

# Certificates

Certificates add a layer of security and trust between devices. Below are some articles that discuss how various certificates are used within FileWave.

- [Root Trusted SSL Certificate \(Using and Renewing\)](#)
- [Let's Encrypt Setup for FileWave Server \(Debian\)](#)
- [APNs](#)
  - [Apple Push Notification Service](#)
  - [APNs Certificate Creation & Renewal on macOS Computers](#)
  - [APNs Certificate Creation & Renewal on Windows Computers](#)
- [Self-signed SSL Certificates](#)
  - [Self-Signed SSL Certificates Going Forward](#)
  - [Renew FileWave Server Self-signed Certificate](#)
  - [Renew MDM self signed SSL certificate with iOS devices](#)
- [Troubleshooting](#)
  - [Export .p12 SSL Certificate from Windows](#)
  - [Determine Correct Intermediates Bundle for SSL Certificate](#)
  - [Self Signed Certificate Error during iOS OTA Enrollment](#)
  - [SSL Server Certificates - iOS 13 and macOS 10.15](#)

# Root Trusted SSL Certificate (Using and Renewing)

## Description

To communicate with devices, a certificate is required. Our recommendation is for a root-trusted SSL certificate to be implemented. If you are currently using a self-signed certificate, we suggest moving to a trusted root certificate; wildcard certificates are supported. This article will discuss both self-signed as well as a certificate from an authority, and the process to renew the certificate.

Since Filewave v12+, the Admin console indicates when a certificate is self-signed; Preferences > Mobile tab > HTTPS Certificate Management

Warning: You are using a self-signed certificate. Your devices will need to trust the certificate to use FileWave. iOS 10.3+ devices require explicit trust before manual enrollment.

Details...

Upload PKCS12 Certificate

Get Current Certificate

iOS 10.3+ devices can be enrolled with a self-signed certificate. Over the Air (OTA) though will experience warnings or errors; [Self Signed Certificate Errors](#).

Apple provide a list of certificates automatically trusted per OS and OS version: [Apple Trusted Certs](#)

## Information

Root trusted SSL certificates can be purchased from a Certificate Authority (CA). Apple provides lists of trusted root certificates: <https://support.apple.com/en-gb/HT204132>

CA Vendors include:

- GoDaddy
- Digicert
- GlobalSign / AlphaSSL
- Trustwave
- and many more...

As FileWave supports wildcard certificates, if you already have a wildcard certificate this could be uploaded without additional purchase. Wildcard certificates are indicated by a \* before the domain name. e.g.

- Wildcard cert: \*.initech.com
- Dedicated cert: filewave.initech.com

If you already have a certificate bundle but it isn't in the .p12 format, you can use this link to convert the file - [Digicert: How to convert a certificate into the appropriate format](#)

## Requirements

Obtaining an official 3rd party root trusted SSL certificate will be dependent upon the Server's current domain. Only a domain that includes an official 'Top Level Domain' (TLD) may qualify for a root trusted SSL certificate and the root domain must be registered to purchase a certificate. Where the Server uses an internal-only domain, it is not possible to transition to an official certificate without first changing the domain where the Server belongs. See migration below. Example TLD:

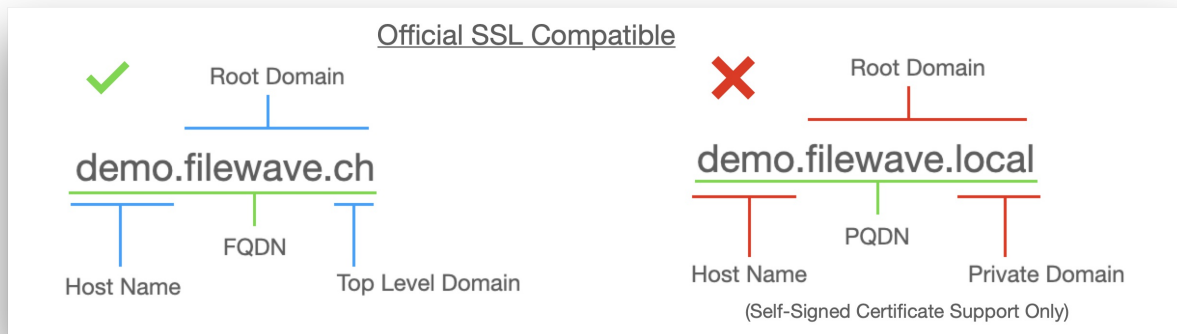
- .com
- .org
- .edu

In the context of a Website, the root domain refers to the highest level of the hierarchy, e.g apple.com, microsoft.com, google.com

For example:

- demo.filewave.ch is a Fully Qualified Domain Name (FQDN) of a server called 'demo' in the root domain 'filewave.ch', the TLD being 'ch'.
- demo.filewave.local does not have a TLD and instead is using a private internal domain of '.local'. Without a TLD they are known as a Partially Qualified Domain Name (PQDN)

Often the term FQDN is used as a way to indicate the idea of writing the Server name along with its connected domain. Strictly speaking, with internal private-only domains, this should be referred to as PQDN. Not only does FileWave recommend using an FQDN, but regardless of using an FQDN or PQDN, always specify the Server name along with its connected domain name (rather than just the hostname) when setting any preferences, be that for Server, Clients, Boosters, etc. We also discourage the use of IP in settings.



## Migration

Migration of certificates will not pose any issues as long as:

- Wildcard cert: domain matches the domain name of the previous self-signed certificate
- Dedicated cert: Server name and domain matches the name of the previous self-signed certificate

**i** If during migration the Server's Host Name and/or Domain Name changes, all MDM devices will lose MDM communication with the FileWave Server and require re-enrolment into MDM

Case	Current Certificate	New Certificate	Result
Any certificate to any certificate (changing name)	Self-signed cert = <code>filewave.initech.com</code>	Root trusted cert = <code>fw.initech.com</code>	CHANGING THE FQDN WILL REQUIRE DEVICES TO BE ENROLLED AGAIN
Self-signed to root trusted (keeping the same name)	Self-signed cert = <code>filewave.initech.com</code>	Root trusted cert = <code>filewave.initech.com</code>	This will NOT require devices to be enrolled again
Self-signed to wildcard	Self-signed cert = <code>filewave.initech.com</code>	Wild Card cert = <code>*.initech.com</code>	This will NOT require devices to be enrolled again
Root trusted to root trusted	Root trusted cert = <code>filewave.initech.com</code>	Root trusted cert = <code>filewave.initech.com</code>	This will NOT require devices to be enrolled again

## Procedure

There are 3 key steps.

1. Create a CSR and Key to request a certificate from a CA
2. Create CRT files from the downloaded certificates
3. Convert the certificate to p12 to upload to the FileWave Server

### Renewing Certificates

When renewing a current expiring certificate with a CA, step 1 is not required. You will however require the key in step 3. If you have not stored the key elsewhere, the key should always be accessible on your current FileWave Server in

**i** `/usr/local/filewave/certs/`.

If you have not stored the key safely and the Server was to break such that the key was not retrievable, the whole process would need to be repeated instead.

### Certificate Expiry

**i** Certificate expiry should be avoided. Renewing certificates should be done in advance to maintain full working order. If the certificate expires before you have a chance to renew, managed devices will not be able to connect to the FileWave Server. However, once it is renewed your devices will check back in.

## Example process

For example:

- FileWave Server FQDN = **fw.initech.com**
- Files will be saved to created folder Certificates
- The certificate was purchased from AlphaSSL

Pre-requisite: OpenSSL. Unix-based systems have this by default. To follow this process on Windows will require an appropriate version of [OpenSSL](#)

## Step 1

Create the CSR and . KEY from OpenSSL

From a command prompt type the following:

macOS and Unix

```
sudo openssl req -new -newkey rsa:2048 -nodes -keyout /certificates/fw.initech.com.key -out /certificates/fw.initech.com.csr
```

Windows

```
C:\OpenSSL-Win64\bin\openssl.exe req -new -newkey rsa:2048 -nodes -keyout C:\certificates\fw.initech.com.key -out C:\certificates\fw.initech.com.csr
```

You will be prompted for the following:

- Country Name (2-letter code)
- State or Province Name Locality Name (eg, city)
- Organization Name (eg, company)
- Organizational Unit Name (eg, section)
- Common Name (e.g. Server FQDN or YOUR name)
- Email Address
- A challenge password
- An optional company name

For this example the details should be:

- Common Name: **fw.initech.com**
- Do not enter a password

The Certificates folder should now show:

```
fw.initech.com.csr  
fw.initech.com.key
```

The KEY should be held safely. The CSR will need to be uploaded to the CA during the request of the certificate creation. You should receive confirmation from the CA, regarding domain ownership and how to retrieve the generated certificate along with some general instructions.

## Step 2

Create CRT files from the downloaded certificates

It is typical, that the SSL certificate will also require an intermediate certificate. These should be readily available from the CA's website. If required, contact the CA for details of which intermediate you will require.

Once the SSL and intermediate certificate have been downloaded, instructions can be followed to create the CRT files. In the case of the email from AlphaSSL, only steps 1-4 should be followed.

Sample email from AlphaSSL.



## QUICK INSTALLATION GUIDE

---

1) Using a text editor, copy the SSL Certificate text from the bottom of this email (including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----lines) and save it to a file such as yourdomain.txt

2) Retrieve the Intermediate Certificate (selecting SHA-1 or SHA-256 as appropriate) from the Support Center at:

<https://www.alphassl.com/support/install-root-certificate.html>

3) Using a text editor, copy the Intermediate Certificate text (including the --BEGIN CERTIFICATE----- and -----END CERTIFICATE----- lines) and save it to a file such as intermediate\_domain\_ca.txt

4) Copy these .txt files to your server and then rename them with .crt extensions

5) Install the Intermediate and SSL Certificates

6) Restart your server

7) To test for installation errors please use our SSL Configuration Checker Tool located at: [https://sslcheck.globalsign.com/en\\_US](https://sslcheck.globalsign.com/en_US)

8) Install your Site Seal with the instructions show at:

<http://www.alphassl.com/support/ssl-site-seal.html>

9) We suggest you back-up your SSL Certificate and Private Key pair and keep it safe, all IIS users can use the Export Wizard

We hope that your application process was quick and easy and you have enjoyed the AlphaSSL experience.

Thank you for choosing AlphaSSL, if you have any questions or issues please do not hesitate to contact us.

These 2 CRT files can be copied to the Certificates folder from Step 1.

From the example:

- SSL certificate: purchasedcert.crt
- Intermediate certificate: AlphaSSLCA.crt

The certificates folder should now show:

```
AlphaSSLCA.crt  
fw.initech.com.csr  
fw.initech.com.key  
purchasedcert.crt
```

## Step 3

Convert the certificate to p12 to upload to the FileWave Server

The necessary files are now available to create the p12. From the command line type the following:

macOS/Linux

```
sudo openssl pkcs12 -export -out /certificates/fw.initech.p12 -inkey /certificates/fw.initech.key -in  
/certificates/purchasedcert.crt -certfile /certificates/AlphaSSLCA.crt
```

```
C:\OpenSSL-Win64\bin\openssl.exe pkcs12 -export -out C:\certificates\fw.initech.com.p12 -inkey  
C:\certificates\fw.initech.com.key -in C:\certificates\purchasedcert.crt -certfile C:\certificates\AlphaSSLCAS.crt
```

The p12 certificate can be uploaded to the Server through the Admin console: Preferences > General Tab > SSL Certificate Management. Once uploaded, check the 'Common Name' on the General tab matches the Server name in the Mobile Tab. For wildcard certificates, only the domain should match.

Apache web server service will automatically restart and the FileWave Server is now ready for MDM.

## Command Overview

### Explanation of OpenSSL command

OpenSSL pkcs12 #Create a p12 (also know as pkcs12)

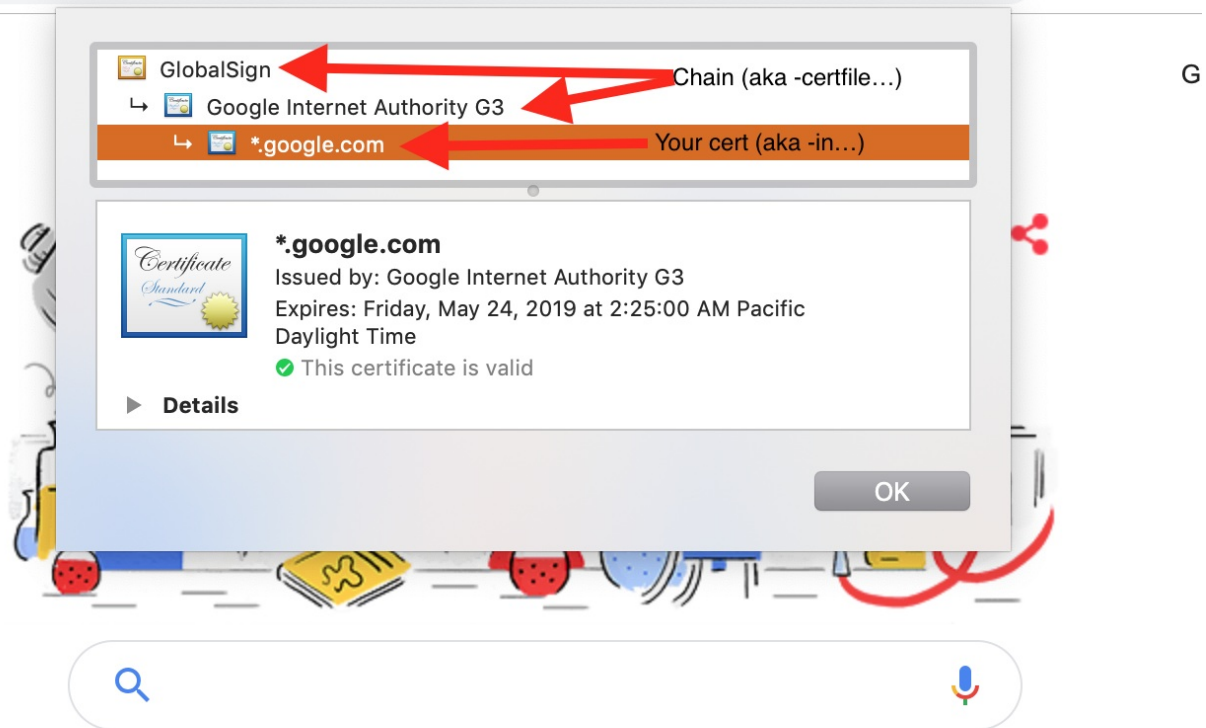
-export -out /certificates/fw.initrode.us.p12 #Output location and name of p12 to upload to FileWave

-inkey /certificates/fw.initrode.us.key #Location and name of the private key file used to generate the CSR

-in /certificates/purchasedcert.crt #Location and name of the purchased certificate provided by the 3rd party supplier

-certfile /certificates/AlphaSSLCA.crt #Location and name of Intermediate certificate, (will often contain more than one cert)

.google.com/?safe=active&ssui=on



# Let's Encrypt Setup for FileWave Server (Debian)

## What

This Knowledge Base (KB) article discusses the use of a shell script to automate the process of setting up Let's Encrypt SSL certificates for a FileWave server. Let's Encrypt is a free, automated, and open Certificate Authority (CA) that provides SSL certificates required for secure (HTTPS) connections.

