Technical Document

Niagara Histories Guide

About histories

Histories are ordered collections of time-stamped records. A single history record (or history) is a collection of specific data values from a component within any station local or remote. For example, a data log is a history.

History services

Each station's **History** space supports three history services. These services appear by default under the **Config > Services** node in the Nav tree.

- The HistoryService manages the **History** space, collecting and storing all histories in the station database. This service is required to provide database support for histories in each station.
- The AuditHistoryService monitors station configuration. A change made by a user to any component property creates a standard audit event record, and a change to a security-related property creates an additional security audit event record in the **History** space. Authentication events (log in and log out) create security event records in the **History** space.
- The LogHistoryService collects some of the messages available in a station's standard output for troubleshooting purposes.

History space

The **History** space provides ways to view and working with histories in the history database. The **History** node in the Nav tree visually represents the **History** space.

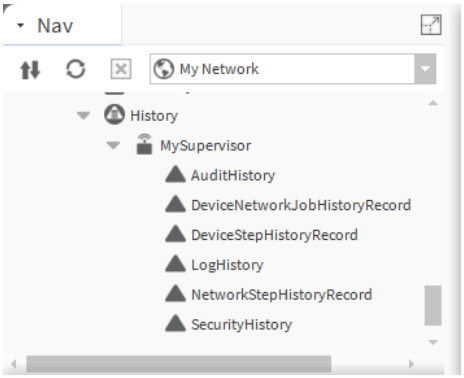
Once you have a history service running, you can access histories in the database using the history ORD scheme. The unique history scheme name "history" and each unique history ID provide identification for the individual histories. This unique ID identifies all history collections. To access a history in the **History** space, expand the Nav tree or click the file open icon and click **Ord**. This opens the **ORD** window.

History views present history information in various formats for both analysis and editing. Views on the **History** space include the following: **Chart**, **History Chart Builder**, **Database Maintenance**, and **Nav Container View**.

History records

Histories are organized by their source station (device).

Figure 1. History records in a Supervisor station



The screen capture shows the history records that appear, by default, under the **History > local** (station name)

node in the Nav tree.

History configuration

Using histories involves a process of collecting, storing and archiving data. You can configure the history collection process to collect the data that you need and store the history records where you want them - locally or remotely.

History configuration includes working with properties such as ID, History Source, Timezone, Record Type, and more.

The History Grouping component uses history properties to customize the organization and display of **History** space contents.

History Nav shortcuts provide convenient navigation links to histories.

History data editing

You can edit and filter the history data in Workbench using the **History Editor** view.

History archives

Creating history import and export descriptors is how you save (archive) histories to a different location (station) from where they originated.

For example, an originating history in a remote controller station may collect only the last 500 records. If imported to a Supervisor station, you can configure the history (using its history import descriptor) with unlimited record capacity.

- **About the history process**
  There are essentially three steps in the history logging life cycle: data collection, data storage, data archiving.
- **About history names**
  By default, when a history extension is added to a component, a history format default string is set to the following: `%parent.name%`. This string automatically names any histories with the name of the parent component and appends a sequential number to additional names, as necessary.
- **About history policies**
  The **NiagaraNetwork**'s History Policies (History Network Ext) node holds an On Demand Poll Scheduler that affects imported histories, if set up for on-demand polling. History Policies also functions as the container for Config Rules that are used when remote histories are exported into the local station. Unlike imported histories, which let you define (and adjust later, if needed) the Capacity and Full policy settings in each History Import descriptor, histories that are exported into the station have no associated component—only the history itself. The Capacity and Full policy for each history is set only at creation time, using the local history policies.
- **About history grouping**
  History grouping sets up alternate navigation for your **History** space based on History properties.
- **About delta logging**
  When you are logging data, such as electric consumption (kWh) or other information that uses a running total, you may want to know the difference between consecutive timestamped values instead of the actual running total. The delta logging feature is provided for this type of calculation.
- **About sampling**
  Sampling uses a simple roll-up technique to get large data sets down to a manageable number of points. This improves chart rendering performance and it smooths out the chart data somewhat which can make it easier to interpret.
- **About security history**
  These histories keep track of security-related events and changes to security-related properties. Separating security history from standard audit history draws attention to the importance of

maintaining a secure system and prevents authentication events from overwhelming the audit trail.

- **Audit trail management**
  An important aspect of station security is preserving an audit trail of user activity. The station commissioning process installs and enables the **AuditHistoryService** by default. This service creates a record for each user-initiated change to each component in the station.
- **About editing history data**
  The **History Editor** view provides the ability to review and modify the history data that you have collected.

About the history process

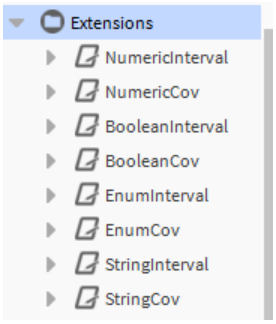
There are essentially three steps in the history logging life cycle: data collection, data storage, data archiving.

Figure 1. Simplified history process

- Collecting data involves defining the properties that specify what data to record and how often to record it. For example, you can collect data whenever a change of value occurs - or at a regular time interval that you specify. To collect history information you need to:
  - Add history extensions to components.
  - Configure the extensions.
  - Use a valid history name (part of the configuration).
- Storing data involves defining the properties of the history database file. For example, you can customize the name of the database file, define the maximum number of records to save, and choose metadata to add to the records.
- Archiving data includes importing and exporting (transferring) records from one station to another station. For example, you can limit your local station records to a small number, which you specify while archiving all collected records to another station.

To extend the functionality of the component, you add extensions to a component's **Property Sheet**. By adding a history extension, you can collect a time-stamped entry in the associated history table for a the real-time value or the status of the component's output. The history palette makes history extensions available.

Figure 2. History extensions in the history palette



The history table is not stored as part of the component's data but is a separate collection of data referred to as the "history."

Parent topic: About histories

About history names

By default, when a history extension is added to a component, a history format default string is set to the following: `%parent.name%`. This string automatically names any histories with the name of the parent component and appends a sequential number to additional names, as necessary.

For example, a history extension on a NumericWritable component creates the default history name: `NumericWritable`. Then, another numeric writable receives the same name incremented to NumericWritable1.

Note:

If you use a literal name and not a script to name a history, you lose the automatic incrementing feature that the script provides. Duplicating extensions that have a literal name duplicates the exact literal name and, therefore, creates an invalid (non-unique) name. In this case, you must rename each duplicated history manually to create a unique name and avoid a fault condition.

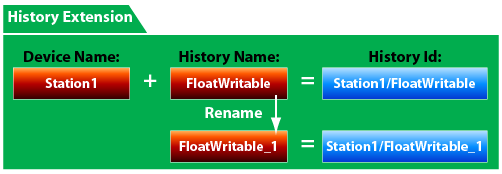
- The character limit for Histories is 200 characters.
- Starting with Windows 10 version 1607, the MAX_PATH limitations have been removed from common Win32 file and directory functions. However, you must take action to opt-in to the new behavior. Refer to the following article for detailed instructions: "https://docs.microsoft.com/en-us/windows/win32/fileio/maximum-file-path-limitation?tabs=registry#enable-long-paths-in-windows-10-version-1607-and-later". The following points summarize the requirements that must be met for enabling the new long path behavior:

    1. The registry key `Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem\LongPathsEnabled (Type: REG_DWORD)` must exist and be set to 1.
    2. The longPathAware element should be included in the application manifest.

    When you enable this opt-in behavior, the directory management functions and file management functions will not have the MAX_PATH restrictions.

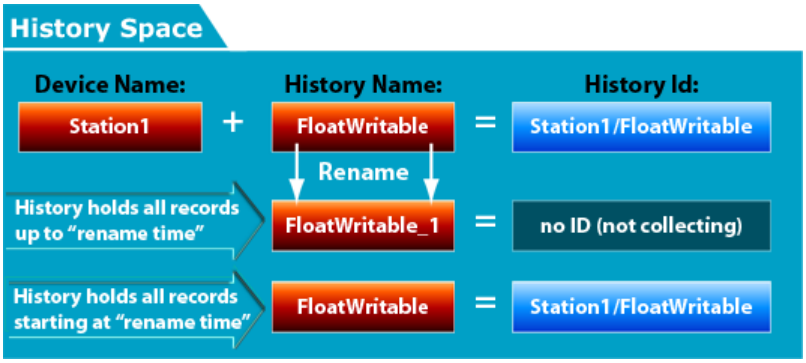
As illustrated in the following image, history names are part of the unique history extension property Id. When you rename a history at the history extension, you are renaming the history at its source. Therefore, the history configuration and the history Id both change. This concept is illustrated here:

Figure 1. Renaming a history in the history extension



If, however, you rename a history in a **History** space view, such as under the **History** space node in the Nav sidebar, or in the Nav container view, you are changing the name of the history as it has been saved in the **History** space not at the configuration (or source) level. Therefore, the history Id does not change and the history extension continues to produce records under the original history name as long as that history extension is active and enabled. This results in a history split where the station no longer updates the newly-named history, as of the time of the renaming, which it contains all the records up to that time. In this scenario, a history under the original name begins with the first record after the renaming and continues recording as configured. This concept is illustrated here:

Figure 2. Renaming a history in the history space



Renaming Summary:

- *No history split*: If you rename a history in either the **Property Sheet** view or the **History Extension Manager** view, you are editing the actual history extension (the source) and, therefore. not forcing a history split.
- *History split*: If you rename a history in either the Nav side bar view or the Nav container view you are editing the name in the **History** space and not actually changing the history Id – the history is split.

Parent topic: About histories

About history policies

The **NiagaraNetwork**'s History Policies (History Network Ext) node holds an On Demand Poll Scheduler that affects imported histories, if set up for on-demand polling. History Policies also functions as the container for Config Rules that are used when remote histories are exported into the local station. Unlike imported histories, which let you define (and adjust later, if needed) the Capacity and Full policy settings in each History Import descriptor, histories that are exported into the station have no associated component—only the history itself. The Capacity and Full policy for each history is set only at creation time, using the local history policies.

Note: You export histories into the station working under a remote station meaning, from a view of the Histories device extension under the station that represents this (local) station.

Default history policies

The On Demand Poll Scheduler contains a set of standard poll component properties as described in the *Drivers Guide*. These properties affect the operation of the on-Demand polling for histories so enabled.

Default Rule contains a set of properties with wildcard matches to all stations and history names, specifying “unlimited” capacity and a “roll” fullPolicy. So that any history that is exported into the station (from a remote station) is archived using a local history configured with unlimited capacity.

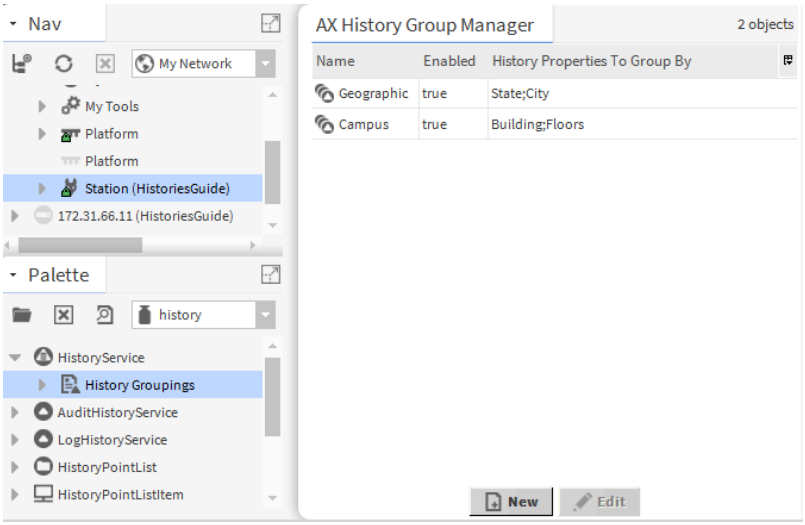
Given the vast storage capacity of a Supervisor host PC, the default rule may be acceptable on the target station of most exported histories. However, if exporting histories to controller stations, you should definitely change the Default Rule of the History Policies under its **NiagaraNetwork** to specify smaller capacities. Even for a Supervisor station, you may wish to change the default rule, and/or add additional optional config rules, as needed.

Parent topic: About histories

About history grouping

History grouping sets up alternate navigation for your **History** space based on History properties.

Figure 1. History Groupings component comes as part of the HistoryService



The History Groupings component, located under the **HistoryService**, specifies which History properties to filter on for organizing a station’s **History** space. These alternative navigation displays are based strictly on the History properties that you select and are a departure from the device- or driver-centered navigation model that builds a Nav tree based upon networks, devices, and points.

With History properties, you can use one or more HistoryGroup components to set up Central Region, functional, or any other desired organization of your **History** space. You can create multiple navigation schemes and use them, as needed, or use the default history navigation structure, which always remains visible. History grouping allows you to simultaneously display histories in multiple Nav paths.

Parent topic: About histories

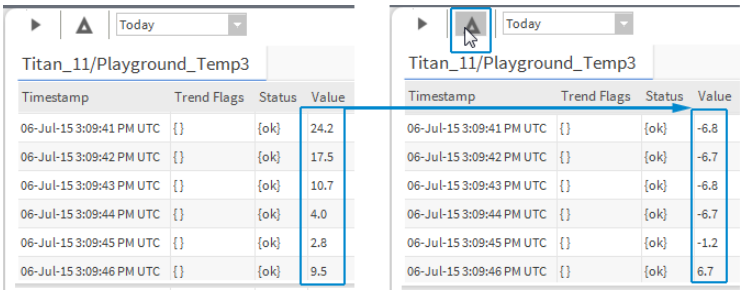
About delta logging

When you are logging data, such as electric consumption (kWh) or other information that uses a running total,

you may want to know the difference between consecutive timestamped values instead of the actual running total. The delta logging feature is provided for this type of calculation.

For delta logging, data is logged (as normal) using the appropriate NumericCOV (Numeric Change Of Value) or NumericInterval extension. Then, in the **Chart** view, you simply select the **Delta** command to plot delta values. Similarly, in an exported CSV file, you can review the delta values instead of the running total value, as shown below. Delta values are computed by taking the difference between one numeric record and the next. The timestamp of the last record (of the two) is used as the timestamp for the delta value.

Figure 1. Delta logging history values shown in History Table view



Two other properties that apply to delta logging are related to the concept of rollover.

Rollover occurs when a running total reaches a defined maximum number and then resets to zero or another defined number. The defined maximum number is represented in the history extensions by the Max Rollover Value property. The reset value (which is often zero) is represented in the history extensions by the Min Rollover Value property. These properties allow you to specify the behavior of the delta logging when the rollover occurs. If you do not know these values or if they are not specified, then select the null option for these properties.

Consider the following example. If you are logging energy consumption with the Max Rollover Value property set to 999,999 and the Min Rollover Value set to 100, then when a rollover is detected, the delta logging bases its delta calculations on a maximum value of 999,999 and a subsequent initial value of 100.

Parent topic: About histories

About sampling

Sampling uses a simple roll-up technique to get large data sets down to a manageable number of points. This improves chart rendering performance and it smooths out the chart data somewhat which can make it easier to interpret.

Rollup (or Rollup Interval) is an interval of time that is used to determine what (and how) data is presented in your chart. The effect is that rollup groups the data into auto-configured intervals. Each point displayed, using the rollup, represents a designated time interval before the specified plot time. This interval is a stat that can be seen in the Settings window **Sampling Period** property (only visible once sampling has begun).

When the focused point array is larger than 2500, roll-up buckets are created and calculated based on the available time based on total duration/2500. The roll-up amount is rounded up to the next highest time increment. For example, if the calculated roll-up bucket is 2.5 hours, a roll-up bucket of 3 hours is used and the roll up will start at an even increment. So if the first entry is 2:35 am, then the first rollover bucket will be 2:00 and the next bucket will start at 3:00.

Auto-Sampling turns on automatically if the focused data set exceeds 2500 and turns off automatically once the focused data set is below the 2500 threshold.

Sampling is on if the **Chart** view **Sampling** command is selected. Alternately, in the settings **Sampling** tab, you

can turn on sampling by disabling Auto Sampling and setting Sampling to true. Additional settings allow you to configure the Facets Limit Mode, Show Start Trend Gaps or hide them, and Show Data Gaps.

Sampling enhancements

In Niagara 4.1 and later, enhancements in sampling protect against an unlimited number of points in a web chart consuming all available memory on the PC. The number of points are configurable with a system.property, as are the limits for when to start auto sampling and when to force auto sampling on.

Data chunking, which is used for all data, accommodates large histories (those exceeding configured size limits) resulting in improved performance. Chunking limits the amount of memory consumed while data is loading. The chart displays once information about the series is received and data displays in the chart as it comes in from a chunked response.

Note too, that an added chart command (in Niagara 4.1 and later) functions as described here:

- The **Stop** command (■), becomes visible only while data is loading. At any point during data loading, you can press the button to stop the chunking process. Press **Stop** once to halt data coming from the server. While stopped, the button displays a red border. Press **Stop** a second time to reload the data.
- Changing the **Time Range** while the page is loading also triggers the **Stop** command followed by a page reload.
- For performance reasons, you cannot turn off sampling once the number of points in the focused data set exceeds 50,000 (or the configured default maxSamplingSize). If you attempt to turn off sampling a popup alerts you that “The chart has too many points (>50,000). Sampling cannot be turned off until the page is focused on fewer points.” You can change the **Time Range** to focus on fewer points.

The following configurable system.properties (!defaults/system.properties) allow you to fine tune sampling defaults:

- `#niagara.webChart.autoSamplingSize=2500`

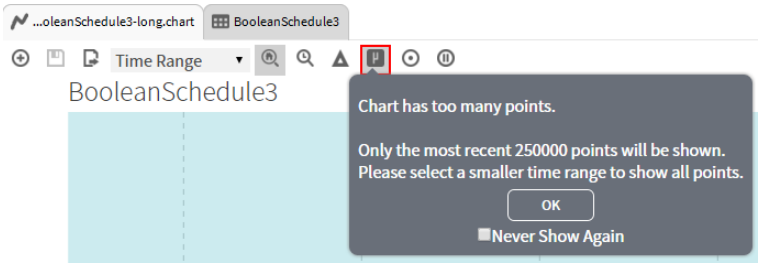
  This property sets the default auto sampling size. This applies when sampling is turned on to improve web browser interaction.

- `#niagara.webChart.maxSamplingSize=50000`

  This sets the default maximum sampling size. When exceeded, sampling cannot be turned off.

- `#niagara.webChart.maxSeriesCapacity=250000`

  This sets the maximum capacity for a data series. When the indicated **Time Range** loads more points than the configured maximum capacity (default=250,000), the **Sampling** command displays a red border and a popup (shown) alerts you that only the most recent 250,000 points will be shown in the chart.



Parent topic: About histories

About security history

These histories keep track of security-related events and changes to security-related properties. Separating security history from standard audit history draws attention to the importance of maintaining a secure system and prevents authentication events from overwhelming the audit trail.

Niagara 4.9 introduced security history. Similar to audit history, security history contains only security-related events, such as log-ins and —outs (authentication) and changes to security-related properties, such as changes to the properties that enable and disable secure communication.

The Security History Service provides the same properties as the standard History Service provides. You configure both sets of properties using the **AuditHistoryService**.

The station stores security-related property changes in both the security history and audit history logs. It stores authentication events only in a security history. This prevents frequent connection events from flooding the audit history.

Security facet

The Security History Service provides a facet labeled “security.” Wherever you can configure facets on a property, you can add this security facet to the property. Any user changes to the configuration of the property will generate a security history record.

Default security-related properties are those that appear on the security dashboard. The Niagara 4.9 release designates several frozen properties in the **FoxService** and **WebService** as security-related properties. These include:

- **FoxService**: Fox Enabled, Foxs Enabled, Foxs Only Foxs Min Protocol, and Cipher Suite Group
- **WebService**: Http Enabled, Https Enabled, Https Only Https Min Protocol, Cipher Suite Group, Require Https For Passwords and X Frame Options

To view security facets, open a component’s **Slot Sheet**. You cannot remove the security-related facet configured by default on a frozen property.

Parent topic: About histories

Audit trail management

An important aspect of station security is preserving an audit trail of user activity. The station commissioning process installs and enables the **AuditHistoryService** by default. This service creates a record for each user-initiated change to each component in the station.

Setting up an effective audit trail involves establishing your company’s history policies. How often are you going to export histories from each controller station to the Supervisor station?

There are two sides to an audit trail configuration:

- Setting up a controller station involves deciding what information to track. From the controller side, backing up histories to a Supervisor station is a push process. The controller station creates an export descriptor for each history and can then push the history to the Supervisor station manually or automatically on a regular schedule.
- Managing all station histories in the Supervisor station involves creating folders to group histories, which quickly can number in the thousands. This is a pull process. The Supervisor discovers the histories available from each controller station and creates an import descriptor for each history.

Parent topic: About histories

About editing history data

The **History Editor** view provides the ability to review and modify the history data that you have collected.

In some cases, you may want to search through the history data and look for unusual data or “outliers.” An *outlier* is a data value that is far apart from the rest of the data; an extreme value that is either much lower or much higher than the rest of the values in the data set. Outliers are known to skew means or averages, so it may be helpful to identify and edit or hide this data. This does not mean that the data point is necessarily bad, but in most cases the information is more helpful without the inclusion of this unusual data.

CAUTION: It is possible to alter good data and miss filtering some bad data points using the History Editor view.

Workbench provides the ability to find and edit outliers based on property that you specify. The **Configure Outliers** window (shown below) appears when you click the **Configure Outliers** icon in the toolbar menu.

Figure 1. Configure Outliers window

- **Configure Outliers** check box

  Outlier filtering is disabled by default. Select the check box to enable the outlier filtering feature and use the properties that are displayed in the window box. Clear the check box to disable outlier data filtering. When outlier properties are enabled, the **Window size** and the **Percent of Std Deviation** properties are available and allow you to specify the intensity of the search for outliers in the data.

- Window

  Enter an integer in the Window property to define the number of surrounding data points to consider when determining whether a given point is an outlier. For example, if you use the default value of “4”, it will look at the two points before and after the point under investigation (PUI). This is a surrounding range of 4 points from which a standard deviation will be calculated and used with the percentage properties, as described, below.

- Percentage

  Enter a value in this property to specify the percent of standard deviation (calculated from the window of points) to apply for identifying whether or not the PUI should be considered a valid value (not an outlier). If the PUI falls outside of this valid range, then it is considered to be an outlier and its value is replaced by the linear interpolation of the surrounding 2 valid points. If the PUI falls within the range, then the data point is used and considered valid.

Parent topic: About histories

Common history tasks

The following topics describe how to configure history components, as well as how to view and work with histories in the history database.

- **Adding a history extension to a component**
  Extensions come in different types to match the data type of the component and the collection method. You add history extensions by dragging them from the history palette and dropping them onto a component **Property Sheet** or onto the component in the Nav tree.
- **Configuring a history extension**
  When a history extension is first added to a component, by default it is set to be disabled. All that is really necessary to enable data collection is to change the Enable property value to true. Of course, you will want to make other property changes to better configure data collection to suit your actual needs.
- **Configuring rollover properties for delta logging**
  Delta logging records the difference between consecutive time-stamped values instead of recording the actual running total. You can configure rollover properties on a history extension so that rollover occurs when a running total reaches a defined maximum number and then resets to zero or another defined number.
- **Using relative history extension Ords with the HistoryPointList**
  You can use relative paths to point to history extension ords when using a HistoryPointListItem component. This "relativized" format facilitates portability, allowing you to copy the device (with HistoryPointList) to other locations without having to correct the Ords for each new location.
- **Adding a metadata property to a history extension**
  You can add a new property with a Metadata flag to a history extension using the **Slot Sheet** view. You might use metadata property values to identify the location and function of a history's source. History sources may be filtered or organized by the values of the metadata properties. This procedure describes adding a new slot to a history extension for ZipCode data and assigning a metadata flag to the slot.
- **Configuring custom navigation for the history space**
  Using properties in the HistoryGroup component under the History Service, you can configure alternate navigation of your **History** space. You can use one or more HistoryGroup components to set up geographic, functional, or any other desired types of organization of your **History** space to simultaneously display histories in multiple Nav paths. This procedure describes how to create a geographical navigation path by configuring history groups and then configuring history extensions with slots for those history groups.
- **Setting up an alternate navigation tree**
  The HistoryGroup component properties allow you to organize alternate navigation presentations for a station's **History** space.
- **Creating History Nav Shortcuts**
  The default property values for the History Nav Shortcuts component are set to display links to all histories that are under the Nav shortcut ancestor device's name. However, you can configure those properties in the **Property Sheet** view to display only the histories you want.
- **Discovering and matching histories**
  Unless working offline, you can use the learn process to export histories in the station.
- **Editing history data to filter outliers**
  Using the **History Editor** view you can review and modify collected history data for purposes of filtering unusual or "outlier" values. Outliers are known to skew means or averages, so it may be helpful to identify and edit or hide this data.
- **Viewing a component in live mode**
  You can view live data for a component by opening a **Chart** view. The **Chart** view displays a continuing live plot that updates according to the configurable sample rate. If a point has a history extension, the history data is charted and live point values are added as they come in. If the point does not have a history extension, the chart contains only live point values.
- **Viewing security history data**
  You can view any **History** space data as a table of information. A chart is the best way to view most standard historical information. A table is best for security history. This procedure specifically documents viewing security history data.
- **About exporting and importing histories**
  Exporting and importing histories serves several purposes. Archiving is the process of saving one or

more histories to a different location (station), other than where it originated for the purpose of backing up the history. Some applications benefit from exporting histories, processing the data outside of the framework and importing them back in for reporting purposes.

