this option is specified, if a user's browser is set to use a language not included in the `availableLanguages` list, the default `en.json` will be used. This option can be used if, for example, administrators edit their language files to use specific terminology and want to limit the number of language files they need to maintain.

Connect Webapp3 currently supports the following languages:

en	English
cs	Czech
cy	Welsh
da	Danish
de	German
es-ES	Spanish (Spain)
es-US	Spanish (Americas)
fi	Finnish
fr-CA	French (Canada)
fr-FR	French (France)
ga	Irish
id	Indonesian
it	Italian
ja	Japanese
ko	Korean
nb	Norwegian (Bokmål)
nl	Dutch
pl	Polish
pt-BR	Portuguese (Brazil)
sv	Swedish
th	Thai
vi	Vietnamese
zh-Hans	Chinese (Simplified)
zh-Hant	Chinese (Traditional)

Accessing a conference or making a call

Connect Webapp3

To access a conference or make a call using Connect Webapp3, users enter into their browser's address bar the IP address or domain name of their nearest Conferencing Node or reverse proxy.

System administrators and conference organizers can also [provide a preconfigured link](#) to a meeting, with one or more options (such as the meeting name) already configured.

Connect Webapp3 offers two different joining flows:

- By default, users are simply asked to enter the **Meeting ID** (if not already included in the URL), which is the alias of the conference or person they want to call. They then get the opportunity to check their setup (camera, microphone and speakers) before selecting **Join**.

This flow is aimed at users who are familiar with the web app and want to join a meeting quickly.

- For occasional or less experienced users who might require assistance with selecting and checking their camera, microphone and speakers, you can include `&role=guest` in the URL you give them to join the meeting. This offers them an alternative join flow that takes them through the setup of their camera, microphone and speakers before they are able to **Join** the meeting. For more information, see [Creating preconfigured links to launch conferences via Connect apps](#).

Connect Webapp2

To access a conference or make a call using Connect Webapp2, users enter into their browser's address bar the IP address or domain name of their nearest Conferencing Node (or reverse proxy if, for example, it is being used to host a customized version of the web app), followed by `/webapp/home` (for example, `confnode.pexample.com/webapp/home`). Users are then presented with the home screen, from where they can check their setup and then select **Call** to enter the alias of the conference or person they want to call.

System administrators and conference organizers can also [provide a preconfigured link](#) to a meeting, with one or more options (such as the meeting name) already configured.

Enabling access for external users

If your Pexip Infinity deployment is located inside a private network and you want to allow Connect app users who are located outside your network to connect to your deployment, see [Using the Connect app from outside your network](#).


Hardware requirements

The performance of the Connect web app typically depends on a combination of factors including the choice of browser, which other applications are currently running on the device, and the device's GPU and CPU specifications.

We recommend that your client device has a minimum of 4 GB of RAM.

In addition, note that use of background effects (blur and replacement) incur a significant local processing overhead which could affect the performance of your device.

About the Connect desktop app

 The Connect desktop app is released separately to Pexip Infinity, and may have been updated since this Administrator Guide was released. For the most up-to-date Connect desktop app user documentation, see [Introduction to Connect app](#).

The Pexip Connect desktop app is a stand-alone video client that provides access to Pexip Infinity services. It is currently supported on:

- Microsoft Windows 10
- macOS 10.11 and later
- Ubuntu Linux 16.04 and later
- Citrix virtual desktops
- Citrix virtual apps

Note that 32-bit operating systems are not supported with the Connect desktop app.

We recommend that you use the [latest available version](#).

Hardware requirements

The performance of the Connect desktop app typically depends on a combination of factors including which other applications are currently running on the device, and the device's GPU and CPU specifications.

We recommend that your client device has a minimum of 4 GB of RAM.

In addition, note that use of background effects (blur and replacement) incur a significant local processing overhead which could affect the performance of your device.

Installing the Connect desktop app

i No special privileges are required to install the Connect desktop app, as it is installed in a per-user context.

To install the Connect desktop app, go to the [Pexip App download page](#) and download and install the appropriate file for your operating system as described below.

Note that 32-bit operating systems are not supported with the Connect desktop app.

Windows

(Supported on Windows 10.)

Download the `pexip-infinity-connect_<release>_win-x64.msi` file for Windows.

Double-click on the .msi file to install the Connect desktop app and then follow the instructions in the installation wizard. During the installation process the Connect app icon is added to the desktop, and entries are added to the Windows registry to allow links prefixed with `pexip:` and `pexip-provision:` to open automatically in the Connect desktop app.

macOS

(Supported on macOS 10.11 and later.)

Download the `pexip-infinity-connect_<release>_darwin-x64.dmg` file for macOS.

To install the macOS client, open this file and drag the **Pexip Infinity Connect.app** into the **Applications** folder.

Linux

Download the `pexip-infinity-connect_<release>_linux-x64.tgz` file for Linux.

To install the Linux client:

1. Create a new directory. For example, to install the client for a single user "alice":

```
mkdir /home/alice/pexapp  
cd /home/alice/pexapp
```
2. Download the Connect desktop app tgz file to that directory and extract the archive. For example:

```
tar -xzf pexip-infinity-connect_<release>_linux-x64.tgz
```
3. Copy the .desktop file to the appropriate location for making the application available for this user as per freedesktop.org-compliant desktop guidelines (see <https://developer.gnome.org/integration-guide/stable/desktop-files.html.en> for more information). For example:

```
cp pexip-infinity-connect_linux-x64/pexip-infinity-connect.desktop /home/alice/.local/share/applications/pexip-infinity-connect.desktop
```
4. Using your preferred text editor, modify the `Exec` line to point to the location of the pexip-infinity-connect binary on your system.

For example:

```
emacs /home/alice/.local/share/applications/pexip-infinity-connect.desktop
```

and make it look something like this:

```
[Desktop Entry]
Name=Pexip Infinity Connect
Exec=/home/alice/pexapp/pexip-infinity-connect_linux-x64/pexip-infinity-connect
Terminal=false
Type=Application
Icon=application-x-executable
```

Note that if you want to install the application for all users (rather than just a single user), follow the same instructions but instead copy the .desktop file into the `/usr/share/applications` directory (you may need root privileges to do this).

Virtual environments

We do not recommend installing Connect desktop app on a virtual environment (Windows or Linux) other than [Citrix](#). In all cases when running in a virtual environment, you must disable hardware acceleration using either of the following methods:

- Set the `ELECTRON_DISABLE_HARDWARE_ACCEL` user environment variable to 1 before launching the Connect desktop app. For example, using the command prompt:

```
set ELECTRON_DISABLE_HARDWARE_ACCEL 1
```

Note that this is a one-time action.

- Launch the Connect desktop app with the command line switch `--disable-gpu`. For example:

```
C:\Users\alice\AppData\Local\Apps\Pexip-Infinity-Connect\pexip-infinity-connect.exe --citrix --disable-gpu
```

If you use this method, you must use it every time you launch the app. You may wish to create a shortcut that launches the Connect desktop app using the `--disable-gpu` switch, which you can then use every time you launch the app.

Citrix

Users can securely access the Connect desktop app via Citrix Virtual Desktops or via Citrix Virtual Apps using the Citrix Workspace app to join VMRs, call through the Pexip Gateway to Microsoft Teams, or simply place point-to-point calls.

See [Using the Infinity Connect desktop client for Citrix virtual desktops](#) for more information on how to install the Connect desktop app in a Citrix environment.

Registering the Connect desktop app

After the Connect desktop app has been installed, it can be registered to a Conferencing Node. The administrator can also provision individual users with their registration details and automatically apply those registration settings to their Connect desktop app.


See [Registering and provisioning the Connect app](#) for more information.

Accessing a conference or making a call

When users open the Connect desktop app, they are presented with the home screen, from where they can check their setup and then select **Call** to enter the alias of the conference or person they want to call (for example `meet.alice@vc.example.com`).

System administrators and conference organizers can also [provide a preconfigured link](#) to a meeting, with one or more options (such as the meeting name) already configured.

About the Connect mobile apps

 The Connect mobile apps are released separately to Pexip Infinity, and may have been updated since this Guide was released. For the most up-to-date Connect mobile app user documentation, see [Introduction to Infinity Connect](#).

The Connect mobile apps can be used by conference participants to control the conference and view presentations from their own personal Android or iOS device, even when they are using a separate telephone or video endpoint to participate in the conference.

Users also have the ability to join a conference from their Android or iOS device, as either an audio-only or a full audio and video participant, allowing them to participate in a conference from anywhere they have an internet connection.

Prerequisites

Connect mobile apps require deployments with HTTPS and valid, trusted certificates.



Connect mobile apps use the Pexip client API, so you must ensure access to this is enabled in your deployment (**Platform > Global Settings > Connectivity > Enable Support For Pexip Infinity Connect And Mobile App**).

Protocols

Connect mobile apps use the WebRTC protocol, so you must ensure this is enabled in your deployment (**Platform > Global Settings > Connectivity > Enable WebRTC**).

Installing the Connect mobile app for Android



The Connect mobile app for Android is available for free from the Google Play store at <https://play.google.com/store/apps/details?id=com.pexip.infinityconnect>. Follow the instructions to download and install the Connect mobile app on your device.

 If you search for "Pexip" in the Google Play store, you will see two apps. The version labeled **Pexip Connect app** with the  icon is the one you should download for use with Pexip Infinity deployments. You will also see an app called **Pexip (My Meeting Video)**; this is for use by customers of the [Pexip Service](#) and **should not be used** in Pexip Infinity deployments.

The Connect mobile app for Android requires **Android 7.0** or later.

Installing the Connect mobile app for iOS

The Connect mobile app for iOS is available for free from the Apple Store at <https://itunes.apple.com/us/app/pexip/id1195088102>. Follow the instructions to download and install the client on your device.

 If you search for "Pexip" in the Apple Store, you will see both the legacy and the latest versions of the app. The version labeled **Pexip Connect app** with the  icon is the latest version and the one you should download. You will also see an app called **Pexip (My Meeting Video)**; this is for use by customers of the [Pexip Service](#) and **should not be used** in Pexip Infinity deployments.

The latest version of the Connect mobile app for iOS, v1.9, is compatible with any iOS device running **iOS 15.2** or later.

Accessing a conference or making a call


When users open the Connect mobile app, they are presented with the home screen, from where they can check their setup and then select **Call** to enter the alias of the conference or person they want to call (for example `meet.alice@vc.example.com`).

System administrators and conference organizers can also [provide a preconfigured link](#) to a meeting, with one or more options (such as the meeting name) already configured.

Registering and provisioning the Connect desktop app

The Connect desktop app can register to a Pexip Infinity Conferencing Node. This enables it to:

- receive calls (as well as place them)
- use directory services to filter and lookup the contact details (phone book) of other devices or VMRs that are set up on the Pexip Infinity platform, making it easier to call those addresses.

 Registration is optional. You do not need to register your device in order to make calls.

The Connect desktop app can also be provisioned with branding details, allowing it to use the same branding as used by the web app.

The mobile clients do not support provisioning, registration or branding/customization.

This topic covers the client [authentication options](#), the [DNS requirements](#), how to [provision the clients](#), some [example provisioning email template content](#), and a description of the associated [user experience](#).

Connect app authentication options

When registering a Connect app to Pexip Infinity, the alias being registered by the Connect app must match one of the entries on the Management Node under **Users & Devices > Device Aliases**. When configuring a device alias, you can specify whether and how a Connect app that is attempting to register with that alias should authenticate itself (authentication is optional but recommended):

- **SSO**: the Connect app uses Single Sign-On (SSO) services such as AD FS to authenticate the registration.
- **Non-SSO**: the username and password credentials associated with the device alias are used to authenticate the registration.

For any given alias, we recommend that you enable Connect app registrations for either SSO or non-SSO authentication, not both.

Setting up appropriate DNS records

The Connect desktop app uses its configured **Registration Host** and performs a DNS SRV lookup on **_pexapp._tcp.<registration host address>** to locate a Conferencing Node to which it can send its registration request.

You must therefore ensure that appropriate DNS records have been set up.

Provisioning the Connect desktop app with registration and/or branding details

Users can manually enter their registration details (alias, credentials, registration host address) into their Connect desktop app. However, as an administrator you can simplify this process by provisioning individual users with their registration details and automatically applying those registration settings to their Connect desktop app.

You can also provision the Connect desktop app with instructions to use the same app branding that has been uploaded to Pexip Infinity (and which is being used automatically by Webapp2). Note that the Connect mobile apps do not support provisioning, registration or branding.

You perform these provisioning tasks by supplying each user with a provisioning URI in the format:

`https://<node_address>/api/client/v2/provision?data=<Base64 encoded name-value pairs>&message=<Base64 encoded message>`

where:

- **<node_address>** is the address of a Conferencing Node. You must ensure that when the end-user attempts to provision their client that they are able to reach the specified node.
- **<Base64 encoded name-value pairs>** are the data values used to provision the Connect app, and are described below.
- **<Base64-encoded message>** is the provisioning message that is displayed to the user. The **message** parameter is optional and by default is "Your Pexip App should have opened and asked to be provisioned. You can now close this window."

Base64 encoding is used to ensure that the data does not get modified by email clients. Note that Base64-encoded data is not encrypted.

For example, the provisioning URI might look like this:

`https://px01.vc.example.com/api/client/v2/provision?data=ZzUmVkaXJl...etc...D%3D&message=bkgY3VzdG9tIG1lc3Nh`

This provisioning URI can be inserted into email messages without the risk of the link being disabled (unlike the [alternative pexip-provision://](#) URI scheme). This means users will have a directly clickable link without needing to copy and paste the link into their web browser.

Provisioning name-value pairs

The name-value pairs that can be provisioned in the **data** query string parameter are described in the following table. If you use Pexip Infinity to bulk provision device aliases and generate emails to each user, you can use the provided template variables and custom Pexip filters to obtain the values for some of the data items and generate the relevant URIs for each user/client.

Each name-value pair must be separated by an **&**. For example (prior to Base-64 encoding):

`name=Alice®istrationHost=px01.vc.example.com®istrationAlias=alice@example.com®istrationUsername=alice®istrationPassword=password123`

The table shows the common data items, and the additional data items that are used for AD FS SSO authentication:

Name	Value	Suggested sync template variable
name	The name of the user as it will appear to other conference participants.	device_username
registrationHost	The domain, IP address or FQDN of the Conferencing Node to which the client should register, for example <code>px01.vc.example.com</code> .	There is no suitable variable for this, as it is not a user specific value.

Name	Value	Suggested sync template variable
registrationAlias	The alias of the device to register to Pexip Infinity.	device_alias
registrationUsername	The username associated with the device alias (registrationAlias). This does not apply if you are using SSO services.	device_username
registrationPassword	The password associated with the device alias (registrationAlias). This does not apply if you are using SSO services.	device_password
brandingURL	A reference to a directory (on an accessible server) that contains customized branding configuration. You typically use this to instruct the Connect desktop app to use the same branding as Webapp2. Prior to version 1.8 of the Connect desktop app, the branding package could be hosted on a Conferencing Node. From 1.8 this is not allowed and the package must be hosted on a different external server. See Customizing and branding the Connect apps for more information.	There is no suitable variable for this, as it is not a user specific value.

Additional data items when using AD FS SSO authentication

adfsFederationServiceName †	The Federation Service name e.g. adfs.example.com.	There are no suitable variables for these items, as they are not user specific values.
adfsResource †	The Resource Identifier e.g. https://pexipappsso.local.	
adfsClientID †	The Client ID e.g. a2a07b42-66d7-41e4-9461-9d343c25b7f3.	
adfsRedirectURI	<p>This is the URI you want the user to be redirected back to after they sign into AD FS. It does not correspond with a value configured on the Management Node but it must be one of the redirect URIs you set up when configuring AD FS on your Windows Server. We recommend you use:</p> <p><code>https://<address>/api/client/v2/oauth2_redirect</code> where <address> is the FQDN of a Conferencing Node or reverse proxy, for example <code>https://px01.vc.example.com/api/client/v2/oauth2_redirect</code>.</p> <p>When the <code>oauth2_redirect</code> page loads it opens the Connect app to complete the sign-in process. The <code>oauth2_redirect</code> page will remain open but it displays a message which by default is "You have successfully signed in. You can now close this window."</p> <p>You can change this message by including the optional base64-encoded <code>message</code> parameter on the <code>oauth2_redirect</code> page URL. For example, the message "my custom message" is "bXkgY3VzdG9tIG1lc3NhZ2U=" when base64-encoded. You would then specify the <code>adfsRedirectURI</code> as follows:</p> <p><code>https://confnode.example.com/api/client/v2/oauth2_redirect?message=bXkgY3VzdG9tIG1lc3NhZ2U=</code></p>	

† These AD FS related data values should correspond to what you have configured in Pexip Infinity (**Users & Devices > AD FS Authentication Clients**) for the OAuth 2.0 Client.

Notes:

- You do not have to provision all of the common name-value data items — if you supply a subset of the data, the user can manually enter the additional data if required.
- When using AD FS SSO provisioning, all of the AD FS data items must be included in the provisioning data.

Example device email template content

The following example content for a device provisioning email template shows how you can build the relevant URI with base64-encoded provisioning data (using device provisioning variables populated from LDAP) and provide a clickable link for the recipient of

the email that will provision their client. The first line in this example defines and sets various variables and the second line incorporates those variables in the paragraph text and link that is displayed to the recipient.

```
{%set provisiondata = "name=" + device_username|capitalize +
"&registrationHost=confnode.example.com&registrationAlias=" + device_alias +
"&registrationUsername=" + device_username + "&registrationPassword=" + device_password
%}

<p>You can open <a href="https://confnode.example.com/api/client/v2/provision?{{pex_url_encode('data', provisiondata|pex_base64)}}">this link</a> to automatically configure your client.</p>
```

Remember to substitute **confnode.example.com** with the address of your Conferencing Node.

You can extend the previous example and include the **message** URL parameter (set to 'Provision your app' in this example) in the provisioning link (the **%set** statement is identical to the previous example):

```
{%set provisiondata = "name=" + device_username|capitalize +
"&registrationHost=confnode.example.com&registrationAlias=" + device_alias +
"&registrationUsername=" + device_username + "&registrationPassword=" + device_password
%}

<p>You can open <a href="https://confnode.example.com/api/client/v2/provision?{{pex_url_encode('data', provisiondata|pex_base64), ('message', 'Provision your app'|pex_base64)}}">this link</a> to automatically configure your client.</p>
```

AD FS SSO examples

This is an example of a provisioning link which can be used to set up Single Sign-On via AD FS:

```
{%set provisiondata = "name=" + device_username|capitalize +
"&registrationHost=confnode.example.com&registrationAlias=" + device_alias +
"&adfsFederationServiceName=adfs.example.com&adfsResource=https://pexipappsso.local&adfsClientID=a2a07b42-66d7-41e4-9461-9d343c25b7f3&adfsRedirectURI=https://confnode.example.com/api/client/v2/oauth2_redirect"
%}

<p>Simply open <a href="https://confnode.example.com/api/client/v2/provision?{{pex_url_encode('data', provisiondata|pex_base64)}}">this link</a> to configure your client automatically.</p>
```

Remember to substitute **confnode.example.com** with the address of your Conferencing Node, and to set the **adfsFederationServiceName**, **adfsResource** and **adfsClientID** variables with the appropriate values for your AD FS service.

This next example shows how to include the "successfully signed in" message URL parameter (set to 'Successfully signed-in message' in this example) in the **oauth2_redirect** link:

```
{%set provisiondata = "name=" + device_username|capitalize +
"&registrationHost=confnode.example.com&registrationAlias=" + device_alias +
"&adfsFederationServiceName=adfs.example.com&adfsResource=https://pexipappsso.local&adfsClientID=a2a07b42-66d7-41e4-9461-9d343c25b7f3&adfsRedirectURI=https://confnode.example.com/api/client/v2/oauth2_redirect?"+ pex_url_encode('message', 'Successfully signed-in message'|pex_base64)) %}

<p>Simply open <a href="https://confnode.example.com/api/client/v2/provision?{{pex_url_encode('data', provisiondata|pex_base64)}}">this link</a> to configure your client automatically.</p>
```

This final example shows how the "successfully signed in" message (on the **oauth2_redirect** URL) and the "provision your app" message (on the **provision** URL) can be customized:

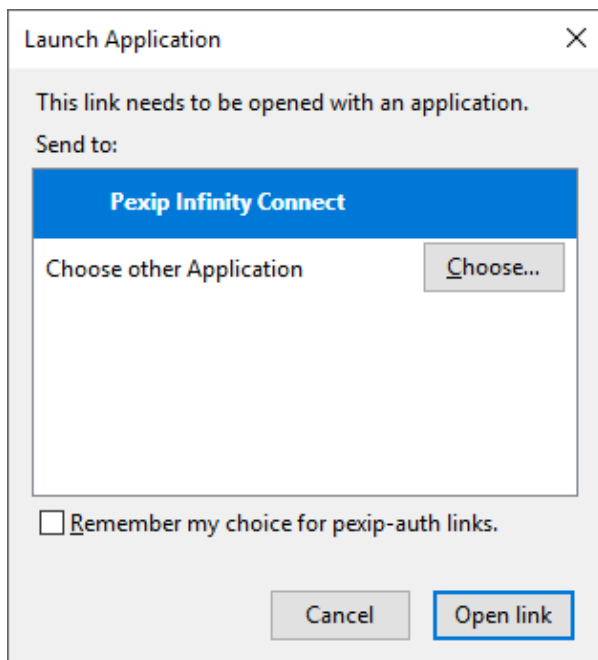
```
{%set provisiondata = "name=" + device_username|capitalize +
"&registrationHost=confnode.example.com&registrationAlias=" + device_alias +
"&adfsFederationServiceName=adfs.example.com&adfsResource=https://pexipappsso.local&adfs
ClientID=a2a07b42-66d7-41e4-9461-
9d343c25b7f3&adfsRedirectURI=https://confnode.example.com/api/client/v2/oauth2_
redirect?" + pex_url_encode(('message', 'Successfully signed-in message'|pex_base64)) %}

<p>You can open <a href="https://confnode.example.com/api/client/v2/provision?{{pex_url_
encode(('data', provisiondata|pex_base64), ('message', 'Provision your app'|pex_
base64))}}">this link</a> to automatically configure your client.</p>
```

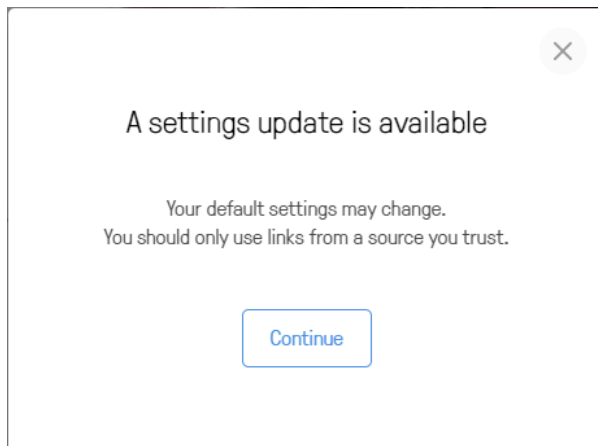
User experience when using the provisioning link

Non-SSO provisioning

When the user clicks on the provisioning link, they are typically asked to confirm or authorize the launch of the Connect app (the exact nature of the request varies according to the platform and the method of launching the link) and then the Connect app will launch and present the user with a confirmation screen:



1. Select **Open Link** to launch Connect app.

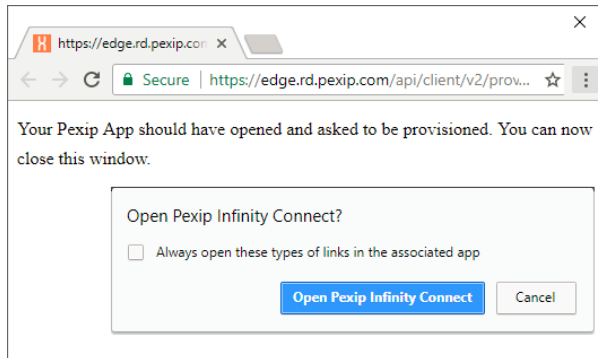


2. Select **Continue** to apply and save the settings contained in the provisioning link.

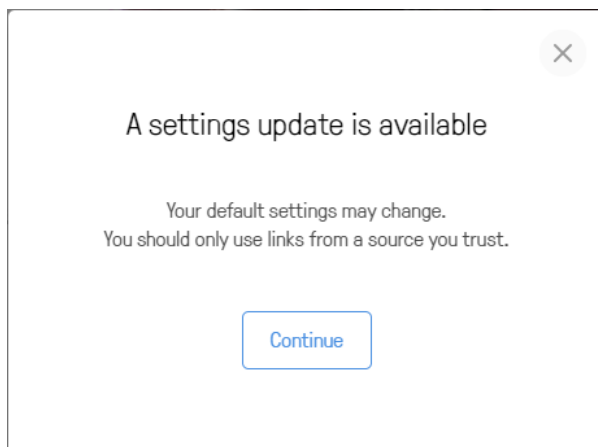
The registration settings in the client are read-only when the client is successfully registered — you must **Unregister** if you want to change them.

