Open Source Sorcerers

Professor Nick Murphy

User Interaction Engineering – SWE 4783

Kennesaw State University – Spring 2025

Kiwix – Group Research

Kiwix is all about education. The target audience for users using Kiwix software can be classified as people who are seeking education. To go into further detail, Kiwix is aimed to assist learners who are in remote areas. In remote areas, there may be limited access or no access to high quality internet. This can make learning on the internet a challenge that Kiwix aims to solve. Along with users in remote areas, additional audiences for Kiwix can be prisoners, students, and educators of remote areas. They do this primarily by deeply compressing websites and articles that can fit into a single .zim file. These zim files can be stored on mobile phones and read without an internet connection.

There is one common workflow in Kiwix. This workflow can be utilized for a multitude of different purposes. It can be used by individual learners to learn on their own, or it can be used by educators who are downloading a lot of content to be used in a lesson for their class. The common workflow is for users to browse which content, article, or website they would like to compress into the .zim file, then download the content onto their devices for when they are offline and without an internet connection. All in all, browsing which content you wish to download onto your device is how many users spend most of their time using the software.

Our proposed redesign of the software aims to help improve browsing for content on the software. We noticed that there is not an “Undo Closed Tab” option on the software. We believe that implementing this change will greatly increase user satisfaction and minimize frustration. With these effects in place, user engagement and efficiency will improve.

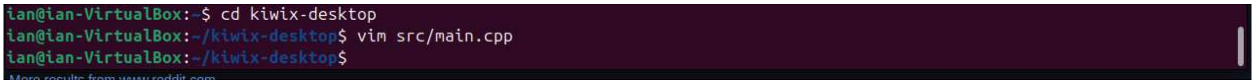
There are many different applications used with Kiwix that users utilize frequently. Kiwix users can practically download content from any website in the world. Most of the content downloaded from Kiwix is educational content from numerous different websites around the globe, so websites are other applications that Kiwix would be utilizing alongside Kiwix. There are also numerous other offline readers that allow users to view content without an internet connection, and we hope that once we implement our proposed redesign, we will help make Kiwix a better application and system for all to use and enjoy.

We noticed that Kiwix users' feedback for the software and the user flow was generally positive with little complaints. However, a problem that Kiwix has recently been dealing with is the issue with server problems. Over the last two months, they have had problems with their server provider dropping them, and with the struggles of dealing with them or potentially finding a new server provider, their users have expressed their concerns with the reliability of the software. Despite this problem being the main issue for Kiwix currently and not the user flow and interaction, we still plan on implementing the "Undo closed tab" feature. We plan on making a great user interaction into an even better one.

