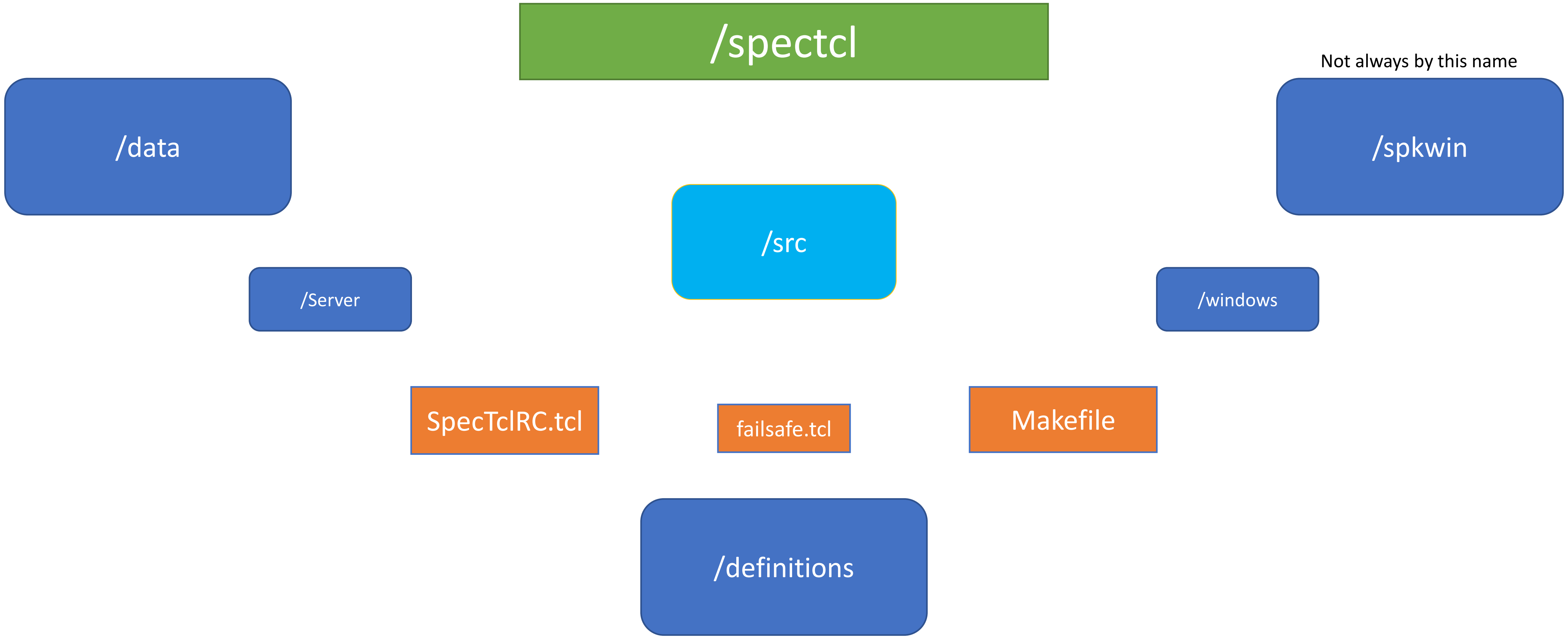
SpecTcl/SpecTk

Kenny Haak @ MSU

ToC

Level 0 – Software

SpecTcl Layout



SpecTcl Layout 2

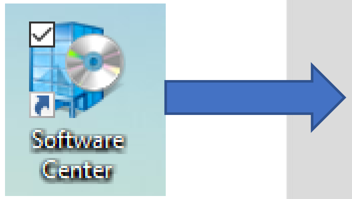
Big 3: Definitions – Data – Windows/SpecTk →

