Xubuntu Contributor Documentation

The Xubuntu documentation team. Xubuntu and Canonical are registered trademarks of Canonical Ltd.

Xubuntu Contributor Documentation

Credits and License

This documentation is maintained by the Xubuntu documentation team.

The contributors to this documentation are:

- Kev Bowring (flocculant)
- Krytarik Raido (krytarik)
- Pasi Lallinaho (knome)

This documentation is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE AS DESCRIBED IN THE DISCLAIMER.

Welcome!

This documentation is a reference for all Xubuntu contributors. The chapters of this documentation provides information on the development processes - both social and technical - that the Xubuntu contributors use as a guideline in their work.

There are two main appendices for this documentation:

- Appendix A, *Strategy Document*, which is the primary guideline in Xubuntu development.
- Appendix B, *Common Reference*, which describes many technical tasks that the Xubuntu developers need to use continuously.

> ### Note
>
> This documentation is written with the assumption that the user knows the basics of working with Xubuntu. For example, it's expected that the reader knows how to work with a terminal and understand when they need to use one, without an explicit mention.

Table of Contents

Chapter 1. Getting Started

Important Places

#xubuntu-devel

The IRC channel #xubuntu-devel is where most of the daily communication and cooperation about Xubuntu development happens. It's also the place to get to know people and get yourself known.

Status Tracker

The development for the current Xubuntu release is tracked in the Xubuntu development tracker [http://dev.xubuntu.org/]. In addition to showing the current status for blueprints and their individual work items, the tracker displays some general progress. Finally, you can access the team calendar, Xubuntu wiki, development mailing list and join the development IRC channel #xubuntu-devel from it.

Part I. Subteam Documentation

Table of Contents

Chapter 2. Development

Xubuntu packageset

The Xubuntu team maintains a subset of packages in the Ubuntu archive. The Xubuntu packageset for the current development release can be found listed at the packageset archive [http://people.canonical.com/~ubuntu-archive/packagesets/].

Development PPAs

The Xubuntu developers use several PPAs (personal package archives) for staging, testing and previewing new packages and package versions. These PPAs can be found under the Xubuntu Developers [https://launchpad.net/~xubuntu-dev] page on Launchpad.

Currently, the official PPAs for development use are:
PPA: Xubuntu Staging [https://launchpad.net/~xubuntu-dev/+archive/ubuntu/xubuntu-staging]
Description: A staging PPA for Xubuntu. The packages and package versions in this PPA are being prepared for inclusion in Xubuntu. The packages are uploaded for the convenience of people testing these new package versions and features. Some of them are daily builds used in testing.
PPA: Xubuntu Daily Builds [https://launchpad.net/~xubuntu-dev/+archive/ubuntu/ppa]
Description: Git/Bzr daily builds for packages related to Xubuntu and/or Xfce4.
PPA: Xubuntu Extras [https://launchpad.net/~xubuntu-dev/+archive/ubuntu/extras]
Description: A preview PPA for Xubuntu. The packages in this PPA are considered for inclusion in the Ubuntu repositories and/or Xubuntu at a later time. The packages are uploaded for the convenience of people willing to preview the new features.

Chapter 3. Documentation

The Xubuntu documentation oversees the user documentation and contributor documentation as translations for the documentation and other packages important to Xubuntu. In addition the team works together with other teams in producing content and more.

Tasks

User documentation

The documentation team works with the developers to ensure any new features that are part of the core functionality of the operating system are covered in the documentation. In addition, the documentation team makes sure features highlighted in the installer slideshow or any other marketing material are appropriately documented. When completely new sections are written, the documentation team makes sure the Xubuntu team has a chance to review and propose changes and fixes to the new sections. When any documented features are removed or changed, the team makes sure the documentation does not refer to any removed functionality.

> **Note**
>
> The team should be aware of changes to new releases, particularly on LTS releases and potential SRUs, as those might affect documentation.

The documentation team is responsible for making sure the documentation is updated before the Documentation String Freeze prior to uploading. Once the upload is done, the team coordinates translation efforts and makes sure the documentation gets another upload before the Translation Freeze. The team should also request the online documentation [http://docs.xubuntu.org/] to be updated shortly before the release.

The documentation team should also look to improve and expand the existing documentation, especially during the development cycles when most uploads are done for maintenance purposes. The team should involve the community in these efforts the best way they see fit.

Contributor documentation

The documentation team lead the efforts to write the contributor documentation. While the various subteams are responsible for their own sections, the documentation team should help the teams to get their content pushed to the main branch and where needed, help contributors to format the content in the DocBook markup.

The online contributor documentation [http://docs.xubuntu.org/contributors/] should be kept updated especially when major changes are made; the documentation team should request an update whenever they see fit.

Other tasks

In addition to the above, the documentation team is strongly encouraged to do the following:

- Work with the Quality Assurance team to write and maintain testcases

- Work with the website team to improve the website content and write FAQ or other documentation related articles
- Work with the marketing team on various projects, like flyers

Working with the documentation

Editing the documentation

First make sure you have the Bzr branch **lp:xubuntu-docs** and the build dependencies for **xubuntu-docs**.

Changes to the user documentation should be made to the .xml files under **user-docs/C/**. Once edited and saved, push changes to your local branch as detailed in the section called “Using Bazaar (Bzr)”.

Similarly, changes to the contributor documentation should be made to the .xml files under **contributor-docs/C/**.

Notes on the contributor documentation

For various reasons, the contributor documentation is more complex in structure than the user documentation. Particularly, pay attention to the following when working with the subteam documentation:

- If you want to create an introductory paragraph that is shown on the subteam main page, simply do not wrap it inside a **<section>** element. Otherwise this will be chunked into a separate HTML file.
- If you want a subsection of a section to be visible in the Table of Contents, add the **<role="toc">** attribute for the **<section>** element. For an actual example, see the **qa-*** files.
- The sections are chunked up to the first level; this means all direct **<section>** ascendants of **<chapter>** elements will be chunked into separate HTML files. If you want to avoid this (for example if your chapter is very short), insert the **<?dbhtml stop-chunking?>** after the **<title>** element of your chapter. Note that doing this will not stop the Table of Contents from being printed with each section. You can use the **<simplesect>** element again to hide sections from the Table of Contents. If you do this, make sure you don't insert any elements after the **<simplesect>** elements in your chapter, as this will invalidate the document.

Translations

Packages to be translated

These are packages that the Xubuntu team considers important, and should take precedence when translating.

Packages that only exist in Xubuntu

These packages only exist in Xubuntu, so their translation is solely dealt with by Xubuntu translators. These are of high importance for the Xubuntu team.

- Xubuntu Documentation [https://translations.launchpad.net/xubuntu-docs]
- Xubuntu Default Settings [https://translations.launchpad.net/xubuntu-default-settings]
- Ubiquity Slideshow for Xubuntu [https://translations.launchpad.net/ubuntu/zesty/+source/ubiquity-slideshow-ubuntu/+pots/ubiquity-slideshow-xubuntu]

Used by Xubuntu

These packages are used by Xubuntu and provide features to the Xubuntu experience. When these are translated, the base system is immediately more usable for more people.