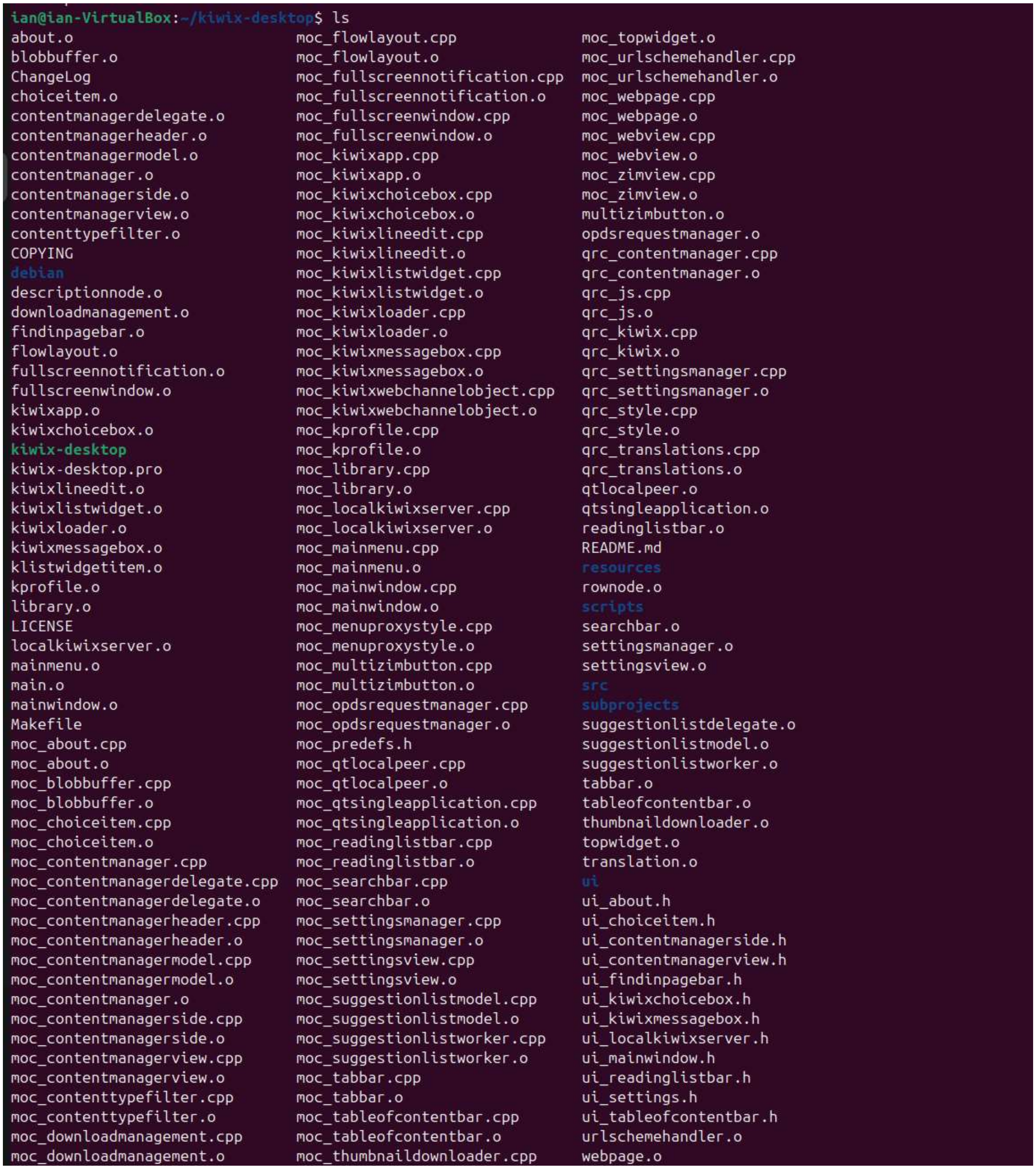
The main metric we aim to improve with this software is efficiency. While browsing, the users may accidentally close out an entire tab or website that they were using and learning on. In this scenario, the user may become frustrated because they must personally remember the content they were viewing and access it again. We will measure how much efficiency has improved in the software with our change by timing how long it takes to bring back all of the contents that were on a tab that closed versus the time it takes to press a few buttons to fully retrieve the content they were previously viewing before they closed it.

The next page will include proof of our team being able to build source code for Kiwix and make changes. We are excited to start working on our proposed implementation, and we hope to make a difference for the users of Kiwix. Thank you very much for reading.

