DSCE at LSO

Chet Preston

Contents

[Config Files 4](#_Toc160467284)

[Channel Lists 4](#_Toc160467285)

[Sensor Scaling 4](#_Toc160467286)

[INI’s 4](#_Toc160467287)

[Data Reduction 4](#_Toc160467288)

[OPC Variables 4](#_Toc160467289)

[Organizational Only 4](#_Toc160467290)

[VEEx Wirelist 4](#_Toc160467291)

[VEEx Video Config 4](#_Toc160467292)

[VEEx Pwr Sply 4](#_Toc160467293)

[Opto Wirelist 4](#_Toc160467294)

[VEEx PLC Channel Assignment 4](#_Toc160467295)

[VEEx Bustec Analog 4](#_Toc160467296)

[Cameras 5](#_Toc160467297)

[New Bolin Cameras 5](#_Toc160467298)

[Add Cameras to Exacqvision 5](#_Toc160467299)

[New Cameras through Kiloview 5](#_Toc160467300)

[New Exacqvision Server 6](#_Toc160467301)

[New Bolin Controllers 6](#_Toc160467302)

[New Pelco-D Cameras 6](#_Toc160467303)

[Adding New Cameras to Bolin Controllers 7](#_Toc160467304)

[VEEx VideoController 7](#_Toc160467305)

[Troubleshooting 7](#_Toc160467306)

[Example 1: No video or Exacqvision 7](#_Toc160467307)

[Serial Communications 8](#_Toc160467308)

[Console Pages 9](#_Toc160467309)

[Adding RSV’s 9](#_Toc160467310)

[Adding Sensors 9](#_Toc160467311)

[Adding PCV’s 9](#_Toc160467312)

[Adding Set points controls 9](#_Toc160467313)

[Updating with BuildBot 9](#_Toc160467314)

[Hacking RSV’s with OPTO22 and OPC 9](#_Toc160467315)

[Messing with PLCs 10](#_Toc160467316)

[Allen Bradley – Studio 5000 10](#_Toc160467317)

[Opening PLC scripts 10](#_Toc160467318)

[DSCE Variables 12](#_Toc160467319)

[Modifying the OPC 13](#_Toc160467320)

[Connect to the OPC 13](#_Toc160467321)

[KepServerEX 6 14](#_Toc160467322)

[Updating OPC variables 14](#_Toc160467323)

[Data 15](#_Toc160467324)

[Data Logging 15](#_Toc160467325)

[Data Archiving 15](#_Toc160467326)

[Data Server 15](#_Toc160467327)

[VEEx STM Server (Admin) 15](#_Toc160467328)

[Data Reduction 16](#_Toc160467329)

[Time Syncing 17](#_Toc160467330)

[Software 18](#_Toc160467331)

[S: 18](#_Toc160467332)

[Custom Software 18](#_Toc160467333)

[Releasing Software 18](#_Toc160467334)

[Running Software 19](#_Toc160467335)

[Automation 21](#_Toc160467336)

[TRA 21](#_Toc160467337)

[NI Test Stand 21](#_Toc160467338)

[Imperio 21](#_Toc160467339)

[Restarting Imperio 21](#_Toc160467340)

[Adding Write Consoles 22](#_Toc160467341)

[Power Automate 22](#_Toc160467342)

[VEEx Terminal Room 23](#_Toc160467343)

[VEEx Control room and GVB IT Room 23](#_Toc160467344)

[VEEx Terminal Room 23](#_Toc160467345)

[VEEx Channel List 23](#_Toc160467346)

# Config Files

## Channel Lists

The channel list can only be modified if you have both admin access to it on your account and the file is write accessible. The file will need to be locked before it can be edited.

* Get lock for channel list
  + If you can’t lock then you probably don’t have edit privileges on your account
* Make changes
* Commit changes
* Run Punchlist Export
  + Commit any folder that receive changes due to this export
* Run Max Import
  + Need to be on Mission Network
  + If there were DAS changes, a stop/start on DAQ in console must be performed
* Run Tag Bot
  + Generally, run as a “Fix” unless the update is major
  + OPC variables require only tagging (export/import is handled on VEEx-OPC)

## Sensor Scaling

## INI’s

## Data Reduction

## OPC Variables

## Organizational Only

The following sheets are for organizational purposes only. Some are kept up a little, others not at all. In any case, since these sheets have no integrative function, their unreliability is only a hindrance from an organizational standpoint. Given that, for the most part, they are unused, that setback is moot. Nevertheless, they do exist.

### VEEx Wirelist

### VEEx Video Config

Video Config is the newest of the organizational sheets. To that extent it is

### VEEx Pwr Sply

### Opto Wirelist

### VEEx PLC Channel Assignment

### VEEx Bustec Analog

# Cameras

## New Bolin Cameras

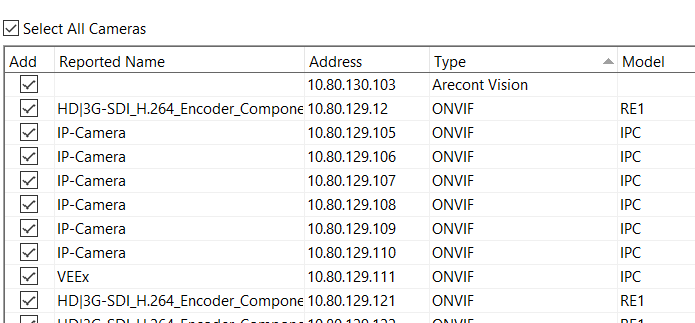
1. Connect via ethernet to the camera
   1. This can be either directly (by cord) or indirectly via the same hub
2. Manually set your IP address, mask and gateway to match the camera
   1. By default the camera is 192.168.0.13
   2. If refurbishing the camera, get the old IP address assignment and assign your ethernet card’s address to that network.
3. Connect to the cameras browser interface by typing in it’s IP address into chrome, firefox, etc.
4. Credentials are usually admin:admin or admin:(nothing)
   1. If refurbished, these credentials may not work. Check the manifests for credentials or ask an adult.
5. In the web UI, assign new network address:
   1. IP: 10.80.129.xx since .129 is the video network
   2. Subnet mask 255.255.255.0
   3. Gateway 10.80.129.254
   4. DNS1 10.1.1.30
   5. DNS2 10.1.1.27

