Ubuntu Factory Packaging Guide

Mario Limonciello
Mario_Limonciello@Dell.com

July 23, 2009

Contents

1 Introduction

Within Dell, Ubuntu is shipped across nearly all LOBs. The requirements for these LOBs will vary, but for the most part they leverage a common framework for performing a factory installation. The success of this framework requires that all teams that use it understand how to properly create releases for I-Drive. This guide should help create the proper structure and framework for all I-Drive releases created for any Ubuntu release that uses the UBX token.

2 Glossary

Depending on the background of the reader, this document uses a lot of terminology that may be foreign to new or non-Dell employees.

- **LOB** - Line of Business
- **FI** - Factory installation
- **SDR** - A file that calls out all software packages for a system
- **SRV** - A software package that can be installed on a system
- **I-drive** - A central tool that Dell and vendors use to host software packages
- **Release** - A particular version of a software package hosted on I-drive
- **FISH** - The team responsible for activating software packages to be factory installable
- **FID** - The team responsible for activating the base OS to be factory installable

3 UBX Token

Starting with Ubuntu 8.10, the Linux FI team has created a token that will always be used when performing an Ubuntu install, the **UBX** token. This token will support multiple versions, but it's important that the I-drive release that is paired with it is intended for the correct version. When creating an I-Drive release, remember to include in the description what version of Ubuntu it is targeted for. When building an SDR, this token contains all possible Ubuntu SRVs.

3.1 FI Basic layout

The UBX token will have a framework laid out in the following structure:

- `./grub`
- `./docs`
- `./preseed`
- `./scripts`
- `./scripts/chroot-scripts`
- `./scripts/chroot-scripts/os-pre`
- `./scripts/chroot-scripts/fish`
- `./scripts/chroot-scripts/os-post`
- `./debs`
- `./debs/main`
- `./isolinux`

The only two main directories relevant to I-drive releases that will go through FISH are `./debs` and `./scripts/chroot-scripts/fish`.

4 Preparing an I-Drive Release

When creating an I-drive release, any modifications should be reflective as though they were to be extracted on *top* of the existing UBX FI framework. Be sure to always place this directory structure in your ZIP or TAR.GZ on I-Drive.

4.1 Generic DEBs only

If this release is providing an application or a standalone set of DEBS that don't need any system integration, they should placed in `./debs/main`. All debs placed in this directory are automatically installed by the UBX FI Framework. Anything but the corner cases should generally be included here.

4.2 AMD & NVIDIA Graphics Drivers

Graphics drivers are distributed soley as debs, but are a corner case because a post installation procedure needs to be ran. If the version of the AMD or NVIDIA graphics driver that shipped with the Ubuntu DVD is sufficient, you can skip shipping an updated driver as the UBX FI framework will install the one with the DVD by default. When shipping an updated driver,some expectations are made in terms of where the debs will be found.

4.2.1 AMD Graphics Drivers

For AMD graphics, you will require four debs in this structure:

- `./debs/fglrx/xorg-driver-fglrx*.deb`
- `./debs/fglrx/*kernel-source*.deb`
- `./debs/fglrx/fglrx-amdcccle*.deb`
- `./debs/fglrx/fglrx-modaliases*.deb`

If you are working on Ubuntu 9.04, due to a bug, you may also need to update the fish script in `./scripts/chroot-scripts/fish/01-graphics.sh`.

4.2.2 NVIDIA Graphics Drivers

For NVIDIA graphics, you will require four debs in this structure:

- `./debs/nvidia/nvidia-glx*.deb`
- `./debs/nvidia/*kernel-source*.deb`
- `./debs/nvidia/nvidia-*-libvdpau*.deb`
- `./debs/nvidia/nvidia-*-modaliases*.deb`

4.3 Updated core kernel drivers

If you require an updated driver that is built into the kernel image itself, you will have to obtain an updated kernel image from Kernel Engineers at Canonical. If just a single module needs to be rev'ed, please see below in *Updated Kernel Module*. If you really **do** need to FISH an updated kernel package, follow these two steps: First, include the updated four debs in:

- `./debs/kernel/linux-headers*_all.deb`
- `./debs/kernel/linux-headers*_i386.deb`
- `./debs/kernel/linux-image*.deb`
- `./debs/kernel/linux-libc-dev*.deb`

If working on Ubuntu 9.04 or earlier, you will also have to update the restricted driver because they are tied together by an ABI version. Again, Canonical will have to provide this to you. Follow the directions below for the updated Broadcom Wireless driver to update it. Be sure that both updates are placed in a common I-Drive release with the structure intact.