AD FS SSO provisioning

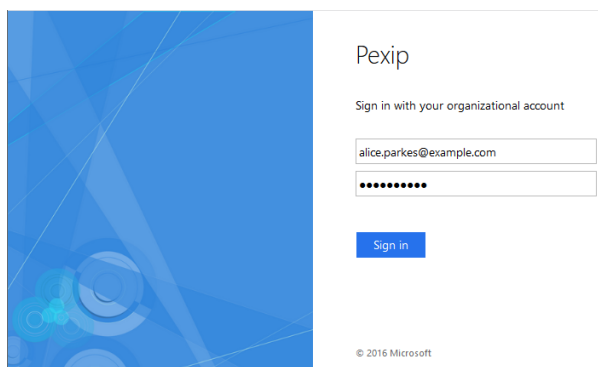
When AD FS SSO provisioning is used, the user is also prompted to sign in to AD FS with their AD credentials. Here are some examples of the screens that are displayed during the provisioning process (the exact nature varies according to the platform, browser and whether the messages have been customized):



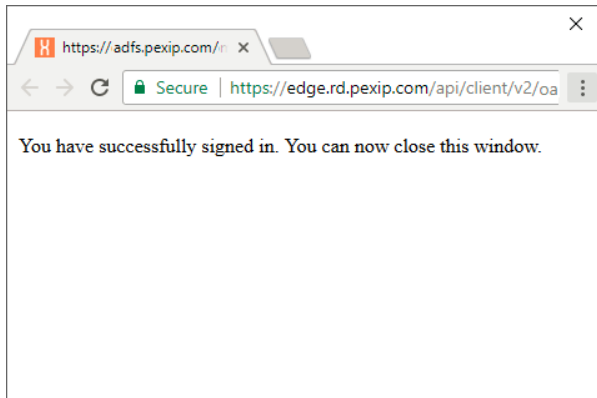
1. Confirm to open the Connect app.



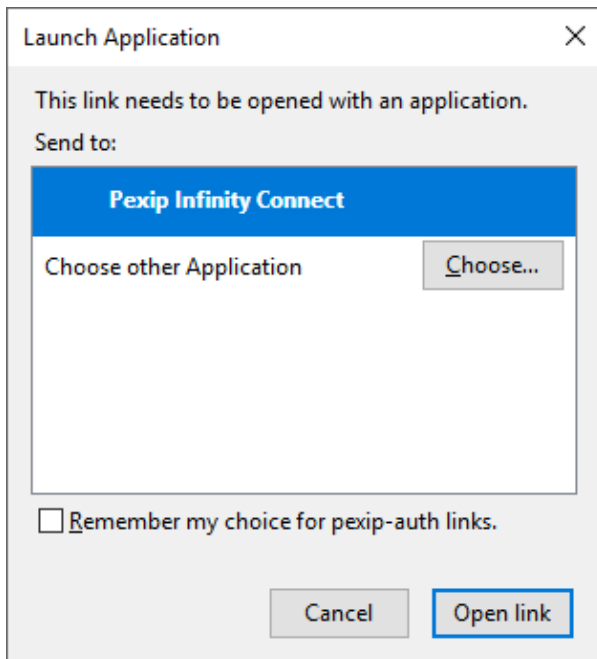
2. Select **Continue** to proceed with provisioning the Connect app.



3. Sign in to AD FS.



4a. AD FS sign-in successful.



4b. Select Open Link to launch the Connect app and complete the sign-in process.


When a client has been configured (provisioned) with SSO registration information, the user name / password fields are blank and the registration settings can only be modified by resetting the Connect app.

Alternative pexip-provision:// URI provisioning scheme

When the Connect desktop app installs, it registers itself to the **pexip-provision://** URI scheme. This provides an alternative provisioning URI that can be used to configure the Connect app with personalized settings for each user. This URI takes the following format:

pexip-provision://settings/?data=<Base64 encoded name-value pairs>

where **data** is set to the same set of name-value pairs as described above.

-  We recommend using the https://<node_address>/api/client/v2/provision style links instead of the **pexip-provision://** style links, as some mail clients (such as gmail) disable embedded **pexip-provision://** style links and other mail clients (such as Outlook) may present users with a security notice warning that the hyperlink may be unsafe and users must choose to continue in order to launch the application.

The following example content for a device provisioning email template shows how you can build the relevant pexip-provision:// URI with base64-encoded provisioning data (using device provisioning variables populated from LDAP) and provide a clickable link for the recipient of the email that will provision their Connect app.

```
{%set provisiondata = "name=" + device_username|capitalize +
"&registrationHost=px01.vc.example.com&registrationAlias=" + device_alias +
"&registrationUsername=" + device_username + "&registrationPassword=" + device_password
%}

<p>You can open <a href="pexip-provision://settings?data={{provisiondata|pex_base64}}">
this link</a> to automatically configure your client.</p>
```

The generated URI for "this link" will take the form `pexip-provision://settings?data=bmFtZT1...etc...HVhcA==`

Using Connect apps to share content

You can use the Connect apps to [share your screen](#) with other participants.

You can also use the text/chat box in the side panel to share videos and images with other Connect app users — just paste the URL of the content you want to share into the chat box (content may be blocked if you are using a reverse proxy with HTTP Content Security Policy (CSP) enabled).

i Webapp2 users can also share [images and PDFs](#). If they are already in a call using another video endpoint, they can join the call using the Connect app [just to share content](#) — for example, if they have joined the conference from a meeting room with a dedicated endpoint, and they want to show a presentation from their laptop without worrying about finding and connecting the correct cables.

Note that:

- An administrator can configure individual Virtual Meeting Rooms and Virtual Auditoriums so that Guest participants are not allowed to present into the conference (they can still receive presentation content from other Host participants). By default, Guests are allowed to present content.
- When someone is **sharing their screen**, their content is sent to other participants at 2 fps by default. However, the presenter can change this rate prior to sharing their screen by selecting **Settings > Advanced Settings > Screen sharing quality**. Note that this setting does not influence the frame rate used when **sharing files and images**, which are only updated each time the file or image changes.

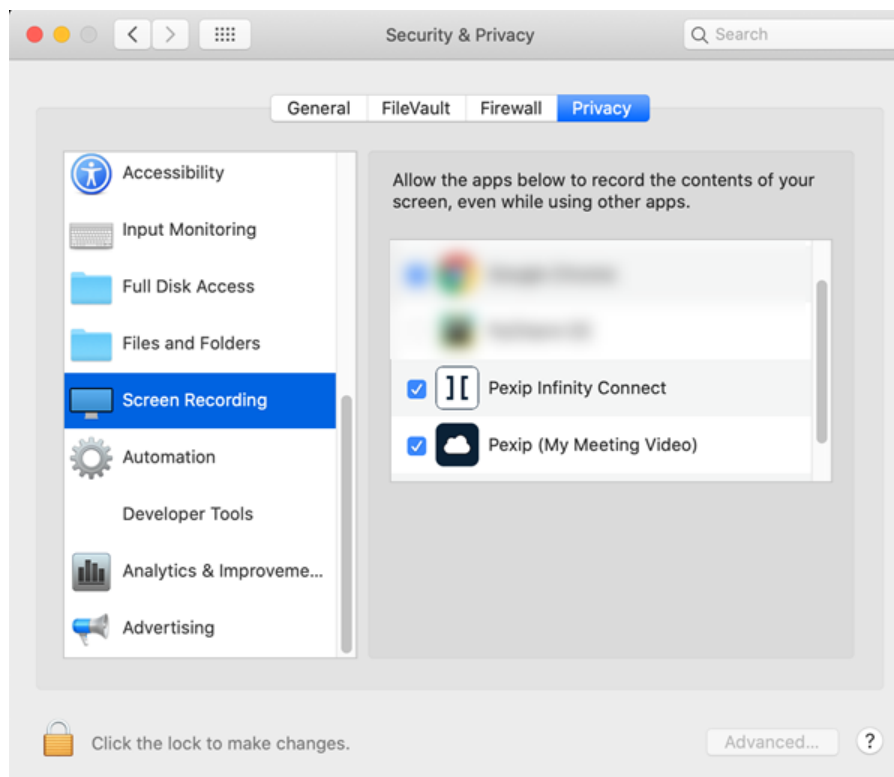
Sharing your screen

Screen sharing is available when using the Connect desktop app, and the Connect web app on the latest desktop browsers (minimum versions: Chrome v72, Firefox v52, Edge Chromium).

Screen sharing is not supported in Safari on macOS, Safari on iOS/iPadOS, or any other browser on iOS/iPadOS.

macOS permissions

Users of macOS 10.15 and later must explicitly grant permission to individual apps to access the device's screen sharing functionality. This permission must be granted to the Connect desktop app or to any browsers used to access the Connect web app, in order for screen sharing to be enabled. This is done via the device's **System Preferences > Security & Privacy > Screen Recording** setting:



Frame rate

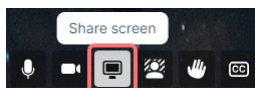
(Available in Webapp2, Connect desktop app and Connect mobile app.)

You can set the frame rate to use when sharing your screen. A lower frame rate produces *sharper* images and is best for static presentations, whereas a higher frame rate is less sharp and is best for content where there is more *motion*. You must set the frame rate to use before you join the conference via **Settings > Advanced Settings > Screen sharing quality**.

Connect web app via Chrome or Edge

You can choose to share the whole screen, an individual application, or an individual tab. To share your screen:

1. From the toolbar at the bottom of the screen, select either:
 - Share screen Webapp3:

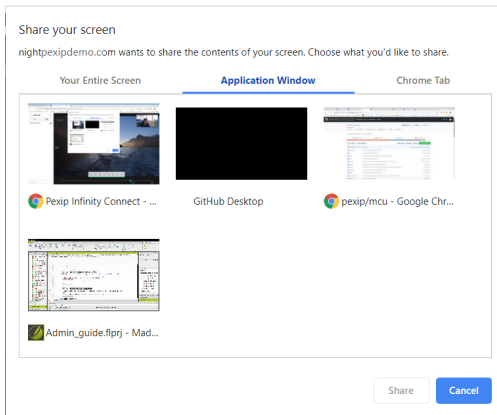


- Share my screen (Webapp2):



2. From the **Your Entire Screen, Application Window**, or browser **Tab** options, select what you want to share.

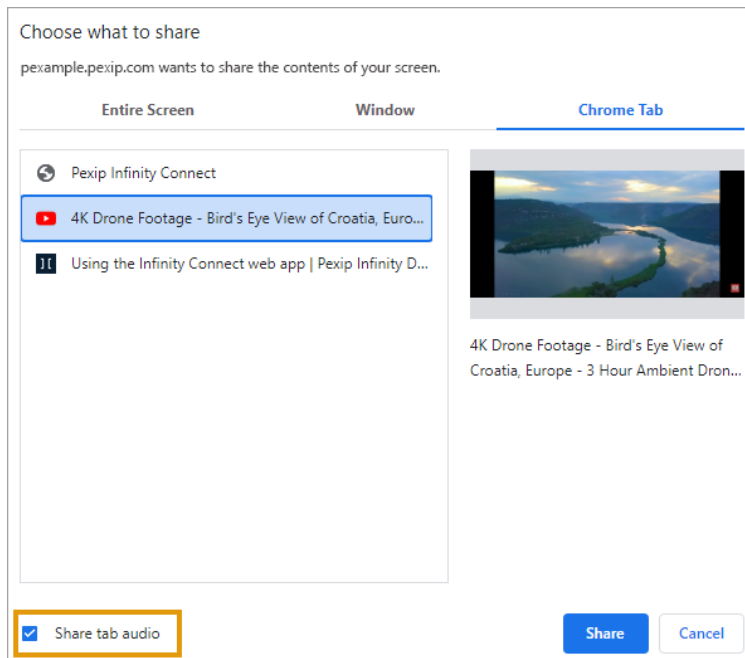
i Any applications that are currently minimized won't appear on the list.



Sharing audio

When you select **Share my screen** in a conference, there is also an option to **Share system audio** if you share your entire screen, or **Share tab audio** if you share a browser tab. On Mac and Linux, you can only share audio from a browser tab. On Windows you can share either system audio or browser tab audio.

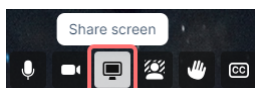
You must have joined the conference with audio to be able to share audio. Muting your microphone does not also mute shared audio.



Connect web app via Firefox

You can choose to share the whole screen, or you can select an individual application window to share. To share your screen:

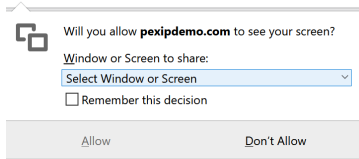
1. From the toolbar at the bottom of the screen, select either:
 - **Share screen Webapp3:**



- **Share my screen (Webapp2):**



2. Select the window or screen you want to share (any applications that are currently minimized won't appear in the list):



Connect desktop app

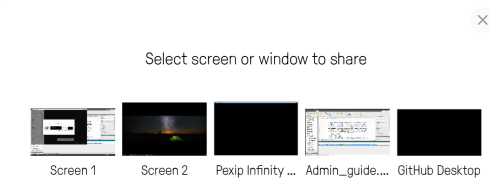
You can choose to share the whole screen, or you can select an individual application window to share. To share your screen:

1. From the toolbar at the bottom of the screen, select **Share my screen**:



2. From the **Your Entire Screen** or **Application Window** options, select what you want to share.

i Any applications that are currently minimized won't appear on the list.



Sharing images and PDFs

(Available in Webapp2, Connect desktop app and Connect mobile app.)

When sharing images or PDFs:

- Supported image formats are JPEG, BMP, PNG and GIF.
- You can share **PDFs** directly from the Connect desktop app, Connect web app (except when used on iOS/iPadOS), and Connect mobile app for Android.
- You can't share **PowerPoint** presentations directly via the Connect apps. To share a PowerPoint presentation, either save the presentation as a PDF and share that, or open the presentation as a slide show and then share your screen.

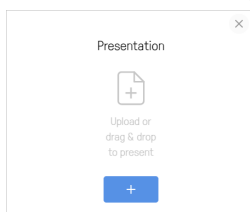
Selecting the images or PDFs to share

To share images or PDFs:

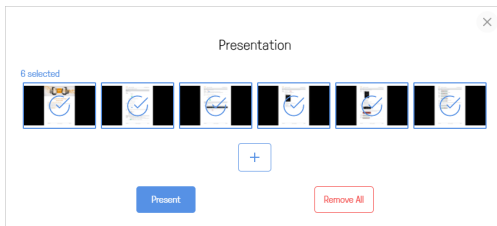
1. From the toolbar at the bottom of the screen, select **Present files**:



The **Presentation** screen appears:



2. Select +, or drag and drop the file(s) you want to share into the **Presentation** window. You can add multiple files, and they can be a combination of images (JPEG, BMP, PNG or GIF) and PDFs (if supported by your device). Each image and PDF page is converted into an individual slide.
3. By default, every slide is selected for presenting, but you can click on individual slides to select and deselect them:



4. When you have selected all the slides you want to share, select **Present**. Use the left < and right > on-screen controls, or the arrow keys on your keyboard, to scroll through the slides. You also have the option to **View presentation in separate window**.
5. To stop sharing the slides, from the toolbar select **Stop presenting**:



Any files you share remain yours — they are not available for other participants to download during or after the conference.

Using the Connect app for presentation, chat and conference control only

If you are already in a conference using an endpoint other than the Connect app (for example, a dedicated meeting room system), you can still access the additional features available to Connect app users (such as conference control, chat, content sharing and viewing, and viewing the participants list) by using the Connect app to join the conference without sending or receiving audio or video — in other words, as a presentation and control-only participant.

The instructions below do not apply to Webapp3 users. For Webapp3 to join as a presentation and control-only participant they must use a **callType** of "none", specified either in the join URL or via the branding manifest. For more information, see [Creating preconfigured links to launch conferences via Connect apps](#) and [Advanced Connect app branding and customization](#).

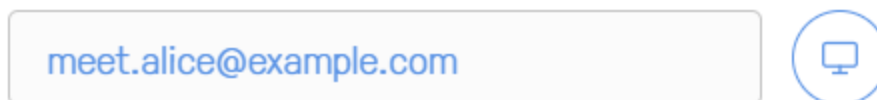
To join a conference as a presentation and control-only participant:

1. Open the Connect app on your computer or mobile device.



2. Select **content**.

3. In the box at the top right of the window, enter the address of the meeting you wish to join:



4. Either click on the icon to the right of the box, or press enter.

The Connect app joins the conference as a presentation and control-only participant — it does not send or receive any audio or video from other participants.

You can now [share your screen](#) or [share images and PDFs](#), and view content being shared by other participants. You can also send and receive chat messages, view the participant list, and (if you are a Host) control aspects of the conference such as adding participants, muting participants, disconnecting participants, and locking the conference.

At any point in the call you can also start sending and receiving audio or video from the Connect app. To do this, select **Start video** or **Start audio** from the toolbar at the bottom of the screen:




Locking a conference and allowing participants to join a locked conference

You can lock a conference if you want to prevent any further participants from joining the meeting after it has started. A conference can be locked and unlocked by Host participants [using the Connect app](#) or [using DTMF-enabled endpoints](#), or by administrators [using the Administrator interface](#).


When a conference is locked, any new participants who attempt to join the conference are held at a waiting screen. They can be [allowed in](#) via the Connect app by Host participants who are already in the conference.

The exact locking behavior depends on whether or not the Virtual Meeting Room or Virtual Auditorium has a Host PIN.

If the service **does not have a Host PIN**:

- Participants can join the conference until it is locked.
- When the conference is locked:
 - A conference locked indicator  is displayed.
 - Any further participants who attempt to join the conference (including any Automatically Dialed Participants and manually-invited participants who have been given a role of Guest) are held at the **Waiting for the host** screen. However, any ADPs and manually-invited participants with a role of Host will join the conference immediately.
 - All participants who are already in the conference are notified of any participants who are attempting to join the locked conference, and can [allow the waiting participants to join](#). Notifications take the form of an on-screen message and an audio message/alert for each participant attempting to join.
- If the conference is unlocked, any participants who are still waiting will automatically join the conference.

If the service **has a Host PIN**:

- Host and Guest participants can join the conference until it is locked.
- When the conference is locked:
 - A conference locked indicator  is displayed to Host participants.
 - New participants who enter the Host PIN will join the conference immediately — locking does not apply to them.
 - Any new Guest participants (including any Automatically Dialed Participants and manually-invited participants who have been given a role of Guest) are held at the **Waiting for the host** screen.
 - All Host participants who are already in the conference are notified of any Guest participants who are attempting to join the locked conference, and can [allow the waiting Guest participants to join](#). Notifications take the form of an on-screen message and an audio message/alert for each participant attempting to join.
- If the conference is unlocked, any Guest participants who are still waiting will automatically join the conference.

All of the on-screen indicators, messages and the **Waiting for the host** screen can be fully customized via the theme associated with your services.


Locking using the Administrator interface

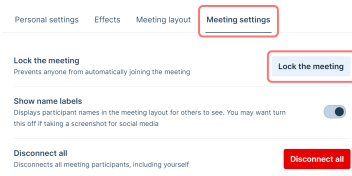
To lock or unlock a conference from the Administrator interface:

1. Log into the Pexip Infinity Administrator interface.
2. Go to **Status > Conferences**.
3. From the **Service name** column, select the conference you want to lock or unlock.
4. At the bottom left of the page, select **Lock conference** or **Unlock conference** as appropriate.

Locking using Connect apps

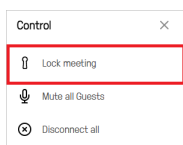
Webapp3

Host participants using the Connect app can lock and unlock the conference they are in by selecting the **User menu**  at the top right of the screen, and from the **Meeting settings** tab selecting **Lock meeting** or **Unlock meeting** as appropriate:



Webapp2/desktop/mobile clients

Host participants using the Connect app can lock and unlock the conference they are in by going to the side panel, selecting **Control** ●●● and then selecting **Lock meeting** or **Unlock meeting** as appropriate:



Host participants using the Connect app can also use the commands `/lock` and `/unlock`.

Locking using DTMF

If DTMF controls are enabled, Host participants using telephones or SIP/H.323 endpoints can lock and unlock the conference using DTMF. The default DTMF entry to do this is `*7` but this may have been customized.

Allowing waiting participants to join a locked conference

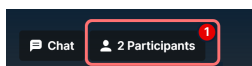
When a new participant attempts to join a locked conference, all Host participants (on any endpoint) in the conference are notified that a participant is waiting to join. However, only Host participants who are using Connect app can admit individual participants into the conference.

 If a locked conference is then unlocked, all participants who are still waiting will automatically join the conference.

Webapp3

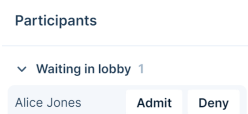
You'll receive a notification on the **Participants** button whenever any participants are waiting in the meeting lobby because they are:

- waiting to join a locked conference, or
- waiting to join a meeting that requires authentication, but they are using an endpoint that does not support authentication.