## Add Cameras to Exacqvision

If cameras fail on exacqvision (fail to connect):

1. Note down its info (IP, login, etc)
2. Delete it
3. Add it again manually via its IP address

## New Cameras through Kiloview

1. Connect via ethernet to the camera
   1. This can be either directly (by cord) or indirectly via the same hub
2. Manually set your IP address, mask and gateway to match the camera
   1. These don’t have default IP addresses
   2. Easy way to get this is by:
      1. opening Exacqvision
      2. Checking the available cameras
      3. 
      4. Removing the ethernet cable from the camera
      5. Updating the list to see what was removed
      6. Plug the camera back in and update list again to verify the IP address is back
      7. Once that IP is found go to next step
3. Connect to the cameras browser interface by typing in it’s IP address into chrome, firefox, etc.
4. Credentials are usually admin:admin or admin:(nothing)
   1. If refurbished, these credentials may not work. Check the manifests for credentials or ask an adult.
5. In the web UI, assign new network address:
   1. IP: 10.80.129.xx since .129 is the video network
   2. Subnet mask 255.255.255.0
   3. Gateway 10.80.129.254
   4. DNS1 10.1.1.30
   5. DNS2 10.1.1.27
6. Refresh the camera list in Exacqvision and find your newly addressed camera and add it to your licensed cameras
7. Append to the name of the camera (the IP address) the tag “#transport=udp” and hit Apply
   1. Example: 10.80.129.143#transport=udp

## New Exacqvision Server

## New Bolin Controllers

Bolin Controllers have three ethernet ports on the back. From left to right they correspond to:

* PTZ control – to veex-data-bkup
* PTZ control – to veex-data-bkup
* IP assigning on video network

## New Pelco-D Cameras

1. Cameras will need to be added to the video network and the analog input needs to be assigned a channel on the Bolin Cameras
2. Find the IP address of the camera via Kiloview to add to exacqvision
3. For Bolin controllers, IP address is not needed, just assign the comms to Pelco-D
   1. Any baudrate will work, but 9600 is preferred
4. For PT control, following step 3 should suffice
5. For Z control, an ethernet to serial converter is necessary.
   1. Lantronix is our usual device
   2. Install DeviceInstaller from their website
   3. Use Telnet config to disable login password
   4. Assign an IP address on Video Network
   5. Can login via that IP address to change settings

## Adding New Cameras to Bolin Controllers

1. Press Setup
   1. PW is 0000
2. Use L/R and U/D knobs to adjust settings

## VEEx VideoController

Video Controller is a LabView software running on a windows computer (VEEx-DATA-BKUP) that provides information related to cameras. This includes:

1. The video display layout for the VEEx control room
2. The PTZ Control for specific cameras
3. Inserter control for time encoding

## Troubleshooting

Lots of shit can get wrong with cameras. This is further compounded by the fact that we use various types of cameras. Therefore, figuring out where an error has occurred can be difficult. Below is a subset of examples of video malfunctions. In most cases the handheld IP Camera Viewer in the I&C shop can come in handy by intercepting the video signal at multiple nodes from camera.

### Example 1: No video or Exacqvision

In the case that there is no video or exacqvision then we know that the underlying problem is likely upstream of the Kiloview. This could include the multiplexer, demultiplexer, or cameras themselves.

First, check the cameras input into the terminal room using the IPC viewer. If there is no signal, then go to the cameras themselves. They will first terminate at an enclosure nearby where they are positioned. Intercept the signal there using the IPC viewer as well. If no signal, try resetting the 120AC source to the cameras and wait for them to reboot.

If there is signal at either of the two above nodes, then check for connection clarity along the way between each node on the way to the GVB. There are many different wires and connections, so its always possible something gets either miswired or a wire breaks.

# Serial Communications

# Console Pages

## Adding RSV’s

## Adding Sensors

## Adding PCV’s

## Adding Set points controls

## Updating with BuildBot

1. Update and commit all changes in the console folder
2. Navigate to either geex-igs-6-bkup, tx-z6-1016, tx-z6-1146 or tx-zb-1348 and update the console folders
3. Run buildbot from the buildbot folder
4. Tag and Commit; use Fix

## Hacking RSV’s with OPTO22 and OPC

1. Create an RSV button using the “NC short as numeric.ctl” file
2. Name that RSV as RSV\_X####\_cmd\_manualon
   1. The RSV\_X#### doesn’t actually have to follow that format
3. Add a link on the KepServer on Veex-opc
   1. Add a new Tag under VTR – Epic VTR
   2. Name it the same as the name of the console button
   3. Have the address be [CONT|ip|tcp:10.80.130.200:22001]I32;VALUE;NAME
      1. Replace NAME with the name of the variable on the OPTO22
   4. Type **Long** w/ **Read/Write**
4. Create the variable on the OPTO22
   1. Open PAC Control and load the Terminal strategy
   2. Add a new variable under Numeric Variables tab
   3. Make it type **Integer 32**
   4. Press add public access
      1. Make readable
      2. Make writeable
      3. Represent as a Boolean

