Seamlessly Composing Music Through Native Windows on Arm

Case Study

Company snapshot

Name: **StaffPad**

Description: **StaffPad is an award-winning music notation app, designed for pen and touch, and built for composers. The app is built by an international team of developers from a multitude of backgrounds, including musicians and composers, who are incredibly passionate about creating the most exciting composition environment on earth.**

Website:
https://www.staffpad.net

Goal

Design a music notation app with high performance, long battery life, and multiple-device connectivity.

Solution

Windows on Arm

Benefits

- Improved user experience
- Excellent performance
- Superior low latency

A Natural Way to Compose Music

StaffPad is a music notation app designed to make the process of writing and performing music easier and more natural by letting users write music in their own handwriting, touch edit, and hear the score played back. It features handwriting recognition for music notation, lifelike playback capabilities, and a connected "Reader" feature, which connects a score across multiple devices and can be updated in real-time. StaffPad for Windows on Arm is available for tablets and 2-in-1 laptops with pen and touch capabilities, including the Microsoft Surface Pro X built on an Arm-based custom processor.

The speed, battery life, instant-on, low-cost and continuous connectivity of Windows on Arm devices all benefit the experience of using StaffPad. The instant-on feature means that when musicians and composers have a creative idea, they have the tools they need available immediately. They do not want to wait for the device to boot. The long battery life means no wires in the way before users start composing music through the app and no interrupting their creative flow with charging alerts from the device. Finally, the continuous connectivity of Windows on Arm devices means users can use the StaffPad app wherever they are, with no need to log-in to public Wi-Fi networks if they are away from their own.

A Painless, Low-Effort Process

StaffPad for Windows is built on the Universal Windows Platform (UWP), using Visual Studio Enterprise. The app is split into two layers: “Core” and “Native Experience.” Core, which is the main logic of the application, is written in pure C++. Native Experience, which handles all of the UI, interaction and native functionality, is written using XAML and C#, and uses Win2D and other UWP frameworks.

> “Once the toolchain is set up, the process of building apps for Windows on Arm is painless and low-effort.”
>
> David William Hearn, Founder and Director at StaffPad

As StaffPad is built as a UWP application, this provides a common app platform on every device that runs Windows 10. This means that the UWP core APIs are the same on all Windows devices. The only challenge for StaffPad was needing to individually re-write or upgrade its own libraries or any third-party libraries for Windows on Arm native performance.

To help ensure that its existing library work would also run native on Windows on Arm devices, StaffPad re-developed some of the audio engine vector acceleration, in particular the engine libraries with FFT processes that had previously been optimized using the IPP libraries. StaffPad decided to rebuild its own FFT code in pure C++ to provide ultimate control and take advantage of more general, native optimizations that are not so specific to a single architecture. More details about how to make specific libraries work native on Windows on Arm are available in this article.

Once the libraries were updated, StaffPad had to recompile a few audio codecs and other components. The process of returning to the source code and rebuilding for ARM64 before linking into the VS Project and referencing in the ARM64 build target was straightforward and took just a couple of hours. The final step was to compile the whole application project and deploy the app on the Microsoft Surface Pro X for testing and evaluation.

Superior Windows on Arm Experience

It is important to have apps available as native ARM64, as the fewer number of apps using emulation layers means a far better overall user experience for Windows on Arm users. Compared to apps using emulation layers to run on Arm-based devices, native Arm apps provide faster load times and reduced latency.

“The responsive pen and touch capabilities, long battery life and connectivity capabilities are highly desirable tools for many composers and musicians,” said David William Hearn, CEO and founder of StaffPad.

Microsoft Surface Pro X customers who use StaffPad speak very positively about the app in forums and online reviews, highlighting that it often outperforms the other version, and is snappy and smooth to use without glitches during complicated music scores.

“We feel strongly that other developers should build natively for Windows on Arm. Arm and Microsoft have both stepped up to the plate, so now it’s time for developers to do the same,” Hearn said .

> “Not only will they gain more customers because of the exceptional user experience, but they will be able to take advantage of the increasingly diverse and expanding device ecosystem due to the capabilities of Windows on Arm.”

Learn more about StaffPad.

Learn more about developing for Windows on Arm.