- **Rdb Archive History Provider**
  The Rdb Archive History Provider feature was added starting in Niagara 4.11. It allows queries against local history records to be supplemented by archived history records that have been previously exported to a remote RDBMS using the standard RDBMS drivers (rdbSqlServer, rdbMySQL, and rdbOracle). Because it plugs in at a low level in the framework architecture (for example, the history module), existing views that query histories can benefit from this functionality without any additional changes.
- **Batch history capacity**
  The Archive History Provider feature allows queries against local and archived history records. The archived records come from an external data store, typically a relational database. Once the provider is configured and operational, and history data are available from an external data store, you have the opportunity to update the capacity of locally-stored histories to reduce local storage requirements.

Adding a history extension to a component

Extensions come in different types to match the data type of the component and the collection method. You add history extensions by dragging them from the history palette and dropping them onto a component **Property Sheet** or onto the component in the Nav tree.

- Open history palette

1. Open a component **Property Sheet** view.
2. In the history palette, expand the Extensions folder to find the extension that matches your component data type and collection method.
3. Click and drag the history extension from the palette to the bottom of the **Property Sheet** view.

   Optionally, you can use the right-click menu to copy an extension in the palette and paste it onto the property sheet.

4. In the **Name** window, enter the desired history name (or you can use the default name) and click **OK**.

The extension is added.
Parent topic: Common history tasks

Configuring a history extension

When a history extension is first added to a component, by default it is set to be disabled. All that is really necessary to enable data collection is to change the Enable property value to true. Of course, you will want to make other property changes to better configure data collection to suit your actual needs.

A history extension has been added to a component.

This procedure describes steps to enable the history extension, configure a unique name and active period, as well as the capacity and full policy.

1. Double-click on the history extension on the component to open the **Property Sheet** view.
2. In the Enabled property, click on the drop-down list and click true.
3. In the History Name text property, edit the default name (`%parent.name%`) to give it a unique name. For example, if configuring the name for a history extension on the point, SpaceTemp1, that is under the device, AHU-1. You could add %parent.parent.name% just prior to the default name to end up with this:%parent.parent.name%_%parent.name%, which results in the unique name: `AHU-1_SpaceTemp1`.

Note: You can expand the History Config property to review the read-only Id value resulting from your entry for the History Name property. If you detect an error string here then the History Name property is incorrectly configured.

4. In the Active Period property, if desired click to change the default Days of week and Time Range values.
5. Expand the History Config property, and in the Capacity property, enter the maximum number of history records for the extension.

Note: Setting the capacity is important if you intend to archive history records to another location (station). Capacity is an arbitrary number that you determine based on the expected change frequency. A best practice is to set the capacity to hold two days worth of collected data on the JACE.

6. Also under History Config, confirm that the Full Policy property is set to Roll to prevent the extension from halting data recording when the specified capacity is reached. If not, click the drop-down list and click **Roll**.

Note: The Roll option implements a first-in-first-out method of overwriting the oldest records. Once the specified history capacity is reached, the next record collected overwrites the oldest record.

The history extension is configured and collecting data that is written to records in the **History** space. In a **Chart** view of the history you should see live data plotting and after a period of time historical data will be available as well.
Parent topic: Common history tasks

Configuring rollover properties for delta logging

Delta logging records the difference between consecutive time-stamped values instead of recording the actual running total. You can configure rollover properties on a history extension so that rollover occurs when a running total reaches a defined maximum number and then resets to zero or another defined number.

The station **History** space contains existing histories.

1. Double-click the point extension and expand the **History Config** slot.
2. In the MinRollover slot, click on the Null value check box and then enter the desired minimum value.
3. In the MaxRollover slot, click on the Null value check box and then enter the desired maximum value.
4. Click **Save**.

For example, if you are logging energy consumption with the Max Rollover Value property set to 999,999 and the Min Rollover Value set to 100, when the station detects a rollover, the delta logging bases its delta calculation on a maximum value of 999,999 and a subsequent initial value of 100.
Parent topic: Common history tasks

Using relative history extension Ords with the HistoryPointList

You can use relative paths to point to history extension ords when using a HistoryPointListItem component. This “relativized” format facilitates portability, allowing you to copy the device (with HistoryPointList) to other locations without having to correct the Ords for each new location.

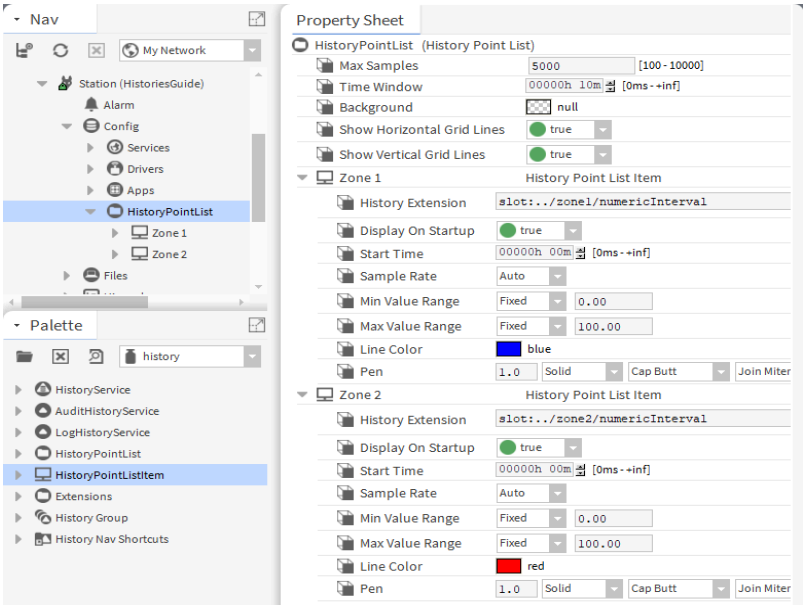
- One or more devices in the station logic.
- One or more points (with history extensions) are under the device(s).

1. Drag a HistoryPointList component from the palette to a device (Air Handling Unit, VAV, etc.).
2. To open the component’s **Property Sheet**, double-click the HistoryPointList component.
3. Drag a HistroyPointListItem component from the palette to the HistoryPointList component and double-

click it to open its **Property Sheet** view.

4. Click on the ORD property and use the **Component Chooser** to select the desired history extension. The absolute Ord for the selected history extension displays. For example, `station:|slot:/Logic/Playground/batchHistories1/ZoneTemp1/NumericInterval`
5. Edit the absolute Ord so that it is relative to the history extension's parent Ord. For example, slot:../ZoneTemp1/NumericInterval
   In this example, the starting point for relativizing the Ord is the HistoryPointList component. Entering ../ has the effect of stepping up one directory in the path to the parent component containing both HistoryPointList and ZoneTemp1
6. Repeat these steps as needed to populate the HistoryPointList.
7. Copy the device to one or two other locations, enable the duplicated history extensions and note the automatic changes to HistroyPointListItem Ords.

The HistoryPointList component takes care of resolving the Ord path for each HistroyPointListItem. The example below shows relative Ords to history extension in HistroyPointListItem.



Parent topic: Common history tasks

Adding a metadata property to a history extension

You can add a new property with a Metadata flag to a history extension using the **Slot Sheet** view. You might use metadata property values to identify the location and function of a history's source. History sources may be filtered or organized by the values of the metadata properties. This procedure describes adding a new slot to a history extension for ZipCode data and assigning a metadata flag to the slot.

The station logic contains history extensions.

1. In the Nav tree, right-click on a history extension and select **Views > SlotSheet**.
2. In the **Slot Sheet** view, right-click and select **Add Slot**.
3. In the **Add Slot** window, enter the Name for the slot (Ex.: ZipCode), click the **Metadata** flag checkbox and click **OK**.
   You can scroll to the bottom of the **Slot Sheet** view to confirm the new slot has been added.
4. Click **Views > Property Sheet**, then scroll to the bottom and enter a value in the new ZipCode property and click **Save**.

You can use the **Metadata Browser** view to add, edit, delete, or view metadata tags. To access the view, right-

click the station Config folder and click **Views > MetadataBrowser**.

Parent topic: Common history tasks

Configuring custom navigation for the history space

Using properties in the HistoryGroup component under the History Service, you can configure alternate navigation of your **History** space. You can use one or more HistoryGroup components to set up geographic, functional, or any other desired types of organization of your **History** space to simultaneously display histories in multiple Nav paths. This procedure describes how to create a geographical navigation path by configuring history groups and then configuring history extensions with slots for those history groups.

- The station has a History Groupings component under the station's **HistoryService**
- Station logic contains points with history extensions already set up.
- Existing histories are already in the station's **History** space.

1. In the Nav tree, expand the station HistoryService and double-click on the History Groupings component.
2. In the **History Group Manager** view, click **New** and in the **New** window click **OK**.
3. In the 2nd **New** window, configure the following properties as shown:
   - Name, enter States
   - History Properties To Group By, enter State in the 1st property, City in the 2nd (as shown here), and click **OK**.

   The new history group named "States" is immediately visible under the History Groupings component in the Nav tree.
4. Under the station logic in the Nav tree, expand a point's history extension, right-click on History Config and open the **Slot Sheet** view.
5. In the History Config **Slot Sheet** view, right-click and select **Add Slot**.
6. In the **Add Slot** window, configure the following properties and click **OK**.
   - Name, enter State
   - Type, select baja:String
7. Add another slot and configure the following properties, and click **OK**:
   - Name, enter City
   - Type, select baja:String
8. Click on the **View** drop-down list and select **Property Sheet** view, then scroll to the bottom and enter values for the newly added State and City properties and click **Save**.
9. Repeat steps 7-8 as needed to add and configure the State and City slots for several history extensions.

   Note: You can configure the History Config properties on multiple points at one time using the **Batch Editor** in the ProgramService.
10. In the Nav tree, right-click on the **History** space and select **Refresh Tree Node**, and expand the **History** space to view the new geographical Nav path.

Note: The default device- and point-centered navigation remains visible under the **History** space named, "Default (All)".

Parent topic: Common history tasks

Setting up an alternate navigation tree

The HistoryGroup component properties allow you to organize alternate navigation presentations for a station's **History** space.

1. To filter by the country, state, and city, History properties that exist in your histories, create three Group properties.

   Note: The order of the History Properties To Group By (in this case: Country, State, and City) determines the history sub-folder ordering.

2. Type in the exact (case-sensitive) property names and click **Save**.
   The corresponding navigation tree is created under the **History** space node.
3. To add or delete additional groupings use the **+** button (to add) and the **x** button (to delete).

Parent topic: Common history tasks

Creating History Nav Shortcuts

The default property values for the History Nav Shortcuts component are set to display links to all histories that are under the Nav shortcut ancestor device's name. However, you can configure those properties in the **Property Sheet** view to display only the histories you want.

- Station logic contains points with history extensions.
- The history palette is open.

1. Open the **Property Sheet** view on a history extension.
2. Drag a History Nav Shortcuts component from the palette to the **Property Sheet**.
3. In the **Property Sheet**, expand the History Nav Shortcut property.
4. Click in the History Query Predicate property and enter a text string for the BQL query predicate value.
   For example, enter the following text string: where state='Georgia' (note the single quotes) to filter for all histories that have a state property with the value set to "Georgia".
5. Click **Save**.

Parent topic: Common history tasks

Discovering and matching histories

Unless working offline, you can use the learn process to export histories in the station.

You are connected to a station.

1. Expand the **Config > Drivers > NiagaraNetwork** right-click the **Histories** node in the Nav tree and click **Views > Niagara History Export Manager**.
   The **Niagara History Export Manager** view displays the histories in the (local) station that are candidates for export.
2. Click the **Discover** button.
   The top **Discovered** pane is a collapsed tree structure of all discovered local histories organized by station name.

   Histories under the same station name as the local station originated in that local station. Histories under any other station nodes represent histories currently imported (or exported) to the local station.

   The **Database** (bottom pane) lists history export descriptors and history system export descriptors currently in the station database.

3. Click to expand, select the histories to export and click **Add**.
   A window opens that allows you to edit properties before the history export descriptor(s) are created in the station.

The **Add** button is available in Learn Mode when you have one or more items selected (highlighted) in the top **Discovered** pane. The **Add** window is identical to the history export descriptor **Edit** window.

4. Edit the properties and click **OK**.
   The station creates history export descriptors under its **Histories** container.

   - Each NiagaraHistoryExport descriptor has a one-to-one associated history, exported to that remote station.
   - Each NiagaraSystemHistoryExport descriptor may, and often does, result in many associated exported histories in the remote station. Note in the Database pane you can spot these export descriptors by the blank History Id value, as well as a text value in the System Tag Patterns column.

5. As necessary, drag the border between the two panes to resize.
6. At any time, to toggle between the two-pane **Learn Mode** and the **single-pane (Database)** view click the Learn Mode tool in the toolbar or use the **Learn Mode** command in the Manager menu.

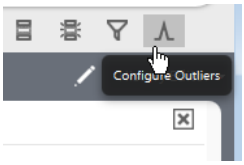
Parent topic: Common history tasks

Editing history data to filter outliers

Using the **History Editor** view you can review and modify collected history data for purposes of filtering unusual or "outlier" values. Outliers are known to skew means or averages, so it may be helpful to identify and edit or hide this data.

The station has existing histories in the station **History** space.

1. In the Nav tree, under the station's **History** space, right-click on a history and select **Views > History Editor**.
2. In the **History Editor** view, select one or more histories and click the **Configure Outliers** icon in the Workbench toolbar (upper right corner), as shown here.



3. In the **Configure Outliers** window, click the check box (far left) to enable outlier filtering, and configure the following values to define the intensity of the search for outliers:
   - Window size — enter an integer to specify the number of surrounding data points to consider when determining whether a given point is an outlier (or use the default value).
   - Percent of Std Deviation — enter a value in this property to specify the percent of standard deviation which is calculated from the window of points (or use the default value).
4. Click **OK**.

If the value of a selected record falls outside of the valid range, then it is considered to be an outlier and its value is replaced by the linear interpolation of the surrounding 2 valid points.
After outliers are configured you can click **Select Outliers** to select all records with outlier values and then either Hide the data (click the toolbar Hide Selected Records icon) or configure filters (click the Configure Filters toolbar icon) to remove the outlier values from the history.
Parent topic: Common history tasks

Viewing a component in live mode

You can view live data for a component by opening a **Chart** view. The **Chart** view displays a continuing live plot

that updates according to the configurable sample rate. If a point has a history extension, the history data is charted and live point values are added as they come in. If the point does not have a history extension, the chart contains only live point values.

You are connected to a station that contains points.

You can open a **Chart** view on the following types of components to view a continuing live plot:

- a Boolean, Enum or Numeric point (with or without a history extension)
- a record in the **History** space
- a schedule
- a Chart widget on a Px page
- a chart file that points to any of the above items

Note: The system cannot accommodate an unlimited amount of points in a webChart.

---

1. Expand the station home and right-click on a Boolean, Enum, or Numeric point.
2. Select **Views > Chart**

The **Chart** view opens and begins plotting live data in the chart according to the component's configuration.
Parent topic: Common history tasks

Viewing security history data

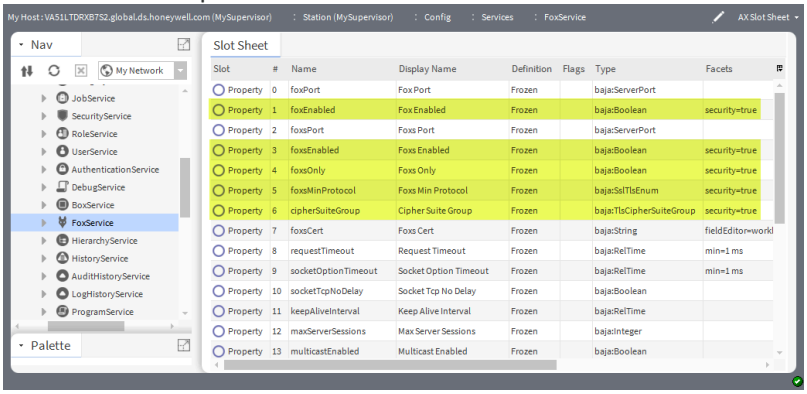
You can view any **History** space data as a table of information. A chart is the best way to view most standard historical information. A table is best for security history. This procedure specifically documents viewing security history data.

The **History** space contains one or more security histories.

1. Expand **History > local** (station name) and double-click **SecurityHistory**.
   The history table opens.

   This screen capture shows mostly authentication history records. The last row reports a **FoxService** property change.

2. To test that the log is working, navigate to a component, such as the FoxService or WebService, right-click the component and click **Views > Slot Sheet**.
   The **Slot Sheet** opens.



The highlighted properties are security-related as identified by `security=true` in the Facets column of the table. Changing one of these properties generates an audit record in the security history.

3. Note which properties are security-related, return to the **Property Sheet**, change the configuration of the property you noted and click **Save**.
4. Go back and refresh the history.
   The station updates the history indicating that the property was `Changed`.

Security-related property changes show up in the standard audit history, but authentication events do not. This prevents frequent authentication events from filling up the limited number of records in the audit history. To change the number of records allowed in any audit history, use the **AuditHistoryService Property Sheet**.
Parent topic: Common history tasks

About exporting and importing histories

Exporting and importing histories serves several purposes. Archiving is the process of saving one or more histories to a different location (station), other than where it originated for the purpose of backing up the history. Some applications benefit from exporting histories, processing the data outside of the framework and importing them back in for reporting purposes.

Export (push)

Exporting histories pushes data from a source station to a target station or to an external file.

Exported histories from remote stations may create histories in the Supervisor that have an unreachable source. If the source station that contains the history is unreachable, and you modify any history configuration properties using the histories **Property Sheet** under the Supervisor's **NiagaraNetwork**, the next export from the remote station may overwrite the changed properties.

If you added metadata to a history extension, an export of a history from a remote source station does not overwrite or drop metadata values with the history in a Supervisor station unless the metadata property has the same name in both stations.

If you are exporting histories, it is better to add system tags at the remote station so that they are exported up to the Supervisor station.

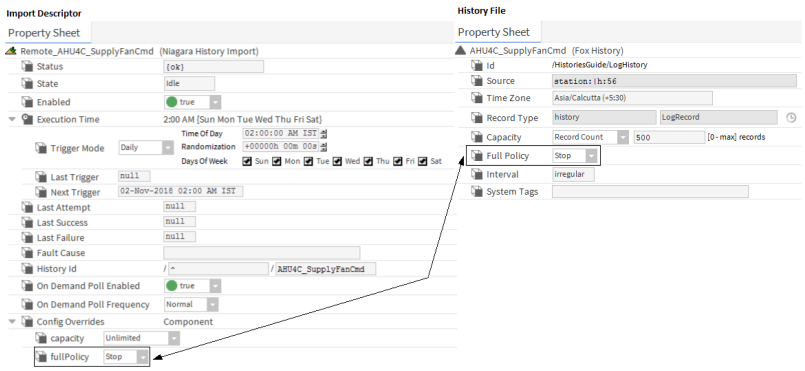
Exporting data to an external file and importing it back in is also usually done by a Supervisor station. Enhancing the data and importing it back may improve reporting, which is almost always the function of a Supervisor.

Import (pull)

Importing pulls data from the source station or from a file (such as a CSV file) to a target station. Usually a Supervisor station imports histories from one or more remote stations for the purpose of archiving the data. This direction of data flow works best because you have more control over the timing of a data transfer when importing from the Supervisor than is possible when exporting from a remote station.

An import descriptor in the Supervisor station configures properties for each remote source history. Changing the Full Policy on the history **Property Sheet** view in the remote station, changes the same property on that history's import descriptor under the Config Overrides property in the Supervisor station.

Figure 1. Full Policy property in an import descriptor and remote history component



On the left is the import descriptor in the Supervisor station for the remote history. On the right is the history file's **Property Sheet** in the remote station.

Local histories behave in the same way since local they and their history import descriptors should always be available to the Supervisor station.

- **Setting up NiagaraNetwork history policy**
  On a regular basis, the Supervisor station collects history data from its subordinate stations for archival purposes and to create reports. This procedure sets up history policy properties on the **NiagaraNetwork** in preparation to import and export histories.
- **Discovering histories to import**
  Importing histories (pulling data) starts by creating a history import descriptor usually in a Supervisor station. The actual import takes place at the time scheduled in the import descriptor.
- **Discovering histories to export**
  Exporting histories (pushing data) starts by creating a history export descriptor in the source station. The actual export takes place at the time scheduled in the export descriptor.
- **Manually setting up a single history descriptor**
  Both import and export use history descriptors. While history discovery is available from both the **Niagara History Import Manager** and **Niagara History Export Manager**, you may create a descriptor manually. A descriptor may apply to a single history or, using tags, you can set up a descriptor that applies to all histories in the station.
- **Setting up to transfer many histories at once**
  System tags provide an alternate way to import and export histories from remote stations. Special system history descriptors make this possible. Instead of using the manager's learn mode (discovery) to add history descriptors, you manually add new descriptors using the **Niagara History Import Manager** or **Niagara History Export Manager**.

Parent topic: Common history tasks

Setting up NiagaraNetwork history policy

On a regular basis, the Supervisor station collects history data from its subordinate stations for archival purposes and to create reports. This procedure sets up history policy properties on the **NiagaraNetwork** in preparation to import and export histories.

- The **NiagaraNetwork** is configured with at least two stations, such as a Supervisor target (receiving) station and a remote (sending) controller station.
- A client connection is established between the two stations.
- The source station contains histories to be exported or imported.
- Workbench is connected to the Supervisor (target) station.

1. To identify the stations from which to import and the names of the history files, expand **Config > Drivers**, right-click **NiagaraNetwork** and click **Views > AX Property Sheet**
   This exposes the **NiagaraNetwork** configuration properties.
2. Click **History Policies** or expand **History Policies**.
   The **History Network Ext** (extension) opens.
3. Click the **Default Rule** or expand **Default Rule**.
   The **Config Rule Property Sheet** opens.

   This is the primary rule for identifying the histories to export from each remote station.

4. Configure the Device Pattern based on your device naming convention.
   This sets up in the Supervisor station the remote devices with history extensions. You may use the asterisk (*) as a wildcard or enter a series of device tags each delimited by a semicolon (;).
5. Configure the History Name Pattern based on your history naming convention.
   This identifies the names of the histories in the remote station(s). You may use the asterisk (*) as a wildcard or enter a series of history extension tags each delimited by a semicolon (;).
6. Select a capacity value and, if you choose Record Count, enter the maximum number of records to save.
   Unlimited assumes that your Supervisor PC can store large quantities of history data.

   Record Count limits the number of records to store.

   Which option to choose depends on your unique situation and how frequently each remote station collects device and event data.

7. If you selected Record Count for capacity, choose a fullPolicy option and click **Save**.
   You do not need to choose a fullPolicy option if capacity is unlimited.

   Roll deletes the oldest records and replaces them with the newest.

   Stop saves exported records until it reaches the total Record Count after which it stops saving records.