# Messing with PLCs

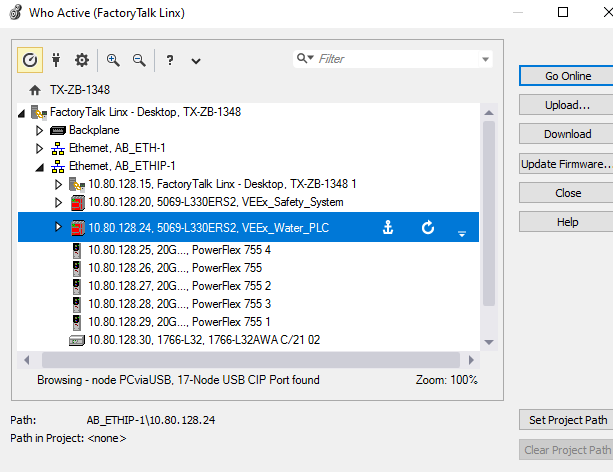
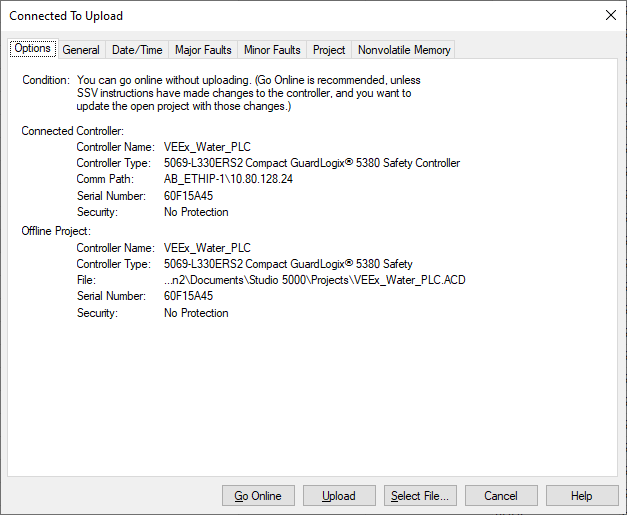
## Allen Bradley – Studio 5000

### Opening PLC scripts

* Scripts are generally saved in \*/Documents/Studio 5000/Projects
* Scripts can be received from a PLC
  + This is referred to as Uploading
* Or sent to a PLC
  + This is referred to as Downloading

#### Uploading Scripts

In order to upload a script that is currently on a PLC to your computer:

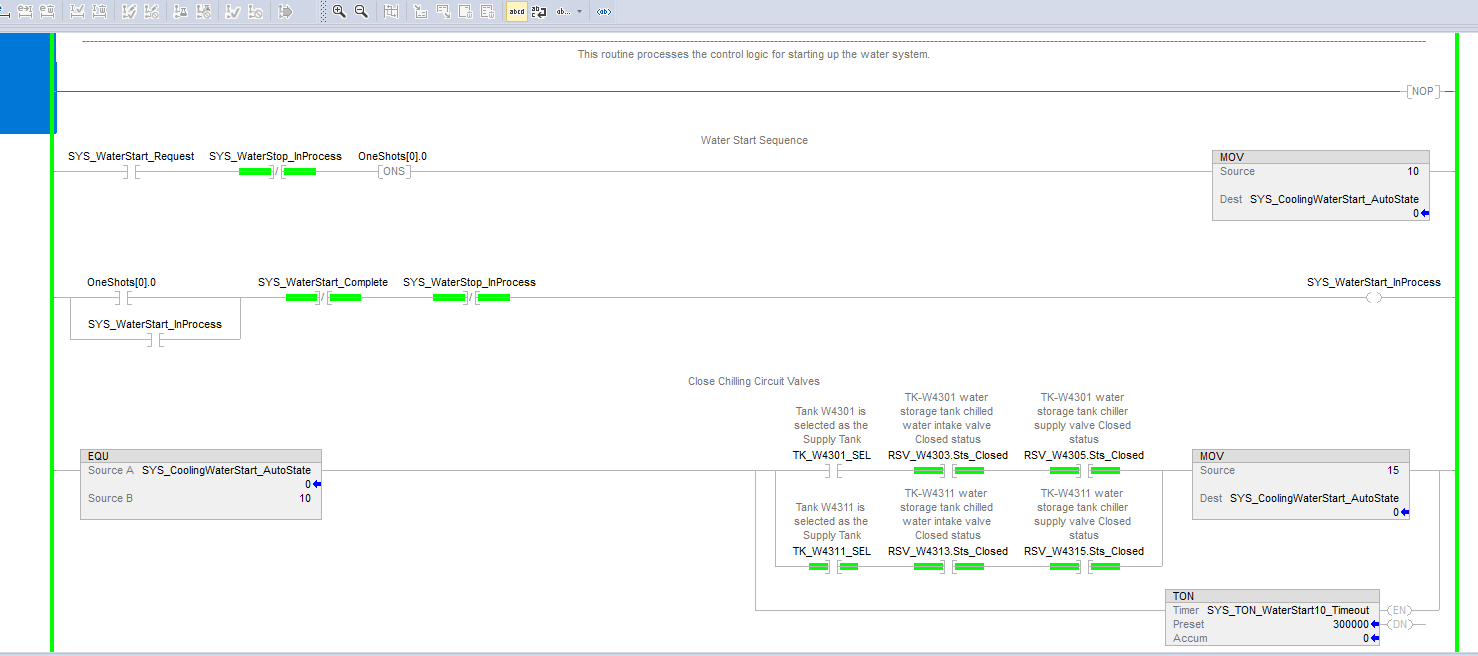
1. Open Studio 5000
2. Press Communications 🡺 Who Active
3. Navigate to the network the PLC’s are located
   1. For VEEx this is 10.80.128.xx
   2. For GEEx this is
   3. For XEEx this is
4. 
5. Choose the PLC you want to receive the logic script from and press Upload
6. 
7. If Upload is not available at the bottom, press Select File to choose a name to save the script as

#### Going Online

Going online with a script will actively connect you with the PLC and allow you to both monitor the states of the controller and actively change the energizations. This can be performed in two easy ways:

1. Follow the steps for uploading, but press Go Online instead of upload, Or…
2. After uploading the script on a particular PLC open the Communications menu and select Go Online

After going online, green bars will appear on the left and right of the script to indicate the actively online status.



