

KDE Connect for Windows port

GSoC 2019 proposal for KDE by Piyush Aggarwal

PRIMARY OBJECTIVE

To develop and deliver a release-able installer for KDE Connect for Windows, to be published on the Windows Store.

INTRODUCTION

Thanks a lot to the KDE organization for taking the [KDE on Windows Initiative](#), which is an ongoing project to port the KDE applications to Windows.

There is still a large portion of the user community unable to use KDE Connect because of two main reasons:-

OS Platform Statistics

2019	Win10	Win8	Win7	Vista	WinXP	Linux	Mac	Chrome OS	Mobile
February	50.7%	5.2%	18.0%	0.1%	0.3%	5.9%	10.3%	0.4%	9.4%
January	49.9%	5.4%	18.8%	0.1%	0.3%	6.1%	10.1%	0.3%	9.2%

1. The Windows Operating system is installed on more than 55% of the world's consumer-targeted computers. [\[source\]](#)

While the KDE principles for promoting use of Linux stands proud, KDE softwares ported to Windows are usually followed by good recognition for the software itself, and even more for KDE as the parent organization of the project.

2. There is minimal presence of KDE Connect on the web.
Every well - known KDE project is supplemented with a website of its own, where various links for documentation, support and download links for the latest build are available to the users in one place. A similar one-stop site for all information about KDE Connect has been proposed as an optional milestone, if time allows.
The current mockup is live at: (<https://brute4s99.gitlab.io>).

SOLUTION OVERVIEW

Thanks to the cross-platform nature of Qt, the User Interface of KDE Connect is rendered nicely on Windows systems, with the full set of icons and correct placing of different elements.

However, the real problems of using KDE Connect on Windows systems arise when there is a need to interface with the Operating System, for eg: rendering notifications, controlling media players, volume control and sharing files with other devices. The fixes to the respective plugins by using Windows APIs are proposed.

Apart from the plugins, some add-ons which allow easier accessibility and better integration with the operating system are also proposed.

The roadmap to publish the built software to the Windows App Store is also proposed.

SOLUTION SPECIFICATIONS

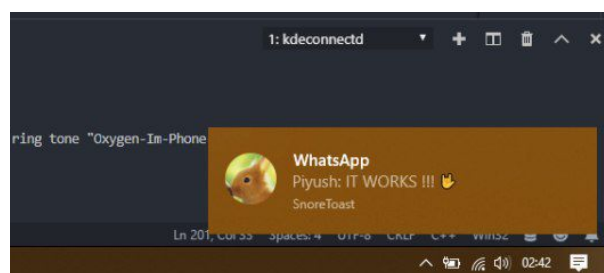
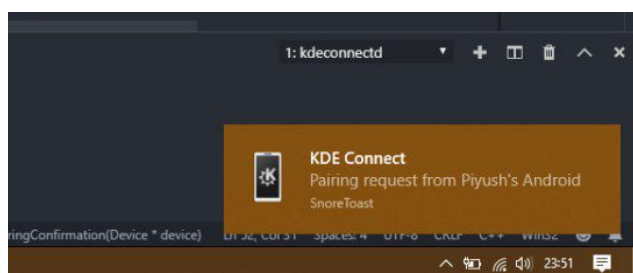
For the sake of simplicity, let's call the peer device(eg: an android phone) as "peer device" and the personal computer running the desktop version of KDE Connect as "the PC". For good documentation, code changes will also be supplemented with inline documentation wherever relevant.

1. Notifications Plugin

This plugin allows notifications received on the peer device to be forwarded to the PC.

A newly developed in-house open source library by [Hannah Von Reth](#), for rendering Native Windows 10 notifications, called LibSnoreToast can be employed for this purpose.

Proof of Concept: An executable, named SnoreToast, built using the same codebase as the (new) library was used in rendering these notifications:-



2. SFTP Plugin

The current SFTP Plugin needs to be paired with a stable backend for Windows. This should allow the Windows port access to file systems of peer device, just like the linux build. In practice, this would mean mounting the peer device's file-system, displaying entries for respective mounts in the native File Explorer of Windows OS. One of the candidate backends is [dokan-sshfs](#).

3. Other plugins

There are many other plugins that could use fixes to allow expected usage on Windows PCs.

a. Contacts

This plugin allows access to contacts of peer device to the PC by downloading the entire list of contacts or by requesting a specific contact.

In Windows systems, the VCards data received from the peer device can be sent to a folder (say, in %TEMP%) on the PC that contains all KDE Connect data specific to the particular peer device, wherefrom it may be picked up by any supporting application.

b. MPRIS Control

This plugin allows remote controlling media player controls and volume by the peer device.

The basic functionality (i.e pause, play, stop, next, previous) works, but music metadata is missing. This can be handled by using SystemMediaTransportControls class.

c. Pause Music

This plugin allows pausing current playing media in the event of an incoming call on the peer device.

