Niagara 4.15 – Fox Over Websocket

July 17, 2025

Q&A

1. **Is Niagara Deamon also going to be using Websocket and 443?**

   Niagara Daemon runs a separate web server from the station's web server. Niagara Daemon is already using HTTPS and WebSocket, however, it must use a different port than the web server running in the station process since they are separate processes. Because the station web server (configured in the station's WebService) typically uses standard web port 443, the Niagara Daemon must use a different port (by default, 5011). However, you can change these ports as needed.

2. **Is it possible to get notified if the failover to Websocket happens?**

   When connecting from Workbench, you will notice that the fox icon badge is different when a Fox over WebSocket connection is made. You will also see that any ORDs to which you navigate will have a "foxwss" session ORD scheme (instead of "foxs" or "fox").

   For station-to-station connections, you will know if a NiagaraStation in the NiagaraNetwork is using Fox over WebSocket when the new "Fox over WebSocket In Use" readonly boolean property on the NiagaraStation's Client Connection is set to true. A notification could be made by linking from this property to a BooleanWritable that has a change of state alarm extension that's configured to send an alert on a true value.

3. **Can you continue to use the WebServer's DoS protection mechanisms while using Fox over Websocket?**

   Yes, since Fox over WebSocket is connecting through the station's WebService, it will honor any web server configuration settings that you have in place there.

4. **Can I use Websocket to connect a JACE internal Lan of customer with internet connection and a supervisor on another server external customer LAN reachable on 443?**

   In any scenario that the station's WebService is accessible via HTTPS, then Fox over WebSocket can also be used to connect to that station since it is using the station's Web Server to make the connection. Fox over WebSocket is a new way to communicate the existing Fox protocol for Workbench-to-station and/or station-to-station connections, just like foxs has done in the past. The only difference is that Fox over WebSocket routes the Fox traffic through the station's WebService and not directly to the station's FoxService (like foxs does). It is not a mechanism for "tunneling" a connection through one device to another downstream device.

5. Can you discuss what latency is encountered using the web socket connection method and what's the impact to the Stations host CPU etc. please?

   Fox over WebSocket (foxwss) does involve a little additional latency overhead when compared to foxs. That's because the fox data has to be packaged and sent over the WebSocket. The latency overhead is likely not noticeable for most use cases, however, foxs is not going anywhere, so if you want to maximize performance, you can continue to use foxs, which also utilizes TLS for security. CPU performance should remain similar to how it is now. The fox traffic is simply being routed through the station's existing WebService (via HTTPS).

6. Is the FoxSS only property only for outgoing connections or also for incoming?

   The "Foxs Only" property on the station's FoxService is unchanged in behavior from what it did before. It is used for incoming (server) connections made to the station, not outgoing (client) connections made from the station to another station. If the non-TLS (legacy) "Fox Enabled" property is set to true, then the "Foxs Only" property set to true will cause any non-TLS connection attempts to the station to be redirected to the foxs (TLS) port for secure communication. Tridium recommends not to enable the legacy, non-TLS "Fox Enabled" property, but if you must, then we would recommend keeping the "Foxs Only" property enabled as well. These properties are unrelated to the new Fox over WebSocket behavior.

7. Any reduction in throughput with Websockets? Slower or faster subscription updates etc.?

   There can be a slight reduction in latency when using Fox over WebSocket compared to foxs, but it is typically not noticeable. See similar question and answer above.

8. Can the platform be accessed by Fox Over Websocket?

   The Niagara device's platform process is a separate process from the station (application) process. The platform process has its own web server (typically running on port 5011) that is independent from the web server running in the station process (which typically runs on port 443). Fox over WebSocket (and foxs for that matter) is NOT used for a connection to a Niagara device's platform, it is only used for a connection to the station process. The platform connection is based on HTTPS and WebSocket, however it does not use the Fox protocol at all.

