DSCE at LSO

Chet Preston

Contents

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
    a. 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
    a. By default the camera is 192.168.0.13
    b. 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)
    a. 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:
    a. IP: 10.80.129.xx since .129 is the video network
    b. Subnet mask 255.255.255.0
    c. Gateway 10.80.129.254
    d. DNS1 10.1.1.30
    e. DNS2 10.1.1.27

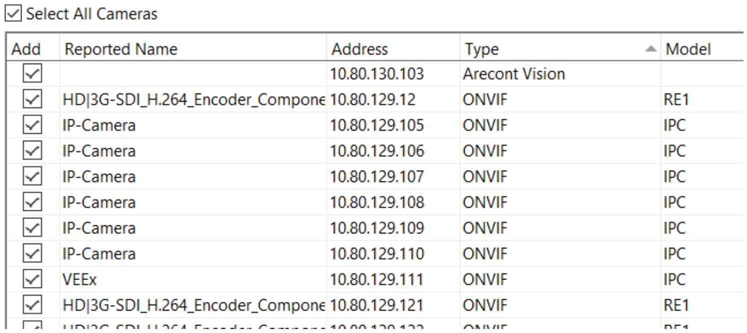
Add Cameras to Exacqvision

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
    a. 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
    a. These don't have default IP addresses
    b. Easy way to get this is by:
        i. opening Exacqvision
        ii. Checking the available cameras
        iii.



☑ Select All Cameras

Add	Reported Name	Address	Type	Model
☑		10.80.130.103	Arecont Vision	
☑	HD|3G-SDI_H.264_Encoder_Compone	10.80.129.12	ONVIF	RE1
☑	IP-Camera	10.80.129.105	ONVIF	IPC
☑	IP-Camera	10.80.129.106	ONVIF	IPC
☑	IP-Camera	10.80.129.107	ONVIF	IPC
☑	IP-Camera	10.80.129.108	ONVIF	IPC
☑	IP-Camera	10.80.129.109	ONVIF	IPC
☑	IP-Camera	10.80.129.110	ONVIF	IPC
☑	VEEx	10.80.129.111	ONVIF	IPC
☑	HD|3G-SDI_H.264_Encoder_Compone	10.80.129.121	ONVIF	RE1
☑	HD|3G-SDI_H.264_Encoder_Compone	10.80.129.122	ONVIF	RE1

        iv. Removing the ethernet cable from the camera
        v. Updating the list to see what was removed
        vi. Plug the camera back in and update list again to verify the IP address is back
        vii. 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)
    a. 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:
    a. IP: 10.80.129.xx since .129 is the video network
    b. Subnet mask 255.255.255.0
    c. Gateway 10.80.129.254
    d. DNS1 10.1.1.30
    e. 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
    a. 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
    a. 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.
    a. Lantronix is our usual device
    b. Install DeviceInstaller from their website
    c. Use Telnet config to disable login password
    d. Assign an IP address on Video Network
    e. Can login via that IP address to change settings

Adding New Cameras to Bolin Controllers

1. Press Setup
    a. 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
    a. The RSV_X#### doesn’t actually have to follow that format
3. Add a link on the KepServer on Veex-opc
    a. Add a new Tag under VTR – Epic VTR
    b. Name it the same as the name of the console button
    c. Have the address be [CONT|ip|tcp:10.80.130.200:22001]I32;VALUE;NAME
        i. Replace NAME with the name of the variable on the OPTO22
    d. Type **Long** w/ **Read/Write**
    e.
4. Create the variable on the OPTO22
    a. Open PAC Control and load the Terminal strategy
    b. Add a new variable under Numeric Variables tab
    c. Make it type **Integer 32**
    d. Press add public access
        i. Make readable
        ii. Make writeable
        iii. Represent as a Boolean

Messing with PLCs

Allen Bradley – Studio 5000

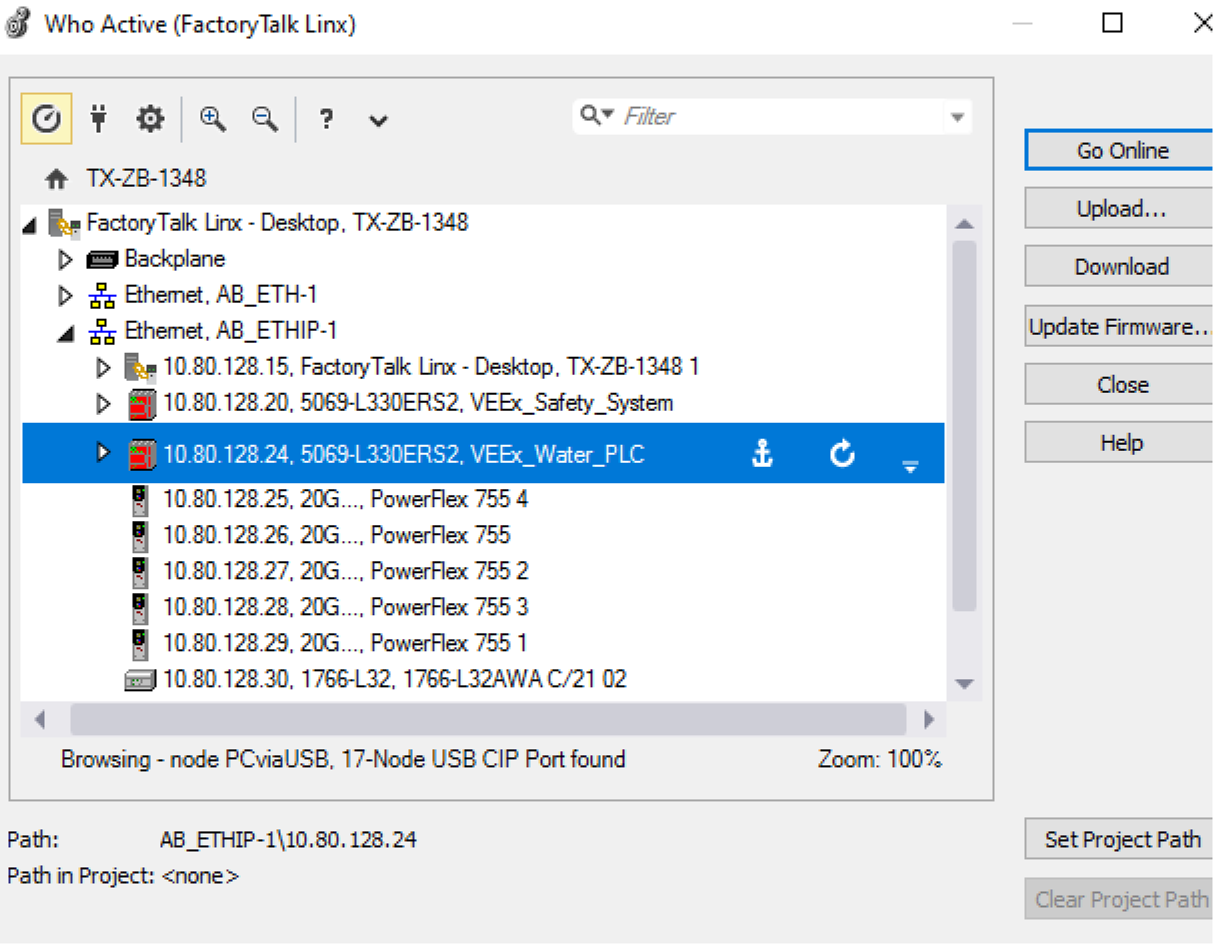
Opening PLC scripts

- 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
   a. For VEEx this is 10.80.128.xx
   b. For GEEx this is
   c. For XEEx this is