## When/Why

FileWave administrators would need to set up an SSL certificate for the FileWave server to ensure secure communication. Using Let's Encrypt is a popular choice because it's free and can be automated. However, the setup process has multiple steps and can be prone to human error. This script simplifies the process by automating most of the steps.

**Note that this documented process is for Debian systems. It could be adapted for macOS.**

## How

The script needs to be run on a FileWave server with root permissions. The server should be Debian 12+ or compatible.

Here are the steps to use the script:

1. Download the script below.
2. Make the script executable by running `chmod +x scriptname.sh`.
3. Run the script as root with `--install` as the argument.  
Example: `sudo ./scriptname.sh --install`
4. The script will automate the rest of the process. It will:
  - Check if the provided FQDN is resolvable.
  - Backup any existing certificates.
  - Install necessary tools (if not already installed).
  - Request a new certificate from Let's Encrypt.
  - Set up a script to automatically renew the certificate and apply it to the FileWave server.
  - Set up a daily cron job to renew the certificate.
5. Check the output of the script to ensure there were no errors during the process.
6. If you want to remove Let's Encrypt you can do the following and then go in to FileWave Central and replace your certificate with one from a different source.  
Example: `sudo ./scriptname.sh --uninstall`

**Please replace `scriptname.sh` with the actual name of the script.**

Download Script: [linux-letsencrypt-v2.0.sh](#)

### ▼ linux-letsencrypt-v2.0.sh (Debian)

```
#!/bin/bash
#26-May-2022 - Removed the old update dep certs file that would break a 14.7.x server updating its dep
profiles
#19-June-2020 - added parameter verification ; made DEP script injection conditional ; added firewall
exceptions ; made sure certificate is injected at initial run ; added cronjob scheduling
#10-July-2023 - Updated the script to replace the old renewal hook if it exists, made the script work with --
uninstall without additional arguments, changed the script to ask for the hostname and email during
installation, added Debian OS check.
#16-Feb-2024 - Updated for Debian server
#08-Mar-2024 - Refinements for Debian use. Switched to SNAP for certbot and more help if errors.
#23-Apr-2024 - Added code to set the cert as trusted

scriptname="$BASH_SOURCE"

# check if script is running on Debian
if ! grep -q 'Debian' /etc/os-release ; then
echo "This script is designed to run on Debian. Exiting."
exit 1
fi
```

```

# root or moot
if [ $(whoami) != "root" ] ; then
echo "root rights required - please rerun as"
echo "sudo ${BASH_SOURCE} --install"
exit 1
fi

#letsencrypt installation for Debian Linux
if [ -z "$1" ]; then
    echo "This script automates the process of obtaining and installing a Let's Encrypt SSL certificate for a
FileWave server on Debian."
    echo "Available arguments:"
    echo "--install: Install a new Let's Encrypt SSL certificate."
    echo "--uninstall: Uninstall the Let's Encrypt SSL certificate and remove associated scripts."
    exit 1
fi

# Check if uninstall switch is provided
if [ "$1" == '--uninstall' ]; then
    echo "Uninstalling..."
    rm -f /etc/letsencrypt/renewal-hooks/deploy/filewave-server-cert.sh
    rm -f /etc/cron.daily/letsencrypt-filewave
    snap remove certbot
    echo "Uninstallation complete. The renewal hook script, daily cron job, and Certbot have been removed."
    exit 0
fi

# Proceed with installation
if [ "$1" == '--install' ]; then
    read -p "Enter your fully qualified domain name (hostname): " hostname
    read -p "Enter your email address: " emailaddress

    # verify whether this is a resolvable public hostname ; abort if not
    if ! [ -x "$(command -v nslookup)" ]; then
        echo "Installing 'dnsutils' for Debian."
        apt install -y dnsutils
    fi

    nslookup $hostname 8.8.8.8
    public=$?
    if [ $public -ne 0 ] ; then echo "$hostname not resolvable by 8.8.8.8 (Google) ; exiting." ; exit 0 ; fi

    # Confirm hostname and email address
    echo "Hostname: $hostname"
    echo "Email Address: $emailaddress"
    read -p "Are these correct? (yes/no) " confirm
    if [ "$confirm" != "yes" ]; then
        echo "Aborting due to user confirmation."
        exit 1
    fi

    #Backup existing certs
    BACKUP_DATE=$(date +%Y-%m-%d-%H-%M)
    mkdir /usr/local/filewave/certs/backup-$BACKUP_DATE
    cp -p /usr/local/filewave/certs/server.* /usr/local/filewave/certs/backup-$BACKUP_DATE

    # Update Debian system and Install Certbot if not already installed
    apt update -y
    DEBIAN_FRONTEND=noninteractive apt-get upgrade -y -o Dpkg::Options::="--force-confold"

    if ! [ -x "$(command -v certbot)" ]; then
        echo "Installing SNAP."
        apt install -y snapd
        snap install core
        echo "Installing Certbot."
        apt remove certbot python3-certbot-apache
        snap install --classic certbot
        ln -s /snap/bin/certbot /usr/bin/certbot
        # Test again for the certbot command

```

```

        if ! [ -x "$(command -v certbot)" ]; then
            echo "Certbot installation failed. Exiting."
            exit 1
        fi
    fi

    #request certificate initially
    # I've seen this fail and there is no harm in the command running 2x the first time to be sure it worked.
    certbot -n --agree-tos --standalone certonly -d "$hostname" -m "$emailaddress"

    echo "Certificate for $hostname should be created. If there was an error try running:"
    echo 'certbot -n --agree-tos --standalone certonly -d "'$hostname'" -m "'$emailaddress''

# PostgreSQL command to update ios_preferences
echo "Updating iOS preferences in PostgreSQL..."
/usr/local/filewave/postgresql/bin/psql -d mdm -U django -c "INSERT INTO ios_preferences (key, value) VALUES ('mdm_cert_trusted', TRUE) ON CONFLICT (key) DO NOTHING; UPDATE ios_preferences SET value = 'true' WHERE key = 'mdm_cert_trusted';"
if [ $? -eq 0 ]; then
    echo "iOS preferences updated successfully."
else
    echo "Failed to update iOS preferences in the database."
fi

# Ensure renewal-hooks and deploy directories exist
mkdir -p /etc/letsencrypt/renewal-hooks/deploy

# Remove old renewal hook if exists
rm -f /etc/letsencrypt/renewal-hooks/deploy/filewave-server-cert.sh

# Create new renewal hook
cat>/etc/letsencrypt/renewal-hooks/deploy/filewave-server-cert.sh<<EOT
#!/bin/bash
cp /etc/letsencrypt/live/$hostname/fullchain.pem /usr/local/filewave/certs/server.crt
cp /etc/letsencrypt/live/$hostname/privkey.pem /usr/local/filewave/certs/server.key
chown apache:apache /usr/local/filewave/certs/server.*
echo "Restarting FileWave Server..."
/usr/local/bin/fwcontrol server restart
echo "Updating DEP profile certificates..."
(yes 2>/dev/null) | /usr/local/filewave/python/bin/python /usr/local/filewave/django/manage.pyc
update_dep_profile_certs 2>/dev/null
echo "DEP profile certificates updated."
EOT
    chmod a+x /etc/letsencrypt/renewal-hooks/deploy/filewave-server-cert.sh

    echo "injecting certificate into filewave server"
    /etc/letsencrypt/renewal-hooks/deploy/filewave-server-cert.sh
    echo "injection done"

##Install firewalld for configuring port 80
#   if ! [ -x "$(command -v firewalld)" ]; then
#       echo "Installing firewalld."
#       apt install -y firewalld
#       systemctl enable --now firewalld
#       # Test again for the firewalld command
#       if ! [ -x "$(command -v firewalld)" ]; then
#           echo "firewalld installation failed. Exiting."
#           exit 1
#       fi
#   fi

#   echo "making firewall opening for port 80 permanent to allow for automatic renewals"
#   firewall-cmd --add-port 80/tcp --zone=public --permanent
#   firewall-cmd --reload
#   echo "firewall settings modified"

    echo "scheduling automatic renewal of certificate"
    cat>/etc/cron.daily/letsencrypt-filewave<<'EOT'
#!/bin/bash
sleep ${RANDOM % 7200}

```

```

/usr/bin/certbot renew --quiet
EOT
  chmod a+x /etc/cron.daily/letsencrypt-filewave
  echo "scheduling complete"

  echo " "
  echo "-----"
  echo "List of files created by this script:"
  echo "/etc/letsencrypt/renewal-hooks/deploy/filewave-server-cert.sh"
  echo "/etc/cron.daily/letsencrypt-filewave"
  echo "Certificate for $hostname should be created. If there was an error try running:"
  echo 'sudo certbot -n --agree-tos --standalone certonly -d "$hostname" -m "$emailaddress"'
  echo 'And upon success run:'
  echo "sudo certbot renew --force-renewal"
  echo "The main reason this process can fail is if the server is not reachable on TCP 80"
else
  echo "Invalid argument. Please run with --install or --uninstall."
fi

```

## Troubleshooting

**1** Please note that using Let's Encrypt requires your server to be reachable from the internet on port 80, as Let's Encrypt uses the HTTP-01 challenge to verify your server's identity.

If you see errors when it tries to do the below command that registers with Let's Encrypt or if you just see that there is no active cert you can run the command below again. Just replace \$hostname with fwjoshlab.filewave.net if that is your server, and \$emailaddress with your email. Keep the " marks. It'll tell you if it is successful or already was previously successful.

Once you have success with registering then running the renew will tell the FileWave Server that the cert is updated.

```

sudo certbot -n --agree-tos --standalone certonly -d "$hostname" -m "$emailaddress"
sudo certbot renew --force-renew

```

If you have noticed that on your <https://server:20443> you see 2 options where 1 is to download a certificate then you may have used an old version of this script. The current script handles this. The fix is to do the below in a Terminal session on the server:

```

/usr/local/filewave/postgresql/bin/psql mdm diango
insert into ios_preferences values('mdm cert trusted', TRUE);
update ios_preferences set value='true' where key='mdm_cert_trusted';
\q

```

An image of what this looks like:

```

admin@ip-172-30-3-220:~$ /usr/local/filewave/postgresql/bin/psql mdm django
psql (12.18)
Type "help" for help.

mdm=# insert into ios_preferences values('mdm_cert_trusted', TRUE);
INSERT 0 1
mdm=# update ios_preferences set value='true' where key='mdm_cert_trusted';

UPDATE 1
mdm=# \q

```

## Related Links

- [Let's Encrypt Documentation](#)
- [GitHub - nycon/filewave-installer: Filewave AIO installer](#)
- [Review My Notes: FileWave and Let's Encrypt | Version 12.0 \(punkstuff.com\)](#)

# APNs

MDM/DDM communication relies upon Apple's APNs cloud service.

# Apple Push Notification Service

## What

- Like to know a new message has been sent?
- Want to see how many messages are unread from the Home Screen, per App?

✓ The following is really just for information, describing APNs.

Push Notifications are mostly designed to allow 3rd party Apps the ability to inform users through their App, e.g. messages, sounds, etc. some relevant detail. Users control which messages are silenced or visible and how they are visible through Settings.

Developers of Apps requiring this service register their App with Apple. This process requires an APNs token, integrated into the App's Server.

! Generation of an APNs token itself is a required action by FileWave Admins as per the other KB articles in this chapter.

For APNs to succeed, the App and 3rd party server must be able to trust Apple's APNs Cloud Service. Hence, Trust Stores must include Apple's APNs Root Certificate.

## APNs Certificate Update:

At times the Root Certificate used by APNs will require replacing, prior to expiry.

APNs Cert	Service	Up to Date	From Date	Expiry Date
<a href="#">AAA Certificate Services root certificate</a>	Sandbox	Jan 2025	-	Dec 31 23:59:59 2028 GMT
	Production	Feb 2025	-	
<a href="#">SHA-2 Root : USERTrust RSA Certification Authority certificate</a>	Sandbox	-	Jan 2025	Jan 18 23:59:59 2038 GMT
	Production	-	Feb 2025	

Apple will supply information when this occurs, ensuring developers of Apps and providers of 3rd party servers update their products.

✓ FileWave Server already includes both of the above listed certificates within its Trust Store.

## 3rd Party Apps

The act of installing an App requiring APNs, registers that App with APNs and the device receives a Unique Device Token.

Messages pushed can include:

- Display Alert Message to User
- Apply Badge Icon to App's Icon
- Play a Sound
- Deliver Notification Silently

Both Message and Unique Device Token are sent by the App's Server when attempting to initiate a notification.

Notifications are relayed through Apple's APNs service. On receipt of the notification, the device will act accordingly, e.g. display a message to user.

In essence, the message payload therefore consists of:

- APS Dictionary: Message content
- Alert Keys: Assist notification processing, e.g. an identifier to a particular conversation of a messaging app.
- Device ID: Unique Device Token

! The App should contain the current APNs Root Certificate within its Trust Store

## MDM/DDM

MDM communication also relies upon the APNs service and therefore is an example of this process, but key aspects are:



- The act of enrolment is equivalent to installing the App, initiating the receipt of the Unique Device Token.
- The App in question is a binary, included in the Operating System by Apple: '/usr/libexec/mdmclient'.
- APS dictionary should not be included in the payload from an MDM server.

MDM APNs messages are nothing more than a request for the device to contact the MDM server. Any commands are subsequently sent directly to the device, once the device responds back to the MDM server from this APNs request.

Since Apple are the developers of the 'mdmclient', Apple manage its Trust Store. Apple's list of supported Root Certificates per OS version are available from their KB:

<https://support.apple.com/en-gb/103272>

# APNs Certificate Creation & Renewal on macOS Computers

## Description

Apple Mobile Device Management (MDM) requires an Apple Push Notification service (APNs) certificate; renewable yearly.

- APNs Expiry  
If APNs certificates are allowed to expire, all MDM communication will be lost, until renewed.

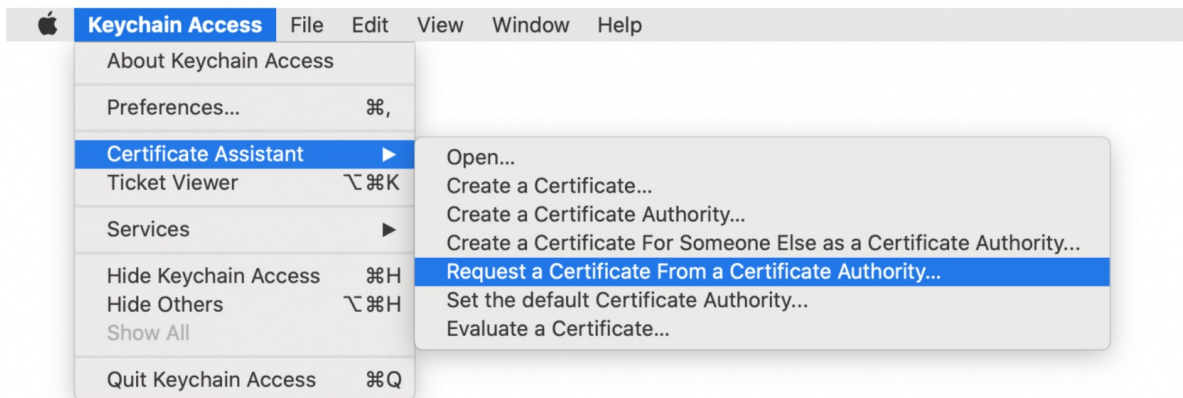
The following guide provides the steps to create and renew an APNs certificate using macOS.

