# Troubleshooting

- Checking what version of iOS/iPadOS App Portal is being pushed out (15.3+)
- Removing pre-15.3 Kiosk Customizations (macOS/Windows)
- Kiosk with macOS in a VM: Enabling Metal support
- Resolving SSL and Manifest Validation Errors with FileWave Kiosk Installation (15.3+)

# Checking what version of iOS/iPadOS App Portal is being pushed out (15.3+)

## What

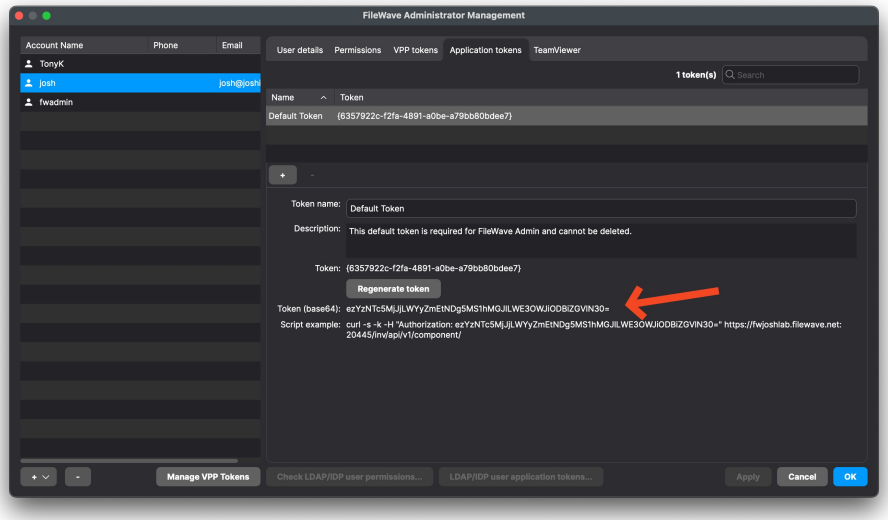
How do I know what version of App Portal (Kiosk) is being pushed out to iOS/iPadOS devices?

## When/Why

After a FileWave upgrade, there likely will be a new version of the iOS/iPadsOS IPA pushed out through the automatic upgrade process. Because this process is a bit invisible, verifying what the server believes is the version it is pushing out is sometimes necessary.

## How

In FileWave Central, go to Assistants -> Manage Administrators and then pick your user account, though depending on permissions, you may want to use fwadmin. Grab the base64 token seen below in the image, including the = at the end of it.




Now you can ask the server. Replace `ezk5ODxxxmE0OyyyyyyyOX0=` below with your token, and replace `support2.filewave.net` with your server's DNS name.

Here is a command you can run in Terminal.app on a macOS system or any system that has curl installed:

```
curl -H "Authorization: ezk5ODxxxmE0OyyyyyyyOX0=" https://support2.filewave.net/filewave/api/kiosk/internal/kiosk-status
```

Here is a PowerShell example of the same command:

```
# Define the URL to make the request
$url = "https://support2.filewave.net/filewave/api/kiosk/internal/kiosk-status"

# Define the headers, including the Authorization token
$headers = @{
    "Authorization" = "ezk5ODxxxmE0OyyyyyyyOX0="
}

# Use Invoke-WebRequest to make the GET request with headers
$response = Invoke-WebRequest -Uri $url -Headers $headers -Method Get

# Output the content of the response
$response.Content
```

The output of the command will be something like below.

