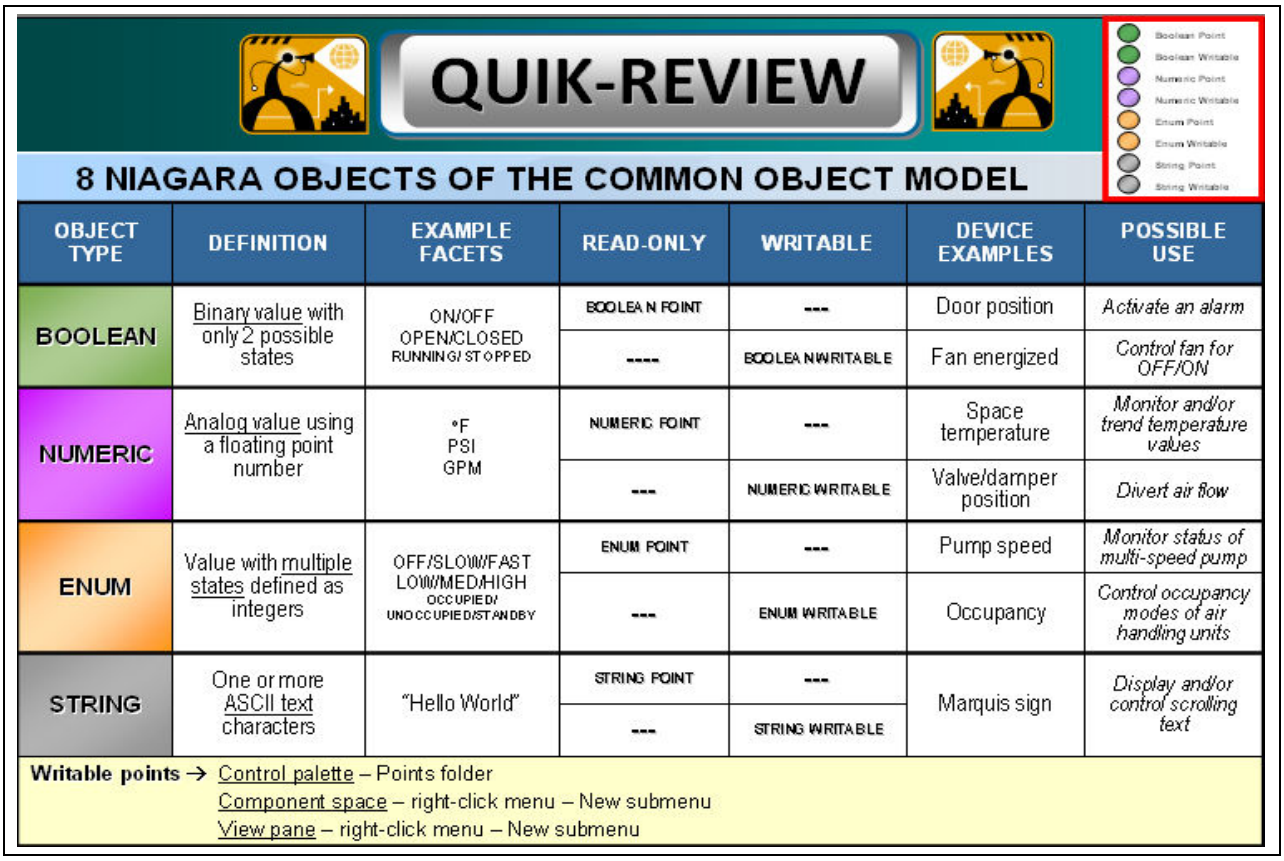
Common Object Model and Components

FROM: Topic 2.2, Normalization	
Common Object Model	The Niagara^{AX} Common Object Model makes it possible for diverse connected systems to talk to each other and to the enterprise. The Framework takes the data elements such as the following from the various devices: ▪ Inputs ▪ Outputs ▪ Setpoints ▪ Schedules ▪ Control parameters 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.
Niagara objects	The device drivers for a particular family of smart devices specify how individual data points map to a collection of **8 simple Niagara objects**. These data points can be any data from a remote object, such as: ▪ space temperature ▪ lighting status ▪ valve position ▪ pump speed The decision to establish a connection to one or more data points then becomes a matter of selecting which of these Niagara objects is best suited for the task. Those objects can then be configured and presented graphically to more easily monitor and control. These objects are **integral to the driver architecture** that is the foundation on which all device integration rests. They are either read-only or writable and are color-coded for ease of identification: ▪ **Boolean** - green ▪ **Numeric** - purple ▪ **Enumerated** - orange ▪ **String** - gray
Read-only	**READ-ONLY** objects have output only, and are used for **monitoring** only. They can be fed as inputs to writable objects.
Writable	**WRITABLE** objects represent data items that can be written (programmed) as well as read. - 16 different input properties correspond to priority levels such as *emergency* and *operator override*.

Common Object Model and Components

