

Our expert

Jono Bacon is the Ubuntu Community Manager.

BIRTH OF THE UBUNTU PHONE

Back in 2004, Linux was a considerably more technical world than it is today. Even the more novice desktop flavours involved installation procedures, choosing tools, limited documentation and, at times, complex configuration. Though many of us dreamed of the power of Linux simplified for all, it was, by definition, a geeky domain.

Back in 2004 Canonical's distro came in one form: Ubuntu for the desktop. Nine years

and 19 releases later, Ubuntu is overseeing a far more ambitious mission: to create a single converged operating system that runs across the desktop, phone, tablet and TV, as well as the cloud and the server.

Of all of these platforms, right now the focus is very much on building the first version of the Ubuntu Phone. Fortunately, due to the convergent nature of this business, the majority of this work benefits each of the other devices too.

Jono Bacon walks us down Canonical's long road to the Ubuntu Phone.

Ubuntu has always been a traditional Linux distro that pulled together a collection of open source projects and integrated them to make them work well together. Although Ubuntu's core code was sourced in other projects – the Linux kernel, Gnome, *X Window System* and *Libre Office* – the integration work was complex. It involved bringing together all those disparate components, user interfaces, dependencies and more into a single consistent system.

Ubuntu quickly became the most popular desktop Linux distro and it attracted attention not just from the Ubuntu community, but from the mainstream media as well. The decisions Ubuntu made impacted on the wider Linux world almost as much as they did on the distro.

The growth in Ubuntu also generated commercial interest. Before long, the four largest OEMs in the world were shipping Ubuntu pre-installed on their systems, and the introduction of Ubuntu Server was generating interest from server hardware manufacturers and service providers too. Remember, this was back, largely, in the days when most hardware came in the form of desktops, laptops and servers.

But then the market changed. With the arrival of increasingly powerful smartphones – largely driven by companies wanting a piece of Apple's pie – the hardware world started to evolve. Before long, we had a new generation of netbooks, mobile internet devices and tablets. The mobile industry was hotting up and the hardware industry was exploring new avenues; it became obvious to Canonical that in order for Ubuntu to stay competitive, it needed to harness this new revolution, not merely dig its heels into the desktop and refuse to acknowledge change.

Promoting Unity

The start of Ubuntu's evolution into a new converged world was the introduction of the Unity desktop environment in late 2009. Shortly before it introduced Unity, Canonical had been exploring refinements and improvements to Ubuntu in the form of the messaging menu and new notifications. These changes were not merely engineering ideas promoted to shippable state, they were design-driven, user-tested features that solved many of the problems the developers had witnessed users experiencing.

These early refinements transitioned into a wider plan in the form of Unity, so named to deliver a unifying experience. Unity was intended to simplify the desktop, maximise screen real estate, and challenge many of the presumptions people had about the traditional Linux desktop world.

Canonical continued to share work that contributed to the vision of one familiar interface that would work on multiple platforms. Ubuntu for Android enabled a docked Android phone to boot a full Ubuntu desktop. Much of this work was possible due to Canonical's close relationship with Linaro, an organisation that consolidates and optimises open source Linux software, ensuring that the Ubuntu archive could run on ARM devices.

Ubuntu TV demonstrated the design principles of Unity applied to television, but the convergence story really kicked off in January, when Canonical announced Ubuntu for phones, quickly followed by Ubuntu for tablets. With

“The Ubuntu phone doesn't require any buttons on the front.”

Global network

Canonical recently announced the formation of the Ubuntu Carrier Advisory Group to help shape Ubuntu for mobile phones. With representatives from all over the world the current CAG membership list includes Deutsche Telekom, Everything Everywhere, Telecom Italia, China Unicom, Korea Telecom, LG UPlus, Portugal Telecom, MTN Group, SK Telecom, Verizon Wireless and Smartfren.



Ubuntu available side-by-side for comparison on a desktop, phone, tablet and TV, the convergence was visible and intriguing for onlookers, particularly at the Mobile World Congress (MWC), where the entire vision was exhibited.