{"installed":{"ipa_url":"https://fw-kiosk-v2-ipas.filewave.cloud/15.3.1/App_Portal_15.3.1.ipa","ipa_md5":"81628b83dad72e274ef93ea031db1484","build_date":"2024-03-27T20:08:49.192563Z","bundle_version":"15.3.1"},"latest":{"ipa_url":"https://fw-kiosk-v2-ipas.filewave.cloud/15.3.1/App_Portal_15.3.1.ipa","ipa_md5":"81628b83dad72e274ef93ea031db1484","build_date":"2024-03-27T20:08:49.192563Z","bundle_version":"15.3.1"},"available_versions":["15.3.1"]}%

# Related Content

- Self-Service Kiosk Overview

# Removing pre-15.3 Kiosk Customizations (macOS/Windows)

Prior to FileWave 15.3 you had a method of customizing the Kiosk that is not used in 15.3 and beyond. Some of the customizations will be ignored with 15.3, but there are things you'll want to potentially remove and you may want to stop pushing out customizations that are no longer enforced. This article is a paired down version of the old article and simply contains information on what customizations you might have had in the past so you can look to potentially remove the files/edits.

The look of the FileWave desktop Kiosk was customized using Qt Style Sheets. You would have had to create a file named fwGUI.qss and deploy it to clients in the right location. Several icons were also customizable by placing your custom icon in the right place with the right file name. All of this can be deployed via a Fileset.

Check out Setting the Primary Color, Name and Logo in Kiosk/App Portal (15.3+) and Applications Preventing Reboot (macOS/Windows) for the things that can be customized in FileWave 15.3 and beyond.

# Deploy

fwGUI.qss -- You would have created a Fileset containing the `fwGUI.qss` and the icons, then associate it to the desired clients. The directory where these files was placed depends on the operating system. In 15.3 and beyond you don't want to push the qss file or icons out as they will have no effect on the Kiosk.

### macOS

```
/usr/local/sbin/FileWave.app/Contents/Resources/fwGUI.app/Contents/custom
```

### Windows

```
C:\Program Files\FileWave\custom (might be "Program Files (x86)", depending on the platform)
```

The changes will only be visible in the desktop Kiosk after the user logs out and logs in again.

# Menu Bar/System Tray Settings

settings.ini -- contained generic kiosk ui settings. Especially important is that if `hide_system_tray = false` that you will end up with 2 icons for the Kiosk where one is the old Kiosk if you are on FileWave 15.3.

### Example

```
window_title = "Your New Window Title"
show_top_downloads = true
hide_system_tray = false
```

- The "window_title" attribute was displayed in the menu bar menu item - as well as in the kiosk window title area.
- The "show_top_downloads" attribute was used to hide the top downloads UI on a per client basis.
- "hide_system_tray" determined whether or not there will be an icon displayed in the system tray. Setting 'false' will force the Kiosk to show even with no current association.

> Example Filesets that you might have used to push out these settings:
> Windows (64) - Kiosk_Customizer_for_Windows64.fileset.zip
> Windows (32) - Kiosk_Customizer_for_Windows32.fileset.zip
> macOS - Kiosk_Customizer_for_macOS.fileset.zip
> macOS/Windows - Applications Preventing Reboot

# Customizable icons

You might have included icon files that you used to replace them in Kiosk before 15.3. These icons will not appear in 15.3 or newer.

| File name | Description | Example | Pixel Size |
|---|---|---|---|
| action_back.png | Icon for the "Back" action in the Software Installation dialog |  | 48x48 |
| action_forward.png | Icon for the "Forward" action in the Software Installation dialog | | 48x48 |

| | | | |
|---|---|---|---|
| | |  | |
| background_icon.png | Icon used for the watermark | (FileWave icon) | 512x512 |
| rating_star_half.png | Icon for half a rating star |  | 40x40 |
| rating_star_off.png | Icon for a rating star that is off |  | 40x40 |
| rating_star_on.png | Icon for a rating star that is on |  | 40x40 |
| tray_icon.png | System tray icon. If missing, the background_icon.png is used. | (FileWave icon) | 16x16 |
| warning.png | Warning icon, displayed when a fileset has problems (e.g. missing VPP user) |  | 64x64 |

# Related Content

- Setting the Primary Colour, Name and Logo in Kiosk/App Portal (16.0+)

# Kiosk with macOS in a VM: Enabling Metal support

## Description

The new Kiosk requires 'Metal' support.  VMware does not support metal by default.  It is possible to add support, however the details provided are Beta and not only may the VM become unstable, but the host may also Kernel Panic.

