

Enhanced Graphics with Tag Based Visualization

Frequently Asked Questions

Q. What kind of resource impact is this going to have on a station? 1-in a JACE and 2-in a web supervisor?

A. See Response Below for "What impact does this feature have on a JACE's resources"

Q. What is the responsiveness compared to bound ord and what is the resource load on CPU

A. See Response Below for "What impact does this feature have on a JACE's resources"

Q. What impact does this feature have on a JACE's resources?

A. NEQL-based Ords do take more time to resolve than slot path ords, as it must be resolved on the station-side instead of on the client-side. The amount of time and load on the CPU depends on how many items must be searched and how tags are applied. These queries are not executed across the entire station but are scoped to descendants of or components related to the component on which the graphic is placed. In our testing, graphics did take longer to load but the loading time was still acceptable. Once the graphic is loaded, the queries are not resolved again and the graphic should be as responsive as a graphic based on slot path Ords.

Q. What happens if the NEQL ord matches more than one point based on the tag? What are the errors messages?

A. At this time, one of those points will be chosen for the binding and there is no indication that there were multiple results. The point that is chosen is indeterminate. In the future, we may enhance the single scheme to throw an exception if multiple results are returned by the query and that would trigger the degrade behavior on the graphic. We are also considering ways to order NEQL results and that would make the result selection more determinate.

Q. Are there any built-in tools to help troubleshoot issues around NEQL returning multiple results?

A. Not currently but there is something on our roadmap for that.

Q. How do you add tags to individual points?

A. There are multiple ways to add direct tags. One way is to right-click a component and open the Edit Tags dialog. The batch editor and some manager views also support tag editing. Tags can also be implied using TagRules in the TagDictionaryService. For more details, see docTagging in the help system.

Q. Are you working on automatically adding graphics on equipment depending on their tags?

A. That is a great idea and it is on our roadmap.

Q. Do the tag-based graphics make use of a new NEQL Select statement that is similar to the old Bql Select Statement? Has this been around for a while?

A. NEQL has been around since 4.0 and is documented in "module://docDeveloper/doc/neql.html". NEQL has a Select statement but the "select" keyword is usually left off, such as "neql:n:point". The Select statements searches descendants of a scope component for those that match the filter. NEQL also has a Traverse statement that searches the endpoints of a relation for those that match an optional filter.

Q. Is there a document detailing NEQL commands and queries?

A. See Response Above for documentation requests

Q. Is detailed documentation on the NEQL going to be available?

A. Existing documentation exists at "module://docDeveloper/doc/neql.html". There will also be user documentation around converting Ords to NEQL-based Ords within the PX editor.

Q. Are the NEQL commands going to be like the ones used for the hierarchy feature?

A. Yes, the same NEQL syntax is shared between these NEQL-based Ords, hierarchies, and search.

Q. Will there be support for compound bql queries? Such as query the results of query?

A. Compound BQL queries are supported. If possible, we recommend combining compound queries into one. Since 4.4, it is possible to pipe a

BQL query onto a NEQL query. You cannot pipe a NEQL query onto a BQL query.

Q. Is there a way to search for a specific phrase? Like wrapping it in quotes?

A. NEQL supports a "like" expression that compares a String tag value to a regular expression (regex). See ["module://docDeveloper/doc/neql.html"](#) for more detail.

Q. Can we use string tags?

A. Yes, value tags, including String value tags, are supported in NEQL queries and by bindings that use NEQL ords.

Q. What about VAVs applications?

A. Any graphic representing equipment can incorporate NEQL ords in its bindings.

Q. What components are in the folder "AHU02" in the Niagara Network?

A. The "AHU01" folder in the Niagara Network was a PointFolder that contained proxy points from a single device that belonged to an AHU. The points were named differently under this folder than in the folder where the graphic was initially placed but because the points were tagged, the bindings still resolved. The "AHU02" folder was a DeviceFolder that contained 2 devices that together contained the points that belonged to an AHU. The points were named differently and were at different relative locations than in the folder where the graphic was initially placed. Again, because the points were tagged, the bindings still resolved.

Q. Are tag-based graphics available now?

A. They will be included in the 4.9 release and are available to try through the Early Access and Beta Testing Programs. Please email tridiumbetatest@tridium.com to participate in either or both programs.

Q. You said come see us at NS2020 for more info. Where would I go for "entry-level info" on all this? Any specific workshops at NS2020 for intro to this and hs?

A. Please visit www.niagarasummit.com for further information on the 2020 Niagara Summit.

Q. Is there any ability for a supervisor to be able to recognize and import metadata tags which have been created in a remote Niagara point database when using the point discovery?

A. Tags can be fetched from remote NiagaraStations by setting the persistFetchedTags property to true and by adding/configuring a tagsToFetch property. This is described in the following help documents:

"module://docAnalytics/doc/AcquiringTagsFromRemoteStations-4E7275C4.html" and

"module://docDrivers/doc/AboutPersistingFetchedTagsdocDriver-7CCE5AD0.html".

Q. Tags to fetch feature is cumbersome when using multi-level Niagara architecture, how does it impact the tag-based graphics?

A. Once the tags to fetch feature is configured, specified tags on the points in the remote NiagaraStations will be added as direct tags on the proxy points in the supervisor. These direct tags can be used to resolve the NEQL ords within a tag-based graphic. We will consider how importing tags from NiagaraStations can be less cumbersome.

Q. Is it possible to import tags from something like SkySpark (or something else implementing the Haystack REST protocol) if we already have tags defined there on our points?

A. The nhaystack driver does not yet create Niagara tags on the nhaystack proxy point based on the tags imported over the Haystack REST protocol.

Q. How much take time/effort to tag points in the gray field/ existing project?

A. Tagging points at a gray field site is a challenge. Implied tags using tag rules in custom SmartTagDictionaries can help here. The batch editor can be used to more quickly apply direct tags. There are additional tools on our roadmap for viewing and editing tags that we hope will be available soon.

Q. How will such a graphic work when deployed via a template?

A. As long as the points deployed via a template are tagged correctly, a graphic also deployed with the template should work as well.

Q. Can tags be used when using Virtuals between Supervisors and JACEs?

A. The right solution for tags on virtuals will most likely involve SystemDb. Integration between graphics and SystemDb is not there yet.



Q. How does this operate with regards to the Export Tags?

A. Tag-based graphics exported via export tags are not yet supported.

Q. The new view did not show the overridden state. Is there a setting originally where the status effect should have been set to color?

A. The same ValueBindings are being used for these NEQL ords so the statusEffect property can be used as well. In the demo view, statusEffect was set to none.

Q. Does the limitation of 43 characters of history naming addressed in 4.9?

A. This has not been addressed yet but is on our roadmap.