8. To create another configuration rule, right-click the **Default Rule**, click **Duplicate**, give the new rule a name, click **OK**, expand the new rule and configure the four properties.
   You need more than one rule to apply different device and history name patterns.

   After setting up configuration rules it is time to create import descriptors.

Parent topic: About exporting and importing histories

Discovering histories to import

Importing histories (pulling data) starts by creating a history import descriptor usually in a Supervisor station. The actual import takes place at the time scheduled in the import descriptor.

- The **NiagaraNetwork** is configured with at least two stations, such as a Supervisor target (receiving) station and a remote (source) controller station.
- A client connection is established between the two stations.
- The source station contains histories to be imported.
- Workbench is connected to the target station.

1. In the Nav tree of the target station to which you intend to import the history data, expand **Config > Drivers**.
2. Double-click **NiagaraNetwork**.
   The **Station Manager** opens.

3. To discover the histories that need import descriptors, click **Discover**.
   The **Discovered** table in the upper portion the window lists discovered station names.
4. Double-click the import extension icon ( ) for the source station.
   The **Niagara History Import Manager** opens with all of the histories stored in the **History** space of the selected station.
5. Select the history descriptors to import and click **Add**.
   The **Add** window opens.

   This window sets up each history descriptor.

6. To set up when to import the histories, configure the Execution Time properties as desired and click **OK**.
   The **Database** table in the lower portion of the view displays the imported history(ies).

   The imported histories reside under the **NiagaraNetwork** of the target station, in the **Histories** space of the station name representing the source station, as shown here.

Parent topic: About exporting and importing histories

Discovering histories to export

Exporting histories (pushing data) starts by creating a history export descriptor in the source station. The actual export takes place at the time scheduled in the export descriptor.

- The **NiagaraNetwork** is configured with at least two stations, such as a target Supervisor station and a source remote controller station.
- A client connection is established between the two stations.
- The source station contains histories to be exported.
- Workbench is connected to the remote source station.

1. In the Nav tree of the source station (pushing data), expand **Config > Drivers > NiagaraNetwork**.
2. Expand the target **NiagaraStation**, right-click the **Histories** node and select **Views > Niagara History Export Manager**.
3. In this view, click **Discover**.
   The **Discovered** table in the upper portion the window lists histories in the station that you can export.
4. Select the history(ies) to export and click **Add**.
   The **Add** window opens.
5. Configure the Execution Time properties as desired and click **OK**.
   These properties configure when the source station exports the selected history(ies) to the target station.

The export descriptors display in the **Database** table in the lower portion of the view. Exported histories are located under the **NiagaraNetwork** of the source station, in the **Histories** space of the station name representing the target station.

Parent topic: About exporting and importing histories

Manually setting up a single history descriptor

Both import and export use history descriptors. While history discovery is available from both the **Niagara History Import Manager** and **Niagara History Export Manager**, you may create a descriptor manually. A descriptor may apply to a single history or, using tags, you can set up a descriptor that applies to all histories in the station.

You are using Workbench connected to a Supervisor station.

1. Expand **Config > Drivers > NiagaraNetwork > NiagaraStation** and do one of the following:
   - To create an import history descriptor, double-click **Histories**.
   - To create an export history descriptor, right-click **Histories** and click **Views > Niagara History Export Manager**.
2. The manager view opens.
3. To create the descriptor, click **New**.
   The first **New** window opens.
4. To create a descriptor that applies to a single history, select Niagara History Import or Niagara History Export based on the type of descriptor and click **OK**
   The second **New** window opens.
5. To create a single descriptor for a single history file, configure the Name of the history and its History Id (station and file path) and click **OK**.
   This type of descriptor does not use the System Tag Patterns. Instead, you configure a single History Id. Once created, you can add system tags by editing this descriptor.

   Once set up, the data transfer occurs when scheduled by the Execution Time properties.

6. To manually import or export a history, right-click its import or export descriptor and click **Actions > Execute**.

Parent topic: About exporting and importing histories

Setting up to transfer many histories at once

System tags provide an alternate way to import and export histories from remote stations. Special system history descriptors make this possible. Instead of using the manager's learn mode (discovery) to add history descriptors, you manually add new descriptors using the **Niagara History Import Manager** or **Niagara History Export Manager**.

You are working in Workbench and are connected to your Supervisor station. Remote history extensions (for histories to be imported) are configured with system tags.

1. Expand **Config > Drivers > NiagaraNetwork > NiagaraStation** and double-click **Histories**
2. Click **New**.
   The **New** window opens.
3. To create a descriptor that applies to all the histories in the station, select Niagara System History Import or Niagara System History Export from the drop-down Type to Add menu and click **OK**.
   A second **New** window opens.

   The screen capture is for a system history import descriptor. An export descriptor is similar with fewer properties. This type of descriptor does not use the History Id. Instead, it uses the System Tag Patterns to configure which histories to transfer.

4. To create a single descriptor for the entire station, enter as many system tags as needed in the System Tag Patterns property.

   For example, the System Tag Patterns property in the screen capture configures the import job to include histories from history extensions with system tags that begin with the mnemonic: AHU, or that include these tags: Ltg1, Ltg3, or Ltg5. This configuration transfers histories from history extensions tagged with: AHU1, AHU_1, AHU2, and so on, as well as those tagged with Ltg1, Ltg3 and Ltg5. It excludes those tagged with: Ahu1, Ltg2, and Ltg 1.

   Other properties, such as Execution Time, Capacity, Full Policy, and On Demand Poll Enabled apply to the associated histories, and operate the same as for regular descriptors.

   Once set up, the data transfer occurs when scheduled by the Execution Time properties.
5. To manually import or export multiple histories, right-click its import or export descriptor and click **Actions > Execute**.

Parent topic: About exporting and importing histories

Rdb Archive History Provider

The Rdb Archive History Provider feature was added starting in Niagara 4.11. It allows queries against local history records to be supplemented by archived history records that have been previously exported to a remote RDBMS using the standard RDBMS drivers (rdbSqlServer, rdbMySQL, and rdbOracle). Because it plugs in at a low level in the framework architecture (for example, the history module), existing views that query histories can benefit from this functionality without any additional changes.

This feature can allow you to seamlessly compare baseline years of historical trends with new data using a WebChart.

The standard RDBMS drivers (rdbSqlServer, rdbMySQL, and rdbOracle) can export local history records periodically to a remote database. The Rdb Archive History Provider plugs in to the station's **HistoryService**, making it possible for any existing views that query histories to benefit from both local and archived records.

While the station still stores local histories, once the driver exports those history records to a relational database, you can reduce the capacity of those local histories to free up resources in your Supervisor station. At history query time, the Rdb Archive History Provider can easily retrieve those exported (older) archived history records residing in an Oracle, SQL server or MySQL database.

License update prerequisite

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

- A general historyArchive license feature that covers any archive history provider implementation
- A specific rdbHistoryArchive license feature that covers your chosen RDB using the Rdb Archive History Provider against any supported RDBMS.

- **Chart example, local data**
  The history being charted in this example has not been configured yet to pull data from an Rdb Archive History Provider.
- **Setting up an Rdb Archive History Provider**
  For history query purposes, an archive history provider pulls archive data into a station from a remote database on-demand (it does not persist the retrieved archive data locally, but only uses the data in the displayed query results).
- **PX example, local and archive data**
  All views that use history queries can benefit from the archive history provider.

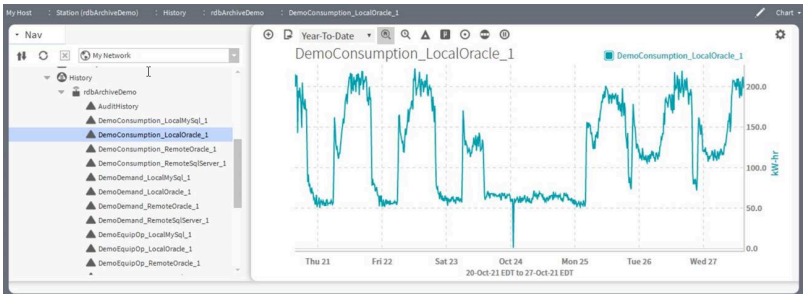
Parent topic: Common history tasks

Chart example, local data

The history being charted in this example has not been configured yet to pull data from an Rdb Archive History Provider.

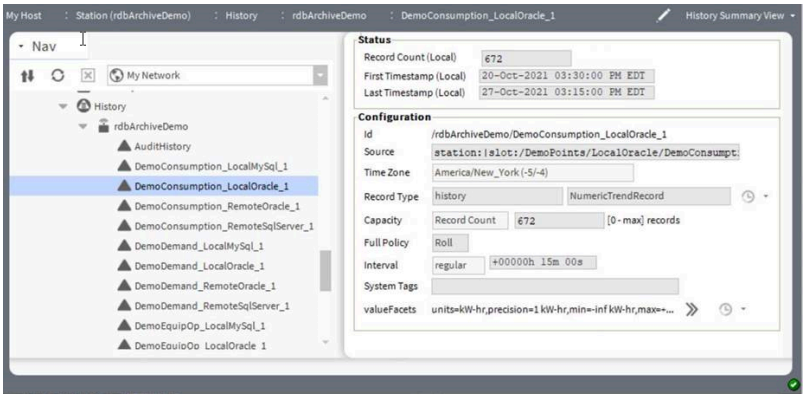
This history has only a week's worth of local history data to display, even though the time range is configured for year-to-date. More records could be available from an archive.

Figure 1. Local data



The **History Summary View** confirms the local history data is confined to a 672 record capacity (rolling). To access this view, click the drop-down list in the upper right corner of the chart.

Figure 2. Local History Summary View



The local data consists of only 672 records, basically a week's worth of 15-minute interval data.

Parent topic: Rdb Archive History Provider

Setting up an Rdb Archive History Provider

For history query purposes, an archive history provider pulls archive data into a station from a remote database on-demand (it does not persist the retrieved archive data locally, but only uses the data in the displayed query results).

You are working in Workbench connected a Supervisor station.

1. Open the rdb palette and expand the **HistoryService Archive Provider** folder.
2. In the station, expand **Config > Services > HistoryService**.
   The **HistoryService** contains an **Archive History Providers** container.
3. From the palette, add a **RdbArchiveHistoryProvider** to the **Archive History Providers** container under **HistoryService** and double-click the provider you just added.
   The component's **Property Sheet** opens.

   This **Property Sheet** configures the Rdb Archive History Provider.

4. Use the button to the right of the Ord to Rdbms property to open the **Component Chooser**, locate the relational database in your driver network and click **Save**.
5. Configure Max Archive Results Per Query and Archive Limit Notifications if needed.
   Max Archive Results Per Query determines the maximum number of history records to read from the RDBMS for any history time range query that taps into it. If more history records are available beyond this limit at history query time, the Archive Limit Notifications property defines the behavior of a subset

of Workbench views, but not all of them. Web Chart and HTML5 History Table views (accessible from the browser and Workbench) provide their own notification when a history query exceeds this limit. When the limit is reached for a query, in addition to the warning, you get truncated archive history results that always consider the most recent history records first.

The Archive Limit Notifications specifies what happens when a history query made from a Workbench user connected to the station exceeds the Max Archive Results Per Query limit.

Note: This setting does not apply to HTML5 history views, including HTML5 views accessed within Workbench, such as the Web Chart view. It only applies to native Workbench views that perform history queries, such as the AX History Chart or AX History Table views.

---

6. To complete the configuration, click **Save**

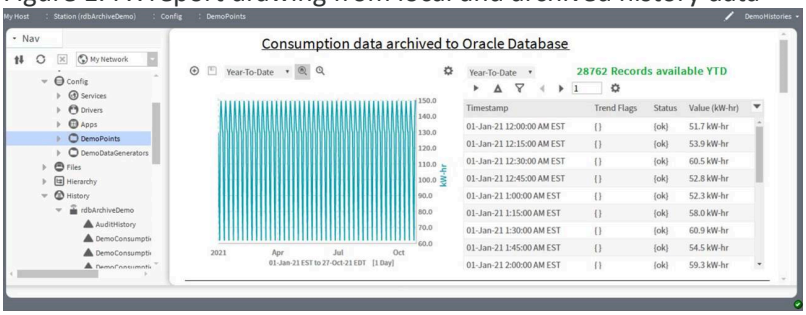
Parent topic: Rdb Archive History Provider

PX example, local and archive data

All views that use history queries can benefit from the archive history provider.

The following PX graphic looks at history data that was pulled from both local and archived sources using an archive provider for an Oracle database. You can configure MySQL and SQL server databases accordingly. All that is required is to set up the Rdb Archive History Providers to reference those databases as well.

Figure 1. PX report drawing from local and archived history data



To debug queries, you can turn on the "rdb.archiveHistoryProvider" logger to FINE level.

Parent topic: Rdb Archive History Provider

Batch history capacity

The Archive History Provider feature allows queries against local and archived history records. The archived records come from an external data store, typically a relational database. Once the provider is configured and operational, and history data are available from an external data store, you have the opportunity to update the capacity of locally-stored histories to reduce local storage requirements.

Local histories are faster to query than archived histories, so you should consider a local history capacity setting that balances your local storage requirements with your common history query time ranges. You should also choose a local history capacity that is acceptable even on occasions when the archive data source is temporarily unavailable (for example, when the remote archive data source is down for maintenance).

Changing each capacity property individually could be a tedious process. This section documents how to use existing tools to make updating the capacity properties as easy as possible.

A locally-stored history can be:

- A local history residing on the station
- A remote history imported to the local station
- A remote history exported to the local station

Each of these requires a different set of steps, which are documented in the three task topics that follow.

- **Updating the capacity property of multiple local histories**
  A station's storage capacity for multiple local histories is limited. To reduce local storage requirements, this procedure uses the **Batch Editor** of the **ProgramService** to configure at one time how many history records a station can store.
- **Updating the capacity property of multiple imported histories**
  A way to include multiple archived histories in the local station is to import them as a batch from the remote database. This procedure uses the **Batch Editor** to configure how many archived histories to import from a remote database.
- **Updating the capacity of remote exported histories**
  A way to include multiple archived histories in the local station is to export them as a batch from the remote database. Configuring the Default Rule, a property of the stations History Policies, controls the number of archived history records that a remote database can export to a station.

Parent topic: Common history tasks

Updating the capacity property of multiple local histories

A station's storage capacity for multiple local histories is limited. To reduce local storage requirements, this procedure uses the **Batch Editor** of the **ProgramService** to configure at one time how many history records a station can store.

You are connected to a station. The Archive History Provider is configured and operational and history data are available from multiple local data stores.

1. Expand **Station > Config > Services**, and double-click **ProgramService**.
   The **Batch Editor** opens.
2. To locate the histories, click **Find Objects**.
   The **Bql Query Builder** opens.
3. For Of Type, select **(Custom Type)** from the drop-down list.
   Of Type moves to left end and two more drop-down lists appear.
4. Select history and HistoryConfig from the drop-down lists and click **OK**.
   The **Batch Editor** displays the history files it found.The screen capture shows two audit history files and a single log history file.
5. To configure the number of records to store for a specific history, select the history row and click the **Edit Slot** button at the bottom of **Batch Editor**.
   The **Edit Slot** window opens.
6. In the **New Value** pane, fill in the number of records and click **OK**.
   The **Batch Editor Results** opens along with the capacity values for each history.

Parent topic: Batch history capacity

Updating the capacity property of multiple imported histories

A way to include multiple archived histories in the local station is to import them as a batch from the remote database. This procedure uses the **Batch Editor** to configure how many archived histories to import from a remote database.

You are connected to a remote station that is ready to receive (import) history data. The Archive History Provider is configured and operational.

1. Expand **Station > Config > Services**, and double-click **ProgramService**.
   The **Batch Editor** opens.
2. To locate the histories to configure, click **Find Objects**.
   The **Bql Query Builder** opens.
3. Click the search icon ( ) beside the In property.
   The **Choose Root** window opens.
4. Select **NiagaraNetwork** and click **OK**.
   The **Bql Query Builder** selects the **NiagaraNetwork**.
5. To add a search criterion, click the add icon ( ) on the **Match** bar, change the drop-down list to name, type configOverrides and click **OK**.
   The **Batch Editor** displays the files it found based on your search criteria.
6. To configure the number of records to store for a specific history, select the history row and click the**EditSlot** button at the bottom of the **Batch Editor**.
   The **Edit Slot** window opens.
7. In the **New Value** pane, fill in the number of records and click **OK**.
   The **Batch Editor Results** window opens along with the new values for capacity.

Parent topic: Batch history capacity

Updating the capacity of remote exported histories

A way to include multiple archived histories in the local station is to export them as a batch from the remote database. Configuring the Default Rule, a property of the stations History Policies, controls the number of archived history records that a remote database can export to a station.

You are connected to a remote station that is ready to receive the exported history data. The Archive History Provider is configured and operational.

1. Expand **Station > Config > Drivers**, right-click **Niagara Network** and click **Views > AX Property Sheet**.
   The **Property Sheet** opens.

   History Policies has a Default Rule (you can add additional config rules) and each config rule has a capacity property.

2. In the capacity property, fill in the number of history records and click **Save**.
   The updates to the capacity are effective from the next time the remote station exports histories.

Parent topic: Batch history capacity

History components

Components include services, folders and other model building blocks associated with a module. You may drag them to a Property or Wire Sheet from a palette.

Descriptions included in the following topics appear as context-sensitive help topics when accessed by:

- Right-clicking on the object and selecting **Views > Guide Help**
- Clicking **Help > Guide On Target**

- **History Property Sheets**
  These views display standard **Property Sheets** for each history. In addition to the default history configuration properties, any slots that are added using the **slot sheet** view or the **metadata browser** display in these views.
- **Audit History Service (history-AuditHistoryService)**
  When enabled, this service registers itself as the auditor for the system at system startup and monitors regular and security-related events separately creating a record for each user-initiated change to each component in the station.
- **history-AuditRecord**
  The AuditRecord keeps a history of changes made by users. If enabled, it registers itself as the Auditor for the system when the service is started.
- **history-ConfigRule**
  This component determines the overrides for an existing history configuration. Its functionality is provided by the history palette.
- **history-ConfigRules**
  This container for rules determines the configuration of histories that are pushed to the local device. The station applies configuration rules when it creates a history. Changing a rule has no effect on existing histories. The ConfigRules functionality is provided by the history palette.
- **history-FoxHistory**
  This component is the implementation of BIHistory that works with the FoxHistorySpace.
- **history-FoxHistorySpace**
  FoxHistorySpace provides access to a History database using the fox protocol.
- **history-HistoryConfig**
  This component configures a history in the History database.
- **history-HistoryDevice**
  HistoryDevice represents a source device for histories.
- **history-HistoryEditorOptions**
  The HistoryEditorOptions stores the options used to configure history options.
- **history-HistoryId**
  The HistoryId component is a container for History id.
- **history-HistoryGroup**
  This component organizes alternate navigation for a station's **History** space. Use the properties in this component to specify metadata properties for grouping histories. Add the HistoryGroup component to the HistoryGroupings container-component by dragging and dropping it from the history palette or by clicking the **New** button in the **History Group Manager** view.
- **history-HistoryPointList**
  This component is a container that holds HistoryPointListItem components. The default view of this component is the **Live History Chart** view. The **Property Sheet** view is where you configure properties for the HistoryPointList chart display.
- **history-HistoryPointListItem**
  Each HistoryPointListItem links to a single history extension and is configured in the **Add Item** window or in the HistoryPointListItem **Property Sheet**. This component is available in the history module.
- **history-HistoryService**
  Each station contains a single **HistoryService**. This service provides http access to all of the histories in a station and is responsible for creating the history database, as well as enabling the collection and storage of histories in the database.

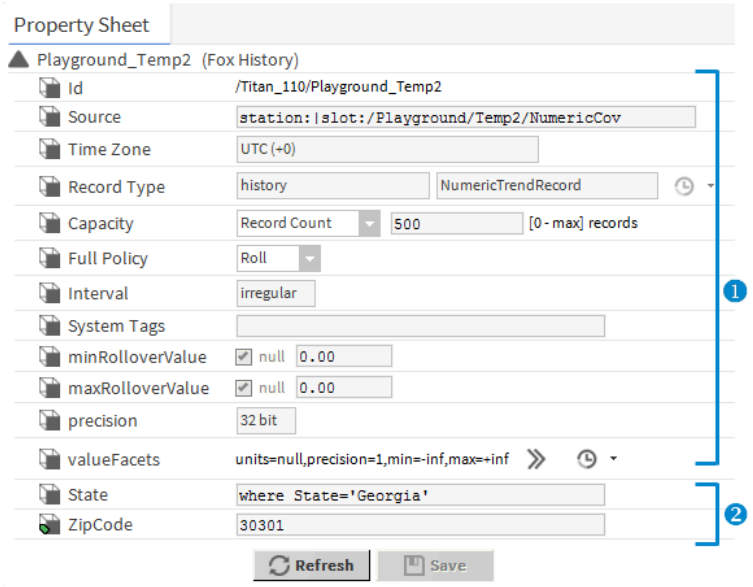
- **history-HistoryShortcuts**
  History Nav shortcuts provide convenient navigation links to histories. You can place these shortcuts anywhere in a station to provide a filtered list of individual history shortcuts. The default property values display links to all histories that are under the Nav shortcut ancestor device's name. You edit properties in the **Property Sheet** view to display only the histories you want. History Nav shortcuts have a **Nav Container** view and a **History Chart Builder** view.
- **history-IntervalAlgorithm**
  This component logs a value periodically at a fixed interval. The IntervalAlgorithm is available in the history palette Extensions directory under some Interval extensions as Collector.
- **history-LocalDatabaseConfig**
  This component configures a LocalHistoryDatabase.
- **history-LogHistoryService**
  When enabled, this service maintains a buffered history (LogHistory) of some of the messages seen in the station's standard output. When troubleshooting, this log provides a history of recent error messages.
- **History extensions**
  Extensions come in different types to match the data type of the component and the collection method desired. You can add an extension to a component by dragging the extension onto your **Property Sheet**or onto your Nav side bar pane from the history palette.

History Property Sheets

These views display standard **Property Sheets** for each history. In addition to the default history configuration properties, any slots that are added using the **slot sheet** view or the **metadata browser** display in these views.

To add properties to a history (and designate them as metadata, if desired), use the **Slot Sheet** view or the **Metadata Browser** view. An example of a history **Property Sheet** view is shown here.

Figure 1. Example of a history Property Sheet view



1	Default properties
2	Added Dynamic properties

If the source is accessible, using a **Property Sheet** to change any of the properties that can be edited, changes the properties at the actual source of the history. For example, if you change a history Full Policy property from

Stop to Roll, and if the source is accessible, the history Full Policy property changes on the appropriate history extension or History Import component (Config Overrides property).

Note: If the history is imported from a NiagaraHistoryImport descriptor or NiagaraSystemHistoryImport descriptor that was made via export tags (HistoryImportTag, SystemHistoryImportTag) in the source station, the next export tag Join to or from that station overwrites whatever Full Policy or other property value changes initiated from a history **Property Sheet**.

Parent topic: History components

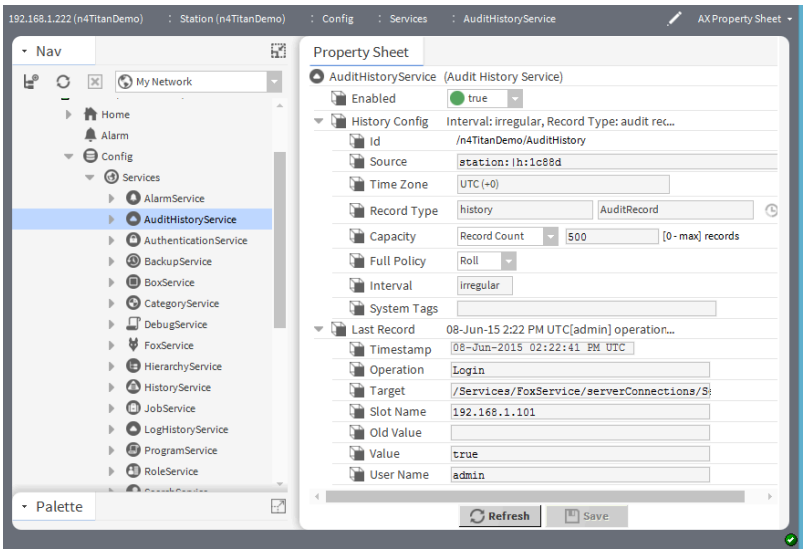
Audit History Service (history-AuditHistoryService)

When enabled, this service registers itself as the auditor for the system at system startup and monitors regular and security-related events separately creating a record for each user-initiated change to each component in the station.