VK_MEDIA_KEY can be used to fix the pause functionality on Windows.

d. Remote Keyboard

This plugin allows sending key-presses from PC to peer device.

Currently, the implementation does not support Windows, since it accepts inputs from within a Plasmoid.

This can be fixed by taking inspiration from the KCAApp's (an alternative mobile-friendly interface of KDE Connect targeted at mobile linux platforms) usage of the plugin.

e. Remote Commands

This plugin allows updating the list of predefined commands that can be triggered remotely with Run Commands plugins.

In its current state, the commands list is not updated on the peer device.

This can be fixed by minor modifications in the code.

f. Screensaver Inhibit

This plugin allows to inhibit the screensaver on the user device as long as the peer device is connected to the PC.

This can be fixed by using SystemParametersInfoA function.

g. Send Notifications

This plugin allows to get the desktop notifications received on the PC, to be forwarded to the peer device.

This can be fixed by using the UserNotificationListener class.

4. Add-ons for Windows build

There are many OS specific integration that can allow a better integration with the Windows Operating System and allow more accessibility to users.

a. Context Menu integration

An integration into the right click (context) menu of the Windows explorer can allow for easier file sharing across devices.

This can be done by creating shortcuts for each peer device(with device specific command arguments preloaded) and saving the said shortcuts into `shell:sendto` folder. The Windows OS automatically picks up the entries from the folder, and allows real time deletion and addition of new device entries.

b. Auto - start on Windows

KDE Connect starts automatically on Linux. It would be great to have the same feature for Windows, so that people can start using KDE Connect as soon as they log in the Windows system. This saves the user from the trouble of starting KDE Connect every time the system boots.

This can be done easily by placing a shortcut to `kdeconnect-indicator.exe` in `shell:startup` folder. This automatically registers the application to be started automatically when the user logs in the PC. This behaviour can be scripted through the NSIS installer scripts.

5. Tests

KDE Connect uses a series of tests on the continuous integration systems to keep a check on software quality. The LanLinkProvider test currently fails on Windows build. New code will be supplemented with tests as and when new behaviour and features are added into the port.

6. Integrate LibSnoreToast into KNotifications

KNotifications is an abstraction for system notifications. It provides an easy to use library to render multi - purpose system - level notifications.

Currently, there is minimal to no support for KNotifications on Windows. We can integrate LibSnoreToast to extend KNotifications for the benefit of all projects being built for Windows systems. This also allows much better support to future projects being ported to Windows OS. This can be done by creating a plugin for KNotifications. The KNotifications team has already been contacted about extending KNotifications with LibSnoreToast, and we have their support.

7. App Publishing to Windows Store

The Windows Store accepts UWP (Universal Windows Package), which are typically packaged with the .appx extension.

UWPs are typically written using C#, which allows their deployment as universal apps that run on Windows 10, Windows 10 Mobile, Xbox One and HoloLens without the need to be re-written for each.

The current KDE Connect Windows build is packaged using NSIS. NSIS (Nullsoft Scriptable Install System) is a tool that allows programmers to create installers for Windows.

Since we plan to support desktop Windows OS, we can simply submit a (.appx) conversion of our current Windows build. For this purpose, the application will be passed through a **Desktop App Converter**. After the necessary paperwork, we will be able to release the new Windows build for desktop Windows 10 on the Windows App Store.

PROPOSED TIMELINE AND OBJECTIVES

May 7, 2019 - May 27, 2019

Community Bonding

- Interaction with mentors and team, discussing ideas regarding various possibilities for features to be added to the project.
- Setting up the development environment, and other essentials.
- Probing more into Windows APIs in order to understand the development architectural details.
- Understanding how packaging and deployment work for Windows Store..

May 27, 2019 : Coding Period Begins

May 27, 2019 - June 3, 2019

First Milestone

- Start developing LibShoreToast extension for KNotifications
- At the end of this week, LibShoreToast extension for KNotifications will be in testing.

June 3, 2019 - June 10, 2019

Second Milestone

- Finish developing LibShoreToast extension for KNotifications
- At the end of this week, LibShoreToast extension for KNotifications will be under review by KNotifications team.

June 10, 2019 - June 17, 2019

Third Milestone

- Start fixing the SFTP plugin to work with Windows 10
- At the end of this week, the Windows port will be able to share files and browse android device filesystems natively with use of a suitable backend.

June 17, 2019 - June 24, 2019

Fourth Milestone

- Finish fixing the SFTP plugin to work with Windows 10
- At the end of this week, the Windows port will be able to share files and browse android device filesystems natively with use of a suitable backend.

June 24 to June 28, 2019 : First Evaluations

June 28, 2019 - July 8, 2019

Fifth Milestone

- Start fixing other KDE Connect plugins
- At the end of this week, more than half of the "Other plugins" will be fixed according to the specifications shared.