- APNs Topic  
An APNs certificate has a unique topic, in the form of a hexadecimal string, and belongs to the Apple ID used to create the certificate. When renewing, the topic must match to ensure devices continue to communicate with the server. As such, not only must the same Apple ID be used when renewing an APNs certificate, but the current certificate must also be selected for renewal.

## Step-By-Step Guide

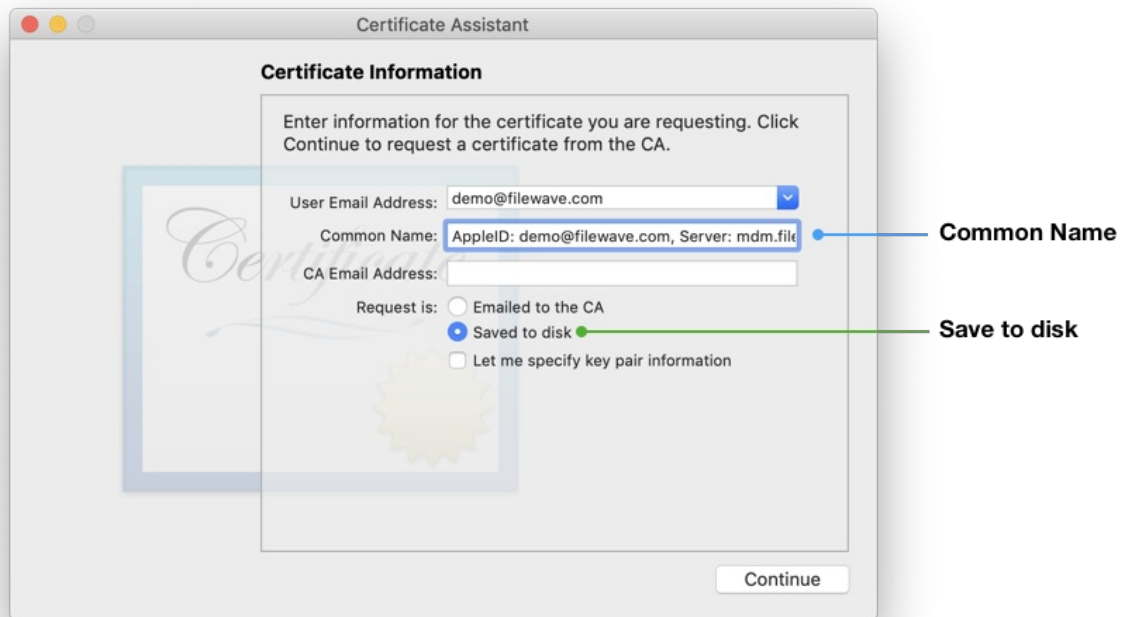
### Creating the Certificate Signing Request (CSR)

1. Open Keychain Access, located in: Applications > Utilities > Keychain Access.app.
2. Create a CSR. Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority...



3. Enter the AppleID and Server name that you are going to be associating with this certificate in the "Common Name" field.

- Common Name  
Certificate Private Key names are visible in Keychain and the Common Name is used to set the Private Key name. Supplying the Apple ID and Server as the Common Name, ensures the Apple ID used to generate the certificate will be stored for future reference.




4. Select the radio button "Saved to disk" and click Continue.
5. Save the CSR request, ready to upload to FileWave in the next section.

**Certificate Storage**  
 Consider creating a secure location to store the created certificates and sub divide them using the date or year, e.g folder named: 'MDM APNs certificates 2020'.

## Sign the CSR


CSR requests must be signed before uploading to Apple. FileWave has a portal for this process, which requires an active FileWave account.

1. Navigate to <https://csr.filewave.com/> and login.
2. Upload the previously created CSR.
3. 'Download signed CSR' should list this uploaded and now signed CSR.
4. Download this newly signed CSR, ready for upload to Apple in the next section. Again consider where this certificate is stored.



Multi-platform management

Contact Sales |



PRODUCTS

SOLUTIONS

SERVICES

SUPPORT


PARTNERS

NEWS

EVENTS

ALLIANCE

STAFF

 search...

# Signed CSR list

Original CSR filename	CSR Upload Date	Download signed CSR
CertificateSigningRequest.certSigningRequest	July 16, 2012, 2:18 p.m.	<a href="#">Download</a>
CertificateSigningRequest.certSigningRequest	Aug. 5, 2014, 11 a.m.	<a href="#">Download</a>

## Upload the signed FileWave CSR to Apple

### Creating a new Certificate

If you are renewing a certificate then jump to [Renewing a Certificate](#)

1. Navigate to: <https://identity.apple.com/pushcert/> and log in with an Apple ID.  
 This Apple ID will own the certificate and is required for every renewal. Do not use a personal Apple ID, to avoid complications if that person where to leave the business or institution.

2. Click 'Create'.
3. 'Accept' Apple's 'Terms of Use'.

## Apple Push Certificates Portal

### Create a New Push Certificate

Upload your Certificate Signing Request signed by your third-party server vendor to create a new push certificate.

Notes

Vendor-Signed Certificate Signing Request

no file selected

### Renewing a Certificate

1. Navigate to: <https://identity.apple.com/pushcert/> and log in with the Apple ID used to initially create the certificate.
2. Confirm the Certificate to renew.
3. Select 'Renew'.

To confirm the certificate, compare the Subject DN (Topic) and current certificate.

Clicking the 'i' button will show the certificate details, including the Topic:

Apple Push Certificates

Serial Number : b4555371ea21ea2

Subject DN : C=GB, CN=APSP:bb78e23d-9b51-4d83-ef5d-dd92a43b0930, UID=com.apple.mgmt.External.bb78e23d-9b51-4d83-ef5d-dd92a43b0930

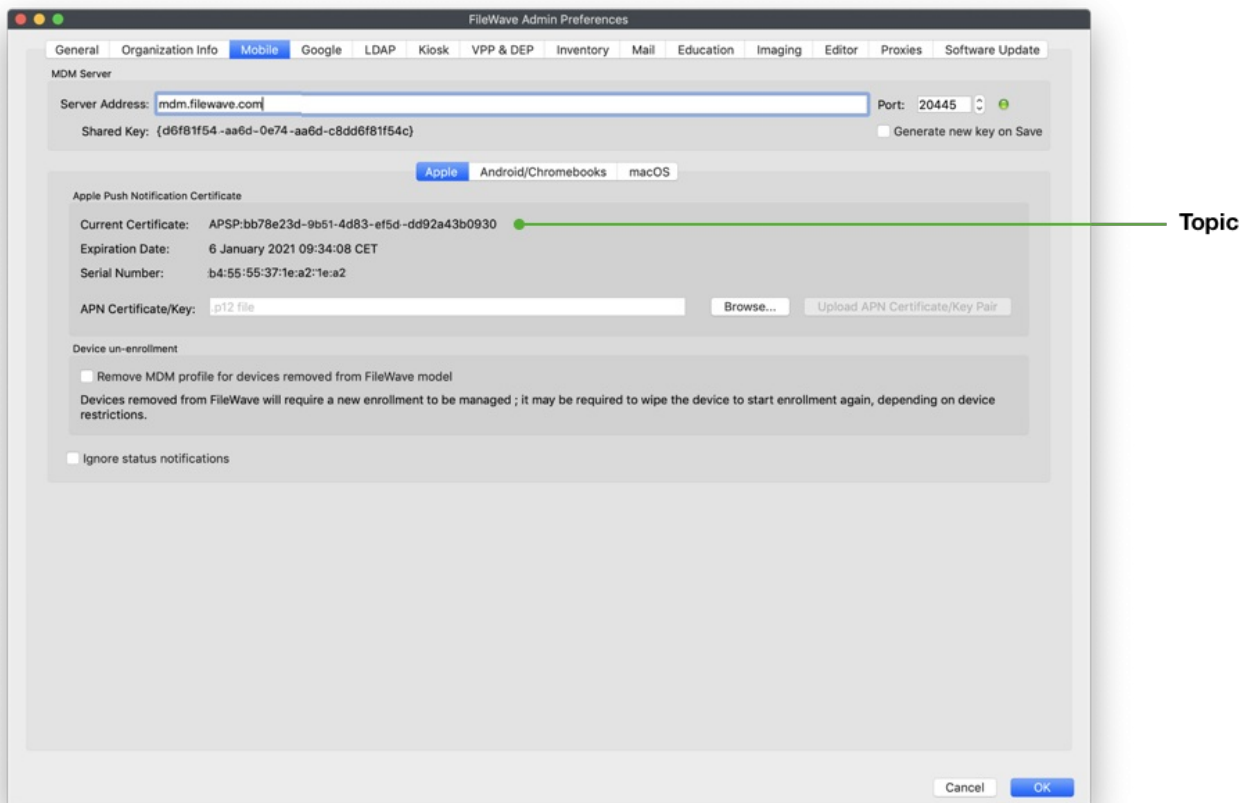
Notes :

Service	Vendor	Date	Status	Actions
Mobile Device Management	FileWave (Europe) GmbH	Jan 6, 2021	Active	<input type="button" value="Renew"/> <input type="button" value="Download"/> <input type="button" value="Revoke"/>

**Topic** (points to Subject DN)

**Info** (points to 'i' icon)

Ensure this matches with the 'Current Certificate' in FileWave Admin > Preferences > Mobile > Apple Push Notification Certificate:



The image shows the 'FileWave Admin Preferences' window with the 'Mobile' tab selected. The 'MDM Server' section contains a 'Server Address' field with 'mdm.filewave.com' and a 'Port' dropdown set to '20445'. Below this is a 'Shared Key' field with a long alphanumeric string. The 'Apple Push Notification Certificate' section is active, showing details for a current certificate: 'Current Certificate: APSP:bb78e23d-9b51-4d83-ef5d-dd92a43b0930', 'Expiration Date: 6 January 2021 09:34:08 CET', and 'Serial Number: b4:55:55:37:1e:a2:1e:a2'. There is a 'Browse...' button for the 'APN Certificate/Key' and an 'Upload APN Certificate/Key Pair' button. The 'Device un-enrollment' section has a checkbox for 'Remove MDM profile for devices removed from FileWave model' and a checkbox for 'Ignore status notifications'. A green line points from the word 'Topic' to the 'Current Certificate' field.

**!** If the 'Topics' do not match do not continue. If the correct certificate is not in the list on Apple's website, this is the wrong Apple ID. If this guide was followed in creating the original certificate, the previously used Apple ID will be viewable from the certificate "Private Key".

Click 'Choose File' and browse to the signed FileWave CSR from the previous section.

Click 'Upload' and Apple will return a 'Confirmation'.

## Confirmation

You have successfully created a new push certificate with the following information:

Service	Mobile Device Management
Vendor	FileWave (Europe) <u>GmbH</u>
Expiration Date	Jan 6, 2021

Manage Certificates
Download

Click 'Download' and save the ".pem" file. Again consider where this certificate is stored.



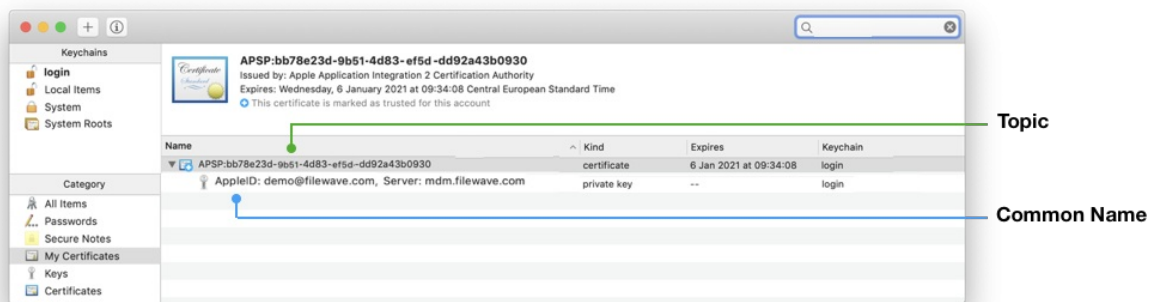
## Create a ".p12" from the Signed CSR

1. Open Keychain Access app, select login from the Keychains list and then choose 'My Certificates' tab.

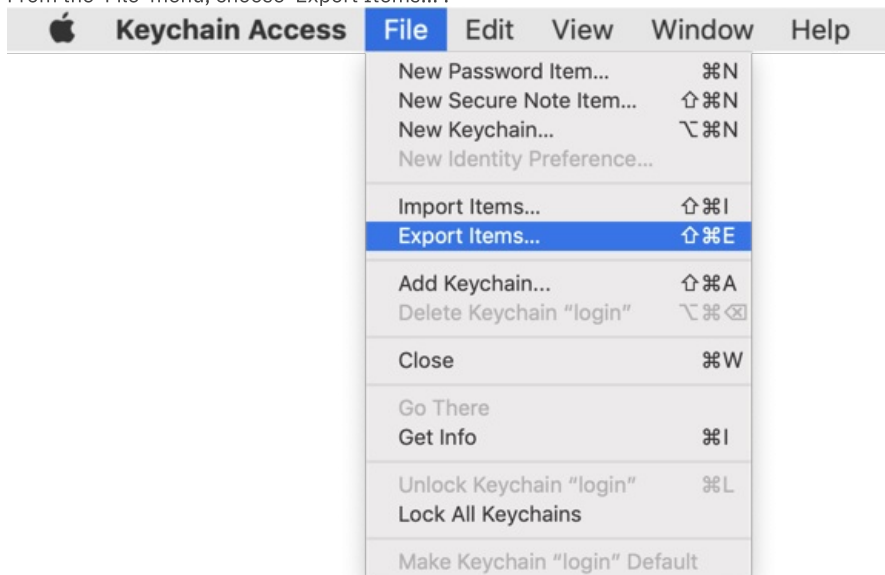
**Keychain**  
 ⚠ If imported into the System Keychain, the Private Key will not be accessible. If 'All Items' tab is selected, private keys will not be available!

2. Drag the downloaded PEM file into the Keychain main window.
3. Locate the imported certificate. It will begin with "APSP:".
4. Click the disclosure triangle and select the expanded private key.

**Common Name and Topic**  
 ✓ The name of the Private Key will show the value defined as the "Common Name" from the creation of the CSR. Where recommendation was followed, this should list the Apple ID and Server name. Additionally the name of the Certificate is the same as the Topic.





