

# Building Your CD

## Summary

This documents how to build your own ISO, complete with custom kickstart (system default configuration), set of RPM and splash screens.

## Step 1. Download base CentOS base ISOs

Download the first CD ISO image from the CentOS operating system for 32 and/or 64 bit. Mirror lists are here

<http://isoredirect.centos.org/centos/>

Put the ISOs into a single directory and do not rename them, In this example we'll use `~/downloads/centos`

## Step 2. Download or build sipXecs RPMS

Obtain a copy of all the RPMS needed to install sipXecs including the rpms built from the `./lib` directory such as `ruby-postgres`, `sipx-freeswitch`, etc. into a directory. It can be organized anyway you want, in fact it can be a directory that also acts as a yum repository. For this example, we'll put these into `~/build/sipxecs`

## Step 3. Configure

Run the standard script to setup the configuration script

```
autoreconf -if
```

Then run configure pointing to your CentOS ISO and sipXecs rpms. In our example this will be.

```
./configure --enable-centos-iso ISO_DIR=/home/buildbot/Downloads RPM_DIST_DIR=/home/buildbot/sipx-master-4.2
```

## Step 4. Build

If you're building RPMS, go ahead and run your target. For example `make distro.centos-5-x86_64.rpm distro.centos-5-i386.rpm`.

## Step 5. Build ISO



### sudo privs

Unfortunately the make process needs to mount the ISO and unless you configure mount only root can do that.

```
make iso
```

if you want to just build the 32 bit or the 64 bit ISO, you can run `make iso-32` or `make iso-64` respectively.

## Step 6. Contribute

If you have can changes you'd like to submit back, please post on sipx-dev mailing list and/or initiate git pull requests.

## Customizing

Reasons to customize the ISO

- ISO makes a lot of assumptions about your system. US locale for example, size of the disk for auto-formatting is greater than 13GB, the default size of the partitions, etc. To change these, you must customize the ISO
- If you want to brand the ISO with your company logos, colors or welcome messages
- If you want to include extra RPMS by default

If you don't you'll produce the standard SIPfoundry ISO and may be all that you desire. If you do decide to customize, then create a separate directory to hold your custom files and configure your build to use this directory. For example `./configure ... OEM_DIR=~/work/ezuze`

Contents of `~/work/ezuze`

**iso.mak.in** - override various make variables used to create the ISO. See `Makefile.in` for a list of all the variables you could override. Here is an example

```
SPLASH_FILE = @OEM_DIR@/splash.lss
ISO_LABEL ?= ezuce
VOLUME_LABEL ?= "ezuce @PACKAGE_VERSION@"
APPLICATION_LABEL ?= "ezuce @PACKAGE_VERSION@"
PUBLISHER_ID ?= dhubler@ezuce.com
PREPARER_ID ?= dhubler@ezuce.com
```

**oem.m4** - Contains M4 macros that you can use to extend the kickstart file. Look in `ks.m4` for an idea of the hooks that are currently in place you can alter. Here is an example

```
define(`spx_packages',
spx_packages()
ezuce
)
define(`welcome_message',

Welcome to eZuce
=====
First time logon: user = root      password = setup

)
define(`repo_contents',
[ezuze]
name=ezuce CentOS \${releasever} - \${basearch}
baseurl=http://download.example.com/ezuze/PACKAGE_VERSION()/CentOS_\${releasever}/
enabled=1
gpgcheck=0
)

```

**splash.lss** - this is a graphic that is the initial splash screen and is in a format that is not well supported, but google around for how to edit this file on your OS.