TX-ZB-1348

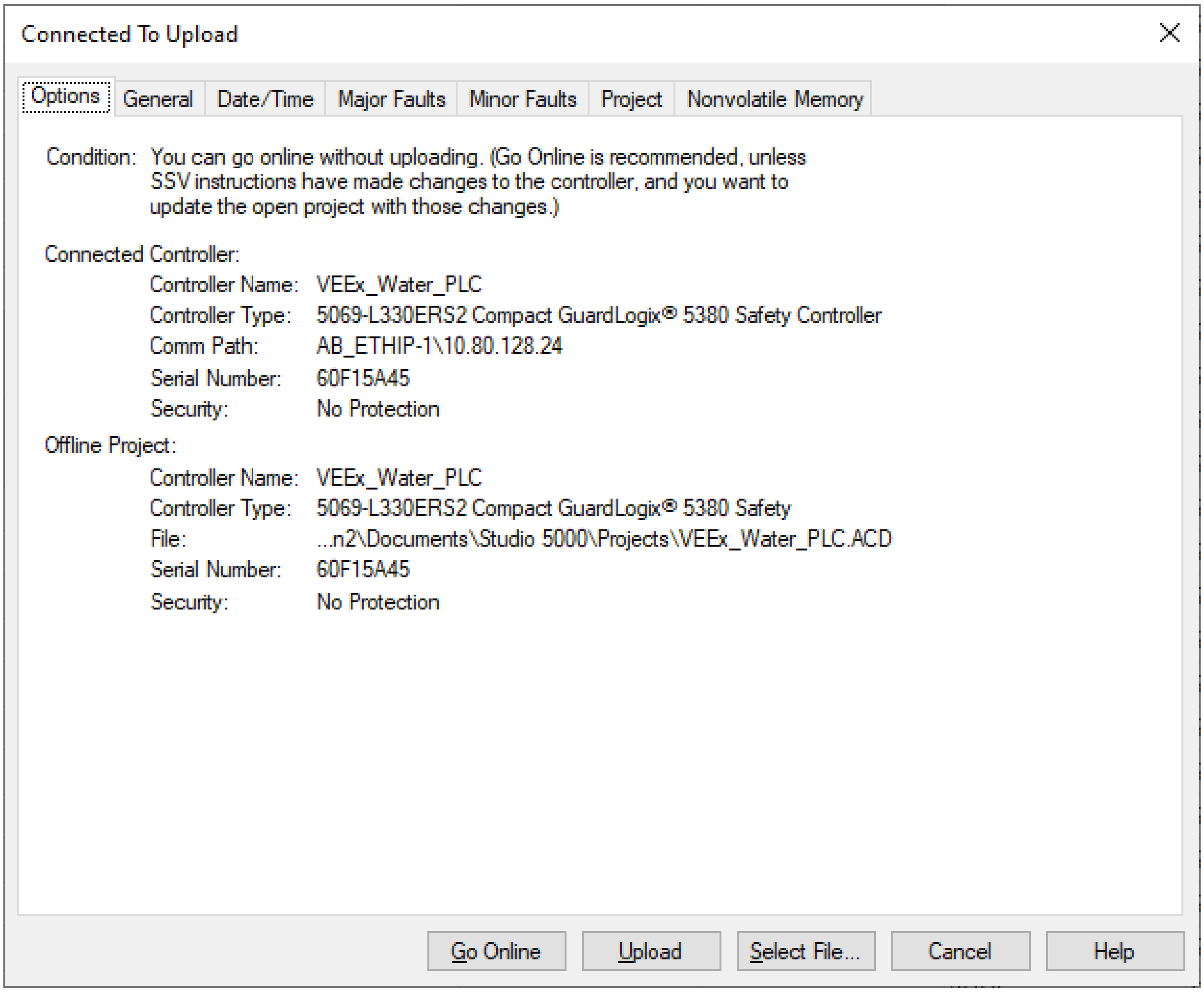
FactoryTalk Linx - Desktop, TX-ZB-1348
Backplane
Ethernet, AB_ETH-1
Ethernet, AB_ETHIP-1
10.80.128.15, FactoryTalk Linx - Desktop, TX-ZB-1348 1
10.80.128.20, 5069-L330ERS2, VEEx_Safety_System
10.80.128.24, 5069-L330ERS2, VEEx_Water_PLC
10.80.128.25, 20G..., PowerFlex 755 4
10.80.128.26, 20G..., PowerFlex 755
10.80.128.27, 20G..., PowerFlex 755 2
10.80.128.28, 20G..., PowerFlex 755 3
10.80.128.29, 20G..., PowerFlex 755 1
10.80.128.30, 1766-L32, 1766-L32AWA C/21 02

Browsing - node PCviaUSB, 17-Node USB CIP Port found   Zoom: 100%

Path: AB_ETHIP-1\10.80.128.24

Path in Project: <none>

4.
5. Choose the PLC you want to receive the logic script from and press Upload



Condition: You can go online without uploading. (Go Online is recommended, unless SSV instructions have made changes to the controller, and you want to update the open project with those changes.)

