



QPoint

Robotic Solutions

KICKOFF

PROJECT INITIATION

- Each installation has a different set of Robo-Apps, and customization options
- Customizable options
 - This file: SmartViewLayout-Glass.csv contains the rules for the example shown in the video sent.
- We also have a set of system Robo-Apps that are displayed in a menu.
 - The file ModuleLayout-Glass.csv defines those.
- I don't want to change the ModuleLayout file, but I am very open to a change in the structure and definition of the SmartViewLayout file.
- I am sure it can be improved to better match your menu structure, and simplify functionality.
- These are files that we create, and edit when deploying a new installation
- I expect that my supervisory application will be responsible to manage the subset of items that are to be in the active menu, and not make you get into the weeds of all the rules of our supervisory application, but I will give you a little background in the interest of understanding

Application Background

- Our software platform is designed to manage robotic applications with a higher level functionality.
- Instead of hard coding each step and rule, we create a set of “actions” that blend robot motion, a hand or tool for the robot, and rules for the zone.
- By defining these actions as independent items, all of the software that would normally be needed to manage motion restrictions, handshaking, and mixed flow rules and sequencing are automatically generated.
- By nature of this design we can move the robot in a map structure much like you would find a route to drive your car (destination based, and all the turns are automatically calculated)
- The “Actions” are a combination of a robot, a TOOL Robo-App and a PROCESS Robo-App.
- Each of these apps have various low level devices and functions.
- The menu structure provides access to these manual/convenience controls, as well as triggering higher level actions.

Menu Structure Content

- The Operator level AND Machine Mode filters what menu items are available (either visually, or as an enabled action)
 - Operator Level - OMSCD
 - Operator
 - Maintenance
 - Style Edit
 - Config
 - Display Only (no actions, not part of menu structure, but may be shown if the menu heading is active)
 - Machine Mode - CPRS
 - Config
 - Prepare
 - Running
 - Service(Manual)
- I can perform the filtering of the available items to put in the tree, or specify the mode and operator level that is allowed. This is part of the information in the SmartViewLayoutFile – columns A and B
- Current Menu Organization
 - Custom Heading
 - Collection of items Under that heading
 - Collection of items Under that heading
 - Custom Heading 2
 - Collection of items Under that heading
 - Collection of items Under that heading
 - Robo-Apps
 - List of the visible apps
 - List of the visible apps
 - Background Robo-Apps
 - List of the background apps
 - List of the background apps
- Would like to have sub menus under the headings so you can have multiple levels, right now there is just 1 level.
- Each Robo-App has a set of
 - Actions (higher level routines)
 - Device controls (lower lever manual control of machine elements)
 - Monitoring (info about the App)
- When a Robo-App is selected, we currently show the detail of that app at right (large box of the LabView code front end)
 - Not in scope of this project for you, but we want a much better display of the apps for generic view, this will be part of phase 2

Tree Needs

- Would like to have sub menus under the headings so you can have multiple levels, right now there is just 1 level allowed for the configuration file
- Would like to somehow mark something as a favorite and thus allow for a favorite's menu heading that would list the actions that were selected. Even better if there were a set of favorites, where each operator could add to a list (sort of like saving favorite web pages, you can throw them in a group folder)
- Knowledge of what is actively displayed so we can provide dynamic updates to displays as needed, rather than continuously updating the entire tree (but not needed for first step)

Menu Definition / Dynamic status

Elements for an item Appearance

- Heading
- Sub Text
 - *Will change dynamically based on supervisory system
- Button
 - Action Type (designates coloring/appearance)
 - Menu Expansion, Detail view of Item, Device control, Robotic movement
 - See columns K and L in config file
 - ROBOT MOTION
 - MOVE TO,<RoboApps>\RoboApp_ConveyorPickZone.vi::Motion Loop R1::PATHEND
 - DEVICE CONTROL
 - <RoboApps>\RoboApp_Modbus Vac Tool.vi,Grip
 - SYSTEM,Clear Alarms
 - Button text
 - *Will change dynamically based on supervisory system (usually due to enabled state)
 - Enabled
 - *Will change dynamically based on supervisory system
- Image

FIRST STEP GOALS

- Show one or more looks of the tree using these concepts, doesn't need to work or be right, just need to get a feel for the visual. Ideally you would provide an HTML file I can drop into my system to allow it to be visible, in shape and size, ignoring all of the rules for populating and maintaining the tree.

Use this as your guide

- SmartViewLayout-Glass.csv
- Also Apps to Include
 - ModuleLayout-Glass.csv
 - The ones marked with a 5 in column F would be in the sub category of BACKGROUND Apps
 - (behave a lot like a programming running that you only see through your windows service tab on your desktop, not an active program running on our desktop)
 - Icons for the apps listed in column E, they are not necessarily ever a proper pixel size, but would need to auto scale for good visual. You will find these (and a bunch more legacy stuff) in the file _Graphics.zip
- Please add a few extra things to show sub options, creativity welcome

ADDITIONAL FEATURE to think about

- One operator mode that we use is a CHECKLIST process.
- This involves walking the operator through a series of sequential steps in order to follow the proper rules to get the machine going.
- This has an additional criteria in that a mechanism of enabling the option is that the PREREQUISITE STEPS are complete
 - We could identify these with a name prefix on the title, or something else.
 - New columns in the configuration file (or change to the file altogether)
- It would appear different, perhaps simply with an image of a check of unchecked step.