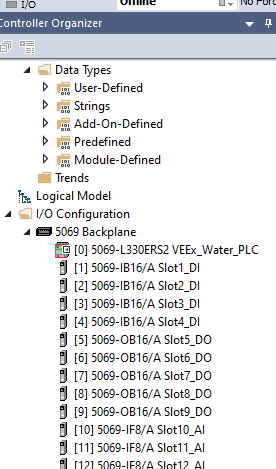
### DSCE Variables

Some variables (typically inputs) are configured through multiple avenues. These can be split into the controller tag, the I/O variable and the Asset Addon Instruction.

#### Controller Tags

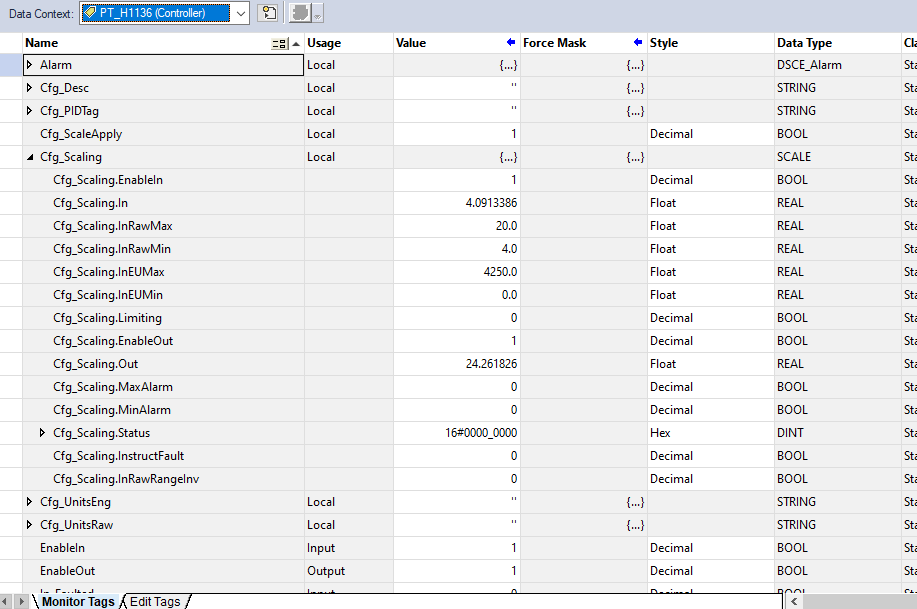
Controller tags make up the majority of variables stored on a typical AB PLC. These can be of numerous data types and can be used for I/O variables, local variables, or remote variables (used to communicate with other services such as console or another PLC). For example, when linking a variable between the OPC and an AB PLC, it is the controller tag that is generally used on the PLC side.

#### I/O Variables

I/O variables are found at the bottom of the Controller Organizer. These variables correspond to the inputs/outputs on the cards that are plugged into the PLC. These cards are then numbers and displayed in the PLC’s “I/O Backplane”. These variables are by default unnamed, but they do correspond to controller tags. The controller tags they represent can be found in the *Local:x:y etc,* section of the Controller Tags tab. A breakdown of these variable’s components can be found both in this backplane and in the Controller Tags tab. To assign a name to these data entries, choose the i/o point your want, for example “Local:26:I:Ch06.Data” and make an alias for it in the controller tags. This example would reference the value sampled on input card 26 channel 6.

#### Asset Add-On Instructions

These Add-On instructions are wrappers used by DSCE to represent and interact with I/O variables. Each one contains a standard set of inputs, outputs and calculations such as scaling.



## Modifying the OPC

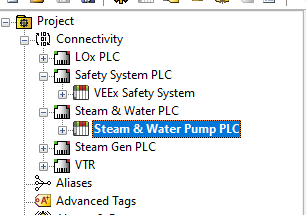
The OPC links the STM with the PLCs. Effectively, it will link names that will appear as variables in Console with the registers that hold the value or command on the PLC. These can be named the same thing but do not have to. This linking occurs through the use of different tools such as the UA Configuration Manager, the KepServer, and the OPC\_Client. Currently, VEEx runs OPC\_Client 5.0 in tandem with KepServer 6.0. This is the standard config as of February 2024. The OPC\_Client consumes an OPC.tom file in system\_configs/INI in order to associate itself correctly with a test stand.

### Connect to the OPC

The VEEx OPC is named veex-opc and is located on the Mission network (10.80.128.61). Use remote desktop to connect with a PC on the same network.

### KepServerEX 6

KepServer is used to organize the names and addresses of the different variables and commands between the STM and PLCs. Each PLC on the system is shown on KepServer.



New Tags can be added for each PLC.