9. If you have a supervisor and 4 JACEs at a remote location, can you use Websocket to do tunneling?

   Fox over WebSocket is a new way to communicate the existing Fox protocol for Workbench-to-station and/or station-to-station connections, just like foxs has done in the past. The only difference is that Fox over WebSocket routes the Fox traffic through the station's WebService and not directly to the station's FoxService (like foxs does). It is not a mechanism for "tunneling" a connection through one device to another downstream device.

10. Is the failover feature for larger installations when you're busy changing things over and updating JACEs here and there, so it's kind of like a catch-all?

    Yes, it can be. It can also be useful for existing Workbench bookmarks and hyperlinks to remote ORDs in Px graphics accessed from Workbench. In those cases, if the existing absolute ORD in the link currently contains the "fox" or "foxs" ORD scheme, and then you either turn off fox and foxs (or the firewall starts preventing access to those ports, typically 1911 and 4911), then the fall-through to foxwss will allow those existing absolute ORD links to continue to resolve. So, it can save the hassle of finding and replacing all of your ORD bookmarks (and ORD px bindings) to change "foxs" to "foxwss" in the session ORD scheme part.

11. If you have an internet connection to the remote Supervisor, can you tunnel into the JACEs using your Workbench?

    No. Fox over WebSocket is a new way to communicate the existing Fox protocol for Workbench-to-station and/or station-to-station connections, just like foxs has done in the past. The only difference is that Fox over WebSocket routes the Fox traffic through the station's WebService and not directly to the station's FoxService (like foxs does). It is not a mechanism for "tunneling" a connection through one device to another downstream device.

12. Assume if a JACE using FOXWSS is trying to reach a JACE/Supervisor across WAN, would this still be an issue since the remote side firewall isn't expecting the incoming request on 443 unless the firewall has a rule to forward that data to the Sup behind the firewall?

    As long as the firewall allows connections to the configured HTTPS port (443 by default), then Workbench-to-station and station-to-station connections should work using Fox over WebSocket. If you already have browser access to the station via HTTPS, then you can expect Fox over WebSocket to work too (since all Fox traffic is routed through the station's WebService).

13. Are there plans to add similar support for the platform connection?

    The Niagara device's platform process is a separate process from the station (application) process. The platform process has its own web server (typically running on port 5011) that is independent from the web server running in the station process (which typically runs on port 443). Fox over WebSocket (and foxs for that matter) is NOT used for a connection to a Niagara device's platform, it is only used for a connection to the station process. The platform connection is based on HTTPS and WebSocket already (so no need to do something similar to foxwss for it).

14. If we enable Foxs Websocket and also have the WebService run on 443, can I still connect to the station through the browser to see graphics?

    Yes, absolutely! The browser experience should be unchanged. If you can connect from a browser via HTTPS to your station, starting in Niagara 4.15, you can now also access that station via Fox over WebSocket to that same HTTPS port for Workbench-to-station and station-to-station connections as well.

15. Is 4.15 required on both ends of the connection to support FOX over Websocket?

    Yes, Niagara 4.15+ is required on both sides of the Fox over WebSocket connection.

16. Once the failover occurs to the Websocket port, will the system detect a reestablished Foxs connection and reconnect?

    Not immediately. The foxwss session will continue for the duration of the session. Once closed, on the next connection attempt, it will again try the foxs connection first (and fall-through to foxwss as needed).

17. Can WebSocket connections using port 443 connect through VPNs that limit traffic to port 443 only?

    As long as you can connect to the station's configured HTTPS port (default 443 as configured in the station's WebService), then you should be able to connect to the station via Fox over WebSocket. As an easy test, if you can currently connect to the station via HTTPS from a browser, then you should also be able to connect to that station via Fox over WebSocket in Niagara 4.15.

18. Does this affect Linux based servers that need to use internal forwarding to 8443?

    It should be fine. As an easy test, if you can currently connect to the station via HTTPS from a browser, then you should also be able to connect to that station via Fox over WebSocket since it uses the same type of connection to the same port.

