INTERACTIVE GLOSSARY

for the
Niagara^{AX} Technical Certification Program CBT

Click anywhere to continue.

Close this glossary at any time by clicking the ESCape key on your keyboard.

GLOSSARY

for the
Niagara^AX Technical Certification Program CBT

NAVIGATION ICONS

Use these icons when they appear to navigate through this glossary.

A to G
Letter buttons

Home Page

Index

Move Backward

Move Forward

Return to Previously Viewed

QUIK-REVIEW
Open a related Quik-Review

HOME PAGE

START HERE BY CLICKING ON THE
LETTER RANGE
FOR THE TERM YOU WISH TO LOOK UP.

YOU MAY ALSO CLICK ON
INDEX

OR ON
QUIK-REVIEW
FOR A LIST OF BRIEF OVERVIEWS.

A to G

H to M

N to R

S to Z

INDEX

QUIK-REVIEW

These buttons are "live" throughout this Glossary.

GLOSSARY

A to G

- **Action** – a "slot" that defines a component's behavior that may be invoked either by a user command or an event
- **Alarm extension** – an extension used to monitor off-normal values and show alarm indication when a limit or value is met or exceeded
- **Analog** – of a system of measurement in which a continuously varying value, as sound, temperature, etc., corresponds proportionally to another value, esp. a voltage.

  A typical analog device is a clock where the hands move continuously around the face. This type of clock is able to indicate every possible time of day. In contrast, a digital clock is able to represent only a finite number of times (every tenth of a second, for example). In general, people experience the world analogically. Vision, for example, is an analog experience because we perceive infinitely smooth gradations of shapes and colors.

  In Niagara, analog values are represented by numeric objects, which are color-coded PURPLE.

QUIK-REVIEW

H to M

N to R

S to Z

GLOSSARY

A to G

- ***ASCII*** – American Standard Code for Information Interchange. ASCII is a code for representing English characters as numbers, with each letter assigned a number from 0 to 127.

  For example, the ASCII code for uppercase M is 77. Most computers use ASCII codes to represent text, which makes it possible to transfer data from one computer to another.

- **BAJA** – Building Automation Java Architecture.
- **Binary value** – A value that can have only 2 possible states, such as:
  - ✓ ON/OFF
  - ✓ YES/NO
  - ✓ OPEN/CLOSED
  - ✓ *OCCUPIED/UNOCCUPIED*

  Binary values are typically shown as a 1 (True) or 0 (False). In the binary number system, there are only these two values. In Niagara, binary objects are color-coded GREEN.

QUIK-REVIEW

H to M

N to R

S to Z

GLOSSARY

A to G

- **.bog file** – (Baja Object Graph) a special file that describes components in a database. It can be a complete database or any collection of components.
- **Boolean object** – represent a binary value with only 2 possible states, typically coded as a TRUE or FALSE condition. [Color-coded GREEN]

  QUIK-REVIEW

- **Child** – a descendant; a property or component that is subordinate to another property or component (the parent) in a hierarchy.

  For example: every Boolean Writeable (parent) object is equipped with a Proxy Extension (child) that appears as a separate object on the wiresheet.

- **Commissioning** – the process used by Niagara to configure a new JACE when placing it into service. QUIK-REVIEW
- **Communication protocol** – a standard way of communicating across a network. A protocol is the "language" of the network; a method by which two dissimilar systems can communicate. TCP is a protocol which runs over a network.

H to M

N to R

S to Z

GLOSSARY

- **Component** – a special class of objects that are primary building-blocks of the Niagara framework used to assemble applications with graphical programming tools. QUIK-REVIEW QUIK-REVIEW
- **Composite** – the process by which lower level properties (*child slots or components) can be exposed on* the glyph (object shape) of a higher level parent (composited) object. This can simplify linking and promote reuse of control logic.
- **config_backup file** – a backup copy of a station's configuration database file, usually one generation older than the current config.bog.
- **config.bog file** – the configuration database file for a Niagara^AX Station. (Located in Station / Files container) An .xml file, this file describes and organizes the components that make up the station.
- **Container** – a type of component used to logically group (store) other components.

GLOSSARY

- **Control extension** – one of three extensions that are found in the Control Module. The three control extensions include:
  - ✓ *Proxy Extension*
  - ✓ *Discrete Totalizer Extension*
  - ✓ *Numeric Totalizer Extension*
- **Control point** – a normalized data point that represents a particular system variable. The control point is used to display real-time status of the variable, and may also be used to implement operator control of a corresponding system component.
- **Device driver** – a program that controls a particular type of device that is attached to a Niagara platform (host); a set of software routines that work with and control a specific hardware device.

