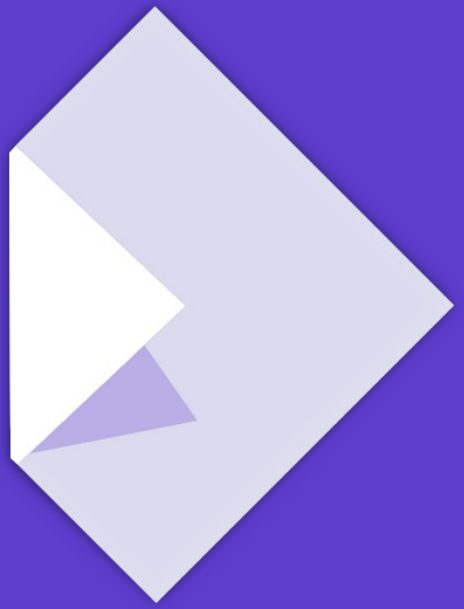
FOSDEM 2021

Porting LibreOffice to macOS on Apple Silicon



Collabora

Tor Lillqvist

Software Engineer at Collabora

@TorLillqvist

Introduction: About myself

I am Tor Lillqvist. I work for Collabora as a contractor

- I have worked on LibreOffice and its predecessor for over ten years
- I enjoy travelling by train and photography
- I am an atheist and a feminist
- I live in Helsinki, Finland, with family, including a dog

LibreOffice on macOS history

LibreOffice on macOS history

- Back in ancient times (early 2000s), OpenOffice.org used X11 also on Mac OS X (as it was called then)
- Later it was changed to use platform-specific APIs on Mac
- For some reason these APIs were referred to in the naming of classes etc as *Aqua* in the OpenOffice.org codebase, even if Aqua is not the name of any API. It is (was) the name of a visual theme
- *Aqua* still persists in identifiers in LibreOffice, which is sad

Fast forward to 2020

Fast forward to 2020

- For many years, rumours that Apple will switch from Intel to ARM on Macs
- Just like they earlier switched from PowerPC to Intel, and before that, from Motorola 68000 to PowerPC
- macOS had for some time already been 64-bit-only, before that it had supported both 32-bit (i386) and 64-bit (x86_64) Intel code
- The iOS product line had used ARM all the time, and had also switched to being 64-bit some years ago

Summer 2020

- In the summer of 2020 Apple finally announced such a switch
- The Mac product line would transition to “Apple Silicon”
- The transition was said to take several years
- No actual Apple Silicon based machines were announced
- Registered developers could rent “Development Transition Kits"
- End-user hardware was promised before the end of the year

Fall 2020

- In November, the first Apple Silicon Macs were announced and became order-able shortly after
- Low-end machines (Mac Mini, MacBook Pro, and MacBook Air) with a system-on-a-chip (SoC) called “M1”
- Despite being “low end”, their CPU performance is pretty impressive
- Also very reasonable power consumption, and thus much less fan noise than the Intel-based machines they replaced

LibreOffice porting starts

- x86_64 builds work on arm64 thanks to the instruction set translation functionality in macOS (Rosetta 2)
- Porting LibreOffice to macOS on Apple Silicon was started by me in the summer
- The first commit was on June 23, *Initial WIP steps for building for macOS on Apple Silicon*
- At that time I was using an Xcode beta and cross-compiling from macOS on x86_64 to macOS on arm64

There are two hard problems...

- One hard question is what to call the architecture: arm64 or aarch64
- Just like with amd64 vs x86_64, there is no consensus on what the architecture is called
- Apple itself in material for the general public uses “Apple Silicon”
- In more technical contexts, they use “arm64”
- The LibreOffice and 3rd-party configury and build systems use either name depending on context

How the port proceeded

- The pace of the port was initially very slow, intentionally
- I did it mostly for fun, in my spare time
- In August the pace accelerated a bit when Collabora got a Development Transition Kit, but still it was not a full-time effort
- When actual end-user hardware was announced and becoming available, more time was spent to have LibreOffice actually work fully
- The final commit to make core LibreOffice functionality work properly was on November 12, *Improve the C++/UNO bridge for macOS on Apple Silicon*

The easy bits

The easy bits

- Most of LibreOffice's own code already is portable and doesn't rely on odd architectural details
- The architectural details on arm64 aren't much different from other 64-bit platforms anyway
- The sizes of data types, byte order, etc in macOS on arm64 is identical to those in macOS on x86_64

The hard bits: 3rd-party libraries

The hard bits: 3rd-party libraries

- The 3rd-party libraries that LibreOffice bundles are more problematic
- Especially the large, old, and complex ones like Python, NSS, OpenSSL, PostgreSQL, and Firebird
- Each of them has its own special snowflake build system
- Not surprisingly, none of them were yet prepared to be built on macOS on arm64

The hard bits: 3rd-party libraries

- Some of them even still contain “support” for Mac OS X on PowerPC
- In one case the 3rd-party configury even deduced that since this is a Mac, but it isn't Intel, it must be PowerPC
- Coming up with suitable patches for 3rd-party libraries was tedious but in retrospect not really hard after all
- Just boring inspection of what is going on in configure scripts, and some trial and error

The hard bits: 3rd-party libraries

- But: It might build, but how much of the often obscure and not unit-tested 3rd-party code works in macOS on arm64?
- Good question
- Next question please

More hard bits: The C++/UNO bridge

More hard bits: The C++/UNO bridge

- The hairiest part in LibreOffice itself is the C++/UNO bridge with its run-time generation of machine code and construction of C++ exception objects
- Luckily LibreOffice was already ported to Linux on arm64
- Initially I just used the exact same Linux arm64 bridge code, and it indeed worked at least for light experimentation

More hard bits: The C++/UNO bridge

- The parameter passing in macOS on arm64 is slightly different than that in Linux on arm64
- Eight 64-bit registers are used first, that is identical
- Following parameters are allocated stack space differently
- Linux uses a full 64-bit stack slot for each parameter
- macOS packs them tightly, according to their natural alignment

More hard bits: The C++/UNO bridge

- The arm64 bridge code that handles parameter passing just needed to be modified to take that into account for macOS
- In retrospect, was straightforward, but I was scared to start

More hard bits: The C++/UNO bridge

- The arm64 bridge code that handles parameter passing just needed to be modified to take that into account for macOS
- In retrospect, was straightforward, but I was scared to start

Current state

Current state

- LibreOffice Vanilla and Collabora Office in the Mac App Store are now universal apps
- Just like for x86_64, Java bits are not included
- I assume TDF will also start distributing arm64-only or universal app builds
- Especially the bundled 3rd-party libraries probably have problems still, help needed

Collabora Productivity

Thank you!

Tor Lillqvist

@TorLillqvist

tml@collabora.com