Connected Controller:

Controller Name: VEEx_Water_PLC
Controller Type: 5069-L330ERS2 Compact GuardLogix® 5380 Safety Controller
Comm Path: AB_ETHIP-1\10.80.128.24
Serial Number: 60F15A45
Security: No Protection

Offline Project:

Controller Name: VEEx_Water_PLC
Controller Type: 5069-L330ERS2 Compact GuardLogix® 5380 Safety
File: ...n2\Documents\Studio 5000\Projects\VEEx_Water_PLC.ACD
Serial Number: 60F15A45
Security: No Protection

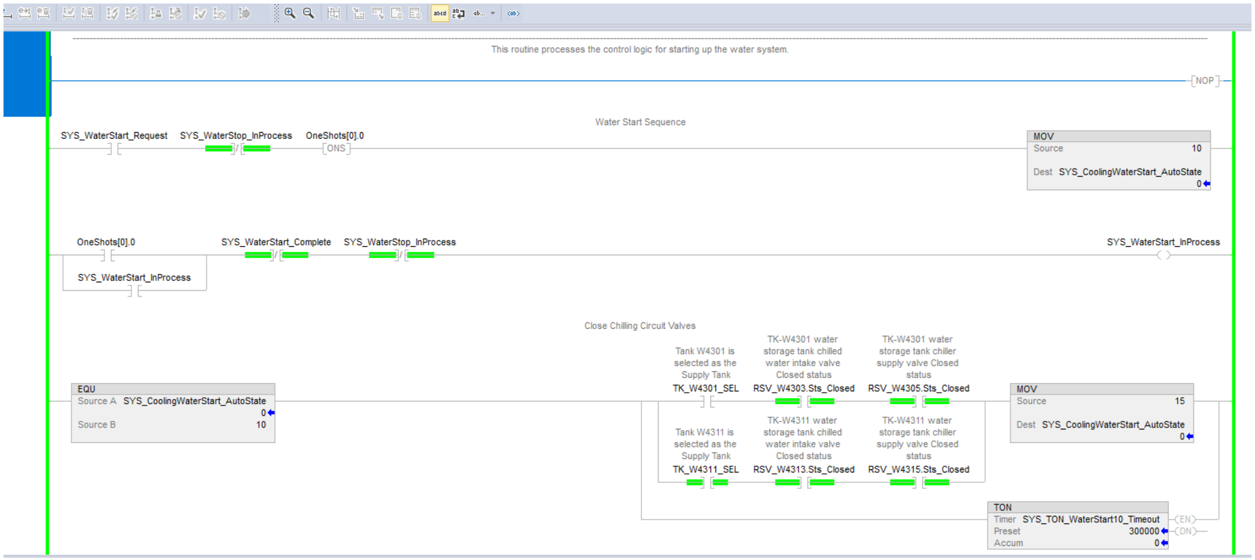
6.
7. If Upload is not available at the bottom, press Select File to choose a name to save the script as

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.



This routine processes the control logic for starting up the water system.

Water Start Sequence

Close Chilling Circuit Valves

NOP
SYS_WaterStart_Request
SYS_WaterStop_InProcess
OneShots[0].0
ONS
MOV
Source 10
Dest SYS_CoolingWaterStart_AutoState
0
OneShots[0].0
SYS_WaterStart_Complete
SYS_WaterStop_InProcess
SYS_WaterStart_InProcess
SYS_WaterStart_InProcess
EQU
Source A SYS_CoolingWaterStart_AutoState
0
Source B 10
Tank W4301 is selected as the Supply Tank
TK_W4301_SEL
TK-W4301 water storage tank chilled water intake valve Closed status
RSV_W4303.Sts_Closed
TK-W4301 water storage tank chiller supply valve Closed status
RSV_W4305.Sts_Closed
Tank W4311 is selected as the Supply Tank
TK_W4311_SEL
TK-W4311 water storage tank chilled water intake valve Closed status
RSV_W4313.Sts_Closed
TK-W4311 water storage tank chiller supply valve Closed status
RSV_W4315.Sts_Closed
MOV
Source 15
Dest SYS_CoolingWaterStart_AutoState
0
TON
Timer SYS_TON_WaterStart10_Timeout
Preset 300000
Accum 0
EN
DN

DSCE Variables

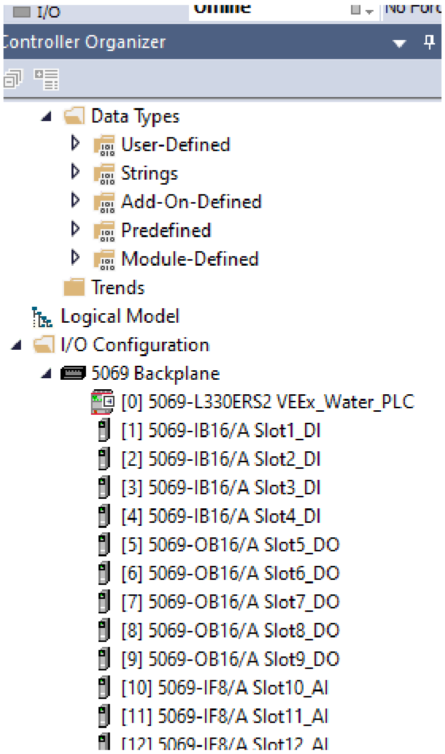
Some variables (typically inputs) are configured through multiple avenues. These can be split into the controller tag, the I/O variable and the Asset Addon Instruction.

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.