H to M

N to R

S to Z

GLOSSARY

A to G

- **Discovery (Learn) function** – allows you to find items that are defined using a device driver's framework; online "device learns" are possible using the *Device Manager* for many drivers.

  For example: NiagaraNetwork, BacnetNetwork, LonNetwork, and NdioNetwork

  Most device learns in NiagaraAX are a two-step process where you first:

  ✓ Discover device candidates for inclusion in the station database.

  ✓ Select and Add from those candidates, creating device components in the network.

QUIK-REVIEW

GLOSSARY

A to G

- **Driver** – a software program used to enable the Niagara^AX framework to communicate and interact with specific external devices and networks.
- **Enumerated object** – represents multiple states (more than one) such as a multi-speed fan or pump. Unlike Boolean objects that can only have 2 states, enumerated objects can have multiple (more than two) states, such as:
  - ✓ *OFF / SLOW / FAST*
  - ✓ *HIGH / MED / LOW*
  - ✓ *POSITIVE / NEUTRAL / NEGATIVE*

  In Niagara, multiple states are represented by enumerated objects, which are color-coded ORANGE.

QUIK-REVIEW

H to M

N to R

S to Z

GLOSSARY

A to G

- **Extension** – additional building blocks that "extend" and change the behavior of an individual control point.

  Point Extensions:
  - ✓ *are added as dynamic properties (slots) on a control point*
  - ✓ *process and modify the value of a control point whenever it executes*
  - ✓ *are always invoked in the order they are declared*

- **Facet** – contains additional data used to modify the presentation of a value. "Unit of Measurement" is one example of a facet. Facets for a data point are edited from the object's Property Sheet view.

- **Framework** – something composed of parts fitted together and united; a structural frame, a basic structure; Niagara^{AX} uses a universal software framework targeted at solving the challenges associated with modern smart devices.

- **Glyph** – a visual representation of a component on the Wire Sheet.

H to M

N to R

S to Z

GLOSSARY

- **Graphical user interface (GUI)** – a type of user interface which allows people to interact with system components. A *GUI* offers graphical icons, and visual indicators, as opposed to text-based interfaces or command labels, to fully represent the information and actions available to a user. The actions are usually performed through direct manipulation of the graphical elements.

GLOSSARY

H to M

- **History extension** – an extension used for any data point for which you want to log historical and/or trend data, typically for changes of value or status and on a repeating time interval. *[Found in the history palette.]*
- **Host** – any workstation or JACE on a *network* that is a repository for services available to other workstations or JACEs on the network, with each host having a unique IP address – in the Nav Tree, the host is represented by a computer icon and IP address. In Niagara, the terms "host" and "platform" are often used interchangeably.
- **Integration** – also known as systems integration: the use of software and computer systems to bring together a set of enterprise computer applications or devices. Allows data from one device to be read or manipulated by another, resulting in ease of use.

A to G

N to R

S to Z

GLOSSARY

- **Learn mode** – a mode that enables Niagara^{AX} to learn (or “discover”) which data points are available for manipulation. Online "point learns" are possible in some driver networks:

  For example: NiagaraNetwork, BacnetNetwork, LonNetwork, and NdioNetwork

  Whenever available, this method is the easiest way to accurately add proxy points to the station database. QUIK-REVIEW

- **Make Widget Wizard** – a tool that automates and simplifies the process of adding graphic visualizations (widgets) to a Px file, is available whenever Px Editor is in the active view, and automatically launches when you drag a component onto the Px Editor canvas pane.

GLOSSARY

N to R

- **Navigation (Nav) Tree** – a structural parent/child view of the entire Niagara system formatted as a hierarchy tree, and located as a Navigation Sidebar in the Left-Sidebar Pane. All files, containers and component views are easily accessible with a point and click of the mouse.
- **Niagarad** – the name of an executable file that activates whenever the Niagara platform daemon is installed and started.
- **Normalize** – to cause to conform to a standard or norm; the Framework takes the data elements from the various devices - inputs, outputs, setpoints, schedules, control parameters, etc. - and processes these items into "normalized" software components. This conversion normalizes the attributes of the devices (both data and behavior), creating a database of objects that talk to and work coherently with each other in real time. QUIK-REVIEW
- **Numeric object** – represents an analog value such as a temperature, current, rate (or similar floating point number), or varying count (integers → 1, 2, 3, etc.) In Niagara, analog values are represented by numeric objects, which are color-coded PURPLE. QUIK-REVIEW

A to G H to M S to Z

GLOSSARY

- **Object** – a discrete item that can be selected and manipulated, such as a read-only Boolean point or a writable Numeric point. In object-oriented programming, objects include data and the processes necessary to operate on that data. The Niagara^{AX} common object model consists of a collection of 8 basic object types, while the Niagara^{AX} component model consists of many other specialty objects. The terms Object and Component are interchangeable.
- **ORD (Object Resolution Descriptor)** – universal identification system used throughout the Niagara framework to identify any resource; unifies and standardizes access to all information. With an ORD, you can refer to the precise location of any object, file, view, or other resource.
- **Palette** – a module-specific library of components used to implement and customize that module's core functionality. These components are copied out of the palette that is accessible through the Palette Sidebar.

