

AX and Sedona Training in KL, Malaysia, 21-25 March 2011

Infocon HQ Training Room



Technical Certification Program

Student Guide

Day 1

Installation and Start-up
Fundamentals
Niagara AX Workbench Interface
Customizing the Workbench Environment
Security, User Administration and Licensing

Day 2

Data and Control Model
KitControl
Schedule Objects

Day 3

Driver Architecture
Field Bus Integrations
Histories
Alarms

Day 3

Graphics

Day 3

Certification Test

Copyright Notice: The software described herein is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

© Copyright 2005 Tridium, Inc. All rights reserved.

This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc., 3951 Westerre Parkway, Suite 350, Richmond, Virginia 23233.

The confidential information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and is not to be released to, or reproduced for, anyone else; neither is it to be used for reproduction of this Control System or any of its components.

All rights to revise designs described herein are reserved. While every effort has been made to assure the accuracy of this document, Tridium shall not be held responsible for damages, including consequential damages, arising from the application of the information given herein. The information in this document is subject to change without notice.

The release described in this document may be protected by one of more U.S. patents, foreign patents, or pending applications.

Trademark Notices: Microsoft and Windows are registered trademarks, and Windows 95, Windows NT, and Internet Explorer are trademarks of Microsoft Corporation. Java and other Java-based names are trademarks of Sun Microsystems Inc. and refer to Sun's family of Java-branded technologies. Communicator and Navigator are registered trademarks of Netscape Communications Corporation. Echelon, LON, LonMark, LonTalk, and LonWorks are registered trademarks of Echelon Corporation. Tridium Niagara, the Niagara Framework, WorkPlace Pro, Web Supervisor, and JACE-NP are trademarks of Tridium Inc. All other product names and services mentioned in this publication that are known to be trademarks, registered trademarks, or service marks have been appropriately capitalized and are the property of their respective owners.

Day 1

Installation and Start-up

This section covers installing Workbench software and software necessary on the JACE.

Installing Niagara AX

- Installing Workbench/Web Supervisor (PC)

Installing JACEs (JACE Install & Startup Guide, Tab 2)

Pg 1

- JACE-4/5 Series (Embedded models)
- JACE-NX/NU Series (Win32 models)

Licensing

- License request dialog box

Fundamentals

Basic Niagara concepts, such as architecture and terminology are addressed here.

Niagara architecture (AX User Guide, Tab 3)

Pg 6

- Software
- Network

Niagara AX building blocks

Pg 8

- Modules
 - Module characteristics
 - Module benefits
- Components
 - Slots
 - Master/slave components
 - Point components
- Palettes
- Stations
- BOG Files

Niagara AX terms

Pg 485

- ORDs
- PX
- Views
- Stations

Niagara Platform (Niagara Platform Guide, Tab 4)

Pg 1

- Distribution file installer
- File transfer client
- Lexicon installer
- License manager
- Module manager
- Resource Manager
- Platform administration
- Commissioning a host
- Rebooting a JACE
- Station copier
- Station director
- TCP/IP configuration
- User Manager
- Remote File System

Types of tools available with Workbench

Pg 327

- New module wizard
- New station wizard

Niagara AX Workbench Interface

Pg 25

When you start Workbench, you will see the Workbench screen.

The primary window of the Workbench GUI is divided into three main areas:

- Niagara Workbench Window
- Side Bar Pane
- View Pane
- Console

Basic Workbench Tasks

Pg 335

- Start Workbench
- Close a Workbench window
- Exit Workbench
- Open a Platform
- Connect to a Platform
- Close a Platform
- Disconnect from a Platform
- Start a Station
- Stop a Station
- Open a Station
- Close a Station
- Connect to a Station
- Disconnect from a Station
- New Driver Wizard

Workbench Window Controls

Pg 25

The Workbench window provides typical Windows-type controls plus other features unique to Niagara.

You may create additional windows after starting Workbench—all have these basic features:

- Title bar
- Borders
- Scroll bars
- Status bar

Workbench Menus

Pg 460

The menus provide one of the options to access the features of the system. The menu provided is based on the current Selection and mode.

- File Menu
- Edit Menu
- Search Menu
- Tools Menu
- Window Menu
- Help Menu

Interface Tool Bars

Pg 38

- The Toolbar
- The Locator Bar
- The View Selector

The side bar pane is the first area on the left, below the locator bar. The side bar pane is normally visible only if a side bar is open.