- Catfish [https://translations.launchpad.net/catfish-search]
- LightDM GTK+ Greeter [https://translations.launchpad.net/lightdm-gtk-greeter]
- LightDM GTK+ Greeter Settings [https://translations.launchpad.net/lightdm-gtk-greeter-settings/]
- MenuLibre [https://translations.launchpad.net/menulibre]
- Mugshot [https://translations.launchpad.net/mugshot]
- Whiskermenu [https://www.transifex.com/projects/p/xfce4-whiskermenu-plugin/]

Translation guidelines for Xubuntu packages

These guidelines apply mostly to the Xubuntu documentation. They can generally be applied to any translation with minor modification. If unsure, ask the Documentation team members for assistance.

The language-specific translation teams [https://wiki.ubuntu.com/Translations/Contact/Teams] usually want to encourage some conventions; please contact the appropriate language-specific team before translating.

More general translation guidelines can be found at the Launchpad translations guide [https://help.launchpad.net/Translations/Guide].

Maintain translation validity

Tags should not be translated. Where you see a tag in the source string (for example: "<xref linkend="&xubuntu-web;" />"), the tag and anything inside the tag itself should not be translated. It is very important to copy and paste these tags exactly as they appear in the source string. Note: The exception to this is where a tag contains a URL that has an equivalent, translated version. In those cases the translator should use discretion about whether to localise the URL.

Entities should not be translated. Where you see a phrase expressed like "&gt;" or "&xubuntu-web;", do not change this.

Always maintain the order of tags. When you see different tags nested inside each other (for example: "<menuchoice><guimenu>System</guimenu><guimenuitem>Administration</guimenuitem><guimenuitem>Users and Groups</guimenuitem></menuchoice>"), you must always preserve that order carefully - copying and pasting from the source language (English) is the best way to ensure that you do this correctly.

> **Note**
>
> Different programming languages and software might have different variable syntaxes; make sure you are familiar with the appropriate software variable syntax when translating. More information on this can often be found in the translation string comment.

Follow good and existing conventions

Use application and label names found in the (graphical) UI. When translating UI labels and application names, use the term or name that is used on the graphical UI. This makes it easier for the user to follow the instructions. Running the system in both the translation target language and English can help with this.

Do not change the amount of spacing. When translating, do not change the amount of spacing inside our outside tags. Keeping the spacing as it is makes the technical reviewing of the translation easier.

Do not change the type of quotes used inside the tags. The double quotes (") doesn't have the same significance in a DocBook, XHTML or XML tag as the guillemets (« »). Using different kind of quotes can potentially develop into validity problems as well.

Never change a tag to another. If you think a tag in the source is invalid or not semantic, file a bug against the package itself instead of changing it in the translation.

Chapter 4. Quality Assurance (QA)

> **Note**
>
> There are various QA trackers, wiki and Launchpad pages, the links to which can be found on the section called “Useful QA Links”.

The Xubuntu QA team was formed to ensure that the quality of a released Xubuntu conforms to the parameters laid out in the Xubuntu Processes documents. In order to successfully accomplish this, close team working relationships, especially with the development team and the Xubuntu Council are paramount. Xubuntu's success is based on close working amongst all of its various teams.

In addition, the team gives people contributing through testing of Xubuntu the opportunity to become part of the Xubuntu Team [https://launchpad.net/~xubuntu-team]. For that to be of practical use, the QA team should keep an eye on testing reports on the trackers and propose users they have seen taking a keen interest in the work of the QA team.

Excluding testing itself, control of the testcases that we use for ISO and Package testing, along with communicating the testing requirements for any particular development cycle (hereafter cycle) to the community, makes up the bulk of the teams work in any given cycle.

Members of the QA team should check the current Xubuntu QA blueprint, and assign themselves to tasks they feel able to undertake.

Testing

At the start of a cycle, the Release Team will discuss which ISO Milestones we will participate in. Then, during a Community Meeting, members of Xubuntu Team will discuss and then ratify Xubuntu's participation during the cycle.

Along with general testing of our OS, dealt with further in the section called “Exploratory Testing” and the section called “Using Development PPAs”, further responsibilities lie with ensuring that:

- Sufficient testing takes place prior to ISO Milestone releases
- Sufficient package testing takes place following calls to testers
- Sufficient testing has taken place by Final Release
- Bugs reported to our trackers are confirmed, or where unable to confirm, further information is requested from the reporter
- Where appropriate, confirmation of bugs can be asked of members of Xubuntu Team in the team devel IRC channel

Members of the QA team might find it useful subscribing to the Xubuntu Bugs [https://launchpad.net/~xubuntu-bugs] team in order to be aware of bugs being reported by users.

Testing Infrastructure

Getting ready to test

To successfully test for Xubuntu, you will need at a minimum 4 things: a Launchpad account, mailing list subscription, an image and something to test it on. This page will give you the basic information in order to do those things and more.

Contacts

You need a Launchpad account, in order to join the necessary team(s) on it, and log into the 2 QA trackers.

We use one mailing list to detail what we intend to test during a cycle, we use the same list when we are calling for tests. This list is also used by the team to discuss the development of Xubuntu. You will need to join the Xubuntu Devel Mailing [https://lists.ubuntu.com/mailman/listinfo/xubuntu-devel] list.

While it is not necessary to do so, we make much use of IRC, so having an account you can use to access the main #xubuntu-devel channel will help you, alternatively you can access the channel via the IRC tab of the Xubuntu development [http://dev.xubuntu.org/] tracker.

Join the Xubuntu Testers Launchpad [https://launchpad.net/~xubuntu-testers] team.

Consider adding your details to the Hardware [http://wiki.xubuntu.org/qa/teamhardware] page. We can use this data to request specific testing from specific testers.

Getting the ISO

ISOs can be obtained through a few methods. The simplest method is to visit the ISO [http://iso.qa.ubuntu.com/] tracker. Follow the testcases through version to Xubuntu's group of tests. Then choose either 32 or 64 bit. Download links are available there.

When testing more regularly, you might find (especially where bandwidth is an issue for you) using the zsync [https://help.ubuntu.com/community/ZsyncCdImage] link more appropriate. This is will just grab the difference between your local copy and the current one.

Methods of testing

Now you've set up your basic contact(s) and have your ISO, you are in a position to test it.

There are two main methods of testing the ISO:

- Using a virtual machine enables you to carry on using a normal release and to test our ISOs. You can install a virtual machine by directly using your downloaded ISO.
- Install the ISO to hardware where that possibility is available. This is more important towards the end of the development cycle. You will necessarily need to have the ISO in some bootable medium. A member of Xubuntu's Website team details a simple method for preparing a bootable USB [http://princessleia.com/journal/2016/03/xubuntu-16-04-iso-testing-tips/] stick with Gnome Disks.