Phones 4 Ubuntu

One of the unique features of the Ubuntu phone is that it doesn't require any buttons on the front. Instead, you swipe the edge of the screen to access different features. After pressing the Power button, a swipe of the left edge of the screen brings up the familiar Unity launcher with a stacked collection of icons to load different applications. As with the desktop, these icons collapse as you slide up and down to provide space for as many of your favourite applications as you like.

At the bottom of the launcher is an Ubuntu button, which unlocks the phone and takes you to the Home view. In much the same way as the Dash works on the desktop, there are different views (called Master Scopes) for Home, Apps, Music and Video. Each of these scopes provides access to content either on your devices, in the cloud, or available to purchase. Sliding

left and right jumps between these different content types and you can then slide up and down to browse, tap an item to view more information or view/buy/download. This functionality is subtle but important: it integrates content right into the phone. You no longer need to load an application to access your music, videos or apps – they're all right there at the core of the phone.

If you load an application, such as Gallery which shows your photos and videos, you can swipe up from the bottom of the screen to access that app's functionality. For example, if you're viewing a photo, an upwards swipe will show options such as Edit or Share.

The HUD works in much the same way as it does on the desktop, except instead of pressing Alt, a long upwards slide brings up a small search icon that you can use to see the phone's equivalent HUD functionality.

If you now open a variety of applications, such as the browser, gallery and camera app, the right edge comes into play. Swiping from the right switches between the different open applications. This makes multitasking as simple as a swiping from the right to the left of the screen. On the

» Canonical is putting a lot of faith in its dream of convergence – that is, one OS to control every device you own.



› Ubuntu for phones looks slick and elegant.

tablet, sliding out from the right edge shows the side stage, which enables you to run phone applications within a smaller window and be able to easily interact with tablet applications (such as dragging content from the tablet web browser and saving it in the note-taking app in the side stage).

The top and final edge of the screen is where you configure the phone's most common parts. In much the same way as you have the indicators in the top bar on the Ubuntu desktop, you have the same indicators on the phone; sliding down from the top screen lets you navigate between them and view controls for configuring those items. As an example, you can select a wireless network, change the volume, adjust power settings, and more.

As with the Ubuntu desktop the messaging menu is also present. This little envelope icon lights up when you receive a text message, Facebook/Twitter direct message, missed call, or other content. All of your messages are in one place and when you select a message you can respond from within the messaging menu. This means that you can reply to messages without having to load a separate app or even exit an existing one.

Another dimension

The development of Ubuntu across these devices has not only involved producing a single and consistent design experience in Unity but also a great deal of convergence and unification at the software level too.

Traditionally the Unity that has shipped on Ubuntu Desktop has been the Unity 7 codebase, written using

“A single Unity codebase displays the appropriate UI.”

Compiz and the Nux toolkit (the latter designed and built specifically for Unity). The new converged Unity is being rewritten using the popular Qt QML framework. This codebase is known as Unity 8.

Importantly though, there is not a different Unity for each device form-factor. Instead there's a single Unity codebase that displays the appropriate user interface depending on the screen size and configuration of the device. As an example, if you take Unity 8 and run it on a device with the dimensions of a phone, you will see the phone experience. If you then widen the same instance of Unity 8 to the dimensions of a tablet, you will see the tablet experience. Likewise, the same will happen for desktop dimensions.

This has a few notable benefits. Firstly, when a bug is fixed or performance is further improved in Unity 8 it instantly applies to all form-factors. Secondly, it means that Unity can easily and effectively switch between these user interfaces when required. As an example, you might want to take your Ubuntu phone and plug it into a monitor with an HDMI cable and then see the Ubuntu desktop experience. This means that although much of the work going on right now is getting Unity ready for the phone, all of that work ultimately benefits all of the different form-factors.