The station commissioning process installs and enables this service by default.

Regular events include station configuration changes, such as adding and deleting users. Security events include each time someone logs in and out, and each time someone changes component properties that have been identified as security-related properties. Separating security-related audit records from regular audit records emphasizes the importance of monitoring security and ensures that the audit history file maintains a manageable size. For example, frequent user log-in and -out events may quickly fill a history with unnecessary records. Recording authentication events in a security history prevents a regular audit history from filling up too quickly.

Figure 1. Audit History Service properties



To open this **Property Sheet**, expand **Config > Services** and double-click on the AuditHistoryService in the Nav tree.

The component is designed to audit all property modifications and all action invocations. These events are subject to audit:

- Property changed
- Property added

- Property removed
- Property renamed
- Property reordered
- Action invoked

History Config properties

These properties configure the audit function. A separate set under the heading Security Audit History Source applies specifically to security-related events, such as authentication and changes to security-related properties.

Property	Value	Description
Enabled	true or false	Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.).
HistoryConfig	additional properties	Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties.
Id	read-only	Displays string results from a value configured in history extension's History Name property. An error string here indicates the History Name property is incorrectly configured.
Time Zone	read-only	Displays the time zone of the active history extension.
Record Type	read-only	Displays the data that the record holds in terms of: extension type (`history`) and data type (`BooleanTrendRecord`, `NumericTrendRecord`, and so on).
Capacity	Record Count: nnn (500 default), Unlimited	Specifies the number of trend log records (histories) to store in the histories database. When capacity is reached, newer records overwrite the oldest records.
Full Policy	Roll (default), Stop	Specifies what happens when a trend log (history) reaches capacity. Applies only if Capacity is set to Record Count. When capacity reaches record count, the newest records overwrite the oldest records. Roll ensures that the latest data are recorded. Stop terminates recording when the number of stored records reaches capacity. Full policy has no effect if Capacity is

Property	Value	Description
		Unlimited.
Interval	read-only	Defines, for Interval-based data collection, the cycle time, or how often the history properties are checked. Changing this property creates a new history (or splits-off) from the original history because histories with different intervals are not compatible.
System Tags	Text	Assigns additional metadata (the System Tag) to a history extension. This identifier is then available for selective import or export of histories using the **Niagara System History Import** or **Niagara System History Export** option (using the System Tag Patterns). Each System Tag is separated by a semicolon. For example: NorthAmerica;Region1;Cities.
Last Record	read-only	Stores a copy of the most recent record successfully appended to the history for this extension.

Last Record properties

Property	Value	Description
Timestamp	read-only	Reports when the event occurred.
Operation	read-only	Identifies the type of event.
Target	read-only	Reports the modified Ord.
Slot Name	read-only	Identifies the host IP address.
Old Value	read-only	Reports the value before the change.
Value	read-only	Reports the new value.
User Name	read-only	Identifies the person who made the change.

Parent topic: History components

history-AuditRecord

The AuditRecord keeps a history of changes made by users. If enabled, it registers itself as the Auditor for the system when the service is started.

The AuditRecord is available in the History palette under the HistoryService component.

Parent topic: History components

history-ConfigRule

This component determines the overrides for an existing history configuration. Its functionality is provided by the history palette.

Each **Config Rule** under the **NiagaraNetwork**'s **History Policies** has the following configuration properties:

Property	Value	Description
Device Pattern	(defaults to a wildcard (*))	Defines a string that matches the name of one or more stations from which to export histories. The string matches all station names.
History Name Pattern	(defaults to a wildcard (*))	Defines a string that matches the names of histories being exported. This means it matches all named histories. Note: Both Device Pattern and History Name Pattern must apply for the rule to be used—otherwise the next rule down (in order) in History Policies is evaluated.
Capacity	Record Count: nnn (500 default), Unlimited	Specifies the number of trend log records (histories) to store in the histories database. When capacity is reached, newer records overwrite the oldest records.
Full policy	Roll (default), Stop	Specifies what happens when a trend log (history) reaches capacity. Applies only if Capacity is set to Record Count. When capacity reaches record count, the newest records overwrite the oldest records. Roll ensures that the latest data are recorded. Stop terminates recording when the number of stored records reaches capacity. Full policy has no effect if Capacity is Unlimited.

Parent topic: History components

history-ConfigRules

This container for rules determines the configuration of histories that are pushed to the local device. The station applies configuration rules when it creates a history. Changing a rule has no effect on existing histories. The ConfigRules functionality is provided by the history palette.

When a station exports a history to another station, it evaluates the history Config Rules to set up the local (archived) history's config properties: Capacity and Full Policy. It uses the first matching rule. The Default Rule is always at the top and cannot be deleted or renamed.

Note: Rule priority is set by order: the Default Rule is always first, it is highest priority. If you create additional

rules in Workbench (right-click a rule, then click **Duplicate**), you can edit, rename, and reorder rules as needed.

Parent topic: History components

history-FoxHistory

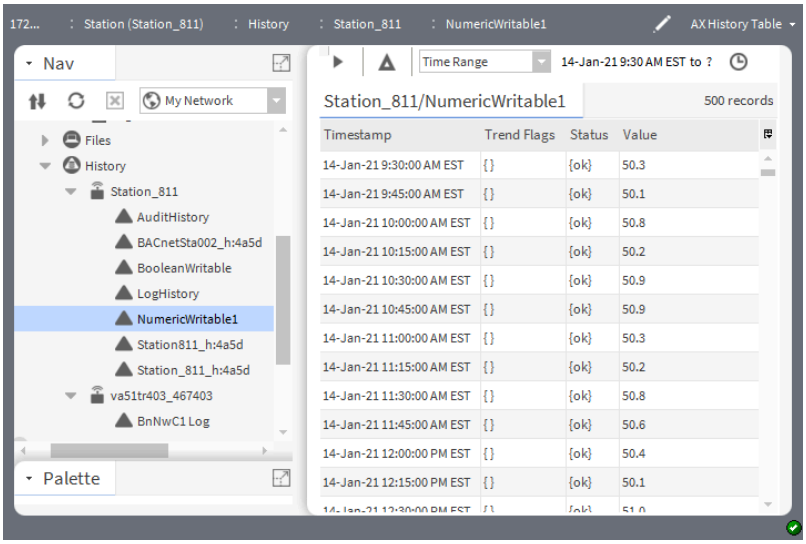
This component is the implementation of BIHistory that works with the FoxHistorySpace.

The primary views for this component are the Chart and History Chart views, which are documented in the *Plugins* chapter of this guide.

AX History Table and History Table

These views list the records in the selected history.

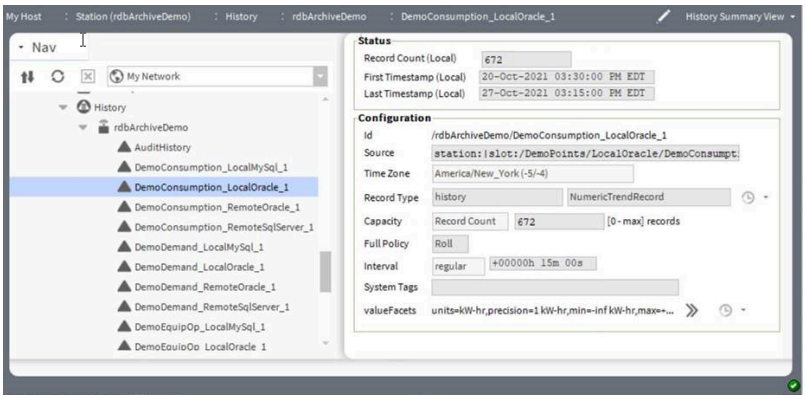
Figure 1. Example of an AX History Table



History Summary View

This view summarizes the attributes of each individual history.

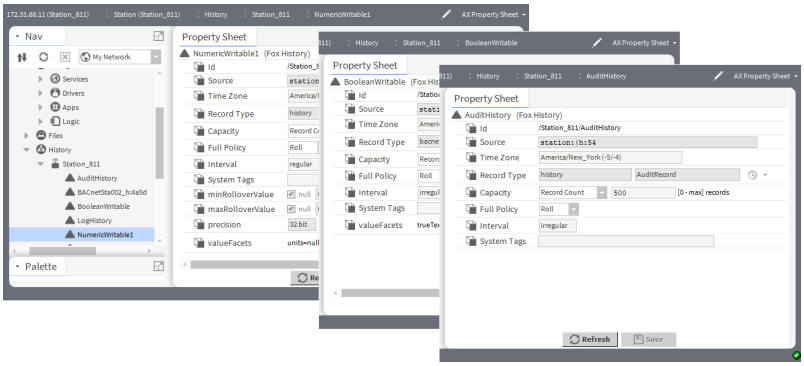
Figure 2. Example of a History Summary View



Property Sheets

You can use this component's **AX Property Sheet**s to review and edit properties for individual histories.

Figure 3. Examples of history properties



To access this view, expand **Config > History**, expand the station, right-click a history and click **Views > AX Property Sheet**

Property	Value	Description
Id	read-only	Displays string results from a value configured in history extension's History Name property. An error string here indicates the History Name property is incorrectly configured.
Source	read-only	Displays the ORD of the active history extension.
Time Zone	read-only	Displays the time zone of the active history extension.
Record Type	read-only	Displays the data that the record holds in terms of: extension type (`history`) and data type (`BooleanTrendRecord`, `NumericTrendRecord`, and so on).
Capacity	Record Count: nnn (500 default), Unlimited	Specifies the number of trend log records (histories) to store in the histories database. When capacity is reached, newer records overwrite the oldest records.
Full Policy	Roll (default), Stop	Specifies what happens when a trend log (history) reaches capacity. Applies only if Capacity is set to Record Count. When capacity reaches record count, the newest records overwrite the oldest records. Roll ensures that the latest data are recorded. Stop terminates recording when the number of stored records reaches capacity.

Property	Value	Description
		Full policy has no effect if Capacity is Unlimited.
Interval	read-only	Defines, for Interval-based data collection, the cycle time, or how often the history properties are checked. Changing this property creates a new history (or splits-off) from the original history because histories with different intervals are not compatible.
System Tags	read-only	Assigns additional metadata (the System Tag) to a history extension. This identifier is then available for selective import or export of histories using the **Niagara System History Import** or **Niagara System History Export** option (using the System Tag Patterns). Each System Tag is separated by a semicolon. For example: NorthAmerica;Region1;Cities.

Unique properties

Poperty	Value	Description
minRolloverValue	read-only	Reports the starting point for calculations for cumulative logging after a running total maximum value is reached.
maxRolloverValue	read-only	Reports the maximum value for calculations when the software detects a rollover by the history logging process.
precision	read-only	Reports the number of decimal places for limiting units.
valueFacets	additional properties	Configures how data are represented.

Parent topic: History components

history-FoxHistorySpace

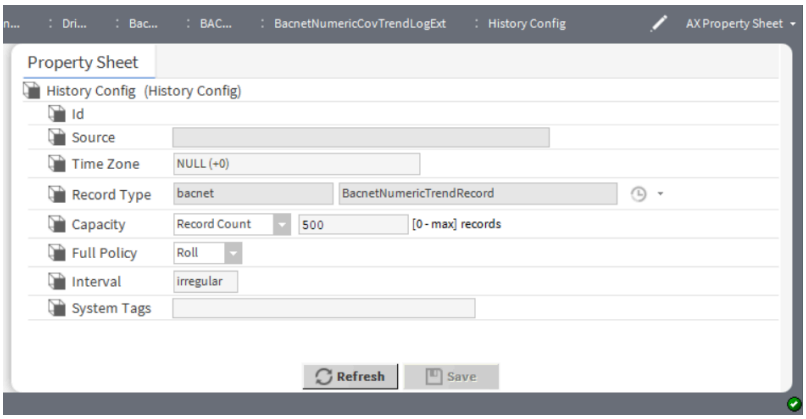
FoxHistorySpace provides access to a History database using the fox protocol.

Parent topic: History components

history-HistoryConfig

This component configures a history in the History database.

Figure 1. History Config properties



The **History Config** component is available in the history palette under **LogHistoryService**. The following table summarizes the History Config properties.

Property	Value	Description
History Config	additional properties	Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties.
Id	read-only	Displays string results from a value configured in history extension's History Name property. An error string here indicates the History Name property is incorrectly configured.
Source	ORD	Displays the ORD of the active history extension.
Time Zone	read-only	Displays the time zone of the active history extension.
Record Type	read-only	Displays the data that the record holds in terms of: extension type (`history`) and data type (`BooleanTrendRecord`, `NumericTrendRecord`, and so on).
Capacity	Record Count: nnn (500 default), Unlimited	Specifies the number of trend log records (histories) to store in the histories database. When capacity is reached, newer records overwrite the oldest records.
Full Policy	Roll (default), Stop	Specifies what happens when a trend log (history) reaches capacity. Applies only if Capacity is set to Record Count. When capacity reaches record count, the newest records overwrite the oldest records. Roll ensures that the latest data are recorded. Stop terminates recording when the

Property	Value	Description
		number of stored records reaches capacity. Full policy has no effect if Capacity is Unlimited.
Interval	read-only	Defines, for Interval-based data collection, the cycle time, or how often the history properties are checked. Changing this property creates a new history (or splits-off) from the original history because histories with different intervals are not compatible.
System Tags	read-only	Assigns additional metadata (the System Tag) to a history extension. This identifier is then available for selective import or export of histories using the **Niagara System History Import** or **Niagara System History Export** option (using the System Tag Patterns). Each System Tag is separated by a semicolon. For example: NorthAmerica;Region1;Cities.

Parent topic: History components

history-HistoryDevice

HistoryDevice represents a source device for histories.

Parent topic: History components

history-HistoryEditorOptions

The HistoryEditorOptions stores the options used to configure history options.

These are stored under C:\Users\userName\Niagara 4.0\brandName\etc\options\history-OutlierParams.options.

Where userName is a name on your computer and softwareVern.n is the version of Niagara.

Parent topic: History components

history-HistoryId

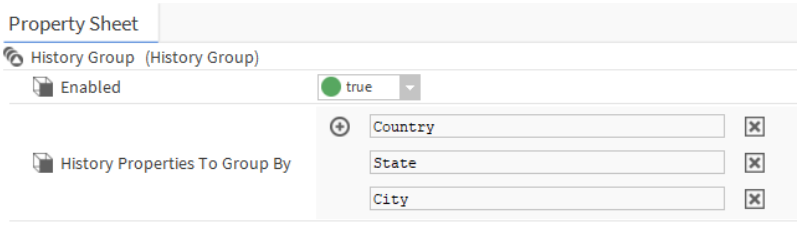
The HistoryId component is a container for History id.

Parent topic: History components

history-HistoryGroup

This component organizes alternate navigation for a station's **History** space. Use the properties in this component to specify metadata properties for grouping histories. Add the HistoryGroup component to the HistoryGroupings container-component by dragging and dropping it from the history palette or by clicking the **New** button in the **History Group Manager** view.

Figure 1. HistoryGroup property sheet



Name	Value	Description
Enabled	true or false	Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.).
History Properties To Group By		Provides properties that you can use to organize/filter the display of histories. A semicolon separates Individual properties. Each completed property represents a specific property name that must match exactly to a history property name that belongs to one or more histories. The order of the History properties to group by determines the sub folder ordering in the **History** space.

Parent topic: History components

history-HistoryPointList

This component is a container that holds HistoryPointListItem components. The default view of this component is the **Live History Chart** view. The **Property Sheet** view is where you configure properties for the HistoryPointList chart display.

In addition to the **HistoryPointList** properties, the **Property Sheet** view displays all associated HistoryPointListItem components below the properties. If you invoke this view from the HistoryPointList component the items in the list can be selected/unselected and the view will update as needed.

The HistoryPointList component properties include the following;

Type	Value	Description
Max Samples	5000 (default)	Defines the maximum number of samples to store for each history extension displayed.
Time Window	00000h 10m (default)	Defines the time range to display on the Time Axis while viewing the live chart. If the chart is "zoomed" you will be able to pan and view all the collected data defined by Max Samples.
Background	color chooser (defaults to Null)	Specifies the background fill color. Solid opens the **Color Chooser** window. Gradient opens the **Gradient Editor**

Type	Value	Description
		window. Image opens the **Image Brush Editor** window. Click the Browse icon () to open the File Chooser, Ord Chooser, or other method of selecting an image file. Null indicates no color (white).
Show horizontal grid lines	true (default), false	Shows (true) or hides (false) horizontal grid lines on the chart.
Show vertical grid lines	true (default), false	Shows (true) or hides (false) horizontal grid lines on the chart.

Parent topic: History components

history-HistoryPointListItem

Each HistoryPointListItem links to a single history extension and is configured in the **Add Item** window or in the HistoryPointListItem **Property Sheet**. This component is available in the history module.

Properties you can configure include the following.

Name	Value	Description
History Extension	Ord	Defines the Ord to the history extension of the control point you wish to view. This MUST be the history extension and NOT the control point or its corresponding History component.
Display on Startup	true (default), false	Indicates whether or not to automatically display the chart for this item when the view first comes up.
Start Time	00000h 00m	Indicates how much of the collected history data you wish to view. The time that is entered in this property will be subtracted from the current time to get the initial history data. Note: The following conditions apply: If no data is found within the given Start Time the LAST collected value is retrieved for the item. If the Start Time is 0 hours 0 minutes ALL collected history values is retrieved for the item.
Sample Rate	Auto, Fixed: 00h 00m 00sCov	Defines the rate at which to sample the live data. Auto follows the interval defined in the history extension configuration. Fixed follows a defined interval that is specified by the user in terms of hours,

Name	Value	Description
		minutes, and seconds. COV configures updates to plot when there is a change of value.
Minimum Value Range	Auto, Fixed 0.00 (default)	Serves as the minimum value to display on the value axis. This only applies to numeric points (Booleans and Enums do not use this property). Auto: uses the minimum value from the data collected. Fixed: uses the value specified by the use
Maximum Value Range	Auto, Fixed 100.00 (default)	Serves as the maximum value to display on the value axis. This only applies to numeric points (Booleans and Enums do not use this property). Auto: uses the maximum value from the data collected. Fixed uses the value specified by the user. Note: If Min Value Range and Max Value Range are Fixed AND have the same values charts may share the value axis. If these two conditions do not exist, additional value axes may be added to the chart, as needed.
Line Color	Null (default)	Specifies the desired color for the plot line. Click on the property editor to invoke the Color Chooser.
Pen	Line weight: 1.0 (default)Line type: Solid (default), Dotted, DashedCap type: Cap Butt (default), Cap Round, Cap SquareCorner type: Join Miter (default), Join Bevel, Join Round	Specifies the line-weight and type to use for the plot line.

Parent topic: History components

history-HistoryService

Each station contains a single **HistoryService**. This service provides http access to all of the histories in a station and is responsible for creating the history database, as well as enabling the collection and storage of histories in the database.

To use histories, the **HistoryService** must be installed on the station. If you do not have the service in your active station, you can add it by dragging and dropping a copy from the history palette, to the **Services** node in the Nav tree or to the bottom of the **Property Sheet** view of the **Services** node, as shown here.

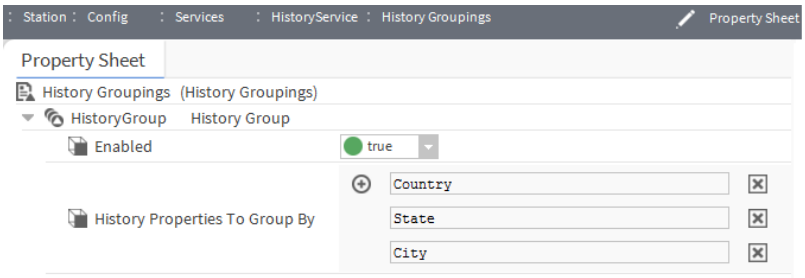
This service manages histories in ways that may not always be visible to you. The **HistoryService**:

- Identifies all existing histories when a station starts and adds them to its list.
- Handles history life cycle management, such as the creation and deletion of individual histories.
- Maintains the naming convention (namespace) for all histories.

- Maintains a global default configuration for the histories.

The **History Extension Manager** and the **HistoryService Property Sheet** provide two views for working with history extensions.

Figure 1. History service properties



To access these properties, expand **Config > Services**, right-click **HistoryService**, click **Views > AX Property Sheet**, and expand **History Groupings**.

Property	Value	Description
History Groupings, History Properties To Group By	three properties for Country, State, City	Configures geographical locations for grouping histories.

Actions

History service actions are available from the popup menu when you right-click the HistoryService node in the Nav tree.

- **Save Db** initiates a save of all histories to the history database.
- **Close Unused Histories** closes any histories that have not been accessed within the Max Open Time (this time is set in the **HistoryService Property Sheet**).

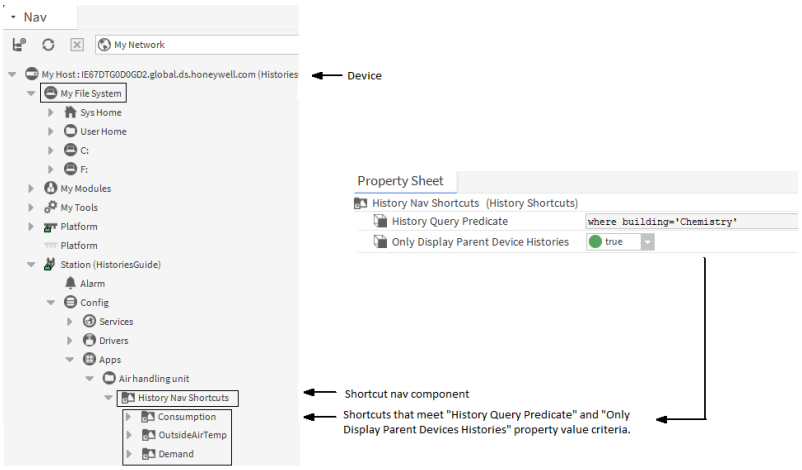
Parent topic: History components

history-HistoryShortcuts

History Nav shortcuts provide convenient navigation links to histories. You can place these shortcuts anywhere in a station to provide a filtered list of individual history shortcuts. The default property values display links to all histories that are under the Nav shortcut ancestor device's name. You edit properties in the **Property Sheet** view to display only the histories you want. History Nav shortcuts have a **Nav Container** view and a **History Chart Builder** view.

Note: The history shortcuts are not actual histories but links that provide convenient access to a history when you double-click on the shortcut. You cannot drag history shortcuts onto a Px page and relativize them.

Figure 1. Example History Nav Shortcut property values



The history shortcut has the following properties:

Type	Value	Description
History Query Predicate	Text string	Filters the histories by entering a value as a BQL query predicate value. For example, you could enter the following text string: `where state='Georgia'` (note the single quotes) to filter for all histories that have a state property with the value set to "Georgia". This property works in conjunction with the Only Display Parent Device Histories property.
Only Display Parent Device Histories	true (default), false	Restricts the history shortcut filtering to those histories that match the ancestor device of the HistoryNavShortcut component. If set to true, only shortcuts to histories that are identified with the history shortcuts parent device name are displayed. If set to false, shortcuts to all valid (those that match the "BQL query predicate value") are displayed. In cases where the parent device name is not a valid history device name, you can add a dynamic "historyDeviceName" property. This property works in conjunction with the History Query Predicate property.
historyDeviceName	Text string	Adds a historyDeviceName string property to the **HistoryNavShortcuts Property Sheet** using the History Nav Shortcuts component **Slot Sheet** view. This property filters for an alternate history device name (in case the HistoryNavShortcuts parent device's name is not a valid history device). For example, if you place a HistoryNavShortcut under a device that has a name such as `SqlServerDatabase` and a history ID

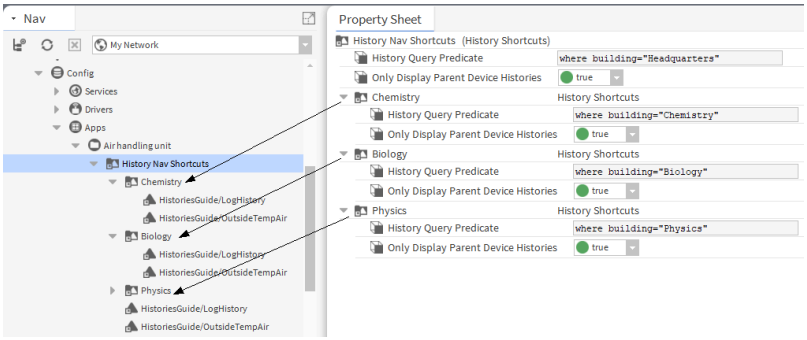
Type	Value	Description
		from that RDBMS device named /myStation/currentTemp, then a shortcut for this history is not displayed when the Only Display Parent Device Histories is set to true. However, if you add a HistoryDeviceName property to the component and set a value of myStation for the added property, a shortcut for that history displays under the History Nav Shortcuts component.