> **Note**
>
> While it is possible to use VirtualBox, past experience has shown that issues found when testing in VirtualBox are hard to replicate elsewhere. You might want to look into using KVM [https://

> help.ubuntu.com/community/KVM]. Members of Xubuntu QA have some experience with using this for testing and can help.

Reporting your result

There are two QA trackers we use.

- The ISO tracker mentioned above for any testing done with an ISO either an install or an upgrade test.
- The Package [http://packages.qa.ubuntu.com/] tracker, which is used when we are testing applications during a cycle.

These trackers form the backbone of what we are able to gain from your testing. It is important that you report your results.

The next two sections of these documents go into more detail on how you can test for us and how to report bugs for us.

Package and ISO Testing

ISOs

In general, links for specific ISOs for daily or milestone use can be found at the ISO Tracker [http://iso.qa.ubuntu.com/]. A link exists for the current [http://cdimage.ubuntu.com/xubuntu/daily-live/current/] development daily ISO. This can be useful, especially when setting up methods to download your testing ISO, such as with zsync [https://help.ubuntu.com/community/ZsyncCdImage]

> **Note**
>
> You can find the last known boot and installation status of our ISOs at QA Status [http://dev.xubuntu.org/#tab-qa] of the development status tracker

Exploratory Testing

In addition to testing the images and software with predefined testing actions (static testing), community members can take part in exploratory testing as well.

In essence, exploratory testing means running the development release and doing your daily tasks with the system and finally, filing bugs when you find them. This allows Xubuntu to get a much larger spread of testing than is possible with predefined tests.

A useful way to deal with this is dual (or multiple) boots. By dual (or multiple) booting you can ensure you are able to access at all times a working version while testing the development system. In this way you can do as much work in the development release as is practical – all the while watching for bugs and regressions in the development release.

Issues which the development team are keen to address include, in addition to normally reported issues, usability bugs, missing icons, inconsistent functionality.

Using Development PPAs

We have 3 PPAs which we use regularly to test packages. These are:

- Shimmer Themes [https://launchpad.net/~shimmerproject/+archive/ubuntu/daily] for daily builds of the Shimmer Project's projects
- Xubuntu Staging [https://launchpad.net/~xubuntu-dev/+archive/ubuntu/xubuntu-staging] for packages and package versions that are being prepared for inclusion in Xubuntu
- Xubuntu daily builds [https://launchpad.net/~xubuntu-dev/+archive/ubuntu/ppa] for daily Git and Bzr builds for packages related to Xubuntu and/or Xfce4

> **Note**
>
> At times the development team use other PPAs to test specific packages. Details of these will be made known to testers when appropriate

Installing these three PPAs means that you will be using and testing packages that developers are currently working with, this means that regressions found by you will not be present once they are released.

Removing an in use PPA. Using ppa-purge causes apt to disable a PPA source list and then change affected packages back to the default versions.

- Run **`sudo apt-get install ppa-purge`**
- Run **`sudo ppa-purge ppa:ppaname`**

Reporting bugs with PPAs can be problematic, please see further information on reporting bugs with PPAs in the section called "Reporting bugs with PPAs".

Static Testing

Testing the development versions of Xubuntu and packages in it helps keep up the every day quality. If you have spare hardware resources or are able to run a virtualized testing environment, you can help.

- Check the devel mailing list close to the start of the cycle, where the decision as to which milestones we will participate in will be discussed.
- While milestone testing is in progress, please watch for rebuilds. A rebuilt ISO will have **.something** appended to its name (e.g. **.1**). These will take place either when we have reason to rebuild, e.g. following a bug fix landing, or more likely when the Ubuntu Release Team have cause to do so.

Milestone Testing

Stand alone image testing for milestones is useful in the 2 days before the release. To see how these milestones are scheduled, refer to the Ubuntu release schedule.

- Live Session: Boot with the image and ensure that basic application testing (open, close, saving etc.) passes.
- Installation tests required pass. These test only the installation.

Daily Testing

The importance of daily image testing lies in the main in knowing that boot or installation regressions haven’t appeared. Where it appears they have, if you are able to boot with a different flavours image, this can help prove a global or local to Xubuntu issue. If unsure contact the Xubuntu team, preferrably on IRC.

- Live Session: Boot with the image and ensure that basic application testing (open, close, saving etc.) passes.
- Installation tests required pass. These test only the installation.

Upgrade Testing

About half way through any development cycle (around the beta milestones) we need to test upgrades from the previous release to the new development version. When the development version is intended to be the next LTS release, we need to test both upgrade paths: LTS to LTS and regular to LTS.

The ISO tracker has a section for upgrade tests which does not change daily, but runs from Friday to Friday. This enables us to test upgrades for a week rather than a day. During these tests it is preferred that the image used for testing is the one built on the first Friday, except where bugs are reported and fixes landed.

- Upgrade: tested using the update-manager
- Upgrade (image): tested using the daily image

Package Testing

For releases where we are making use of the Package Tracker, calls for package testing will be made to the xubuntu-devel mailing list as required. This could be a call from QA or Developers.

A schedule of planned tests will be mailed to the Xubuntu Devel mailing list close to the cycle start. Reminders of upcoming package test requirements are mailed to the Xubuntu devel mailing list as required. *Specific developer testing requirement* are mailed as they are available, often these packages will be those found on one of our PPAs.

Dealing with bugs

Using tags for Xubuntu bug reports

So that we can easily find bugs for Xubuntu during a development cycle, there a few tags you can add to your bug reports. We assume that a tag for the cycle codename is included, if it is not please add one.

- xubuntu-exp: always add this tag
- ppa: add this tag when you've reported a bug against a package from one of the 3 develeopment PPAs
- usability: add this tag if the issue you are reporting is not a standard bug, but one which might be deemed wishlist, such as multiple selection of results in the Catfish results window

Normal bug reporting

Anybody using Xubuntu can file (or in other words, report) bugs, even with no experience at all. Filing bugs is important because in some cases, bugs can go unnoticed unless an end user files them.

Before submitting a bug, you should look at the existing bug reports [https://launchpad.net/ubuntu/+bugs] and release notes to verify the bug hasn't been reported already. If the bug has not already been reported, you should file a new bug report. It is sensible to read through the bug reporting guidelines [https://help.ubuntu.com/community/ReportingBugs] before filing your first bug.

In the most situations, it is easiest to file a bug by opening a new Terminal and type ubuntu-bug package-name, where package-name is the package you want to file bug against. If you don't know which package you should file the bug against, refer to the instructions on finding the right package [https://wiki.ubuntu.com/Bugs/FindRightPackage]. When filing the bug, it is better to have too much information than too little.

> **Note**
>
> If you are unsure of the package name to report against, you can use your mouse to click on the application window; in a terminal run **`ubuntu-bug -w`**, click to close the message window and then click on the application window.