* Name: This will be what Console sees as a variable name
* Description: This is for KEPserver only
* Address: This is the name of the register you want to read or write to found on the PLC the tag is located on
* Data Type: (use Double for Ints)
* Client Access: Only use Read/Write if this is a command

### Updating OPC variables

* After making changes, save and close the KEPserver
* Close the OPC\_Client
* Re-open the OPC Client
* Save the new OPC variables locally when prompted there has been a change
* Copy this OPC\_Variables file over to its directory in the system configs on an admin PC
  + This will generally be the one you use to remote into the OPC with
* Commit the OPC folder on the admin PC
* Run Tag Bot on the admin PC

# Data

## Data Logging

Logs all facility data at 1Hz and 5Hz. Reduces and pushes data at midnight everyday to BDMS and the shared drive (S:\).

Also creates TDMS files every night that is stored for 30 days. These files contain information that can be parsed in console to help provide timestamped feedback on historical data. Data Logger is run on VEEx-DATALOGGER (10.80.128.64). The PC it is run on is in the Terminal Room

## Data Archiving

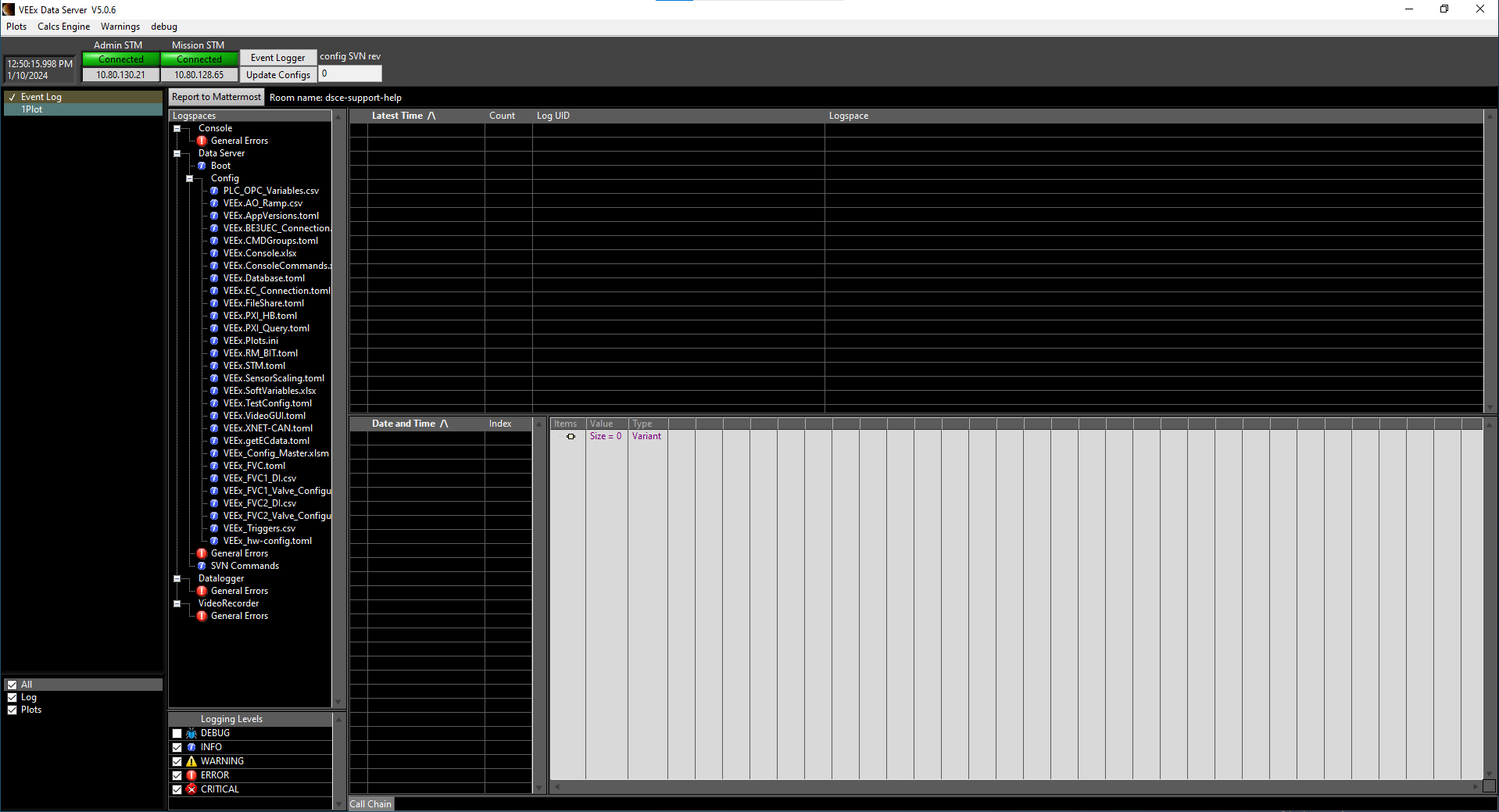
The data archiver (also known as Data Backup) logs the data for the data system (DAS, FVC, OPC etc) during tests and tares while also monitoring the health and errors of the system. It is located on VEEx-Data-BKUP (10.80.128.62). Tare logs are created whenever a tare operation is run. Similarly, A Test log is created whenever a Log start/stop command is run from console. The data archiver stores data for 7 days.