About nesting history shortcuts

History Nav Shortcuts can be nested in order to build hierarchies of shortcut navigation. In the following illustration, note the following points:

- The History Query Predicate property filters to show the shortcuts in the Nav tree. Since the Only Display Parent Device Histories value is set to true and the shortcut is not a child of any of the station devices (Station1, Station2, etc.) these shortcuts include histories that have a building property set to a value of Headquarters and are included anywhere in the **History** space. If the shortcut is placed under Station1 with these same property settings, then only the Station1 Consumption, OutsideAirTemp, and Demand histories would display.

  Figure 2. Example nested History Nav Shortcuts



- The History Nav Shortcuts named Chemistry shows all shortcuts to histories with a building property set to Chemistry.
- The History Nav Shortcuts named Biology shows shortcuts to histories with a building property set to Biology. Since the Only Display Parent Device Histories value is set to true and the shortcut does not have a parent device these are histories that have a Biology property set to a value of Headquarters and are included anywhere in the **History** space.

Parent topic: History components

history-IntervalAlgorithm

This component logs a value periodically at a fixed interval. The IntervalAlgorithm is available in the history palette Extensions directory under some Interval extensions as Collector.

Parent topic: History components

history-LocalDatabaseConfig

This component configures a LocalHistoryDatabase.

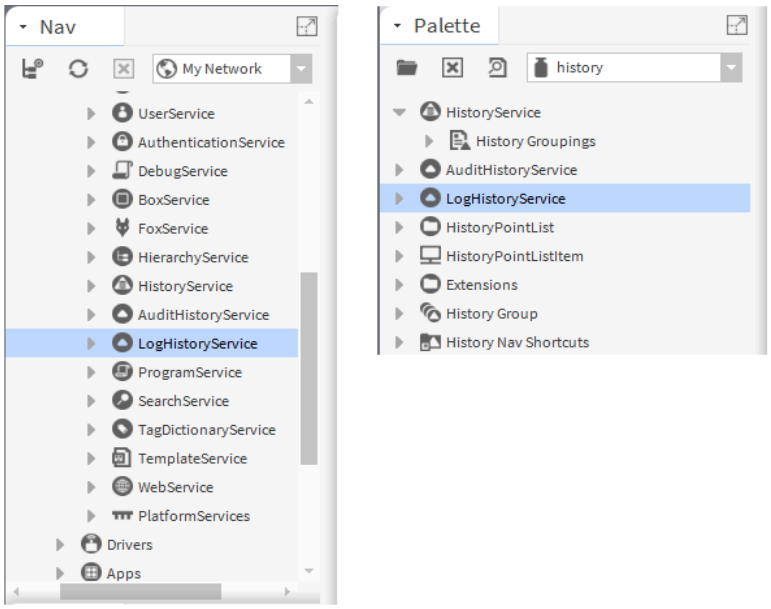
Parent topic: History components

history-LogHistoryService

When enabled, this service maintains a buffered history (LogHistory) of some of the messages seen in the station's standard output. When troubleshooting, this log provides a history of recent error messages.

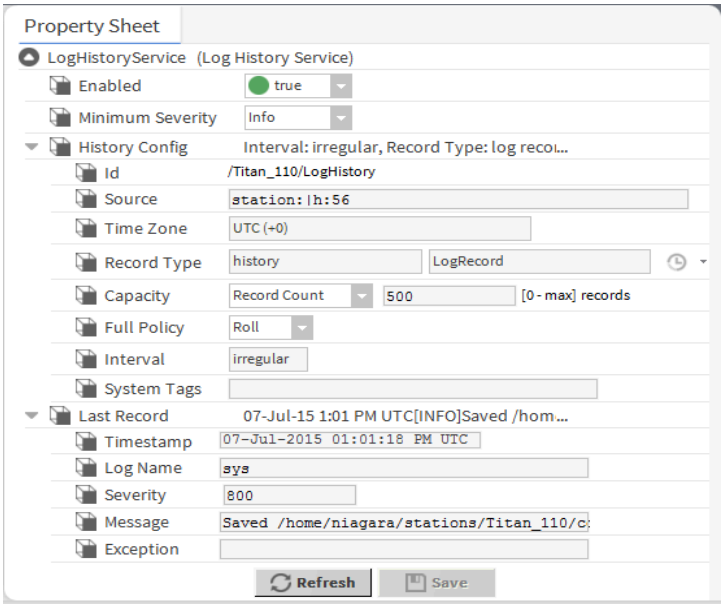
The LogHistoryService is available in the history palette. If it is not already present, drag it from the history palette to the **Services** node in the Nav tree, as shown here.

Figure 1. Adding the LogHistoryService



You can edit LogHistoryService properties and set configuration properties in the **Property Sheet**, as shown here.

Figure 2. Log history service properties



To open this **Property Sheet** expand **Config > Services** and double-click LogHistoryService.

Properties

Property	Value	Description
Enabled	true or false	Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.).
Minimum Severity	Error (default), Warning., Trace, or Message	Sets the lowest-level station output message that you want to log. Choose the level of output that you want to record.
History Config	additional properties	Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties.
Id	Text string	Displays string results from a value configured in history extension's History Name property. An error string here indicates the History Name property is incorrectly configured.
Time Zone	display or drop-down list	Displays the time zone of the active history extension.
Record Type	read-only	Displays the data that the record holds in terms of: extension type (`history`) and data type (`BooleanTrendRecord`, `NumericTrendRecord`, and so on).
Capacity	Record Count: nnn (500 default),	Specifies local storage capacity for histories. In general, 500 (default record

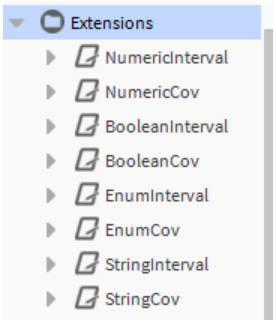
Property	Value	Description
	Unlimited	count) or less is adequate for a controller station because those records are usually archived (exported) to a Supervisor station. For this reason, a very large number, such as 250,000 is acceptable for Supervisor stations. Unlimited is not the wisest choice even for a Supervisor station.
Full Policy	drop-down list	Specifies what happens when a trend log (history) reaches capacity. Applies only if Capacity is set to Record Count. When capacity reaches record count, the newest records overwrite the oldest records. Roll ensures that the latest data are recorded. Stop terminates recording when the number of stored records reaches capacity. Full policy has no effect if Capacity is Unlimited.
Interval	read-only	Defines, for Interval-based data collection, the cycle time, or how often the history properties are checked. Changing this property creates a new history (or splits-off) from the original history because histories with different intervals are not compatible.
System Tags	text	Assigns additional metadata (the System Tag) to a history extension. This identifier is then available for selective import or export of histories using the **Niagara System History Import** or **Niagara System History Export** option (using the System Tag Patterns). Each System Tag is separated by a semicolon. For example: NorthAmerica;Region1;Cities.
Last Record	read-only	Stores a copy of the most recent record successfully appended to the history for this extension.

Parent topic: History components

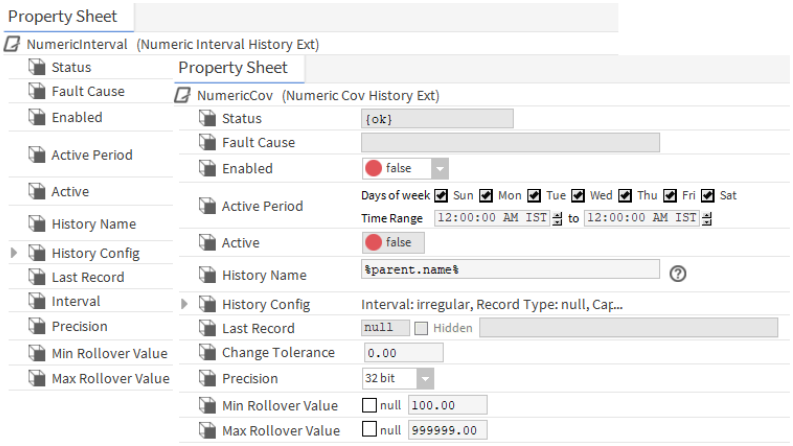
History extensions

Extensions come in different types to match the data type of the component and the collection method desired. You can add an extension to a component by dragging the extension onto your **Property Sheet**or onto your Nav side bar pane from the history palette.

Figure 1. History extensions in Workbench history palette



The following properties apply to history extensions.
Figure 2. History extensions in the Property Sheet view



Property	Value	Description
Status	read-only	Reports the condition of the entity or process at last polling. {ok} indicates that the component is licensed and polling successfully. {down} indicates that the last check was unsuccessful, perhaps because of an incorrect property, or possibly loss of network connection. {disabled} indicates that the Enable property is set to false. {fault} indicates another problem. Refer to Fault Cause for more information.
FaultCause	read-only	Indicates the reason why a system object (network, device, component, extension, etc.) is not working (in fault). This property is empty unless a fault exists.
Enabled	true or false	Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.).
Active Period	Days of the week: All (default), Time range: `12:00:00 AM` to `12:00:00 AM`	Specifies, using check boxes, the days of the week as well as start and end time

Property	Value	Description
	(default)	`hours:minutes:seconds` for data collection.
Active	read-only	Indicates whether or not the data collected are active (as defined by the Active Period property.
History Name	A history format default string, `%parent.name%` (default)	Uses a formatting convention to consistently name histories using a standardized formatting pattern. The format `%parent.name%` is the default history name format. This string automatically names any histories with the name of the parent component and appends a sequential number to additional names, as necessary.
HistoryConfig	additional properties	Serves as a container for sub-properties used to configure the attributes of the history record stored in the History space. A separate topic documents these properties.
Last Record	read-only	Stores a copy of the most recent record successfully appended to the history for this extension.
Change Tolerance	text	Specifies the minimum change in value that must occur before the system writes a record to the database.
Min Rollover Value	number (defaults to null)	Defines a maximum value for calculations when a rollover is detected by the history logging process. Using this parameter and the Min Rollover Value parameter helps you avoid getting negative numbers when you are logging total data, such as energy usage.
Max Rollover Value	number (defaults to null)	Defines a maximum value for calculations when a rollover is detected by the history logging process. Using this parameter and the Min Rollover Value parameter helps you avoid getting negative numbers when you are logging total data, such as energy usage.

- **BooleanChangeOfValue**
  BooleanCovHistoryExt is a history extension that is used for recording on change of value for boolean point data. This extension is available in the history palette Extensions directory.
- **BooleanInterval**
  BooleanIntervalHistoryExt is a history extension that is used for recording on intervals for boolean point data. This extension is available in the history palette Extensions directory.
- **NumericInterval**
  NumericIntervalHistoryExt is a history extension used to record on intervals for Numeric point data. This extension is available in the history palette Extensions directory.
- **EnumChangeOfValue**

EnumCovHistoryExt is a history extension used to record on a change of value for Enum point data. This extension is available in the history palette Extensions directory.

- **EnumInterval**
  EnumIntervalHistoryExt is a history extension used to record on intervals for Enum point data. This extension is available in the history palette Extensions directory.
- **StringChangeOfValue**
  StringCovHistoryExt is a history extension used for collecting a string control value on change of value. This extension is available in the history palette Extensions directory.
- **StringInterval**
  StringIntervalHistoryExt s a history extension for collecting a string control value at intervals. This extension is available in the history palette Extensions directory.
- **CovAlgorithm**
  CovAlgorithm determines when to log a point's value according to change of value. The CovAlgorithm is available in the history palette Extensions directory under some Change Of Value extensions as Collector.

Parent topic: History components

BooleanChangeOfValue

BooleanCovHistoryExt is a history extension that is used for recording on change of value for boolean point data. This extension is available in the history palette Extensions directory.

Parent topic: History extensions

BooleanInterval

BooleanIntervalHistoryExt is a history extension that is used for recording on intervals for boolean point data. This extension is available in the history palette Extensions directory.

Parent topic: History extensions

NumericInterval

NumericIntervalHistoryExt is a history extension used to record on intervals for Numeric point data. This extension is available in the history palette Extensions directory.

Parent topic: History extensions

EnumChangeOfValue

EnumCovHistoryExt is a history extension used to record on a change of value for Enum point data. This extension is available in the history palette Extensions directory.

Parent topic: History extensions

EnumInterval

EnumIntervalHistoryExt is a history extension used to record on intervals for Enum point data. This extension is available in the history palette Extensions directory.

Parent topic: History extensions

StringChangeOfValue

StringCovHistoryExt is a history extension used for collecting a string control value on change of value. This extension is available in the history palette Extensions directory.

Parent topic: History extensions

StringInterval

StringIntervalHistoryExt s a history extension for collecting a string control value at intervals. This extension is available in the history palette Extensions directory.

Parent topic: History extensions

CovAlgorithm

CovAlgorithm determines when to log a point's value according to change of value. The CovAlgorithm is available in the history palette Extensions directory under some Change Of Value extensions as Collector.

Parent topic: History extensions

History plugins

Plugins provide views of components. There are many ways to view plugins (views). One way is directly in the Nav tree. In addition, you can right-click on an item and select one of its views.

In Workbench, access the following summary descriptions on any plugin by selecting **Help > On View** (F1) from the menu, or pressing F1 while the view is open.

- **workbench-WebChart**
  This view plots historical data, live historical data, and live data, as well as schedules. It is the default history view for history records in Workbench and in Hx, and a secondary view on schedules and Enum, Numeric, and Boolean points. Legacy charts, those created in earlier releases, are available as secondary **History Chart** views on history records.
- **Collection Table view**
  This view shows a table of any data (in this case history data) that you can export and view in the following formats: PDF, CSV, Text.
- **Database Maintenance view**
  This view is one of the views available on the **History** space node in the Nav tree. Using this view, you can clear records and delete complete histories from your history database.
- **Device Histories View**
  This view is available on any device that supports import and export of histories (for example, BACnet devices, and others). The view shows a filtered list of history shortcuts for the particular device.
- **History Chart Builder view**
  This view is the default view of the **History** space node (in the Nav tree). As one of the available views, you can build any one of several types of chart options from the data that are stored in one or more histories. This view displays in Workbench, in all Workbench (Wb) Web profiles, and in the Default Hx, and Basic Hx profiles as well.
- **History Chart view**
  This view plots the historical data of the selected history log along X and Y axes. With this view you can display and configure charts of history records. Charts created in an earlier release are available as secondary **History Chart** views on history records.
- **History Editor view**
  This view edits data and filter histories, including batch editing (selecting multiple rows and using the right-click menu or the **Edit** menu to make changes).
- **History Extension Manager view**
  As the default view of the History Service, the **History Extension Manager** provides a tabular view of all history extensions in the station and displays information about the control point, extension type, name and the current status.
- **History Group Manager view**
  The default view of the HistoryGroup component provides a tabular view of all history groups. It is the default view of the HistoryGroupings component and provides information about the groups contained in the HistoryGroup component.
- **History Group Ux Manager view**
  This is the default view for adding groups to the station using the **HistoryGroupManager** view.
- **History Slot Sheet view**
  This view displays a standard slot sheet for a history. It is available on all histories to add slots and designate them as metadata slots, if desired. To designate slots as metadata, you assign a metadata flag to a slot.
- **History Summary view**
  This view displays read-only values for each history record's status and configuration properties.
- **History Table view**
  This view shows a table of history data that you can export and view in the following formats: PDF, CSV, Text.
- **Live History Chart view**
  This view displays historical data (trend data) as the default view for the HistoryPointList component and a secondary view on history extensions. It combines the historical plot of the History Chart with a continuing live plot that updates according to a sample rate that can be configured. It begins with a

start time to the current time and continuously updates as the source generates new data.

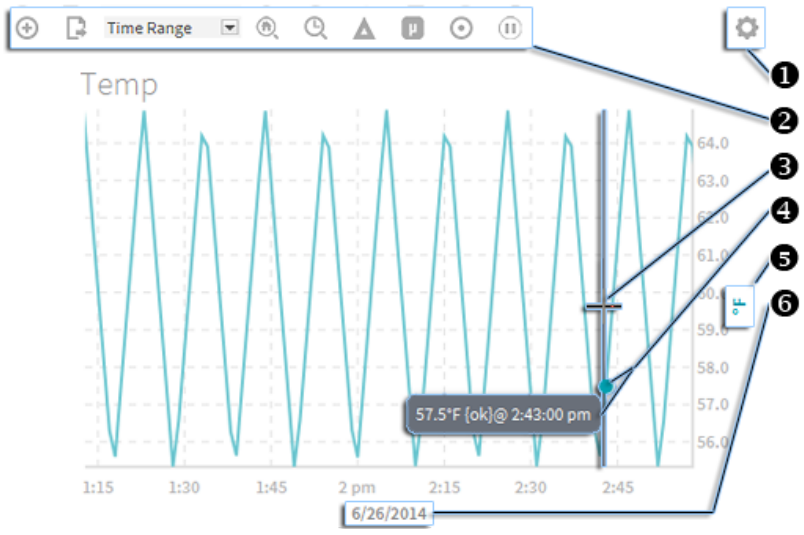
- **Nav Container view**
  This view displays all histories in the station. It is one of the views available on the **History** space node in the Nav tree. It displays a row for each station that is represented in the **History** space. You can double-click on any of the station icons to display the individual histories that are children to the station that is represented in the Name column of the view. Double click on a history in this view to display the history in the **Chart** view.
- **Niagara History Export Manager view**
  This view is available only on a station's **Histories** extension (apart from database device components in some RdbmsNetwork drivers) of the **NiagaraNetwork** (niagaraDriver).
- **Niagara History Import Manager view**
  This is the default view on a station's **Histories** extension.
- **Metadata Browser view**
  This view is available for working exclusively with metadata properties on components and histories. The interface provides a convenient way to add metadata slots to one or more components or histories. Although you can add metadata slots directly to a component or a history from the slot view, the **Metadata Browser** has the ability to add metadata slots to thousands of objects in one batch job.
- **On Demand History view**
  The On Demand history feature enables polling of both local history sources and Niagara History imports for live data when displaying history chart or history table views. This feature works in addition to, and does not replace, a standard polling schedule. For example, you would typically still have your History Import descriptors scheduled to archive at some daily interval (such as every night), even though you might be displaying **On Demand** history views.

workbench-WebChart

This view plots historical data, live historical data, and live data, as well as schedules. It is the default history view for history records in Workbench and in Hx, and a secondary view on schedules and Enum, Numeric, and Boolean points. Legacy charts, those created in earlier releases, are available as secondary **History Chart** views on history records.

In Niagara 4 the **Chart** view is the default view for histories. While the **History Chart** view is a secondary view for legacy charts created an earlier release. Although the two views have a different look and feel, both offer many of the same controls and options.

Figure 1. Chart view description



- ❶ Settings icon — click to access chart **Settings** window

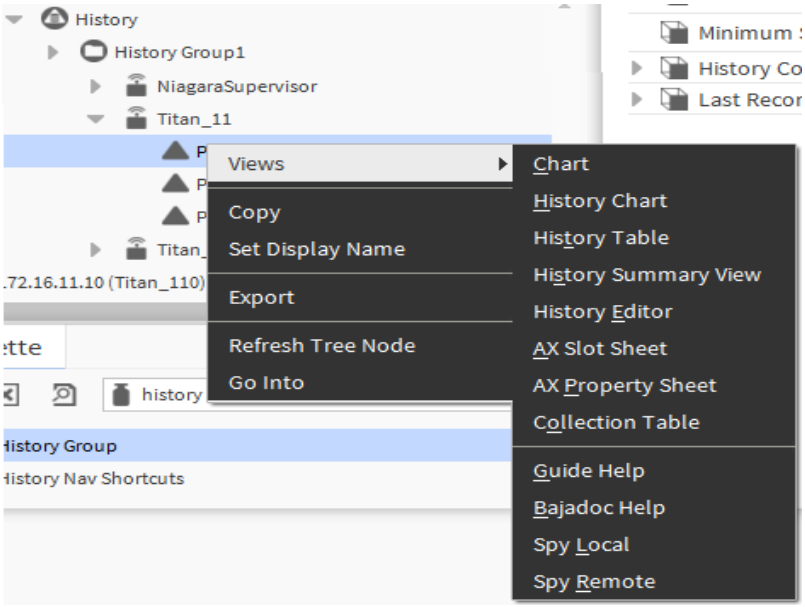
- ❷ Command bar — click icons to launch chart commands
- ❸ Cursor position indicator
- ❹ Data Value popup — displays when cursor is on a point
- ❺ Y-Axis label — default orientation of Y-axis for primary data
- ❻ X-Axis label — default orientation of X-axis. Once you have defined a specific Time Range for the chart, you can click this label to reopen the **Time Range** window to modify the range.

Export icon in Workbench view:

Export icon in browser view:

You can view histories in different ways in Workbench.

Figure 2. History views available from popup menu



The screen capture shows a menu of views that are available using either the Workbench view selector or a view popup menu.

Chart types

Although you can configure Chart Type via the **Settings > Series** window, the default chart type is determined by the type of data being presented. For example:

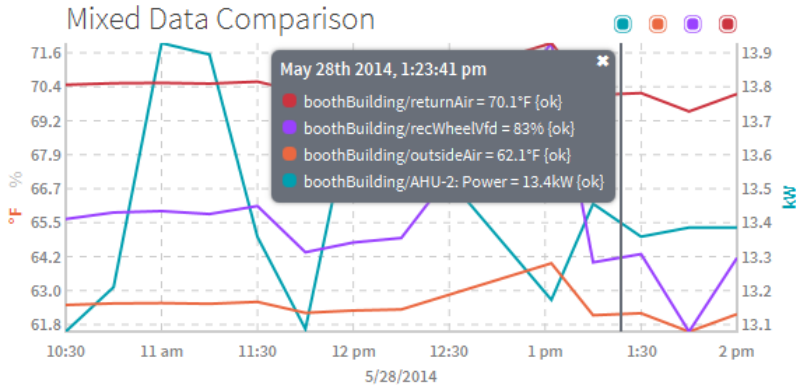
Component type	Default chart type
Numeric histories and points	Render as lines with interpolation and display as a line chart.

Component type	Default chart type
Numeric schedules	Render as discrete lines with no interpolation and display as a line chart.
Boolean and Enum points	Render as shaded areas referred to as swim lanes and displayed as a shaded chart. The ordinal of the Enum determines the opacity of the swim-lane fill.
Boolean and Enum schedules	Render as shaded areas referred to as swim lanes and displayed as a shaded chart. The ordinal of the Enum determines the opacity of the swim-lane fill.

Different types of data (Numeric and Boolean or Enum) can be combined on the same chart. To allow you to more clearly view the lines representing the numeric data, the swim lanes representing Boolean and Enum data display with dimmed opacity. Also, you can modify the default chart type of one or more components in a chart. For example, you can set a Boolean writable point to display bars while the data for another component plots a line.

The interactive Chart view allows you to modify the chart while it is rendering. For example, while viewing, you can add one or more points, history records, schedules and even containers of data. When adding data to a chart, the Y-axis automatically adjusts the units and can accommodate different units of measure by displaying multiple Y-axes.

