

# Upgrading from 2.0.0

## Performing the upgrade

This is a manual process due to the wide range of possible site configurations. So you will need to adapt to the specific needs of your HUBzero installation as necessary. The steps outlined below have been tested with the HUBzero 2.0.0 VMWare virtual machine image.

Some instructions will differ depending on how your site was installed. We have highlighted items like passwords, template names, and hub names which you may need to substitute with values specific to your hub.

All actions listed here must be performed as the root user.

You changed your root password right? If hubzero2015 is still your root password you really must change it as soon as possible.

Using a terminal, log into the host.

```
example.com login: root
Password: hubzero2015
```

The first thing that needs to be done is update the available package list, otherwise package installations will fail with obsolete package references.

```
apt-get update
```

In the case of the VMWare virtual machine image the Network Time Protocol daemon isn't running so the time and date on the site could be wrong. While not critical, updating this can eliminate a number of warning errors that might otherwise be confusing. Let's install the ntp package.

```
apt-get install -y ntp
```

Then manually have it update the time and wait 10 seconds to give it enough time to take effect.

```
ntpd -gq
```

## UPGRADING FROM 2.0.0

---

```
sleep 10
```

Now we will back up the HUBzero databases and the HUBzero CMS installation. We do not back up the "site" directory here, you may do so as you see fit. Be careful, some of these lines are wrapping around but are supposed to be on a single line. Remember to replace "mytemplate" with the name of any custom template you may be using. Custom and modified templates now are located in /var/www/example/app/templates. Preexisting Stock templates exist in /var/www/example/app/templates.

```
cd /root
```

```
mysqldump --defaults-file=/etc/mysql/debian.cnf --default-character-set=utf8 --skip-extended-insert --compact --order-by-primary --result-file=example.`date +"%Y%m%d"`.sql example
```

```
mysqldump --defaults-file=/etc/mysql/debian.cnf --default-character-set=utf8 --skip-extended-insert --compact --order-by-primary --result-file=example-metrics.`date +"%Y%m%d"`.sql example_metrics
```

```
tar -czf template-mytemplate  
.`date +"%Y%m%d"`.tar.gz -C /var/www/example/app/templates mytemplate
```

```
tar -czf cms.`date +"  
%Y%m%d"`.tar.gz -C /var/www/example --exclude=app/site/* .
```

Next we will make sure your HUBzero 2.0.0 and Debian 7 installation is completely up to date to ensure the upgrade to HUBzero 2.1.0 will go smoothly. If you are running Debian 6 you will need to upgrade to Debian 7 first (which is outside the scope of this document). The next step will rewrite your package repository configuration, if you have customized yours you can just

## UPGRADING FROM 2.0.0

---

update the lines related to packages.hubzero.org to change the HUBzero package repository "ellie-deb7". Copy and paste the all of the following as one command:

```
cat << HERE > /etc/apt/sources.list
deb http://ftp.us.debian.org/debian/ wheezy main contrib non-free
deb-src http://ftp.us.debian.org/debian/ wheezy main contrib non-free

deb http://ftp.us.debian.org/debian/ wheezy-updates main contrib non-
free
deb-src http://ftp.us.debian.org/debian/ wheezy-
updates main contrib non-free

deb http://security.debian.org/ wheezy/updates main contrib non-free
deb-src http://security.debian.org/ wheezy/updates main contrib non-
free

deb http://download.openvz.org/debian wheezy main

deb http://packages.hubzero.org/deb ellie-deb7 main
deb-src http://packages.hubzero.org/deb ellie-deb7 main
HERE
```

Now we can do the actual package upgrade.

```
apt-get update
```

```
apt-get dist-upgrade -y
```

You may be prompted with a message during the update of some packages. Enter "q" to quit the message.

Do some cleanup.

```
apt-get clean
```

## UPGRADING FROM 2.0.0

---

```
apt-get autoremove -y
```

Update the existing HUBzero 2.0.0 installation with the most recent version (updated during previous step).

```
hzcms update
```

Install the HUBzero 2.1.0 package.

```
apt-get install -y hubzero-cms-2.1.0
```

Apply the HUBzero CMS 2.1.0 package to the current instance and let it reapply any configuration that may have been undone by the operating system upgrade. We also re-enable webdav and openvz so that their configuration files get updated (they were replaced during the operating system upgrade).

```
hzcms update
```

Remove old hubzero-cms packages that are no longer needed.

```
apt-get purge -y hubzero-cms-2.0.0
```

Clean up cached packages one more time then reboot to make sure everything is working correctly. This reboot should be relatively quick.

```
apt-get clean
```

```
reboot
```

## UPGRADING FROM 2.0.0

---

Update the tool container image, if you don't have any Debian 7 tool container images you can skip this section.

```
chroot /var/lib/vz/template/debian-7.0-amd64-maxwell
```

```
cat << HERE > /etc/apt/sources.list
deb http://http.us.debian.org/debian/ wheezy main contrib non-free
deb-src http://http.us.debian.org/debian/ wheezy main contrib non-free

deb http://security.debian.org/ wheezy/updates main contrib non-free
deb-src http://security.debian.org/ wheezy/updates main contrib non-
free

deb http://http.us.debian.org/debian/ wheezy-updates main contrib non-
free
deb http://packages.hubzero.org/deb ellie-deb7 main
deb-src http://packages.hubzero.org/deb ellie-deb7 main
HERE
```

```
apt-get update
```

Install the hubzero-policyrcd package which will prevent all future package installations and updates for this container image from stopping or restarting services (which would start/stop services running on the host which would not be a good thing).

```
apt-get install -y hubzero-policyrcd
```

Some packages require access to the /proc filesystem to get information about the system. Mount this special filesystem inside the container image chroot environment.

```
mount -t proc proc /proc
```

Upgrade packages.

## UPGRADING FROM 2.0.0

---

```
apt-get upgrade -y
```

After the upgrade is complete unmount the temporary /proc mount and remove any packages that are no longer needed.

```
umount /proc
```

```
apt-get autoremove -y
```

Remove any cached packages from the image in order to conserve disk space and exit the chroot environment.

```
apt-get clean
```

```
exit
```

If you modified a core template in-place instead of creating your own see the last section to upgrade your modified core template.

If you were using your own template you will find it has been moved to /app/templates/yourtemplate and automatically modified it to conform with the new release. If your template is more complex it may require additional work to be made compatible with HUBzero 2.1.0. If you need to switch to the default template until you complete upgrading your template you may do so as follows:

```
Login to /administrator with the username 'admin' and the JOOMLA-ADMIN password in /etc/hubzero.secrets.
```

```
From the main menu is /administrator go to  
Extension Manager -> Template Manager -> Click on the star for the Ki
```

## UPGRADING FROM 2.0.0

---

mera (site) template in the default column.

That's it. Your hub should now be upgraded to HUBzero 2.1.0!

### Upgrading a custom template

If you modified a default HUBzero template (hubbasic2012, hubbasic2013) without copying it to a new name you will need to restore it from the backups we made earlier as the upgrade process overwrites the standard HUBzero templates. Restoring this old template will likely break your site as your template will need to be upgraded to be compatible with this version of the HUBzero CMS. The web developer [upgrade notes](#) outline the basic changes that will be needed for your template, this should include renaming your template as it will now conflict with a core template name.

```
cd /var/www/example/app/templates
```

```
tar -xvpf /root/template-mytemplate.`date +"%Y%m%d"` .tar.gz
```