The Data Archiver program (VEEx\_Databackup) is driven by configuration files in our test stand’s files found at system\_configs\INI\_files\VEEx\_LogManager.toml

The VEEx-DATA-BKUP is located on the video rack in the GVB IT room

## Data Server

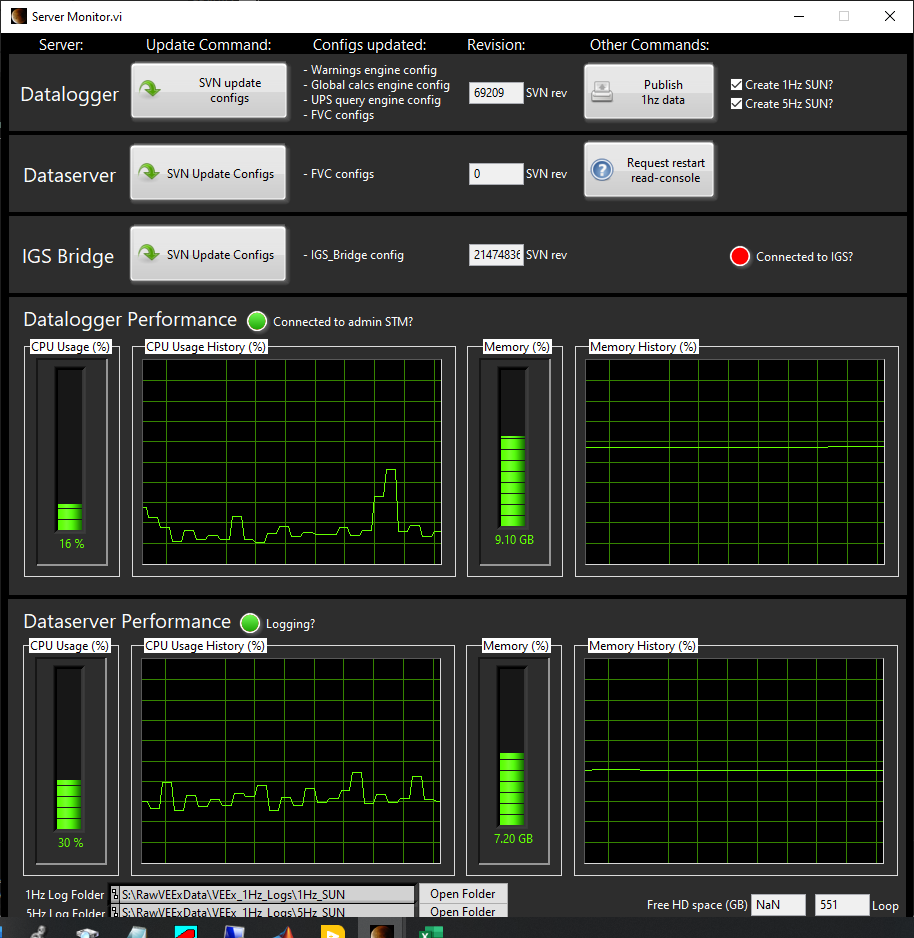
Located on VEEx-Data (10.80.128.65). Located in the Terminal Room

The data server is a 5Hz 90 minute long buffer for all the data on the network. It is this source of data that is used for data plotting on console. 

## VEEx STM Server (Admin)

This Labview app located on VEEx-Data (10.80.128.65) is responsible for read console data display.

1. The Mission STM server is repeated onto an Admin STM server
2. If the Admin Server STM drops, it will not affect Write Consoles
3. Displays all consoles and apps currently connected to the Admin STM Server



## Data Reduction

This is driven by a configuration file in a test stands user\_configs/sensor\_scaling folder.

## Time Syncing

# Software

The software we use generally comes from four different sources. The main two sources are S: and custom software. The S: hosts most of the commercial software and their licenses (or at least info to get them) that we use. However, much of the software that we currently run our test stand with are custom labview based programs that we create iterative releases for. The last two software repositories we use are software center and just grabbing stuff from the internet. Although IT does suggest we limit our non-approved software (stuff not in software center or S:), you’ll find that a lot of great software we use almost every day will come from there such as PyCharm and Discord.

## S:

The S: is linked to our network. Insofar as you are on site or connected via the VPN, you should have access to it. Although there are multiple directories containing software, the largest repository is found at: S:\ICE\Software Installations

## Custom Software

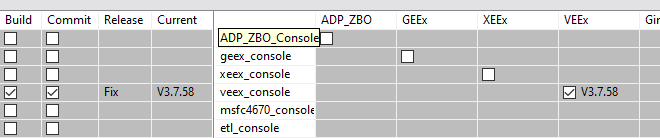
The VEEx test stand runs almost entirely on custom labview software for all of its control and data acquisition. This software is run through iterative releases that are linked to via batch files, i.e. loading these programs occurs through running targeted batch files. These batch files point to the svn location of the most recently tagged release version of the software. Therefore, from the user’s perspective, even when the software is updated, running the program remains the same and they don’t have to make any changes.

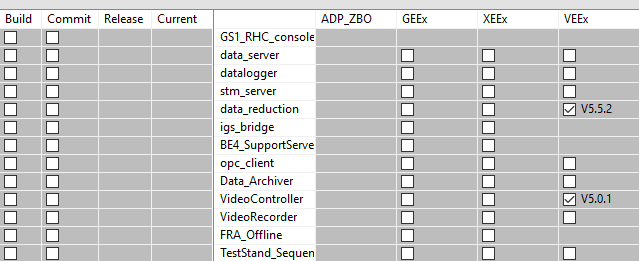
### Releasing Software

Software releases occur through running buildbot. Buildbot is a labview program and can be found at <https://subversion.blueorigin.com/svn/operations-software/blue_test_stands/trunk/ModularProjects/ConsoleCore/trunk/Source/projects/buildbot>

#### Buildbot

Buildbot itself is not test stand specific. Therefore, one must take care to only and correctly modify the releases for their particular stands. Buildbot releases can be broken into two main uses: 1) releasing new console updates, and 2) updating auxiliary software.

