



# Niagara 4.11 Feature Sneak Peak

## Introducing Archive History Provider

October 21, 2021

### Q&A

**1. Is there a set plan for which future versions will be earmarked as LTS?**

There isn't currently. Our current consideration is to take the pulse of the community to decide when enough new features have been added to ear mark the next LTS version.

**2. If you do a query for 6 years of data but, say, rolled up by month, does that still count as a massive query?**

Yes, it could depend on the collection interval. That's where the 'Max Archive Results Per Query' property could come into play and limit the query results that are retrieved from the relational database for a large time range query. If the limit is reached, the results would be truncated, favoring the most recent records first.

**3. Any ideas floating around to also export the indexed model/object data for the station to the RDB? Like the structure of everything and their tags? Similar to the index created for SystemDB.**

We've been having discussion about the best way to provide remote access to tag and model data, which is not ideal for a relational database table with a fixed schema. Currently the way to accomplish this request is to use the SystemDb on the supervisor. You can run NEQL queries against the SystemDb (via Search or Hierarchies) to find components (e.g. local points or Niagara virtual points), and once you've found those points, you can directly navigate to the histories for those points. Those histories, starting in Niagara 4.11, could then have their data supplemented by the new RDB Archive History Provider.

**4. Are analytic functions working?**

Because any archive history provider implementation (e.g. the RDB Archive History Provider) plugs in at the history service level, any existing applications, views, etc that make history queries will benefit. So this means that Niagara Analytics and any other tools that utilize history queries will immediately benefit from the RDB Archive History Provider because the response to any history query can now include history records sourced locally or from a remote archive. The application that submitted the history query doesn't need to be changed since it just continues to submit and process the results of the history query the way it always has.

**5. Will we be able to get the recording or a copy of the slides?**

After this Talk, the recording will be posted on Tridium.com.

## 6. When will 4.11 be released?

Our goal is to release 4.11 in the upcoming weeks. This is subject to change as we continue to move through the Beta run.

## 7. Is fully indexing a large DB an expensive operation? If so, any impacts server side to the running station? Also, once indexed, are the indexes Niagara-specific, or can they be leveraged by other clients querying the DB?

Using the default configuration of the existing RDB drivers for Niagara, indexes will already be created for your exported tables (although some existing indexes may still benefit from migration for optimum performance going forward, especially if you are using the 'By History Type' export mode). Only if the default configuration was changed (e.g. the 'Use Last Timestamp' property was set to true) would Niagara not have been creating indexes on exported tables in the past. So it's likely that existing installations already have table indexes in place for Niagara exported history tables. In terms of adding missing indexes to existing relational database tables for exported history data, it does take some additional storage in the database. Also, while the indexes are being generated, there could be some additional load on the database. So you will probably want to discuss with your DBA (Database Administrator) prior to using the new 'Migrate to Optimized Data Indexes' action (you may also want to discuss taking a backup first). Also note that indexes (and migration to the new index form) is completely optional. The RDB Archive History Provider will still function without optimized table indexes, but you may notice that query performance is slower without them. The table indexes created by Niagara are not Niagara-specific. Any other client/application making a SQL query to the exported tables could also benefit from the indexes, specifically if the queries are based on a time range.

## 8. Are there any performance enhancements retrieving histories this way?

Querying for local history data will always be the fastest approach. However, offloading older (archived) history data to an external database can be helpful for scalability reasons in your supervisor. You can then optionally reduce your local history capacities to something that satisfies your most common queries for *recent* history data (also make sure the local capacity is large enough to withstand any database down times), freeing up space on your supervisor. You can then still support history queries that go much further back in time, but accessing those older (archived) records may take a little longer since a connection to the external database must be made to retrieve those records on-demand.

## 9. What happens if we reduce the records in the RDB history table for instance by creating stored procedure that create an average record every 4 records in order to save space in the database?

This activity is not recommended on a table that Niagara exported. If you have another client/application that desires this rollup data, you could create a separate table/view in the database to support it. Keeping the table that Niagara exported unmodified will allow Niagara to reliably export new data and read data back at query time.