4.4 Broadcom Wireless driver

Depending on the Ubuntu version you are updating, the procedures will differ for updating the Broadcom driver.

4.4.1 Ubuntu 9.10 or later

For Ubuntu 9.10, the Broadcom driver is sanctioned off in it's own package. Place the updated deb in:

- `./debs/bcmwl/bcmwl-kernel-source*.deb`

4.4.2 Ubuntu 9.04

Because of the way that the Broadcom wireless driver was built into Ubuntu 9.04, Canonical will have to deliver an updated linux-restricted-modules package with the new version of the driver. To FISH this package, you have to follow two steps.
First, include the updated two debs in:

- `./debs/restricted/linux-restricted-modules-common*.deb`
- `./debs/restricted/linux-restricted-modules*.deb`

Next, include an updated `./scripts/chroot-scripts/fish/04-kernel.py`. Unfortunately, you will not be able to update the kernel and restricted driver in separate releases because they are tied together by an ABI version. If you need to update the kernel too, be sure to place both together in the same release.

4.5 Updated Kernel Module

If you are just providing an updated kernel module, it must be packaged via DKMS and placed in the generic debs directory, `./debs/main`. It is important that you add a directive to the DKMS control file that indicates when the module is available in Ubuntu so that DKMS deactivates at the proper time in the field.
Please look at these external documents for information on how to create DKMS packages:
`http://wiki.centos.org/HowTos/BuildingKernelModules#head-d313bd351f90d4f25a2143b7bbcff73f927`
`https://help.ubuntu.com/community/Kernel/DkmsDriverPackage`

4.6 Scripts

If your release provides any type of postinstall script, be sure to place it in **./scripts/chroot-scripts/fish**. This will ensure that it runs at the proper point in installation.

4.7 Documentation

If you have relevant documentation for the product you are including, this documentation should be included as PDF in the `./docs` directory if your release. All documentation in this directory gets placed in a folder with a symbolic link on the userś desktop.

5 Testing a Release

When you have finished assembling an I-Drive release, it's a good idea to test it and ensure that it won't cause the factory process to fail. Depending on the status of the system in question, there are three ways to test the release.

5.1 Recovery Partition

All Dell Ubuntu releases create a recovery partition that is between 3 and 5 GB. Adding content to this recovery partition will cause that content to be used when a recovery is ran or recovery media built.

To add a release to a recovery partition, follow these steps:

1. Start with a system that has a clean installation of the Dell BTO image for the Ubuntu release in question

2. Mount the recovery partition to /mnt using `sudo mount /dev/sda2 /mnt`

3. Extract the I-drive release tarball over the recovery partition at /mnt.

4. Reboot the system and in the GRUB2 menu, select to "Reinstall Operating System"

5.2 Manual USB Injection

If you don't have a fresh copy of the BTO image loaded on the system in question, but have a spare flash drive, you can use USB Creator to help out.

1. Using USB Creator, copy the Dell BTO image for the Ubuntu release in question to your USB stick

2. Double click the flash drive to open it up in a Nautilus window.

3. Double click the I-drive tarball to open it up in File Roller.

4. Copy the contents of the tarball onto the USB stick

5. Install the system using this USB stick

5.3 Manual ISO Injection

As a last resort, you can manually extract the ISO and rebuild it using the command line.

1. Mount the ISO using `sudo mount -o loop FILE.iso /mnt`

2. Create a temporary directory using `mktemp -d`

3. Extract your I-Drive release in that directory

4. Rebuild the ISO substituting in your directory and this command:

   ```
   genisoimage
   -o file.iso -input-charset utf-8 -b isolinux/isolinux.bin -c isolinux/boot.catalog
   -no-emul-boot -boot-load-size 4 -boot-info-table -pad -r -J -jolie-long
   -N -hide-jolie-trans-tbl -cache-inodes -l -publisher Dell Inc.
   -V Dell Ubuntu Reinstallation Media /mnt/ TEMPDIR/
   ```