5. From the 'File' menu, choose 'Export Items...'



6. Export as a .p12 file. Again consider where this certificate is stored.
7. Click Save.


Save As:


Tags:

Where:   

File Format:

8. Leave the password blank.


 **Enter a password which will be used to protect the exported items:**

Password:  

Verify:

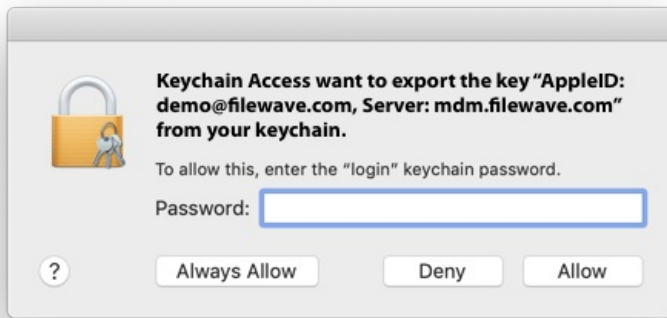
[Password Strength:](#) Weak

☐ Show password



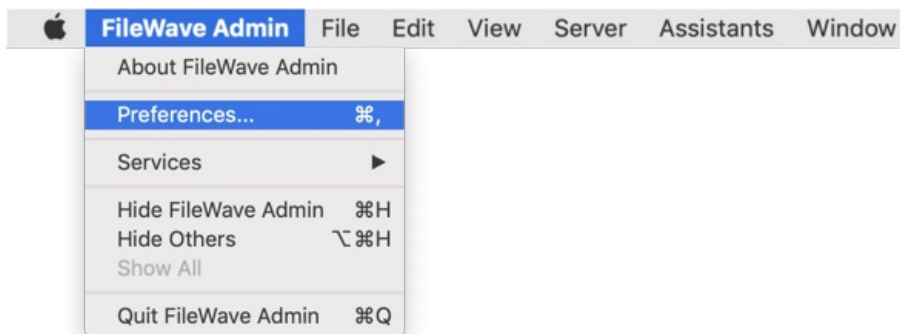
9. Enter your local admin account, when prompted, allowing Keychain to export.



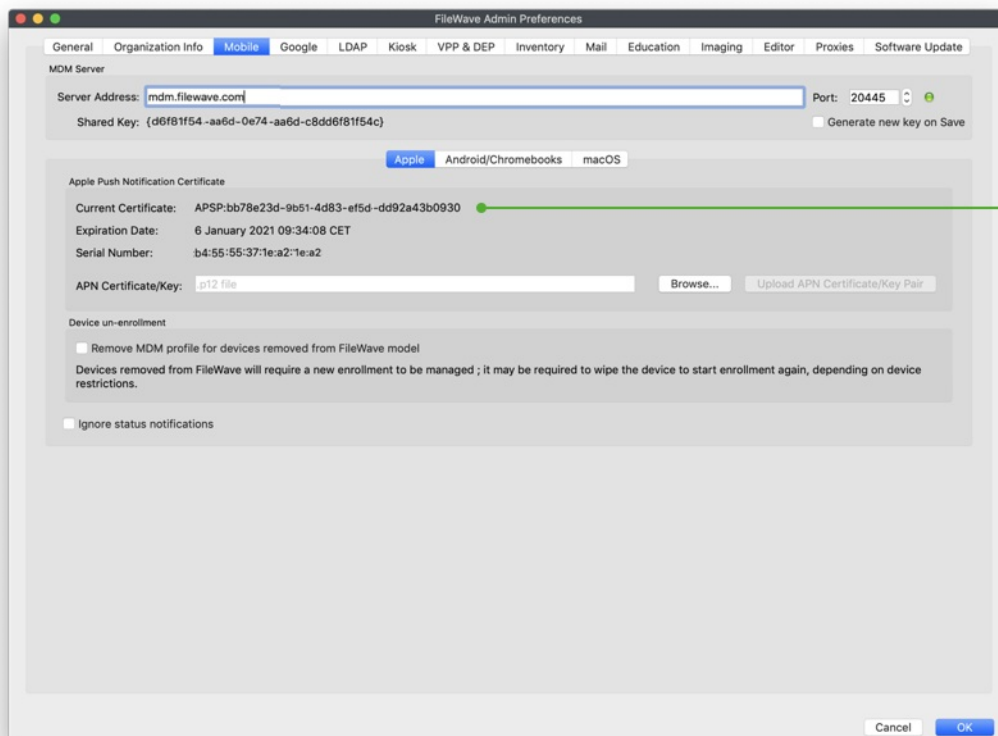


## Uploading the Certificate into FileWave

1. Launch the FileWave Admin and login to the FileWave server.
2. Open the FileWave Admin Preferences.



3. Select the 'Mobile' tab.
4. Click 'Browse' and navigate to the saved ".p12" APNs certificate.
5. Select the exported ".p12" certificate.
6. Click 'Upload APN Certificate/Key Pair'.
7. The topic should match the previous topic.



Topic

8. That is it! FileWave may now manage Apple devices using Apple's Push Notification Service.





APNs certificates require yearly renewals. Through FileWave Admin > Dashboard > Alert Settings, automated emails may be configured. Consider adding 'APN for MDM'. Note this requires the Email preferences in Admin to be configured.

## Related articles


- [APNs Certificate Creation & Renewal on Windows](#)

# APNs Certificate Creation & Renewal on Windows Computers


## Description

The following guide provides the steps to create and renew an APNs certificate using Windows.

### APNs Topic

 An APNs certificate has a unique topic, in the form of a hexadecimal string, and belongs to the Apple ID used to create the certificate. When renewing, the topic must match to ensure devices continue to communicate with the server. As such, not only must the same Apple ID be used when renewing an APNs certificate, but the current certificate must also be selected for renewal.

### APNs Expiry

 Apple Mobile Device Management (MDM) requires an Apple Push Notification service (APNs) certificate; renewable yearly. If APNs certificates are allowed to expire, all MDM communication will be lost, until renewed.

## Information

### Requirements

- An appropriate copy of [OpenSSL](#), which must be downloaded and installed.

Note, that the light version does not include the necessary configuration files.



### CMD Commands

The cmd.exe application should be opened with 'Run as an Administrator' for all commands in this KB

## Step-By-Step Guide

- [Creating the Certificate Signing Request \(CSR\)](#)
- [Sign the CSR](#)
- [Upload the signed FileWave CSR to Apple](#)
  - [Creating a Certificate](#)
  - [Renewing a Certificate](#)
- [Create a ".p12" from the Signed CSR](#)
- [Uploading the Certificate into FileWave](#)
- [Related articles](#)

## Creating the Certificate Signing Request (CSR)

1. Open cmd.exe as an Administrator
2. Create a CSR. Enter the following command, which will result in two new files on the Desktop: request.csr and privateKey.key:

```
"C:\Program Files\OpenSSL-Win64\bin\openssl.exe" req -out "%USERPROFILE%\Desktop\request.csr" -new -newkey  
rsa:2048 -nodes -keyout "%USERPROFILE%\Desktop\privateKey.key" -config "C:\Program Files\OpenSSL-  
Win64\bin\cnf\openssl.cnf"
```



Certificate Private Key names are visible from openssl commands and the Common Name is used to set the Private Key name. Supplying the Apple ID and Server as the Common Name, ensures the Apple ID used to generate the certificate will be stored for future reference.

```
Administrator: Command Prompt
C:\WINDOWS\system32>"C:\Program Files\OpenSSL-Win64\bin\openssl.exe" req -out "%USERPROFILE%\Desktop\request.csr" -new -newkey
rsa:2048 -nodes -keyout "%USERPROFILE%\Desktop\privateKey.key" -config "C:\Program Files\OpenSSL-Win64\bin\cnf\openssl.cnf"
Generating a RSA private key
.....+++++
writing new private key to 'C:\Users\Administrator\Desktop\privateKey.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Indiana
Locality Name (eg, city) []:Fishers
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Demo
Organizational Unit Name (eg, section) []:Demo
Common Name (e.g. server FQDN or YOUR name) []:AppleID: demo@filewave.com, Server: mdm.filewave.com
Email Address []:demo@filewave.com

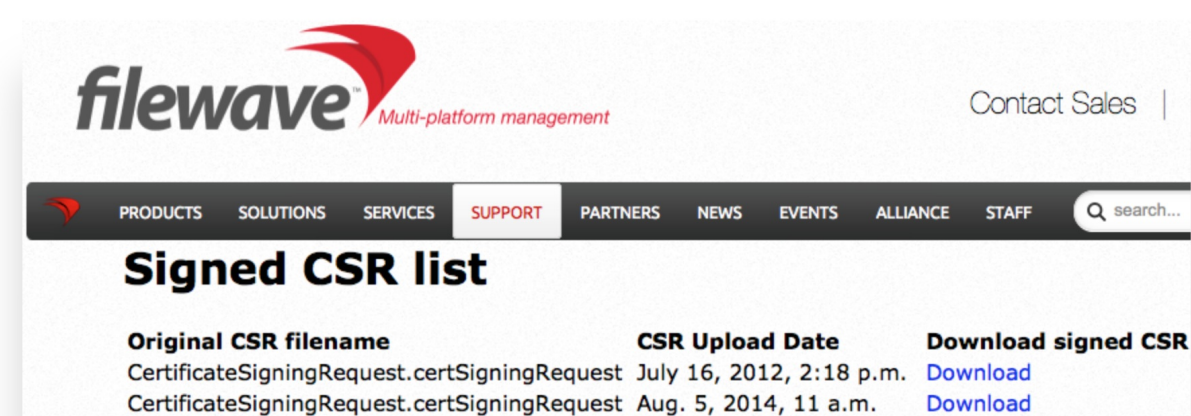
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

C:\WINDOWS\system32>
```

## Sign the CSR

CSR requests must be signed before uploading to Apple. FileWave has a portal for this process, which requires an active FileWave account.

1. Navigate to [https://csr.filewave.com/list\\_csr](https://csr.filewave.com/list_csr) and login.
2. Upload the previously created CSR.
3. 'Download signed CSR' should list this uploaded and now signed CSR.
4. Download this newly signed CSR, ready for upload to Apple in the next section. Again consider where this certificate is stored.



## Upload the signed FileWave CSR to Apple

### Creating a Certificate

1. Navigate to: <https://identity.apple.com/pushcert/> and log in with an Apple ID.

✓ This Apple ID will own the certificate and is required for every renewal. Do not use a personal Apple ID, to avoid complications if that person were to leave the business or institution.

1. Click 'Create'.
2. 'Accept' Apple's 'Terms of Use'.

# Apple Push Certificates Portal

## Create a New Push Certificate

Upload your Certificate Signing Request signed by your third-party server vendor to create a new push certificate.

Notes

Vendor-Signed Certificate Signing Request

no file selected

Cancel

Upload

## Renewing a Certificate

1. Navigate to: <https://identity.apple.com/pushcert/> and log in with the Apple ID used to initially create the certificate.
2. Confirm the Certificate to renew.
3. Select 'Renew'.

To confirm the certificate, compare the Subject DN (Topic) and current certificate.

Clicking the 'i' button will show the certificate details, including the Topic:

Apple Push Certificates

Serial Number : b45555371ea21ea2  
Subject DN : C=GB, CN=APSP:bb78e23d-9b51-4d83-ef5d-dd92a43b0930, UID=com.apple.mgmt.External.bb78e23d-9b51-4d83-ef5d-dd92a43b0930  
Notes :

Cancel Update Note

Service	Vendor	Expiration Date	Status	Actions
Mobile Device Management	FileWave (Europe) GmbH	Jan 6, 2021	Active	<i>i</i> Renew Download Revoke

Topic

Info

Ensure this matches with the 'Current Certificate' in FileWave Admin > Preferences > Mobile > Apple Push Notification Certificate:

FileWave Admin Preferences

General Organization Info Mobile Google LDAP Kiosk VPP & DEP Inventory Mail Education Imaging Editor Proxies Software Update


MDM Server

Server Address:  Port:  ☒ Generate new key on Save

Shared Key:

Apple Android/Chromebooks macOS

Apple Push Notification Certificate

Current Certificate:  

Expiration Date: 06 January 2021 09:34:08

Serial Number: b4:55:55:37:1e:a2:1e:a2

APN Certificate/Key:

Device un-enrollment

☐ Remove MDM profile for devices removed from FileWave model

Devices removed from FileWave will require a new enrollment to be managed ; it may be required to wipe the device to start enrollment again, depending on device restrictions.

☐ Ignore status notifications

Topic

If the 'Topics' do not match do not continue. If the correct certificate is not in the list on Apple's website, this is the wrong Apple ID. If this guide was followed in creating the original certificate, the previously used Apple ID will be viewable from the certificate "Private Key".

Click 'Choose File' and browse to the signed FileWave CSR from the previous section.

Click 'Upload' and Apple will return a 'Confirmation'.

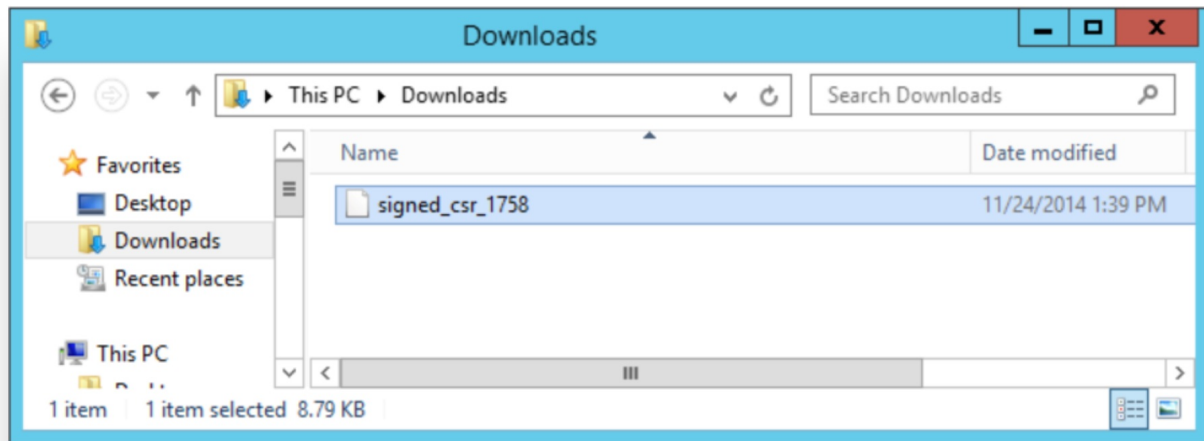
## Confirmation



You have successfully created a new push certificate with the following information:

Service	Mobile Device Management
Vendor	FileWave (Europe) <u>GmbH</u>
Expiration Date	Jan 6, 2021

Click 'Download' and save the ".pem" file. Again consider where this certificate is stored.