Select the button to open the **Participants** panel, and locate the participant in the **Waiting in lobby** section. You then have two options:

- To allow a participant to join the conference, select **Admit**.
- If you do not want them to join, select **Deny**. The participant will be disconnected from the meeting.



Webapp2/desktop/mobile clients

Participants who are waiting to join a locked conference are shown in the **Participant** list with a tick and cross next to their names. To allow these participants to join the conference, select the green tick. If you do not want them to join, select the red cross.



Note that if the Host has joined as presentation and control-only (and there are no other Host participants), the Host is not offered the option to allow or deny the waiting participants. However, they can use the **Start the meeting** menu option, which will let in all Guest participants.

Rejecting a request to join a locked conference

If a Host (who is using the Connect app) does not want a waiting participant to join the conference immediately, they have two options:

- To reject the request completely, the Host participant must click on the **Deny** button / red cross icon next to the waiting participant's name. The waiting participant's call will be disconnected.
- To leave the participant at the waiting for Host screen, the Host participant should do nothing. The waiting participant will remain at the waiting screen until:
 - a Host participant chooses to let the waiting participant join the conference, or
 - the conference is unlocked (after which the waiting participant automatically joins the conference), or
 - the participant has been waiting for longer than the specified waiting time (after which the participant is disconnected), or
 - the conference finishes (after which the waiting participant is disconnected).

Administering Connect app

Customizing and branding the Connect apps

The branding and styling of the Connect web apps and Connect desktop app can be customized. This changes the look and feel of the Connect app regardless of which service is being accessed. (However, the theme-based elements of each individual service may also have been customized — a theme changes the look and feel of the actual conference you have joined, or are trying to join.)

Connect app customization can be used to control:


- default settings such as bandwidth, screen sharing frame rate and so on
- the ability to display an image/logo and accompanying welcome text on a landing page, and to use a custom favicon
- language translations and the default language
- the color scheme for buttons, icons and other graphic indicators; elements can be customized individually or a general color scheme can be applied to all similar items.

To customize the **Connect web apps** you typically create a branding package and then upload it to the Management Node. You can use Pexip's branding portals to quickly and easily create branding packages, or you can create them manually. Separate branding packages are required for each of the web app versions in use in your deployment.

If you have created customized branding for Webapp2, you can also use this to customize the **Connect desktop app**. To do this you need to use Pexip Infinity's provisioning features to instruct those clients to override their built-in branding and use the customized branding instead.

This topic explains the two main steps in applying branding to Connect web apps in your deployment:

1. Creating a branding package:
 - using the [Webapp3 branding portal](#)
 - using the [Webapp2 branding portal](#)
 - by [downloading an existing package](#) for editing (including any of the branding packages that are provided by default).
2. [Uploading the branding package](#) to the Management Node so that it is available to all users.

 For information on creating a branding package manually, see [Advanced branding and customization](#).

It also includes information on:

- [Updating an existing package](#)
- [Removing a package \(reverting to default branding\)](#)
- [Applying branding to the desktop clients](#)

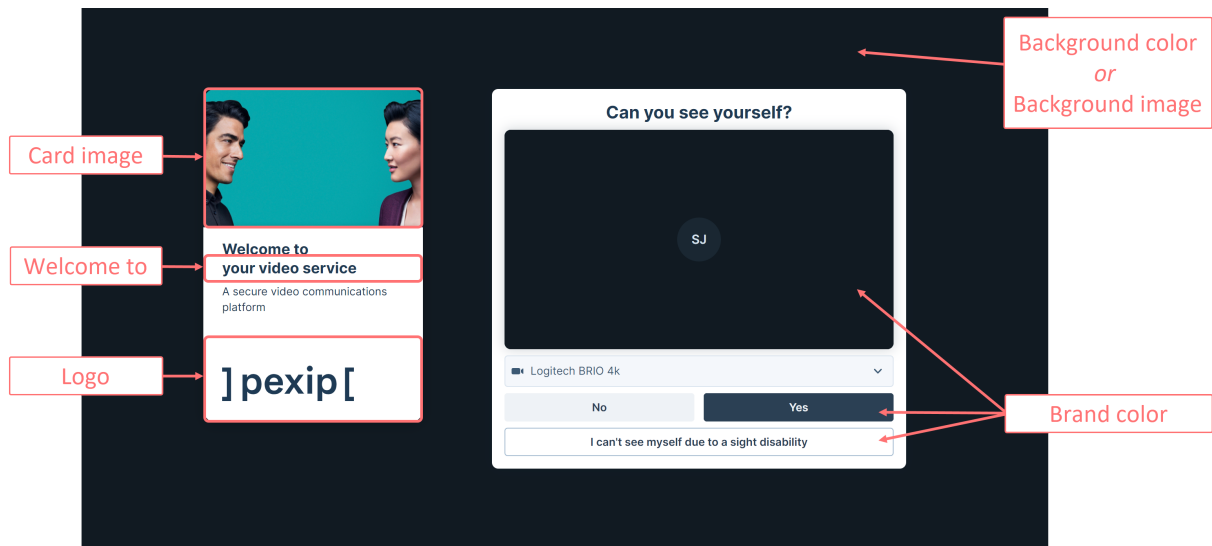
Creating a branding package using the portals

You must create a branding package before you can upload it to the Management Node or use it to brand the desktop client. Our recommended method for creating a branding package is to use the branding portals for either [Webapp3](#) or [Webapp2](#) (although you can also create the required files manually). These web-based portals guide you through the selection of your image files and colors without having to edit individual CSS files etc., and then generates the customized branding package for you.

Using the Webapp3 branding portal

1. Go to the Pexip branding portal for Webapp3 (<https://branding.pexip.io/>).
2. Select one of the options:
 - to create a completely new package, select **New brand**
 - to use an existing package as the basis of your new package, select **Load existing brand**.
3. From the panel on the left, select each option and edit it to suit your brand's requirements.

The image below shows what elements of the Connect web app can be customized. The table that follows describes each option in more detail. If you are intending to edit the resulting branding package manually, it also gives the section within the manifest.json file where that element is defined.




Option	Description	Definition in manifest.json
Welcome to	Specifies the text that appears under the card image on the landing page.	"brandName"
Logo	<p>An optional image/logo file that can be displayed at the bottom of the area on the left side of the landing page.</p> <p>This file must meet the following requirements:</p> <ul style="list-style-type: none"> transparent maximum displayed width: 168px maximum displayed height: 80px format: we recommend .SVG 	"logo"
Brand color	<p>Specifies a color that the branding portal will use as a base to automatically generate the range of colors used for interactive elements of the user interface, such as buttons and text.</p> <p>You can use the color picker tool to select a color from any of the images you have already uploaded.</p>	"baseColor" "colorPalette"
Background color	<p>Specifies the landing page's background color if no background image is specified.</p> <p>You can use the color picker tool to select a color from any of the images you have already uploaded.</p>	"backgroundColor"
Card image	<p>An optional image/logo file that can be displayed at the top of the area on the left side of the landing page.</p> <p>This file must meet the following requirements:</p> <ul style="list-style-type: none"> resolution: 640x360 size: 70-140 kB format: .JPG, .JPEG, .PNG or .WEBP 	"jumbotron"

Option	Description	Definition in manifest.json
Background image	<p>An optional image to display behind all other elements on the landing page.</p> <p>This file must meet the following requirements:</p> <ul style="list-style-type: none"> resolution: 1920x1080 size: 500-600 kB format: .JPG, .JPEG, .PNG or .WEBP 	"background"
Overlay	Makes the background image less prominent, either by darkening or lightening it.	"overlay"
Overlay opacity	Determines the degree to which the background image is darkened or lightened.	"overlayOpacity"
Application title	Specifies the text that appears on the browser tab when using the Connect web app.	"appTitle"
Terms of service	Specifies the URL to which the "terms and conditions" text on the user name step of the join flow is directed.	"termsAndConditions"
Disconnect destination	Specifies the URL to which users are directed when a call is completed (instead of returning to the app home page).	"disconnectDestination"
Handle OAuth 2.0 Redirects	<p>(Supported in Pexip Infinity v34 and later)</p> <p>Enables support for features (such as plugins) that require authentication to a third party using an OAuth 2.0 / OpenID Connect flow, with a redirect destination of <code>webapp3/oauth-redirect</code>.</p>	"handleOAuthRedirects"

- When you have finished configuring your branding, from the top right of the page select **Download**. This creates and downloads a **brand.zip** file containing your client customizations.
- If you wish to make further customizations you can unpack the file, edit it, and then re-zip it. See [Manually customizing Connect Webapp3](#) for full information.
- [Upload the branding package](#) to your Management Node.

Using the Webapp2 branding portal

You can use the Webapp2 branding portal to customize both Webapp2 and the Connect desktop app.

- Go to the Pexip branding portal for Webapp2 (<https://webapp2-branding.pexip.io/>).
- From the title bar at the top of the page, select which version of Pexip Infinity you have installed, so that the relevant branding and customization features can be offered.
- From the bar at the bottom of the page, select the **Download Builds** button  to view the options for building a package for subsequent downloading.
- You can choose to create new customizations, or edit an existing customization that you have previously created. Configure your customization as required, selecting the relevant image files, colors and settings:
 - The **App Editor** changes the look and feel of the Connect apps, including enabling an image/logo on the landing page.
 - The **Customizations** section controls the client's configuration settings, including default options, languages and plugins.
 - The **Splash Screens** section doesn't directly affect the Connect apps. It is used to customize the Pexip Infinity themes (which are used when you join a VMR or other service either via a Connect app or other endpoint) and generates a separate ZIP package when built.
 - The **Languages** section allows you to set up additional languages for the Connect app, or to create a modified version of the default English text strings. When creating a new set of language strings the **Name** is the name you will see within the portal, and the **Label** is the name users will see within the app; the **Locale** enables that language to be used automatically if it matches the browser's default language. If you set up new language option then you must use the **Customizations** section to select the new/modified languages you want to include in your branding package (and deselect the original **English** language strings if required).

5. When you have finished configuring your branding, go to the Dashboard and from the **Download Builds** section at the bottom of the page select the relevant **App Edits and Customizations** and then **Build** your customization package. If you have added new languages they are automatically included in your build depending upon which languages are selected in the **Customization**. This creates and downloads a `branding_<date>.zip` file containing your client customizations.
 6. If you intend to use this branding on a v31 or later Pexip Infinity deployment, you must restructure the contents of the .zip file so that the contents sit within a `webapp2/branding` folder. For full details see [Using a pre-v31 Webapp2 branding package](#).
 7. If you wish to make further customizations you can unpack the file, edit it, and then re-zip it. See [Manually customizing Connect Webapp2](#) for full information.
 8. [Upload the webapp2.zip file](#) to your Management Node.
- i** This branding package is used to customize the Connect web app by default, but you can also automatically [apply the same branding to the Connect desktop app](#).

Downloading an existing package

As an alternative to using the branding portals to create new branding packages, you can download, edit, and upload an existing branding package. This might be a default branding package downloaded from the Management Node, a package previously created using the branding portals, or a package downloaded from previous versions of Pexip Infinity.

If you are using a Webapp2 branding package created for a v30 or earlier version of Pexip Infinity, you must amend the file structure before uploading it — see [Using a pre-v31 Webapp2 branding package](#).

Manually editing an existing package is useful if you have very specific modifications that you want to apply to the branding files, or if you are including plugins. Note that manual configuration requires knowledge of core web-design technologies such as HTML, JavaScript and CSS.

- i** This section covers how to obtain branding packages. For full information on manually editing these packages, see [Advanced branding and customization](#).

Downloading branding packages from the Management Node

Pexip Infinity includes a number of default branding packages that you can download and edit to create your own customized branding. . In addition, if you have existing custom branding files uploaded for any of the three web apps, you can also download and use these as the basis on which to apply your modifications.

You may also wish to download an existing branding package in order to upload it to an external server if you are hosting the Connect web app externally.

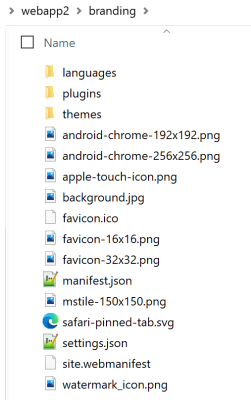
To download an existing branding package from the Management Node:

1. Go to **Web App > Web App Branding**.
2. Select the Pexip branding package for the relevant web app version and then from the bottom of the detail page select **Download**. A ZIP file containing the selected branding files is downloaded to your local file system.
3. Unpack the downloaded file and apply your modifications to the relevant files.
4. Repackage your branding files into a single ZIP file. This file will need to have a different name to any packages already uploaded. If you are editing an existing branding package, after uploading simply redirect the current path to point to this new package.
 - i** The ZIP file must contain the complete set of branding files. You must retain the original file/folder structure in the rebuilt ZIP file.
 - i** You must include the `manifest.json` file in the `webapp2/branding` or `webapp3/branding` folder.
5. [Upload the branding package](#) to your Management Node.

Using a pre-v31 Webapp2 branding package

If you are using an existing Webapp2 branding package that was created for Pexip Infinity version 30 or earlier, you will need to change the structure of the files and folders within the branding package before it can be uploaded to the v31 or later Management Node. To do this:

1. Unpack the contents of the .zip file. This will be unpacked to a `/webapp2` folder.
2. Within the `/webapp2` folder, create another folder called **branding** and move the contents of the `/webapp2` folder into this subfolder. This will give you a structure that looks like the following:



3. If you wish to make further customizations you can edit the contents of the folder, and then re-zip it. See [Manually customizing Connect Webapp2](#) for full information.
4. [Upload the webapp2.zip file](#) to your Management Node.

Editing an existing package

You can also use as the basis of your new branding packages any other existing branding packages. These might be downloaded from previous versions of Pexip Infinity, from either of the branding portals, or be packages that you created entirely manually. Simply ensure that these packages meet the requirements outlined in [Manually customizing Connect Webapp3](#) or [Manually customizing Connect Webapp2](#) as appropriate, before zipping and [uploading](#) the new package.

Uploading the branding package

Branding packages are uploaded as .ZIP files. The files must contain the required branding files in a `webapp3/branding` or `webapp2/branding` subfolder.

To upload a branding package to your Management Node:

1. Go to **Web App > Web App Branding**.
2. From the bottom of the page, select **Add Webapp branding package**.
3. On the **Add Webapp branding package** page, enter the following information about the package you wish to upload:

Name	The name for this branding package.
Description	An optional description of this branding package, to help you identify it easily.
Web app version	Select the web app version to which this branding package will apply.
Branding package to upload	Select Choose File and select the ZIP file containing your customizations.

4. Select **Save**.
The branding package will be verified and uploaded.
5. After the branding package is uploaded, in order to use it you must select it for use with one or more web apppaths, creating a new path if necessary.

i Any changes you make to branding packages and paths must be replicated out to all Conferencing Nodes before being available (typically after approximately one minute).

Updating an existing package

To make changes to a branding package already in use in your deployment:

1. [Download](#) the old branding package and edit it with your updates.
 2. Save the updated branding package as a ZIP file.
 3. Upload the updated branding package (**Web App > Web App Branding**), giving it a different name to the existing version.
 4. Edit all paths that use the old version of the branding package so that they use the new version instead.
 5. Delete the old branding package.
- i** Don't delete the old branding package before uploading the new package and redirecting to it. If you do, all the paths that used the deleted package will revert to using the default branding, meaning that participants will experience the default branding until you have redirected the relevant paths to use the new package.

Removing a package (reverting to default branding)

Default branding packages cannot be deleted. You can remove customized branding in one of two ways:

- If you wish to keep the branding for possible later use, simply edit the paths that currently use that package, so that the paths either use a different branding package, or use the default branding.
- If you no longer wish to keep the branding package, delete it. Any paths that point to deleted branding packages will revert to using the default branding.

To delete a branding package:

1. On the Management Node, go to **Web App > Web App Branding**.
2. Select the tick boxes next to the branding packages you wish to remove.
3. From the **Action** drop-down, select *Delete selected Web app branding packages*.

Wait for the customized branding to be removed from all Conferencing Nodes (typically after approximately one minute). After this time, all new participants accessing the path will see the default branding. Note however that any participants currently using the deleted package will continue to see the deleted branding until they refresh their browser.

Applying branding to the desktop clients

Any branding package that is uploaded to the Management Node is only applied to the relevant **Connect web app**.

If you have created customized branding for Webapp2, you can also use this to customize the **Connect desktop app**. To do this you need to use Pexip Infinity's provisioning features to instruct those clients to override their built-in branding and use the customized branding instead. This is achieved by specifying the **brandingURL** provisioning parameter when you construct each individual desktop client user's provisioning URI.

- The **brandingURL** parameter must refer to a directory on an accessible server that contains the branding package.
- The branding package must be signed, and the client must upload a trusted (public) key before the branding can be applied.
- The branding package must be presented as a **branding.zip** file and an associated **branding.zip.sig** file.

For example, if **brandingURL** = **pexample.com/foo**, then you need to provide **pexample.com/foo/branding.zip** and **pexample.com/foo/branding.zip.sig**.

After a Connect app has been provisioned with a **brandingURL** provisioning parameter, every time it launches it checks the contents of the branding files at the brandingURL location to see if the branding has changed (it checks to see if the **brandingID** in the **manifest.json** file has changed). If the branding has been updated, the client fetches and caches the relevant files.

Note that the desktop client's favicon, taskbar/tray icons and app name cannot be updated via branding as these elements are fixed during the installation of the client software.

See [Registering and provisioning the Connect desktop app](#) for full instructions about how to set up provisioning URIs. Note that the client does not need to be registered in order to use the branding provisioning feature.

Note that as of version 1.8 you cannot apply branding to the mobile clients, and the desktop client branding can no longer be hosted on Conferencing Nodes.

Creating and signing a branding package for the desktop clients

The branding package in the **brandingURL** location must be presented as a **branding.zip** file plus an associated **branding.zip.sig** file that contains the package's signature.

Contents of branding.zip

Typically we recommend that you use a **branding.zip** file produced by the Pexip branding portal as this is a suitable zip file/format and contains all of the relevant content (although you must still sign it yourself).

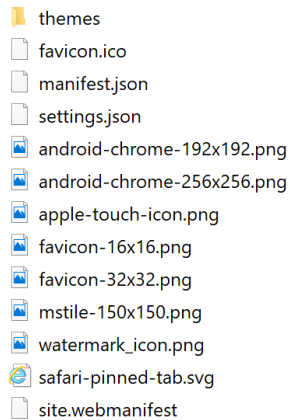
The **manifest.json** is automatically generated by the Pexip branding portal and includes the **brandingID** timestamp and also indicates which parts of the app are customized.

If you want to create your own **branding.zip** file then it must contain a **webapp2** folder as its root folder and that must then have the following structure/contents:

- **manifest.json** (mandatory)
- **settings.json** (optional)
- **watermark_icon.png** (optional)
- **favicon** files (optional, applies only to the web app)
- **site.webmanifest** (optional)
- **themes** directory containing **styles.css** (both optional)

as shown below:

Name



Signing the branding package

You must use JSON Web Token (JWT) to sign the package. (JWT is an open standard that defines a way for securely transmitting information between parties as a JSON object.)

As part of the process to sign the branding package you need a public/private keypair. You may already have a keypair that you can use for this process, or you can use a third-party tool such as PuTTYgen to generate a keypair. The key must be in RSA format and at least 2048 bits. Alternatively you can log in to your Management Node over SSH and run the following commands to generate a private and public key pair:

```
openssl genrsa -out /dev/shm/privatekey.pem 2048
openssl rsa -in /dev/shm/privatekey.pem -pubout -out /tmp/publickey.pem
```

To sign the branding package and create your .sig file:

1. Create your **branding.zip** file.
2. Using a plain text editor, create a shell script file called **mkjwt.sh** containing the following code:

```
#!/bin/sh

set -e

if [ $# -ne 2 ]; then
    echo "Usage: $0 <privatekey.pem> <branding.zip>" >&2
    exit 1
fi
```

```

HEADER="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9"
HASH=$(openssl dgst -sha256 $2 | sed -e 's/^[^ ]* //' )
PAYLOAD=$(echo -n "{\"sha256\":\"${HASH}\"}" | base64 -w0 | sed -e 's/\+/-/g' -e 's/_/_/g' -e 's/=//g')
SIG=$(echo -n "${HEADER}.${PAYLOAD}" | openssl dgst -sha256 -sign $1 | base64 -w0 | sed -e 's/\+/-/g' -e 's/_/_/g' -e 's/=//g')

echo "${HEADER}.${PAYLOAD}.${SIG}"

```

- Copy your private key file (named **privatekey.pem**), the **branding.zip** file and the **mkjwt.sh** file into the **/dev/shm** directory on the Management Node using an SCP (Secure Copy) client, for example WinSCP.
- Connect over ssh into the Management Node as user **admin** with the appropriate password.
- Run the following commands:


```
cd /dev/shm
chmod 0755 mkjwt.sh
```
- Run the following command to generate the **.sig** file:



```
./mkjwt.sh privatekey.pem branding.zip >branding.zip.sig
```
- Run the following command to remove the private key file:


```
rm ./privatekey.pem
```
- Use the SCP client to copy the generated **branding.zip.sig** file to your local machine.

Using the branding package on the desktop client

The client will not automatically use the customized branding package (as referred to at the provisioned **brandingURL** location).


Each client user must first import a trusted key via **Settings > Advanced settings > Import trusted key** and confirm that they want to apply the branding. The trusted key file they need to import (i.e. that you need to distribute) is the public key file used as part of the key pair used to create the JWT signature.

-  Only distribute the public key. Do not distribute the private key.

Creating path-based web app branding

Your Pexip Infinity deployment uses URL paths (such as **/webapp**, **/webapp2**, and **/webapp3**) to provide users with access to the web app. For each path, you can configure which web app version and branding users are offered when accessing the web app via that path, and you can create and configure additional paths to offer users a variety of differently-branded web app experiences within your environment.

This topic describes how to [create and configure web app paths](#) to determine which combinations of web apps and branding are offered in your deployment, in addition to the [default paths](#).

-  Additional parameters can be appended to the path to allow users to join specific meetings and pass in other information specific to the call being placed. For more information, see [Creating preconfigured links to launch conferences via Connect apps](#).

Prerequisites

You must have already created and uploaded to the Management Node any branding packages to be used.

Note that the branding package content and requirements for each of the three versions of the web app are different, so you may need to upload more than one branding package if you want to create web app paths for more than one branded web app version.

Creating a web app path

To create a specific web app path and apply a branding package to it:

- From the Management Node, go to **Web App > Web App Paths**.
- From the bottom of the page, select **Add Webapp path**.