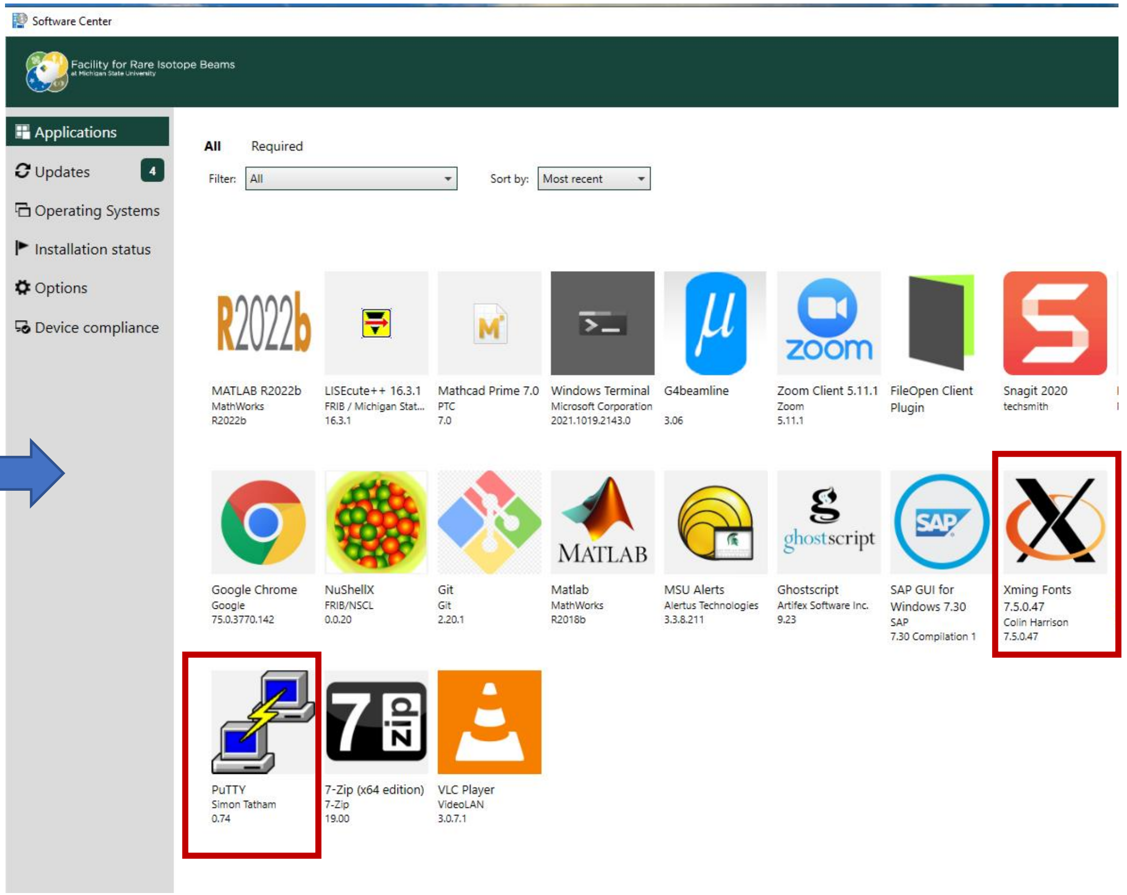
- Makefile
    - SpecTcl is a program written with Tcl/Tk language
    - But analysis is performed by code written in C^{++} so we need to compile
- SpecTclRC.tcl
    - .tcl extension means Tcl/Tk language
    - RC stands for "Remote Control"
    - This is where you can add on Tcl/Tk functionality such as:
        - Modify SpecTcl into a server to host SpecTk connections
        - Change gui (button size/color/function)
        - Apply calibrations upon start up
- Server/
    - Directory which contains code to compile SpecTcl as a server
    - May need to setup yourself

- **definitions/**
    - Saved configurations for SpecTcl
    - Includes:
        - Variables (calibration values)
        - Gates
        - Spectra (and which gates are applied to which spectra)
- failsafe.tcl
    - The 'autosave' definition file
    - Always there, comes with compiling (maybe?)
    - I've never used it
- **data/**
    - Contains paths to your data (you don't want it to actually be stored here)
    - Best method is to construct cluster files (.clu extension)
- windows/
    - Saved configurations for the default display program: "**Xamine**"
- **spkwin/**
    - Saved configurations for the optimal display program: "**SpecTk**"

