

Navigation Strategies in Vision

Navigation Strategy

Setting up a navigation strategy allows you to navigate between different windows in the runtime Client. Ignition provides several different types of runtime navigational strategies you can choose from when designing your project. Before selecting the proper navigation strategy for your project, there are several things to consider. These considerations will help you determine the best navigation strategy to use for your project and your users. Once you address these considerations, then you can choose the best navigation strategy from the types below.

- Does your project have a lot of windows?
- How complex is your project structure?
- Is your project structure organized?
- What types of things are you doing?
- Do you want to use navigation windows or fill the screen?

Types of Navigation Strategies

To help you select the right strategy that fits your project structure, here is a brief description of each navigation strategy that Ignition provides. Keep in mind your project structure, size, organization, and types of things you are doing while you are reviewing these strategies so you can select the best runtime strategy for your project.

- **Tab Strip** is a simple strategy used for small structures regardless of how organized your project is. It lends itself perfectly to only having a few windows and showing all of them on a navigation window. Having too many tabs does not work well with the Tab Strip because of size limitations. You want your users to see all the navigation tabs immediately on the first screen. The Tab Strip works by clicking to swap one main window for another.
- **2 Tier** is similar to the Tab Strip, but is good for small and regular size project structures where windows are grouped. It contains a second level of tabs allowing you to navigate around various areas of your project. This strategy has a docked window that contains tabs that are always open to do navigation, and the main window which fills the rest of the space.
- **Tree View** is excellent for large project structures. You can view the entire project structure at a glance allowing you to navigate to any structure within the multi-tier Tree View component.
- **Back and Forward Buttons** are perfect if you have a small process with ordered steps. It is one big main window that has Back and Forward buttons to step through each process step or operation one right after the other.
- **Drill Down** is ideal if you have different geographical locations, whether it's in a local facility or facilities sprinkled around the world. The project opens with an overview that has areas on the that correspond to specific locations/areas in your facility. With the Drill Down strategy, you can select a specific area representing the facility, and the client swaps windows to display details pertaining to that specific area.
- **Menubar** is ideal for maximizing the usable screen space, while still having the ability to navigate to any window at any time by selecting from a list of windows.
- **Retargeting** enables navigation between multi-project operations: a simple script can push the user between many different projects, even on different gateways.

On this page

...

- [Navigation Strategy](#)
 - [Types of Navigation Strategies](#)
 - [Tab Strip Navigation](#)
 - [2-Tier Navigation](#)
 - [Tree View Navigation](#)
 - [Back and Forward Button Navigation](#)
 - [Drill Down Navigation](#)
 - [Menubar Navigation](#)
- [Navigation Operations - Swapping vs. Opening](#)
 - [Opening](#)
 - [Swapping](#)
- [Common Navigation Mistakes](#)
 - [Multiple 'Main' Windows](#)
 - [Swapping a Main Window with a Docked Window](#)

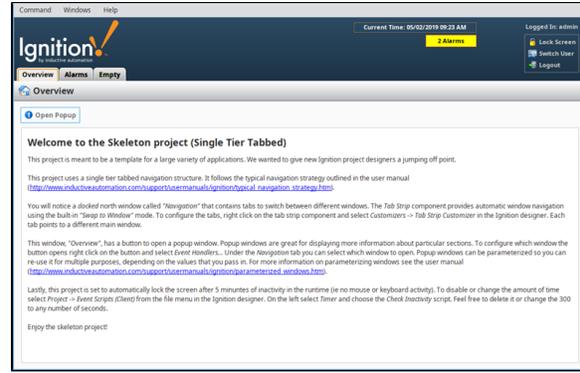
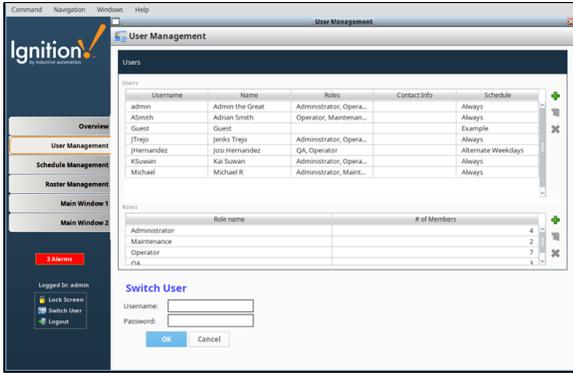


Navigation Strategies

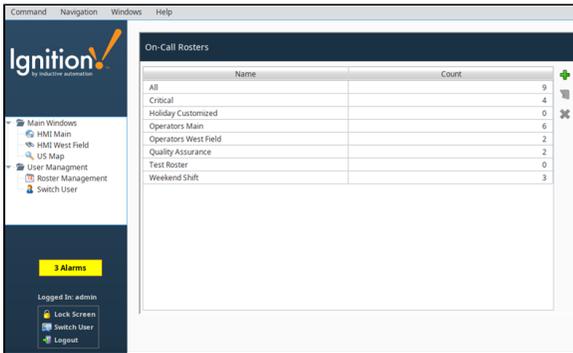
[Watch the Video](#)