Reporting bugs with PPAs

When you encounter a bug with a package from a PPA, you'll need to file the bug report manually. Once you've got the offending package name, go to **https://bugs.launchpad.net/ubuntu/+source/PACKAGENAME/+filebug**.

Crash logs related to the package can be found in ***/var/crash/***. This directory sometimes requires superuser (**sudo**) permissions to view and extract the contents. These logs can be particularly valuable to include in your bug report, but be sure to review them as they may contain sensitive or personally identifiable material.

Be sure to follow the above general guidelines, and also add the ppa tag so it is clear to developers that this is an unsupported package.

Triaging bugs

Triaging bugs is getting bug reports in a state where they are useful for developers by making sure bug reports have useful titles, descriptions, appropriate logs and more. To get started, read the page on triaging bugs [https://wiki.ubuntu.com/Bugs/Triage] in the Ubuntu wiki.

After you have made yourself familiar with the aspects of triaging bugs, you can start triaging Xubuntu bugs [https://bugs.launchpad.net/~xubuntu-bugs/+packagebugs]. Please note that the bug status and importance can only be changed by members of the Ubuntu Bug Control [https://wiki.ubuntu.com/UbuntuBugControl] team – in the beginning, you will need to report triaged bugs for the team in #ubuntu-bugs on Freenode. After you have demonstrated your ability to triage bugs, you will gain more responsibilities from the Bug Control team.

As you gain experience on triaging Xubuntu bugs, you may want to take a look at new bugs that mention Xubuntu. Triaging new bugs is recommended for those who are more familiar with both triaging and Xubuntu generally, since not all of the new bugs mentioning Xubuntu are actually Xubuntu bugs.

Forwarding bugs upstream

When bugs are forwarded upstream, all users of the relevant software will benefit from the debugging, triaging and patching work carried out by bug triagers and developers.

Once it has been determined that a bug is not caused by a change in Xubuntu, bugs can be forwarded to their respective upstream projects to be reviewed. For example, all appropriate Xfce bugs should be filed in the Xfce bug tracker [https://bugzilla.xfce.org/]. In addition, to track the status of the bug in Ubuntu, the bug in Launchpad should be linked with the upstream bug.

The Ubuntu wiki has an extensive page on forwarding bugs upstream [https://wiki.ubuntu.com/Bugs/Upstream] along with instructions on how to report bugs in the upstream bug trackers. Detailed information on the way to add an upstream bug to Launchpad can be found on the Bugs/Watches [https://wiki.ubuntu.com/Bugs/Watches] page on the Ubuntu wiki.

Working with trackers

The initial decisions on what and when to test, will be taken following discussion with the members of the Xubuntu Release [https://launchpad.net/~xubuntu-release] team.

At the start of a cycle, the QA team needs to ensure that:

- Image testcases we use are still correct.
- When the intention is that package testing will take place during the cycle, package testcases required during the cycle are still correct.
- The testsuites on the Package Tracker make sense for what we intend to test during the cycle. Differences between regular and LTS releases are often, but not always, needed.
- Scheduling of ISO, and when appropriate Package, testing should take place amongst the QA team.

When there are changes to a package we test, following for example a bug fix, a further check of the testcase involved should take place. Further testing calls for that package should be made to check for regression during the cycle.

When a package during test constantly fails, or bug reports indicate a failure in a package for something not tested, the testcase for that package can be disabled temporarily. The QA Lead is responsible for ensuring tests are both disabled and re-enabled when appropriate.

Xubuntu ISOs are built and added to the tracker at approximately 02:00 UTC. This time can be changed by amending our build times on the ubuntu-cdimage crontab [http://bazaar.launchpad.net/~ubuntu-cdimage/ubuntu-cdimage/mainline/files/head:/etc/] and proposing the change.

Working with testcases

Information on the basic method of working with testcases can be found at the Ubuntu QA Team Manual Testcases [https://wiki.ubuntu.com/QATeam/ContributingTestcases/Manual] page.

We are only concerned with a specific set of tasks: grabbing the branch, making edits and then pushing the changes we need to the main branch. We have people in the Testcase Admin team for the LP

Testcases, in addition any member of our Release Team can edit the tracker, this helps ensure that changes are moved through to the trackers quickly.

To edit a testcase:

- Report [https://bugs.launchpad.net/ubuntu-manual-tests/+filebug] the required change as a bug against the testcase project
- Assign yourself to the bug
- Create a local branch: **`bzr branch lp:ubuntu-manual-tests`**
- Make changes to the testcase in your local branch
- Commit the changes: **`bzr commit -m "Fix LP bug #BUGNO."`**
- Push to a personal remote branch: **`bzr push lp:~username/ubuntu-manual-tests/bug-BUGNO`**

Once pushed to a personal remote branch, propose the change for merging - the section called “Submitting merge proposals”.

Respond to any requests for changes when asked by the Testcase Admins in order to get the required change through in a timely manner.

Communicating with Testers and Users

While we have two sets of people in the community that we contact about required testing, Testers will get regular contact from us, but we should only, in general, call on Users at later stages.

The QA Lead will be an administrator on the Testers Launchpad page and can contact those users via LP. Copies of testing calls sent to the devel mailing list should go to this group each time.

Any member of the QA team can:

- Mail the devel list with a testing call
- Just prior to an upcoming ISO Milestone testing call, warn on the devel list
- Make an ISO Milestone testing call

Xubuntu Users will be contacted for ISO testing at later ISO Milestones, at the earliest the Beta 1 milestone, depending on the state of the current ISO and our packages.

Release Responsibilities

Much of the responsibility for the QA team at any release lies with the QA Lead.

However, *any* member of the QA team can:

- Work with the testing wiki Release Note [https://wiki.ubuntu.com/Xubuntu/Testing/ReleaseNote]
- Check status of bugs listed on the above draft

- Check status of work items on the QA blueprint, marking as appropriate

Post Release Tasks

Following release, there are a few tasks that need to be done before the next release cycle begins.

- QA Lead should set up the blueprints for both the QA team and the Bugs that the whole team uses
- Check that the draft Release Note is up to date

Chapter 5. Website

Introduction to the Xubuntu web presence

Xubuntu.org

The website is the main output for end-users. The content on the website should be easily understandable and approachable.

The website is a WordPress instance on a server maintained by the Canonical IS. The website content is overseen and administrated by the Xubuntu website team. The website team can submit tickets to the Canonical IS for any problems. Bugs that are related to the website can be submitted on Launchpad, under the bugs for xubuntu-website [https://bugs.launchpad.net/xubuntu-website/+bugs].

The source code for the website is available at **lp:xubuntu-website** (the theme) and **lp:xubuntu-website/plugins** (the plugins).

Xubuntu wiki

The Xubuntu wiki is used for some developer communication and serves as an archive for older release roadmaps and specification.