3. Complete the following fields:

Path identifier	<p>The string that, when appended to the IP address or domain name of a Conferencing Node or reverse proxy, will provide the URL used to access this branded version of the web app.</p> <p>For example, if users normally access the web app via <code>webapp.pexample.com</code> and you enter <code>sales</code> here, users will be able to access this branding via <code>webapp.pexample.com/sales</code></p> <p>Permitted characters are a-z, A-Z, 0-9, - and _.</p>
Description	An optional description of this path.
Web app version	The version of the web app accessed using this path.
Web app bundle	<p>If you want this alias to direct to a specific revision of the selected Web app version, select the associated software bundle to use.</p> <p>If you want this alias to always use the revision of the selected Web app version that is included with Pexip Infinity (rather than any subsequent revision), then use the default <i>Use default web app bundle</i> option. If you select this option, when you upgrade to a later version of Pexip Infinity, this will automatically use the bundle shipped with that later version.</p>
Branding package	<p>Select the branding package to use for this path.</p> <p>The list of options will include the available branding packages for the selected Web app version.</p> <p>If you don't want to apply a branding package and instead want to use the default Pexip branding for the selected web app version, select <i>Use default branding</i>.</p>
Enabled	Determines if the path is enabled or not. Use this setting to test configuration changes, or to temporarily disable specific paths.
Set as default	Select this option if you want users who have entered the address of a Conferencing Node or reverse proxy, either on its own or with <code>/webapp</code> appended, to redirect to this particular path and associated branding. For more information, see Redirecting "/webapp" (or no path)

4. Select Save.

You can now test the branding by entering the path in your browser and using it to join one of your Pexip Infinity services.

Default paths and default branding

The paths and associated branding that are included by default in your Pexip Infinity deployment, and which of those paths users are redirected to if they either don't specify a path or use `/webapp`, will depend on whether you are installing a new v31 or later deployment, or upgrading to v31 or later from v30 or earlier.

Redirecting "/webapp" (or no path)

Users access the web app by entering the IP address or FQDN of Conferencing Node or reverse proxy in your deployment, either on its own or with a path appended. You can configure where to redirect users who either do not enter a path or enter the `/webapp` path (supported by previous versions of Pexip Infinity).

If you don't [manually select a default path](#), the path that `/webapp` (or no path) redirects to depends on whether this is a new v31 or later Pexip Infinity deployment or an upgrade from v30 or earlier, and if the latter, the setting for the now-deprecated **Default web app** (v30) or **Do not default to the legacy Web App** (v29 and earlier) fields under **Platform > Global Settings > Connectivity**, as follows:

New v31 or later installation

- `/webapp` (or no path) redirects to `/webapp3`

Upgrade from v30

- If **Default web app** was set to *Always use latest at upgrade* or *Webapp3*, `/webapp` (or no path) redirects to `/webapp3`
- If **Default web app** was set to *Webapp2*, `/webapp` (or no path) redirects to `/webapp2`

- If **Default web app** was set to **Webapp1**, note that this is no longer supported from v32 onwards. Please contact your Pexip authorized support representative if you still require access to Webapp1.

Upgrade from v29 and earlier

- If **Do not default to the legacy Web App** was previously enabled (the default), **/webapp** (or no path) redirects to **/webapp2**
- If **Do not default to the legacy Web App** was previously disabled, **/webapp** (or no path) redirects to **/webapp1**. However, this is no longer supported from v32 onwards. Please contact your Pexip authorized support representative if you still require access to Webapp1.

Manually changing the default path

To see which path is currently used as the redirect, go to the list of web app paths (via **Web App > Web App Paths**). The currently selected path is indicated by a tick in the **Default path** column.

To change the path to be used, select the path and then select [Set as default](#).

- i** If none of the paths are set as the default, users who either don't specify a path or use **/webapp** will be presented with a 404 error.

Default path configuration

Your Pexip Infinity deployment includes the following paths and associated branding. You can delete and edit these default paths, as well as [creating new paths](#).

Path	Web app version	Branding package name
/webapp1	Webapp1 i This is no longer supported from v32 onwards. Please contact your Pexip authorized support representative if you still require access to Webapp1.	Pexip webapp1 branding
/webapp2	Webapp2	Pexip webapp2 branding
/webapp3	Webapp3	Pexip webapp3 branding

Default branding packages

The default branding packages included with Your Pexip Infinity deployment will depend on whether this is a [New v31 or later installation](#) or an [Upgrade from v30 or earlier](#).

- i** You can't delete or edit any of these branding packages, but you can download them to use as the basis of your own branding packages.

New v31 or later installation

Every Pexip Infinity deployment includes the following branding packages:

- **Pexip webapp1 branding**: containing the default branding for Webapp1
- **Pexip webapp2 branding**: containing the default branding for Webapp2
- **Pexip webapp3 branding**: containing the default folder structure for Webapp3 branding files.

Upgrade from v30 or earlier

Each upgraded Pexip Infinity deployment includes the branding packages listed above for a [New v31 or later installation](#). In addition, if your previous deployment included custom branding for one or more web apps, you'll also have the associated branding packages listed below.

Note that these additional branding packages will not initially be associated with any web app path; if you wish to continue to use this branding in your deployment you'll need to either apply the package to the relevant [default path](#) (replacing the default branding for that path), or create a new path and apply the branding package to it.

- **webapp1 custom branding:** containing your previous custom branding for Web app 1
- **webapp2 custom branding:** containing your previous custom branding for Web app 2
- **webapp3 custom branding:** containing your previous custom branding for Web app 3

Advanced Connect app branding and customization

Most customization requirements for the Connect apps can be implemented by using the Pexip branding portal to generate a package of branding files and then applying that branding by uploading the branding package to the Management Node (see [Customizing and branding the Connect apps](#)). However, for advanced customization requirements you may need to make manual changes to the branding files.

This topic covers how to manually customize both the current [Webapp3](#) and the previous [Webapp2](#) versions of the Connect web app.

You may also need to manually customize these files if you are hosting the web app on an external server or reverse proxy. For information about how to apply customized branding in this situation, see [Hosting the web app externally](#).

Note that manual configuration requires knowledge of core web-design technologies such as HTML, JavaScript and CSS.

Hosting on the Conferencing Nodes

The standard method for applying branding to the Pexip Infinity platform is to upload a branding package to the Management Node and associating the package with a unique web app path. The package is then pushed out to all Conferencing Nodes, from where those customizations are served to all web app users joining using that path.


Each web app version requires separate branding customizations. You can create multiple unique paths where each path is used to host different combinations of web app version, revision and branding.


Branding customizations that are applied to the Connect web app via the Management Node will persist over upgrades to subsequent versions of Pexip Infinity software (although you may need to adapt the customization to cater for any new features when upgrading to a new major release).


Manually customizing Connect Webapp3

This section describes how to manually customize the latest web app, Connect Webapp3. You do this by manually creating a client branding package, uploading it to the Management Node, and then specifying the web app path used to access the branding.

The complete set of files that can be included in a branding ZIP package for upload via the Management Node, or manually copied into the **branding** directory when hosting the app on a different server, are summarized below, and are then explained in more detail in the subsequent sections.

- The [manifest.json](#) controls which customized settings take effect. It also contains the application and default user settings.
 -  The [manifest.json](#) file is the only mandatory file. Your customizations will not take effect if this file is not included.
- [landing page background image](#)
- [jumbotron image](#)
- [logo](#)
- [participant background image](#)
- any new or edited [language files](#)
- the content shown in the [body of the custom step](#) (if enabled)

 When creating your branding package ZIP file, ensure that [manifest.json](#) and any of the optional files listed above that you are including are contained in a **webapp3/branding** folder inside the ZIP file for uploading to the Management Node.

 When editing the configuration files, you must use a text editor that does not apply "smart quotes" or make any automatic text changes, as the files are sensitive to correct formatting. Use a code editor or simple file editor instead of word processing software.

Application manifest (manifest.json)


The `manifest.json` file controls which customized settings take effect, and which image files are used. When customizing Webapp3, you must include the `manifest.json` file in the `webapp3/branding` folder (creating it, if necessary, in the first instance).




i This file must use correct JSON syntax, otherwise it will fail validation when uploaded to the Management Node. We recommend that you use a JSON validation tool (such as <https://jsonlint.com>) to check the syntax.

In particular, note that items listed inside braces `{ }` or brackets `[]` must be separated by commas `,`, but there **must not** be a comma after the last item in the list, before the closing brace/bracket.

The properties that can be included in the `manifest.json` are described below, starting with those that are required:

Property	Description
Required	
version	<p>Specifies the manifest schema version.</p> <p>Currently this should be set to <code>0</code>.</p>
meta	<p>Specifies the manifest metadata. It is not used by the app, and should use the settings given in the example file below, i.e.:</p> <pre>"meta": { "name": "DEFAULT", "brandVersion": "n/a" },</pre>
images	<p>Specifies the files to be used for the custom background, logo and jumbotron images on the landing page, using the format:</p> <pre>"images": { "background": "../background.jpg", "logo": "../logo.svg", "jumbotron": "../jumbotron.jpg" },</pre> <p>The actual file name of each of image does not matter.</p> <p>The "images" property must be specified; if there aren't any images to include you must specify an empty list, i.e.:</p> <pre>"images": {}</pre> <p>For more information on image size and format, see Images.</p>
translations	<p>Specifies any supported languages that have been customized, and/or any additional languages, along with the files that provide the translations for each, in the format:</p> <pre>"translations": { "en": "../en.json", "af": "../af.json" }</pre> <p>The "translations" property must be specified; if there aren't any translations to include you must specify an empty list, i.e.:</p> <pre>"translations": {}</pre>
Optional	
applicationConfig	<p>Specifies any overrides of the application's default settings.</p> <p>For full details, see Overriding the default application settings.</p>
appTitle	<p>Specifies the text that appears on the browser tab when using the Connect web app.</p>

Property	Description
availableLanguages	<p>Restricts which of the supported languages and additional languages are available to users. When this option is specified, if a user's browser is set to use a language not included in the <code>availableLanguages</code> list, the default <code>en.json</code> will be used. This option can be used if, for example, administrators edit their language files to use specific terminology and want to limit the number of language files they need to maintain.</p> <p>For example:</p> <pre>"availableLanguages": ["en", "af"]</pre> <p>When this optional setting is not included, all supported and additional languages are available to users.</p>
backgroundColor	Specifies the landing page's background color if no background image is specified
brandName	<p>Specifies the text that appears under the jumbotron image on the landing page, and can be used as a variable in the translations files, for example:</p> <pre>"header": "Welcome to {{brandName}}"</pre> <p>You can edit the rest of the text on the jumbotron using the translation file — see Text on the landing page "jumbotron".</p>
colorPalette	Specifies the range of colors used for interactive elements of the user interface, such as buttons and text
defaultUserConfig	<p>Specifies the defaults to be used for the options that are configurable by the user, including the image used when background replacement has been enabled.</p> <p>For full details, see Overriding the default user-configurable settings.</p>
overlay	Determines how the background image is made less prominent, either by darkening or lightening it. Options are <code>dark</code> or <code>light</code> .
overlayOpacity	Determines the degree to which the background image is darkened or lightened. This must be a decimal number between <code>0.1</code> and <code>0.9</code> .
plugins	Specifies which plugins are available. For more information on implementing plugins, see Creating and deploying Connect app Plugins and the Pexip Developer Portal .
favicon	Specifies the image file to be used for the favicon, using the format:
*	<pre>"favicon": { "href": "./favicon.webp" },</pre> <p>The actual file name of does not matter.</p> <p>Supported formats are <code>.ICO</code>, <code>.SVG</code>, <code>.WEBP</code>, and <code>.PNG</code>.</p> <p> This option cannot be configured via the branding portal.</p>

Property	Description																		
customStepConfig	Enables and specifies details of an additional step in the join flow that can be used to display a card with additional information, and an optional confirmation, to users.																		
*	<p>The card is made up of:</p> <ul style="list-style-type: none">fixed elements such as a title, "Next" button, and checkbox; the text of these can be configured via the language files. For more information, see Text of the custom step.an iframe, within which the content specified by "source" is displayed. For more information, see Body of the custom step. <p>The format is:</p> <pre>"customStepConfig": { "active": true, "subTitle": true, "width": "80%", "height": "80%", "mobileHeight": "100%", "mobileWidth": "100%", "source": { "default": "../index_default.html", "en": "../index_en.html", "de": "../index_de.html" }, "confirmation": "checkbox", "mandatory": true }</pre> <p>where:</p> <table><tr><td>active</td><td>is <code>true</code> to enable the custom step. The default is <code>false</code>.</td></tr><tr><td>subTitle</td><td>is <code>true</code> to enable a subtitle (in addition to the title) for the card that is displayed. The default is <code>false</code>.</td></tr><tr><td>width</td><td>specifies the width of the card.</td></tr><tr><td>height</td><td>specifies the height of the card.</td></tr><tr><td>mobileHeight</td><td>specifies the height of the card when shown on mobile devices.</td></tr><tr><td>mobileWidth</td><td>specifies the width of the card when shown on mobile devices.</td></tr><tr><td>source</td><td><p>specifies the path of the content that appears within the card. The content can be an HTML file, text, an image, a video, or anything else that can be displayed inside an iframe. For more details and examples, see Body of the custom step.</p><p> To change the text of the elements outside of the iframe, such as the title, subtitle (if enabled), "Next" button, or checkbox label, see Text of the custom step.</p><p>Paths must start with <code>../</code> and are relative to the branding folder.</p><p>You must define a <code>default</code> path, and you can optionally define additional paths on a per-language basis. The default path will be used unless the user's browser's display language is set to one of the defined languages, in which case the path specified by that language will be used.</p><p>If this value is not set or does not start with <code>../</code>, the card is deactivated.</p></td></tr><tr><td>confirmation</td><td><p>is <code>checkbox</code> if the user must confirm the step's content by ticking a checkbox. Doing so will also enable the Next button.</p><p>If not set or set to <code>none</code>, no active confirmation is required.</p></td></tr><tr><td>mandatory</td><td>is <code>true</code> if the card should be shown even if direct join is used.</td></tr></table>	active	is <code>true</code> to enable the custom step. The default is <code>false</code> .	subTitle	is <code>true</code> to enable a subtitle (in addition to the title) for the card that is displayed. The default is <code>false</code> .	width	specifies the width of the card.	height	specifies the height of the card.	mobileHeight	specifies the height of the card when shown on mobile devices.	mobileWidth	specifies the width of the card when shown on mobile devices.	source	<p>specifies the path of the content that appears within the card. The content can be an HTML file, text, an image, a video, or anything else that can be displayed inside an iframe. For more details and examples, see Body of the custom step.</p> <p> To change the text of the elements outside of the iframe, such as the title, subtitle (if enabled), "Next" button, or checkbox label, see Text of the custom step.</p> <p>Paths must start with <code>../</code> and are relative to the branding folder.</p> <p>You must define a <code>default</code> path, and you can optionally define additional paths on a per-language basis. The default path will be used unless the user's browser's display language is set to one of the defined languages, in which case the path specified by that language will be used.</p> <p>If this value is not set or does not start with <code>../</code>, the card is deactivated.</p>	confirmation	<p>is <code>checkbox</code> if the user must confirm the step's content by ticking a checkbox. Doing so will also enable the Next button.</p> <p>If not set or set to <code>none</code>, no active confirmation is required.</p>	mandatory	is <code>true</code> if the card should be shown even if direct join is used.
active	is <code>true</code> to enable the custom step. The default is <code>false</code> .																		
subTitle	is <code>true</code> to enable a subtitle (in addition to the title) for the card that is displayed. The default is <code>false</code> .																		
width	specifies the width of the card.																		
height	specifies the height of the card.																		
mobileHeight	specifies the height of the card when shown on mobile devices.																		
mobileWidth	specifies the width of the card when shown on mobile devices.																		
source	<p>specifies the path of the content that appears within the card. The content can be an HTML file, text, an image, a video, or anything else that can be displayed inside an iframe. For more details and examples, see Body of the custom step.</p> <p> To change the text of the elements outside of the iframe, such as the title, subtitle (if enabled), "Next" button, or checkbox label, see Text of the custom step.</p> <p>Paths must start with <code>../</code> and are relative to the branding folder.</p> <p>You must define a <code>default</code> path, and you can optionally define additional paths on a per-language basis. The default path will be used unless the user's browser's display language is set to one of the defined languages, in which case the path specified by that language will be used.</p> <p>If this value is not set or does not start with <code>../</code>, the card is deactivated.</p>																		
confirmation	<p>is <code>checkbox</code> if the user must confirm the step's content by ticking a checkbox. Doing so will also enable the Next button.</p> <p>If not set or set to <code>none</code>, no active confirmation is required.</p>																		
mandatory	is <code>true</code> if the card should be shown even if direct join is used.																		


* This is new in version 34.

An example `manifest.json` file that you can copy and edit for your own use is given below:

```
{
  "version": 0,
  "meta": {
    "name": "DEFAULT",
    "brandVersion": "n/a"
  },
  "brandName": "Pexip Connect for Web",
  "backgroundColor": "#181818",
  "colorPalette": [
    "#E9F2FB",
    "#A3C8EE",
    "#5C9FE1",
    "#2475C5",
    "#174B7F",
    "#0A2138",
    "#0A2035",
    "#091E33",
    "#091D30",
    "#081B2E",
    "#08192B"
  ],
  "overlay": "light",
  "overlayOpacity": 0.9,
  "images": {
    "background": "./background.jpg",
    "logo": "./logo.svg",
    "jumbotron": "./jumbotron.jpg"
  },
  "translations": {
    "en": "./en.json"
  },
  "defaultUserConfig": {
    "backgroundBlurAmount": 16,
    "isAudioInputMuted": true,
    "isVideoInputMuted": true,
    "showSelfView": true
  },
  "applicationConfig": {
    "disconnectDestination": "https://www.meet.pexample.com",
    "bgImageAssets": [".branding/bg_light.png", ".branding/bg_dark.jpg"],
    "showLiveCaptionsFeature": false,
    "bandwidths": [
      "576",
      "1264",
      "2464",
      "6144"
    ],
    "termsAndConditions": {
      "en": "https://www.pexample.com/terms"
    }
  },
  "appTitle": "Pexip",
  "plugins": [
    { "src": "<path-to-your-plugin>/index.html" }
  ]
}
```

Overriding the default user-configurable settings

The `defaultUserConfig` section of the manifest file specifies the defaults to be used for the options that are configurable by the user. Options include:

<code>bgImageUrl</code>	The URL of the default image used when background replacement has been enabled by the user.
	 This has now been superseded by bgImageAssets .

backgroundBlurAmount	<p>Controls the level of blurring when background blur has been enabled. Valid values are between 0 and 100, with a default of 16.</p> <p>i As the value increases, the level of local processing increases but the difference in blur becomes less noticeable. For this reason we do not recommend setting this value too high.</p>
callType	<p>Determines whether the participant will be able to send or receive audio or video.</p> <p>i In all cases, the participant can still access the conference controls and chat, and send and receive presentations.</p> <p>Options are:</p> <p>0: ("none") to join as a presentation and control-only participant, i.e. the user will not send or receive any audio or video.</p> <p>6: ("audioonly") to join as an audio-only participant, i.e. send and receive audio but not send or receive video.</p> <p>4: ("audiorecvonly") to receive but not send audio, and not send or receive video.</p> <p>2: ("audiosendonly") to send but not receive audio, and not send or receive video.</p> <p>24: ("videoonly") to send and receive video, but not send or receive audio.</p> <p>16: ("videorecvonly") to receive but not send video, and not send or receive audio.</p> <p>8: ("videosendonly") to send but not receive video, and not send or receive audio.</p> <p>10: ("audiovideosendonly") to send audio and video, but not receive audio or video.</p> <p>20: ("audiovideorecvonly") to receive audio and video, but not send audio or video.</p> <p>30: ("video") to join as a full (send and receive) audio and video participant.</p> <p>If this value is not set, or is set to a value other than one listed above, the default 30 (full send and receive video and audio) is used.</p>
isAudioInputMuted	Determines whether the user's microphone is muted (<i>true</i>) or not muted (<i>false</i>) when they first join a meeting.
isVideoInputMuted	Determines whether the user's camera is disabled (<i>true</i>) or enabled (<i>false</i>) when they first join a meeting.
segmentationEffects	<p>Determines which, if any, background effect is applied. Options are:</p> <p><i>none</i>: no effects are applied.</p> <p><i>blur</i>: background blur is enabled.</p> <p><i>overlay</i>: background replacement is enabled.</p>
showSelfView	Determines whether the user's self view is shown (<i>true</i>) or hidden (<i>false</i>) by default.
preferPresInMix	Determines whether the Prefer presentation in mix option is enabled (<i>true</i>) or disabled (<i>false</i>) by default.
*	

* This is new in version 34.

Overriding the default application settings

The `applicationConfig` section of the manifest file specifies any overrides of the application's default settings. Options include:

audioProcessing	<p>Determines whether the advanced audio features are enabled. These features include the detection of audio when a microphone is muted (which triggers the "Trying to speak? Your microphone is muted" in-call notification, and the notification when testing a muted microphone before joining a call), and the software-based noise suppression feature.</p> <p>This option is enabled by default; to disable these features, set</p> <pre>"audioProcessing": false</pre>
bandwidths	<p>An array that specifies the bandwidths to be used for the <i>Low</i>, <i>Medium</i>, <i>High</i>, and <i>Very High</i> options when a user selects the bandwidth to be used for their call.</p>
disconnectDestination	<p>Specifies the URL to which users are directed when a call is completed (instead of returning to the app home page).</p>
node	<p>(Only required if you are hosting the web app on an external server)</p> <p>The FQDN of the Conferencing Node to which requests should be sent. You can only specify a single address.</p>
bglImageAssets	<p>Specifies one or more background images that users choose from when enabling the background replacement feature.</p>
showLiveCaptionsFeature	<p>Determines whether the option to enable live captions is available to users. This option is enabled by default where supported; to hide this option from users, set</p> <pre>"showLiveCaptionsFeature": false</pre>
showTermsAndConditionsLink*	<p>Determines whether the "terms and conditions of use" text (as defined in the language file by "next-terms-and-conditions") is shown on the user name step of the join flow. This option is enabled by default; to disable it set</p> <pre>"showTermsAndConditionsLink": false</pre>
termsAndConditions	<p>Specifies the URL to which the "terms and conditions" text on the user name step of the join flow is directed. By default, this is https://www.pexip.com/terms. You can change this for all users, or you can specify different URLs depending on the user's browser language. For example:</p> <pre>"termsAndConditions": { "en": "https://www.pexample.com/terms", "es": "https://www.pexample.es/terminos", "fr": "https://www.pexample.fr/termes" }</pre>
videoProcessing	<p>Determines whether the option to blur their background is available to users. This option is enabled by default where supported; to hide this option from users, set <code>"videoProcessing": false</code></p>
shouldMaskConference	<p>Determines whether the meeting name remains visible in the URL after joining (e.g. <code>.../meet.alice</code>, or is encoded (e.g. <code>.../73dca0e3-8d52-880c-4ca3-a938c88a00af</code>) so that any screenshots of the meeting do not reveal the meeting name.</p> <p>This option is disabled by default; to enable it set</p> <pre>"shouldMaskConference": true</pre>

hiddenFunctionality	Specifies any UI elements that should be hidden, using the element's <code>data-testid</code> .
*	<p>To find the <code>data-testid</code> of an element, right-click on the element and select Inspect. From the panel on the right select the Elements tab. The code will expand to show the definition of the element; from within this find the <code>data-testid</code> and copy the value.</p> <p>Common options include:</p> <ul style="list-style-type: none"> • <code>add-participant</code>: hides the Add participant button • <code>link-mute-all-guests</code>: hides the Mute all Guests button • <code>tab-settings-meeting-layout</code>: hides the Meeting layout tab from the meeting settings options • <code>button-participants</code>: hides the Participants button, therefore preventing users from opening the Participants panel to view or control other participants • <code>button-rejoin</code>: hides the post-call "Rejoin" button <p>Values must be specified within an array.</p> <p>For example:</p> <pre>"hiddenFunctionality": ["add-participant", "link-mute-all-guests"]</pre>
handleOAuthRedirects	Enables support for features (such as plugins) that require authentication to a third party using an OAuth 2.0 / OpenID Connect flow, with a redirect destination of <code>webapp3/oauth-redirect</code> .
*	<p>This option is disabled by default; to enable it set:</p> <pre>"handleOAuthRedirects": true</pre>

* This is new in version 34.