Setup

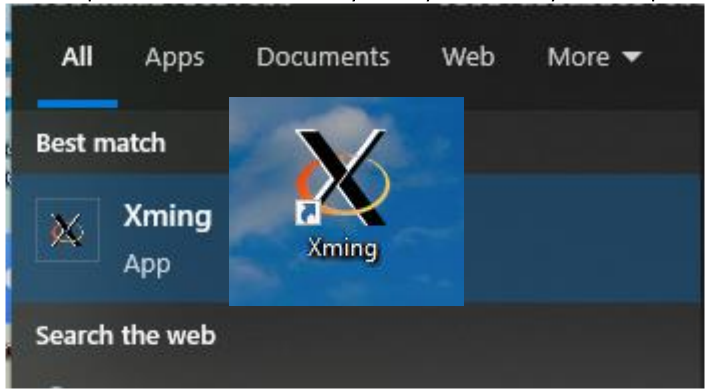
1) Install Xming and Putty from the Software Center shortcut on your desktop



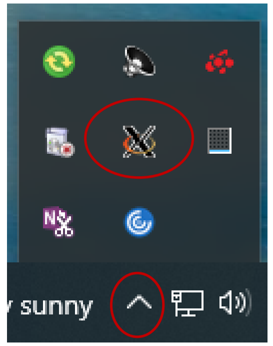

Software Center
Facility for Rare Isotope Beams
Applications
Updates 4
Operating Systems
Installation status
Options
Device compliance
All
Required
Filter: All
Sort by: Most recent
MATLAB R2022b MathWorks R2022b
LISEcute++ 16.3.1 FRIB / Michigan Stat... 16.3.1
Mathcad Prime 7.0 PTC 7.0
Windows Terminal Microsoft Corporation 2021.1019.2143.0
G4beamline 3.06
Zoom Client 5.11.1 Zoom 5.11.1
FileOpen Client Plugin
Snagit 2020 techsmith
Google Chrome Google 75.0.3770.142
NuShellX FRIB/NSCL 0.0.20
Git Git 2.20.1
Matlab MathWorks R2018b
MSU Alerts Alertus Technologies 3.3.8.211
Ghostscript Artifex Software Inc. 9.23
SAP GUI for Windows 7.30 SAP 7.30 Compilation 1
Xming Fonts 7.5.0.47 Colin Harrison 7.5.0.47
PuTTY Simon Tatham 0.74
7-Zip (x64 edition) 7-Zip 19.00
VLC Player VideoLAN 3.0.7.1

2) Run Xming either through windows search or desktop shortcut

This step needs to be done every time you restart your computer



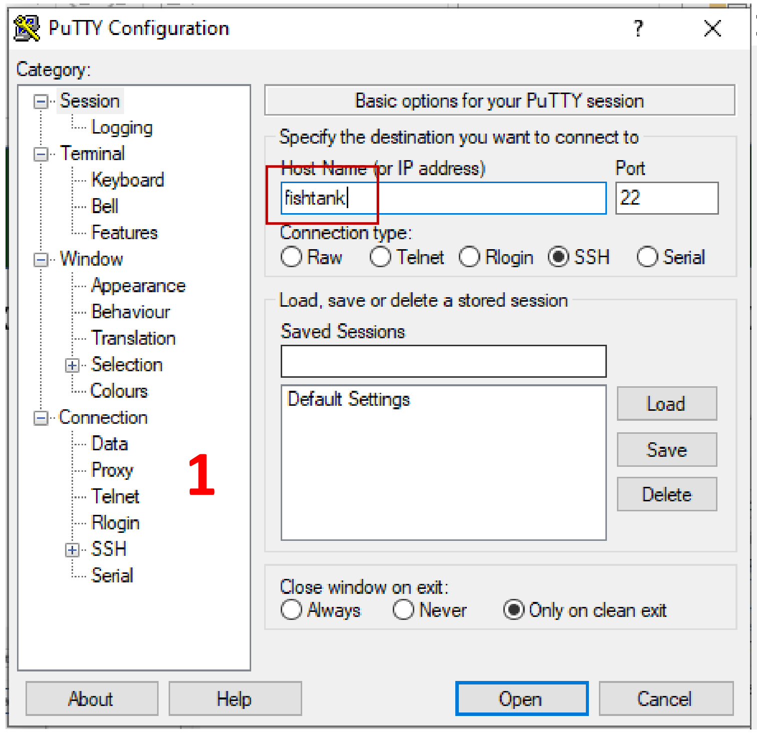
When running, you should see this icon in the icons tray in the bottom right of you screen