> Use cautiously

Without the necessary configuration, the new Kiosk will not show and the user logs will show something like:

```
2024-05-13 10:40:15.324 FileWave Kiosk[1241:5674] Could not acquire Metal device.
2024-05-13 10:40:25.936 FileWave Kiosk[1241:5674] Unable to create FlutterView; no MTLDevice or MTLCommandQueue
available.
```

## Configuration

The following configuration requires the VM to be shutdown (not paused) prior to editing.

Locate the containing folder of the VM.  Within this will be a .vmx file.  For example, if the containing VM Folder were called 'macOS 12.vmware, the file to edit should be called 'macOS 12.vmx':

```
/Users/sholden/Documents/Virtual\ Machines/macOS\ 12.vmwarevm/macOS\ 12.vmx
```

Add the following two lines:

```
appleGPU0.present = "TRUE"
svga.present = "FALSE"
```

Once added, the VM may then be booted.  To remove this configuration, again shutdown the VM, edit the file, this time removing these two entries.

It may also be necessary to define the aspect ratio in the same file with the following two lines.  1920 x 1080 used as an example:

```
appleGPU0.screenWidth="1920"
appleGPU0.screenHeight="1080"
```

> Although this may work, in practice the VM crashed too often or became seemingly unusable.  Since the host may also Kernel Panic, recommendation would be to avoid configuring this on VMware test devices, unless absolutely necessary.  Even then, consider undoing this once the test has been completed.

# Resolving SSL and Manifest Validation Errors with FileWave Kiosk Installation (15.3+)

## What

This article addresses an issue encountered during the installation of the new FileWave Kiosk v.15.3.1 on macOS and Windows devices. Users may encounter SSL errors and manifest validation failures that prevent the Kiosk from installing correctly.

## When/Why

These installation errors typically occur when upgrading to FileWave v.15.3.1 and are primarily due to conflicts with certain content filters or proxy settings that block or misclassify necessary URLs. This is particularly relevant for organizations using content filtering solutions like Lightspeed, which may categorize essential URLs as unknown, thus blocking them.

```
"InstallApplication     command error           2024-04-22T07:36:38     2024-04-22T07:36:38
Could not validate manifest..An SSL error has occurred and a secure connection to the server cannot be made.
com.filewave.ios.app.kiosk2     "
```

## How

To resolve these installation issues, follow the steps below:

1. Check Proxy and Firewall Settings: Ensure that your organization's proxy or firewall settings are not blocking access to FileWave URLs.
2. Whitelist Necessary URLs: Add the following URLs to the whitelist in your content filter or proxy settings:
   - `https://fw-kiosk-v2-ipas.filewave.cloud/`
   - `*.filewave.cloud`

   This change allows devices to communicate securely with FileWave servers and access the necessary resources for installing the Kiosk.
3. Reattempt Installation: After updating your settings, reattempt the installation of the FileWave Kiosk on the affected devices.

```
# Example command to verify if the URL is accessible from your network
curl -Iv https://fw-kiosk-v2-ipas.filewave.cloud/
```

If the issue persists, check your SSL certificate settings and network configuration for any discrepancies that might be interfering with secure connections.

## Related Links

- Default TCP and UDP Port Usage

## Digging Deeper

The SSL error typically indicates an underlying issue with the secure connection setup between the client device and FileWave servers. This can be caused by SSL certificate verification failures, misconfigured proxies, or stringent network security policies that incorrectly classify or block legitimate URLs required for FileWave operations. Adjusting content filtering policies or proxy settings often resolves these issues, but further investigation into SSL trust settings may be required for complex network environments.