  For example, the Alarm Module contains an Alarm Palette that contains a collection of standard alarming system components required for the configuration of a Niagara station's Alarm Service.

A to G H to M S to Z

GLOSSARY

N to R

- **Parent** – a higher-level object that contains other lower-level (child) objects, and which often contributes substantially to the structured environment within which the child object co-exists.
- **Parent/Child relationship** – In some cases, a critical parent-child relationship imposes certain restrictions on lower level objects. For example, a child object often inherits its security classification from its parent. In another example, a Bacnet device can be a child of a Bacnet network, but cannot exist as a child of a LonWorks network.
- **Platform** – the hardware and all software components installed on a Niagara^AX host that is not part of a Niagara^AX station. QUIK-REVIEW
- **Platform daemon** – a compact executable program (or server process) that enables a user to access the platform tools through the Workbench. The Niagara platform daemon runs in the background, and when the Workbench requests access, acts like a traffic cop to either grant or deny access to the appropriate platform tools.

A to G H to M S to Z

GLOSSARY

N to R

- **Platform tools** – a set of Workbench utilities available through a platform connection that are used to configure settings on a Niagara^AX host. These tools are available as functional "views" in the Workbench View Pane. Different tools are available, depending on whether the platform is local or remote.

QUIK-REVIEW

Local Host Platform

Platform	
Name	**Description**
Application Director	Control applications and access console output
DDNS Configuration	Configure the way DDNS operates.
Dialup Configuration	Configure the way the remote host uses dialup networking
Lexicon Installer	Install lexicons to support additional languages
License Manager	Manage licenses and certificates
Platform Administration	Update the platform daemon's port or credentials, or set its date and time
Software Manager	Install software to the remote host
TCP/IP Configuration	Manage the host's TCP/IP settings
User Manager	Manage the local OS users and groups
Remote File System	The remote host's file system

Remote Host Platform (JACE)

Platform	
Name	**Description**
Application Director	Control applications and access console output
DDNS Configuration	Configure the way DDNS operates.
Dialup Configuration	Configure the way the remote host uses dialup networking
Distribution File Installer	Install distribution files to the remote host
File Transfer Client	Transfer files to and from the remote host
Lexicon Installer	Install lexicons to support additional languages
License Manager	Manage licenses and certificates
Platform Administration	Update the platform daemon's port or credentials, or set its date and time
Software Manager	Install software to the remote host
Station Copier	Transfer stations to and from the remote host
TCP/IP Configuration	Manage the host's TCP/IP settings
Remote File System	The remote host's file system

A to G

H to M

S to Z

GLOSSARY

N to R

- **Pop-up menu** – a list of choices; the Niagara Framework uses both pull-down menus and popup menus which appear when the mouse interacts with the Nav Tree as well as various component views.
- **Properties** – the visible data associated with a component, and are accessed on the Property Sheet View; they provide the primary means of interacting with components.
- **Property Sheet** – the default view for the Config container (component space) and most components in the component space; shows the facets, outputs, inputs, and extensions for each component (or at the folder level – shows all components in a folder).
- **Protocol** – a standard way of communicating across a network. A protocol is the "language" of the network; a method by which two dissimilar systems can communicate. TCP is a protocol which runs over a network.
- **Proxy extension** – a child component of every extendable object. The proxy extension indicates how the point's data originates, including details specific to the parentage of the point's network and communications (driver).

A to G H to M S to Z

GLOSSARY

- **Px Editor** – a Px view that enables you to build Px files, and create the desired visualization of your control logic without programming skills.
- **Px View** – a custom graphical view that you define in a Px file; the purpose of a Px view is to provide a visualization of information in a rich, dynamic format that is easy to create and to edit. When attached to a component, the Px View becomes the default view (1st on the View menu list) for the component.
- **QNX** – a commercial Unix-like real-time operating system, aimed primarily at the embedded systems market.

  As a *microkernel-based* OS, QNX is based on the idea of running most of the OS in the form of a number of small tasks, known as *servers*. This differs from the more traditional *monolithic* kernel, in which the operating system is a single very large program composed of a huge number of "parts" with special abilities.