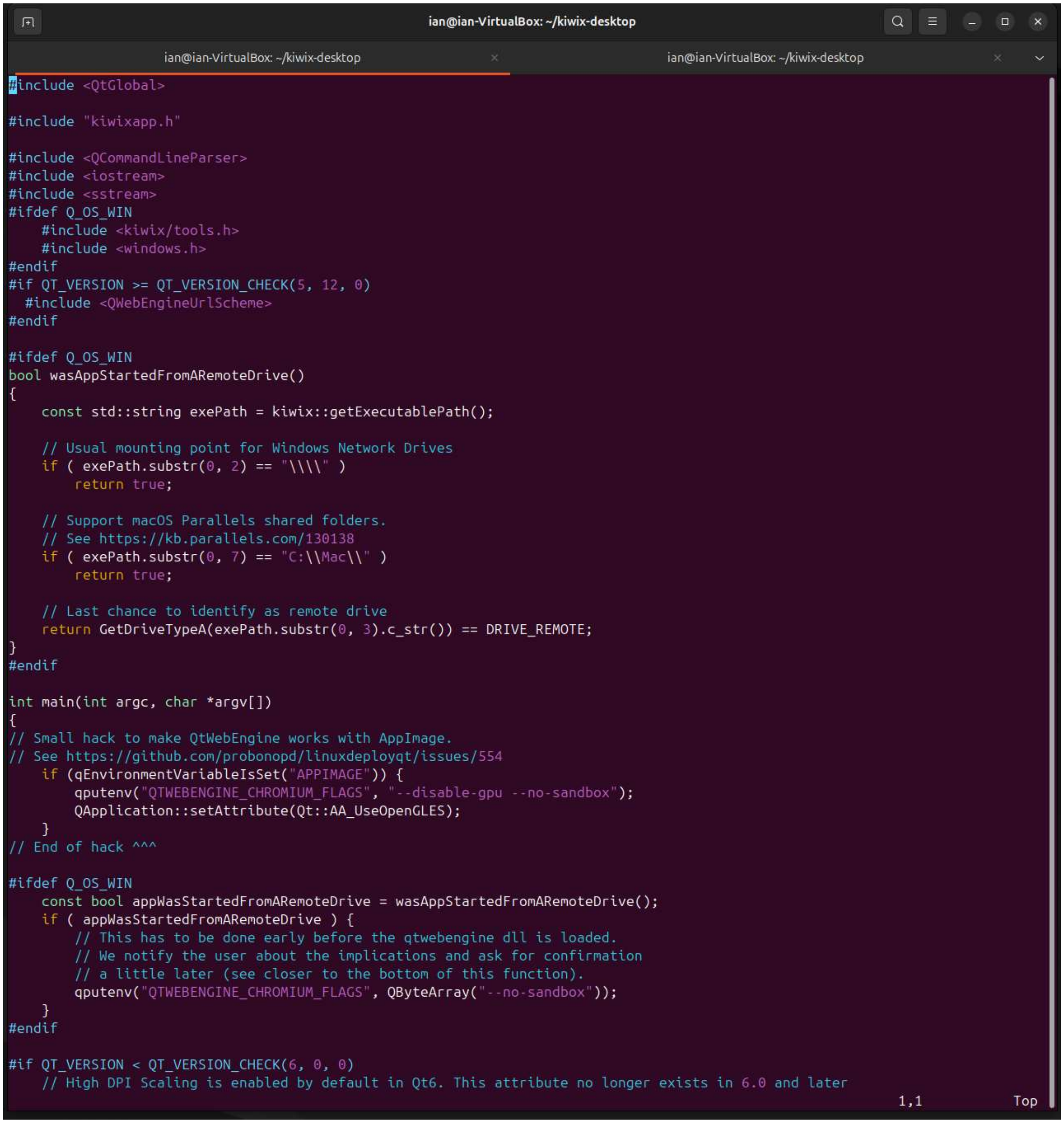
Proof of Building Kiwix in the Form of Screen Captures



```
ian@ian-VirtualBox:~$ cd kiwix-desktop
ian@ian-VirtualBox:~/kiwix-desktop$ vim src/main.cpp
ian@ian-VirtualBox:~/kiwix-desktop$
```



```
ian@ian-VirtualBox:~/kiwix-desktop$ ls
about.o                         moc_flowlayout.cpp              moc_topwidget.o
blobbuffer.o                    moc_flowlayout.o                moc_urlschemehandler.cpp
ChangeLog                       moc_fullscreennotification.cpp  moc_urlschemehandler.o
choiceitem.o                    moc_fullscreennotification.o    moc_webpage.cpp
contentmanagerdelegate.o        moc_fullscreenwindow.cpp        moc_webpage.o
contentmanagerheader.o          moc_fullscreenwindow.o          moc_webview.cpp
contentmanagermodel.o           moc_kiwixapp.cpp                moc_webview.o
contentmanager.o                moc_kiwixapp.o                  moc_zimview.cpp
contentmanagerside.o            moc_kiwixchoicebox.cpp          moc_zimview.o
contentmanagerview.o            moc_kiwixchoicebox.o            multizimbutton.o
contenttypefilter.o             moc_kiwixlineedit.cpp           opdsrequestmanager.o
COPYING                         moc_kiwixlineedit.o             qrc_contentmanager.cpp
debian                          moc_kiwixlistwidget.cpp         qrc_contentmanager.o
descriptionnode.o               moc_kiwixlistwidget.o           qrc_js.cpp
downloadmanagement.o            moc_kiwixloader.cpp             qrc_js.o
findinpagebar.o                 moc_kiwixloader.o               qrc_kiwix.cpp
flowlayout.o                    moc_kiwixmessagebox.cpp         qrc_kiwix.o
fullscreennotification.o        moc_kiwixmessagebox.o           qrc_settingsmanager.cpp
fullscreenwindow.o              moc_kiwixwebchannelobject.cpp   qrc_settingsmanager.o
kiwixapp.o                      moc_kiwixwebchannelobject.o     qrc_style.cpp
kiwixchoicebox.o                moc_kprofile.cpp                qrc_style.o
kiwix-desktop                   moc_kprofile.o                  qrc_translations.cpp
kiwix-desktop.pro               moc_library.cpp                 qrc_translations.o
kiwixlineedit.o                 moc_library.o                   qtlocalpeer.o
kiwixlistwidget.o               moc_localkiwixserver.cpp        qtsingleapplication.o
kiwixloader.o                   moc_localkiwixserver.o          readinglistbar.o
kiwixmessagebox.o               moc_mainmenu.cpp                README.md