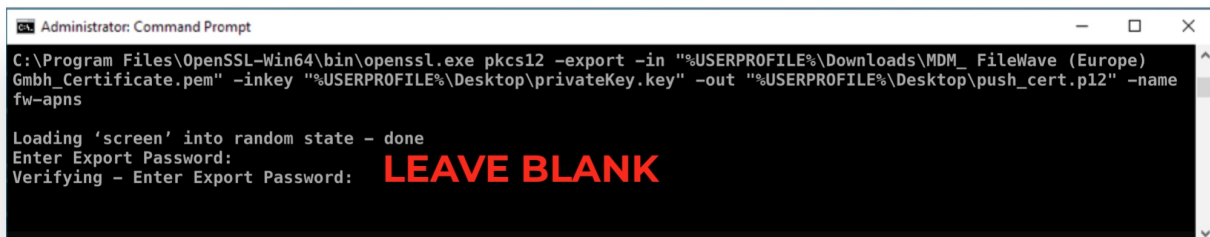
## Create a ".p12" from the Signed CSR

1. Open cmd.exe as an Administrator
2. Create a ".p12". Entering the following command will create the ".p12" on the Desktop:

```
"C:\Program Files\OpenSSL-Win64\bin\openssl.exe" pkcs12 -export -in "%USERPROFILE%\Downloads\MDM_ FileWave (Europe) Gmbh_Certificate.pem" -inkey "%USERPROFILE%\Desktop\privateKey.key" -out "%USERPROFILE%\Desktop\push_cert.p12" -name fw-apns
```

- 1. If the output errors in creating the .p12 certificate file, replace the %USERPROFILE% location by pathing out the exact file location instead.

1. Leave the 'Export Password' blank



1. Certificate details may be checked:

- 1. Common Name and Topic  
The name of the Private Key will show the value defined as the "Common Name" from the creation of the CSR. Where recommendation was followed, this should list the Apple ID and Server name. Additionally the name of the Certificate is the same as the Topic.

```
"C:\Program Files\OpenSSL-Win64\bin\openssl.exe" pkcs12 -info -in C:\Users\Administrator\Desktop\push_cert.p12
```

Note, below image has been edited to remove some details and highlight the two key items of interest.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.18363.836]
(c) 2019 Microsoft Corporation. All rights reserved.

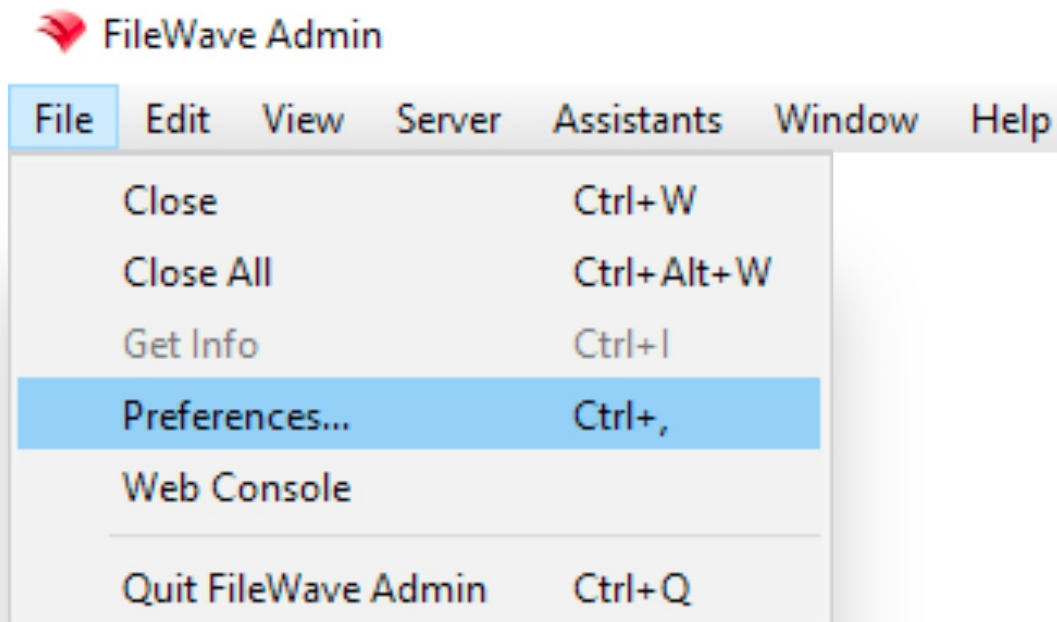
C:\WINDOWS\system32>"C:\Program Files\OpenSSL-Win64\bin\openssl.exe" pkcs12 -info -in C:\Users\Administrator\Desktop\push_cert.p12
Enter Import Password:
MAC: sha1, Iteration 1
MAC length: 20, salt length: 8
PKCS7 Encrypted data: pbeWithSHA1And40BitRC2-CBC, Iteration 2048
Certificate bag
Bag Attributes
    friendlyName: APSP:bb78e23d-9b51-4d83-ef5d-dd92a43b0930
PKCS7 Data
Shrouded Keybag: pbeWithSHA1And3-KeyTripleDES-CBC, Iteration 2048
Bag Attributes
    friendlyName: AppleID: demo@filewave.com, Server: mdm.filewave.com
```

Topic

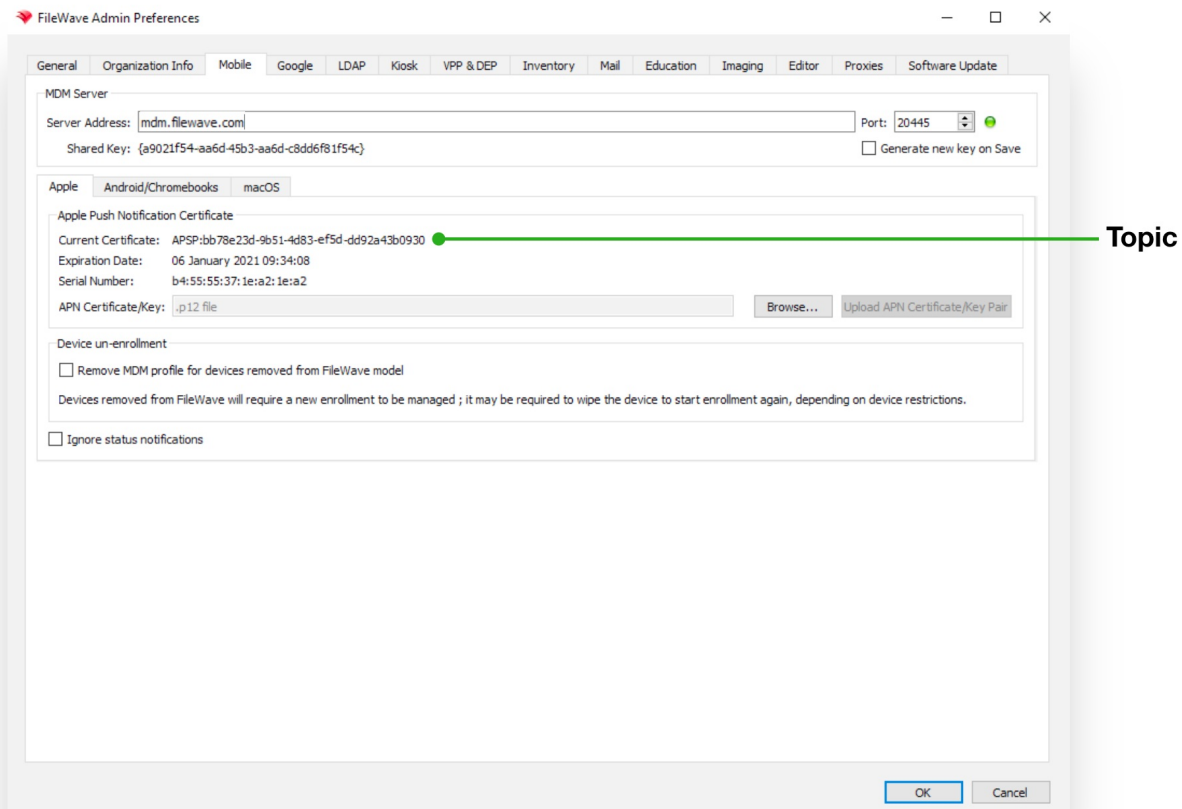
Common Name

## Uploading the Certificate into FileWave

1. Launch the FileWave Admin and login to the FileWave server.
2. Open the FileWave Admin Preferences.



1. Select the 'Mobile' tab.
2. Click 'Browse' and navigate to the saved ".p12" APNs certificate.
3. Select the exported ".p12" certificate.
4. Click 'Upload APN Certificate/Key Pair'.
5. The topic should match the previous topic.



1. That is it! FileWave may now manage Apple devices using Apple's Push Notification Service.

✓ APNs certificates require yearly renewals. Through FileWave Admin > Dashboard > Alert Settings, automated emails may be configured. Consider adding 'APN for MDM'. Note this requires the Email preferences in Admin to be configured.

## Related Articles

[APNs Certificate Creation and Renewal on macOS](#)



# Self-signed SSL Certificates

# Self-Signed SSL Certificates Going Forward

Using a self-signed certificate is not the recommended option and needs to be given a second thought before implementation. Having a certificate trusted by a Global Certificate Authority (CA) is not only the most recommended and most secure option but also becoming more of a requirement for a lot of processes in the tech world.

Having a certificate trusted from a CA will also make sure all of your FileWave communication is as secure and user experience as simplified as possible. If you're FileWave server is going to be managing Chromebooks then a root trusted certificate is required, where as managing iOS devices were self-signed certs can work, you will have to manually trust the certificate during OTA enrollment for the device to communicate with FileWave.

Of course there are some use cases where a self-signed certificate makes sense such as a test or evaluation server.

## FileWave Clients

When using a self-signed certificate your client devices will need this certificate to trust for proper and secure communication with FileWave.

### Initial Install

If the FileWave Client has never been installed on your macOS or Windows devices then you will need to create a custom PKG/MSI. This custom package will need to be filled out with your server address, booster info, and other important data to make sure your clients connect successfully to the FileWave Server. One of those options is Server Certificate, you will need to upload your self-signed certificate into this option so that your new client devices will be trusted by the FileWave server.

- [macOS Custom PKG](#)
- [Windows Custom MSI](#)

Server Certificate

no file selected

### How do you get the self-signed certificate to upload?

To get the self-signed certificate that needs to be uploaded just follow the steps below:

1. Log into the FileWave Admin
2. Go to FileWave Admin → Preferences
3. While in the General Tab find the SSL Certificate Management pane
4. Finally click the Get Current Certificate button, this will download the current SSL certificate you have in FileWave

**Warning: You are using a self-signed certificate. Your devices will need to trust the certificate to use FileWave. iOS 10.3+ devices require explicit trust before manual enrollment.**

[Details...](#)

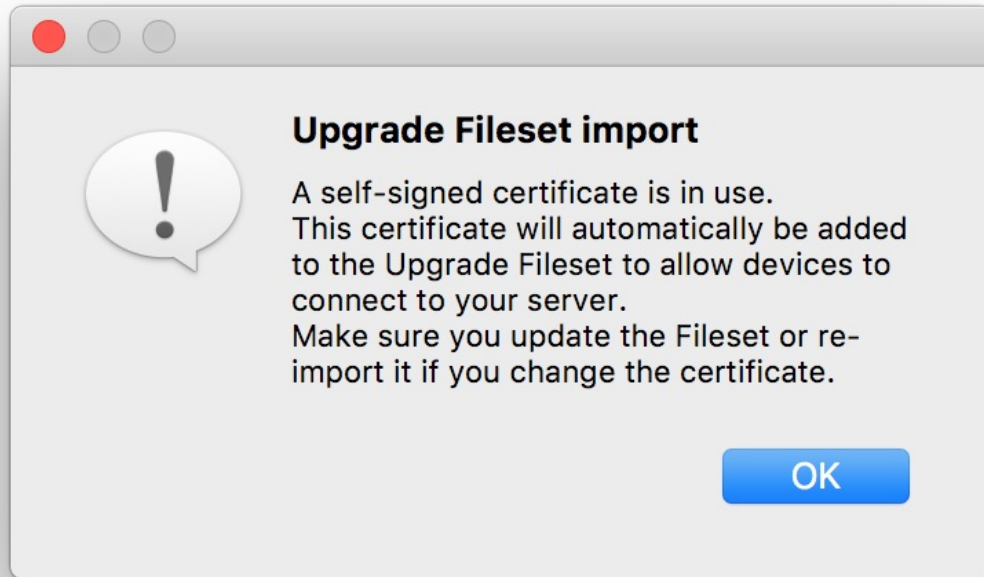
[Upload PKCS12 Certificate](#)

[Get Current Certificate](#)

iOS devices will enroll normally during DEP but, during OTA enrollment the FileWave certificate will need to be trusted manually. Please refer to the [KB article linked here](#) for more information.

## Upgrade

All macOS and Windows clients on FW version 12.9.1 and below will still communicate with the FileWave server, but once upgraded to version 13 the self-signed certificate will need to be pushed to the devices. This will be done automatically when you upload the FileWave version 13 upgrade Fileset into the Filesets section the FileWave Admin.

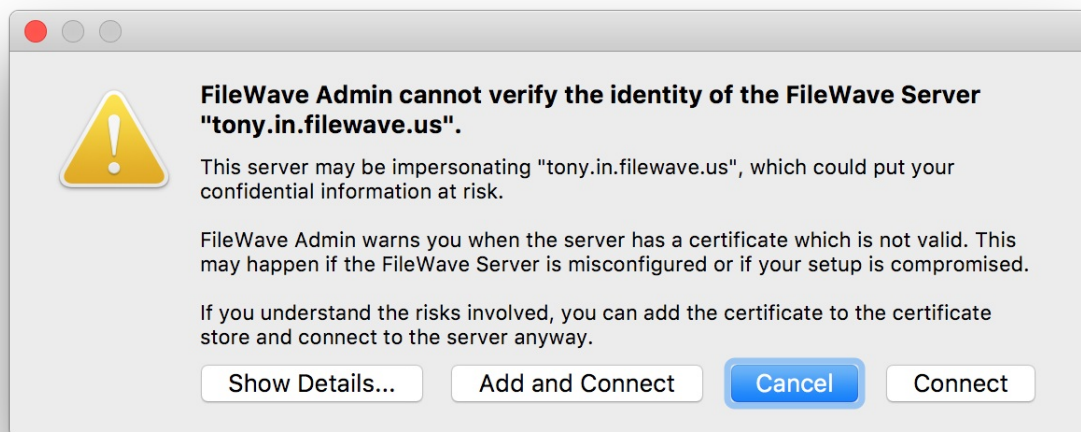


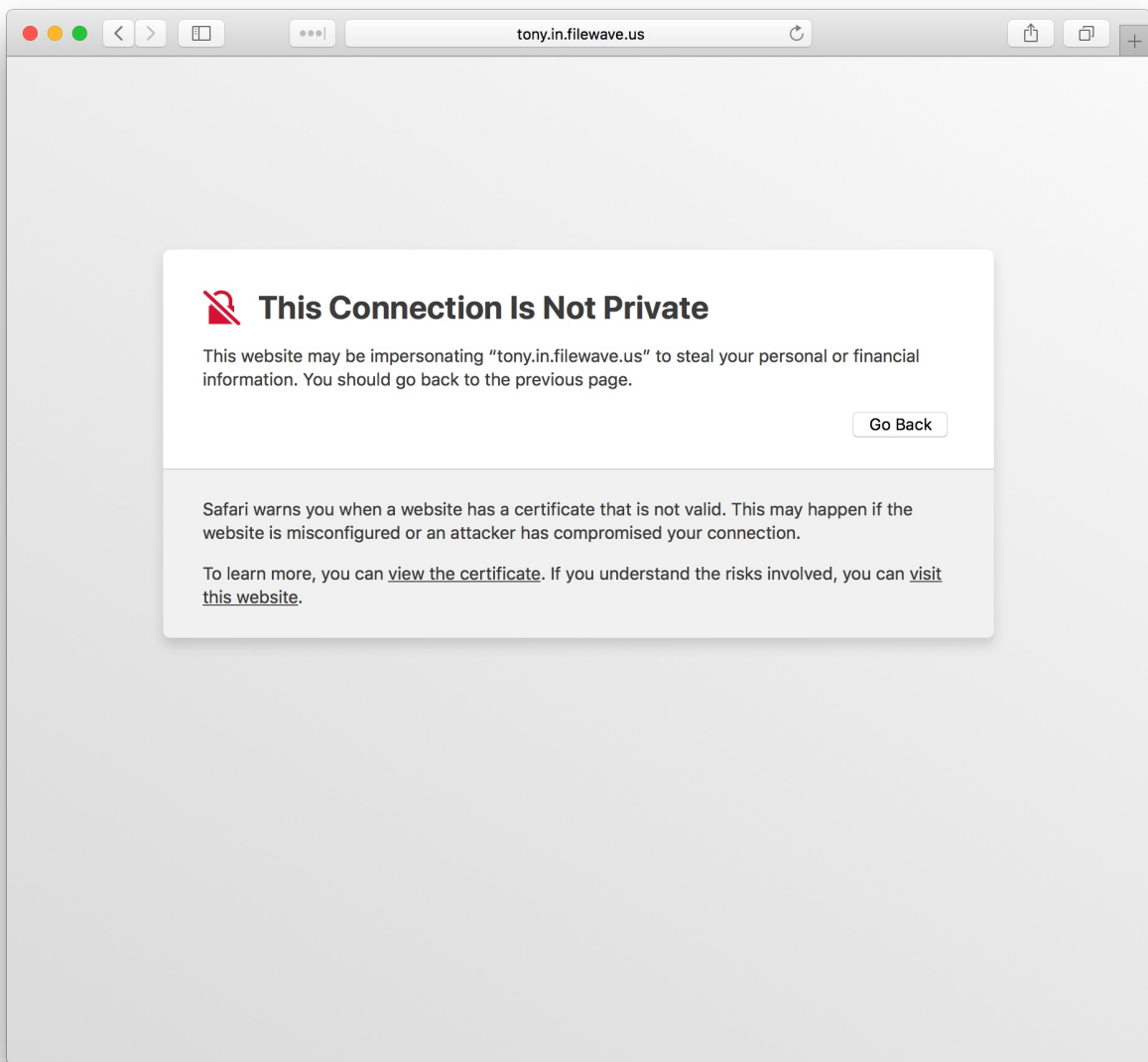
iOS devices will not need anything pushed out, when the FileWave server is updated. But keep in mind during OTA enrollment the FileWave certificate will need to be trusted manually. Please refer to the [KB article linked here](#) for more information.

If you need to renew your self-signed certificate please refer the [KB article linked here](#) for those steps.

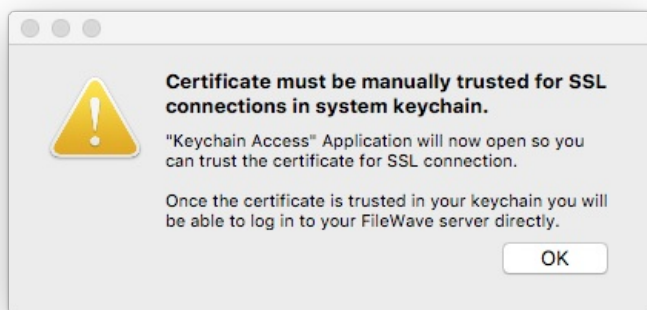
## FileWave Admin

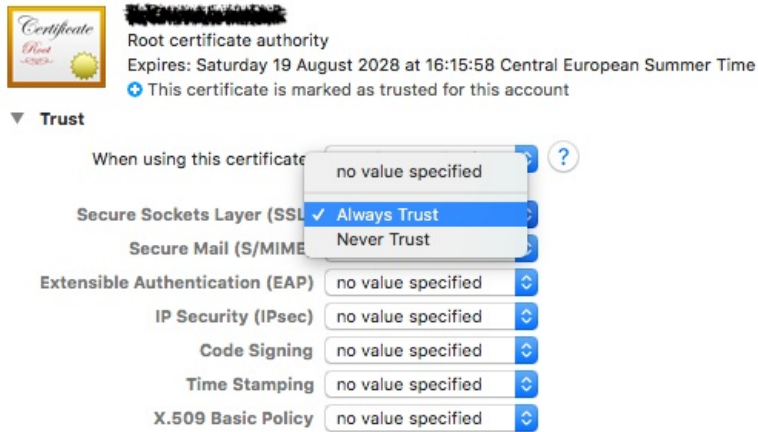
If using a self-signed certificate the FileWave Admin won't be able to verify the identity of the server. When you log into the Admin you will be prompted that the server doesn't trust the certificate and you have the option to continue with the connection being untrusted or you can add the certificate to your trust store then connect. Also when you connect via the Web Console you will be warned that the connection is not private.





On macOS, certificates manually added to trust store require explicit "Trust for SSL" permission.

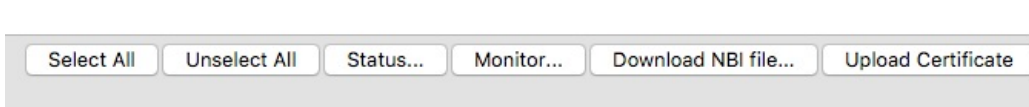




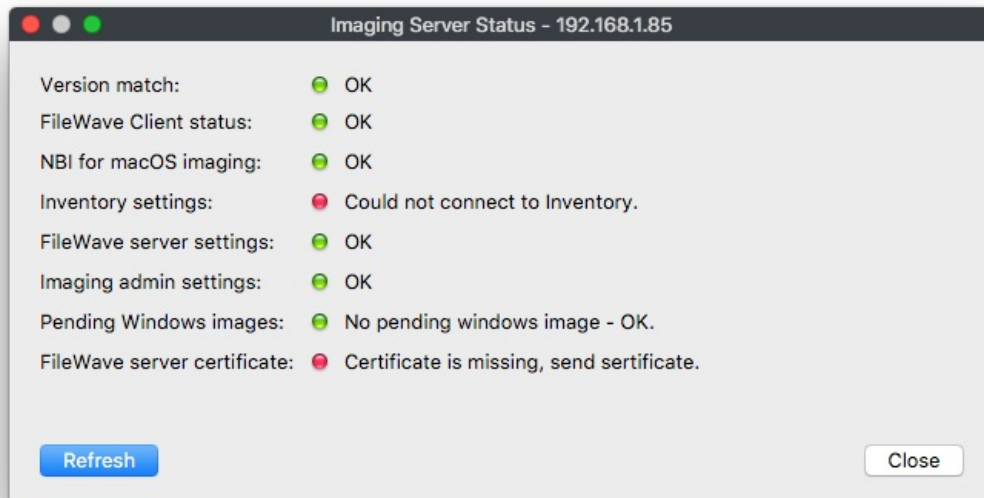
## Imaging Virtual Server

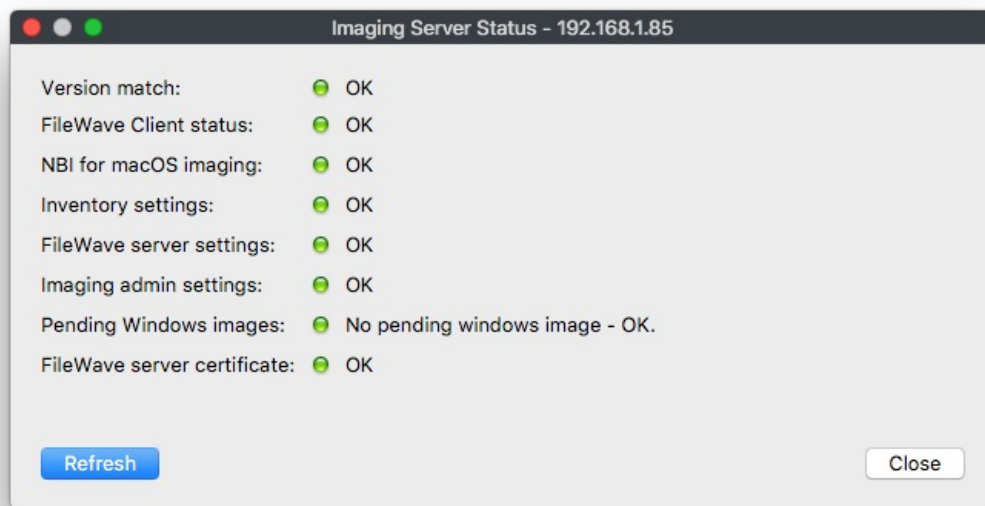
When using self-signed certificates the FileWave server will automatically transfer the certificate to a newly created IVS, but existing imaging servers will need to be pushed the certificate.

1. Log into the FileWave Admin
2. Go to FileWave Admin → Preferences → Imaging
3. Select an imaging server then the Upload Certificate button at the bottom right of the pane



This will send the SSL certificate to the IVS, you have to do this for any existing IVS you have attached to your FileWave server. You can check the status of the IVS to see whether or not the certificate is uploaded, by selecting the IVS and clicking the Status... button.






## Related Content

- [Let's Encrypt Setup for FileWave Server \(Debian\)](#)
- [FileWave Server SSL Certificate from Windows](#)

# Renew FileWave Server Self-signed Certificate

## Description


For simplicity, we should recommend [Renewing with an Official SSL certificate](#) or [Let's Encrypt Setup for FileWave Server \(Debian\)](#)

 Using a self-signed certificate is strongly discouraged for a production server.

## Information

A self-signed certificate may not be trusted by devices out of the box. Instead, the device requires a local copy to be able to trust the certificate. Prior to FileWave 13, this has only affected Mobile devices: [Renew MDM self signed SSL certificate on iOS](#)

However, FileWave uses the certificate for additional security for non MDM communication and initial installation or upgrading to FileWave 13 from a release of 12 or lower.

 Renewal though requires additional steps to ensure device communication is not lost.

## Directions

The 'fwcontrol' command for creating certificates is now a 2 step process, where 'fqdn' should be the Fully Qualified Domain Name of your FileWave Server, e.g. demo.filewave.ch:

