

UPDATE: Apple Pay Certificates Generation

Process Certificate

Step 1: Generate an ECC Private Key

You will generate a new ECC private key using the prime256v1 curve.

```
1 openssl ecparam -genkey -name prime256v1 -out yourdomain.key
```

 Note: the private.key file will be shared from the merchant side

This command generates an ECC private key using the prime256v1 curve and saves it to a file named `yourdomain.key`. Replace `yourdomain` with your domain name or any relevant identifier for your use case.

Step 2: Generate a CSR using the ECC Key

Next, you will generate a CSR using the ECC private key you just created.

```
1 openssl req -new -key yourdomain.key -out yourdomain.csr
```

During this process, you will be prompted to enter information that will be included in your certificate request, such as your country name, state or province, locality (city), organization name, organizational unit, common name (your domain name), and email address. Fill these in as appropriate for your situation.

 Step 3: Generate the Apple Payment Certificate:

1. Open the Apple Developer Portal
2. create a new merchant identifier or use an existing one
3. From the merchant identifier page click the "Create Certificate" button

Certificates, Identifiers & Profiles

[← All Identifiers](#)

Edit or Configure Merchant ID

[Remove](#)

[Save](#)

Name

packman

Identifier

merchant.packman

You cannot use special characters such as @, &, *, ' , " , ~ , .

Apple Pay Payment Processing Certificate

To configure Apple Pay Payment Processing for this Merchant ID, create a Payment Processing Certificate. Apple Pay Payment Processing requires this certificate to encrypt transaction data. Use the same certificate for Apple Pay Payment Processing in apps or on the web.

Create an Apple Pay Payment Processing Certificate for this Merchant ID.

[Create Certificate](#)

Apple Pay Payment Processing on the Web

To configure Apple Pay Payment Processing on the web for this Merchant ID, you must register and verify the domains that will process transactions. You must also create a Apple Pay Merchant Identity, which authenticates your web sessions with the Apple Pay Payment Processing servers.

Incorporation of Apple Pay Payment Processing into your website is subject to these [Apple Pay Payment Processing Web Merchant Terms and Conditions](#) and [Acceptable Use Guidelines](#). Failure to comply with any of these Terms and Conditions or guidelines may

 Note: the merchant identifier will be shared from the merchant side

4. Upload your CSR "`yourdomain.csr`" and then download the generated payment certificate

Certificates, Identifiers & Profiles

[← All Certificates](#)

Create a New Certificate

[Back](#)

[Continue](#)

Certificate Type

Apple Pay Payment Processing Certificate

Upload a Certificate Signing Request

To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac.

[Learn more >](#)

Choose File

Step 4: Conversion Command: Use the following command in the terminal:

```
1 openssl x509 -inform DER -in apple_pay.cer -out certificate.pem
```

i Note: the certificate.pm file will be shared from the merchant side

Explanation:

1. `-inform DER` : Specifies the input format is DER (used by `.cer`).
2. `-in apple_pay.cer` : Input file (your `.cer` certificate).
3. `-out certificate.pem` : Output file in PEM format.

Once this is done, you'll have the certificate in `.pem` format.

Merchant Certificate: [↗](#)

To generate an RSA Certificate Signing Request (CSR) on Ubuntu, follow these steps:

1. Install OpenSSL (if not already installed) [↗](#)

```
1 sudo apt update
2 sudo apt install openssl
```

2. Generate a Private Key [↗](#)

Generate a 2048-bit RSA private key:

```
1 openssl genpkey -algorithm RSA -out private.key
```

i Note: the private.key file will be shared from the merchant side

3. Generate the CSR [↗](#)

Use the private key to generate a CSR:

```
1 openssl req -new -key private.key -out request.csr
```

During this process, you will be prompted to provide information for the certificate. Here's a breakdown of the fields:

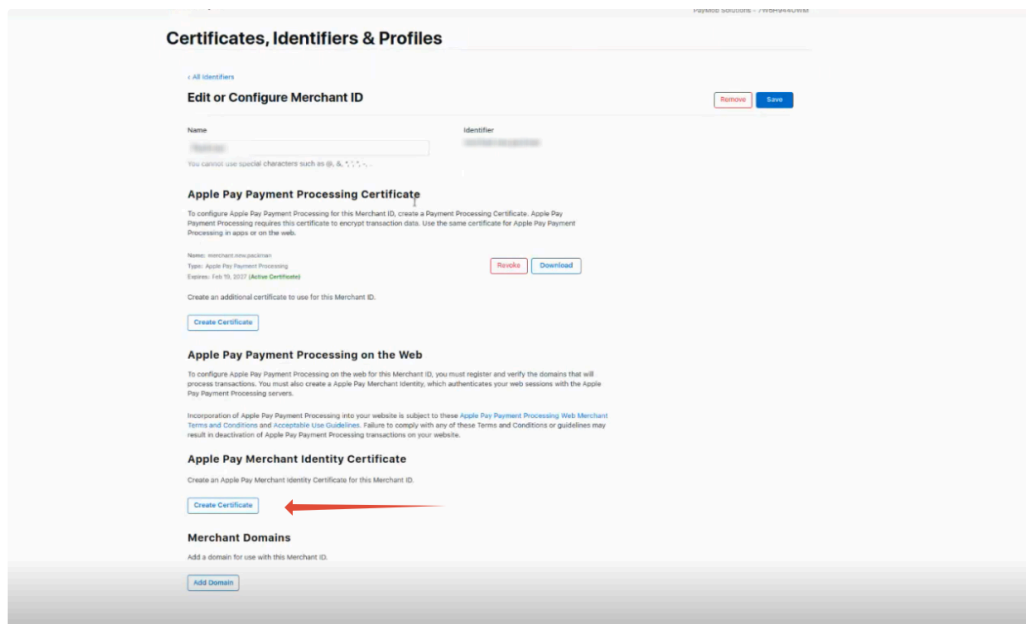
- **Country Name** (2-letter code)
- **State or Province Name** (full name)
- **Locality Name** (e.g., city)
- **Organization Name** (company name)

- **Organizational Unit Name** (section or department)
- **Common Name** (domain name or server name)
- **Email Address** (email for contact)

Optionally, you can skip fields by pressing `Enter`.

i Generate the Apple Payment Certificate:

1. Open the Apple Developer Portal
2. Use the existing merchant identifier
3. From the merchant identifier page click the "Create Certificate" button



4. **Conversion Command:** Use the following command in the terminal:

```
1 openssl x509 -inform DER -in merchant_id.cer -out certificate.pem
```

i Note: the certificate.pm file will be shared from the merchant side

Explanation:

1. `-inform DER` : Specifies the input format is DER (used by `.cer`).
2. `-in merchant_id.cer` : Input file (your `.cer` certificate).
3. `-out certificate.pem` : Output file in PEM format.

Once this is done, you'll have the certificate in `.pem` format.

5. Verify the CSR (Optional) [↗](#)

To verify the content of the CSR:

```
1 openssl req -text -noout -verify -in request.csr
```

This will show the details, confirming the CSR was generated correctly.

You now have the `private.key` file (your private key) and the `request.csr` file (the CSR to be submitted to a Certificate Authority).