

# Creating a Code Signing Key

To assist our users in verifying the authenticity of our software releases, we digitally sign them. As of Fedora 3.3, this is part of the [Fedora Release Process](#), and requires that the committer doing the final build for distribution uses their code signing key.

## Requirements

We have borrowed heavily from the [release signing policy used by the ASF](#).

When generating your code signing key:

1. Use a 4096 bit RSA key with SHA512 hash
2. Use your real name, preferred email address, and "CODE SIGNING KEY" as the comment.
3. Use a strong password to protect your key

Once generated, you should:

- Keep your private key file on a safe, secure computer, and make sure you have a secure backup.
- Never use this key for purposes other than code signing or signing other keys.

## 1. Generate Your Key

Carefully follow the instructions [here](#) to generate your key and check that SHA1 is avoided.

Tip: Popular binaries for GnuPG 2.x can be found [here](#):

- [Linux](#)
- [Mac OS X](#)
- [Windows](#)

Note: After initially generating your key with GnuPG 2.x (gpg2), you can work with it using the more commonly-available 1.4.9 release (gpg).

## 2. Publish Your Public Key

To enable people to find your public key, you should publish it to a well-known keyserver. This is a simple command with gpg:

```
gpg --send-key [yourKeyID]
```

...where *yourKeyID* is the last 8 digits of your public key fingerprint.

This will upload your public key to a well-known keyserver, which will then trigger other connected keyservers to get a copy. Afterward, you can verify the general availability of your public key by searching for your name in one of the [keyservers in the SKS network](#).

## 3. Publish Your Key Fingerprint

Add your fingerprint to the [Fedora Contributors](#) page.

## 4. Sign Others Committers' Keys

For each fingerprint on the [Fedora Contributors](#) page:

- Download the key via:

```
gpg --recv-keys [fingerprint]
```

- Sign it via:

```
gpg -u [yourKeyID] --sign-key [fingerprint]
```

- Upload the signature via:

```
gpg --send-key [fingerprint]
```

## 5. Ask Other Committers to Sign Your Key

Email the other committers, notifying them that you've signed their key and uploaded the signature, and they should run:

```
gpg --refresh-keys
```

...then ask them to sign your key as indicated above.

After they have had a chance to sign your key and upload the signature, you should also do a `--refresh-keys` so your local web of trust is up to date.

## 6. Optional: Sign Your Own Key

If you have another key you use for normal communication, you can improve the [web of trust](#) by signing your other key with with your code signing key, and vice-versa.