Images

Landing page background image

You can optionally include a **background** image file to display behind all other elements on the landing page. To do this, save the file you wish to use in the `webapp3/branding` folder and then edit the `manifest.json` to specify its path using the `"background"` value within the `"images"` property.

This file must meet the following requirements:

- resolution: 1920x1080
- size: 500-600 kB
- format: .JPG, .JPEG, .PNG or .WEBP

Jumbotron image

The **jumbotron** image file is an optional image/logo that can be displayed at the top of the "jumbotron" — the area on the left side of the landing page. To include this image, save the file you wish to use in the `webapp3/branding` folder and then edit the `manifest.json` to specify its path using the `"jumbotron"` value within the `"images"` property.

This file must meet the following requirements:

- resolution: 640x360
- size: 70-140 kB
- format: .JPG, .JPEG, .PNG or .WEBP

Logo

The **logo** file is an optional image/logo that can be displayed at the bottom of the "jumbotron" — the area on the left side of the landing page. To include this image, save the file you wish to use in the `webapp3/branding` folder and then edit the `manifest.json` to specify its path using the `"logo"` value within the `"images"` property.

This file must meet the following requirements:

- transparent
- maximum displayed width: 168px
- maximum displayed height: 80px
- format: we recommend .SVG

Participant background replacement images

The background replacement feature allows a video participant to replace their background with one or more selected images — either from a set of default images that you specify, or an image they upload themselves.

To include default background replacement images for all users in your deployment, save the files you wish to use in the **webapp3/branding** folder (or a subfolder that you create) and then edit the **manifest.json** to specify their paths using the **"bgImageAssets"** value within the [applicationConfig](#) section.

These files must meet the following requirements:

- images must be at least 432 pixels high, and 768 pixels wide
- supported formats are .JPG, .JPEG, .PNG and .WEBP
- we recommend you use high-definition images

i Earlier versions of Pexip Infinity supported a single default background replacement image specified using the **"bgImageUrl"** value within the [defaultUserConfig](#) section of the manifest file. To ensure backwards compatibility this option is still supported, but we recommend for new customizations that you use the **"bgImageAssets"** option instead. If both **"bgImageAssets"** and **"bgImageUrl"** are specified, the former will be used and the latter ignored.

Languages/text used in labels and messages

Connect Webapp3 supports over 20 of the most popular languages. If the user's browser is set to use any one of these supported languages, Connect Webapp3 will use that automatically instead of the default English. Alternatively, users can view Connect Webapp3 in any of the supported languages by appending the appropriate language code to the end of the URL.

See [Language support](#) for a complete list of the languages currently supported.

You can change any of the text that is displayed in the application (either in the default English or any of the supported languages), and you can add additional languages. Each language references a .json dictionary file containing a list of token and text string pairs.

- To [change the text of a supported language](#) you create a new .json file for that language that contains the strings you wish to change.
- To [add a new language](#) you create a new .json file for that language.

All .json files must be placed in the **webapp3/branding** subfolder and referenced in the [translations](#) block of the manifest file.

Editing a language file

Each language file contains a list of token and text string pairs. The token remains the same in all language files and the associated string contains the text that is displayed in the app when that language file is used.

The full set of token and text string pairs for each supported language is available from https://docs.pexip.com/files/webapp3_languages/v34/<language>.json where **<language>** is the ISO standard code for the language. For example, the link for the default English file is https://docs.pexip.com/files/webapp3_languages/v34/en.json. You can download this file to use as the basis of your new or edited language files (you may need to right-click and select **Save link as...**).

i These files reflect beta software — all content is subject to change until release.

To change the text that is displayed, simply search in the language file for the text that needs to be changed, edit the text string, and save your changes back to the same file. Do not change the token names.

For example, in the [default en.json](#) the **"Please enter a display name so other people know who's in the meeting"** string is located in the **"username"** block and is associated with the **"usage-purpose"** token:

```
"username": {  
  ...  
  "usage-purpose": "Please enter a display name so other people know who's in the meeting",  
}
```

```
...
}
```

The strings are generally grouped together according to where or when they are displayed. For example, all tokens in the "add-participant" block refer to strings that appear when you are adding a participant to the meeting.

The edited language file does not have to contain the complete set of tokens / text strings. You only need to include the token and text string pairs that you want to be different from the default strings for that language.

Changing the default (English) strings

You can use your own alternative English strings instead of the default strings. To do this, download the default `en.json` file from https://docs.pexip.com/files/webapp3_languages/v34/en.json (you may need to right-click and select **Save link as...**), and edit the strings as required. You only need to include the token and text string pairs that you want to be different from the default strings. Do not change the token names.

i These files reflect beta software — all content is subject to change until release.

Then save the `en.json` file in the `webapp3/branding` folder and reference it in the `translations` block of the manifest, in the same way as you would when [Adding additional languages](#).

Changing translations for other supported languages

The Connect Webapp3 is available in a [selection of other languages](#). We've provided the translations for you, but you can use your own alternative strings for each language instead. To do this:

1. Download the relevant language file from https://docs.pexip.com/files/webapp3_languages/v34/<language>.json where `<language>` is the ISO standard code for the language, for example `de` for German.

i These files reflect beta software — all content is subject to change until release.
2. Edit the strings as required. You only need to include the token and text string pairs that you want to be different from the default strings for that language. Do not change the token names.
3. Save the file with the appropriate file name (for example, `de.json` for any German changes) in the `webapp3/branding` folder and reference it in the `translations` block of the manifest, in the same way as you would when [Adding additional languages](#).

Adding additional languages

The Connect Webapp3 is available in English and a [selection of other languages](#). You can add additional languages as follows:

1. Create a new `<language>.json` file in the `webapp3/branding` folder:
 - a. Download the default [en.json file](#) as a basis for the new language.
 - b. Rename the new file as appropriate for your new language, for example `af.json` for Afrikaans.

We recommend using the ISO standard language codes/locales for the filename as this allows that language file to be used automatically if its name matches the browser's default language. Always use `.json` as the filename extension.

The app also supports different language cultures via branding. For example, it can distinguish between French (`fr-FR.json`), French Canadian (`fr-CA.json`) and French Belgian (`fr-BE.json`).
 - c. Edit the text strings as appropriate for the new language, leaving the token names unchanged.
2. Save the file in the `webapp3/branding` folder.
3. Edit the `manifest.json` file to add a reference to the new language by updating the `translations` block to include the new language file, for example:

```
"translations": {
  "en": "../en.json",
  "af": "../af.json"
}
```

Now, if a user's browser is set to use Afrikaans, they will see the text strings from the `af.json` file. Any strings not specified in `af.json` will be shown in the default English.

Tags

Some of the text strings contain tags, for example `<0>...</0>`, `<1>...</1>`, etc. These tags are used by the app for formatting purposes, for example by changing the text to bold, by adding a hyperlink to the text, or by inserting the text into a button. You must retain the tags, but you can translate any text within the tags into your chosen language.

You cannot create your own tags.

Variable substitutions

Some strings contain variable substitutions, for example:

```
"remove-user-confirmation": "Are you sure you want to forcibly eject {{userName}} from this meeting?"
```

This message appears when a user disconnects a participant. In this case, the application automatically substitutes `{{userName}}` with the participant's actual name as shown in the participant list. Variables always take the format `{{<variable name>}}`. You cannot create your own variables.

Error messages

There is a list of error message in the language file. These messages typically relate to connectivity issues between the Conferencing Node and Connect app, or to conference activities.

The token name is in the format `"code[#pex###]"`, which is used as a common reference for the message regardless of the language used in the message string, for example:

```
"code[#pex121]": "A host ended the meeting"
```

You can change these messages in the same way as you can change the other messages — edit the display text part only; do not change the `"code[#pex###]"` token name part.

Text on the landing page "jumbotron"

The `"infocard"` block of the language file specifies the text that appears underneath the "jumbotron" image on the left side of the landing page. By default this uses the `brandName` variable configured in the [manifest.json](#) file. You can edit both the header and body text:

```
"infocard": {
  "meeting": {
    "body": "A secure video communications platform",
    "header": "Welcome to\n {{brandName}}"
  }
},
```

Text of the custom step

The `"custom-step"` block of the language file specifies the text that appears in the fixed elements (e.g. title, button, checkbox label) of the card displayed when the optional additional customized joining step is enabled.

- i** To use a custom joining step and to set its contents you must first include the optional [customStepConfig](#) block in the [manifest.json](#) file, and then include in the `webapp3/branding` folder the files that are referenced within the [source](#) block, which supply the [body of the custom step](#). You can also change the wording of the other elements of the custom step via the `"custom-step"` block of the language file (as described in this section).

The example below shows an edited `"custom-step"` block used to produce the example shown in [Body of the custom step](#):

```
"custom-step": {
  "checkbox": {
    "aria-label": "Click to agree",
    "caption": "I agree"
  },
  "content-frame": {
    "title": "Content"
  },
  "next": "Proceed",
  "subtitle": "You must read and accept the conditions below in order to join the meeting.",
  "title": "Terms and conditions of use"
},
```

Body of the custom step

- i** To use a custom joining step and to set its contents you must first include the optional [customStepConfig](#) block in the [manifest.json](#) file, and then include in the `webapp3/branding` folder the files that are referenced within the [source](#) block, which supply the [body of the custom step](#) (as described in this section). You can also change the wording of the other elements of the custom step via the `"custom-step"` block of the language file.

The content that is shown within the body of the custom step can be an HTML file, text, an image, a video, or anything else that can be displayed inside an iframe.

You can define different content files for different languages — for more information, see [source](#).

The example below is an HTML file that displays pre-meeting terms and conditions as a set of numbered items. This can be used in conjunction with a checkbox to require that the user confirms they have read and agree to these conditions before they can proceed:

```
<!DOCTYPE html>

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-7">
  <style>
    ::-webkit-scrollbar {
      width: 8px;
      height: 8px;
      overflow: visible;
    }

    ::-webkit-scrollbar-thumb {
      min-height: 18px;
      border-width: 1px 1px 1px 8px;
      border-radius: 20px;
      background-color: #a5b8ca;
      background-clip: padding-box;
    }

    ::-webkit-scrollbar-button {
      width: 0;
      height: 0;
    }

    ::-webkit-slider-thumb element .b-w::-webkit-scrollbar-track {
      border: solid transparent;
      border-width: 0 0 0 6px;
      background-clip: padding-box;
    }

    ::-webkit-scrollbar-track {
      border: solid transparent;
      border-width: 0 0 0 6px;
      background-clip: padding-box;
    }

    ::-webkit-scrollbar-corner {
      background: 0 0;
    }

    ::-webkit-scrollbar-track-piece {
      background: transparent;
    }
  }
  body {
    color: #1e3a54;
    font-size: 14px;
  }
</style>
</head>
<body>
  style="
    font-family: Inter, -apple-system, BlinkMacSystemFont, Segoe UI,
      Roboto, Oxygen, Ubuntu, Cantarell, Fira Sans, Droid Sans,
      Helvetica Neue, sans-serif;
  "
  >
  <ol>
    <li>
      This is the first condition.
    </li>
  <br />
  <li>
```

```
    This is the second condition.  
</li>  
<br />  
<li>  
    This is the third condition.  
</li>  
</ol>  
</body>  
</html>
```

When the example above is used in conjunction with the example "custom-step" block of the language file given in [Text of the custom step](#), the result is the following:

Terms and conditions of use

You must read and accept the conditions below in order to join the meeting.

1. This is the first condition.
2. This is the second condition.
3. This is the third condition.

☐ I agree

Proceed

Uploading the branding package

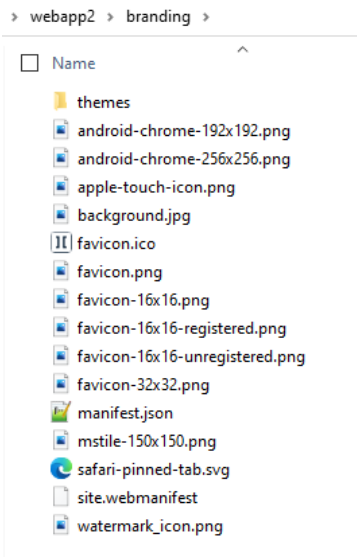
When you have prepared your files, ZIP the folder containing the **webapp3/branding** folder. It is this zipped folder that you then upload to the Management Node.

Manually customizing Connect Webapp2

This section describes the Connect app files for the previous web app, "Connect Webapp2", that can be manually customized.

- i** When editing the configuration files, you must use a text editor that does not apply "smart quotes" or make any automatic text changes, as the files are sensitive to correct formatting. Use a code editor or simple file editor instead of word processing software.
- i** The customizations you make here must be applied manually to the Connect web app. You can use Pexip Infinity's [provisioning](#) features to apply the same branding to the Connect desktop app.

If you have downloaded the default branding from the Management Node, your ZIP file when unpacked will contain the following files within a **webapp2/branding** folder, as shown below:



i Note that if you have downloaded a branding package from the Webapp2 branding portal, or from a Pexip Infinity deployment running v30 or earlier, you will need to edit the structure of the unzipped files and folders to match that shown above. For more information, see [Using the Webapp2 branding portal](#).

Manually creating a client branding package for upload via the Management Node

In addition to the files/subfolders included in the default client branding as shown above, there are some extra files/subfolders that can be included for other customization purposes — some of which may be generated when using the Pexip branding portal.

The complete set of files that can be included in a branding ZIP package for upload via the Management Node, or manually copied into the **custom_configuration** directory when hosting the app on a different server, are summarized below, and are then explained in more detail in the subsequent sections:

i When creating your branding package ZIP file, ensure that all of your files/subfolders are contained in a **webapp2/branding** folder.

- The **manifest.json** file controls which customized settings take effect.
 - i** The manifest.json file is the only mandatory file. Your customizations will not take effect if this file is not included.
- The **settings.json** file* contains the application and default user settings.
- A set of **favicon** files* to associate with the app.
- A **site.webmanifest** file* for websites installed on the homescreen of a device.
- The **watermark_icon.png** file is a transparent image used for applying a watermark to the self-view image displayed on the **Home** page. By default, this image is the same as the base theme's watermark that is applied to the main speaker video in a conference, and is a white Pexip logo with 40% transparency.
- The **background.jpg** file* is an optional image/logo file that can be displayed on a landing page. It can also include some customizable text.
- A **languages** subfolder* containing the text customizations for each language. Currently, English is the only language that is available by default, but you can add more languages if required.
- A **plugins** subfolder* which in turn contains subfolders for each plugin that has been deployed.
- A **themes** subfolder* containing a **styles.css** file that specifies style overrides, such as different colors for selected elements.

* This file/folder does not initially exist in the default branding download.

Application manifest (manifest.json)

The **manifest.json** file controls which customized settings take effect. You must include the **manifest.json** file in the **webapp2/branding** folder. The **manifest.json** file should look like this, with a **brandingID** value and the other settings set to true/false

as appropriate:

```
{
  "brandingID": "1521829718679",
  "isSettingsBranded": true,
  "isWatermarkBranded": false,
  "isStylesBranded": false,
  "isCustomBackgroundBranded": false
}
```

- The **brandingID** value has to be different from the previously uploaded value for the new settings to take effect (this means, for example, that you can easily revert to a branding you created earlier just by re-uploading that ZIP file). In the **manifest.json** file generated by the Pexip branding portal, the **brandingID** is set to a timestamp, but any type of numerical value is valid. You can open your browser console and type `Date.now()` to get a current timestamp value.
- Set **isSettingsBranded** to **true** if you include a **settings.json** file.
- Set **isWatermarkBranded** to **true** if you are supplying a new **watermark_icon.png** file.
- Set **isStylesBranded** to **true** if you have provided CSS overrides in a **themes** folder with a **styles.css** file.
- Set **isCustomBackgroundBranded** to **true** if you are supplying a **background.jpg** file.

Application and default user settings (settings.json)

The **settings.json** file is a JSON dictionary that contains the application and default user settings.

Note that this file is not included in the default branding ZIP file that you can download from the Management Node, but it is included in ZIP files generated by the Pexip branding portal when the relevant settings have been customized. When used, the **settings.json** file must be placed in the **webapp2/branding** folder.

The following items in the **applicationSettings** block can be configured:

Setting	Description
dialOutProtocol	This setting is deprecated. Since version 25 all calls are placed via the "auto" protocol option, and therefore suitable Call Routing Rules must be configured.
languages	<p>Controls the set of languages available to the user. When languages are configured, users get an additional option on the Settings page that allows them to choose their preferred language. For more information, see Languages/text used in labels and messages (en.json) and additional languages.</p> <p>Default: []</p> <p>English (en locale) is used by default when no languages are specified.</p>
bandwidths	<p>Controls the set of bandwidth options available to the user.</p> <p>Default: ["256", "576", "1264", "2464", "6144"]</p> <p>You can specify other values if required. Note that the actual bandwidth limit for any given call will be the lower of either the value set within the app, or the value set for the service (VMR) being called (which if not set specifically is the global bandwidth).</p>
turnServer	<p>This setting provisions the Connect app with a TURN server that it can offer as a relay candidate in ICE negotiations.</p> <p>Default: null</p> <p>To configure a TURN server you must specify the TURN server address and credentials (note that these credentials are not encrypted within the settings file), for example:</p> <pre>"turnServer": { "urls": "turn:turn.example.com:443?transport=tcp", "username": "user", "credential": "pass" }</pre>

Setting	Description
serverAddress	<p>In most deployments you will not need to customize this setting. You should only change this setting if:</p> <ul style="list-style-type: none">• you are hosting the web app on an external web server (rather than on a Conferencing Node or reverse proxy)• you want to direct the desktop client to use a specific Conferencing Node. <p>To configure a specific address, change the serverAddress variable to refer to the relevant Conferencing Node FQDN. You can only specify a single address, for example:</p> <pre>"serverAddress": "conferencingnode1.example.com"</pre> <p>Note that the TLS certificate installed on the server needs to be trusted by the client system (as the client system will not display any certificate trust security alerts).</p> <p>Default: null</p>
registrationEnabled	<p>This setting only applies to the Connect desktop app and controls whether the client is allowed to register to a Conferencing Node.</p> <p>Default: true</p>
disconnectDestination	<p>Defines a URL to redirect the user to when a call is completed (instead of returning to the app home page), for example:</p> <pre>"disconnectDestination": "https://somewhere.example.com"</pre> <p>Note that this setting is not configurable via the branding portal.</p>
defaultToMuted	<p>Controls whether the user's microphone is locally muted on the home page when the app is first launched.</p> <p>Regardless of whether this setting is true or false, the user can still mute and unmute the microphone before joining or during a meeting.</p> <p>Default: false</p>
micSampling	<p>If enabled, the user will see a message in selfview saying "You're muted" if their microphone detects sound while it is locally muted.</p> <p>Note that this setting is not configurable via the branding portal. If enabled it may cause audio quality issues if the client device has high CPU usage.</p> <p>Default: false</p>
showTimeline	<p>Controls whether a timeline that gives a visual overview of the events during the course of the call is shown at the bottom of the screen.</p> <p>Note that this setting is not configurable via the branding portal. If enabled it increases CPU utilization on the client device.</p> <p>Default: false</p>
raiseHandInVMR	<p>Controls whether the raise hand feature is enabled in Virtual Meeting Rooms (in addition to Virtual Auditoriums).</p> <p>Default: false</p>
presentationInMix	<p>This setting controls whether the client is allowed to receive presentation in the layout mix. If false it will always receive presentation content as a separate stream only.</p> <p>Default: true</p>
sortAttendeesAlphabetically	<p>Controls whether the participant list is sorted alphabetically or not (in which case it is sorted in the order that the participants joined).</p> <p>Default: false</p>

Setting	Description
backgroundEffects	Controls whether the option to enable background effects is visible in the app. Default: true

The `defaultUserSettings` block in the `settings.json` file contains the default user settings that are applied to first-time users, or whenever a new customization package is applied to the client. The application subsequently remembers the user's last-used settings. The configurable options are:

Setting	Description
language	Controls which of the languages from the list of <code>languages</code> in the <code>applicationSettings</code> block is used by default. However, the browser's default language is used automatically, and supersedes this setting, if it is available in the <code>languages</code> list. Default: "en"
screenshareFrameRate	Controls the frame rate (in fps) for screen sharing. Default: 2
promptDisconnect	Controls whether to ask the user for confirmation before disconnecting from a conference. Default: true
viewFullMotionPresentation	Controls whether the user views presentations as full motion video or as still images by default, when a presentation is started by another participant. Users can switch between both viewing modes after a presentation has started. The valid values are true (full motion) and false (still images). Default: true
sendStats	Controls whether or not anonymous Connect app usage statistics are sent to Pexip. The valid values are true and false. Note that the <code>Automatically send deployment and usage statistics to Pexip</code> global setting on the Management Node must also be enabled in order to allow the Connect app to send usage statistics. Default: true
bandwidth	The default bandwidth used for video and audio. The value specified here must match one of the values configured in the <code>bandwidths</code> block above it. Default: 1264
showConferenceSidebar	Controls whether the side panel is initially hidden or open when in a call. The user can still use the in-call controls to show or hide the side panel, and then this is remembered for the next call. Default: false
highContrast	When this option is enabled, there is a higher contrast between foreground and background elements of the user interface, making them more legible. Default: false
startInBackground	The Connect app will always start automatically when the device it is installed on starts. This option allows you to select whether it starts minimized (in the background), or maximized (in the foreground). This setting has no effect on the web app. Default: false


Setting	Description
playRingtone	Plays the default ringtone when the client receives an incoming call. Default: true
videoConstraints	Controls whether video constraint settings are applied to prevent lower quality / wrong ratio on cameras. In some environments, where you may experience low frame rates with some cameras, it may be necessary to disable this setting. Note that this setting is not configurable via the branding portal. Default: true
devicePairing	This setting toggles the availability of the option to pair with an alternative device when placing a call. Default: true for web app and desktop clients, and is not supported for the mobile clients.
enableBackgroundEffects	Controls whether background effects are enabled. Default: true
activeEffect	(Valid for Connect desktop app only) Controls which background effect is in use when enableBackgroundEffects is enabled. Options are blur or replacement. Default: blur
enableFecc	This setting controls whether the currently selected camera can be controlled (if it supports pan/tilt/zoom, or zoom-only) by another participant. Default: false


The **plugins** block controls which plugins are available. For more information on implementing plugins, see [Creating and deploying Connect app Plugins](#) and [the Pexip Developer Portal](#).