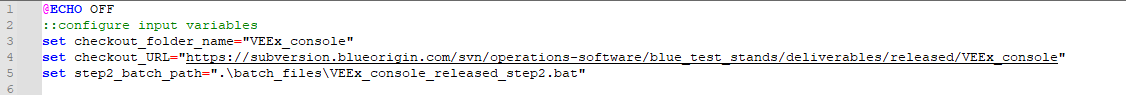
When updating console, buildbot will grab the associated labview files from the GUI directory for the chosen stand. For VEEx this will be the Load VEEx GUIs vi as well as all of the VI’s in the subsequent VEEx folder that Load VEEx GUIs calls. When building a new console version, it is important to check both the build and commit boxes, as well as choose a type of release (Major, Minor, Fix). Just to note, these terms correspond to the iteration of the version number on the new release such that major will increase the first number, minor the second, and fix the last. All in all, the release of a new console version will look like this: 

This is in contrast for simply updating to releases. Updating is most commonly performed when the software team creates and update to a non-test stand specific tool such as the data logger, STM server or OPC Client. In order to update the particular stands release of that application, only the test stand’s associated box needs to be checked before running buildbot, and not the build or commit boxes like shown below. 

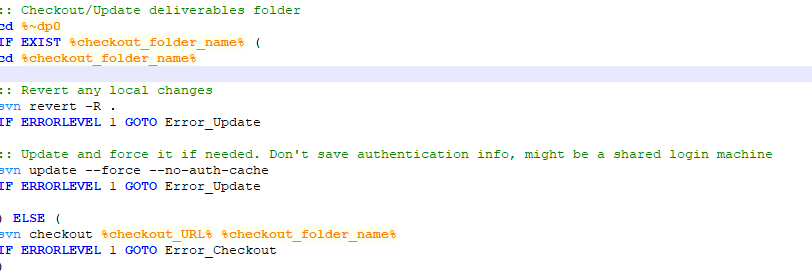
In this example, running buildbot would update the VEEx-specific releases of the data reduction and video controller softwares. After updating these software, they will need to be rebooted on the computers they are run on. Note that this sometimes requires updating the labview runtimes on those computers.

### Running Software

Custom Labview software is typically run through batch files.



These files do two things. First they direct the rest of the file toward the files associated with the particular program to be launched. The example above is for the VEEx console.



The second function is to perform svn updates and checkouts of the files associated with the chosen program. After performing these checkouts, a step2 batch file is loaded which launches the actual application.

# Automation

## TRA

## NI Test Stand

NI Test Stand is an automation package that enables case specific automation which dramatically increases functionality compared to something like TRA with is timing based alone. Since NI Test Stand was wrapped by Imperio, it has only had limited implementation at VEEx in its original form. The automation scripts we created can be accessed at https://subversion.blueorigin.com/svn/propulsion/GaPPS/WTLS\_Testing/Test\_Requests/branches/NI Automation .

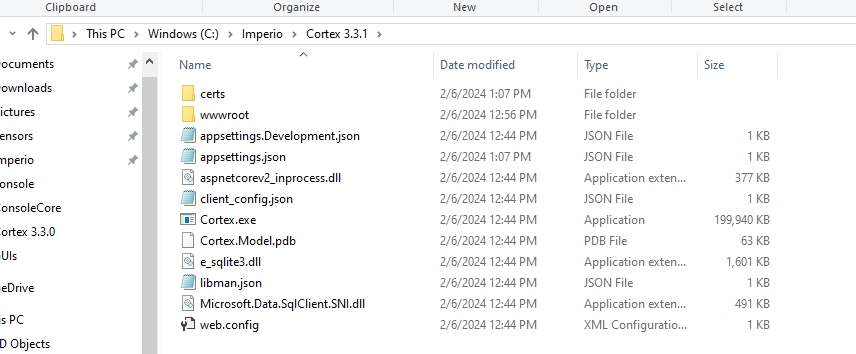
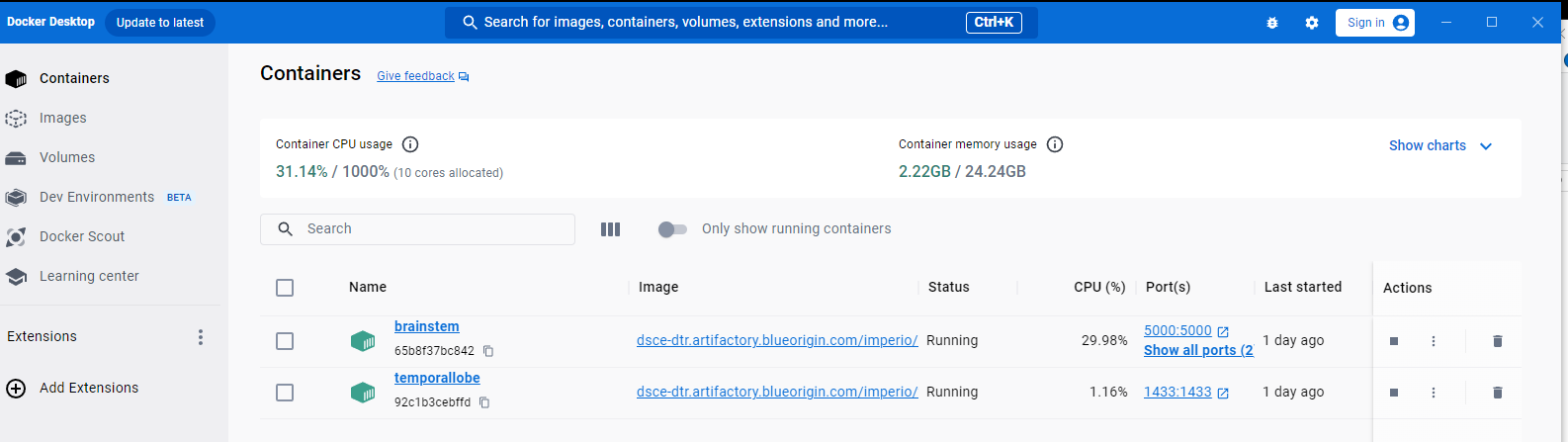
While these scripts exist and can be run on any computer running TSSR, they are mostly obsolete now with the ascension of Imperio. Essentially, NI TestStand and Imperio are the same thing, thus learning one can inform of the other, and to that extent I will not go into detail on NI Test Stand here. It is important to note, however, that Imperio at its core is a subset of NI TestStand; thus TestStand generally has a wider functionality than Imperio. This may not always be true depending on how (if) unique functions are added to Imperio.