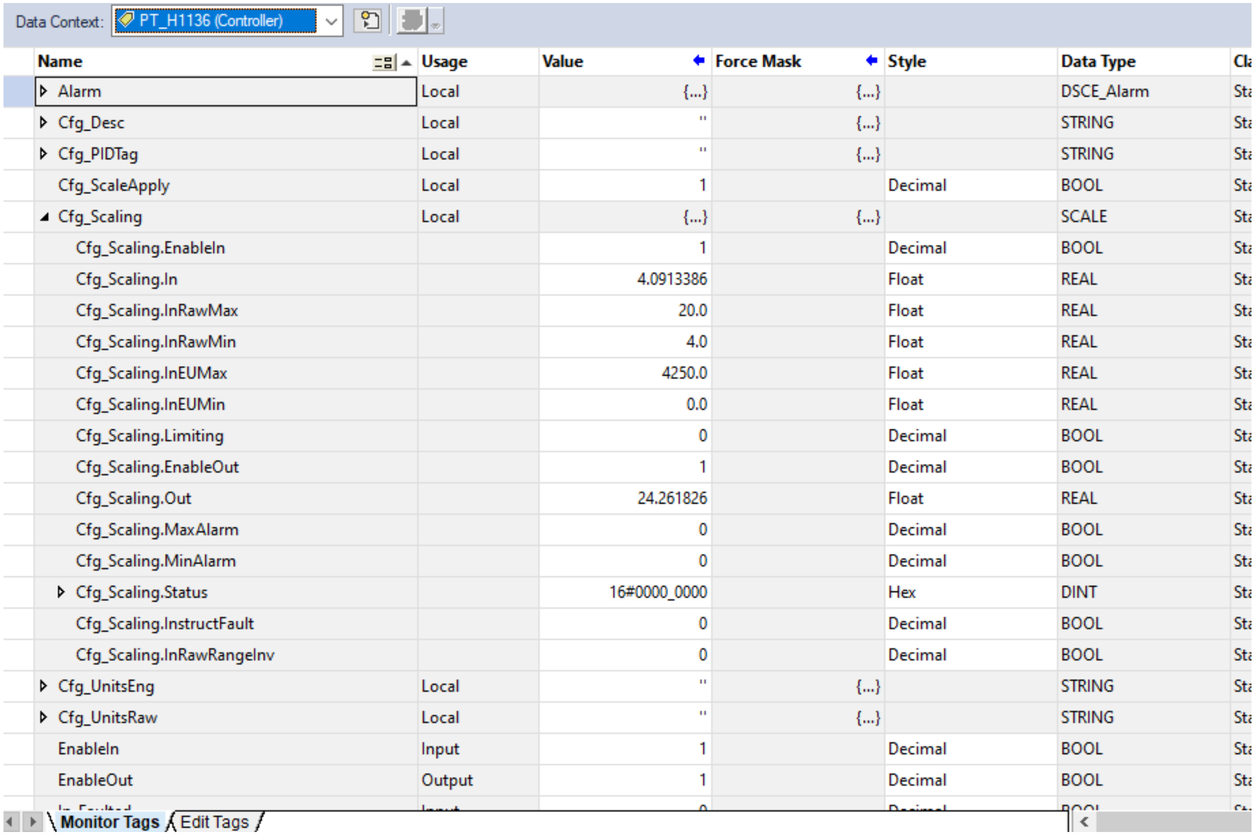
I/O Variables

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.



Name	Usage	Value	Force Mask	Style	Data Type	Cl
Alarm	Local	{...}	{...}		DSCE_Alarm	Sta
Cfg_Desc	Local	''	{...}		STRING	Sta
Cfg_PIDTag	Local	''	{...}		STRING	Sta
Cfg_ScaleApply	Local	1		Decimal	BOOL	Sta
Cfg_Scaling	Local	{...}	{...}		SCALE	Sta
Cfg_Scaling.EnableIn		1		Decimal	BOOL	Sta
Cfg_Scaling.In		4.0913386		Float	REAL	Sta
Cfg_Scaling.InRawMax		20.0		Float	REAL	Sta
Cfg_Scaling.InRawMin		4.0		Float	REAL	Sta
Cfg_Scaling.InEUMax		4250.0		Float	REAL	Sta
Cfg_Scaling.InEUMin		0.0		Float	REAL	Sta
Cfg_Scaling.Limiting		0		Decimal	BOOL	Sta
Cfg_Scaling.EnableOut		1		Decimal	BOOL	Sta
Cfg_Scaling.Out		24.261826		Float	REAL	Sta
Cfg_Scaling.MaxAlarm		0		Decimal	BOOL	Sta
Cfg_Scaling.MinAlarm		0		Decimal	BOOL	Sta
Cfg_Scaling.Status		16#0000_0000		Hex	DINT	Sta
Cfg_Scaling.InstructFault		0		Decimal	BOOL	Sta
Cfg_Scaling.InRawRangeInv		0		Decimal	BOOL	Sta
Cfg_UnitsEng	Local	''	{...}		STRING	Sta
Cfg_UnitsRaw	Local	''	{...}		STRING	Sta
EnableIn	Input	1		Decimal	BOOL	Sta
EnableOut	Output	1		Decimal	BOOL	Sta
In_Faulted	Input	0		Decimal	BOOL	Sta

Modifying the OPC

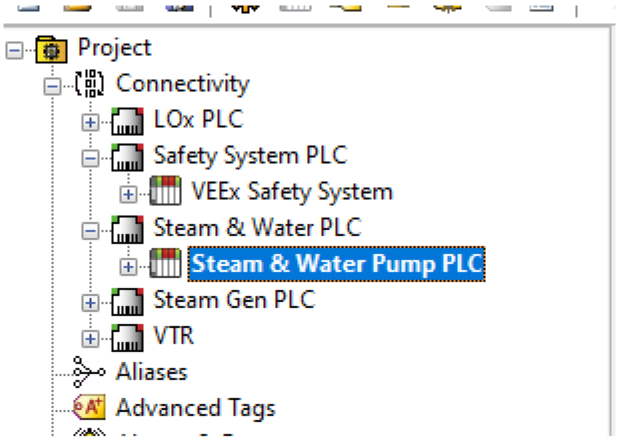
The OPC links the STM with the PLCs. Effectively, it will link names that will appear as variables in Console with the registers that hold the value or command on the PLC. These can be named the same thing but do not have to. This linking occurs through the use of different tools such as the UA Configuration 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.



Project
Connectivity
LOx PLC
Safety System PLC
VEEx Safety System
Steam & Water PLC
Steam & Water Pump PLC
Steam Gen PLC
VTR
Aliases
Advanced Tags

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.

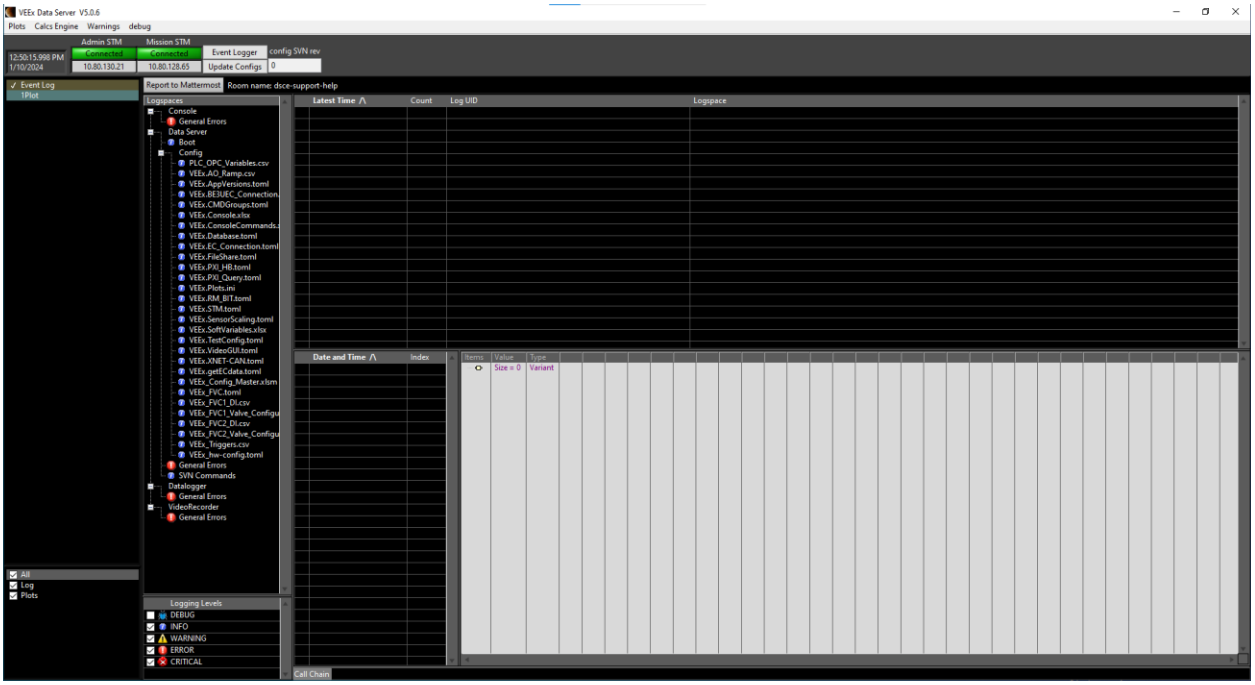
The Data Archiver program (VEEx_Databackup) is driven by configuration files in our test stand's files found at system_configs\INI_files\VEEx_LogManager.toml

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.