Boolean objects	Boolean objects represent a binary value with only 2 possible states, typically coded as a TRUE or FALSE condition. However, each condition can be shown in a way that is most meaningful for the application: ▪ **ON / OFF** ▪ **YES / NO** ▪ **OPEN / CLOSED** ▪ **OCCUPIED / UNOCCUPIED** *Boolean objects are color-coded GREEN.*
Numeric objects	Numeric objects represent an analog value such as a: ▪ Temperature ▪ Current ▪ Rate ▪ (or similar floating point number) *Numeric objects are color-coded PURPLE.*
Enumerated (Enum) objects	Enumerated objects (enum) represent multiple states (more than one) such as a multi-speed fan or pump (or varying count: integers → 1, 2, 3, etc.) ▪ **OFF / SLOW / FAST** *Enumerated objects are color-coded ORANGE.*
String objects	**String objects** represent one or more **ASCII characters**, often with literal meaning. **ASCII characters** *ASCII stands for **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange. 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. *String objects are color-coded GRAY.*

Common Object Model and Components



OBJECT TYPE	DEFINITION	EXAMPLE FACETS	READ-ONLY	WRITABLE	DEVICE EXAMPLES	POSSIBLE USE
BOOLEAN	Binary value with only 2 possible states	ON/OFF OPEN/CLOSED RUNNING/STOPPED	BOOLEAN POINT	---	Door position	*Activate an alarm*
			---	BOOLEAN WRITABLE	Fan energized	*Control fan for OFF/ON*
NUMERIC	Analog value using a floating point number	°F PSI GPM	NUMERIC POINT	---	Space temperature	*Monitor and/or trend temperature values*
			---	NUMERIC WRITABLE	Valve/damper position	*Divert air flow*
ENUM	Value with multiple states defined as integers	OFF/SLOW/FAST LOW/MED/HIGH OCCUPIED/ UNOCCUPIED/STANDBY	ENUM POINT	---	Pump speed	*Monitor status of multi-speed pump*
			---	ENUM WRITABLE	Occupancy	*Control occupancy modes of air handling units*
STRING	One or more ASCII text characters	"Hello World"	STRING POINT	---	Marquis sign	*Display and/or control scrolling text*
			---	STRING WRITABLE		