Chapter 6. Processes

This section describes some of the community processes for Xubuntu development. The processes should be used as guidelines: the main goal of this section is to describe *what we do*, not so much to describe *what we should do*.

In addition to the Xubuntu processes described below, there are several generic Ubuntu processes that are useful for the team and described elsewhere:

- Freeze exception process [https://wiki.ubuntu.com/FreezeExceptionProcess] for getting to upload changes after a relevant freeze
- Sponsorship process [https://wiki.ubuntu.com/SponsorshipProcess] for getting new package uploads in the archive

Meetings

Chairing a meeting

The community meetings are held in **#xubuntu-devel** and chaired by the Xubuntu team leads. The chairing team leader is cycled based on the team name. The actual upcoming chair list can be found in the comments of the Meetings page.

The chair is responsible for scheduling and announcing the meeting on the development mailing list, and for updating the wiki with the meeting time.

During any of the sections in the meeting, the chair can assign action items to individuals or teams with their permission. The chair should take care to copy these items to blueprint work items where applicable.

The meeting minutes will need to be added to the wiki after the meeting has ended. To do this, go to the Minutes startpage [https://wiki.ubuntu.com/Xubuntu/Meetings/Archive/Minutes/] and use the input box for creating a minutes page. You will get the content for the page directly from meetingology Moin output URL.

After you have created a minutes page, don't forget to update the include in the Meetings main page [https://wiki.ubuntu.com/Xubuntu/Meetings/].

Website updates

Update on release

- Update screenshots on the Screenshots [http://xubuntu.org/screenshots/] page
- Rotate the In the Press [http://xubuntu.org/press/] section
- Add new release to the Help & Support [http://xubuntu.org/help/] and Get Xubuntu [http://xubuntu.org/getxubuntu/] pages

- Change front page widget to point to a new release
- When one is set, remove the widget mentioning testing from the front page
- When one is prepared, release a FAQ blog article

Update on point releases

- Update download links on the Get Xubuntu [http://xubuntu.org/getxubuntu/] page
- Update the front page widget

Update on release EOL

- Update the supported releases list on the Help & Support [http://xubuntu.org/help/] page
- Update the documentation subdomain [http://docs.xubuntu.org/] from **lp:xubuntu-website/docs-startpage**

Chapter 7. Release Cycle

This section loosely describes the stages of a development and release cycle.

Planning

Planning the release is conducted in three phases:

- Brainstorming
- Approving blueprints
- Finalizing specifications

During the brainstorming phase the contributors determine what they would like to work on during the release cycle, including proposals for changes in default applications. Anybody can add items to the list. Settling on the goals in advance will make planning, focusing and estimating the likelihood for the features to be implemented easier.

After the brainstorming is over and before or on the Feature Definition Freeze day, the Xubuntu team will approve or reject the proposed blueprints. Any items with no assignee or rationale will be automatically rejected, but having them will not guarantee that the blueprint is approved. Other criteria include, but are not limited to: likelihood of getting the feature implemented, maintenance weight in the future, possible stability issues, influence on other blueprints, et cetera. The approved blueprints compose the roadmap for the release being developed.

After blueprints are approved, they should be finalized and detailed specifications should be written. These specifications should document the proposed changes and help guide the implementation.

Once the detailed specifications are ready, the developer team should coordinate with the QA team about the required scale and schedule of testing. The QA team will then build a schedule for testing for ISO and package tests. Additional requests for testing during the cycle should be sent out only after consulting the QA team.

The Ubuntu freezes define the deadlines for implementation.

Developing

In addition to implementing the features and/or improvements set in the roadmap, there are a number of tasks which will occur during each release cycle:

- Xubuntu packages will be synced or merged with Debian as appropriate
- Bugs reported by users will be reviewed, fixed, and/or passed upstream
- Attempt to reduce our delta by pushing appropriate patches upstream
- Evaluate the seeds to ensure we are shipping the optimal software
- Work on improving and updating the Xubuntu documentation and artwork

Testing

Throughout the release cycle and even more so towards the end, we will test Xubuntu to ensure that Xubuntu is a quality product that we are proud of.

The most important goal of the testing is to find as many bugs as possible and report them with sufficient detail to the Ubuntu QA trackers. To read more about conducting tests and reporting them, refer to the QA tracker wiki page.

ISO testing

The Xubuntu team will do its best to ensure that released milestone ISOs have enough tests completed. We will also test daily builds, upgrading through supported release paths and running the development version regularly to help detect problems and make sure the changes made meet the release goals, work as expected and do not cause regressions.

The following schedule is used for daily testing:
Scheduled time: Beginning of cycle
Action: Send testing schedule via both the developer and users mailing lists as well as the Launchpad testing team.
Scheduled time: Monthly
Action: As required, send out reminders of the testing schedule.

The following schedule is used for each milestone:
Scheduled time: 7 days before release
Action: Pre-reminder for milestone testing
Scheduled time: 2 to 3 days before release
Action: Once appropriate testing area set up, call for testing and start milestone testing
Scheduled time: Release day
Action: Mark images ready on the ISO tracker once the release team is confident to do so

Package testing

When being used the QA team will schedule package testing sprints to happen during the cycle between milestone testing to ensure applications that are included in Xubuntu have sufficient amount of testing conducted.

Regardless of whether package testing is taking place during a development cycle, additional testing should be conducted for a new default application is to be included in the next Xubuntu release, when an existing application has a substantially large update in the middle of the cycle or when developers request specific testing for a specific update to an application.

Calls for package testing sprints will be sent out as required.

Releasing

When it comes time to deploy a release (both milestones and final) several conditions must be met:

- Appropriate testing has been done on the image
- There are no known bugs which cause data loss or damage to hardware
- The image must not be oversized
- Xubuntu must be of sufficient quality that it would not damage Xubuntu's, Ubuntu's, or any of the other flavors' reputation/image.

When a release is made, the Xubuntu Release Team must follow the release process specified below. The release process ensures that the new release has sufficient release notes and release announcement and that all release-specific communication is updated to inform about the new release. Where needed, advice from the Ubuntu release team should be asked for.
Scheduled time: User Interface Freeze
Action: Upload artwork and slideshow packages
Scheduled time: Documentation String Freeze
Action: Update translation templatesUpload documentation packages
Scheduled time: Non-Language Pack Translation Deadline
Action: Upload documentation and slideshow packages
Scheduled time: 14 days before release
Action: Reminder to update the website FAQ (final release only)
Scheduled time: 7 days before release
Action: Start preparing release notes and release announcementMake the tracker start tracking the next release
Scheduled time: Release day
Action: Publish release announcement and notesUpdate the website, IRC channel topics and social media outlets with new release information
Scheduled time: After release (final release only)
Action: Set the new development version as the default in the trackerSet up appropriate branches (documentation, ...) in LaunchpadReview and update this page

Maintaining

After a release is made, bugs and problems are bound to be reported. The period between the release and the start of the next release cycle is the optimal time to perform Stable Release Updates (SRU) for major bugs as long as developers do not forget to fix the issue in the next release as well, once the repository opens. Once the next release cycle has started, primary focus will be on the next release and the severity/importance of the SRU candidates will be more important when it comes to deciding whether an SRU will be performed or not.