12:50:15.998 PM
1/10/2024

Admin STM: Connected, 10.80.130.21
Mission STM: Connected, 10.80.128.65
config SVN rev: 0
Room name: dsce-support-help

```
Logspaces
⊟ Console
   General Errors
⊟ Data Server
   Boot
  ⊟ Config
     PLC_OPC_Variables.csv
     VEEx.AO_Ramp.csv
     VEEx.AppVersions.toml
     VEEx.BE3UEC_Connection.
     VEEx.CMDGroups.toml
     VEEx.Console.xlsx
     VEEx.ConsoleCommands.
     VEEx.Database.toml
     VEEx.EC_Connection.toml
     VEEx.FileShare.toml
     VEEx.PXI_HB.toml
     VEEx.PXI_Query.toml
     VEEx.Plots.ini
     VEEx.RM_BIT.toml
     VEEx.STM.toml
     VEEx.SensorScaling.toml
     VEEx.SoftVariables.xlsx
     VEEx.TestConfig.toml
     VEEx.VideoGUI.toml
     VEEx.XNET-CAN.toml
     VEEx.getECdata.toml
     VEEx_Config_Master.xlsm
     VEEx_FVC.toml
     VEEx_FVC1_DI.csv
     VEEx_FVC1_Valve_Configu
     VEEx_FVC2_DI.csv
     VEEx_FVC2_Valve_Configu
     VEEx_Triggers.csv
     VEEx_hw-config.toml
   General Errors
   SVN Commands
⊟ Datalogger
   General Errors
⊟ VideoRecorder
   General Errors
```

☑ All
☑ Log
☑ Plots

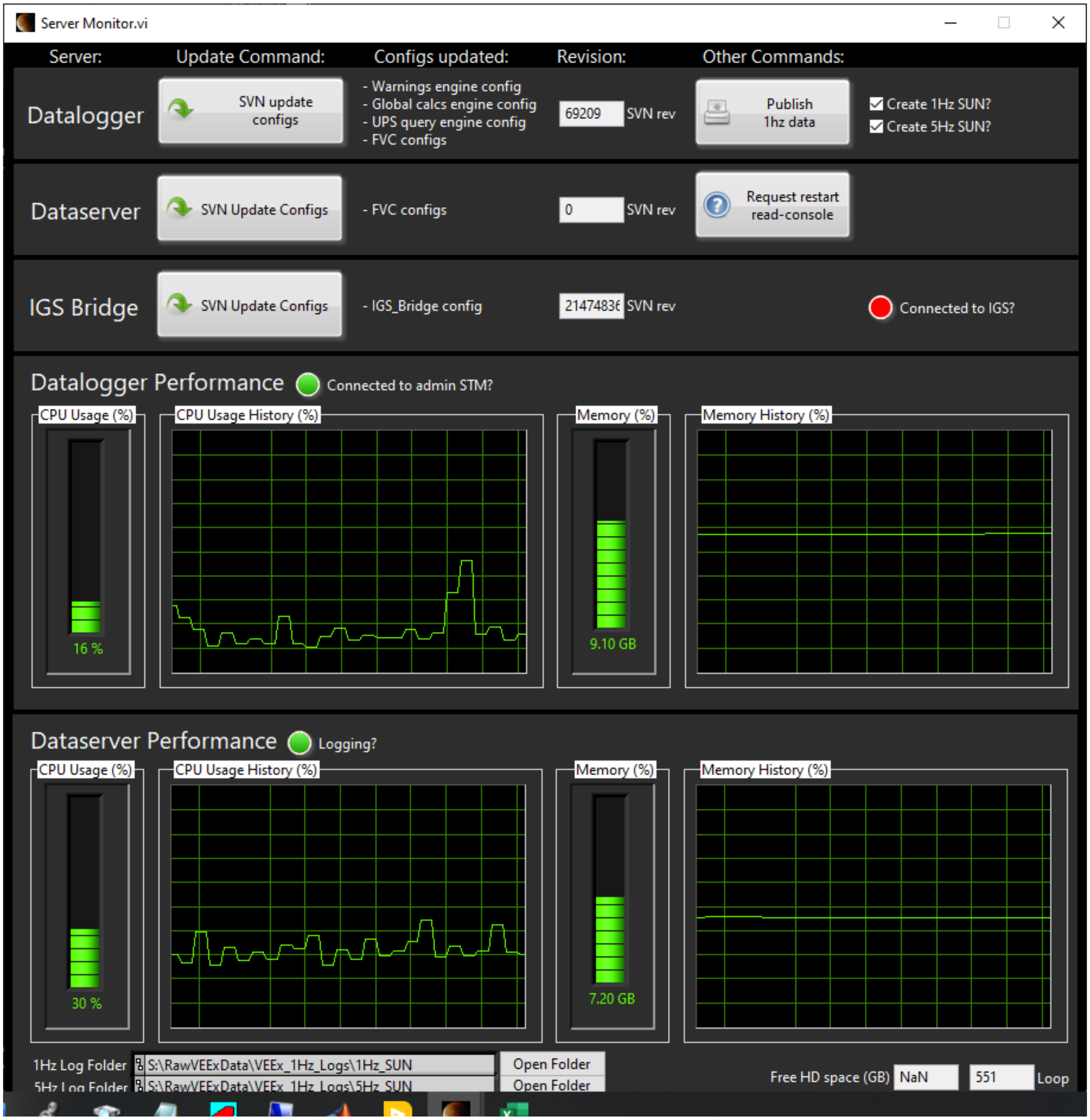
Logging Levels
□ DEBUG
☑ INFO
☑ WARNING
☑ ERROR
☑ CRITICAL

Items	Value	Type
○	Size = 0	Variant

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


Server Monitor.vi

Server: | Update Command: | Configs updated: | Revision: | Other Commands:

Datalogger — SVN update configs — - Warnings engine config - Global calcs engine config - UPS query engine config - FVC configs — 69209 SVN rev — Publish 1hz data — ☑ Create 1Hz SUN? ☑ Create 5Hz SUN?