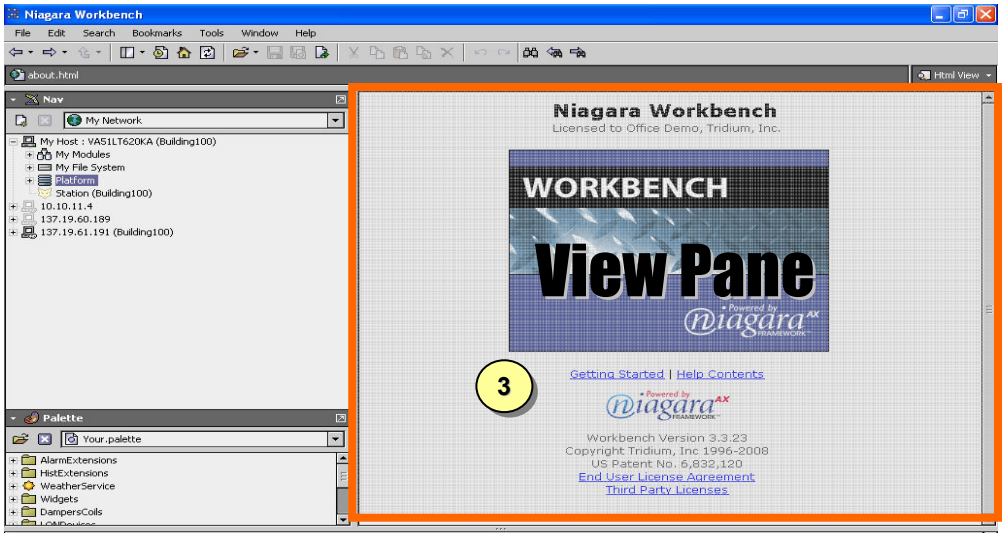
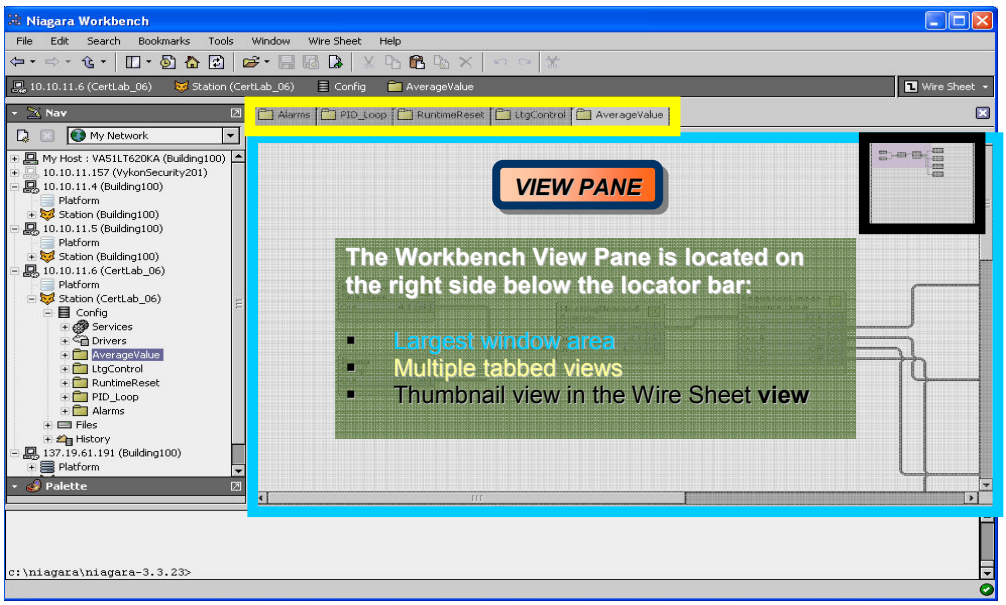
Writable points → Control palette – Points folder
Component space – right-click menu – New submenu
View pane – right-click menu – New submenu