```
sudo fwcontrol server generateSelfSignedCert --create --cn=fqdn [--country COUNTRY] [--state STATE] [--locality LOCALITY] [--organization ORGANIZATION] [--ou ORGANIZATIONAL_UNIT] [--email EMAIL] [--ignore_name_mismatch]
sudo fwcontrol server generateSelfSignedCert --install
```

Bracketed options are not required, but may be specified.

## Step 1

### Certificate Generation

Using demo.filewave.ch as an example:

```
sudo fwcontrol server generateSelfSignedCert --create --cn=demo.filewave.ch. --country Switzerland
```

This first step generates a new certificate, but unlike before, it does not overwrite the current active certificate. Instead, this certificate is in a 'pending' state. You should see the following warning when creating the certificate:

```
WARNING: Self-signed certificates are NOT recommended! If you install one, clients in version 13 or greater will
no longer allow connections with the FileWave Server unless you put the new self-signed certificate in their trust
store.
```

```
A self-signed certificate has been successfully created and is now pending for later installation on the server.
```

#### IMPORTANT!

```
- Before installing it, you must deploy it to the trust store of any device whose FileWave Client is in version 13
or greater, otherwise these clients will no longer be able to connect to the server!
```

```
To do so, you can create a filesset with a copy of /usr/local/filewave/certs/server.crt.pending to be deployed in
the trust store folder.
```

```
- Once you are ready to install the new self-signed certificate and if you understand the risks, please run this
command again with option --install.
running restart apache command
```

Instead a new certificate key/crt pair of files may be seen in the following server folder and will show as 'pending', along with the original key/crt pair:

```
/usr/local/filewave/certs/server.crt
/usr/local/filewave/certs/server.crt.pending
/usr/local/filewave/certs/server.key
```

```
/usr/local/filewave/certs/server.key.pending
```

As indicated by the Important message, all clients will require a copy of this certificate to communicate with the server. During transition, it is important that both original and new certificate are installed on devices. Copy the server.crt.pending and rename appropriately for deployment. e.g. server.2019.04.30.crt

## Mobile Devices

Installing the new certificate on Mobile devices is as before, except a profile needs to be made with this new certificate as well as the current certificate:

[Renew MDM self signed SSL certificate with iOS devices](#)

## Computers

Installing the new certificate on Computers is the same as the process for Upgrading to FileWave 13, but this new certificate needs to be added to a Fileset manually. This could either be the current FileWave Upgrade Fileset or a new Fileset. Location of the file is either:

macOS:


### macOS Client/Booster Trust Store


```
/private/var/FileWave/trust_store
```

Windows:

### Windows Client/Booster Trust Store

```
C:\ProgramData\FileWave\FWClient\trust_store
```

 Set the certificate 'Verification' to 'Ignore At Verify' to ensure it is never removed

 If the new certificate should become live on the server prior to the clients receiving this Fileset, those devices will no longer be manageable through FileWave and a manual process will be required to locally instal the certificate.

Whichever option is chosen, a method should be designed to monitor the installation process. Only once all devices are updated, should the 'pending' certificate become the active server certificate.

Options for monitoring could include:

- Fileset Reports
- Custom Fields

A Custom Field could take the following form (assuming the example file name of 'server.2019.04.30.crt'):

### macOS Example Custom Field

```
#!/bin/bash

server_cert=$(find /var/FileWave/trust_store -name "server*.crt")

if [[ "$server_cert" != "" ]]
then
    echo Yes
else
    echo No
fi

exit 0
```

## Step 2

This second step enables the 'pending' certificate as the active certificate, replacing the original server certificate file.