Dataserver — SVN Update Configs — - FVC configs — 0 SVN rev — Request restart read-console

IGS Bridge — SVN Update Configs — - IGS_Bridge config — 2147483€ SVN rev — ● Connected to IGS?

Datalogger Performance ● Connected to admin STM?

CPU Usage (%) 16 % — Memory (%) 9.10 GB

Dataserver Performance ● Logging?

CPU Usage (%) 30 % — Memory (%) 7.20 GB

1Hz Log Folder S:\RawVEExData\VEEx_1Hz_Logs\1Hz_SUN — Open Folder

5Hz Log Folder S:\RawVEExData\VEEx_1Hz_Logs\5Hz_SUN — Open Folder

Free HD space (GB) NaN — 551 Loop

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.

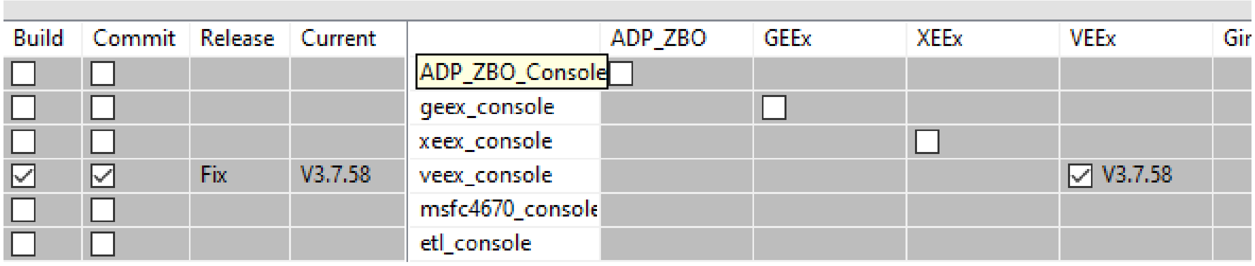
Releasing Software

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

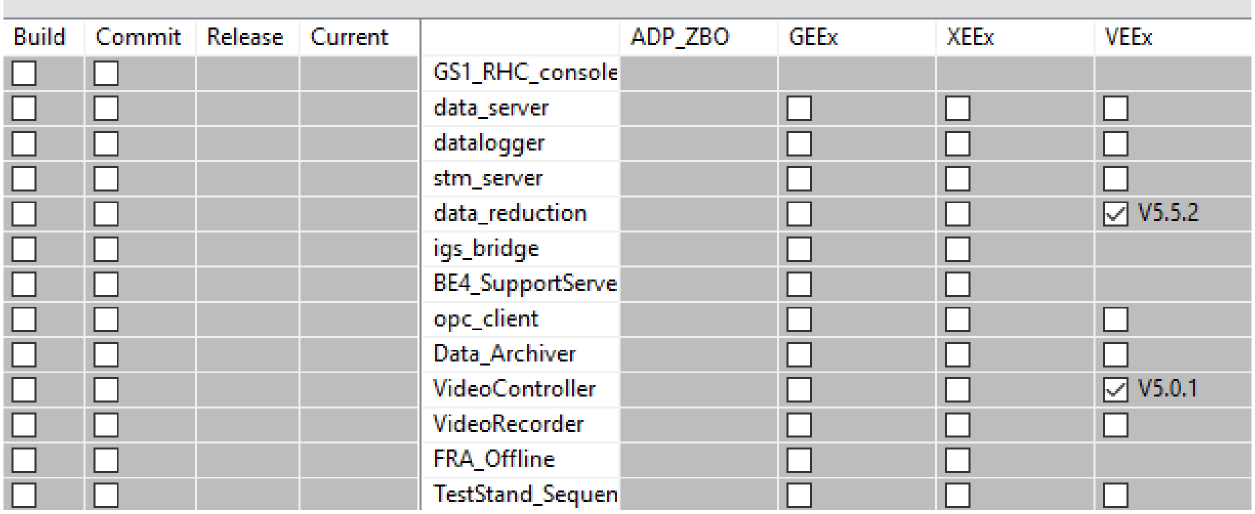
Buildbot itself is not test stand specific. Therefore, one must take care to only and correctly modify the releases for their particular stands. Buildbot releases can be broken into two main uses: 1) releasing new console updates, and 2) updating auxiliary software.

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:



Build	Commit	Release	Current		ADP_ZBO	GEEx	XEEx	VEEx	Gir
□	□			ADP_ZBO_Console	□				
□	□			geex_console		□			
□	□			xeex_console			□		
☑	☑	Fix	V3.7.58	veex_console				☑ V3.7.58	
□	□			msfc4670_console					
□	□			etl_console					

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.

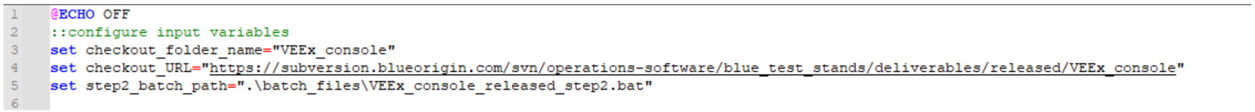


Build	Commit	Release	Current		ADP_ZBO	GEEx	XEEx	VEEx
□	□			GS1_RHC_console				
□	□			data_server		□	□	□
□	□			datalogger		□	□	□
□	□			stm_server		□	□	□
□	□			data_reduction		□	□	☑ V5.5.2
□	□			igs_bridge		□	□	
□	□			BE4_SupportServe		□	□	
□	□			opc_client		□	□	□
□	□			Data_Archiver		□	□	□
□	□			VideoController		□	□	☑ V5.0.1
□	□			VideoRecorder		□	□	□
□	□			FRA_Offline		□	□	
□	□			TestStand_Sequen		□	□	□

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

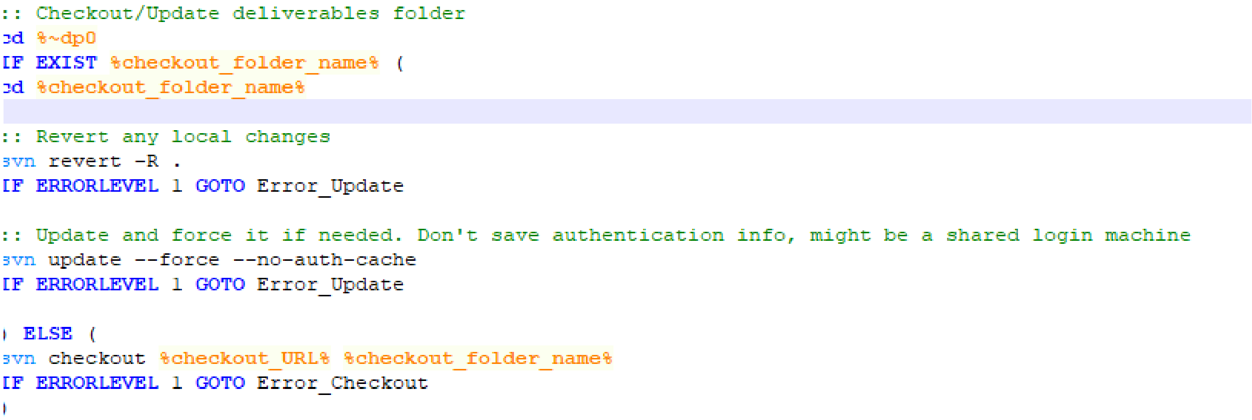
Custom Labview software is typically run through batch files.



```
@ECHO OFF
::configure input variables
set checkout_folder_name="VEEx_console"
set checkout_URL="https://subversion.blueorigin.com/svn/operations-software/blue_test_stands/deliverables/released/VEEx_console"
set step2_batch_path=".\batch_files\VEEx_console_released_step2.bat"