Common Object Model and Components

FROM: Topic 4.2, Building Blocks

Component	Don't confuse components with modules. Modules make up the Niagara software itself. Components are used to build Niagara solutions using the Workbench. For the purposes of this course, any of the following objects are considered "components" -- ▪ Any of the **8 object types** of the Common Object Model: Boolean, Numeric, Enum, and String (both read-only and writable) ▪ An object that acts as a **control point**, such as a pump or fan that an operator can override ▪ **Extensions** – that act to extend the functionality of a point ▪ **Triggers** – that provide periodic action ▪ A **device** that can be monitored and/or controlled such as a Lon or Bacnet controller used to control a pump or valve ▪ **Logic or Math components** – that process an input value and provide an output value ▪ A **network** such as a Lon or Bacnet network ▪ A **container** or **folder** that contains other components
Slots	Niagara components are defined as a series of **slots** that define both the characteristics (Properties) of a component and how those components will behave (Actions) when a user invokes a command or an event occurs.
View	The Workbench **"View Pane"** can display any component you select. Different views of components can provide different editing options. There are many ways to visualize your system and its components. A view is just a "visualization" of a component. These are but a few of the most common examples: **Wire Sheet View** **Property Sheet View** **Category Sheet View** **Slot Sheet View** **Link Sheet View** **Manager Views** **Px Viewer/Editor** **Other Views** Each component has a default view that appears whenever you activate a component (double-clicking, for example) without specifying a particular view.

FROM: Topic 7.3, View Pane

Displays any component	The **View Pane** can display any component view that you select. Different views of components can provide different editing options. Most component editing is done on the View Pane, although many editing functions can be done in the Nav Tree as well. The functions are **context-sensitive** and may be used in several – but not all – views and with several – but not all – types of components. 
Below the Locator bar	The Workbench View Pane is located on the right side below the Locator Bar. It is the largest window area in the Workbench, can have multiple tabs, and can also have a thumbnail view in the Wire Sheet view. **It is in the View Pane that components are most often viewed and acted upon.** 

Common Object Model and Components

Interacting with components

In the Workbench, you can interact with components in the following ways:

COMPONENT ACTION	NAV TREE	VIEW PANE				
		WIRE SHEET	PROPERTY SHEET	SLOT SHEET	PX EDITOR	VIEW SELECTOR
Add new components	x	x	x	**		
Delete existing components	x	x	x	**		
Rename components	x	x	x	**	x	
Duplicate components	x	x	x			
Copy/paste components	x	x	x			
Drag & drop components *	x	x	x			
Reorder components	x	x	x	x	x	
Arrange components		x				
Select component view	x	x	x			x
Create composite view	x	x	x			

* Both left-click and right-click
** The most common tasks associated with the Slot Sheet view are adding, deleting and renaming slots on a component.

Drag & drop components

In each of the following scenarios – for the Property Sheet, Wire Sheet, Nav Tree and Palette Side Bar – the following can be dragged and dropped FROM one TO the other. Note that you cannot drag & drop FROM the Wire Sheet.

SCENARIOS		VIEW PANE		SIDE BAR PANE	
		Property Sheet	Wire Sheet	Nav Tree	Palette Side Bar
1	*From Property Sheet to Nav Tree or Palette Side Bar*				
	FROM	X	NO		
	TO			X	X (Personal Palette only)
2	*From Nav Tree or Palette Side Bar to Property Sheet or Wire Sheet* *From Nav Tree to Palette Side Bar* *From Palette Side Bar to Nav Tree*				
	FROM		NO	X	X
	TO	X	X	X	X

Viewing components

There are many ways to visualize your system and its components.
A view is a "visualization" of a component.

Selecting a component view

Typically, there are 4 ways to select a component view:

- Select a container, folder or component from the **Path Bar**.
- Right-click a container, file or component and choose a view from the **Views submenu**.*
- With a container, file or component selected (and displayed in the Path Bar), choose a view from the **View Selector** menu.
- Double-click on (select) a container, file or component in the **Nav Side Bar** to see its default view.

** Not all views have the same right-click functionality.*

Common Object Model and Components

Different views	The Workbench has 5 standard component views that display in the View pane: • ***Wire Sheet view*** • ***Property Sheet view*** • ***Category Sheet view*** • ***Slot Sheet view*** • ***Link Sheet view*** There are a number of other views that are context-sensitive – they depend on which container, folder, component or file is selected. These include a variety of Manager views and a Presentation Graphics (or PX) view that includes a Workbench-viewer and an editor.