```
sudo fwcontrol server generateSelfSignedCert --install
```

Once all clients have the new certificate within their respective trust stores, the 'pending' server certificate may now become active. When this update of the certificate occurs, any other elements requiring the server certificate should also be updated at this time.



## DEP

The server certificate is stored as an 'Anchor certificate' within any created DEP profile. As with any certificate change, once the certificate is renewed, new DEP profiles should be created; do not duplicate.

## Custom PKG/MSI

The Custom Client Installer also needs to include the certificate. The following links allow for uploading the current server certificate within the 'Options'

- [macOS Custom Client Builder](#)
- [Windows Custom Client Builder](#)

Details highlighted on: [Self-Signed Certificates Going Forward](#)

# Renew MDM self signed SSL certificate with iOS devices

## Self Signed certificate renewal

Renewing MDM self-signed certificate can be done if the current certificate has to be changed:

- the certificate is or is about to expire
- the certificate is not or will not be trusted by devices anymore

The main issue with self-signed certificate is that, by definition, those certificates are not issued by a trusted Certificate Authority (CA), and are not trusted by default on devices. To have devices trust those certificates, the certificate must be added to the trust store. This can be achieved by:

- DEP enrollment, which can add the server certificate
- Deploying a profile
- manually installing and trusting the certificate



In production environment, it is highly recommended to use trusted CA issued certificate ; self-signed certificates should only be used for testing and evaluation purpose. The best and most simple way to solve self-signed certificate renewal issues is to stop using self-signed certificate and use trusted CA certificates. There are free options like [Let's Encrypt](#) to have a trusted cert.

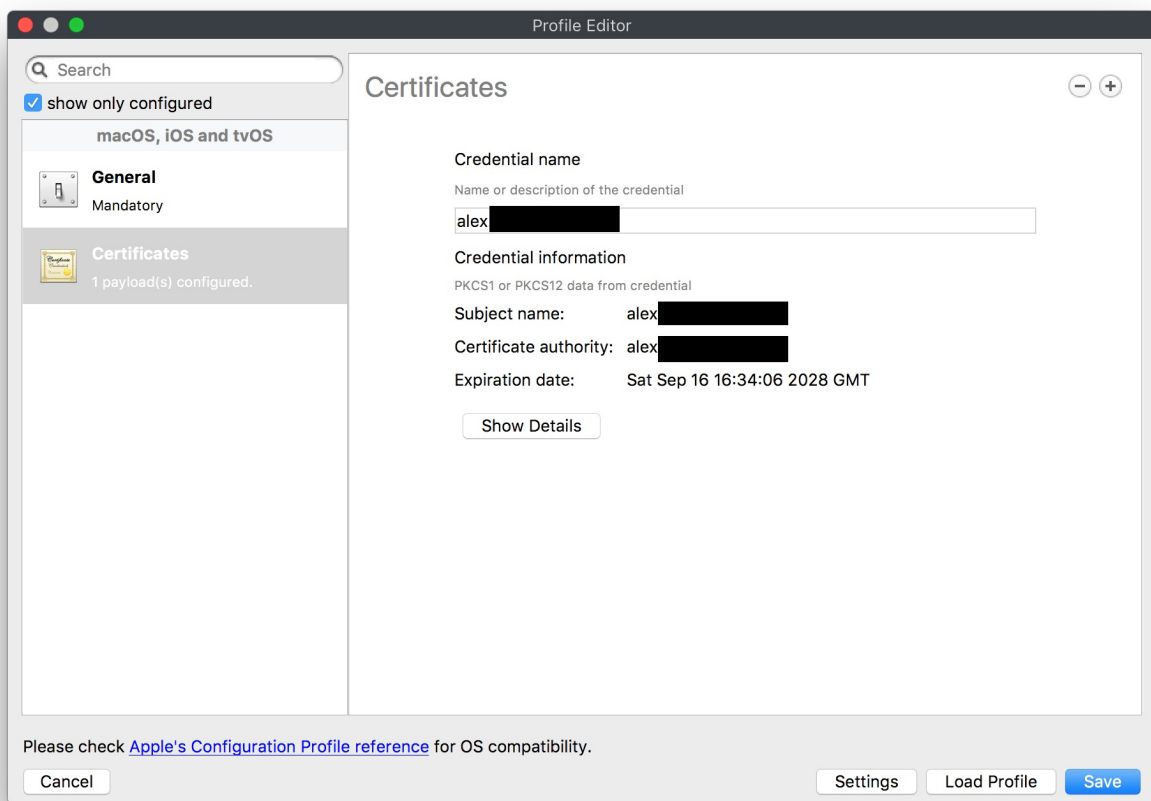
## Planned renewal

In case you need to renew a self-signed certificate, you need to ensure all your devices will trust the new certificate before you renew it ; this implies the following steps:

1. Create a new private key and certificate

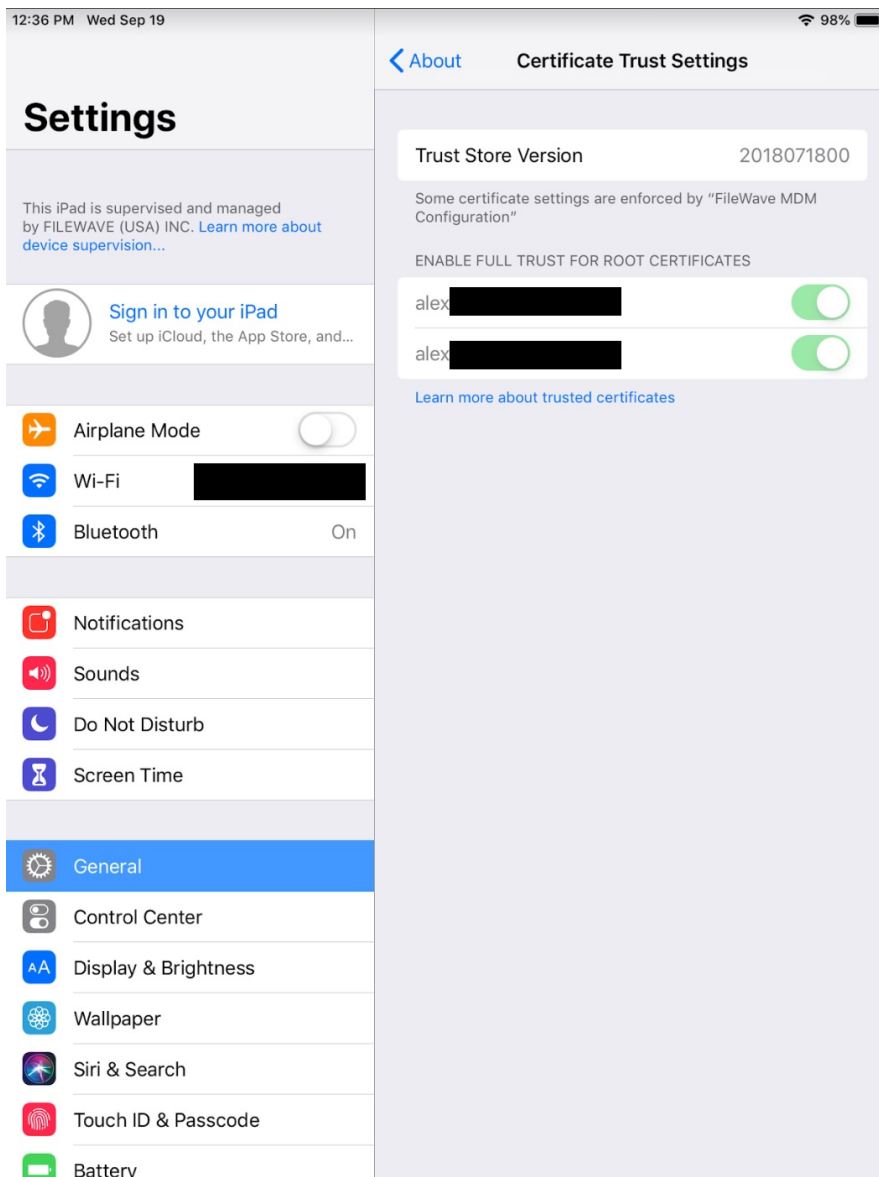
```
$ openssl req -x509 -nodes -sha256 -days 3650 -newkey rsa:2048 -keyout /tmp/server.key -out /tmp/server.crt
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:
State or Province Name (full name) []:
Locality Name (eg, city) []:
Organization Name (eg, company) []:
Organizational Unit Name (eg, section) []:
Common Name (eg, fully qualified host name) []:filewave.acme.org
Email Address []:
```

2. Import server.crt file into a profile filesset



3. Deploy the profile filest to all your devices


4. You are able to confirm that the profile was received and trusted by the device by going to Settings → General → About → Certificate Trust Settings, and should see your old as well as new self-signed certificate listed and trusted. The screen shot below shows what you will see with the device trusting both certificates.




5. Once all devices have the profile, you can switch the key and certificate. The path to your new "server.crt" and "server.key" may change depending on where the certificate is located on your FileWave server:

```
$ cd /usr/local/filewave/certs
$ mkdir old_certs
$ mv server.crt server.key old_certs
$ cp /tmp/server.*
$ fwcontrol apache restart
```

6. Re-create DEP profiles and associations as the DEP profile contains a copy of the certificate and is sent to Apple at association time ; a new certificate implies a new DEP profile.

 Failure to update your DEP profiles to have the new profile will cause trust issues at enrollment

## Unplanned or late renewal

 Worst case possibility using a self-signed cert that expires.

If the current certificate is not trusted by devices anymore (or because some devices did not get the new certificate before the switch), the renewal process remains the same, but with one exception: as devices will stop trusting the server certificate it's not possible to use FileWave to deploy the new certificate.

At this point, the best solution is to move forward with a trusted CA certificate ; your devices will start communicating immediately to your server as soon as the certificate is in place.

In case trusted CA is not possible, you will have to manually add the certificate to each impacted device:

1. deploy the new certificate to devices ; you can either send it via e-mail, or send your users to the usual enrollment page and ask them to install the cert via "step 1"
2. in the trust store, the newly installed certificate must be granted "use for SSL" permission

## Related Content

- [Renew FileWave Server Self-signed Certificate](#)
- [Let's Encrypt Setup for FileWave Server \(Debian\)](#)

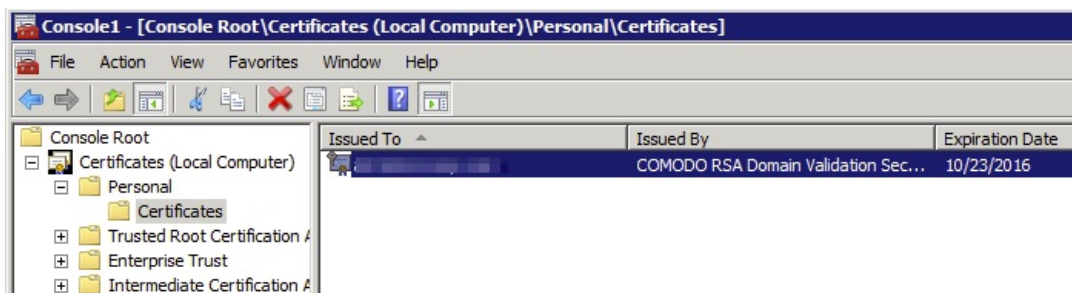
# Troubleshooting

# Export .p12 SSL Certificate from Windows

When managing mobile devices, it is considered best practice to install a root trusted SSL certificate on the FileWave Server. This certificate is located in the FileWave Admin > Preferences > Mobile tab. If you generated the Certificate Signing Request (CSR) for your SSL certificate on a Windows based system and have completed the certificate generation process, the SSL certificate and intermediates bundle can be exported as a .pfx file directly from Windows. This bundle would contain all components (private key, public certificate, Root CA certificate, and intermediate certificate bundle). This .pfx file (after renaming the extension to .p12) can then be uploaded to the FileWave Admin > Preferences > Mobile tab without any modification.

## Step-by-step guide

- Open a Run dialog and enter "mmc".
- Go to File > Add/Remove Snap-in.
- Add the Certificates snap-in and click the Add > button in the middle.
- Add for the Computer account.
- Pick Local computer and click Finish.
- Click the OK button.
- In the MMC console browse to Certificates (Local Computer) > Personal > Certificates on the left. If your certificate is not there, browse the rest of the Certificates (Local Computer) tree until you find it.



- Select your certificate in the middle pane, right-click, and pick All Tasks > Export.
- When prompted pick Yes, export the private key.



- Under Personal Information Exchange - PKCS #12 (.PFX) check Include all certificates in the certification path if possible. Leave the other 2 checkboxes unchecked.



- Click the Next button and specify an export password. The FileWave Admin will prompt you for this password when you attempt to upload the SSL certificate in the Preferences> Mobile tab.
- Save the file to your desktop.
- Change the file extension from .pfx to .p12.
- Upload .p12 file in the Mobile preferences tab of the FileWave Admin.



# Determine Correct Intermediates Bundle for SSL Certificate

Some SSL providers include multiple intermediates certificate bundles along with your SSL certificate. Your SSL certificate must be merged with one of these intermediates bundles along with your private key to generate a .p12 certificate file that can be uploaded into the Mobile preferences tab of the FileWave Admin. If the incorrect intermediates bundle is used, two steps will appear during interactive MDM enrollment rather than one, like in the screenshot below. If the incorrect intermediates bundle is used, client devices will not be able to communicate with the FileWave MDM server correctly. There should normally only be one step listed, the one to "Enroll Device", if there are no certificate trust chain issues.



## Step-by-step guide

Follow the steps below to determine the correct intermediates bundle to pair with your SSL certificate so that only one step appears on the interactive enrollment page.

1. Be sure to choose Apache format when downloading your SSL certificate from the your provider. If the certificate files do not have a .crt extension redownload them again and pick Apache format this time.
2. Go to the Intermediate Certificate Check page at <https://tools.keycdn.com/ssl>.
3. Paste the contents of your SSL .crt file from your SSL provider.
4. Follow it up with the contents of the desired intermediates .crt file right below it. The intermediates bundle may contain multiple certificates. Copy and paste them all into the Intermediate Certificate Check page below your SSL certificate.
5. Click the Validate button.
6. You'll receive a response stating either "No chain issues detected" in green or "Chain issues detected" in brown. If there are chain issues keep replacing the intermediates bundle with another one until there are no chain issues. The intermediates bundle that results in no chain issues is the one you need to use when generating your .p12 file for FileWave.

## Decoder

This SSL check decodes your SSL certificates and validate intermediate certificate issues.

Certificate (PEM format)

```
-----BEGIN CERTIFICATE-----Certificate-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----Intermediate certificate-----END
CERTIFICATE-----
```

Q Validate

No chain issues detected.

**Correct Intermediates Bundle**

1. Subject CN: fw.lanrevcorp.com » Issuer CN: Go Daddy Secure Certificate Authority - G2
2. Subject CN: Go Daddy Secure Certificate Authority - G2 » Issuer CN: Go Daddy Root Certificate Authority - G2
3. Subject CN: Go Daddy Root Certificate Authority - G2 » Issuer CN:
4. Subject CN: » Issuer CN:

## Decoder

This SSL check decodes your SSL certificates and validate intermediate certificate issues.

**Certificate (PEM format)**

