Step by step instructions on how to digitally sign the software.

## Digitally Signing for Windows

**1. Obtain a Code Signing Certificate:**

- Purchase a code signing certificate from a Certificate Authority (CA) like DigiCert, Comodo, or Symantec.

- Follow their process to verify your identity and receive your certificate.

**2. Install Your Certificate:**

- Once you receive your certificate, install it on the machine where you will be signing your application.

- This usually involves importing the certificate into the Windows Certificate Store.

**3. Prepare Your Tools:**

- Ensure you have the Windows SDK installed, as it includes the `signtool`, which is used for signing applications.

- You can download it from the [Microsoft website](https://developer.microsoft.com/en-us/windows/downloads/windows-sdk/).

**4. Sign Your Application:**

- Use `signtool` to sign your application. The basic command format is:

```
signtool sign /f [path-to-your-certificate.pfx] /p [your-password] /tr http://timestamp.digicert.com /td sha256 /fd sha256 [path-to-your-electron-app.exe]
```

- Replace `[path-to-your-certificate.pfx]`, `[your-password]`, and `[path-to-your-electron-app.exe]` with your actual certificate path, certificate password, and application path.

**5. Verify the Signature:**

- After signing, verify the signature with:

```
signtool verify /pa /v [path-to-your-electron-app.exe]
```

## Digitally Signing for macOS

**1. Obtain an Apple Developer ID:**

- Enroll in the Apple Developer Program at [developer.apple.com](https://developer.apple.com/).
- Obtain a Developer ID certificate for application signing.

**2. Install Your Certificate:**

- Download and install your Apple Developer ID certificate on your macOS machine.
- It will be added to your Keychain.

**3. Sign Your Application:**

- Use the `codesign` tool available in macOS to sign your application. The basic command is:

```
codesign --sign "Developer ID Application: Your Name" --deep --force [path-to-your-app.app]
```

- Replace `"Developer ID Application: Your Name"` with the name of your certificate and `[path-to-your-app.app]` with the path to your Electron app.

**4. Notarize Your Application (optional but recommended):**

- Notarizing your app assures users that it has been scanned for malware and is safe to run.
- Use Apple's `altool` to upload your app for notarization:

```
xcrun altool --notarize-app --primary-bundle-id "com.yourcompany.yourapp" --username "your-apple-id" --password "your-app-specific-password" --file [path-to-your-app.zip]
```

- After notarization, staple the ticket to your app:

```
xcrun stapler staple [path-to-your-app.app]
```

**5. Verify the Signature:**

- You can verify the signing by:

```
codesign --verify --deep --strict --verbose=2 [path-to-your-app.app]
```

- Replace `[path-to-your-app.app]` with the path to your app.

## *Notes:*

- The paths and names in the commands need to be replaced with your actual file paths, names, and passwords.
- For macOS, you'll need an Apple Developer account, which has an annual fee.
- The process of notarizing the app for macOS is especially important if you distribute outside of the Mac App Store.
- Always keep your private keys and passwords secure.

Following these steps, you should be able to digitally sign your Electron application for both Windows and macOS.