❑ Side Bar Pane Tasks

Some tasks are shared by all types of Side bar Pane types. This section explains tasks that apply to all Panes and the Side Bar pane as a whole.

- Opening the Side Bar
- Basic Tasks
- Closing the Side Bar(s)
- Resizing the Side Bar

❑ Nav Side Bar Pane

The Nav side bar contains the tree view that provides a hierarchical view of the whole system.

- Opening the Nav Side Bar Pane
- Using the Nav Side Bar (Tree) Pane
- Nav Side Bar Toolbar
- Nav Side Bar Tree Nodes
 - Host Node
 - Module Node
 - File system Node
 - Station Node
 - Config Node
- Refreshing Tree Nodes
- Go Into Command

❑ Help Side Bar Pane

Niagara Workbench provides context-sensitive help. When help is accessed from the Help menu from the Workbench menu bar, the Help interface opens in a new window. Help accessed from the Help Side Bar Pane will open in the View Pane.

- Opening the Help Side Bar Pane
- Using Help

❑ Bookmarks Side Bar Pane

Bookmarks are simply linked "shortcuts" or "favorites" to help you quickly find views in just a couple of clicks. The bookmark side bar provides a convenient place in the side bar pane to add, manage, edit, and remove bookmarks.

- Opening the Bookmarks Side Bar Pane
- Adding Bookmarks
- Managing Bookmarks
- Editing Bookmarks
- Removing Bookmarks

❑ Jobs Side Bar Pane

The Jobs Side Bar Pane shows all the current jobs in all the stations with which you have a connection.

- Opening the Jobs Side Bar Pane
- Jobs Pane Tasks

Palette Side Bar Pane

The palette side bar provides a place to open and view sets of modules. From the palette side bar, you can open multiple palettes (displaying them one at a time), close palettes and view modules within palettes.

- Opening the Palette Side Bar Pane
- Palette Side Bar Tool Bar

 Todo List Side Bar Pane

The Todo list side bar lists action items for your Niagara^{AX} projects by category.

- Opening the Todo List Side Bar Pane
- Adding action items to a Todo list.

The View Pane

Pg 36

The View pane is the largest window area and is located on the right side, below the locator bar. Features of the view pane include tabbed views and a thumbnail view.

Each object that you select (in a view or in the nav tree) has a “default” view that appears in view pane. You can select a different view of the object using the view selector or the right-click menu. The last view of an object becomes its default view during a workbench session. If Workbench is closed and restarted, default views are reset to their original default views.

- Tabbed Views
- The Thumbnail View
- Property Sheet
- Wire Sheet
- Graphic View
- Slot Sheet
- Link Sheet

The Console

Pg 37

The console is located along the bottom of the Workbench interface and may be alternatively hidden or shown by selecting Window: Hide Console or Window: Console in the menu bar.

The console has scroll bars on the right side and the window size may be adjusted by dragging the top border bar. The console provides you with access to a command line prompt without leaving the Workbench environment. From the console you may type in commands directly, including the help command for additional help.

- Using the Console
- Hide Console
- Active Plugin

Customizing the Workbench Environment

Pg 44

You can customize your Workbench GUI as well as many of the settings in the Workbench environment. Some settings require an acknowledgement via the OK button on a dialog box, while others are invoked immediately and saved automatically on exit from Workbench. For example, when you exit Workbench with four tabbed windows in your view pane, those same four windows will be displayed the next time you open Workbench.

Managing Tabs In The View Pane

Pg 45

- Adding Tabs
- Navigating Tabs
- Closing Tabs

Customizing the New Menu

- Adding Options
- Organizing Options

Workbench options

Pg 46

Use the Workbench options dialog box to customize your Workbench GUI and to set other preferences.

- Alarm console options
- Alarm Class Mapping
- Alarm portal options
- Bajadoc options
- General Workbench options
- Wiresheet options
- Lexicon options

Assigning Name Maps

- Language codes
- Adding Name Maps

Browser Workbench

- Access
- Use

Security, User Administration and Licensing

Pg 271

This section reviews setting up users and user security.

Security Service

- ❑ Categories
 - Adding a Category
 - Modifying Category Settings
- ❑ Component Security
 - Permissions
 - Category assignments

User Service

- ❑ Simple User Service View
 - Adding a New User
 - Modifying a User
 - Deleting a User

Day 2

Data and Control Model

Pg 51

The control module provides normalized components for representing control points. All control points subclass from the ControlPoint base class. Control points are typically used with the Driver framework to read and write points in external devices.