Appendix A. Strategy Document

Introduction

About this Document

This document will describe the strategy, vision and direction of the Xubuntu project as well as act as a guide and a reference for the Xubuntu Team.

This revised version has been written by Pasi Lallinaho with the help of Elizabeth Krumbach and Simon Steinbeiß along with numerous people in the community. This version is based on the former Xubuntu Strategy Document by Cody Somerville et al.

We wish thank to Cody Somerville, Jono Bacon, Eero Tamminen, Nico Veenkamp and Lionel Le Folgoc for their contributions to this document. Some content, ideas, and inspiration were derived from existing Ubuntu documentation.

Mission Statement

Xubuntu is a community-developed, Ubuntu-based operating system that combines elegance and ease of use. Xubuntu provides a light, stable and configurable desktop environment with conservative workflows. Xubuntu delivers a polished and unified product ready for end-users.

In addition to the technical aspects, the Xubuntu team focuses on contributor and user communities.

Target

The target audience for Xubuntu consists of users who are interested in having an elegant, easy to use, polished and unified operating system. Xubuntu is a good option for those who want a stable, configurable and/or relatively light desktop environment too. Finally, Xubuntu is an appealing choice for users who prefer conservative workflows over the newest innovations.

Xubuntu does not target users with specific skill sets or aptitudes. We want Xubuntu to be a viable option for users who are new to Linux or the Ubuntu platform, but also be appealing to more experienced users. Xubuntu does not specifically focus on new users or users migrating from Windows, but according to this ease of use, it is a good alternative for those users as well.

Xubuntu does not explicitly target users with low, modest, or high powered machines but instead targets the entire spectrum. Xubuntu's extra responsiveness and speed, among other positive traits, can be appreciated by all users, regardless of their hardware.

While Xubuntu uses Xfce, it is not specifically targeted to Xfce enthusiasts or projects and software being hosted by the Xfce project or associating (officially or unofficially) with Xfce are not guaranteed for inclusion in Xubuntu.

Areas of Focus

This part of the strategy document provides a transparent framework and a set of priorities for the whole Xubuntu development and its tasks, including but not limited to decision making processes, package selection and technical dispute resolution.

The areas of focus are presented in an abstract and simple form but are also concrete enough to be actionable. As the foundation of the Xubuntu vision, the specifications in the strategy document should not be confused with release-specific goals and priorities, but instead recognized as the principles which the release-specific goals are based on.

Focus 1: Usability

Usability is one of the most important parts of an operating system which is used on a daily basis. This is why Xubuntu should be easy to use and have an appearance that doesn't get in the way. The appearance should be an all-round experience, covering all user interfaces from booting to shutting down.

Xubuntu should be localized to allow users to work in their preferred language. Another important aspect of usability is configurability. We believe Xfce gives users the possibility to configure their system in a meaningful way; other applications should live up to this expectation at least moderately well.

While accessibility is not one of the main priorities of Xubuntu, it should be taken into serious consideration as long as it doesn't require unreasonable efforts to integrate, implement or maintain. At least common accessibility tools should be installed in the default Xubuntu installation.

Focus 2: Performance

The Xubuntu team should strive to make Xubuntu lightweight. This means every Xubuntu release should work moderately well on machines that date to a few years before the release date. This ultimately means that newer Xubuntu releases may not able to support older computers. However, it assures that the Xubuntu team is able to work on other improvements and provide a release that is able to fulfill the expectations for a modern operating system.

Users wanting the most lightweight system possible should be pointed at the minimal CD, more lightweight derivatives (such as Lubuntu) or other options.

At the start of the release cycle, the minimum system requirements should be re-evaluated to determine if they continue to be realistic.

Focus 3: Ready to Use

It is important for us to provide a polished and unified product that is ready for end-users. It is also a prerequisite to enabling Xubuntu as a useful, usable and effective desktop. Without the mentioned integration, the Xubuntu desktop will appear rough and unpolished which is unappealing to end-users.

The integration is accomplished also by selecting applications and libraries that work well with each other as well as by applying patches to assure a more bug-free system.

See the section called “Package Selection” for guidelines on how to determine if an application fits in the Xubuntu stack.

Focus 4: Community

The Xubuntu community is an important force in creating Xubuntu and making it as perfect as it can be. It's essential that regular coordination between contributors work well. The infrastructure to allow good communication between the contributors as well as users is described more closely in the section called “Community” and the section called “Development”.

Where possible and appropriate, the Xubuntu team should work together with other communities, including other Ubuntu teams and upstream. If cooperation at a given time is not possible but would be beneficial for both parties, the Xubuntu team should try to allocate some resources to the cooperation at a later time.

Community

Xubuntu Council

The Xubuntu Council (later: council) is responsible for overseeing the development direction of Xubuntu and enabling the community to work with Xubuntu as smoothly as possible. The council serves a term of four releases/two years at a time, ending with an LTS release to allow long-term planning.

The xubuntu-council Launchpad team consists of the council members. The xubuntu-council team is the administrator for all other Xubuntu Launchpad teams, giving the council full administration access for the Xubuntu teams.

Membership

The council consists of 3 members, who are chosen by a CIVS [http://civs.cs.cornell.edu/] vote. All members in the Xubuntu Team [https://launchpad.net/~xubuntu-team] and/or direct members of any moderated subteams can be nominated, either by themselves or anybody else in the Xubuntu community with the nominees consent. The same group of people is eligible to vote.

If a council member becomes inactive for 6 months, the member shall be replaced by a new council member for the remainder of their term, elected by a new CIVS vote.

The council will decide on a chair for the whole council term. The chair will act as the official point of contact for Xubuntu.

Council operation

The council is expected to take action or respond to any issue raised within 2 weeks. If appropriate and fair, the first action can be postponement.

If at any time a majority of the Xubuntu team considers the actions of the council harmful for the project, the issue at hand should be raised to the council for reconsideration. If a resolution can't be found, the council should consult the the Ubuntu Community Council and other appropriate Ubuntu community teams. Similarly, if the council can't reach a consensus even after consulting the Xubuntu team, they should consult the Ubuntu Community Council and other appropriate Ubuntu community teams.

Xubuntu Team

The Xubuntu team is a group of individuals who are primarily responsible for the Xubuntu development.

The Xubuntu team is organized in Launchpad as xubuntu-team [https://launchpad.net/~xubuntu-team]. The Launchpad team is moderated.

The individuals wanting to join the Xubuntu team are required to meet the criteria described below prior to applying or they will be automatically declined. Leaders for subteams can be appointed from this team. The Xubuntu team is moderated by the council.