Click the arrow to expand

Setup

3) Set up fishtank access with X11 forwarding (how Putty communicates with Xming



Specify the destination you want to connect to

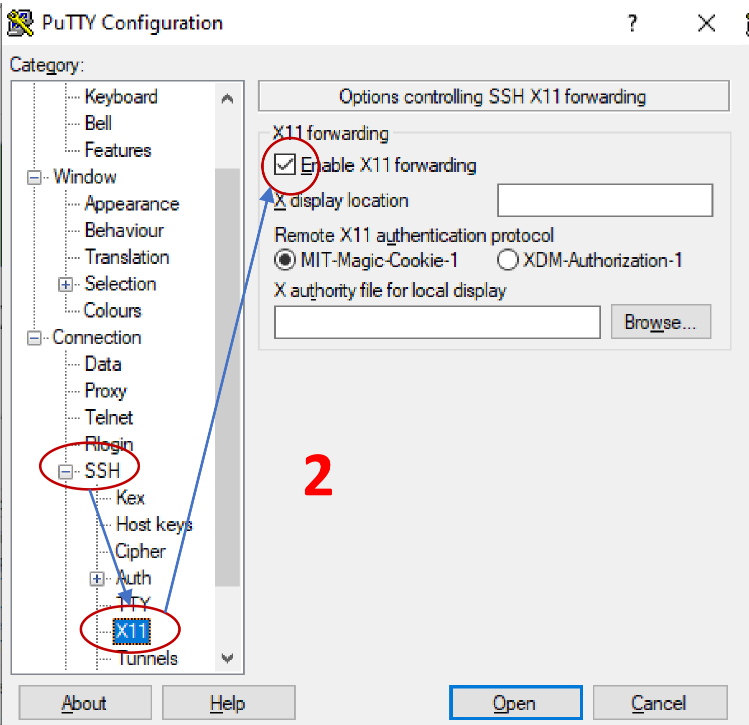
Host Name (or IP address): fishtank | Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: Always Never [✓ Only on clean exit]



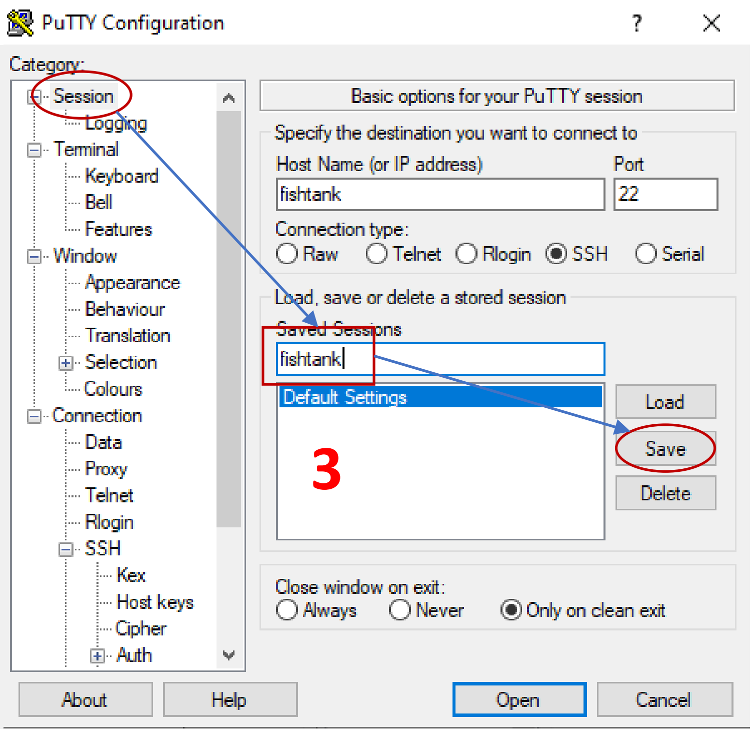
X11 forwarding

[✓] Enable X11 forwarding

X display location:

Remote X11 authentication protocol: [✓ MIT-Magic-Cookie-1] XDM-Authorization-1

X authority file for local display:



Specify the destination you want to connect to

Host Name (or IP address): fishtank | Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

Saved Sessions: fishtank

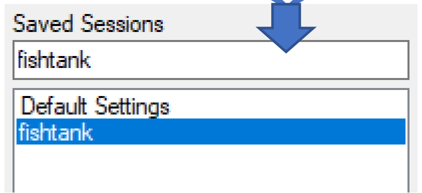
Default Settings

Close window on exit: Always Never [✓ Only on clean exit]

Type "fishtank" into Hostname

- Navigate to the **SSH → X11** Category tab
- Check the "Enable X11 forwarding" box

- Navigate back to **Session** tab
- Type fishtank into saved sessions
- Click Save button



Saved Sessions

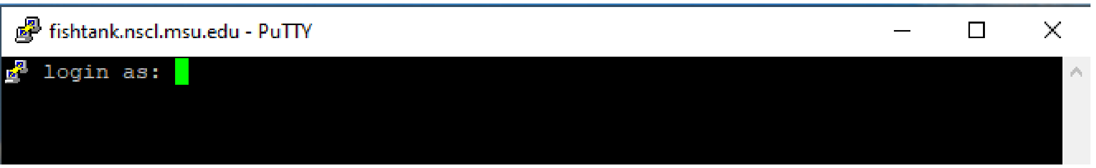
fishtank

Default Settings
fishtank

- Once finished you should always see a "fishtank" option in Saved Sessions
- In the future you only need to double click this option, no need to set X11 forwarding every time

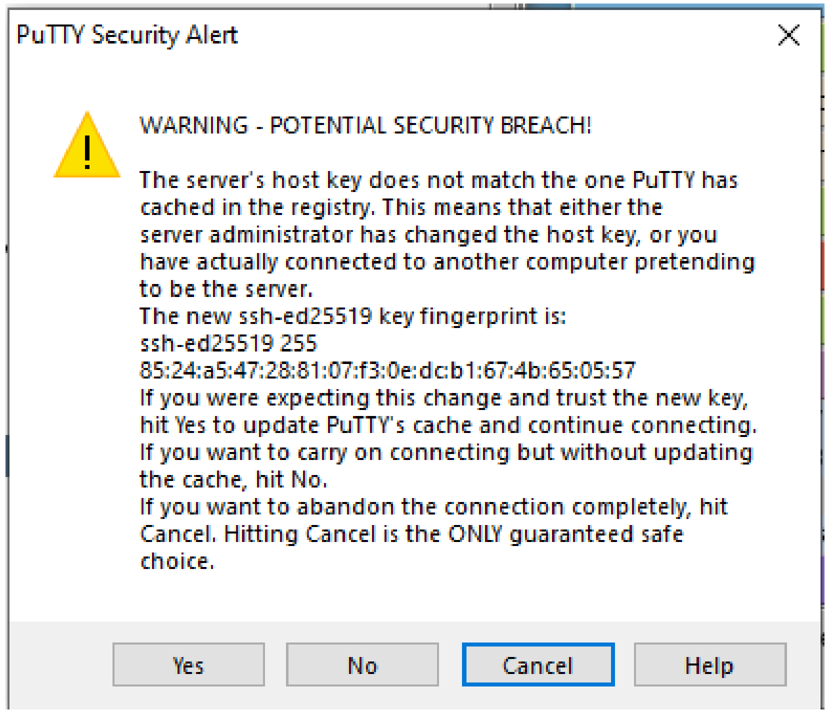
Setup

4) Login with your FRIB credentials



```
fishtank.nscl.msu.edu - PuTTY
login as:
```

You may see the following window:



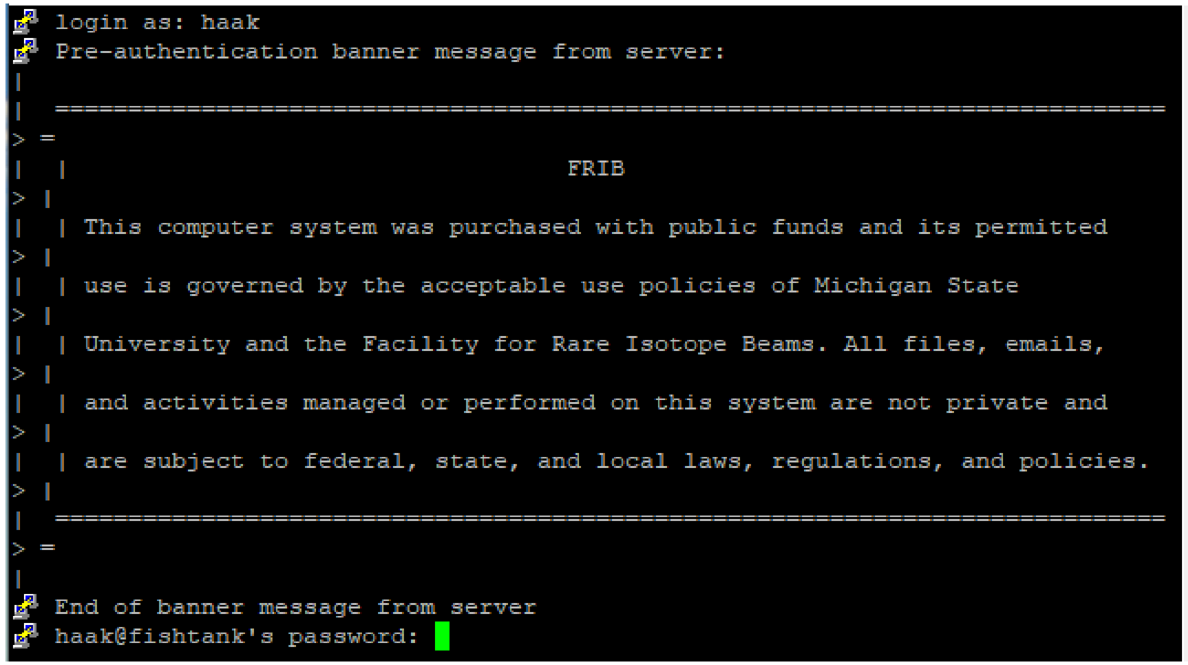
PuTTY Security Alert

WARNING - POTENTIAL SECURITY BREACH!

The server's host key does not match the one PuTTY has cached in the registry. This means that either the server administrator has changed the host key, or you have actually connected to another computer pretending to be the server.
The new ssh-ed25519 key fingerprint is:
ssh-ed25519 255
85:24:a5:47:28:81:07:f3:0e:dc:b1:67:4b:65:05:57
If you were expecting this change and trust the new key, hit Yes to update PuTTY's cache and continue connecting.
If you want to carry on connecting but without updating the cache, hit No.
If you want to abandon the connection completely, hit Cancel. Hitting Cancel is the ONLY guaranteed safe choice.

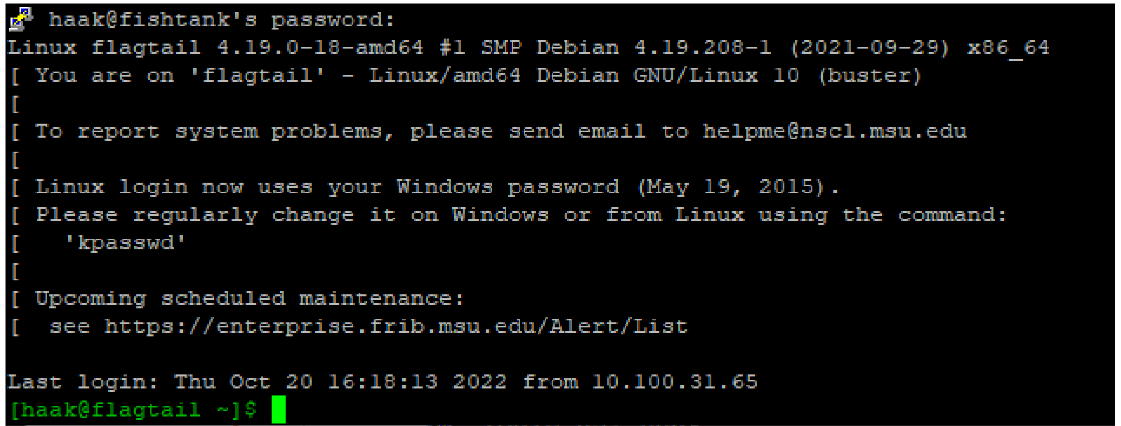
Simply click yes and proceed to enter your FRIB username.