klistwidgetitem.o               moc_mainmenu.o                  resources
kprofile.o                      moc_mainwindow.cpp              rownode.o
library.o                       moc_mainwindow.o                scripts
LICENSE                         moc_menuproxystyle.cpp          searchbar.o
localkiwixserver.o              moc_menuproxystyle.o            settingsmanager.o
mainmenu.o                      moc_multizimbutton.cpp          settingsview.o
main.o                          moc_multizimbutton.o            src
mainwindow.o                    moc_opdsrequestmanager.cpp      subprojects
Makefile                        moc_opdsrequestmanager.o        suggestionlistdelegate.o
moc_about.cpp                   moc_predefs.h                   suggestionlistmodel.o
moc_about.o                     moc_qtlocalpeer.cpp             suggestionlistworker.o
moc_blobbuffer.cpp              moc_qtlocalpeer.o               tabbar.o
moc_blobbuffer.o                moc_qtsingleapplication.cpp     tableofcontentbar.o
moc_choiceitem.cpp              moc_qtsingleapplication.o       thumbnaildownloader.o
moc_choiceitem.o                moc_readinglistbar.cpp          topwidget.o
moc_contentmanager.cpp          moc_readinglistbar.o            translation.o
moc_contentmanagerdelegate.cpp  moc_searchbar.cpp               ui
moc_contentmanagerdelegate.o    moc_searchbar.o                 ui_about.h
moc_contentmanagerheader.cpp    moc_settingsmanager.cpp         ui_choiceitem.h
moc_contentmanagerheader.o      moc_settingsmanager.o           ui_contentmanagerside.h
moc_contentmanagermodel.cpp     moc_settingsview.cpp            ui_contentmanagerview.h
moc_contentmanagermodel.o       moc_settingsview.o              ui_findinpagebar.h
moc_contentmanager.o            moc_suggestionlistmodel.cpp     ui_kiwixchoicebox.h
moc_contentmanagerside.cpp      moc_suggestionlistmodel.o       ui_kiwixmessagebox.h
moc_contentmanagerside.o        moc_suggestionlistworker.cpp    ui_localkiwixserver.h
moc_contentmanagerview.cpp      moc_suggestionlistworker.o      ui_mainwindow.h
moc_contentmanagerview.o        moc_tabbar.cpp                  ui_readinglistbar.h
moc_contenttypefilter.cpp       moc_tabbar.o                    ui_settings.h
moc_contenttypefilter.o         moc_tableofcontentbar.cpp       ui_tableofcontentbar.h
moc_downloadmanagement.cpp      moc_tableofcontentbar.o         urlschemehandler.o
moc_downloadmanagement.o        moc_thumbnaildownloader.cpp     webpage.o
```