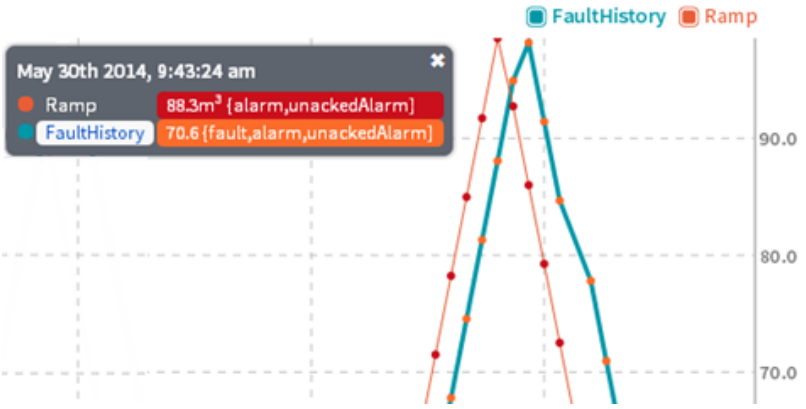
Figure 3. Multiple Y-axes accommodate data with different units of measure



On a chart containing data with three or more different units of measure, such as that shown above, the display still shows dual Y-axes. To switch the units displayed on the secondary Y-axis, click on the dimmed axis label. For example, on the left-side Y-axis, the dimmed % symbol indicates an alternate Y-axis with percent as the unit of measure. Clicking that **%** symbol switches the Y-axis units from displaying degrees to displaying percentage.

You can hide or show specific data and even completely remove data from a chart. Chart settings can customize the appearance of a chart via selectable data colors, chart type per component, axis orientation, and data source zooming, as well as turning the chart grid on or off, changing the background color, configuring data pop-ups and status colors.

Figure 4. Line chart displaying status colors



Web charts use standard Niagara status colors to indicate current status. The chart above invoked the Status Coloring command, a red dot indicating Alarm status to mark each plot in the **Ramp** line while an orange dot indicating Fault status marks each plot in the **FaultHistory** line. Status colors shown in the **Fixed Data** window confirm the status of charted data.

Shade and Bar charts display status colors. When enabled, and if there is a non-ok status, a color band at the top of the shaded area or bar indicates the status.

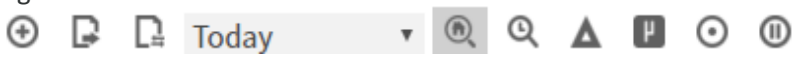
- **Chart commands**
  Options in the **Chart** view **Command Bar** allow you to fine tune data presentation.
- **Chart settings**
  Options in the **Chart** view **Settings** window allow you to make data presentation changes that are of a persistent nature, meaning the changes are retained when the chart is exported or saved.

Parent topic: History plugins

Chart commands

Options in the **Chart** view **Command Bar** allow you to fine tune data presentation.

Figure 1. Command Bar

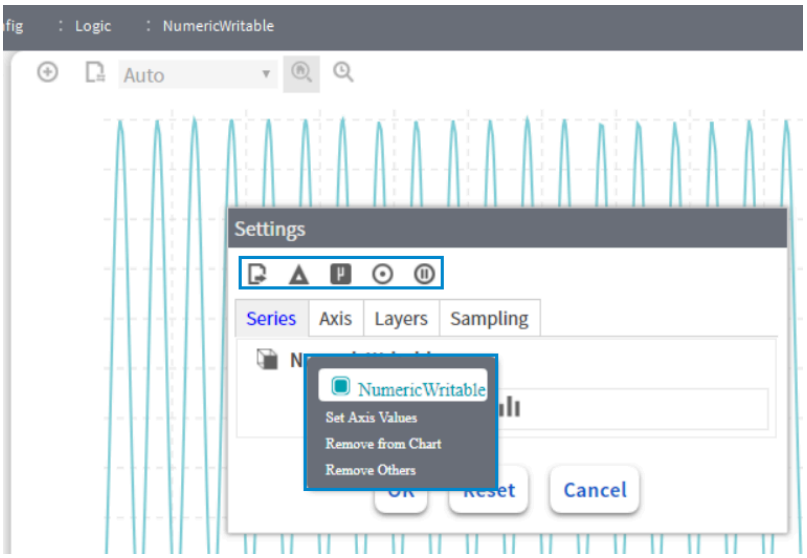


Note: Most options in the **Command Bar** provide fine tuning for viewing purposes only. Changes made with those options are of a temporary nature and are not included when the chart is saved or exported. For example, if you turn on time zoom and Delta using buttons in the command bar and then export the chart, the chart file displays with the original default settings for those options. Exceptions to this are changes made with the Time Range, Sampling, and Status Coloring options, which are included on export or save.

To export the chart and retain all of the changes that you have made, you need to do the following:

1. Export the modified chart to a Station File.
2. Create a **Px** view for this chart and load the exported Station File to this **Px** view. The chart will display with the modifications included.

Figure 2. Narrow chart width changes the chart display



Any time the chart width is less than 800 pixels the following changes in the chart occur. This prevents the chart from appearing overcrowded which helps maintain legibility. Once the chart window is resized to greater than 800 pixels, the changes revert.

- Chart title and data series legend become hidden.
- Several of the commands icons move from the chart **Command Bar** into the **Settings** window.
- In the **Settings** window, a right-click menu is available on data series in the **Series** window. The right-click menu allows you to hide or show specific data or even completely remove data from a chart.

Command Bar	Default	Description
Add Series	opens separate Nav tree	Adds components to the chart. Select one or more components via the File Chooser. Use `Ctrl + Click` to select multiple individual components or select a folder that contains multiple components.
Save Chart	opens save window	Available only when you open an existing .chart file and make changes. Saves the file (chart or csv format) to the station File space (Files/charts/chartName.chart or Files/csv/chartName.csv).
Export Current View or Object	opens exporting window	Available in a new chart and when you open an existing chart file. • Select Exporter opens a window to choose the exported file type. Options are Chart (default), CSV, and, in a browser connection, Print is also available. • Ord Type selects the type of ORD: Absolute or Relative (default). Note: In Niagara, there is added support for relative ORDs to better accommodate Px page reuse.

Command Bar	Default	Description
		• Base Ord specifies a base ORD from which to relate all of the ORDs in the series for that chart. This option applies only to chart exports with the Relative Ord Type. • Select Destination provides two options: Download or Station File. Download exports to your Windows file system. Station File exports the file to the station File space (Files/charts/chartName.chart) or (Files/csv/chartName.csv) • • File Name • View On Export
Time Range **Time Range**	defaults to Auto (default)	Opens a window where you can enter custom Start and End times for the range. Leave the End time property blank for live data to continue plotting on the chart. Options are: • Auto (default) • Time Range • Today • Last 24 Hours • Yesterday • Week To Date • Last Week • Last 7 Days • Month To Date • Last Month • Year To Date • Last Year
Toggle Home Zoom	• On (default) • Off	Turns On/Off Home Zoom. On — zooms to the X-axis of the primary data set. Note: If the primary data set is numeric, it zooms on the Y-axis. Off — reverts to Home Zoom.
Toggle Time Zoom	• On • Off (default)	Turns On/Off Time Zoom. On — zooms X-axis to the time period specified by the **Time Range** drop-down list. Off — reverts to Home Zoom.
Toggle Delta Command	• On • Off (default)	Turns On/Off Delta. On — plots the rate of change between points. Off — resumes plotting data points.

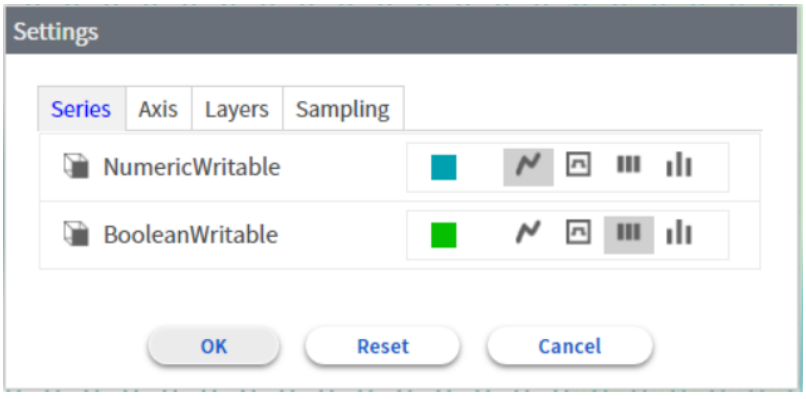
Command Bar	Default	Description
Toggle Sampling Command	• On • Off (default)	Turns On/Off Sampling. On — sampling is enabled Off — turns off sampling and disables auto-sampling behavior.
Toggle Status Coloring	• On • Off (default)	Turns On/Off data Status Coloring. On — displays data points with status colors in a line chart and in shade or bar chart displays a status color band at the top of each bar. Off — hides status coloring, data points/color bands.
Toggle Pause	• On • Off (default)	Turns On/Off pause in live data plotting. On — pauses live data plotting. No longer in live mode when paused Off — resumes live data plotting
Stop	• On • Off (default)	Visible only during data loading. Turns data chunking On/Off. On — stops the data chunking process, halts data coming from the server. While stopped, the button displays a red border. Off — reloads all of the data.

Parent topic: workbench-WebChart

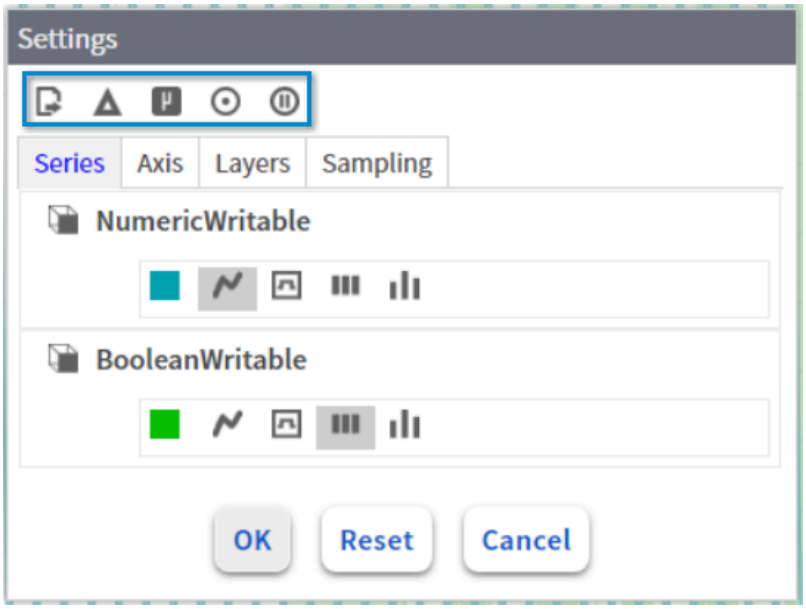
Chart settings

Options in the **Chart** view **Settings** window allow you to make data presentation changes that are of a persistent nature, meaning the changes are retained when the chart is exported or saved.

Figure 1. Settings Window in Workbench



Note: In the Niagara 4.8 Workbench and later, if the chart width is less than 800 pixels several of the chart commands icons are moved into the **Settings** window above the tabs. When the chart is resized wider than 800 pixels, those icons revert back to the Commands Bar in the chart.
Figure 2. Commands icons in the Settings window



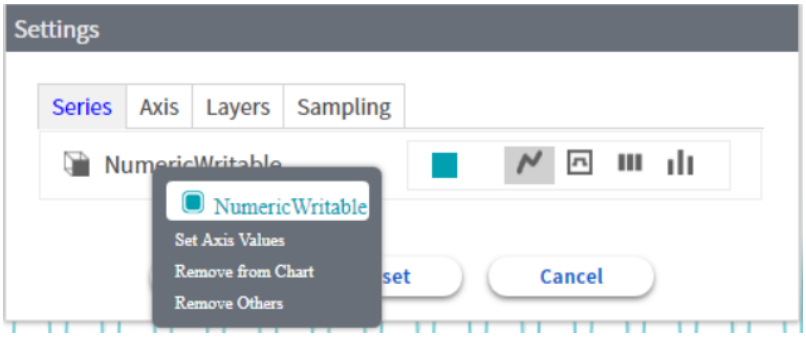
Series tab

Settings	Options	Description
Color	Color block assigned to each component	Change default data color by clicking color block and selecting different color via Color Picker. As of Niagara 4.12, the Color Picker is also available when you are connected to your station via the browser using the HTML5 browser-based implementation of the WebChart view.
Chart type	• Line, • Discrete line, • Shade, • Bar	Line — plots a smooth line with interpolation. The default chart type for Numeric points and histories. Discrete line — plots a “stepped” line without interpolation. Shade — plots shaded areas, known as “swim lanes,” representing state change. The default chart type for Boolean and Enum points. Bar — plots vertical bars. Samples data into common intervals based on available width, When you have more than one component in a chart using bar chart type, they become a Bar Group, where the individual bars are adjacent (no space between). As shown below, clicking on a Bar Group selects the entire group and the values for all components in the group are shown in the Fixed Data Popup. While the mouseover Data Value Popup, shows the value of a single component.

Right-click menu

A right-click menu is available on data series in the **Series** tab. The right-click menu allows you to hide or show specific data or even completely remove data from a chart.

Figure 3. Right-click menu options



Axis tab

Settings	Options	Description
Y-Axis Orientation	• left • right (default)	Aligns Y-axis of primary data to either the left or right side of the chart.
Data Zoom Scope	• primary (default) • all	Sets the Data Zoom Scope to primary or all. Primary — zooms to the X-axis of the primary data set only. If the primary data set is numeric, it zooms on the Y-axis. All — changes the X-axis to accommodate all available data, including live data as it is recorded.
Show Grid	• true (default) • false	Turns on/off the chart grid. true — the grid displays in chart behind data. false — the grid does not display.
Background Color	• On • Off (default)	Turns on/off the background area color for the current theme. On — the background area color displays in chart behind data. Off — the background area color does not display.
Chart Cursor	• Crosshair (default) • None	Sets the appearance of mouse pointer while positioned over a chart. Crosshair — the mouse pointer appears as a crosshair. None — turns off the mouse pointer visibility (while positioned over a chart), hiding it completely.
Facets Limit Mode	• Off (default) • Inclusive • Locked	Configures whether the WebChart uses a point's facets for Min and Max. Off (default) — the WebChart ignores a point's facets for Min and Max. Inclusive — the WebChart includes a point's facets for Min and Max. Locked — forces the WebChart to use a point's facets for Min and Max. Note: In each of these settings chartMin and chartMax facet keys can be used as a higher priority substitute to "min" and "max". Even if the Facet Limit Mode is "Off" it can be overridden for specific series if a facet key of chartLimitMode is supplied with the corresponding values of "Inclusive" or "Locked".

Settings	Options	Description
		Note: Previously, if you were not using a chart file to load a WebChart, there was no way to preset any options. In latest version of Niagara, there is a Default Options WebProperty on a Px page which you can modify to preset WebChart default options. By default, modifications are saved to file:^charts/defaultOptions.chart. Even when not on a px page, non-chart files will load their options from this file if it exists and the user has permissions to view it. This includes the ability to change all options, so even the default time range can be preset.
Show Start Trend Gaps	• Yes (default) • No	Configures the behavior when drawing the chart line, providing a visual indication (a line gap) of an interruption in data collection. For example, a station restart or that history collection was disabled and re-enabled. Yes — if there is a start trend flag on a record the chart does not connect the dot for that record to the previous record, resulting in a gap in the line No — allows the dots to be connected, eliminating any such gaps.
Show Data Gaps	• Yes • No (default)	Configures the behavior when drawing the chart line, it providing a visual indication (a line gap) for records that have either the hidden flag set or invalid values (+inf, -inf, NaN). Yes — if a record has a hidden flag set or invalid values (+inf, -inf, NaN) the record's dots are not connected to adjacent records. No — if a record has a hidden flag set or invalid values (+inf, -inf, NaN) the record's dots are connected to adjacent records.

Layers tab

Settings	Options	Description
Data Popup	• On (default) Displays • OffPauses	Enables/disables the **Fixed Data** popup. On — clicking on chart data displays the recorded date and time, as well as the name, value and status for each component in the chart at the point where you click. The persistent data popup remains visible until you close it. Off — suspends display of fixed data popup.
Data Mouseover	• On (default) • Off	Enables/disables the mouseover **Data Value** popup. On — mouse position on chart data displays the recorded component value, status, and the time for that mouse position. Off — suspends display of mouseover data value popup.
Status Coloring	• On • Off (default)	Turns On/Off data status coloring. On — displays data points with status colors in a line chart and in a bar chart displays a status color band at the top of each bar. Off — hides status color data points/color bands in the chart.

Sampling tab

Settings	Options	Description
Auto Sampling	• true (default) • false	Enables/disables automatic sampling optimizations. true — automatically begins sampling if the focused data set exceeds 2500. false — automatically stops sampling if the focused data set is below 2500.
Sampling Type	• Average (default) • Min • Max • Sum	Sets the Sampling type. Average - samples average values for the selected rollup period. Min - samples minimum values for the selected rollup period. Max - samples maximum values for the selected rollup period. Sum - samples the total of the values in the selected rollup period.
Desired Period	• Best Fit (default) • 1 Minute • 15 Minutes • 30 Minutes • Hour • Day • Week • Month • Year • Custom	Configurable setting allows you to choose the desired sampling interval. By default, set to Best Fit which finds the best sampling period that fits the page that is one the standard collection intervals which are: Year, Month, Day, Hour, 30 minutes, 15 minutes,1 minute, and other smaller common intervals.
Sample Size	2500 (default)	Specifies the number of points in the data set to sample. Range is 1–50000. Note: The default auto sampling size is configurable in the system.properties file.
Sampling	• true • false (default)	Enables/disables sampling for any size data set. true — turns on sampling false — turns off sampling Note: For performance reasons, sampling cannot be turned off once the focused data set exceeds 50,000. This threshold is configurable in the system.properties file.
Data Points	Read only	Displays the maximum number of points in the data set that are available to sample.
Sampling Period	Read only	Visible only once sampling has begun, displays the calculated average of the amount of time between each of the points that have been sampled.

Parent topic: workbench-WebChart

Collection Table view

This view shows a table of any data (in this case history data) that you can export and view in the following formats: PDF, CSV, Text.

This view displays records with columns of data that you can customize by displaying or hiding selected columns. An example of the collection table view is shown here.

Figure 1. Collection Table view

500 rows

Timestamp	Log Name	Severity	Message
12-Nov-18 2:53:10 PM IST	javax.baja.web.BWebServer	1000	failed: java.net.BindException: Failed to bind to https port [443]
12-Nov-18 2:53:10 PM IST	javax.baja.web.BWebServer	800	Scheduling restart in 5 seconds.
12-Nov-18 2:53:16 PM IST	web	800	Jetty stopped.
12-Nov-18 2:53:16 PM IST	javax.baja.web.BWebServer	1000	failed: java.net.BindException: Failed to bind to https port [443]
12-Nov-18 2:53:16 PM IST	javax.baja.web.BWebServer	800	Scheduling restart in 5 seconds.
12-Nov-18 2:53:21 PM IST	web	800	Jetty stopped.
12-Nov-18 2:53:21 PM IST	javax.baja.web.BWebServer	1000	failed: java.net.BindException: Failed to bind to https port [443]
12-Nov-18 2:53:21 PM IST	javax.baja.web.BWebServer	800	Scheduling restart in 5 seconds.
12-Nov-18 2:53:27 PM IST	web	800	Jetty stopped.
12-Nov-18 2:53:27 PM IST	javax.baja.web.BWebServer	1000	failed: java.net.BindException: Failed to bind to https port [443]
12-Nov-18 2:53:27 PM IST	javax.baja.web.BWebServer	800	Scheduling restart in 5 seconds.
12-Nov-18 2:53:33 PM IST	web	800	Jetty stopped.
12-Nov-18 2:53:33 PM IST	javax.baja.web.BWebServer	1000	failed: java.net.BindException: Failed to bind to https port [443]
12-Nov-18 2:53:33 PM IST	javax.baja.web.BWebServer	800	Scheduling restart in 5 seconds.
12-Nov-18 2:53:38 PM IST	web	800	Jetty stopped.

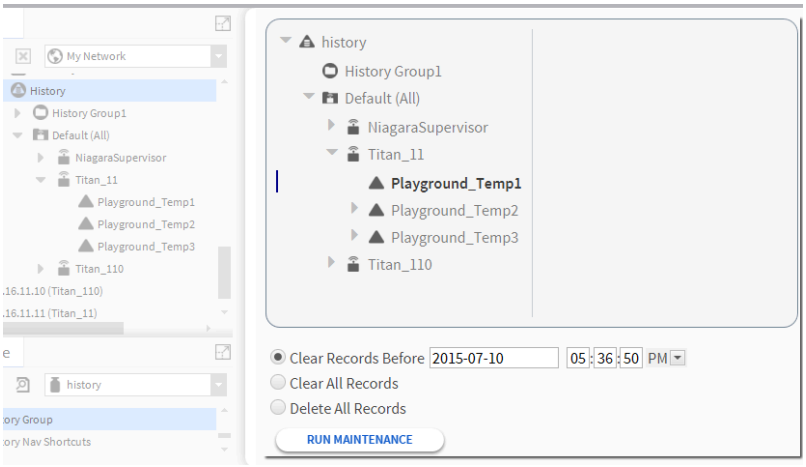
Use the **Table Options** menu in the top right corner of the collection table to modify the table view or to export the data in the view, as desired.

Parent topic: History plugins

Database Maintenance view

This view is one of the views available on the **History** space node in the Nav tree. Using this view, you can clear records and delete complete histories from your history database.

Figure 1. Database maintenance view



Select the view by selecting the **History** space node in the Nav tree and using the Views menu (or the right-click

popup menu) to choose the Database Maintenance view.

The left side of the **histories** area contains the **available histories** window. This window displays all histories that are available in your local station or any station histories that you import by means of the **NiagaraNetwork** or other network driver (for example, **BacnetNetwork**). Histories are grouped under the station by station name.

Note: The available histories are the same histories that are displayed under the **History** space node in the Nav tree.

The right side of the **histories** area contains the targeted histories window. This window displays the histories that are affected when you click the **Run Maintenance** button. Move the histories that you want manage into this window using the control buttons, as described below:

Controls and options for the database maintenance view are described in the following list:

- Add history button (right arrow)

  Click this button to move histories that are selected in the **available histories** window to the targeted histories window.

- Remove history button (left arrow)

  Click this button to move histories that are selected in the **targeted histories** window to the available histories window.

- Clear Old Records option

  Select this option and use the **Before** date selector to remove records, based on date, from the histories that are in the targeted histories window.

- Before date property

  Use this property with the Clear old records option to set the year, month, day, and time properties that you want to use for removing old records.

- Clear all records

  Select this option to delete all records from the selected history database.

- Delete Histories

  Select this option to delete all histories that are in the targeted histories window.

- Run Maintenance button

  Click this button to execute the option that you have selected on the histories in the targeted histories window.

Parent topic: History plugins

Device Histories View

This view is available on any device that supports import and export of histories (for example, BACnet devices, and others). The view shows a filtered list of history shortcuts for the particular device.

The view displays all related shortcuts in a table. You can double-click on any single entry in the table to open that history in the **Chart** view.

Parent topic: History plugins

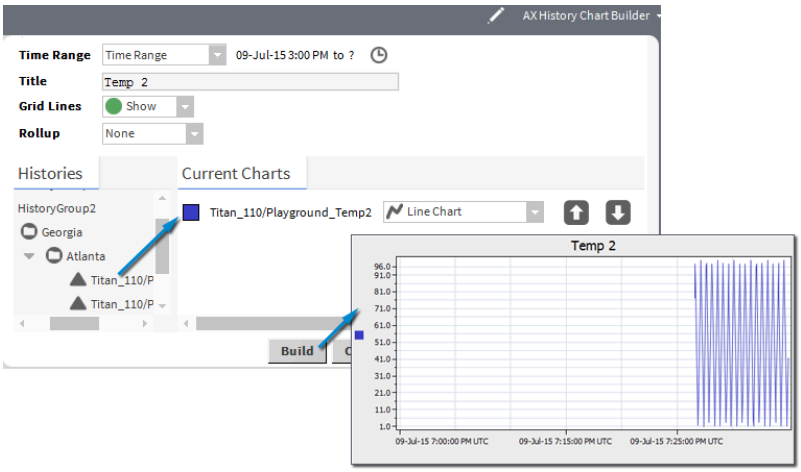
History Chart Builder view

This view is the default view of the **History** space node (in the Nav tree). As one of the available views, you can build any one of several types of chart options from the data that are stored in one or more histories. This view displays in Workbench, in all Workbench (Wb) Web profiles, and in the Default Hx, and Basic Hx profiles as well.