Here is an example `settings.json` file that shows the required structure:

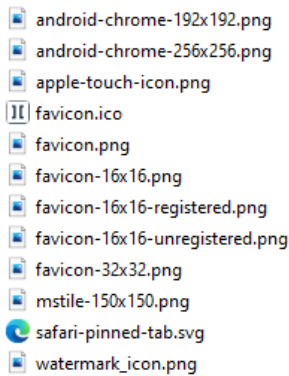
```
{
  "applicationSettings": {
    "dialOutProtocol": "sip",
    "languages": [],
    "bandwidths": [
      "576",
      "1264",
      "2464"
    ],
    "registrationEnabled": true,
    "defaultToMuted": true
  },
  "defaultUserSettings": {
    "bandwidth": "1264",
    "playRingTone": false
  },
  "plugins": []
}
```

Custom favicon files

When you use customized branding in your web app, you must specify the favicon files you want to use, even if you want to continue using the default Pexip favicon ()

 If you do not include the favicon files in your branding package then the web app will not use any favicons at all.

The complete set of expected favicon files is shown below. You can use the Pexip branding portal to automatically generate these files (with the default Pexip favicon image).



Many favicon generator utilities are available on the internet.

A site.webmanifest file

The `site.webmanifest` file is an optional file. A web app manifest provides information about an application (such as its name, author, icon, and description) in a JSON text file for websites installed on the homescreen of a device. See <https://developer.mozilla.org/en-US/docs/Web/Manifest> for more information about the contents of the file.

You can use the Pexip branding portal to automatically generate a `site.webmanifest` with some suggested settings.

Custom background image/logo (background.jpg) and welcome text

You can optionally include a `background.jpg` file and some accompanying welcome text to display to the user on a landing page whenever they use the app. (By default, Pexip Infinity does not supply a background file and the landing page is not displayed.)

To enable the landing page:

1. Include a file called `background.jpg` in the `webapp2/branding` folder (at the same level as the `watermark_icon.png` file). The background image covers the whole browser window and scales if the browser window resizes. We recommend a JPEG image that is approximately 2000x1400 pixels.
2. Set `isCustomBackgroundBranded` to `true` in the `manifest.json` file.
3. Customize the accompanying welcome text as required in the language file. The text strings used on the landing page are in the "ONBOARDING_SCREEN" group of tokens. You can use HTML markup (UTF-8 characters only).

Languages/text used in labels and messages (en.json) and additional languages

All of the text that is displayed in the application can be changed, and additional languages can be added. Note that you can also use the Pexip branding portal (<https://webapp2-branding.pexip.io/>) to set up new languages.

Any files containing customized text strings must be placed in the `webapp2/branding/languages` subfolder. The `en.json` (English) file is supplied by default when you download the default branding from the Management Node.

Editing a language file

Each language file contains a list of token and text string pairs. The token remains the same in each language file and the associated string contains the text that is displayed in the app when that language file is used.

Text customizations are simply a matter of changing the text assigned to each token in your language file. To find the string to change, search in the language file for the text that needs to be changed, edit the text, and save your changes back to the same file. Do not change the token names.

For example, the "Enter your name" string is located in the "HOME" block and is associated with the "DISPLAYNAME_PLACEHOLDER" token:

```
"DISPLAYNAME_PLACEHOLDER": "Enter your name"
```

The strings are grouped together according to where or when they are displayed. For example, all tokens in the "SEARCH" block refer to strings that appear when you are entering or selecting the address to call.

The language file does not have to contain the complete set of tokens / text strings. If required, you can only include the token and text string pairs that you want to be different from the default English strings.

Variable substitutions

Some strings contain variable substitutions, for example:

```
"TITLE": "Are you sure you want to disconnect {{name}}?"
```

This message appears when a user disconnects a participant. In this case, the application automatically substitutes `{{name}}` with the participant's actual name as shown in the participant list. Variables always take the format `{{<variable name>}}`. You cannot create your own variables.

Error messages

There is a list of error message in the language file. These messages typically relate to connectivity issues between the Conferencing Node and Connect app, or to conference activities.

The token name is in the format `PEX###`, which is used as a common reference for the message regardless of the language used in the message string, for example:

```
"PEX120": "A Host ended the meeting."
```

You can change these messages in the same way as you can change the other messages — edit the display text part only; do not change the `PEX###` token name part.

Changing the default English (en.json) strings

You can use your own alternative English strings instead of those supplied in the default (`en.json`) file. To modify the English strings:

1. Edit the text strings as required in the `en.json` file in the `webapp2/branding/languages` folder (it is supplied by default when you download the default branding from the Management Node).
2. As with all customizations, remember to update (or create if necessary) a `manifest.json` file. Set a new value for `brandingID` and set `isSettingsBranded` to `true`.
3. Ensure that the `settings.json` file (which you may need to create) contains a `languages` section within the `applicationSettings` block and that it references the `en` locale:

```
{
  "applicationSettings": {
    "languages": [
      {
        "locale": "en",
        "label": "English"
      }
    ]
  }
}
```

4. Zip up your customizations in the `webapp2` folder and upload the package to the Management Node.

Adding more languages

The Connect Webapp2 and Connect desktop app are in English by default, but you can add extra languages. To add a new language:

1. Create a new `<language>.json` file in the `webapp2/branding/languages` folder:
 - a. Copy the existing `en.json` file as a basis for the new language.
 - b. Rename the new file as appropriate for your new language, for example `es.json` for Spanish.

We recommend using the ISO standard language codes/locales for the filename as this allows that language file to be used automatically if its name matches the browser's default language. Always use `.json` as the filename extension.

The app also supports different language cultures via branding. For example, it can distinguish between French (`fr-FR.json`), French Canadian (`fr-CA.json`) and French Belgian (`fr-BE.json`).
 - c. Edit the text strings as appropriate for the new language, leaving the token names unchanged.
2. Edit the `settings.json` file and add a reference to the new language by updating the `languages` setting in the `applicationSettings` block to include the new language file, for example:

```
"languages": [
  {
```

```
    "locale": "en",  
    "label": "English"  
  },  
  {  
    "locale": "es",  
    "label": "Español"  
  }  
]
```

would mean there are two language files available: **en.json** and **es.json**. The **locale** acts as the reference to the name of the json language file, and to the user's language preferences in their browser when deciding which language to use by default.

3. As with all customizations, remember to update (or create if necessary) a [manifest.json](#) file. Set a new value for **brandingID** and set **isSettingsBranded** to **true**.
4. Zip up your customizations in the **webapp2** folder and upload the package to the Management Node.

When languages are configured, Connect app users get an additional option on the **Settings** page that allows them to choose their preferred language.

Changing the default language

When additional languages have been configured, you can set one of those new languages to be the default language for first-time users. To set the default language for first-time users:

1. Edit the **settings.json** file.
2. Edit the **language** item in the **defaultUserSettings** block to refer to the preferred default language locale, for example:

```
"language": "es"
```

The nominated locale must be in the list of **languages** in the **applicationSettings** block, and have an associated language json file.

3. As with all customizations, remember to update (or create if necessary) a [manifest.json](#) file. Set a new value for **brandingID** and set **isSettingsBranded** to **true**.
4. Zip up your customizations in the **webapp2** folder and upload the package to the Management Node.

Plugins

Plugins offer extensions to the core client functionality. Each plugin has its own subfolder of files, and must also be referenced in the **plugins** block of the **settings.json** file. For more information on implementing plugins, see [Creating and deploying Connect app Plugins](#) and [the Pexip Developer Portal](#).

Styles (CSS) overrides

You can include a **themes** subfolder containing a **styles.css** file that specifies style overrides, such as different colors for selected elements.

Note that this folder and file is not included in the default branding ZIP file that you can download from the Management Node, but it is included in ZIP files generated by the Pexip branding portal when the relevant settings have been customized. When used, the **themes** folder must be placed in the **webapp2/branding** folder.

More information

Participant avatars cannot be branded via the web app, but they can be controlled by using external policy or by configuring user records. For full details about how to integrate Pexip Infinity with an external policy server, see [Using external and local policy to control Pexip Infinity behavior](#).

In addition to customizing the appearance of the Connect web app, you can also use themes to change the voice prompts and images provided to participants when they are accessing a Virtual Meeting Room, Virtual Auditorium or Virtual Reception. For more information, see [Customizing video and voice prompts using themes](#).

If any further information on customizing Pexip Infinity is required, please contact your Pexip authorized support representative.

Hosting the web app externally

There may be situations where you want to host the web app on a reverse proxy or external web server instead of on the Pexip Infinity Conferencing Nodes. If you do this and you also want to customize the web app, you will need to manually upload the customized files.

This topic describes how to [download and copy over the web app](#) for hosting externally, how to [apply customized branding](#) to externally-hosted web apps, and how to [upgrade](#) externally-hosted web apps.

Hosting options overview

There are two methods for hosting the customized Connect web app, either:

- on the Conferencing Nodes (via an upload on the Management Node), or
- on an external web server or reverse proxy.

Hosting on the Conferencing Nodes

This is the standard method for applying branding to the Pexip Infinity platform. It involves uploading a branding package to the Management Node and associating the package with a unique web app path. The package is then pushed out to all Conferencing Nodes, from where those customizations are served to all web app users joining using that path.

Hosting on an external web server or reverse proxy

This hosting method involves downloading a copy of the entire web app from the Management Node and uploading it onto an external web server or reverse proxy (e.g. the Pexip Reverse Proxy) and serving it from that server. This method allows you, for example, to host multiple different branding customizations under different URLs on those external web servers or reverse proxies.

The remainder of this topic deals with externally-hosted web apps.

Copying over the Connect web app

To download the Connect web app files and copy them to an external web server or reverse proxy:

1. From the Management Node, go to **Web App > Web App Download**.
You see the **Infinity Connect web app download page**.
2. In the **Download web app** field, use the drop-down menu to select the version of the web app you want to host externally, and select **Download**.
A ZIP file containing the files for the selected web app is downloaded to your local machine.
3. Unzip the files and locate the relevant directory:
 - for Webapp3: **webapp3/<revision>/web/**
 - for Webapp2: **webapp2/<revision>/web/static/dist/web/**
4. Copy the contents of the **web/** directory to the external web server or reverse proxy.

Examples

Assuming that your external web server (<http://pexample.com>) is serving files from **/var/www/html**:

For Webapp3:


1. Download the Webapp3 ZIP file from Pexip Infinity.
2. Unzip the file to your home directory (`unzip $webapp3.zip`).
3. Copy the contents to **/var/www/html** (`cp -r webapp3/$version/web /var/www/html/webapp3`).
4. Go to <http://example.com/webapp3> to use Webapp3.

For Webapp2:

1. Download the Webapp2 ZIP file from Pexip Infinity.
2. Unzip the file to your home directory (`unzip $webapp2.zip`).
3. Copy the contents to **/var/www/html** (`cp -r webapp2/$version/web/static/dist/web /var/www/html/webapp2`).
4. Go to <http://example.com/webapp2> to use Webapp2.

Uploading branding files

When hosting the web app on an external web server or reverse proxy you must manually upload any customized client files to the appropriate directory on the host server. The name of the directory depends on whether the branding applies to Webapp3 or Webapp2.

-  You obtain the files to be uploaded by [downloading](#) them from the Management Node, using the [branding portal](#) to create a branding package, or creating them manually.

Webapp3

For Webapp3, the branding files must be uploaded to a **branding** folder directly under the directory from which the web app is being served. You must create the **branding** folder if it does not already exist.

For example, assuming that your external web server (<http://pexample.com>) is serving Webapp3 from `/var/www/html/webapp3` and you are using a branding package downloaded from Pexip Infinity:

1. Download the Webapp3 branding package from Pexip Infinity.
2. Unzip the files to your home directory (`unzip $branding.zip`).
3. Copy the branding files to the hosted Webapp3 (`cp webapp3/branding /var/www/html/webapp3/`).
4. Go to <http://example.com/webapp3> to use the branded Webapp3.

Webapp2

For Webapp2, the branding files must be uploaded to a **custom_configuration** folder directly under the directory from which the web app is being served. You must create the **custom_configuration** folder if it does not already exist.

Any branding packages downloaded from Pexip Infinity will have the structure **webapp2/branding**, so you must change the name of the **branding** folder to **custom_configuration** when copying it to the external server.

For example, assuming that your external web server (<http://pexample.com>) is serving Webapp2 from `/var/www/html/webapp2` and you are using a branding package downloaded from Pexip Infinity:

1. Download the Webapp2 branding package from Pexip Infinity.
2. Unzip the files to your home directory (`unzip $branding.zip`).
3. Copy the branding files to the hosted Webapp3 (`cp webapp2/branding/* /var/www/html/webapp2/custom_configuration/`).
4. Go to <http://example.com/webapp2> to use the branded Webapp2.

Note that:

- For Webapp2, if you want to customize the settings you must place your **settings.json** file in the **custom_configuration** directory along with your other customized files.
- For Webapp2, when hosting the web app on an external web server, you must supply a **settings.json** file and change the **serverAddress** variable to point to a Conferencing Node e.g. `"serverAddress": "conferencingnode1.example.com"`. You can only specify one Conferencing Node. This modification is not required if you are hosting the web app on a reverse proxy as it will typically already be configured to forward requests to your Conferencing Nodes.

Maintaining customizations when upgrading Pexip Infinity

If the web app is being hosted on an external web server or reverse proxy, the copy of the web app must be upgraded manually whenever the Pexip Infinity installation is upgraded or you update the web app software bundle. You should migrate the existing customized **branding** directory (Webapp3) or **custom_configuration** directory (Webapp2) on the external web server or reverse proxy onto the new version:

1. Backup the **branding** or **custom_configuration** directory on the external web server or reverse proxy containing your current customizations.
2. Upgrade your Pexip Infinity deployment, or install the software bundle for a new web app version.
3. Download and copy over the upgraded web app files as described in [Copying over the Connect web app](#).
4. Replace the contents of the **branding** or **custom_configuration** directory with your previously customized contents.
5. Check if you need to add any more customizations to support any new features.

When a new version of the web app includes new features, any new customizable elements are added to the default versions of the files in the **branding** or **custom_configuration** directory that is shipped with the new software. Therefore, after an upgrade you should compare your customized versions of these files with the new default versions, to see if any text, styles, colors or resource files should be adjusted.

Obtaining diagnostic information from Connect apps

Users of Connect apps can obtain information about their client's incoming and outgoing audio and video streams, which may be helpful in diagnosing issues with call quality.

To obtain this information, from the top right of the side panel, select **Control** ● ● ● and then select **Get media stats**.

Creating preconfigured links to launch conferences via Connect apps

You can construct URLs or hyperlinks that may be used to automatically launch the Connect app and take the user directly into a specific conference. These URLs can also pass in any additional information specific to that call, such as the caller's name or the PIN needed to enter the meeting.

The URLs are in two formats: one that can be used to launch the [web app](#), and one for use with the [desktop and mobile clients](#).

Alternatively, you can construct a URL that directs the user to a webpage on a Conferencing Node where the full set of join instructions for a specific VMR is shown. These join instructions can optionally include a QR code, which when scanned by a device with a supported Connect app installed (such as Pexip Connect for RealWear or the Connect mobile app) will open the meeting directly in the app. For more information, see [Links to a join instructions page](#).

Security considerations

Although embedding information such as participant names and conference PINs into the URL can make it easier for participants to join conferences, note that these parameters are included in the URL in human-readable format. This means that if a user shares the URL — such as in a screen shot of their meeting invitation — and the URL includes the PIN, anyone with access to the URL can deduce the PIN and enter (and control, if it is a Host PIN) the meeting.

As of version 25 of Pexip Infinity, when a user follows a link to join a conference via the Connect web app, any join parameters, such as a conference PIN, are automatically removed from the URL that is displayed in the browser's address bar.

Links to the web app

Links to the home screen

To open an instance of the Connect web app in the user's default browser and take them to the home screen (not into a specific meeting), use the following link:

`https://<address>/<path>`

where:

<address>	is the IP address or domain name of the Conferencing Node (or reverse proxy if, for example, it is being used to host a customized version of the web app).
<path>	is an optional parameter that specifies the branding path to be used. If no path is specified, users will be redirected to the default path for Webapp3. For more information, see Default paths and default branding .

Links to a specific meeting with no additional parameters

To provide users with a URL that, when clicked, directs them to a specific meeting, but still requires them to enter any information such as their name and the PIN, and does not control their camera or mic mute, use the format:

`https://<address>/<path>/m/<alias>`

where:

<address>	is the IP address or domain name of the Conferencing Node (or reverse proxy if, for example, it is being used to host a customized version of the web app).
<path>	specifies the branding path to be used.
<alias>	is one of the aliases for the conference or service the user will join.


Links to a specific meeting with additional parameters included

To provide users with a URL that, when clicked, takes them straight into a specific conference and also determines in advance settings including their role, the PIN, their camera and mic mute state, construct a URL in the format:


```
https://<address>/<path>/#/?conference=<alias>&name=<name>&pin=<PIN>&role=<role>
&muteMicrophone=<muteMicrophone>&muteCamera=<muteCamera>&callType=<callType>
&callTag=<callTag>&extension=<extension>&bandwidth=<bandwidth>&join=<join>&lng=<lng>
```

where:

<address>	is the IP address or domain name of the Conferencing Node (or reverse proxy if, for example, it is being used to host a customized version of the web app).
<path>	<p>is an optional parameter that specifies the branding path to be used. If no path is specified, users will be redirected to the default path for Webapp3.</p> <p>For more information, see Default paths and default branding.</p>
<alias>	is one of the aliases for the conference or service the user will join.
<name>	is the name of the user who is joining the conference.
<PIN>	is either the Host PIN or Guest PIN, if required (note the Security considerations if these are included).
<role>	<p>For Connect Webapp3 participants, if role=guest is included in the URL, they will be offered an alternative join flow that takes them through the setup of their camera, microphone and speakers before they are able to Join the meeting. For more information, see Support for first-time and infrequent users.</p> <p>For Connect Webapp2 participants, use role=guest if you want to allow Guests to automatically join a conference that allows Guests but has no Guest PIN. In all other cases, participants are asked to enter a PIN to join the conference (unless there is no Host PIN, or the URL already specifies a <PIN>); the PIN determines the participant's role and the <role> is ignored. Note that if role=host, participants are still prompted to enter the Host PIN to join the conference; this parameter cannot be used to bypass PIN entry requirements.</p>
<muteMicrophone>	is true to join without sending audio (the user will still receive audio, and send and receive video).
<muteCamera>	is true to join without sending video (the user will still receive video, and send and receive audio).

<callType>	<p>(Supported in Webapp2, and Webapp3 from Pexip Infinity v33)</p> <p>is one of:</p> <ul style="list-style-type: none"> • none to join as a presentation and control-only participant, i.e. the user will not send or receive any audio or video. • audioonly to join as an audio-only participant, i.e. send and receive audio but not send or receive video. • video (the default) to join as a full (send and receive) audio and video participant. <p>Additional parameters available in Webapp3 only:</p> <ul style="list-style-type: none"> • audiorecvonly to receive but not send audio, and not send or receive video. • audiosendonly to send but not receive audio, and not send or receive video. • videoonly to send and receive video, but not send or receive audio. • videorecvonly to receive but not send video, and not send or receive audio. • videosendonly to send but not receive video, and not send or receive audio. • audiovideosendonly to send audio and video, but not receive audio or video. • audiovideorecvonly to receive audio and video, but not send audio or video. <p> In all cases, the participant can still access the conference controls and chat, and send and receive presentations.</p>
<callTag>	<p>(Supported in Webapp2, and Webapp3 from Pexip Infinity v33)</p> <p>Assigns a call tag for this participant, which is included in logs, policy requests, and participant lists. For more information, see Tracking usage via service and participant call tags.</p>
<extension>	is the Virtual Reception extension, or the Microsoft Skype for Business / Lync Conference ID.
<bandwidth>	is the maximum bandwidth for the call, and the bandwidth at which the initial call attempt will be made, in kbps. It can be any number between 256 and 6144.
<join>	<p>(Not supported in Webapp2)</p> <p>is 1 if you want the participant to automatically join the conference, bypassing the option to check their devices.</p>
<lng>	<p>(Not supported in Webapp2)</p> <p>is the code for one of the supported languages, in order to display Connect Webapp3 in that language. Note that this will override any of the user's own browser language settings.</p>

The URL should always include the **alias** parameter. The remainder of the parameters are optional. If a parameter is not specified in the URL but is required when joining (i.e. **name**, and **PIN** if the conference uses PINs, or **extension** if one is requested), the participant will have to provide the information themselves before they can join the conference.

-  This URL structure will not work on version 24 or earlier of Pexip Infinity, but any URLs using the previously recommended structure (<https://<address>/webapp/conference/<alias>?<parameters>>) will still work on v25 and later, and the join parameters (but not the alias) will be removed from the browser's address bar.

Examples

Assuming the domain name of your Conferencing Node is **vc.pexample.com**, and there is a Virtual Meeting Room with the alias **meet.alice**, which has no PIN:

- the basic URL for someone to join the VMR directly would be:
<https://vc.pexample.com/webapp/#/?conference=meet.alice>
- to set the display name for a participant e.g. "Bob", the URL would be:
<https://vc.pexample.com/webapp/#/?conference=meet.alice&name=Bob>
(Note that if you shared this same link with many participants, they would all join with their display name set to "Bob".)

If we then gave the same Virtual Meeting Room a Host PIN of **1234**, and allowed Guests to join without a PIN:

- the URL for Bob to join it directly as a **Host** would be:
`https://vc.pexample.com/webapp/#/?conference=meet.alice&name=Bob&pin=1234`
- the URL for Bob to join it directly as a **Guest** would be:
`https://vc.pexample.com/webapp/#/?conference=meet.alice&name=Bob&role=guest`

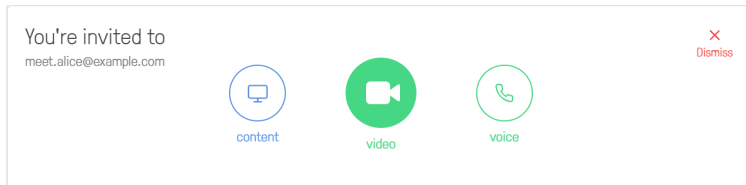
Alternative structure to join with an invitation card

(Supported in Webapp2 only)

You can use an alternative URL structure in the format:

`https://<address>/webapp/home?conference=<alias>`

In this case the web app will launch with an invitation to join the nominated <alias>, and it gives the user an opportunity to modify their settings (such as camera, mic and so on) before joining the conference, and they can select whether they want to join with **video**, **voice**, or **content** and control only.



This can be useful in scenarios where a user has previously set their camera to **None**, and are unable to modify their device settings if they have been taken directly into a conference.

Links to the desktop and mobile clients

You can create a URL that, when clicked, opens the Connect app on that device and provides an invitation to join the nominated conference. The same URL can be used for the desktop client and mobile clients for Android and iOS. This URL can be included in web pages, instant messages or emails (but note that some email clients such as Gmail will strip them out for security reasons).

i The Connect app desktop or mobile client must already be installed on the device.