GLOSSARY

S to Z

- **Side Bar Pane –** the Workbench interface may be customized by adding unique side bars that are designed to fit particular applications:
  - ✓ ***Bookmarks* side bar** - *displays a list of bookmarks*
  - ✓ ***Help* side bar** - *provides a tree view of available help documentation*
  - ✓ ***Jobs* side bar** - *provides a tree view of available help documentation*
  - ✓ ***Navigator* side bar** - *provides a tree view of the system*
  - ✓ ***Palette* side bar** - *provides a tree view of specific module palettes*
  - ✓ ***To Do* side bar** - *provides a customizable list of tasks or notes*

- **Slot –** building block for defining components in terms of properties (what their characteristics are) and actions (how they behave); a means to assign various properties and actions to a Niagara object or component.

- **Slot Sheet –** used to view all of the "slots" of a component. Here slots may be added, deleted, renamed, reordered or configured.

- **Space –** defines a group of objects that share common strategies for loading, caching, lifecycle, naming and navigation.

  Examples of spaces in the Workbench are – the file space (under the Files container) and the component space (under the Config container).

A to G

H to M

N to R

GLOSSARY

S to Z

- **State** – a condition or mode of being; in Niagara, multiple states are represented by enumerated objects, which are color-coded ORANGE.
- **Station** – the main unit of server processing in the Niagara architecture. There is usually a 1-to-1 correspondence between a station and a host machine. (It IS possible to run 2 stations on the same machine if they are configured to use different ports.) *(A station is NOT the hardware.)* QUIK-REVIEW QUIK-REVIEW
- **String object** – represents one or more ASCII characters, often with literal meaning. QUIK-REVIEW
- **View Pane** – Displays additional information and details about whatever is selected in the Navigation tree, and can display a number of different views (e.g., Wire Sheet, Property Sheet, Slot Sheet, Px Editor, etc.). QUIK-REVIEW
- **Widget** – a component that provides for graphic visualization; you can use the Px Editor to work with widget properties in defining user interface functions for control and information display.

A to G H to M N to R

GLOSSARY

S to Z

- **Wire Sheet** – a view that graphically shows devices and objects (as “glyphs”) and links between them as wires; this is the default view for all folders in the Config container (component space).
- **Workbench** – a Windows-based engineering tool used for configuration of the Niagara^{AX} Framework. The Workbench has features of a file explorer and a computer-aided design (CAD) application. It allows Niagara^{AX} installation or maintenance professionals to graphically review and edit the contents and behavior of a Niagara^{AX} station, as well as the configuration of a Niagara platform -- the computer on which the station is running. QUIK-REVIEW
- **WorkPlace^{AX}** – Tridium's branded version of the Workbench; it provides an integrated development environment (IDE) for non-programmers to develop their own customized applications.

GLOSSARY INDEX

Click on the letter range below for the term you wish to look up.
On each page, click on the specific term you wish to see.

QUIK-REVIEW

GLOSSARY INDEX

A to G

H to M

N to R

S to Z

GLOSSARY INDEX

A to G

H to M

N to R

S to Z

GLOSSARY INDEX

A to G

H to M

N to R

S to Z

GLOSSARY INDEX

A to G

H to M

N to R

S to Z

QUIK-REVIEW

Click on the Quik-Review below you wish to see.

To close that Quik-Review, just click the X in the upper right corner.

The associated course module is shown in parenthesis.

- Application Director Functionality (6.2.1)
- Commissioning Process Overview (6.3)
- Component Naming Conventions (7.3)
- Connecting to a Platform – Process Chart (5.4)
- Connecting to a Station – Process Chart (5.5)
- Connection Authentication Services (5.5)
- Discovery Process (2.2)
- Drag & Drop Components (7.3)
- Establishing Station Connections (5.5)
- File Transfer Client vs. Station Copier (6.2)
- Installing a License (6.2.7)
- JACE Equipment (3.3)
- JACE Support Data (3.3)
- Key Features (4.1)
- License Manager Functions (6.2.7)
- Logoff-Disconnect-Close-Exit (7.5)
- Niagara^AX Building Blocks (4.2.9)
- Niagara Objects (2.2)
- Normalization (2.2)
- Platform Connections – Nav Tree (5.4)
- Platform Service Views (6.1)
- Platform Tools (6.2)
- Platform vs. Station Connections (5.3)
- Selecting Component Views (7.3)
- Station Connections – Nav Tree (5.5)
- Working with Components (7.3)
- Workbench Keyboard Equivalents (7.5)
- Workbench Toolbar Equivalents (7.5)