Developer Series: Improve UI Experience with UxMedia

August 5, 2021

Q&A

1. **Future Deployment: What about Mac OS X support?**

   We've heard interest in the past. This is a great request to raise though your sales channels.

2. **Do we need to have JAVA on local machine?**

   No. Java is running the Niagara station itself (as it does in all stations), but the UI is being presented in the Web Browser using only HTML/CSS/Javascript, so the end user does not need to have Java installed.

3. **How is the render performance on devices like tablets/smartphones or industrial touch panel?**

   What if we are running a powerful server machine running RHEL, but the user also has a fast machine with a modern browser?

   You have a Jace-8000 and "$50" burner phone. Will there be any positive improvement, negative improvement, or neutral improvement?

   The performance of UxMedia is all about the capabilities of the user's device relative to the Niagara station. UxMedia offloads the graphics rendering effort from the station to the user's device. If your station is running on a JACE under load, and your users connect using relatively recent devices, then you are likely to get a better performance improvement than if they connect with an older, weaker device and the station is running on a strong PC.

4. **If we are happy with our HxPx HTML5 / JavaScript widgets, and their performance (we use them 99% of the time from a Supervisor) - is there any imperative for us to move to UxMedia?**

   If you are satisfied with your UI as it is, there's no urgency for updating to UxMedia. The initial UxMedia effort is spent on achieving the same functionality as offered by HxPx pages today, just with better performance on a wider variety of devices. Future development work may include UxMedia-specific enhancements you might choose to take advantage of at that time.

   Even when the station is running on a strong Supervisor, one potential advantage of switching to UxMedia is faster real-time updates. Hx pages usually update on a 5-second cycle, but since UxMedia uses BajaScript for updates, it can reflect changes much more quickly.

5. **Are all the Px canvases available for Ux? Does scaling graphics in the browser change?**

   CanvasPane is fully supported in UxMedia, including scaling.

6. Does Tridium plan on updating the documentation on how to properly setup a TDD environment?

   The documentation on grunt-init and grunt-init-niagara in the Building JavaScript Applications page is a little out of date, but "pretty close" – we'll bring this fully up to date in an upcoming release. You can check the READMEs on their respective pages at github.com/tridium for the most up-to-date information.

7. Should we use module/KitN4Svg for graphics or file/KitN4Svg? Both can be cached.

   The choice between copying individual files to your station, vs copying the entire module, comes down to your preference and how much file storage is available on your device. Both ways are supported. To configure the file size threshold at which the Px Editor decides to copy individual files to the station, see the "niagara.ui.px.maxImageModuleFileSize" property in system.properties.

8. Does Tridium plan on posting an example of using the updated APIs on github?

   There are no immediate plans to post on Github, but live coding examples may be added to docDeveloper in a future release.

9. Where can I get a copy of N4.10 version to get familiar with UxMedia?

   4.10 is out and live! Reach out to your OEM or distributor for access to a download via the Niagara Software portal.

10. Will N4.11 have any UxMedia improvements that, if we are thinking of doing some N4.10 UxMedia in the next few months it might be best to wait until N4.11 so we don't have to redo some new feature.

    The only changes to UxMedia in 4.11 are defect fixes – there are no major backwards-incompatible changes.

11. The slowest Px that I have observed is tabbed page views of multiple scheduler P x files. Will UxMedia accelerate this?

    With UxMedia, the browser would download the raw Px data, set up the rendering and data bindings itself, and use BajaScript to retrieve the scheduler data. The station's only responsibility would be to provide the Px data (once) and retrieve the schedules themselves. If the user is using a reasonably capable device, I think you are likely to see good performance. Let us know how it goes!

12. History views, any improvements on that with UxMedia or is that TBD depending on what you have going on?

    There are no UxMedia-specific changes to History Tables. Support for History Tables should be equivalent to HxPx's.

13. How is UxMedia different then the caching? Doesn't caching do the same thing?

    UxMedia is more than just caching – it is a new way of rendering Px pages, where the rendering and layout work is shifted from the station to the browser, where it can be more capably handled. It does make use of caching, however. When you use UxMedia to view a Px page (whether directly, or embedded in another page as a PxInclude), that raw Px data will be downloaded once from the station and cached by the browser. The next time you visit that page, the Px data will be retrieved from the browser's cache instead of retrieving it from the station, which saves a network call.

14. Is there a long-term plan to drop Px support?

    Not in the near term. I would expect it to be quite some time before we have that conversation.

15. Does the TabbedPane in UxMedia render on a per tab basis? I understand that the TabbedPane in HxPx loads all tabs at once - which can be a performance bottleneck.

    The TabbedPane in UxMedia does currently render all tabs at once. However, a web browser is more capable than a JACE at performing this rendering and layout, so you should still see better performance in UxMedia in this case.

16. Can we still build relativized pages with UxMedia?

    Yes, relativized pages can be built with UxMedia.