Another important piece in the stack is the *Mir* display server. *Mir* provides a next-generation replacement for the ageing *X Window System* that has been in commission for the past 25 years or so.

Although some were disappointed that Unity didn't move to *Wayland*, the choice of building *Mir* was born out of creating something lighter than *Wayland* in terms of breadth of features and more orientated towards these multiple devices.

Taking it for a test drive

If you have a Galaxy Nexus or Nexus 4 phone and want to give Ubuntu for phones a whirl, you first need to understand that it is unfinished software. You should back up any data and applications you need as Ubuntu is going to replace it with a new operating system. Fortunately, you can revert to Android if you need to later (albeit, without your original data, hence the back-up!).

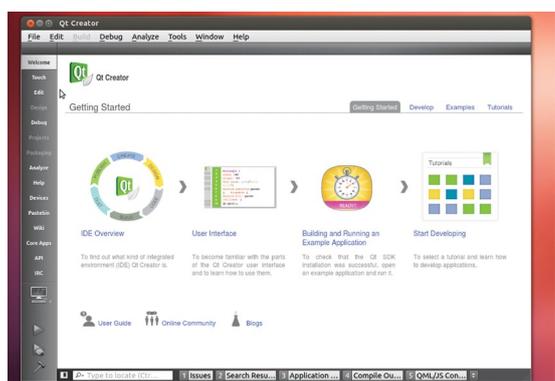
First install the software from the PPA:

```
sudo add-apt-repository ppa:phablet-team/tools
```

Writing Ubuntu phone apps

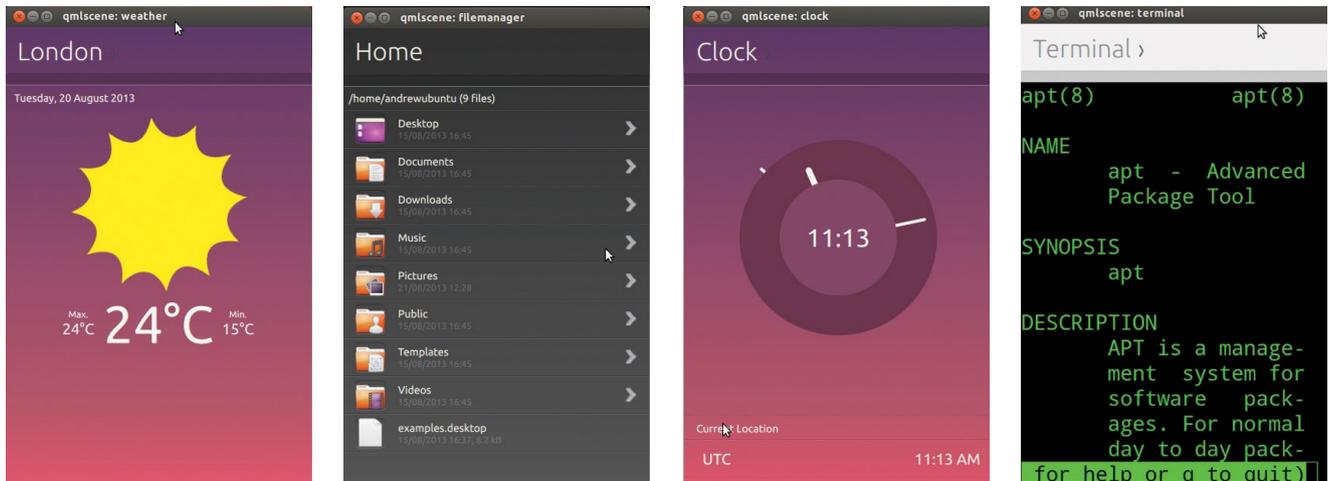
Ubuntu for phones won't be able to take advantage of the huge number of Android applications out there, so it'll need developer buy-in if it's going to be a success. With this in mind, Canonical has tried to make it as easy as possible for developers to get involved. If you want to write an application for Ubuntu running on any of these different devices – not just for phones – you can use the Ubuntu Software Development Kit (SDK) that has recently seen its beta release.