ian@ian-VirtualBox: ~/kiwix-desktop
ian@ian-VirtualBox: ~/kiwix-desktop
ian@ian-VirtualBox: ~/kiwix-desktop

```
#include <QtGlobal>

#include "kiwixapp.h"

#include <QCommandLineParser>
#include <iostream>
#include <sstream>
#ifdef Q_OS_WIN
    #include <kiwix/tools.h>
    #include <windows.h>
#endif
#if QT_VERSION >= QT_VERSION_CHECK(5, 12, 0)
  #include <QWebEngineUrlScheme>
#endif

#ifdef Q_OS_WIN
bool wasAppStartedFromARemoteDrive()
{
    const std::string exePath = kiwix::getExecutablePath();

    // Usual mounting point for Windows Network Drives
    if ( exePath.substr(0, 2) == "\\\\" )
        return true;

    // Support macOS Parallels shared folders.
    // See https://kb.parallels.com/130138
    if ( exePath.substr(0, 7) == "C:\\Mac\\" )
        return true;

    // Last chance to identify as remote drive
    return GetDriveTypeA(exePath.substr(0, 3).c_str()) == DRIVE_REMOTE;
}
#endif

int main(int argc, char *argv[])
{
// Small hack to make QtWebEngine works with AppImage.
// See https://github.com/probonopd/linuxdeployqt/issues/554
    if (qEnvironmentVariableIsSet("APPIMAGE")) {
        qputenv("QTWEBENGINE_CHROMIUM_FLAGS", "--disable-gpu --no-sandbox");
        QApplication::setAttribute(Qt::AA_UseOpenGLES);
    }
// End of hack ^^^

#ifdef Q_OS_WIN
    const bool appWasStartedFromARemoteDrive = wasAppStartedFromARemoteDrive();
    if ( appWasStartedFromARemoteDrive ) {
        // This has to be done early before the qtwebengine dll is loaded.
        // We notify the user about the implications and ask for confirmation
        // a little later (see closer to the bottom of this function).
        qputenv("QTWEBENGINE_CHROMIUM_FLAGS", QByteArray("--no-sandbox"));
    }
#endif

#if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
    // High DPI Scaling is enabled by default in Qt6. This attribute no longer exists in 6.0 and later
                                                                                 1,1           Top
```

Resources

Pearson, Ben. “The Cornerstone of Software Quality and Efficiency | Cycle Time.” *LinearB*. 4 Dec. 2024 https://linearb.io/blog/the-cornerstone-of-software-quality-and-efficiency-cycle-time

Dogra, Tory. “What Are Software Metrics? How Can I Measure These Metrics?” *Lansa*. 18 Dec. 2024 https://lansa.com/blog/app-development/what-are-software-metrics-how-can-i-measure-these-metrics/

https://kiwix.org/en/about-us/#our-story

https://www.reddit.com/r/opensource/comments/1ax3ljj/kiwix_will_be_part_of_this_years_summer_of_code/

https://www.reddit.com/r/Kiwix/comments/1ha5ioa/about_last_weeks_outage/

https://www.reddit.com/r/Kiwix/comments/1ho8a35/global_search_ui/