The URL is in the format:

`pexip://<alias>?host=<domain>&name=<name>&pin=<PIN>&role=<role>&muteMicrophone=<muteMicrophone>&muteCamera=<muteCamera>&extension=<extension>&bandwidth=<bandwidth>`

where:

<alias>	is one of the aliases for the conference or service the user is invited to join.
<domain>	is the IP address or domain name of the Conferencing Node (or reverse proxy if, for example, it is being used to host a customized version of the web app) the client should connect to in order to place the call. Note that this is ignored if the client is registered and Route calls via registrar is enabled.
<name>	is the name of the user who is joining the conference.
<PIN>	is either the Host PIN or Guest PIN, if required (note the Security considerations if these are included).
<role>	is guest if you want to allow Guests to join a conference without having to enter a PIN (providing the conference allows Guests and has no Guest PIN). In all other cases, participants are asked to enter a PIN to join the conference (unless there is no Host PIN, or the URL already specifies a <PIN>); the PIN determines the participant's role and the <role> is ignored. Note that if role=host , participants are still prompted to enter the Host PIN to join the conference; this parameter cannot be used to bypass PIN entry requirements.
<muteMicrophone>	is true to join without sending audio (the user will still receive audio, and send and receive video).
<muteCamera>	is true to join without sending video (the user will still receive video, and send and receive audio).

<extension>	is the Virtual Reception extension, or the Microsoft Skype for Business / Lync Conference ID.
<bandwidth>	is the maximum bandwidth for the call, and the bandwidth at which the initial call attempt will be made, in kbps. It can be any number between 256 and 6144.

The URL must always include **pexip://<alias>**. The remainder of the parameters are optional. If a parameter is not specified in the URL but is required when joining (i.e. **name**, and **PIN** if the conference uses PINs, or **extension** if one is requested), the participant will have to provide the information themselves before they can join the conference.

Example - email footer

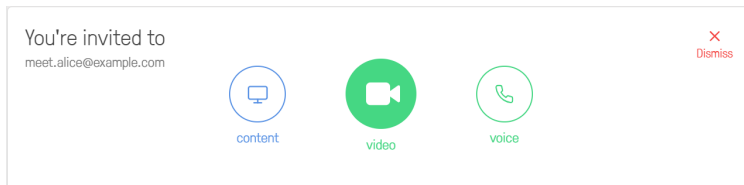
For example, Alice's personal meeting room has the alias **meet.alice@pexample.com** so she includes the following text in her email footer:

- Video: `meet.alice@pexample.com`

which displays as:

- Video: [meet.alice@pexample.com](pexip://meet.alice@pexample.com)

Now, when someone who has a Connect app installed on their device clicks on the link in Alice's email, their client will open automatically with an invitation to join **meet.alice@pexample.com**, and all they need to do is select whether they want to join with **video**, **voice**, or **content** and control only:



Example - Guest PIN

Alice's personal meeting room has a Guest PIN of **1234**. When Alice is chatting with a colleague using an instant messaging client and she wants to move the conversation to video, she sends them the message **pexip://meet.alice@pexample.com?pin=1234**, which automatically appears as a hyperlink. Her colleague clicks on the link and is invited to join Alice's personal meeting room as a Guest.

Example - always join with microphone muted

If you want the participant to join a meeting with a PIN of 1234, and you want their microphone to be muted on joining, the URL would be: **pexip://meet.alice@pexample.com?pin=1234&muteMicrophone=true**

Links to a join instructions page

You can generate a URL to a webpage on a Conferencing Node where the full set of join instructions for a specific VMR is shown. These join instructions can optionally include a QR code, which when scanned by a device with a supported Connect app installed (such as Pexip Connect for RealWear or the Connect mobile app) will open the meeting directly in the app.

The [format of the link](#) is described below.

- To include this URL in invitations to meetings scheduled using VMR Scheduling for Exchange, you must edit the [Personal VMR joining instructions template](#) or the [Single use VMR joining instructions template](#) to include the link.
- To include this URL in [emails sent when a VMR is provisioned](#), you must edit the VMR email body template to include the link.
- To include this URL elsewhere, for example within your email footer, include it within a hyperlink tag, e.g.

```
<a href="https://px.vc.pexample.com/teams/join.html?conf=meet.alice&d=pexample.com&test=test_call&w&qrcode">Contact me on video</a>
```

Formatting the URL

The URL takes the format:

`https://<node_address>/teams/join.html?conf=<alias>&d=<domain>&test=<test_call_alias>&w&qrcode`

where `<node_address>` is an FQDN that resolves to your Pexip Conferencing Nodes (it can also be the FQDN or IP address of an individual Conferencing Node). Note that:

- To view this webpage, the client application used to view the invitation must be able to access the specified Conferencing Nodes (or alternative server) on HTTPS 443/TCP.
- You must ensure that the FQDN used here is resolvable by any internal or external client applications that may be used to view the invitation i.e. that they can access the webpage on the nodes referenced by the FQDN. This means that if these nodes have private addresses, then depending on your internal network routing, you may need appropriate local DNS resolution for the `<node_address>` FQDN for internally-based clients, in addition to external DNS resolution for that FQDN for externally-based clients.

The other parameters are:

Parameter	Mandatory	Description
conf	Yes	An alias that can be used to join the meeting.
d	Yes	The domain name of your Pexip Infinity platform e.g. <code>pexample.com</code> . This is used as the domain for all of the URI-style addresses that are displayed on the webpage.
test	No	<p>Includes a "Test call" option on the webpage, where the value of this parameter is the name part of the alias to dial e.g. <code>test_call</code>. Do not include the domain — this is the <code>d</code> parameter above.</p> <p>This uses Pexip Infinity's inbuilt Test Call Service; therefore you must ensure that <code><test@d></code>, e.g. <code>test_call@pexample.com</code>, matches the name of the alias configured in Pexip Infinity for the test call service.</p>
w	No	<p>Displays the "From a browser" access details on the webpage.</p> <p>There is no value associated with this parameter.</p>
qrcode	No	<p>Includes a QR code on the webpage, which when scanned by a device with a supported Connect app installed (such as Pexip Connect for RealWear or one of the Connect mobile apps) will open the meeting directly in that app.</p> <p>There is no value associated with this parameter.</p>

An example URL value could be: `https://px.vc.pexample.com/teams/join.html?conf=meet.alice&d=pexample.com&test=test_call&w&qrcode`

and that would produce the following webpage:



Video meeting invitation



Join the meeting directly

From a VTC/SIP system, enter: meet.alice@pexample.com



From a browser

Go to: <https://px.vc.pexample.com/webapp/?conference=meet.alice@pexample.com>



From the Pexip app on a supported device:

Scan this with your camera: <pexip://meet.alice@pexample.com?host=nightly.pexip.com>



Test call

To test your connection from a VTC/SIP system, enter: test_call@pexample.com and verify that you can see and hear yourself

Powered by Pexip

Links to the legacy Connect apps

For information on creating links to legacy clients, see the [v17 documentation](#).

Setting up DNS records and firewalls for Connect app connectivity

To ensure that Connect apps can successfully locate and connect to Pexip Infinity you must set up appropriate DNS records and ensure your firewalls are configured correctly.

DNS records

You must set up DNS records so that the Connect apps know which host to contact when placing calls or registering to Pexip Infinity.

The host will typically be a public-facing Conferencing Node (for on-premises deployments where your Transcoding Conferencing Nodes are located within a private network we recommend that you deploy public-facing Proxying Edge Nodes).

To enable access from the Connect desktop apps and Connect mobile apps, each domain used in aliases in your deployment must either have a DNS SRV record for `_pexapp._tcp.<domain>`, or resolve directly to the IP address of a public-facing Conferencing Node.

The SRV records for `_pexapp._tcp.<domain>` should always:

- point to an FQDN which **must** be valid for the TLS certificate on the target Conferencing Nodes
- reference port 443 on the host.

Note that SRV records are not required for the Connect web app — the web app connects to Conferencing Nodes directly via DNS A-records, so no SRV lookup is required.

Ultimately it is the responsibility of your network administrator to set up SRV records correctly so that the Connect desktop app and Connect mobile app know which system to connect to.

You can use the tool at <http://dns.pexip.com> to lookup and check SRV records for a domain.

Firewall configuration

Connect apps connect to a Conferencing Node, so you must ensure that any firewalls between the two permit the following connections:

- Connect mobile app > Conferencing Node port 443 TCP
- Connect app (all clients) > Conferencing Node ports 40000–49999 TCP/UDP
- Conferencing Node ports 40000–49999 TCP/UDP > Connect app (all clients)

Using the Connect app from outside your network

In many cases, your Pexip Infinity deployment will be located inside a private network. If this is the case and you want to allow Connect app users who are located outside your network (for example on another organization's network, from their home network, or the public internet) to connect to your deployment, you need to provide a way for those users to access those private nodes.

Since version 16 of Pexip Infinity, we recommend that you deploy Proxying Edge Nodes instead of a reverse proxy and TURN server if you want to allow externally-located clients to communicate with internally-located Conferencing Nodes. A Proxying Edge Node handles all media and signaling connections with an endpoint or external device, but does not host any conferences — instead it forwards the media on to a Transcoding Conferencing Node for processing.

Further information and connectivity examples

Information on how each of the Connect apps attempt to locate a Conferencing Node when placing a call is described in the following sections. Within each section is an example of the lookup process for that client. The example uses the following records:

Assume that the following `_pexapp._tcp.vc.example.com` DNS SRV records have been created:

```
_pexapp._tcp.vc.example.com. 86400 IN SRV 10 100 443 px01.vc.example.com.  
_pexapp._tcp.vc.example.com. 86400 IN SRV 20 100 443 px02.vc.example.com.
```

These point to the DNS A-records `px01.vc.example.com`, port 443 (HTTPS), with a priority of 10 and a weight of 100, and `px02.vc.example.com`, port 443, with a relatively lower priority of 20 and a weight of 100.

This tells the Connect desktop apps and Connect mobile apps to initially send their HTTP requests to host `px01.vc.example.com` (our primary node) on TCP port 443. The Connect apps will also try to use host `px02.vc.example.com` (our fallback node) if they cannot contact `px01`.

Connect desktop app

Registering

The Connect desktop app uses its configured **Registration Host** and performs a DNS SRV lookup on `_pexapp._tcp.<registration host address>` to locate a Conferencing Node to which it can send its registration request.

In all cases, when performing an SRV lookup on `_pexapp._tcp.<registration host address>`:

- If multiple records are returned, the client attempts to contact each host in turn according to the priority and weight of each returned record.
- If the SRV lookup fails (because either the SRV lookup does not return any records, or the client cannot contact any of the hosts (i.e. Conferencing Nodes) on the list that is returned in the SRV lookup), the client performs a DNS A-record lookup for `<registration host address>`. If that A-record lookup is successful, it attempts to connect to port 443 on the IP address returned from the lookup.

Making calls

The way in which Connect desktop apps decide which Conferencing Node to use when attempting to place a call depends on whether the client is registered, and on the global Route calls via registrar setting at the time of registration.

When placing a call, the Connect desktop app attempts to locate a Conferencing Node by doing **one** of the following, in order of precedence:

- If the client is registered to Pexip Infinity and the global Route via registrar setting is enabled, the client will route all calls directly to the IP address of the Conferencing Node to which it is registered, regardless of the domain being dialed. From there, the call is treated as an incoming call and processed according to the service precedence call routing logic.
- If the client is not registered or **Route calls via registrar** is disabled, and the call was placed [via a URL](#) that specifies a host domain, then the client performs an SRV lookup on **_pexapp._tcp.<host domain>**.
- If a **serverAddress** is specified in the client's application settings file (**settings.json**), the client performs an SRV lookup on **_pexapp._tcp.<serverAddress>**. Note that the **serverAddress** is not configured in the default desktop client provided by Pexip, but an address could have been configured during customization and provisioned to the client.
- If a **serverAddress** was not specified, the client performs an SRV lookup on the domain portion of the address that was dialed, i.e. **_pexapp._tcp.<address domain>**.

In all the above cases, when performing an SRV lookup:

- If multiple records are returned, the client attempts to contact each host in turn according to the priority and weight of each returned record.
- If the SRV lookup fails (because either the SRV lookup does not return any records, or the client cannot contact any of the hosts (i.e. Conferencing Nodes) on the list that is returned in the SRV lookup), the client performs a DNS A-record lookup for the domain in the SRV lookup. If that A-record lookup is successful, it attempts to connect to port 443 on the IP address returned from the lookup.
- If the client successfully contacts a host but that Conferencing Node is in maintenance mode, the client will not make any further attempts to contact any other hosts.
- When the client successfully contacts a host, the host will then check to see if the alias that has been dialed exists in its configuration. This means that the alias does not need to include a domain if, for example, the host has been found via a lookup on the **serverAddress**. It also means that the domain in the alias being dialed does not necessarily need to be the same as the domain of the host.

Example

In this example, when a user attempts to place a call to `meet.alice@vc.example.com`, the client does **one** of the following:

- If the client is registered to Pexip Infinity and the global Route via registrar setting is enabled, the client will route all calls directly to the IP address of the Conferencing Node to which it is registered, regardless of the domain being dialed.

For example, if the client is configured with a Registration Host of `registration.example.com`, then the client will perform an SRV lookup on `_pexapp._tcp.registration.example.com`.

If the SRV lookup fails, or none of the returned hosts in the lookup can be contacted, the client will also attempt to connect directly to that domain, i.e. to `http://registration.example.com:443` (via DNS A-records for `registration.example.com`).

- If the call is being placed via a preconfigured link that specifies a host domain, then the client will perform an SRV lookup on that domain, and attempt to contact one of the hosts returned in that lookup.

For example, if the URL is `pexip://meet.alice@vc.example.com?host=localserver.example.com` then the client will perform an SRV lookup on `_pexapp._tcp.localserver.example.com`.

If the SRV lookup fails, or none of the returned hosts in the lookup can be contacted, the client will also attempt to connect directly to that domain, i.e. to `http://localserver.example.com:443` (via DNS A-records for `localserver.example.com`).

If that also fails, no further lookups are performed, and the client will report that it could not join the host domain.

- If a `serverAddress` has been configured, the client performs an SRV lookup on that domain, and attempts to contact the host(s) returned in that lookup.

For example, if the `serverAddress` is `localserver.example.com` then the client performs an SRV lookup on `_pexapp._tcp.localserver.example.com`.

If the SRV lookup fails, or none of the returned hosts in the lookup can be contacted, the client also attempts to connect directly to that domain, i.e. to `http://localserver.example.com:443` (via DNS A-records for `localserver.example.com`).

If that also fails, no further lookups are performed, and the client will report that it could not join the host domain.

- In all other cases, the client attempts an SRV lookup on the domain portion of the address that was dialed, i.e. on `_pexapp._tcp.vc.example.com`.

If the SRV lookup succeeds, it returns the records shown above, and the client will attempt to contact `px01.vc.example.com` (the record with the highest priority) on TCP port 443.

If it cannot contact `px01.vc.example.com` it next tries to contact `px02.vc.example.com`.

If it fails to contact either host, the client also attempts to connect directly to the domain, i.e. to `http://vc.example.com:443` (via DNS A-records for `vc.example.com`).

If that also fails, the client will report that it has failed to contact a server.

Connect mobile app

Making calls

When placing a call, the Connect mobile app attempts to locate a Conferencing Node as follows:

- If the call was placed [via a URL](#) that specifies a host domain, then the client performs an SRV lookup on `_pexapp._tcp.<host domain>`.
- Otherwise, the client performs an SRV lookup on the domain portion of the address that was dialed, i.e. `_pexapp._tcp.<address domain>`.

In all the above cases, when performing an SRV lookup:

- If multiple records are returned, the client attempts to contact each host in turn according to the priority and weight of each returned record.
- If the SRV lookup fails (because either the SRV lookup does not return any records, or the client cannot contact any of the hosts (i.e. Conferencing Nodes) on the list that is returned in the SRV lookup), the client performs a DNS A-record lookup for the domain in the SRV lookup. If that A-record lookup is successful, it attempts to connect to port 443 on the IP address returned from the lookup.
- If the client successfully contacts a host but that Conferencing Node is in maintenance mode, the client will not make any further attempts to contact any other hosts.

- When the client successfully contacts a host, the host will then check to see if the alias that has been dialed exists in its configuration. This means that the domain in the alias being dialed does not necessarily need to be the same as the domain of the host.

Example

In this example, when a user attempts to place a call to `meet.alice@vc.example.com`, the client does **one** of the following:

- If the call is being placed via a preconfigured link that specifies a **host** domain, then the client will perform an SRV lookup on that domain, and attempt to contact one of the hosts returned in that lookup.
For example, if the URL is `pexip://meet.alice@vc.example.com?host=localserver.example.com` then the client will perform an SRV lookup on `_pexapp._tcp.localserver.example.com`.
If the SRV lookup fails, or none of the returned hosts in the lookup can be contacted, the client will also attempt to connect directly to that domain, i.e. to `http://localserver.example.com:443` (via DNS A-records for `localserver.example.com`).
If that also fails, no further lookups are performed, and the client will report that it could not join the host domain.
- In all other cases, the client attempts an SRV lookup on the domain portion of the address that was dialed, i.e. on `_pexapp._tcp.vc.example.com`.
If the SRV lookup succeeds, it returns the records shown above, and the client will attempt to contact `px01.vc.example.com` (the record with the highest priority) on TCP port 443.
If it cannot contact `px01.vc.example.com` it next tries to contact `px02.vc.example.com`.
If it fails to contact either host, the client also attempts to connect directly to the domain, i.e. to `http://vc.example.com:443` (via DNS A-records for `vc.example.com`).
If that also fails, the client will report that it has failed to contact a server.

Deploying the Connect desktop app under Citrix

Users can securely access the Connect desktop app via Citrix Virtual Desktops or via Citrix Virtual Apps using the Citrix Workspace app to join VMRs, call through the Pexip Gateway to Microsoft Teams, or simply place point-to-point calls. This solution:

- Provides secure, easy to join meetings from any location.
- Requires no download or plugins.
- Allows Connect desktop app users to directly launch their video call from a secure desktop.

The Connect desktop app versions 1.12.x for Citrix Virtual Desktops and Citrix Virtual Apps are compatible with Pexip Infinity version 28 or later.

Limitations

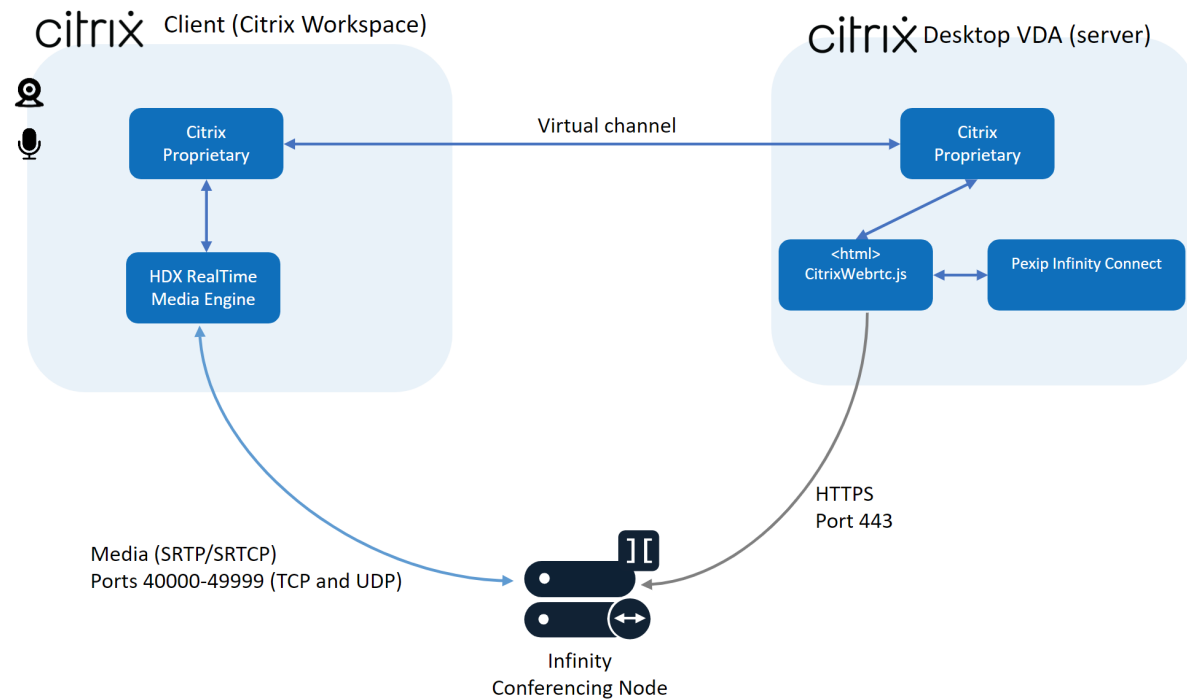
This solution has the following limitations:

- When someone is sharing their screen, their content is sent to other participants at 5 fps by default. This default rate cannot be changed by selecting **Settings > Advanced Settings > Screen**.
- The **Enable media relay on TCP port 443** global settings option is not supported for Citrix virtual desktops.

Architecture overview

The Connect desktop app for Citrix deployment has 2 main components:

- **Citrix Virtual Delivery Agent (VDA)**: server (virtual machine) where the Connect desktop app is hosted. The Citrix Workspace App uses virtual desktops to access the Connect desktop app. When a call is received or made on the Connect desktop app, the VDA connects to a Conferencing Node using the **Citrix WebRTC** JavaScript to manage all the call signaling. All the media processing is performed on the Citrix Workspace App and the Citrix VDA handles the Connect desktop app and call signaling.
- **Citrix Workspace App**: used by the end users to connect to virtual desktops. The **Citrix WebRTC** running on VDA handles the communication to the Workspace App. The Workspace App uses the **HDX Real Time Media Engine** to handle media processing.



Supported Citrix versions

Versions 1.12.x of Connect desktop app support:

- Citrix Workspace App 20.9.5.x
- Citrix Virtual Delivery Agent (VDA) 1912 or later
- Connect desktop app requires your VDA environment to be installed on Windows Server 2016 (build 1607) or later
- For customers on Long Term Service Release (LTSR), Citrix recommends the Workspace App and VDA to be version 2203 LTSR or later

Prerequisites

The following instructions assume that in a Citrix environment you have already:

- Installed a suitable Citrix VDA server
- Installed Citrix Workspace App at the client
- Created a user account to log in to the Citrix Workspace App

Port requirements

These are the port usage rules for call signaling and media between Citrix and Conferencing Nodes:

Source address	Source port	Destination address	Dest. port	Protocol	Notes
Citrix Workspace App	19302–19309	Conferencing Node	40000–49999 **	TCP and UDP	SRTP/SRTCP
Citrix VDA	33000–39999 **	Conferencing Node	443	TCP (HTTPS)	
Citrix Workspace App	<any>	TURN server	3478	UDP	UDP TURN/STUN (optional)
Citrix Workspace App	<any>	TURN server	49152–65535	UDP	TURN relay media (optional)
TURN server	49152-65535	Citrix Workspace App	<any>	UDP	RTP media (optional)

** Configurable via the Media port range start/end, and Signaling port range start/end options.

Installing the Connect desktop app

To install the Connect desktop app:

1. Go to the [Pexip App download page](#) and download `pexip-infinity-connect_<release>_win-x64.msi` file for Windows.
2. Install the `pexip-infinity-connect_<release>_win-x64.msi` file with the following switches:
 - `ALLUSERS=1`
 - `CITRIX=1`

For example:

```
msiexec /i pexip-infinity-connect_<version>_win-x64.msi ALLUSERS=1 CITRIX=1
```

During the installation process the Connect app icon is added to the desktop, and entries are added to the Windows registry to allow links prefixed with `pexip:` and `pexip-provision:` to open automatically in the Connect desktop app.