```
-----BEGIN CERTIFICATE-----Certificate-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----Intermediate certificate-----END  
CERTIFICATE-----
```

Q Validate

### Wrong Intermediates Bundle

Chain issues detected. Possible reasons are missing intermediate certificate or wrong order of the certificates.

×

1. Subject CN: fw.lanrevcorp.com » Issuer CN: Go Daddy Secure Certificate Authority - G2
2. Subject CN: Go Daddy Secure Extended Validation Code Signing CA - G2 » Issuer CN: Go Daddy Root Certificate Authority - G2

×

# Self Signed Certificate Error during iOS OTA Enrollment

This article shows how to resolve an error if you are manually enrolling 10.3+ devices in FileWave with a self-signed certificate.

It is considered a best practise to have a root trusted certificate defined in the FileWave> Preferences> Mobile> HTTPS certificate section. In FileWave v12+ it is easy to determine whether you have a self-signed certificate or not. Simply log into the FileWave Admin, open the preferences, go to the "Mobile" tab, and you will see in the HTTPS section, the following line:

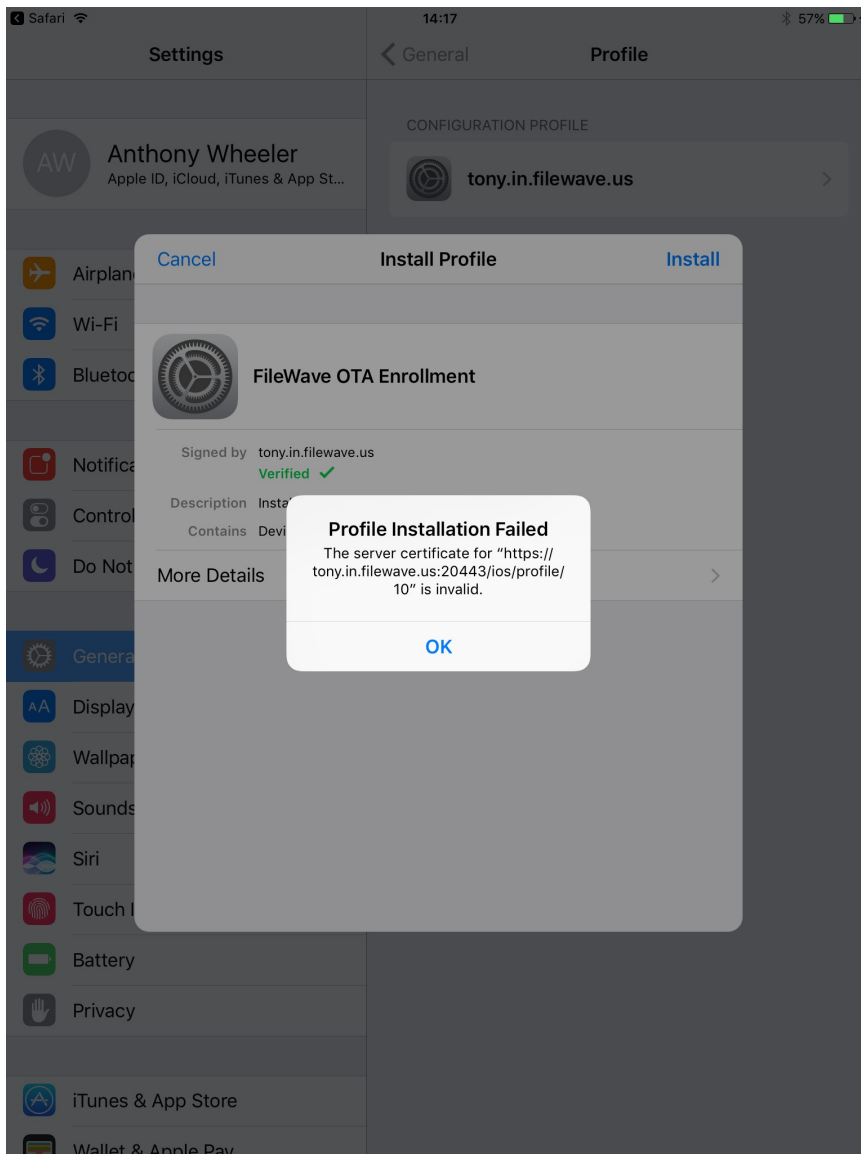
Warning: You are using a self-signed certificate. Your devices will need to trust the certificate to use FileWave. iOS 10.3+ devices require explicit trust before manual enrollment.

Details...

Upload PKCS12 Certificate

Get Current Certificate

If this is the case, you will still be able to enroll iOS 10.3+ devices through DEP. But if the device is iOS 10.3+ and you try a manual web enrollment (OTA), you will get the following error.



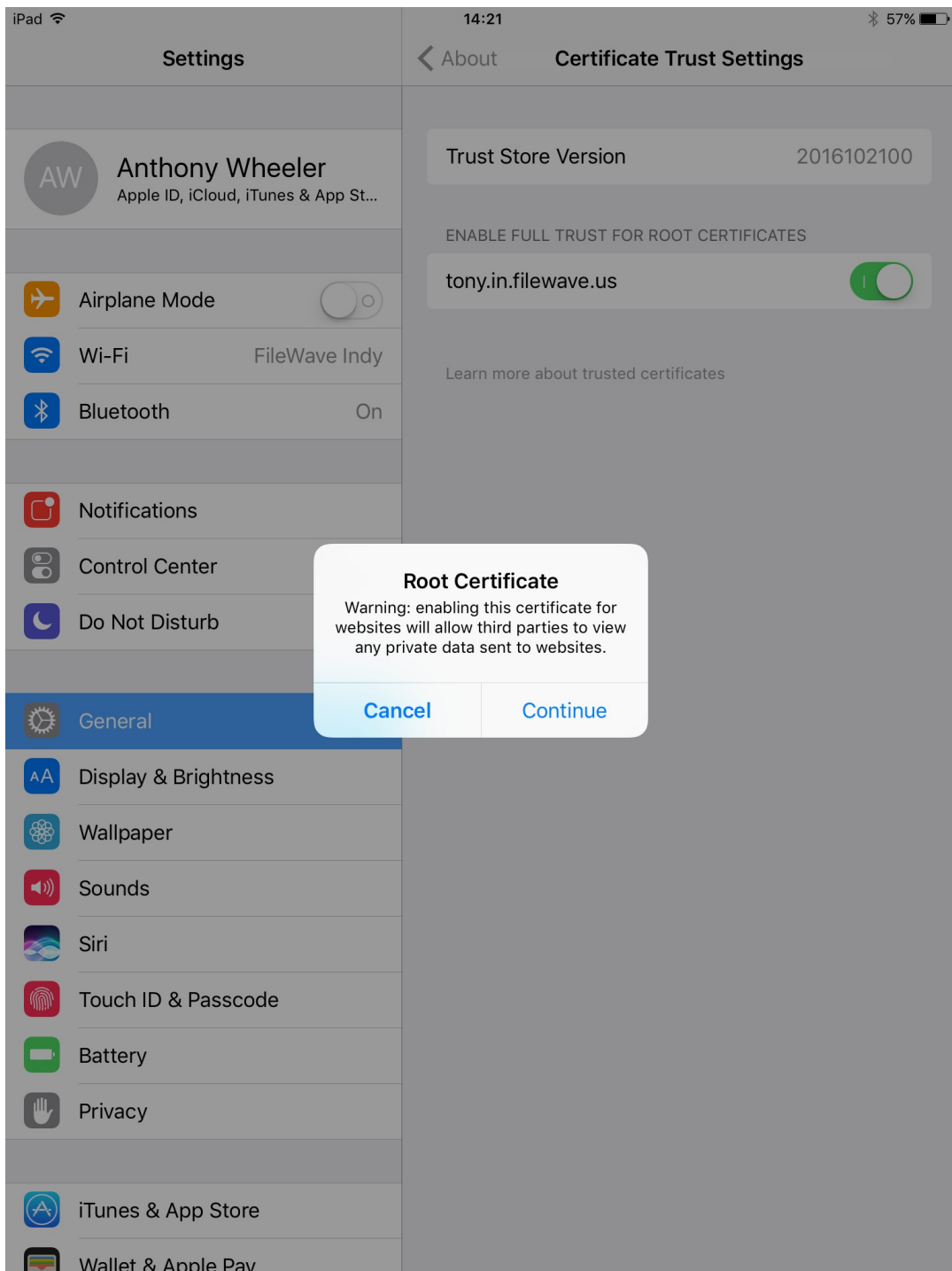
If you choose to retain your self-signed certificate, you will have to use the steps below to resolve the error. Alternatively, you can purchase a root trusted certificate, and you will not encounter this issue. Again, it is highly recommended that you purchase a root trusted certificate (can include a wildcard) so that you don't have to work around this trust issue, as described below.

## Steps to Resolve (if you choose to keep a self signed certificate in place)

1. Navigate to the your manual enrollment address: <https://your.fw.server.DNS.here:20443/ios>
2. Select: "Step 1 - Install Certificate"

<b>step</b> <b>1</b>	<b>Install Certificate</b> Allows enrollment of this device	➤
<b>step</b> <b>2</b>	<b>Enroll Device</b> Get access to mobile services and software	➤

3. Once you have selected step one, the device will ask you to Install the cert, go through those three prompts by hitting Install each time and finally Done.
4. After the certificate has been installed, open the "Settings" app on the iOS device. Do not start Step 2 (This will prompt the error).
5. Go into General => About
6. At the bottom of the "About" section, tap the sub section called "Certificate Trust Settings"
7. You will see an option called ENABLE FULL TRUST FOR ROOT CERTIFICATES
8. Toggle that option for your newly installed certificate



Now go back to the manual enrollment page and finish the steps with "Step 2 - Enroll Device".

# SSL Server Certificates - iOS 13 and macOS 10.15

Apple have updated their requirements for certificates for their new operating system releases: <https://support.apple.com/en-us/HT210176>

The new requirements can be broken down in the 3 major sections:

1. The mandatory presence of a Subject Alternative Name
2. Presence of an OID (1.3.6.1.5.5.7.3.1) designating the use of the certificate for TLS Web Server Authentication
3. Maximum validity period of 825 days

Requirement 1 is confirmed to render MDM clients unable to connect to the MDM server when not being met.

Requirements 2 and 3 are not currently ( as of 24th of September 2019 ) interfering with MDM function when not being met. These two new requirements are not met by newly generated self-signed certificates as of FileWave Server 13.1.3 - so renewing your self-signed certificate will not mitigate this issue permanently. FileWave Server will be updated in a future release to accommodate these new guidelines in order to comply with self-signed certificates.

If you are using a self-signed certificate on a production server we recommend you purchase a valid 3rdparty certificate that has been signed by a [trusted root CA](#).

To verify whether your certificate is affected by a missing subject alternative Name, please run the following command on your Linux/macOS server :

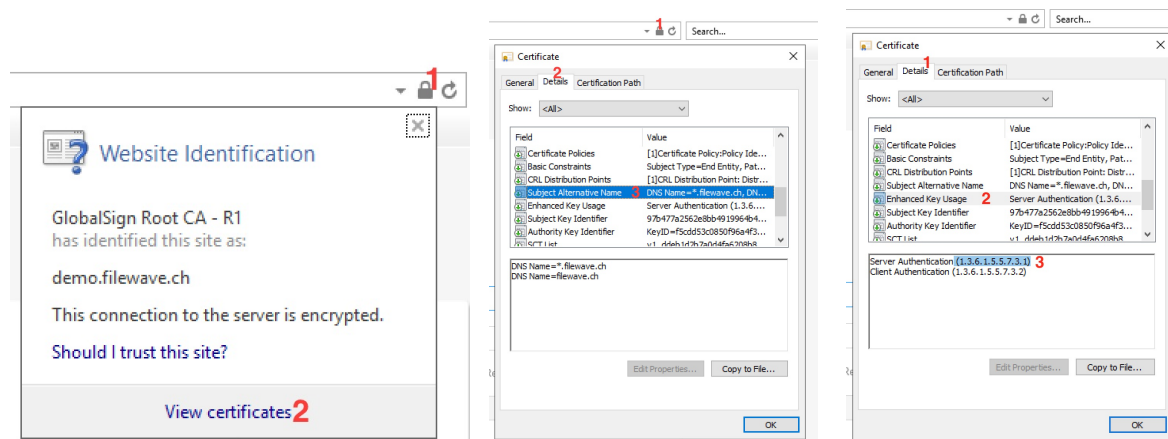
```
certSANCheck=$(openssl x509 -in /usr/local/filewave/certs/server.crt -text -noout | awk '/DNS/'; ); if [ "$certSANCheck" == "" ]; then echo "Certificate requires updating"; else echo "Certificate has SAN, no action required"; fi
```

If the above script returns "Certificate has SAN , no action required" , please verify the presence of the OID extension using the next snippet . Otherwise, please jump to "[Directions](#)" below to read on for instructions on how to mitigate this issue.

```
certOIDCheck=$(openssl x509 -in /usr/local/filewave/certs/server.crt -text -noout | awk '/TLS Web Server Authentication/'; ); if [ "$certOIDCheck" == "" ]; then echo "Certificate requires updating"; else echo "Certificate has OID, no action required"; fi
```

If the above script returns "Certificate has OID , no action required" , you can stop reading now . Otherwise, please check this page for updates on how to mitigate this issue .

To verify a Windows Server based Installation, please browse to your iOS enrollment page and verify the certificate as shown below :



If the above "Subject Alternative Name" is visible in the Certificate Details, and the "Enhanced Key usage" shows the OID 1.3.6.1.5.5.7.3.1, you can stop reading now. Otherwise, please read on for instructions on how to mitigate this issue.

## Description

Apple have updated their requirements for certificates for their new operating system releases:

<https://support.apple.com/en-us/HT210176>

Some of these restrictions were in place with earlier versions of iOS and macOS:

<https://support.apple.com/en-gb/HT209028>

#### Loss of Device Management



This could affect device communication if using non-compliant certificates. Certificate should be updated as per the following guide before updating devices or MDM device management will be lost.

#### Self-Signed and 3rd Party Certificates



Although this is likely to be an issue with older self-signed certificates, official 3rd party certificates could also be affected. Where 3rd party certificates are affected, contact your supplier for an updated certificate.

#### Information

##### Requirements:

- FileWave Server version 13.1.0+

Particular interest should be paid to the following:

- TLS server certificates must present the DNS name of the server in the Subject Alternative Name extension of the certificate. DNS names in the CommonName of a certificate are no longer trusted.

When using self-signed certificates, if the certificate does not have a SAN entry, it will no longer be trusted in Apple's new operating systems.

FileWave has an option to generate self-signed certificates:

```
sudo fwcontrol mdm generateSelfSignedCert --cn=fqdn [--country COUNTRY] [--state STATE] [--locality LOCALITY] [--organization ORGANIZATION] [--ou ORGANIZATIONAL_UNIT] [--email EMAIL] [--replace] [--ignore_name_mismatch]
```

However, earlier versions of FileWave did not generate a certificate with a Subject Alternate Name (SAN).

As of FileWave 13.1.0, fwcontrol generates a certificate that includes a SAN

#### Certificate Generation



Although a newer version of FileWave may be in place now, what is relevant here is the version of FileWave that was running when the certificate was generated.

## Directions

This is a good opportunity to switch to an official SSL certificate, using our guide to ensure device management continuity:

[Root Trusted SSL Certificate \(Using and Renewing\)](#)

If you cannot make the switch at this time, please observe the following KB for distribution in profiles through MDM:

[Renew MDM self signed certificate](#)

For clients, the new certificate needs to be added to the client's 'Trust Store' prior to making the pending generated certificate live. Details found on the following KB.

[Renew Self-signed Certificate - FileWave 13+](#)

#### Recovery

For devices upgraded when the server certificate did not meet requirements there are options:

- Obtain an official SSL 3rd party certificate (highly recommended)
- Manually install and trust the server certificate on each affected device
- Update the self-signed certificate as per the details then re-enrol all affected devices (may involve erasure of device)