19. If using Niagara Remote, can Fox over we socket be used to connect workbench remotely leveraging the Niagara remote session?

    No, not at this time, but stay tuned!

20. Is there any guidance on migrating a system to using WebSocket? Can it be changed in bulk with provisioning or with program service?

    Fox over WebSocket is enabled by default when you upgrade your station to Niagara 4.15+. However, it does require that the station's WebService has its existing "Https Enabed" property set to true. If needed, there are existing provisioning jobs that allow you to set properties on remote stations in a supervisor's NiagaraNetwork.

21. Is Tridium looking into simplifying SSL certification on their platform? Within my organization a SSL setup is difficult due to the way the IT department manage their SSL certs.

    Yes. For example, in Niagara 4.14, the Certificate Signing Service was added. Refer to this Tridium Talk for more details about how Niagara's Certificate Signing Service can help make certificate management easier: https://www.tridium.com/us/en/services-support/events/2024/06/2024-06-13-certificate-signing-service

22. If multiple JACEs wanted to use Fox over Websocket, they would have to use different ports. Are there any other ports besides 443 which would be easy to use with the IT department?

    The license is based on the number of active tenants.

23. Does this impact the Discover feature in any way?

    No, not significantly. Station discovery is still based on UDP multicast, so it is unchanged for the most part. The only small difference you might notice is that discovered stations will report if Fox over Websocket is enabled (foxwss) so that you can establish the connection to the discovered station using foxwss directly.

24. If using self-signed certificates with 1-year expiry, will all devices need to be re-approved?

    Tridium recommends using properly signed certificates whenever possible, as that is a general security best practice. In Niagara 4.14, the Certificate Signing Service was added. Refer to this Tridium Talk for more details about how Niagara's Certificate Signing Service can help make certificate management easier: https://www.tridium.com/us/en/services-support/events/2024/06/2024-06-13-certificate-signing-service.

25. Can the failover work in the other direction? For example, could I change all my supervisor connections to subordinate JACE to try Websocket first, then fail over to regular foxs - then handle the upgrades to the JACEs as we go along over time?

    Interesting idea for sure. Foxs will always be a little more performant than foxwss (since foxwss must wrap the communication in a WebSocket). So if foxs is available, it may be the preferred choice for optimal performance (although the performance difference may be negiligible and not noticeable). So I'm not sure how often you'd need the fall-through to work in the other direction.

26. Is the driver able to get the accumulated total energy and subtract for each month automatically?

    Yes, NTBS invoice calculations handle both totalized and delta logged historical data.

27. Can this tunnel to multiple internal Niagara stations with only one external 443 port exposed?

    No. Fox over WebSocket is a new way to communicate the existing Fox protocol for Workbench-to-station and/or station-to-station connections, just like foxs has done in the past. The only difference is that Fox over WebSocket routes the Fox traffic through the station's WebService and not directly to the station's FoxService (like foxs does). It is not a mechanism for "tunneling" a connection through one device to another downstream device.

28. Are the Fox settings, timeouts, etc. still in play for Websocket connection just like a 'normal' Fox session?

    Most of the configuration settings in the FoxService are still valid for foxwss connections, with the exception of the "Socket Option Timeout" (still used for failsafe checking) and the "Socket Tcp No Delay" properties. Also note that since foxwss connects through the station's WebService, then the configuration settings applicable to HTTPS in the WebService also apply to foxwss.

29. What input will IT have before remote connection Websocket is available?

    In general, Fox over WebSocket should make conversations with IT easier since only HTTPS connections to the station are needed to support all of the functionality. So this should simplify the port conversation with IT, as port 443 (the standard HTTPS port) is all that is needed to support browser, Workbench, and station-to-station (NiagaraNetwork) connections. That said, don't forget that platform connections will still require a different port for HTTPS connections to the device's platform (port 5011 by default), but platform connections may be less often used since they are needed at commissioning time, software upgrade time, etc.