In case a team lead is in the minority when voting on an issue that concerns their area of expertise, the team lead should be given another hearing, after which the team should take extra care in deciding about the issue and may want to consider voting on it again after more discussion. If required, the issue can be raised to the council for further consideration.

Becoming and staying a member

To be accepted to the Xubuntu team, applicants participants must demonstrate their motivation and ability to contribute to Xubuntu. This is to ensure that any Xubuntu team member has a sufficient understanding of the Xubuntu community and its operation. The different steps one must go through will also indicate that the candidate member is able to work within the guidelines, and more important, with other people and communities. To become a member, one must:

- Commit meaningful contributions to one of the subteams, after which one can be approved to the subteam for "probation" by a subteam administrator
- Demonstrate motivation to contribute perpetually, after which one can be approved to the Xubuntu team

To stay a member of the Xubuntu team, the member will have to have desire to continue contributing in the future. Anyone with no contributions for more than a complete cycle (6 months) should deactivate themself from the team as well as all subteams and reapply if/when they are willing to rejoin. The team administrators should purge people with no contributions in more than 6 months from the teams occasionally.

Xubuntu Users

The Launchpad team xubuntu-users [https://launchpad.net/~xubuntu-users] is the formal top-level group of people who are interested in Xubuntu. The team is open for anybody to join.

The members of the official Launchpad group are encouraged to give user support on IRC and on the Xubuntu users mailing list as well as spread the word about Xubuntu anywhere they see fit, including but not limited to blogs, social media and conferences.

In addition to user support and advocacy, the members are free to organize as they like on social media platforms and more. If the group is considered substantial and useful enough, the Xubuntu team can name it official and help with maintaining it. All currently official teams and their administrators are listed on the Xubuntu wiki under the Website Administrators [http://wiki.xubuntu.org/website/admins] page.

Development

Coordination

The direction of the development of the project is coordinated by:

- The Strategy Document
- Release-specific blueprints and specifications
- The community meetings
- The developer communication tools listed below

Communication

For Xubuntu to be successful, its members and community must have the right tools to enable useful and helpful communication and ultimately, to help the community grow. These communication tools include the following "core" tools:

- *Mailing lists*; Xubuntu-users for community support and user discussions and the Xubuntu-devel list for development discussion and coordination
- *IRC channels*; #xubuntu for community support, #xubuntu-devel for development discussion and coordination and #xubuntu-offtopic for general discussion with a more relaxed mood
- *Launchpad* for team organization and granting permissions and access rights via team memberships
- *The Xubuntu website* for news, development articles and general information about Xubuntu
- *The Xubuntu developer area* for tracking development progress and collaborating on development issues and sharing development related information on the wiki

In addition to communicating with each other and the Xubuntu community, the Xubuntu team should communicate continuosly with other Ubuntu contributors, upstream and other projects related to Xubuntu. The team members should take part in the development discussion outside Xubuntu as well to create and maintain healthy relationships with other projects and their developers.

Dispute Resolution

The Code Of Conduct (CoC) is one of the most fundamental documents in the Ubuntu community, including Xubuntu, and it functions as the base for a working community. Everybody and every communicating medium in the Xubuntu community is CoC-compliant at all times.

> **Note**
>
> To read the most recent revision of the Ubuntu Code of Conduct, refer to the Ubuntu Code of Conduct [http://www.ubuntu.com/about/about-ubuntu/conduct] page on the Ubuntu website.

The Ubuntu Community Council can be asked to act as mediators/advisors where their expertise would be useful when the Xubuntu Council requires assistance. The Xubuntu Council and the Xubuntu team are encouraged to talk to other teams in the Ubuntu community if it helps resolving the issue.

When disputes occur, contributors are strongly encouraged to refrain from taking the disputed actions to avoid sabotaging the dispute resolution process.

Release and Support Cycles

The Xubuntu release cycle consists of 6 months from start to release, then an additional 9 month of maintenance for standard releases and 3 years of support for LTS releases. Although the Xubuntu project itself can not provide commercial support or commercial guarantees, the Xubuntu team will make every effort to conform with the acknowledged support period and provide security updates for Xubuntu packages.

For more information about the specifics of the processes related to developing a release, see Chapter 6, *Processes*.

Seeds and Composition

Core Components

The following packages and their technical dependencies are considered essential in composing an operating system that represents the goals set previously in this document:

- xfwm4 (Xfce4 window manager)
- xfdesktop4 (Xfce4 desktop)
- xfce4-panel (Xfce4 panel)
- xfce4-session (Xfce4 session manager)
- xfconf (Xfce4 configuration storage system)
- xfce4-settings (Xfce4 settings manager)
- thunar (file manager)

Unless the components mentioned above are deprecated and no longer work as they are intended, they should not be substituted or removed from the seed.

The **xubuntu-core** metapackage consists of the core packages as well as other important packages which contribute to the Xubuntu experience. The **xubuntu-desktop** metapackage depends on the core metapackage and includes additional packages to provide the full, ready to use, Xubuntu experience.

Package Selection

When considering the package selection for Xubuntu, there are two main guidelines:

- The package must be in line with the mission statement
- The package selection process must be conducted as objectively as possible

While the Xubuntu Technical Lead has the last word on package selection with the approval of the Xubuntu Council, it's recommended that the whole team is involved in the discussion, bringing up well thought out arguments both for and against. When appropriate, tests should be conducted to gather meaningful measurements.

Before seeding an application in Xubuntu, the following aspects should be considered:

- Usability and integration
- Usefulness
- Resource consumption and size
- Reliability
- Localization

When measuring if a package is right for Xubuntu, the decision should not be based on an analysis of the package alone, but the target package along with its dependencies. Any package that is considered for seeding should fulfill the following requirements:

- The package should use the GTK toolkit
- The package should not pull in heavy libraries, especially if they will run and/or start frequently
- The package should be well maintained upstream

The above guidelines can be omitted with the approval of the council when there are extensive benefits to seeding the package.

Appendix B. Common Reference

Using Bazaar (Bzr)

Bazaar (commonly referred to as Bzr) is the version control system primarily used in Xubuntu development.

Setting up Bzr

To set up Bzr, do the following:

- To make sure Bzr is installed on your system, run **`sudo apt-get install bzr`**
- To set up your personal information for Bzr to use, run **`bzr whoami "John Doe <john.doe@gmail.com>"`**

Getting and updating Bzr branches

After you have set up Bzr, you can get a local copy of a branch by running **`bzr branch LOCATION`**. This will create a directory for the branch.

To make sure your local copy is up to date, run **`bzr pull`** inside the branch directory.