Applications can be written for Ubuntu using QML, HTML5 and OpenGL, and the SDK provides a complete integrated development for creating these applications, complete with comprehensive documentation. Best of all, the SDK includes detailed tutorials as well as a small selection of example applications. To get started, visit <http://developer.ubuntu.com>, download the Ubuntu SDK and get started with the tutorials.



› Ubuntu's SDK make a lot more sense (and is a hell of a lot easier to install) than its Android equivalent.

Ubuntu core apps



When Ubuntu for phones was announced there was also a request for community participation to help write the core set of applications that will ship with the phone. The goal of this campaign was to write the main calendar, clock, weather, calculator, email client, RSS reader, music player, file manager, document viewer and terminal applications that most users will need.

More than 1,500 developers volunteered to contribute to the project and since then the core apps have been making steady and significant progress, with many reaching core feature completion.

In addition to core apps we added *Sudoku*, *Dropping Letters* and *StockTicker* apps that other community members were working on to

the line-up. Members of the team at Canonical have been working with each of these community members to help coordinate the projects to get them ready in time, and the community has done a fantastic job. You can track the progress of the project at <https://wiki.ubuntu.com/Touch/CoreApps> – be sure to come and participate.

Now do the following:

```
sudo apt-get update
sudo apt-get install phablet-tools android-tools-adb android-tools-fastboot
```

You now need to unlock the device (if your device is unlocked you can skip this). With the device powered off, power on the device by holding the Power button + volume up + volume down. The device will boot into the bootloader.

Now plug the device into your computer via the USB cable. On your computer, press Ctrl+Alt+T to start a terminal and type:

```
sudo fastboot oem unlock
```

Now on the device screen, accept the terms of unlocking. Your device is now unlocked! Now boot the device by pressing the Power button (pointed by an arrow with Start on the screen). Next, you need to enable USB debugging. This varies depending on which version of Android you are running:

» **Ice Cream Sandwich (version 4.0)** Go to Settings and turn on USB Debugging (Settings > System > Developer options > USB debugging).

» **Jelly Bean (versions 4.1 and 4.2)** Enter Settings, About [Phone|Tablet] and tap the Build number 7 times to see the Developer Options.

On either Android version you must then enable USB debugging via Settings > Developer options > USB debugging. You should now accept a host key on the device. On the computer type:

```
adb kill-server
adb start-server
```

Depending on the installed Android version, a pop-up will appear on the device with the host key that needs to be accepted for the device to communicate with the computer.

Now save the version of the current image on the device as a reference to revert back to. The version can be found by going to Settings > About Phone > Build Number. Fortunately you only need to run through that unlocking process once. Now you can install the Ubuntu phone image to the device easily by just running *phablet-flash*.

The latest version of Ubuntu phone will now be downloaded and automatically installed to the device. When it is complete the phone will automatically boot into Ubuntu phone. If

you have your SIM card in the phone you should be able to make calls, receive text messages, and use the other features of the phone.

As the engineering team works hard to have a first cut of Ubuntu for phones ready by October, this is a story you are sure to want to keep an eye on – you might even want to grab a spare phone so you can have a play about with it. Although there is a long road ahead, the journey has already been rewarding... who knows what the future holds? **LXF**

“When a bug is fixed, it instantly applies to all form-factors.”

Join the community

If you would like to find out more about Ubuntu for phones, keep up-to-date with current progress or get involved with the project, have a look at the following resources:

- » Ubuntu For Phones Homepage www.ubuntu.com/phone
- » Ubuntu Developer Portal <http://developer.ubuntu.com>

» Google+ Ubuntu App Developers <https://plus.google.com/u/0/111697084657487423167/posts>

- » Facebook Ubuntu App Developers Page www.facebook.com/ubuntuappdev
- » IRC Channel [#ubuntu-app-dev on Freenode](https://freenode.net/#ubuntu-app-dev)