## Imperio

Imperio is a C# backend, JavaScript front end tool that wraps NI TestStand on a single host computer in order to simplify and streamline automation at Blue Origin. Imperio is accessed via web browser at veex.blueorigin.com

### Restarting Imperio

In the case that Imperio becomes offline, it can be restarted on its host computer: TX-Z6-1016. This takes two steps:

* + 1. Load cortex.exe in the Imperio folder
    2. 
    3. Run Brainstem and Temporal Lobe from Docker
    4. 

### Adding Write Consoles

Having a write console on Gemini doesn’t automatically mean that same computer has a write console through Imperio. Adding write privileges for each case is unique. For Imperio, connect to TX-Z6-1016, navigate to the Imperio folder, run “”, and add the IP of the computer you want to add to the whitelist. Restart Imperio cortex afterwards in order to manifest the changes.





## Power Automate

# VEEx Terminal Room

The centralized communication terminals at VEEx are located in three primary locations. The first is the VEEx Terminal Room which is a conex at the test site that most of or effectors/affectors are route through. The second is the GVB IT Room which reroutes the majority of our video network as well as interfacing our control room computers to our networks. The third is the control room itself. As it stands during the writing of this, the video network stack is still in the veex control room, but the majority of it will soon be moved to the GVB IT room. How the components end up being split between the two rooms is still in the air.

## VEEx Control room and GVB IT Room

Much of these resources are located in either other parts of this document or other documents in our databases.

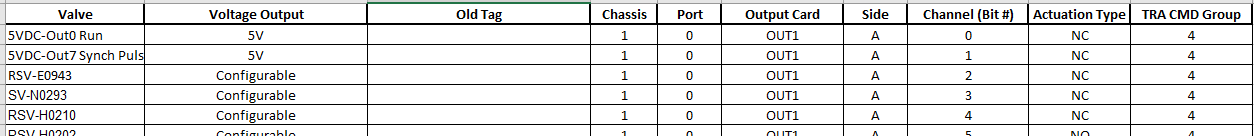
## VEEx Terminal Room

All (almost all?) of the effectors/affectors located in our facilities and test articles are routed through the VEEx Terminal. Most of these are routed from enclosures nearby the facility or test article the device is used for. Most of these devices inevitably terminate from/at our DAQ or valve controller systems before being routed to the STM (also located in the terminal room), but some are further rerouted to the GVB (mostly video stuff).

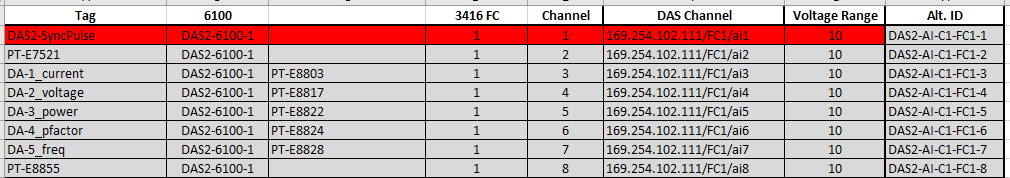
### VEEx Channel List

The channel list is a config file that serves two main purposes. The first is to link the STM server with the DAS and FVC systems so that commands from console are transmitted where desired. The second is to provide an organized way of finding the physical connections related to particular devices. The former is already [addressed here](#_Channel_Lists), so this section is left to address the latter.

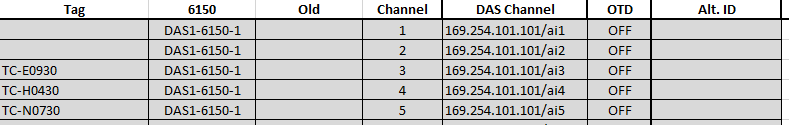
#### FVC



#### DAS 6100



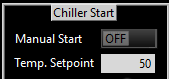
#### DAS 6150



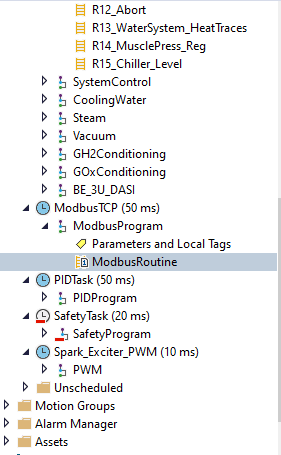
# Other Devices

## Advantage Chiller

VEEx has two chillers for their two water tanks. Right now, both tanks work in parallel to cool one tank at a time. Both chillers share the same PLC on signal. In order to actually start each chiller, the chiller boxes must be on prior to the PLC on signal. The PLC signal is sent from the Manual Start button on the Water console page (shown below). To turn on the chiller boxes, 3 of the four knobs on the box must be on. These are the Main power, Evap Pump and ??? Pump (#2). The Standby knob should remain off.



### Chiller Telemetry

The telemetry associated with the Advantage Chillers is transmitted via ethernet. The ethernet card on the Chiller panel is connected to a switch on the Water PLC. Within Studio 5000, this telemetry is located under the ModBus tab.