Pushing to Bzr branches

When you have worked on your changes and want to push them to a remote branch, do the following in the local branch:

- To rename or move files, run **`bzr mv OLD NEW`**. This helps to keep the resulting diff meaningful, and also works after the fact.
- If you have added files, run **`bzr add .`** in the branch root. This starts tracking any files in the branch that are currently unknown.
- If you have removed files, run **`bzr remove`**. In most cases, this should automatically stop tracking files that are no longer in the branch.
- To check the status of the branch, run **`bzr status`**.
- Commit your changes by running **`bzr commit`**. This will open a text editor to enter the commit message.
- After you have done one or more commits, run **`bzr push LOCATION`** to push your changes to a remote branch.

The commit message should summarize all the changes you have done after the last commit. You can specify it directly by using **-m MESSAGE**, and link a bug report to the commit by using **--fixes lp:BUGNO**.

> **Note**
>
> When you are working on a branch and are making several changes that aren't related to each other, it is recommended that you do several commits. This helps to keep the commit log clean and makes it easier to find any issues that certain changes may have introduced.

> **Note**
>
> When you push to a remote branch the first time, you will need to specify the branch location. In most cases, it will follow the format **lp:~username/projectname/branchname**. This pushes your code to a personal remote branch. On subsequent pushes, you can omit the branch location, as Bzr will have saved your previous push location in the local branch configuration.

Submitting merge proposals

Once you have pushed your changes to a personal branch, you might want to propose them to be merged into the main branch.

To do a merge proposal (often referred to as MP) via web browser, start by opening the Launchpad page of your branch. You can find it by going to your Launchpad branches [https://code.launchpad.net/people/+me] page and clicking on the branch in question. Now, click on *Propose for merging*. This will open a page that asks for some details on the merge proposal you are about to do.

The *Target Branch* is usually set correctly by default, change this only if you know it is wrong. If you have been told to add a specific *Reviewer*, add that reviewer to the appropriate field. Enter a *Description* of your proposed change as well. If in doubt about any of the options, ask the requested reviewer, or if one isn't specified, in the IRC channel.

Finally, click on **Propose Merge**. This submits your merge proposal and the target branch's owner as well as the specified reviewers will be sent a mail. If they need further information or want you to work further on the branch, they will comment on the merge proposal. You will be sent a mail any time an action is taken on the merge proposal, including when it's approved and merged into the main branch.

To do a merge proposal from the command line, using both the remembered submit branch (defaults to the parent branch) and the default reviewer team for that branch, run **`bzr lp-propose-merge`**. This will open a text editor to enter the description of your proposed change. You can override the default reviewer by using **--review NAME** one or more times, and link a bug report to the merge proposal by using **--fixes lp:BUGNO**.

Using Git

Git is the version control system used in Xfce development.

> **Note**
>
> One major difference between Git and Bazaar to note is that, in Bazaar you usually deal only with branches, while in Git you have to differentiate between branches and repositories, which can hold multiple branches. This is also reflected by the various commands you use. Another is that Git doesn't automatically stage your changes for commit, you have to tell it to do so.

Setting up Git

To set up Git, do the following:

- To make sure Git is installed on your system, run:

  ```
  sudo apt-get install git
  ```

- To set up your personal information for Git to use, run:

  ```
  git config --global user.name "John Doe"
  ```

  ```
  git config --global user.email "john.doe@gmail.com"
  ```

Getting and updating Git repositories

After you have set up Git, you can get a local copy of a repository by running **`git clone LOCATION`**. This will create a directory for the repository.

To make sure your local copy of a branch is up to date, run **`git pull`** inside the branch directory.

Pushing to Git repositories

When you have worked on your changes and want to push them to a remote repository, do the following in the local branch:

- To rename or move files, run **`git mv OLD NEW`**. This helps to keep the resulting diff meaningful, while Git should figure this out automatically.
- To remove files, run **`git rm FILE`**. This also stops tracking the specified files, while it's redundant if all changes are staged later anyway.
- If you have added files, run **`git add .`** in the branch root. This starts tracking any files in the branch that aren't currently, and also stages any changes.
- To check the status of the branch, run **`git status`**.
- Commit your changes by running **`git commit -a`**. This stages any changes, and will open a text editor to enter the commit message.
- After you have done one or more commits, run **`git push LOCATION BRANCH`** to push your changes to a remote repository.

The commit message should summarize all the changes you have done after the last commit, you can specify it directly by using **-m MESSAGE**.

Build dependencies

Before you can build a package from source, you will usually need to install its build dependencies.

To do this, simply run **`sudo apt-get build-dep PACKAGE`**.

Launchpad

Launchpad is the primary platform used for Xubuntu development. Historically, the version control system used on Launchpad is Bazaar, but it recently added support for Git [https://help.launchpad.net/Code/Git], too.

Setting up a Launchpad account

For information on setting up a Launchpad account, see Creating a new account [https://help.launchpad.net/YourAccount/NewAccount].

Useful QA Links

- Xubuntu QA [https://launchpad.net/~xubuntu-qa]
- Xubuntu Testers [https://launchpad.net/~xubuntu-testers]
- ISO Tracker [http://iso.qa.ubuntu.com/qatracker]
- Package Tracker [http://packages.qa.ubuntu.com/]
- Ubuntu QA Team Wiki [https://wiki.ubuntu.com/QATeam]
- Ubuntu Manual Tests [https://launchpad.net/ubuntu-manual-tests/]

Appendix C. Creative Commons by Attribution-ShareAlike 2.5

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. **Definitions.**

   a. **"Collective Work"** means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

   b. **"Derivative Work"** means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

   c. **"Licensor"** means the individual or entity that offers the Work under the terms of this License.

   d. **"Original Author"** means the individual or entity who created the Work.

   e. **"Work"** means the copyrightable work of authorship offered under the terms of this License.

   f. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

g. **"License Elements"** means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.

2. **Fair Use Rights.** Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. **License Grant.** Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;

b. to create and reproduce Derivative Works;

c. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

d. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

e. For the avoidance of doubt, where the work is a musical composition:

i. **"Performance Royalties Under Blanket Licenses."** Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

ii. **"Mechanical Rights and Statutory Royalties."** Licensor waives the exclusive right to collect, whether individually or via a music rights society or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

f. **"Webcasting Rights and Statutory Royalties."** For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. **Restrictions.** The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(c), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(c), as requested.

b. You may distribute, publicly display, publicly perform, or publicly digitally perform a Derivative Work only under the terms of this License, a later version of this License with the same License Elements as this License, or a Creative Commons iCommons license that contains the same License Elements as this License (e.g. Attribution-ShareAlike 2.5 Japan). You must include a copy of, or the Uniform Resource Identifier for, this License or other license specified in the previous sentence with every copy or phonorecord of each Derivative Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Derivative Works that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder, and You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Derivative Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Derivative Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Derivative Work itself to be made subject to the terms of this License.

c. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. **Representations, Warranties and Disclaimer**

UNLESS OTHERWISE AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE MATERIALS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. **Limitation on Liability.** EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. **Termination**

    a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

    b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. **Miscellaneous**

    a. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

    b. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

    c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

    d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

    e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, neither party will use the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time.

Creative Commons may be contacted at http://creativecommons.org/.