After entering your username you should see the following prompt. That means you are in the right place



```
login as: haak
Pre-authentication banner message from server:
|
|  ================================================================================
> =
|  |                                    FRIB
> |
|  | This computer system was purchased with public funds and its permitted
> |
|  | use is governed by the acceptable use policies of Michigan State
> |
|  | University and the Facility for Rare Isotope Beams. All files, emails,
> |
|  | and activities managed or performed on this system are not private and
> |
|  | are subject to federal, state, and local laws, regulations, and policies.
> |
|  ================================================================================
> =
|
End of banner message from server
haak@fishtank's password:
```

A successful login

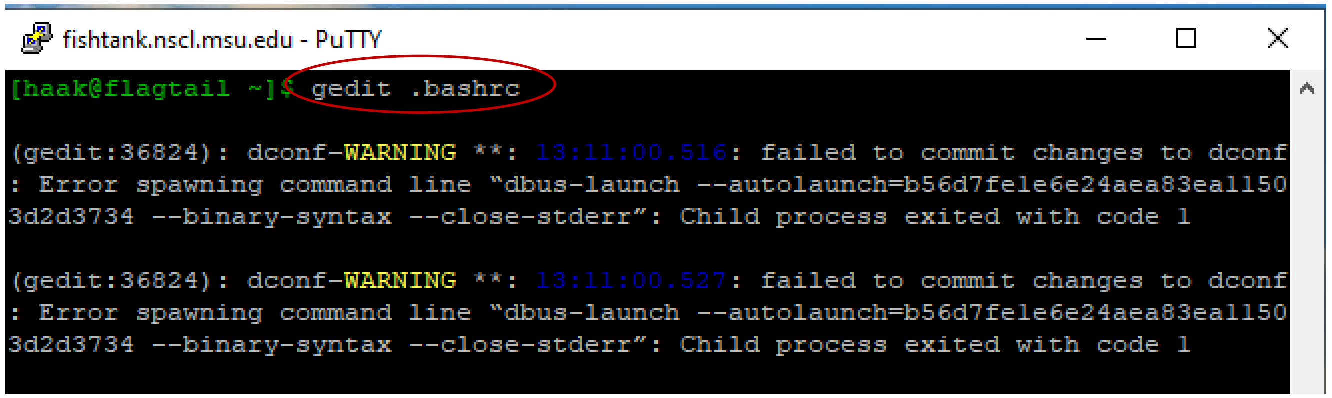


```
haak@fishtank's password:
Linux flagtail 4.19.0-18-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) x86_64
[ You are on 'flagtail' - Linux/amd64 Debian GNU/Linux 10 (buster)
[
[ To report system problems, please send email to helpme@nscl.msu.edu
[
[ Linux login now uses your Windows password (May 19, 2015).
[ Please regularly change it on Windows or from Linux using the command:
[   'kpasswd'
[
[ Upcoming scheduled maintenance:
[  see https://enterprise.frib.msu.edu/Alert/List

Last login: Thu Oct 20 16:18:13 2022 from 10.100.31.65
[haak@flagtail ~]$
```