```

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.



```
:: Checkout/Update deliverables folder
cd %~dp0
IF EXIST %checkout_folder_name% (
cd %checkout_folder_name%

:: Revert any local changes
svn revert -R .
IF ERRORLEVEL 1 GOTO Error_Update

:: Update and force it if needed. Don't save authentication info, might be a shared login machine
svn update --force --no-auth-cache
IF ERRORLEVEL 1 GOTO Error_Update

) ELSE (
svn checkout %checkout_URL% %checkout_folder_name%
IF ERRORLEVEL 1 GOTO Error_Checkout
)
```

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.

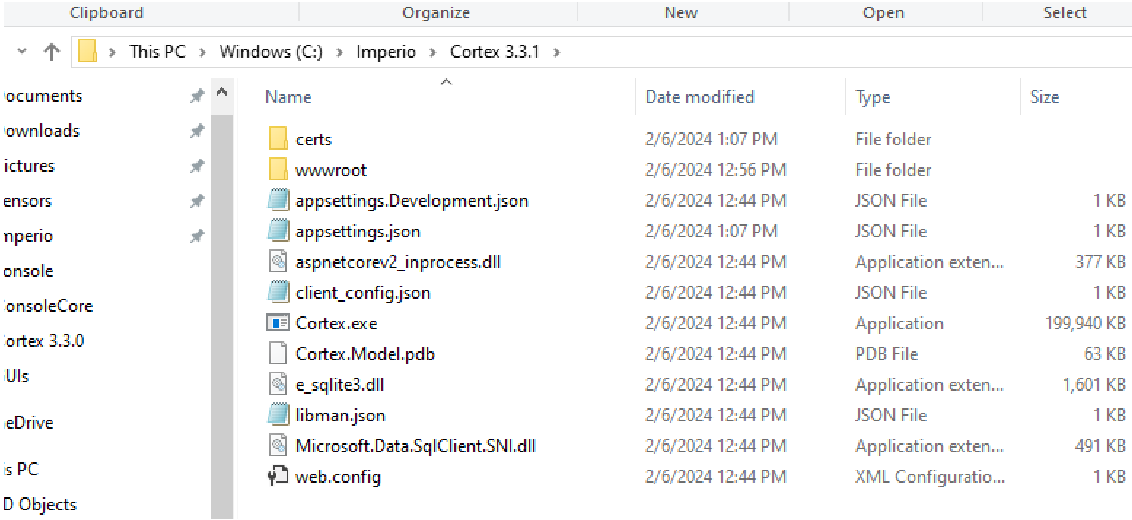
Imperio

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:

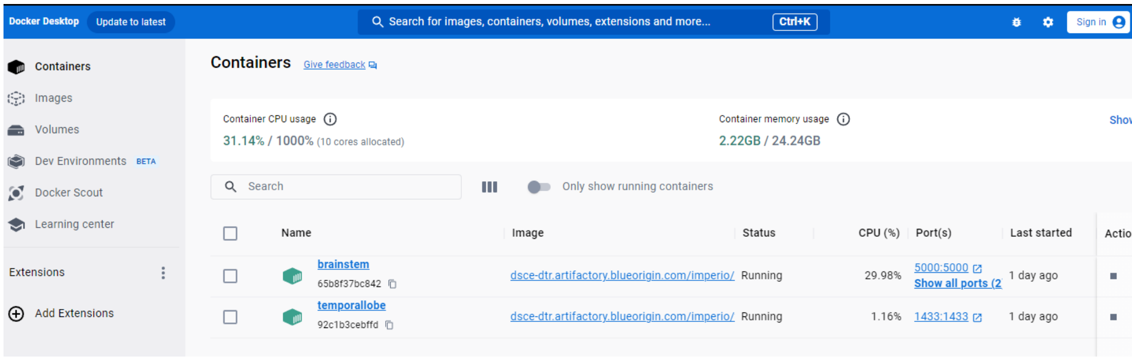
1) Load cortex.exe in the Imperio folder



Name	Date modified	Type	Size
certs	2/6/2024 1:07 PM	File folder	
wwwroot	2/6/2024 12:56 PM	File folder	
appsettings.Development.json	2/6/2024 12:44 PM	JSON File	1 KB
appsettings.json	2/6/2024 1:07 PM	JSON File	1 KB
aspnetcorev2_inprocess.dll	2/6/2024 12:44 PM	Application exten...	377 KB
client_config.json	2/6/2024 12:44 PM	JSON File	1 KB
Cortex.exe	2/6/2024 12:44 PM	Application	199,940 KB
Cortex.Model.pdb	2/6/2024 12:44 PM	PDB File	63 KB
e_sqlite3.dll	2/6/2024 12:44 PM	Application exten...	1,601 KB
libman.json	2/6/2024 12:44 PM	JSON File	1 KB
Microsoft.Data.SqlClient.SNI.dll	2/6/2024 12:44 PM	Application exten...	491 KB
web.config	2/6/2024 12:44 PM	XML Configuratio...	1 KB

2)

3) Run Brainstem and Temporal Lobe from Docker

4) 

Containers

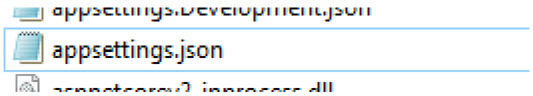
Container CPU usage
31.14% / 1000% (10 cores allocated)

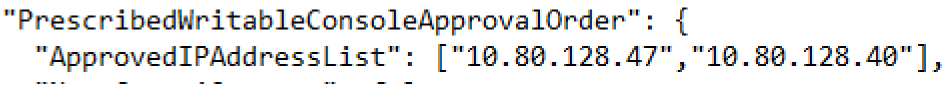
Container memory usage
2.22GB / 24.24GB

Name	Image	Status	CPU (%)	Port(s)	Last started	Actio
brainstem 65b8f37bc842	dsce-dtr.artifactory.blueorigin.com/imperio/	Running	29.98%	5000:5000 Show all ports (2	1 day ago	■
temporallobe 92c1b3cebffd	dsce-dtr.artifactory.blueorigin.com/imperio/	Running	1.16%	1433:1433	1 day ago	■

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.





```
"PrescribedWritableConsoleApprovalOrder": {
  "ApprovedIPAddressList": ["10.80.128.47","10.80.128.40"],
```

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.