3. Ensure that the Citrix VDA server has the following registry settings to support WebRTC media redirection:

```
HKEY_LOCAL_MACHINE\Software\WOW6432Node\Citrix\WebSocketService
"Name": ProcessWhitelist
"Type": REG_MULTI_SZ
"Data": pexip-infinity-connect.exe
```

4. Restart the Citrix VDA server.
5. After restarting the Citrix VDA server, ensure that the `CtxHdxWebSocketService` is running on the server.

Running as a Citrix Published Desktop

When running the Connect desktop app as a Citrix Published Desktop, you must disable hardware acceleration.

To do this, use the following command line options:

```
--disable-gpu --citrix
```

For more information, see [Virtual environments](#).

Running as a Citrix Published App

The Connect desktop app can only run as a Citrix Published App without a window titlebar. Hardware acceleration must also be disabled.

To do this, run the app with the following command-line options:

```
--disable-gpu --citrix --frameless
```

For more information, see [Virtual environments](#).

About Connect web app versions

There are currently three different versions of the Connect web app available to users connecting to a Pexip Infinity deployment. The version that users are offered depends on the URL path that they use to access the web app. As an administrator you can configure web app paths and also decide which version of the web app is offered to users by default.

All three Connect web app versions use the same API to connect to a Pexip Infinity deployment but each version offers a different user interface, supports different Pexip Infinity features, and is customized separately. Within the Administrator interface (e.g. when customizing the web apps or installing software bundles) each version is managed separately and are differentiated as follows:

- **Webapp3:** this is the latest version and was released with Pexip Infinity v30.
- **Webapp2:** this is the previous version, originally released with Pexip Infinity v18. This version is aligned with the current Connect desktop app and Connect mobile app in terms of user interface and functionality, and any customization of Webapp2 can also be applied to the Connect desktop app.
- **Webapp1:** this is the original "legacy" version, and is no longer actively developed or maintained and has been removed from the default installation of Pexip Infinity.

Setting which version is offered by default

Users access the web app by entering the IP address or FQDN of a Conferencing Node or reverse proxy in your deployment, either on its own or with a path appended. Each path specifies the web app version to be used, but you can configure where to redirect users who either do not enter a path or enter the `/webapp` path (supported by previous versions of Pexip Infinity).

For full details, see [Redirecting "/webapp" \(or no path\)](#).

Troubleshooting Connect app error messages

The table below lists the specific messages that may be presented to Connect app users, along with their meaning and suggested resolution (where appropriate). To assist administrators with troubleshooting, the associated admin-facing message (which appears in the admin log, and when viewing historical information about a participant) is also given.

For help with general issues that may occur when using Connect app clients within your deployment, see [Troubleshooting the Pexip Infinity platform](#).

Admin-facing message	User-facing message	Message code	Meaning/resolution
Call Failed: Invalid role	The PIN you entered is invalid - please try again.	#pex100	
Call Failed: Invalid PIN	The PIN you entered is invalid - please try again.	#pex101	The PIN that was entered did not match the Host (or Guest, if configured) PIN.
Call failed: Out of proxying resource	Error connecting to the meeting	#pex109	All Proxying Edge Nodes in the location are out of capacity.
Call Failed: System in maintenance mode	The system you are trying to reach is temporarily unavailable. Please try again shortly.	#pex110	The Conferencing Node is in maintenance mode. Note that if the client encounters a node in maintenance mode while performing an , it will not attempt to contact any other nodes.
Call Failed: 502 Bad Gateway	There is no connection. Please try again.	#pex111	
Call Failed: 503 Service Unavailable	There is no connection available.	#pex112	
Call Failed: Invalid token	Your connection was lost. Please try again.	#pex113	
Call Failed: Out of resource	The system you are trying to reach is over capacity.	#pex114	
transfer failed	Transfer failed.	#pex115	A Host participant attempted to transfer another participant from the current meeting to another meeting, but failed.
Call Failed: Unexpected Response: 503	Call failed - please contact your administrator	#pex116	Pexip Infinity received an Unexpected Response (503) when trying to place the call. If this issue persists, you may wish to send a snapshot to your Pexip authorized support representative.
Call failed: <code>	The call failed. Please try again.	#pex117	Generic failure code.

Admin-facing message	User-facing message	Message code	Meaning/resolution
Could not join localhost:8080	The server cannot be reached.	#pex118	The host server (obtained either as the result of the DNS lookup, or by using the domain part of the dialed alias) could not be found.
Call Failed: Failed to forward request	Call failed: Failed to forward request	#pex119	
Conference host ended the conference with a DTMF command	A Host ended the meeting.	#pex120	A Host participant ended the call using a DTMF command.
Conference terminated by a Host participant	A Host ended the meeting.	#pex121	A Connect app Host participant has selected "disconnect all", or a client API command was used to terminate the conference.
Conference terminated by an administrator	An administrator ended the meeting.	#pex122	An administrator using the Pexip Infinity Administrator interface has selected "disconnect all", or a management API command was used to end the conference.
Disconnected by an administrator	An administrator disconnected you from the meeting.	#pex123	An administrator using the Pexip Infinity Administrator interface has disconnected this particular participant.
Disconnected by another participant	Another participant in the meeting disconnected you.	#pex124	A Host using a Connect app has disconnected a specific participant.
Conference terminated by another participant	A Host ended the meeting.	#pex125	A Connect app Host participant has selected "disconnect all", or a client API command was used to terminate the conference.
Timeout waiting for conference host to join or permit access to locked conference	The meeting Host has not joined or unlocked the meeting.	#pex126	The participant timed out because the conference Host either did not join the conference, or did not permit the participant to join a locked conference.
	This feature has been disabled.	#pex127	The setting to Enable support for Pexip Infinity Connect and Mobile App has been disabled by an administrator.
Call failed: failed to establish media to server. Ensure required firewall ports are permitted.	Call failed: a firewall may be blocking access.	#pex128	An ICE failure has occurred.
Signaling node disconnected	Something went wrong with the meeting. Please try to connect again.	#pex129	The media node lost connectivity to the signaling node.
Media process disconnected	Something went wrong with the meeting. Please try to connect again.	#pex130	The Conferencing Node hosting the media has encountered an unexpected behavior.
Media node disconnected	Something went wrong with the meeting. Please try to connect again.	#pex131	The signaling node lost connectivity to the media node.
Proxied participant disconnected	Something went wrong with the meeting. Please try to connect again.	#pex132	The proxying node lost connectivity to the transcoding node.
No participants can keep conference alive	The meeting has ended.	#pex140	This was the only remaining participant, and they were an ADP that was not configured to keep the conference alive.

Admin-facing message	User-facing message	Message code	Meaning/resolution
All conference hosts departed hosted conference	The meeting ended because the Host(s) left.	#pex141	There are no Host participants remaining in the conference.
Last remaining participant removed from conference after timeout	You were the only participant left in the meeting.	#pex142	This was the only participant remaining, and they were disconnected after the configured amount of time.
Test call finished	The test call has finished.	#pex143	This was a call to the Test Call Service that was automatically disconnected after the specified time.
Call rejected	The person you are trying to call did not answer or could not be reached.	#pex150	The person being called did not answer or could not be reached.
Call disconnected	The other participant has disconnected.	#pex151	A Connect app has been disconnected by themselves or another system other than Pexip Infinity.
Gateway dial out failed	The call could not be placed.	#pex152	The alias matched a Call Routing Rule but the call could not be placed.
invalid gateway routing rule transform	The call could not be placed. Please contact your administrator.	#pex153	The alias matched a Call Routing Rule but the resulting alias was not valid.
Call Failed: Neither conference nor gateway found	"Cannot connect to <alias>. Check this address and try again.	#pex154	The alias that was dialed did not match any aliases or Call Routing Rules.
Could not join <domain part of dialed alias>	Could not join <domain>	#pex155	The domain is not part of a Pexip Infinity deployment. This error can occur if an incorrect serverAddress has been specified during customization. It can also occur if a SSL error is preventing a secure connection to the server.
Participant failed to join conference Reason="No direct route between Edge and Transcoding"	The call could not be placed.	#pex156	There is an issue with media location policy. This typically occurs if restricted routing for Proxying Edge Nodes is enabled and the proxying node cannot forward the media to the nominated transcoding node.
Not Found: The requested URL <address> was not found on this server	Could not join <domain>	#pex157	Check that the URL is structured correctly.
Failed to gather IP addresses.	Call failed: Please disable any privacy extensions on your browser.	#pex170	The browser cannot find the local IP address. This may be due to ad blockers. A Connect app could not determine its IP address. This may be because there are privacy extensions installed.
Call Failed: Error: Could not get access to camera/microphone. Have you allowed access? Has any other application locked the camera?	Your camera and/or microphone are not available. Please make sure they are not being actively used by another app.	#pex171	A Connect app WebRTC participant has not allowed their camera or microphone to be shared, or has no camera or microphone available.
Presentation ended	The presentation ended.	#pex180	
Presentation stream remotely disconnected	The presentation stream was disconnected.	#pex181	

Admin-facing message	User-facing message	Message code	Meaning/resolution
Presentation stream unavailable	The presentation stream is unavailable.	#pex182	
Screenshare canceled	The screenshare was canceled.	#pex183	
Screenshare error	Something went wrong with screenshare. Please try again.	#pex184	
Screenshare remotely disconnected	The screenshare was disconnected.	#pex185	
Timer expired awaiting token refresh	Error connecting to the meeting	#pex190	A Connect app was unable to refresh its token after 2 minutes. This is likely due to network issues.
Resource unavailable	Error connecting to the meeting	#pex191	There was insufficient transcoding or proxying capacity on the Transcoding Conferencing Node or the Proxying Edge Node on which the call landed.
Participant exceeded PIN entry retries	Too many PIN entry attempts	#pex192	The participant exceeded the allowed number of PIN entry attempts (3).
Invalid license	Error connecting to the meeting. Please contact your administrator	#pex193	There is an invalid license, for example a license may have expired.
Participant failed to join conference... Reason="Participant limit reached"	This meeting has reached the maximum number of participants.	#pex194	A user has attempted to join a conference that has exceeded its configured number of participants.
	Error connecting to the meeting. Please contact your administrator.	#pex195	All the existing licenses are currently in use.
	Error connecting to the meeting. Please reconnect.	#pex196	The Connect app's ICE connection failed or was interrupted. Users should attempt to reconnect.
ERROR_SSO_AUTHENTICATION	SSO Authentication Failed	#pex200	Check your username and password and try again.
ERROR_SSO_NO_IDENTITY_PROVIDERS	SSO enabled but no Identity Providers configured	#pex201	SSO setup may not be complete. Contact your administrator.
ERROR_SSO_POPUP_FAILED	Unable to open window for SSO authentication. This may have been prevented by a pop-up blocker.	#pex202	Disable your pop-up-blocker for this website if you can.
ERROR_SSO_AUTHENTICATION_MAINTENANCE	SSO Authentication Failed. The system is in Maintenance mode.	#pex203	Try again later.
Failed SAML SSO Request	SSO authentication failed. SSO is not available from this domain.	#pex204	The domain in the URL (for web app clients) or the domain in the registration address (for desktop clients) is not on the list of allowed Assertion Consumer Service URLs.
	Failed to transfer into a multi-party conference.	#pex210	The escalation transfer failed to complete. Redial and rejoin the meeting.
	Failed to transfer into a one-to-one conference.	#pex211	The de-escalation transfer failed to complete. Redial and rejoin the meeting.

Connect app release notes

For information about the new features, fixed issues, and known limitations the Connect app s see:

- [Connect web app release notes](#)
- [Connect desktop app release notes](#)
- [Connect mobile app release notes](#)

For release notes for the Pexip Infinity platform, see [Pexip Infinity release notes](#).

Connect web app release notes

This topic describes the new features, changes in functionality, and fixed issues in the current and previous supported releases of the Connect web app:

- [What's new?](#)
- [Fixed issues](#)
- [Known limitations](#)

What's new?

New in v34 Webapp3

Following are the new features and changes in Connect Webapp3 in Pexip Infinity version 34:

Feature	Description	More information
New features		
Portrait mode support for WebRTC devices in all layouts *	<p>Any WebRTC device with a portrait aspect ratio (such as 9:16) can now receive a layout specifically designed for a portrait display in all layouts (previously this was limited to the Adaptive Composition layout).</p> <p>This is still a technology preview feature and can be enabled via Platform > Global Settings > Tech Preview Features > Enable Portrait Layouts.</p>	
Custom layout selection	<p>Users can now select any of the custom layouts that are available to the current conference when changing the layout via the Meeting layout tab.</p>	
Presentation in Mix support	<p>Webapp3 users can control whether to view presentation in mix automatically when a supporting layout is in use. This option is off by default.</p> <p>Administrators can customize this option to be on by default.</p>	
Far-End Camera Control (FECC) support	<ul style="list-style-type: none">• Webapp3 users can control the camera of another participant if that participant's camera supports FECC.• Webapp3 users can choose whether their own camera can be controlled by other participants.	
Lower a raised hand	<p>Meeting hosts can "lower" another participant's raised hand.</p>	
Prioritize motion over sharpness	<p>When sharing content with other participants, Webapp3 users can choose whether to prioritize motion (best for videos and moving images) or sharpness (best for static presentations and images).</p>	
Custom favicon	<p>Administrators can customize the image used for the Webapp3 favicon.</p>	
Hide UI elements	<p>Administrators can use customization to hide selected UI components.</p>	

Feature	Description	More information
Support for third-party authentication	Support has been added for features (such as plugins) that require authentication to a third party using an OAuth 2.0 / OpenID Connect flow. The redirect destination is <code>webapp3/oauth-redirect</code> .	
Custom join flow *	Administrators can use customization to add an extra step to the join flow to show additional information (for example, terms and conditions of use) and which can optionally require acceptance or confirmation from the user in order for them to proceed with the call.	
Support for screen capture API updates	Administrators can use customization to override default screen capture parameters, supporting recent changes to W3C's Screen Capture API . For more information, contact your Pexip authorized support representative.	
Changes in functionality		
Processing improvements	The blur algorithm has been improved, reducing the local processing overhead when background blur is enabled.	
Improvements to permission requests	Various improvements to the way in which requests for blocked camera and mic permissions are presented to users.	
Improvements to content sharing	Improvements to the way in which audio and video content is detected and shared.	
Plugin improvements	Various improvements and additions to enhance support for visual elements used in plugins (for example forms, prompts, toasts, buttons).	
* Technology preview only		

New in v34 Webapp2

There are no significant new features in Connect Webapp2 in Pexip Infinity version 34.

Fixed issues

Fixed in v34 Webapp3

Ref #	Resolution
GL4473	If a white or light background color is in use, the post-meeting text is now black rather than white, to improve visibility.
GL4409	Resolves an issue whereby in some situations when the camera selection had toggled between two devices, the final camera selection would not persist.
GL4408	The speaker device selection now persists between meetings. Previously only the camera and microphone selection persisted.
GL4109	Resolves an issue whereby a preconfigured meeting URL resulted in the meeting being joined automatically even when <code>join=1</code> was not included.
GL3822	Language parameters are no longer ignored if they are included in a preconfigured URL that also includes a custom branding path.

Fixed in v34 Webapp2

There were no significant user-facing fixes in this release.

Known limitations

Ref #	Limitation
GL4293	(Webapp3 only) If the <code>defaultUserConfig</code> section of the manifest file specifies settings for <code>isAudioInputMuted</code> or <code>isVideoInputMuted</code> , any settings included in a preconfigured URL for <code>muteMicrophone</code> or <code>muteCamera</code> respectively will not override the <code>defaultUserConfig</code> and are ignored.
19889	When using the Connect web app in a browser on a mobile device in landscape mode, in some cases the video and presentation is cropped. To work around this, we recommend using the Connect mobile apps on mobile devices.
18119	Long display names are truncated when there is space available to show the full name.

Connect desktop app release notes

This topic describes the new features, fixed issues and known limitations in the current and previous supported releases of the Connect desktop app:

- [What's new?](#)
- [Fixed issues](#)
- [Known limitations](#)

What's new?

New in v1.12.5

Version 1.12.5 of the Connect desktop app was released in October 2023. This was the latest version at the time of publishing; to check for updates, see [What's new in Connect app?](#)

Below are the new features and changes in v1.12.5:

Feature	Description	More information
Changes in functionality	Previously the Copy meeting link button copied the meeting's <code>pexip://</code> URL to the user's clipboard. Now the button copies both the meeting's <code>https://</code> URL and its <code>pexip://</code> URL, along with text explaining when to use each link.	

New in v1.12.4


Version 1.12.4 of the Connect desktop app was released in June 2023.

Version 1.12.4 contained no new features or changes.

New in v1.12.3

Version 1.12.3 of the Connect desktop app was released in May 2023.

Below are the new features and changes in v1.12.3:

Feature	Description	More information
Changes in functionality		
Run location	Running the Connect desktop app no longer requires access to a "temp" folder, allowing it to be used in environments where running from temp folders is restricted.	
	 All settings are retained on upgrade.	

New in v1.12.2

Version 1.12.2 of the Connect desktop app was released in April 2023.

Version 1.12.2 contained no new features or changes.

New in v1.12.1

Version 1.12.1 of the Connect desktop app was released in March 2023.

Version 1.12.1 contained no new features or changes.

New in v1.12

Version 1.12 of the Connect desktop app was released in February 2023. Below are the new features and changes in v1.12:

Feature	Description	More information
New features		
Citrix virtual apps support	The Connect desktop app for Citrix virtual apps is now generally available. It was a technology preview feature in the previous release.	
Changes in functionality		
Background blur	It is now possible via branding to remove the option to enable or disable background blur from within the Connect app, using the <code>disableBackgroundBlur</code> application setting.	<code>disableBackgroundBlur</code>
Radio buttons replaced with ticks	Under the Advanced settings menu, users now click on a tick (rather than the previous radio button) to enable and disable each setting.	
Accessibility improvements	General improvements to the user interface to increase accessibility.	
Secure check code	The Secure check code used to verify end-to-end encrypted calls now has a copy button, allowing it to be easily shared between participants.	

Fixed issues

Fixed in v1.12.5

Ref #	Resolution
CVE-2023-4863	Updates to address CVE-2023-4863 .

Fixed in v1.12.4

Ref #	Resolution
33101	(Windows client only) Resolved an issue caused when the app was minimized and then restored by double-clicking the taskbar icon, which meant it did not restore correctly.

Fixed in v1.12.3

There were no significant user-facing fixes in this release.

Fixed in v1.12.2

Ref #	Resolution
29955	For Citrix clients, the UI option to view a presentation in a separate window has been removed as this feature is not supported in these clients.

Fixed in v1.12.1

Ref #	Resolution
32814	Resolved an issue when the Connect desktop app for Windows was not already running, whereby clicking a <code>pexip://</code> or <code>pexip-provision://</code> link did not trigger the app to open, and clicking it twice prompted the app to open as a blank screen.

Fixed in v1.12

Ref #	Resolution
31611	The toolbar will now wrap when required to accommodate additional plugin icons.
31495	Resolved an issue with provisioned Connect desktop apps using AD FS SSO for authentication whereby if the app became unregistered, it would need to be re-provisioned in order to re-register.
30498	Resolved an issue whereby settings enabled via branding were not always applied until after the page was reloaded.
30299	Resolved an issue where under some circumstances when a participant is previewing a presentation from a vertical monitor, the previewed image went off screen.
30086	Resolved an issue where in some circumstances the Connect desktop app window would not close.
29899	When changing the layout, the currently selected layout is now indicated.
29887	When an incoming audio-only call is answered with video, the call no longer fails.
29813	Opening an invalid pexip-auth link will no longer result in an error.
29811	Calls to an alias that does not include a hostname will now result in a user-facing error message.
29364	When two or more participants perform the same action on another participant at almost the same time, the second and subsequent participants will no longer see an error.
29339	Participant names in chat messages in end-to-end encrypted calls now use the name provided by IDP authentication.
29130	Resolved an issue whereby when a call from a participant into a Virtual Meeting Room was transferred to an end-to-end encrypted call, the participant may see the spinning animation instead of the welcome screen while waiting for other participants to connect.
25868	Resolved an error with microphone sampling when enabled via branding.

Known limitations

Ref #	Limitation
28799	<p>Call history and preferences data for versions 1.8.x, and 1.9.1 and later of the Connect desktop app for Windows are stored in a folder under C:\Users\<user>\AppData\Roaming called Pexip Infinity Connect. All other versions (i.e. versions 1.7.x and earlier, and v1.9.0) store this data in the same directory in a folder called pexip-infinity-connect.</p> <p>This means that after upgrading from a version that stores data in one folder to a version that stores data in a different folder, call history and preferences will be lost. To work around this, after installation, copy the contents of the previous folder (i.e. pexip-infinity-connect or Pexip Infinity Connect, as applicable) into the new folder.</p> <p>Note that upgrades between versions that use the same folder for storing data are not affected.</p>
18210	The Connect desktop app may not attempt to re-register if it de-registers while minimized.
18119	Long display names are truncated when there is space available to show the full name.
11854	In some cases when sharing Office apps, the Connect desktop app does not capture all portions of the application window. The workaround is to share the entire desktop rather than those specific applications.

Connect mobile app release notes

This topic describes the new features, fixed issues and known limitations in the current and previous supported releases of the Connect mobile apps:

- [What's new?](#)
- [Fixed issues](#)

- [Known limitations](#)

What's new?

New in v1.10.3 mobile app for Android

Version 1.10.3 of the Connect mobile app for Android was released in August 2023. This was the latest Android version at the time of publishing; to check for updates, see [What's new in Connect app?](#)

This release contained no new features.

New in v1.10.1 mobile app for iOS

Version 1.10.1 of the Connect mobile app for iOS was released in March 2023. This was the latest iOS version at the time of publishing; to check for updates, see [What's new in Connect app?](#)

This release contained no new features.

Fixed issues

Fixed in v1.10.3 mobile app for Android

Ref #	Resolution
35049	Resolved an issue with some Android devices that resulted in an "Invalid Call" error.

Fixed in v1.10.1 mobile app for iOS

Ref #	Resolution
32670	Resolved an issue with screen rotation on some iOS devices.

Known limitations

Ref #	Limitation
17072	(Android app only) <code>pexip://</code> links within the app (for example, links sent via the chat window) are not recognized by the app.