Control points

Pg 51

- Types of control points
- Point value categories
- Point types
- Point Out
- Point actions
- Point facets
 - Facets importance for Enum points
 - Facets affect on point actions

Point extensions

Pg 62

- Extension process overview
- Types of point extensions
- Proxy extension
- Control extensions
 - Types of control extensions
- Alarm extensions
 - Types of alarm extensions
- History extensions
 - Types of history extensions

Control triggers

Pg 68

- How triggers are used

Point status

Pg 69

- Types of status flags
- Setting flags
- Priority of status indication

Proxy points

Pg 74

- Location of proxy points
- How proxy points are made
- Proxy points versus control points

Writable pointsPg 81

- Priority scheme
 - Priority input
 - Priority linking rules
 - Priority level conventions
- Minimum On and Off times

CompositesPg 83

- Some composite examples
 - Point-level composite
 - Folder-level composite
- Composite performance issues

KitControl (kitControl Guide, Tab 5)Pg 1

The kitControl palette is the primary palette that comes with the Niagara AX software.

kitControl components

-
- Application for kitControl components
 - Categories of kitControl components
 - Location for kitControl components

Extensions and kitControl components

-
- Components that cannot receive extensions
 - Conversions components
 - HVAC components
 - Latch components
 - Logic components
 - Math components
 - Select components
 - String components
 - Timer components
 - Util components

Schedule Objects

Pg 293

A deployable schedule that provides continuous data output at predetermined times.

Schedule Object Types

- Boolean Schedule
- Enumerated Schedule
- Numeric Schedule
- String Schedule
- Calendar Schedule
- Trigger Schedule

Creating Schedules

- Schedule Palette
- Weekly Schedule
- Special Events
- Properties
- Summary

Day 3

Driver Architecture**Pg 89**

In any Niagara AX station, one or more driver networks are used to fetch and model data values. Data is modeled with proxy points, lower-tier components in that driver's architecture.

To support a driver's proxy points, the station must have that driver's network architecture. This includes the upper-tier parent network component, and one or more child device components, each with device ext (extension) child components.

Network Architecture**Pg 90**

- Networks
- Devices
- Device Extensions

Driver Manager**Pg 93**

The Driver Manager is the default view for the Driver Container (showing all networks) in a station.

Common Network Components**Pg 95**

- Network status properties
 - Health
- Monitor
- Tuning Policies
- Additional Network components
 - Communication components
 - Poll components

Device Manager**Pg 105**

- Device New and Edit
- Device Learn and Discover
 - Device Match feature

Types of Device extensions**Pg 116**

- Points extension
- Histories extension
- Alarms extension
- Schedule extension

Point Manager**Pg 121**

- Points New and Edit
- Points Learn and Discover
 - Points Match feature
- New Folder for Points
- Other Points views

Histories Extension Views **Pg 132**

- Histories Import Manager
- Histories Export Manager
- Histories Retry Properties

Alarms extension properties **Pg 119**

Alarm extension properties define items such alarm enable (annunciation) transition types, alarm delay times, associated alarm class, and alarm display text for different transition types. You define the actual alarm limits or state(s) in properties in the extension's "Offnormal Algorithm" slot.

Schedules extension views **Pg 142**

- Schedules Import Manager
- Schedules Export Manager
- Schedules Retry properties

Field Bus Integrations **Pg 167**

For purposes here, a field bus integration is any Niagara AX driver besides the Niagara Driver (Niagara Network). All Niagara AX drivers resemble each other in basic architecture, including the Niagara Network.

However, field bus integrations such as Bacnet, Lon, Modbus, and others each have unique characteristics and features. This section provides an overview of topics related to these drivers.