**10. How large of a dataset were you able to performance test against? Say, a WebChart with 10 histories attempting to load 100k records each? Again, does a worst-case simply crash the client, or are there server-side implications?**

We performed testing in various conditions. There are a number of factors that can influence performance: the version/type of the relational database (note that a production/enterprise quality database is recommended, as they typically perform much better than a free/trial version), network conditions, the 'Max Archive Results Per Query' limit, the size of the exported history table ('By History Id' export mode leads to smaller-sized tables which are usually quicker to query, but it creates a table per history in contrast to the 'By History Type' export mode which has fewer, but larger tables which may take longer to query), and more. Since there are so many variables, it's hard to predict the exact performance you will experience. In our testing, we tried various combinations (up to 2 billion records in a single table using the default 50k limit). Performance was quite reasonable for most test cases, but in worst-case scenarios the history view (e.g. Web Chart) would take a while to load/present the data. No server side (database side) issues were observed in our testing. Your performance may vary depending on your setup, and you can also tweak the properties in the RDB Archive History Provider to optimize performance. You may also want to work with your DBA to optimize performance.

**11. Can you describe security best practices for incoming traffic through firewall to retrieve archival data from off-site for analysis on the on-site supervisor unit, or would recommend viewing that via some kind of online Niagara Interface?**

We've made some updated recommendations on security best practices in our RDB Driver guide.

**12. Is it possible to play with this using SQL express and an SI demo license?**

Yes. We don't fully support express but have no reason to believe the RDB Driver wouldn't work against it. The SI demo licenses will have the features once 4.11 is released. It's worth noting that the MySQL offering has a community edition as well and can be used for testing. Also note that production/enterprise database editions tend to perform much better than trial/community versions.

**13. Could you speak to as to best practices on when to archive as change of value vs regular time-stamp data in a Niagara 4.11 environment?**

In terms of exporting Niagara history data to a relational database, this largely depends on the capacity of your local history and how much room you have for recent history records to be stored locally before it rolls over. By knowing this capacity and estimating how many new records will be added to the history per day, you can make a judgement on how to configure the 'Execution Time' of your RDB History Export descriptors so that the exports will happen before any new records roll off due to the local history's capacity. Also add some buffer capacity/time in case the relational database connection is temporarily down, you'll want your local history capacities large enough to store any new records that occur while the database connection is down so that those records can be exported when the connection comes back up.

In terms of querying archived history records, there should be no difference between change-of-value (COV) and interval collected histories other than maybe the number of records.

14. Is a separate license required to remove to local storage device? Or is it available straight away?

To use the RDB Archive History Provider, your license needs two updates:

- A general “historyArchive” license feature that is needed for using any archive history provider implementation.
- A specific “rdbHistoryArchive” license feature that covers using the RDB Archive History Provider against any supported RDBMS.

15. Can we use the Niagara database maintenance to manage also the remote database records?

No. Once the Niagara history data is exported to a remote relational database, it is no longer managed by Niagara tools. You (or your DBA) can use the tools for your particular relational database to manage those table records once they are there.

16. If you query for five histories, year’s worth of data, does the limit apply to each one? Or the chart as a whole?

The limit applies to each history query. Since there are five separate histories, each being queried separately, in this scenario, the limit would apply to each one, not the chart as a whole.

17. From the development point of view is there any code review to perform?

From a developer’s perspective, starting in 4.11, there is a new abstract class in the public API that can be used to create other archive history provider implementations. This new abstract class is called **BArchiveHistoryProvider**, and it is part of the rdb-rt module. You can use the BajaDoc (in Workbench) for this new class to read about what methods need to be implemented. In addition, if you have code that performs a history query, by default it will now tap into archived data when configured and available on the station. Also programmatically, if you don’t want a history query to tap into archived data (for example, if you are exporting history data to some other location and you don’t want the export to include archived history data which could be large), you can pass in a special Context facet when resolving a history query/ORD or making a connection to the Niagara history database. In those (rare) cases, you can apply a boolean facet named **“excludeArchiveHistoryData”** set to true on the Context, or you can use the new **makeExcludeArchiveDataContext(Context)** method on the HistoryQuery class in the public API for a convenient way to create such a Context.

18. Has this been tested for compliance with FDA validated servers (namely 21 CFR)?

Title 21 CFR Part 11 does not have compliance testing for software development, but rather the FDA reviews a particular site for compliance once installed. The electronic signature offering from Tridium Professional Services provides the configuration options needed to be compliant. The Archive History Provider adds more configuration options to meet compliance and customer need.