Note: The following description discusses the **History Chart Builder** view in terms of the Wb Web and the desktop Workbench display profiles. Stations can display the **History Chart Builder** view in the browser using the Default and Basic Hx (non-Java applet) profile views. Views displayed using Hx look different but behave very similarly to the Workbench view. All property descriptions apply to both views.

The **History Chart Builder** Workbench view shown here. Using this view, you can select the histories and History Nav Shortcuts and configure the charts that you want to generate.

Figure 1. History Chart Builder view



The display of the view is divided into three primary areas:

Display configuration properties

Property	Value	Description
Time Range	Time Range, Today (default), Last 24 Hours, Yesterday, Week-To-Date, Last Week, Last 7 Days, Month-To-Date, Last Month, Year-To-Date, Last Year	Specifies a time property option from the drop-down list, including an option that allows you to set a specific time range using the **Edit Time Range** window box.
Title	Text string	Specifies a title for your chart in this text property.
Grid Lines	Hide, Show (default)	Shows or hides the grid on the history chart.
Rollup (or Rollup Interval)	None (default), 1-minute, 5-minutes,	Specifies an interval of time used to

Property	Value	Description
	15-minutes, 30-minutes, Hour, Day, Month, Year	determine what (and how) data are presented in your chart. Each point displayed, represents a designated time interval before the specified plot time. For example, a value of 1 hour presents data at a granularity level of every one hour, while a value of 15 minutes shows data for every 15 minutes of logged data.

Histories pane

This is the lower left area of the **Chart Builder** view. It displays all histories that are available in your local station or any station histories that you import by means of the **NiagaraNetwork** or other network driver (for example, **BacnetNetwork**). Histories are grouped under the station by station name. Double-click (Wb Web profile) or click (Hx Web profile) on a history name to copy it to the Current Charts pane.

The histories that are displayed in this pane are the same histories that are displayed under the **History** space node in the Nav tree.

Current Charts pane

This pane displays the histories that are selected to be plotted. For each history, you may select the type of chart to generate, using the chart type option list, as displayed in .

Adjacent to the histories in this pane are icon-type controls that allow you to reorder histories in the pane or remove histories from the pane.

Control buttons

The following control buttons are located at the bottom of the **Chart Builder** view:

- **Build** builds the chart using the histories that are in the selected Current Histories pane.
- **Clear** removes all histories from the Current Histories pane.

Time zones

When charting multiple histories that include different time zones, the Chart Builder uses a zoneless time range configuration so that it can plot each history with reference to its own time zone. This means that the resulting charts are aligned by local time. For example, if you select a time range of 8:00 AM through 5:00 PM for two histories—one in EST and another in CST—then the values at 8:00 AM align so that the 8:00 AM values may be visually compared.

Parent topic: History plugins

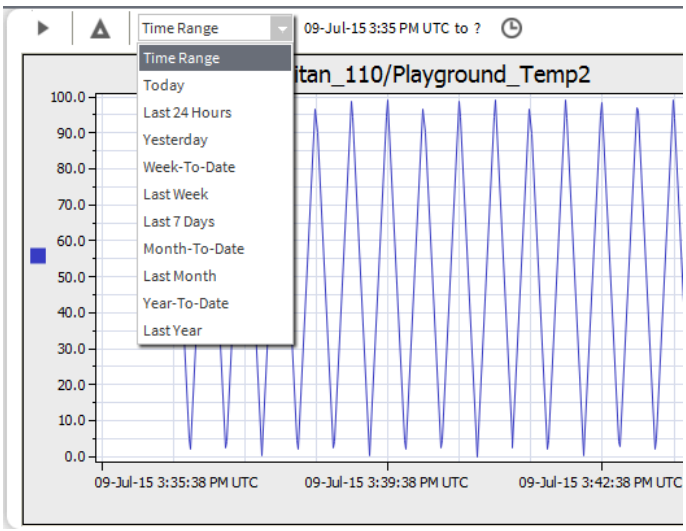
History Chart view

This view plots the historical data of the selected history log along X and Y axes. With this view you can display and configure charts of history records. Charts created in an earlier release are available as secondary **History Chart** views on history records.

To chart any point history in the **History** space, double-click directly on it. An example of the history chart view is

shown here.

Figure 1. HistoryChart view



Buttons

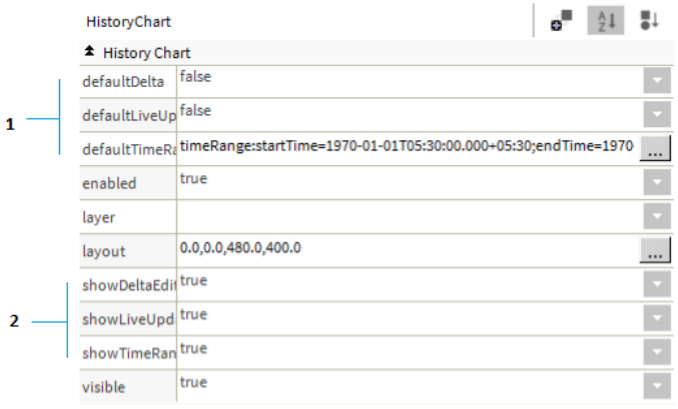
The **History Chart** view contains the standard chart controls and options to help you customize and view the data. In addition to these standard controls, the **History Chart** view has the following buttons (also available on the toolbar):

- **Live Updates** initiates On Demand plotting of the history data.
- **Delta** initiates the plotting of Delta values.

Properties on a Px page

You edit **History Chart** properties in a **Px page** view.

Figure 2. History Chart properties in Px page



1. Choose how to display these chart features on initial view.

2	Choose to show or hide these chart features individually.

Properties include the following.

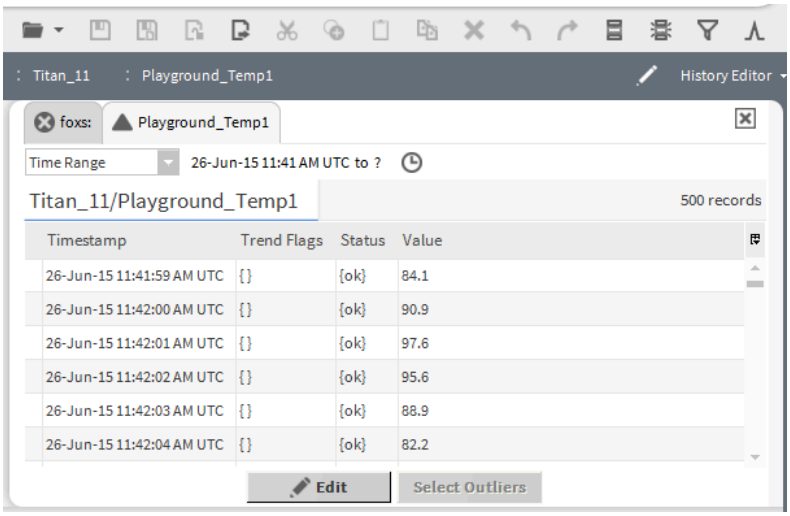
Name	Value	Description
defaultDelta	true or false	Controls the display of initial delta values. true causes the **History Chart** view to plot delta values on initial display and any time the view is refreshed. false leaves the values unchanged.
defaultLiveUpdates	true or false	Configures the **Chart** view for live update mode on initial display and whenever the view is refreshed (true) or to normal mode (false).
defaultTimeRange	selectable time range options	Selects a specific time range option that appears on initial display and whenever the **History Chart** view is refreshed.
enabled	true or false	Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.).
layer		Groups objects in the **Px Editor** view by assigning them to a common layer.
layout	selectable coordinates, 0, 0, 450, 350 (default)	Provides four values (in pixels) that define the size of the history chart view display area.
showDeltaEditor	true or false	Hides (false) or shows (true) the delta mode button on the history chart view.
showLiveUpdates	true or false	Hides (false) or shows (true) the live update button on the history chart view.
showTimeRangeEditor	true or false	Hides (false) or shows (true) the Time Range Editor option list on the history chart view.
visibility	true or false	Hides (false) or shows (true) the history chart view.

Parent topic: History plugins

History Editor view

This view edits data and filter histories, including batch editing (selecting multiple rows and using the right-click menu or the **Edit** menu to make changes).

Figure 1. History Editor view



The **History Editor** view is comprised of the following main areas:

- The title bar displays the history name and number of records in the history.
- The toolbar icons are available on the Workbench toolbar (far right, above the **Views** drop-down list) when the **History Editor** view is open.s
- The time range options menu is located in the top left corner of the view. Use it to select one of the pre-defined times or to select the Time Range and then click (Clock) icon to set a specific time range using the **Edit Time Range** window.
- The Table **Table Options** drop-down list in the top right corner of the view changes which columns are displayed and exports the data in the view.

Columns

In addition to a title bar that displays the history name and number of records in the table, the history table has the following columns.

- Timestamp
- Trend Flags
- Status
- Value

Control buttons

These buttons initiate record editing:

- **Edit** is available when one or more records are selected in the history editor table. When you click this button, the **Edit Records** window box displays, allowing you to configure whether the selected record(s) is hidden or not, and to set a specific numeric value for the selected record(s).
- **Select Outliers** is available when the outlier configuration properties are active (when the check box is selected in the **Configure Outliers** window box). When you click this button, the records in the history table are examined for outlier values (as defined in the **Configure Outliers** window) and any records

with outlier values are selected. At this point, you can use the toolbar icons to **Hide** or **Filter** those records.

Parent topic: History plugins

History Extension Manager view

As the default view of the History Service, the **History Extension Manager** provides a tabular view of all history extensions in the station and displays information about the control point, extension type, name and the current status.

Using this view as both a management and navigational tool, you can double-click on any entry-row to go directly to the **Property Sheet** view of that extension.

This table has the standard table features. Use the **Table Options** menu in the top right corner of the history table to modify the table view or to export the data in the view, as desired.

Figure 1. History extension manager

You can also **Enable**, **Disable**, **Rename**, or **Edit System Tags** for any collection from this view by selecting the desired history extension in the table and using the History Ext Manager menu, popup menu, or toolbar icons.

Note:

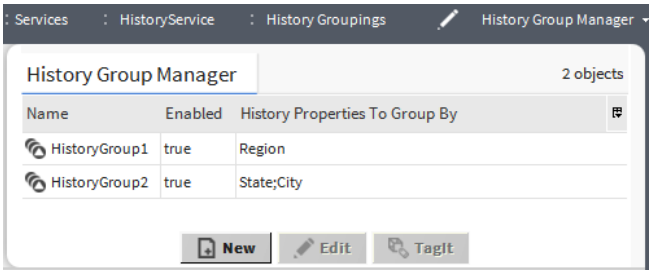
If a history is disabled, its row is dimmed with a gray background.

Parent topic: History plugins

History Group Manager view

The default view of the HistoryGroup component provides a tabular view of all history groups. It is the default view of the HistoryGroupings component and provides information about the groups contained in the HistoryGroup component.

Figure 1. History Group manager view



You can add HistoryGroup components under the HistoryGroupings component by dragging and dropping a HistoryGroup from the History palette or you can add HistoryGroup components to this component by clicking the **New** button. The **History Group Manager** shows all history groups in the station and displays the following columns in a table view:

You can double-click on any entry-row to open the **Edit** window box for editing the group. This table has the standard table features. Use the **Table Options** menu in the top right corner of the history table to modify the table view or to export the data in the view, as desired.

Column name	Description
Name	This is the identifier for the grouping. You can select a row and **Rename** or **Edit** the group using the control buttons at the bottom of the view.
Enabled	This column indicates whether or not the group is actively filtering. If the Enabled value is set to False, then no custom grouping displays under the **History** space node for that group. If no history groups are present or enabled, the default device-oriented history organization displays.
History Properties To Group By	This column displays a summary list of the history properties that are set to filter the presentation of the histories. Individual properties are separated by a semicolon.

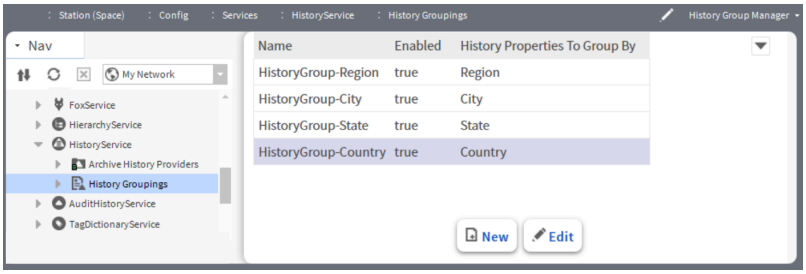
Parent topic: History plugins

History Group Ux Manager view

This is the default view for adding groups to the station using the **HistoryGroupManager** view.

To access this view, **Config > Services > HistoryService** and right-click **HistoryGroupings > views** and click **HistoryGroupManager**.

Figure 1. History Group UxManager view



You can add HistoryGroup components to this component by clicking the **New** button.

You can double-click on any entry-row to open the **Edit Duplicate** and **Remove** window box for editing the group. This table has the standard table features.

Column name	Description
Name	This is the identifier for the grouping. You can select a row and **Rename** or **Edit** the group using the control buttons at the bottom of the view.
Enabled	This column indicates whether or not the group is actively filtering. If the Enabled value is set to False, then no custom grouping displays under the **History** space node for that group. If no history groups are present or enabled, the default device-oriented history organization displays.
History Properties To Group By	This column displays a summary list of the history properties that are set to filter the presentation of the histories. Individual properties are separated by a semicolon.

Buttons

Column name	Description
New	Creates a new history group.
Edit	Opens the Edit window for updating the selected record.
Duplicate	Duplicates the selected history group.
Remove	Removes the selected history group.

Parent topic: History plugins

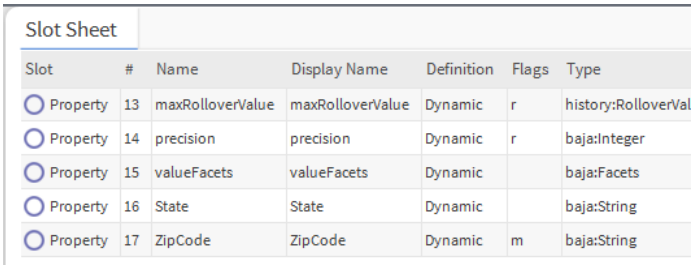
History Slot Sheet view

This view displays a standard slot sheet for a history. It is available on all histories to add slots and designate them as metadata slots, if desired. To designate slots as metadata, you assign a metadata flag to a slot.

Note:

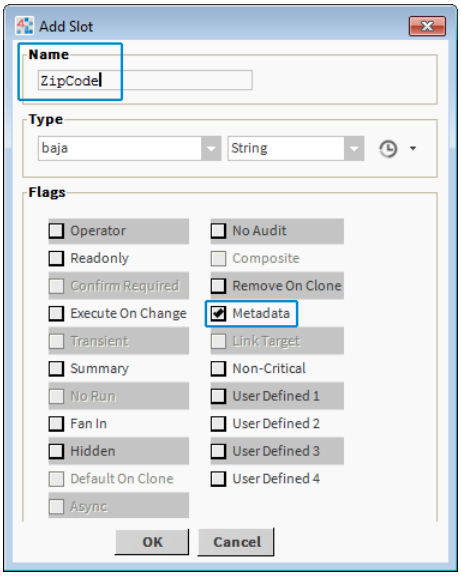
Properties that have the metadata flag are designated as metadata properties. You might use metadata property values to identify the location and function of a history's source. History sources may be filtered or organized by the values of the metadata properties. You can use the **Metadata Browser** view to add, edit, delete, or view metadata tags.

Figure 1. History slot sheet view



To add a new property to a history using the **Slot sheet** view, open the **Add Slot** window box (Ctrl+A) from the history **Slot sheet** view, as shown below.

Figure 2. Select a Metadata flag to create a metadata slot

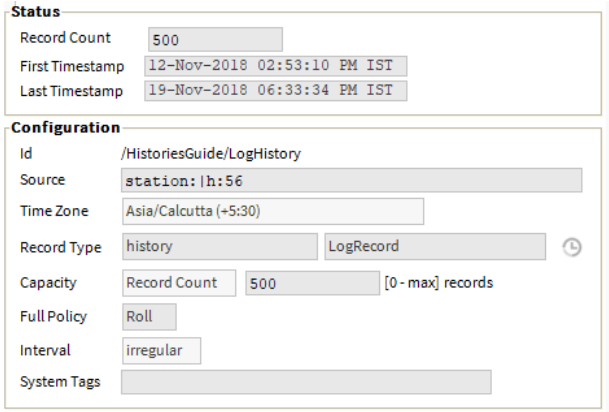


Parent topic: History plugins

History Summary view

This view displays read-only values for each history record's status and configuration properties.

Figure 1. History Summary view



This view displays the details of the history in two groups, as follows:

Status displays updated data as of the time you selected this view.

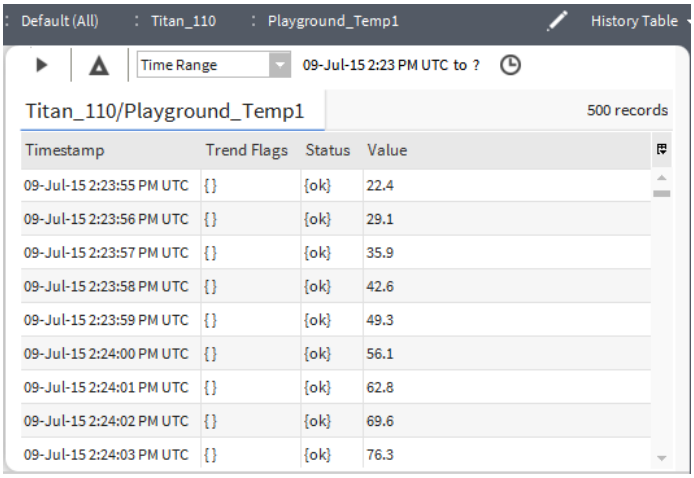
Configuration displays data that identify and characterize the specific history.

Parent topic: History plugins

History Table view

This view shows a table of history data that you can export and view in the following formats: PDF, CSV, Text.

Figure 1. History Table view



Use the **Table Options** menu in the top right corner to show and hide columns, filter the data based on date and time and export the data, as desired.

Columns

In addition to a title bar that displays the history name and number of records in the table, the history table has the following four columns.

- Timestamp
- Trend Flags
- Status
- Value

Buttons

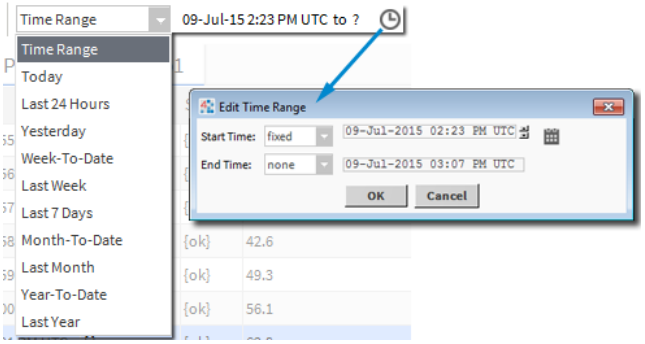
In addition to standard table controls, the history table view has the following buttons (also available on the toolbar):

- ▶ **Live Updates** initiates on-demand plotting of the history data.
- ◭ **Delta** initiates the plotting of delta (change) values.

Edit Time Range window

The query drop-down list selects the period for the query. Selections include:

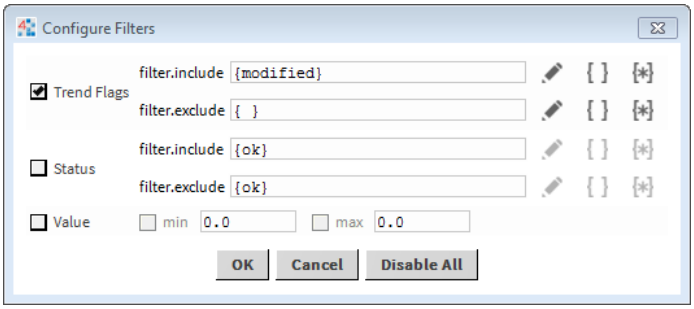
Figure 2. Query drop-down list options and Edit Time Range window



Configure Filters window

The filter icon ( ) opens the **Configure Filters** window.

Figure 3. Configure Filters window



These data are common to several history table views and appear as columns that may be hidden or displayed using the **Table Options** menu.

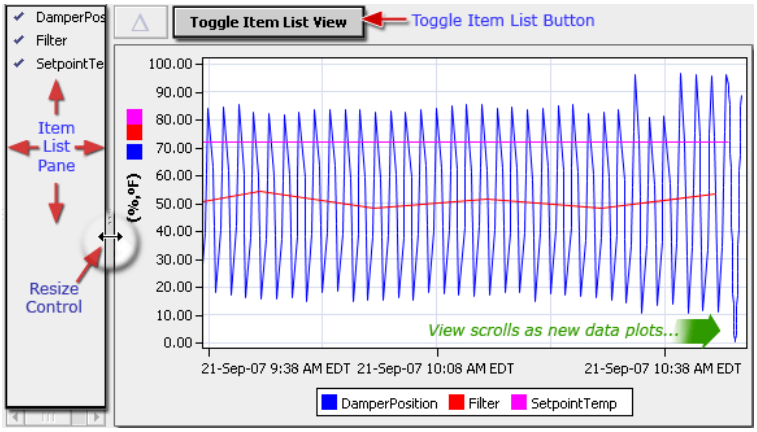
Name	Description
Timestamp	Indicates the time that the recorded value occurred.
Trend Flags	Displays trend flag information about the recorded data—for trend record types. These flags provide extra context information about the record data. For example, Start, OutofOrder, Hidden, Modified, and Interpolated.
Status	Displays the status of the history's parent component; for example, `OK` or `null`.
Value	Displays the record value.

Parent topic: History plugins

Live History Chart view

This view displays historical data (trend data) as the default view for the HistoryPointList component and a secondary view on history extensions. It combines the historical plot of the History Chart with a continuing live plot that updates according to a sample rate that can be configured. It begins with a start time to the current time and continuously updates as the source generates new data.

Figure 1. Live History Chart view



The **Live History Chart** view contains the standard chart controls and options to help you customize and view the data, as well as the following additional features:

- The *Item List Pane* displays all the points that are linked to the **Chart** view. To display or remove an item from the plot, click to select or de-select it.
- The *Resize Control* resizes the pane. Drag the left border to widen or narrow the pane in the view.
- The *Toggle Item List Button* displays or collapses the item list pane.

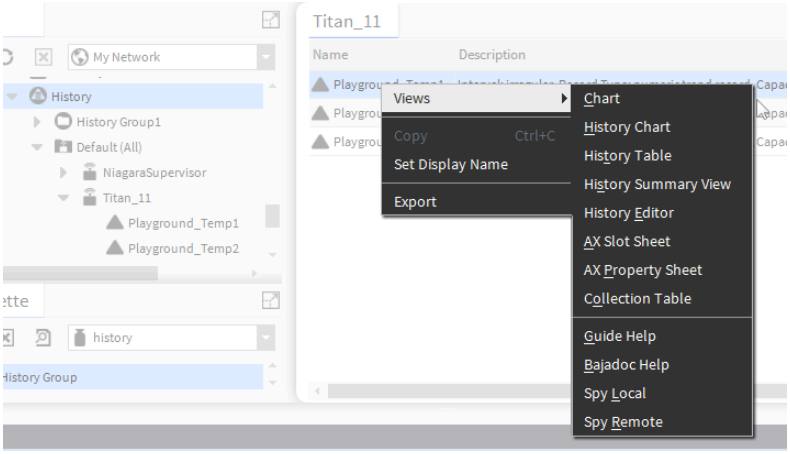
As the chart plots new data the view continuously updates. When data fill the chart area, the screen scrolls horizontally to keep showing the latest data.

Parent topic: History plugins

Nav Container view

This view displays all histories in the station. It is one of the views available on the **History** space node in the Nav tree. It displays a row for each station that is represented in the **History** space. You can double-click on any of the station icons to display the individual histories that are children to the station that is represented in the Name column of the view. Double click on a history in this view to display the history in the **Chart** view.

Figure 1. Nav Container view



In this view you can select any history and switch to any other view of that history using the view selector or the popup menu. In this view you can also rename histories, using the popup menu.