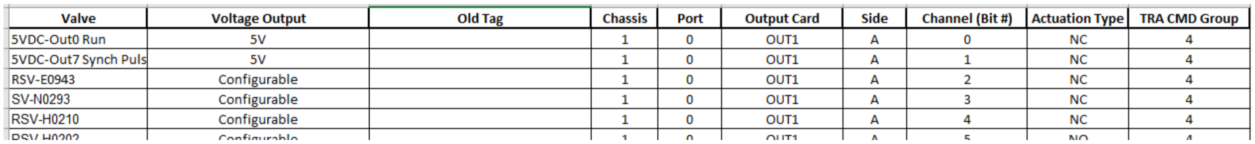
VEEx Terminal Room

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

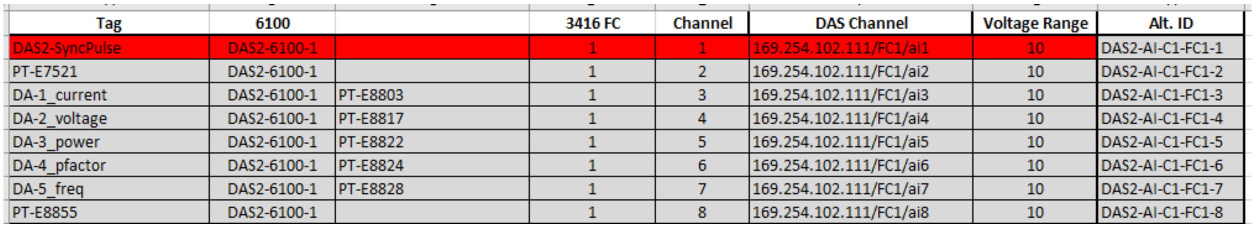
The channel list is a config file that serves two main purposes. The first is to link the STM server with the DAS and FVC systems so that commands from console are transmitted where desired. The second is to provide an organized way of finding the physical connections related to particular devices. The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

FVC



Valve	Voltage Output	Old Tag	Chassis	Port	Output Card	Side	Channel (Bit #)	Actuation Type	TRA CMD Group
5VDC-Out0 Run	5V		1	0	OUT1	A	0	NC	4
5VDC-Out7 Synch Puls	5V		1	0	OUT1	A	1	NC	4
RSV-E0943	Configurable		1	0	OUT1	A	2	NC	4
SV-N0293	Configurable		1	0	OUT1	A	3	NC	4
RSV-H0210	Configurable		1	0	OUT1	A	4	NC	4
RSV-H0202	Configurable		1	0	OUT1	A	5	NO	4

DAS 6100



Tag	6100		3416 FC	Channel	DAS Channel	Voltage Range	Alt. ID
DAS2-SyncPulse	DAS2-6100-1		1	1	169.254.102.111/FC1/ai1	10	DAS2-AI-C1-FC1-1
PT-E7521	DAS2-6100-1		1	2	169.254.102.111/FC1/ai2	10	DAS2-AI-C1-FC1-2
DA-1_current	DAS2-6100-1	PT-E8803	1	3	169.254.102.111/FC1/ai3	10	DAS2-AI-C1-FC1-3
DA-2_voltage	DAS2-6100-1	PT-E8817	1	4	169.254.102.111/FC1/ai4	10	DAS2-AI-C1-FC1-4
DA-3_power	DAS2-6100-1	PT-E8822	1	5	169.254.102.111/FC1/ai5	10	DAS2-AI-C1-FC1-5
DA-4_pfactor	DAS2-6100-1	PT-E8824	1	6	169.254.102.111/FC1/ai6	10	DAS2-AI-C1-FC1-6
DA-5_freq	DAS2-6100-1	PT-E8828	1	7	169.254.102.111/FC1/ai7	10	DAS2-AI-C1-FC1-7
PT-E8855	DAS2-6100-1		1	8	169.254.102.111/FC1/ai8	10	DAS2-AI-C1-FC1-8

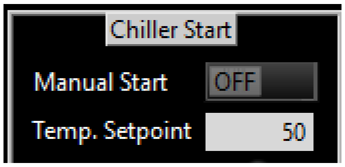
DAS 6150

Tag	6150	Old	Channel	DAS Channel	OTD	Alt. ID
	DAS1-6150-1		1	169.254.101.101/ai1	OFF	
	DAS1-6150-1		2	169.254.101.101/ai2	OFF	
TC-E0930	DAS1-6150-1		3	169.254.101.101/ai3	OFF	
TC-H0430	DAS1-6150-1		4	169.254.101.101/ai4	OFF	
TC-N0730	DAS1-6150-1		5	169.254.101.101/ai5	OFF	

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.



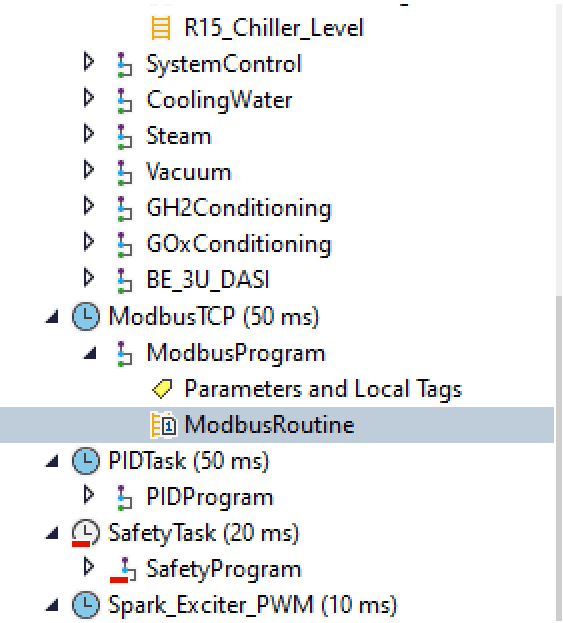
Chiller Start

Manual Start OFF

Temp. Setpoint 50

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.


R15_Chiller_Level
SystemControl
CoolingWater
Steam
Vacuum
GH2Conditioning
GOxConditioning
BE_3U_DASI
ModbusTCP (50 ms)
ModbusProgram
Parameters and Local Tags
ModbusRoutine
PIDTask (50 ms)
PIDProgram
SafetyTask (20 ms)
SafetyProgram
Spark_Exciter_PWM (10 ms)