Tab Strip Navigation

2-Tier Navigation



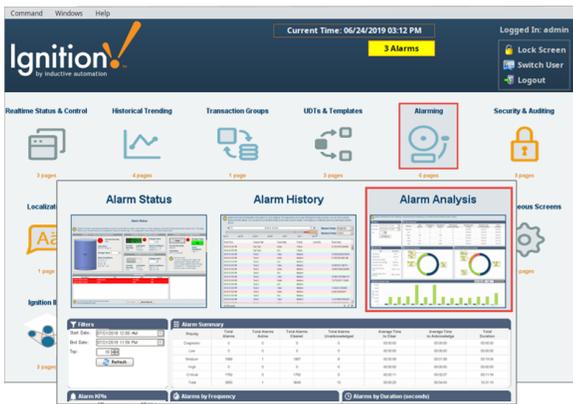
Tree View Navigation



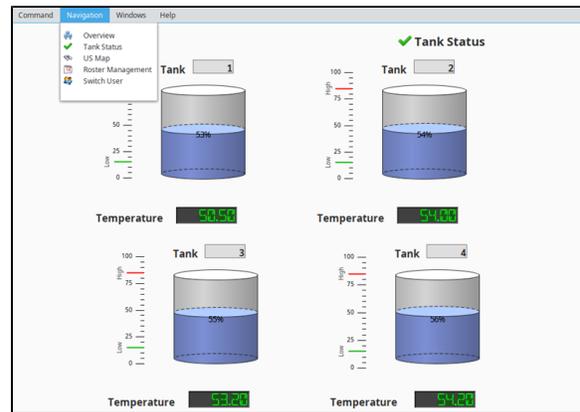
Back and Forward Button Navigation



Drill Down Navigation



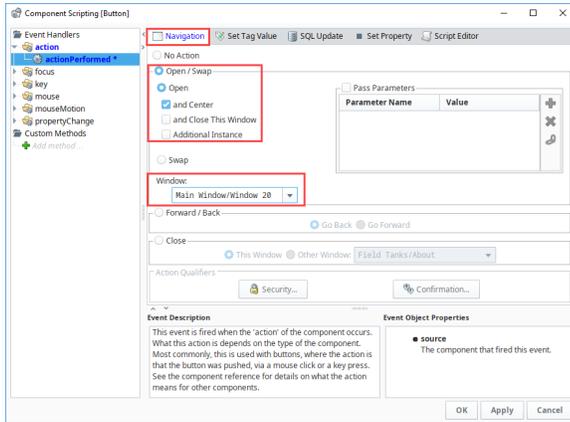
Menubar Navigation



Navigation Operations - Swapping vs. Opening

Any time you open a window, you have to use one of the two navigation operations: **Swapping** or **Opening**. These operations can be performed on any type of window, but are usually reserved for specific cases. Navigating between different windows typically involves some sort of scripting, but the **Navigation Script Builder** makes this a simple task. Otherwise, you can use the specific scripting functions to completely customize navigation.





Watch the Video



Navigation
Functions

Watch the Video

Opening

Opening and by extension closing are the basic window navigation options. Opening a window opens the window at the same size it was in the Designer, unless the **Start Maximized** property is **true** or the **Dock Position** is *not Floating*. This is typically reserved for opening popup windows. They have a scripting function that can open and a function that can close.

- `system.nav.openWindow`
- `system.nav.closeWindow`

Swapping

When Ignition swaps a window, it closes the current main window and then opens another window in its place. This is typically reserved for moving between main windows, as it performs the close action automatically. There are two different scripting functions that can be used to swap windows, depending on what needs to be done.

- `system.nav.swapWindow`
- `system.nav.swapTo`

Common Navigation Mistakes

Multiple 'Main' Windows

The most common mistake that will cause windows to stay open unintentionally is to implement a swapping navigation system using the **system.nav.openWindow()** function on main windows instead of **swapTo**. When you do this, the next time the **swapTo** function is called, it may swap from a window that is hidden behind the current 'Main' window and look like nothing happened. It is easy to check the client's **Windows** menu to see what windows are currently open. If there are more windows listed there than you can currently see, there is a problem in your navigation logic that is failing to close windows properly.

Swapping a Main Window with a Docked Window

Another common mistake that will cause windows to stay open unintentionally is to implement a swapping navigation system using the **system.nav.swapTo()** function on windows that are docked. This will cause your docked windows to be 'swapped in' as a maximized window instead of its usual size. When you do this, the client will not see it as a main window and next time the **swapTo** function is called, it may not have space on screen to open the main window. Logging out and back in to the client (or restarting it) is the only solution to this. Identify the offending button or script that is swapping the docked window and change it accordingly.

In This Section ...