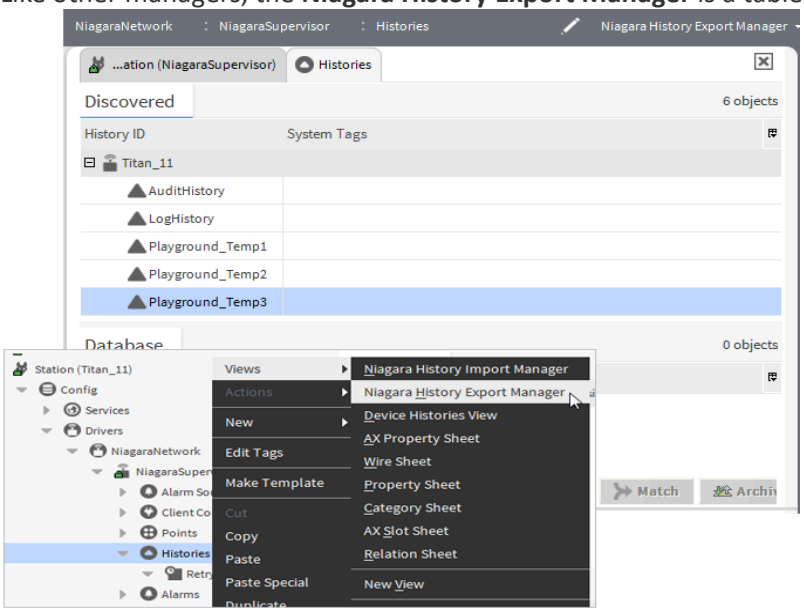
Parent topic: History plugins

Niagara History Export Manager view

This view is available only on a station's **Histories** extension (apart from database device components in some RdbmsNetwork drivers) of the **NiagaraNetwork** (niagaraDriver).

Note: If you are using the BACnet driver, Niagara histories in the local station can also be exposed to all networked BACnet devices as BACnet Trend Log objects. However, this is done using a different view under the Local Device component in the **BacnetNetwork**. Refer to the *Niagara 4 BACnet Driver Guide* for more details.

Like other managers, the **Niagara History Export Manager** is a table-based view.



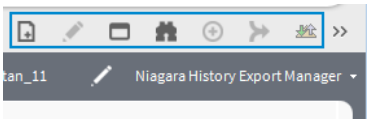
Each row typically represents a history export descriptor. Each descriptor specifies how data from a local history is exported ("pushed") from the station to a selected station, where it appears as a history. You use this view to create, edit, and delete history export descriptors. Each export descriptor you add results in the creation of a history on that remote station.

Following station configuration, this view provides a status summary for exporting local histories. You can also use it to issue manual **Archive** commands to one or more history descriptors. This causes an export (push) of history data into the selected histories at the remote station.

Note: Only history export descriptors appear in the **History Export Manager** view—any other components that may also reside under Histories do not appear. For example, you do not see the default Retry Trigger component, or history import descriptors. However, you can use the histories **Property Sheet** or the Nav tree to access those items.

At the bottom of the view, the **New** button lets you manually create new export descriptors in the station. **Edit** lets you edit one or more export descriptors. **Discover**, **Add**, and **Match** are also available, (these work similarly as in the **Point Manager** view). **Archive** is available to manually export (push data) into one or more selected histories.

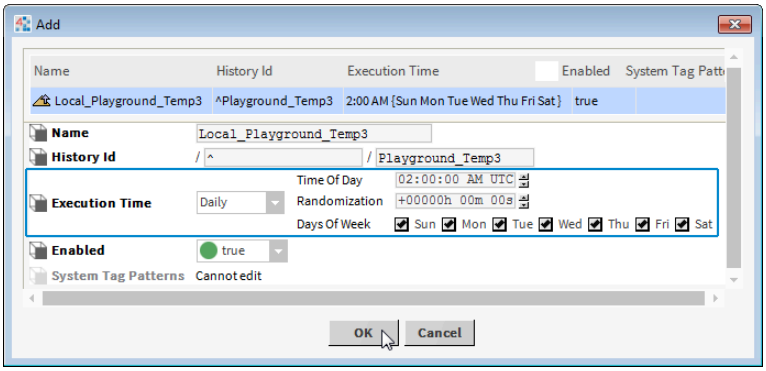
Note: The view toolbar also contains icons for these button commands, as shown here. Positioning the mouse pointer over an icon displays the icon label.



Niagara History Export Properties

The following are properties of history export descriptors available in the **Edit** or **Add** window.

Figure 1. Add export descriptors window



Property	Value	Description
Name	Text string followed by numbers	Defines the history name. For a history originating in the local host station, the name begins with `Local_` . If Discovered for import, typically left at default. For a system history export, originating in the remote station, the name begins with `NiagaraSystemHistoryExport`.
HistoryId	Text in two parts: */stationname/historyname*	Specifies the history name in the local station's **History** space, using two parts: `/<stationName>` and `/<historyName>`. If discovered, station name is "^" (a character representing the device name of the parent container) and history name reflects the source history name. Typically, you leave both properties at their default values, or edit the second (`<historyName>`) property only.
Execution Time — Daily (default)	Time Of Day hours:minutes:seconds AM/ PM timezoneRandomizationDays Of Week	Defines when the daily export or import automatically takes place. The hours follow a 24-hour clock.
Execution Time — Interval	Interval hours:minutes:secondsTime Of DayDays Of Week	Defines the amount of time between automatic exports or imports. Hours may number in the thousands.
Execution Time — Manual	N/A	Requires human intervention to initiate a history export or import.
System Tag Patterns	Text	Specifies one or more text strings matched against text values in "System Tags"

Property	Value	Description
		properties of local history extensions (modifiable only for a NiagaraSystemHistoryExport descriptor). Matching text patterns result in histories exported into the remote **History** space.

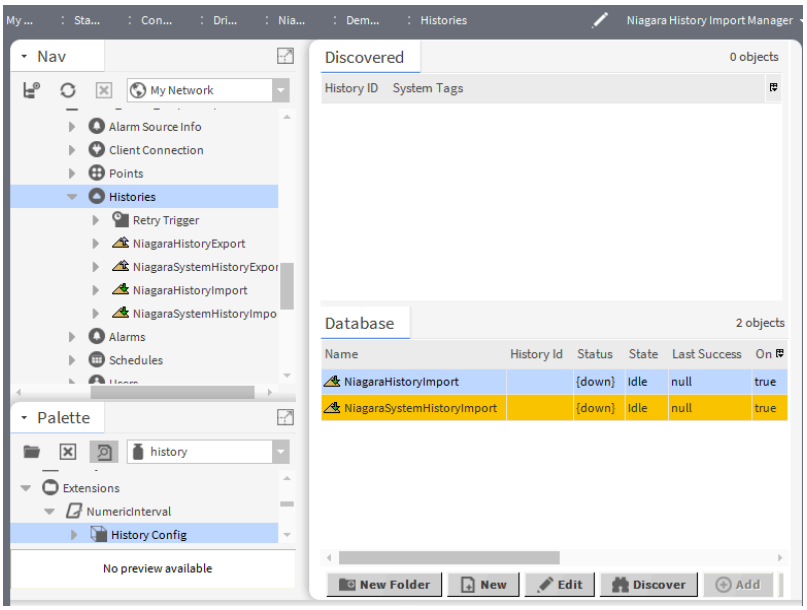
Note: The Capacity and Full Policy of any exported history (created on a remote station) are determined by rules under that station's **NiagaraNetwork** History Policies, and are set at creation time only.

Parent topic: History plugins

Niagara History Import Manager view

This is the default view on a station's **Histories** extension.

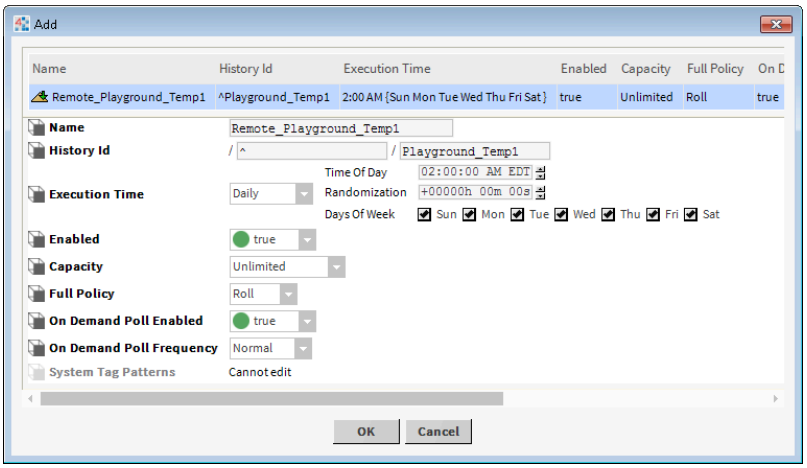
Figure 1. History Import Manager under a **NiagaraNetwork** station



The **New Folder** button is available in this view for adding archive folders to organize import (or export) descriptors. Each folder has its own **history manager** view.

Add (or Edit) history import descriptor window

Figure 2. Add (or Edit) history import descriptor window



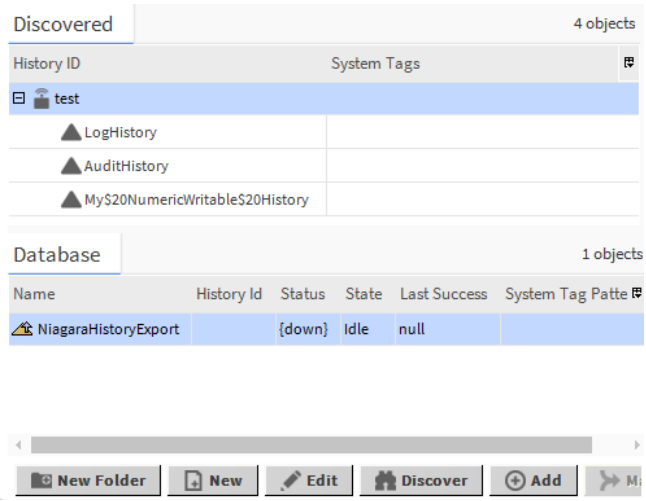
Property	Value	Description
Name	Text string followed by numbers	Defines the history name. For a history originating in the local host station, the name begins with `Local_` . If Discovered for import, typically left at default. For a system history export, originating in the remote station, the name begins with `NiagaraSystemHistoryExport`.
HistoryId	Text in two parts: */stationname/historyname*	Specifies the history name in the local station's **History** space, using two parts: `/<stationName>` and `/<historyName>`. If discovered, station name is "^" (a character representing the device name of the parent container) and history name reflects the source history name. Typically, you leave both properties at their default values, or edit the second (`<historyName>`) property only.
Execution Time — Daily (default)	Time Of Day hours:minutes:seconds AM/PM timezoneRandomizationDays Of Week	Defines when the daily export or import automatically takes place. The hours follow a 24-hour clock.
Execution Time — Interval	Interval hours:minutes:secondsTime Of DayDays Of Week	Defines the amount of time between automatic exports or imports. Hours may number in the thousands.
Execution Time — Manual	N/A	Requires human intervention to initiate a history export or import.
Enabled	true or false	Activates (true) and deactivates (false) use of the object (network, device, point, component, table, schedule, descriptor, etc.).
Capacity	RRecord Count: nnn (500 default), Unlimited	Specifies local storage capacity for histories. In general, 500 (default record count) or less is adequate for a controller station because those records are usually archived (exported) to a Supervisor station. For this reason, a very large number, such as 250,000 is acceptable for

Property	Value	Description
		Supervisor stations. Unlimited is not the wisest choice even for a Supervisor station.
Full Policy	drop-down list	Specifies what happens when a trend log (history) reaches capacity. Applies only if Capacity is set to Record Count. When capacity reaches record count, the newest records overwrite the oldest records. Roll ensures that the latest data are recorded. Stop terminates recording when the number of stored records reaches capacity. Full policy has no effect if Capacity is Unlimited.
On Demand Poll Enabled	true (default), false	Determines user control over polling. true enables a system user to use the **Live Updates** (play) button in History views to poll for live data for the associated imported history(ies). false renders this button unavailable in history views for the associated imported history(ies).
On Demand Poll Frequency	Fast, Normal, Slow	References the On Demand Poll Scheduler rates under the **NiagaraNetwork**'s History Policies container slot. Configures how often this type of poll occurs.
System Tags	Text	Specifies one or more text strings matched against text values in "System Tags" properties of local history extensions (modifiable only for a NiagaraSystemHistoryExport descriptor). Matching text patterns result in histories exported into the remote **History** space.

Discovered selection notes

In the **Niagara History Import Manager**, discovered station histories are under an expandable tree structure organized by station name.

Figure 3. Expand stations to see all histories



Histories under the same station name as the parent Niagara Station (device) component are local histories for that station. Histories under any other stations represent histories that are either imported into (or exported to) that station.

For example, discovered histories (shown in the above image) for Niagara Station subCONT_A include local histories (expanded, top); another imported history from remote station, subCONT_B, is shown below.

Note: From any Niagara Station, you can import both its local histories and already-imported histories, as needed. However, unless circumstances warrant a “relay archive method,” it may be best to import histories directly from the source station whenever possible.

Parent topic: History plugins

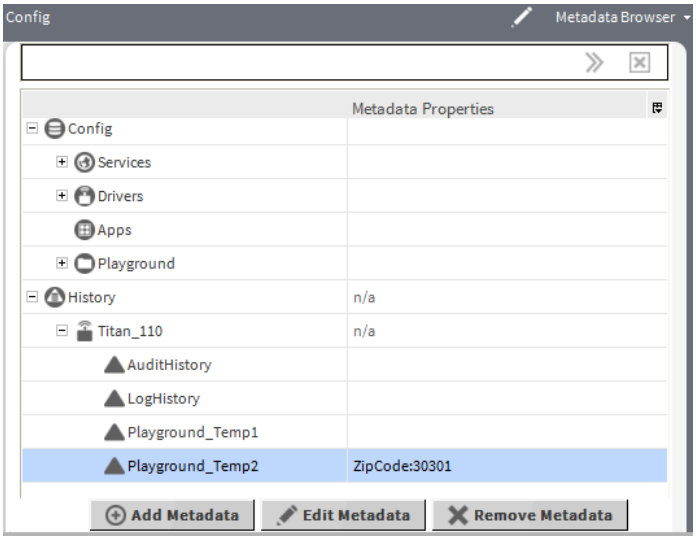
Metadata Browser view

This view is available for working exclusively with metadata properties on components and histories. The interface provides a convenient way to add metadata slots to one or more components or histories. Although you can add metadata slots directly to a component or a history from the slot view, the **Metadata Browser** has the ability to add metadata slots to thousands of objects in one batch job.

The interface (shown in the following image) is a view on the root station component. It filters and navigates through your histories or components so that you can selectively batch-add metadata to all the desired objects. The view runs a job for the metadata operation and reports the results in a job log. If a job runs and is unsuccessful at any point, the job log displays the reason for the failure.

Note: An example of a failure might be that you are trying to add a slot type to an object that does not support that particular slot type. This exception should be indicated in the job log.

Figure 1. Metadata browser view



The following areas comprise the main parts of the **Metadata Browser** view:

The **Job Log** pane displays after you run any of the editing control jobs to add, edit, or remove metadata. Click the >> icon to view the log or click the X icon to remove the most recent job log record.

The Metadata Properties table has two adjustable-width columns:

- The Navigation column provides a navigation tree with root nodes at the Config (station) level and at the History level. The navigation column displays an expandable tree structure representation of the selected station. Expanding a parent node displays additional rows that you can select for editing metadata.
- Metadata Properties displays a summary string of the properties. The metadata property name and its corresponding value are separated by a colon (:) and individual metadata properties are separated by a bar (|). For example: `ZipCode:303011`.

  Note: The history device object always displays an `n/a` in this column because it does not support the use of properties.

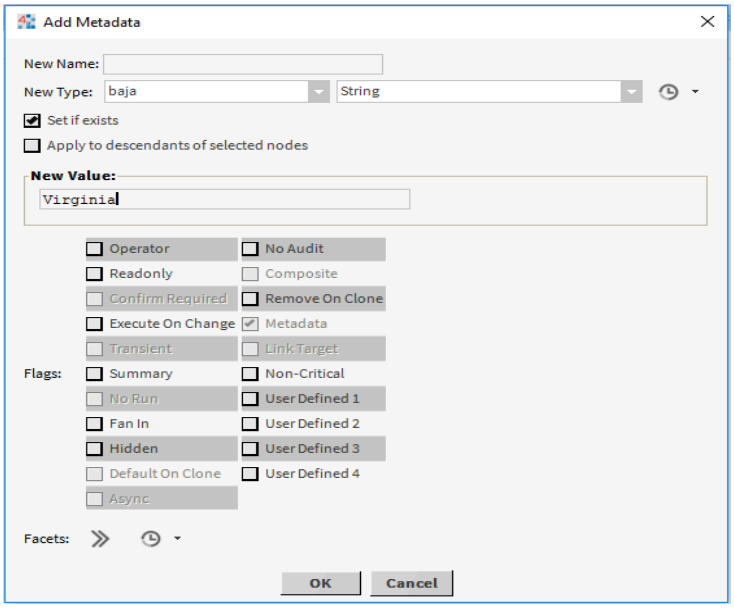
The Editing controls pane contains the following buttons that open window boxes that you can use to add, edit, and remove metadata.

Buttons

- **Add Metadata** opens the **Add Metadata** window.
- **Edit Metadata** is available only when you select an object that already has one or more metadata properties. Clicking this button opens the **Edit Metadata** window.
- **Remove Metadata** is available only when you select an object that already has one or more metadata properties. Clicking this button opens the **Edit Metadata** window.

Add Metadata window

Figure 2. Metadata window



Properties available in this window vary according to the type of property that you select. You can select multiple rows in the Metadata Browser table by using the **Shift** or **Ctrl** keys to select a range or a non-contiguous set of rows, respectively.

Property	Value	Description
New Name	text	Defines a name for the metadata item.
New Type	two drop-down lists (defaults to string)	Select the property type that you want to add.
Set if exists	check box	Choosse to edit an existing property value, if it already is present in or under the selected object(s). If this option is not selected, a new property may be added but an existing property is not edited when the Add Metadata job runs.
Apply to descendants of selected nodes	check box	Runs the Add Metadata job on the selected object and all of its descendant objects. If the option is not selected, only the selected object is edited.
New Value	text	Sets an appropriate value for the property type. For example, a string property displays for the default string type and allows you to enter the string value that is used for the property.
Flags:	check boxes	In addition to the default (required) Metadata flag option, other flags are available.
Facets	additional properties	Use this property to add facets to the slot, if needed, for the chosen property type and use.

Edit Metadata window

This window provides a Property option list that displays all metadata properties assigned to the object and allows you to select the property that you want to edit.

Property	Value	Description
Apply to descendants of selected nodes		Applies any edits you make to this property to all descendant objects under the selected object.
New Value	text	Edits the selected metadata property by specifying a new value for it.

Remove Metadata window

This window provides a Property option list that displays all metadata properties assigned to the object and allows you to select the property that you want to edit.

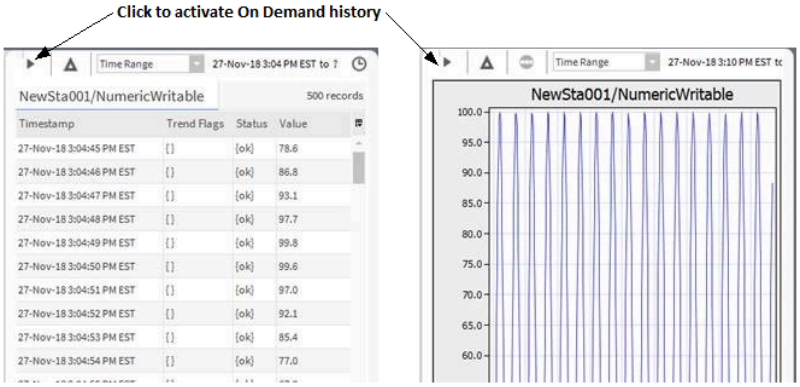
Property	Value	Description
Apply to descendants of selected nodes		Removes this property from all descendant objects under the selected object.

Parent topic: History plugins

On Demand History view

The On Demand history feature enables polling of both local history sources and Niagara History imports for live data when displaying history chart or history table views. This feature works in addition to, and does not replace, a standard polling schedule. For example, you would typically still have your History Import descriptors scheduled to archive at some daily interval (such as every night), even though you might be displaying **On Demand** history views.

Figure 1. On Demand history example view



The On Demand feature provides a **Live Updates** toggle button (check box in Hx view) on both the **History Chart** and **History Table** views. When you click this button, it initiates a history subscription that finds the source of the history and subscribes to that source component. If the source is a history import descriptor, that descriptor starts polling at a frequency defined by its On Demand Poll Frequency property (described below). If the source is a local History Ext, the history chart or table updates when the history extension appends a new record to the

history.

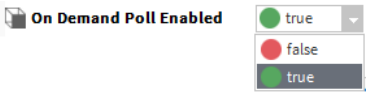
Note: When the On Demand History is active, a timestamp appears in the view's status bar (lower corner) to indicate the time of the last successful update. This does not apply to Hx views.

Since histories eligible for on-demand support are sourced directly from a local history extension or at a history import descriptor, on-demand History availability and update rate are affected by properties described below and shown below:

On Demand history is not available if the source component is not accessible. Since histories can be categorized (for permissions levels) independently from their source components, it may be possible for a user to have access to a history, but not have access to the source component. In this case, the On Demand History feature is disabled (button is dimmed on the **History Chart** or **History Table** view). Also, if the source component is no longer in existence (has been deleted) the **Live Updates** toggle button is dimmed (inaccessible).

On Demand History is not available if the source history import descriptor's On Demand Poll Enabled property is set to false. On Demand Polling may be disabled to limit bandwidth usage.

Figure 2. On Demand Poll setting in the History Import Manager view



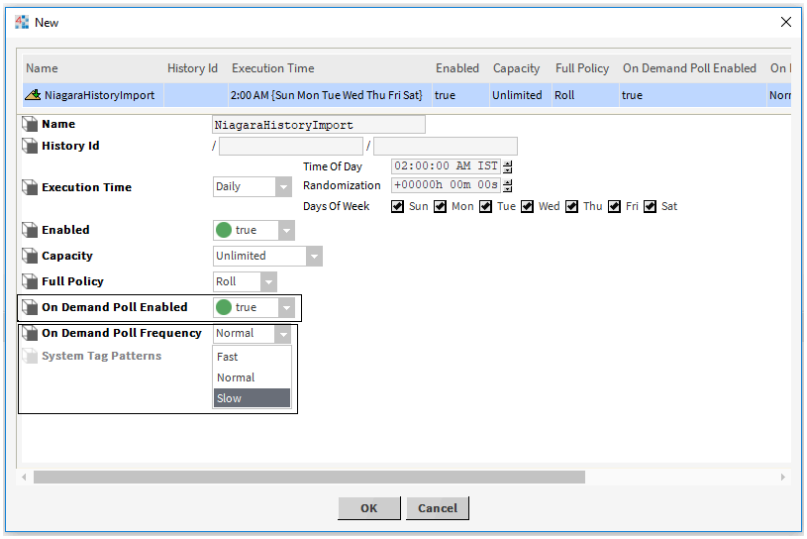
On Demand History is not available if the On Demand Poll Scheduler property is disabled (or the driver network does not have it included in a HistoryNetworkExt)

On Demand History rates are affected by the following property settings:

- On Demand Poll Scheduler (sources using import descriptor): This property is located under the **NiagaraNetwork** History Policies property. You can set standard polling rate values for a Fast Rate, Normal Rate, and a Slow Rate in the properties displayed in the **Property Sheet** view. These rates are available as selection options for individual Niagara History Import descriptors under the **NiagaraNetwork** History Device Extensions.

- On Demand Poll Frequency (sources using import descriptor): This property is located under the **Niagara History Import** property. Select one of the three options (defined in the On Demand Poll Scheduler): Fast, Normal, or Slow.

  Figure 3. On Demand Poll frequency settings for import descriptor-sourced views



- History extension update interval: If the On Demand history is being sourced from a local history extension, then the On Demand chart or table view updates whenever the local history extension appends a new record to the history. This function reflects the presence of a (Last Record) property.

Parent topic: History plugins