July 8, 2019 - July 15, 2019

Sixth Milestone

- Finish fixing other KDE Connect plugins
- At the end of this week, every KDE Connect plugin will be ported to work with Windows 10 natively, as specified.

July 15, 2019 - July 22, 2019

Seventh Milestone

- Developing add-ons for Windows build
- At the end of this week, the Windows port will also support OS-level integrations for better UX.

July 22 to July 26, 2019 : Second Evaluations

July 26, 2019 - August 5, 2019

Eighth Milestone

- Publishing app to Windows Store
- At the end of this week, A draft for KDE Connect for Windows will be submitted to Windows Store for review

August 5, 2019 - August 12, 2019

Ninth Milestone

- Buffer week to implement any new features/ to cover optional milestones.

August 12, 2019 - August 19, 2019

Tenth Milestone

- Buffer week to implement any new features/ to cover optional milestones.

August 19 to August 26, 2019 : Code submission and Evaluations

TOOLS TO BE USED

- VCS - Git
- Programming Language - C++14
- Build System - Craft
- Compiler - MSVC 2017
- GUI Toolkit - Qt 5.12
- Code Editors - Nano and VS Code
- Communication Medium - Jitsi, Email, IRC, Telegram, Matrix
- Development Environment - Windows 10 Pro
- Testing Environment - Windows 10
- Project Management - Phabricator, Gitlab CE instance for KDE

DELIVERABLES

- KDE Connect for Windows published as a Windows App to the Windows Store

ABOUT ME

I am Piyush Aggarwal, a Computer Science Engineering undergraduate student currently in my third year (graduating by June 2020) from MAIT, New Delhi, India. I'm an open source contributor and a developer. Ever since I attended that first meetup of [ILUG-D](#)(India Linux Users Group - Delhi), that community has become my second home. I have been a volunteer, speaker, organizer, contributor, mentor and mentee and spend most of time working with and for FOSS. My interest lies in developing tangible solutions that make good use of the resources at hand , and in my free time I experiment with technologies and tools mostly around this.

AVAILABILITY

I'm willing to work ~7 hours a day for 6 days in a week. Hence, I'm not willing to take any vacation longer than a day per week. My university examinations are scheduled during the first week of June and I'll work on Sunday to compensate for study time. I'll also maintain logs on my blog and KDE wiki about my experience and technical skills I learn during the program.

WHY KDE

I have been spending my time in the community channels since November 2018; ever since my first encounter with kubuntu. I was totally amazed by the great experience offered by KDE Plasma. Later on, whenever I hopped distros, KDE Plasma , Konsole, Dolphin and KDE Connect became totally necessary to make me feel at home. When I started contributing by discussing KDE software in forums, I came to know about KDE Connect's team, and decided to start contributing to KDE Connect with the hope of giving back to the developers' efforts in my own way of a learning developer.

WHY KDE CONNECT

I was looking to build a simple application that has SSH based file transfers from PC to android and back, and KDE Connect had it implemented already along with loads of good stuff! So this is a project I was interested in for my own use, long before I became aware of KDE Connect!

I want to work on the Windows port specifically, because after spending some months with the team, I now know some of the ways I can help KDE Connect, end goal being to make this project more accessible and noticed to an even bigger user base. I personally am also excited to be able to use it on the Windows based PC at my home!

Additionally, working on/with backends will allow me to work with more teams and foster a richer learning and community experience.

OTHER PROJECTS

I'm not looking for any other project apart from this in Google Summer of Code 2019.

RECENT CONTRIBUTIONS

- KDE

- Fix Find My Device plugin (Windows port) [[1](#)] [[2](#)]
- Add menu Icons for Mobile build of KDE Connect [[link](#)]
- Allow disabling persistent notification in Android 8 and up [[link](#)]
- Other code contributions listed at Phabricator [[link](#)]
- Instructions to build KDE Connect on Windows [[link](#)] [original source: [link](#)]
- Other contributions to the KDE Community [[link](#)]

- **Pandas**
 - Documentation: Fix doc-string for `combine` and `combine_first`. [\[link\]](#)
 - Algo: Add support for PEP 3141 numbers [\[link\]](#)
- **PyCon India Mobile App**
 - Some fixes and enhancements [\[link\]](#)

SOME LINKS

- My blog : <https://journal.piyush.tech/>
- Twitter : <https://twitter.com/brute4s99>
- LinkedIn : <https://www.linkedin.com/in/brute4s99/>
- GitHub : <https://github.com/brute4s99>
- KDE Identity : <https://identity.kde.org/index.php?r=people/view&uid=piyushaggarwal>
- KDE Phabricator : <https://phabricator.kde.org/p/brute4s99/>
- My resume : <https://piyush.tech/resume>

CONTACT DETAILS

- Email : piyushaggarwal002@gmail.com
- IRC Nick : brute4s99 at IRC Freenode
- Phone Number : (+91)-9210775170