Common field bus concepts

- Driver architecture framework
- Learn versus New devices and points

Niagara Network **Pg 149**

- Niagara Network components
 - Fox Service
 - Query Service
- Station Manager
 - Station Learn and Discover
- Niagara Station
 - Station status properties
 - Client connection properties
 - Server connection properties
- Bql Query Builder
 - Bql Query Find filters
 - Bql Query Match filters
 - Example Bql Queries
- Station Histories Import and Export
- Station Alarms Properties
- Station Schedules Import and Export

Modbus

- ❑ Modbus network types
 - Modbus Async networks
 - Modbus Slave networks
 - Modbus TCP networks
 - Modbus TCP Slave networks
- ❑ Master Types
 - Modbus proxy (client) points
 - Modbus Preset points
 - Device Level polling
 - Modbus Master register management
- ❑ Slave Types
 - Modbus Slave proxy (server) points
 - Modbus Slave register management

Lonworks

- ❑ LonNetwork Architecture
 - lonworks palette components
 - Poll Service
 - Lon Netmgmt
 - Local Lon Device
- ❑ Lon Network views
 - Lon Device Manager
 - Lon Router Manager
- ❑ Understanding Learn scenarios
 - An unmanaged network
 - A previously managed network
- ❑ Lon device objects
 - Device object properties
 - Device Data
- ❑ Lon proxy points
 - Lon Point Manager tips
- ❑ Managing Links (Lon Netmgmt)
 - Lon Links Manager: NetworkVariableLinks tab
 - Lon Links Manager: MessageTagLinks tab

Bacnet

- ❑ Bacnet Network Architecture
 - bacnet palette components
- ❑ Bacnet Comm
 - Bacnet Comm: Network
 - Bacnet Comm: Network: Ip Port
 - Bacnet Comm: Network: Router
 - Bacnet Comm: Network: Ethernet Port
 - Bacnet Comm: Network: MstpPort
 - Bacnet Comm: Client, Server, and Transport
- ❑ Bacnet Local Device
 - Local Device Properties
 - Time Synchronization Recipients
 - Utc Time Synchronization Recipients
 - Device Address Binding
 - Active COV Subscriptions
 - Bacnet Export Table
- ❑ Bacnet Tuning Policies
 - Bacnet Tuning Policy tips
- ❑ Bacnet Device Manager
 - Bacnet Find Parameters
 - Bacnet Device Learn Tips
- ❑ Bacnet Device object
 - Config Device object
- ❑ Bacnet Proxy Points
 - Bacnet Point Manager Tips
- ❑ Bacnet Device Schedules
 - Bacnet Schedule Tips
- ❑ Bacnet Trend Logs (Histories)
 - Bacnet History Import Manager
- ❑ Bacnet Config objects
 - Bacnet Config object tips

Histories

Pg 169

Histories are ordered collections of time-stamped records. Each history is the collection of specific data values from a component within the station or from another station. All history collections are identified by a unique id.

Histories are explained in the Niagara AX Documentation under Histories.

History Extensions

- Identifying extension types
 - Boolean
 - Numeric
 - Enumerated
 - String
- Adding Extensions
- Configuring Extensions

History Service

- Extension Manager
- History Service Property Sheet
- History Service Actions
 - Save DB
 - Close Unused histories
- Audit History Service
- Log history service

Viewing Histories

- View Types
- Viewing histories in Workbench
- Viewing histories in the Browser

History Maintenance

- Cleaning
- Deleting
- Archiving
 - Push
 - Pull

Alarms

Pg 191

The Alarm module provides core functionality for lifecycle management of alarms within the Niagara framework. Alarms are used to indicate that some value is not within an appropriate or expected range. Alarms may be routed from the system to a variety of external sources, be it email or a printer or a console application.

Alarming extensions

- Types
 - Out Of Range
 - Boolean Change Of State
 - Boolean Command Failure
 - Enumerated Change Of State
 - Enumerated Command Failure
- Adding
- Configuring

Alarm service

The Alarm Service uses Alarm Classes to route all Alarm messages between Alarm Sources and Alarm Recipients. Each Station contains a single Alarm Service.

Alarm Recipients

- Recipient Types
- Creating Recipients
- Configuring Recipients

Alarm Portal

- Configuring
- Using

Viewing Alarms

- Alarm Console
 - Customizing
 - Using
- Locally
- Remotely

Maintenance Alerts

- Viewing
- Managing

Batch editing

- Accessing the Editor
- Editing Values

Day 3

Graphics

Pg 217

The Niagara AX framework provides a powerful presentation architecture based on XML and the Niagara component model. Presentation is a term that describes how Niagara visualizes information (text, graphics, alarms, and so on) across heterogeneous media - such as: Workbench, desktop browsers, handheld devices, and so on. Niagara uses Presentation XML (px) to accomplish this.

Px

- Presentation views
- Graphic page

Page editor (PX editor)

- Menus
- Toolbar
- Popup Menus
- Bindings
- Properties

Web UI concepts

- Web Client
- Viewing Px pages