Setup

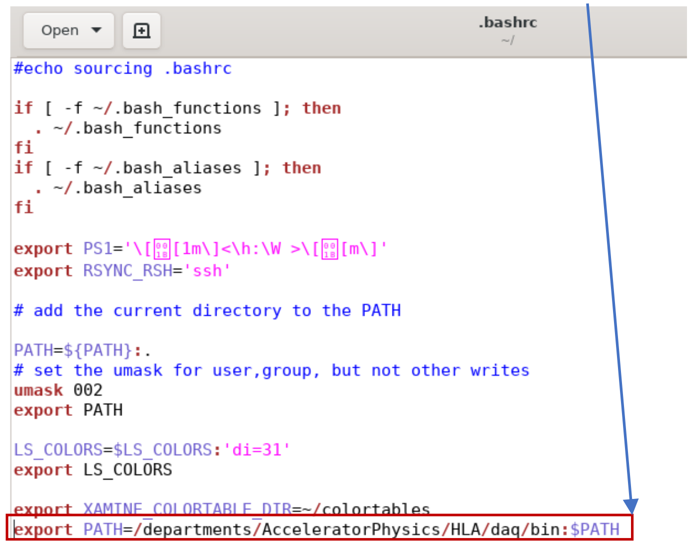
5) Export the path for the SpecTk program

In your home directory, edit the .bashrc file (here I use the gedit program)



```
[haak@flagtail ~]$ gedit .bashrc

(gedit:36824): dconf-WARNING **: 13:11:00.516: failed to commit changes to dconf
: Error spawning command line “dbus-launch --autolaunch=b56d7fe1e6e24aea83ea1150
3d2d3734 --binary-syntax --close-stderr”: Child process exited with code 1

(gedit:36824): dconf-WARNING **: 13:11:00.527: failed to commit changes to dconf
: Error spawning command line “dbus-launch --autolaunch=b56d7fe1e6e24aea83ea1150
3d2d3734 --binary-syntax --close-stderr”: Child process exited with code 1
```

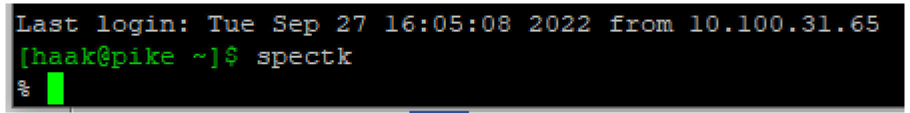
Copy and paste this line of bash code
export PATH=/departments/AcceleratorPhysics/HLA/daq/bin:$PATH



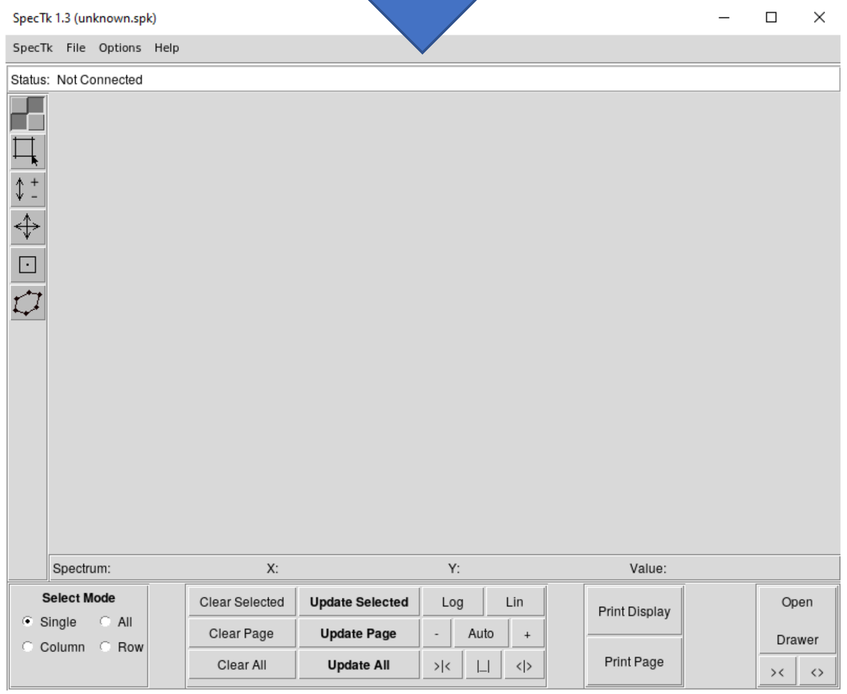
```
#echo sourcing .bashrc

if [ -f ~/.bash_functions ]; then
  . ~/.bash_functions
fi
if [ -f ~/.bash_aliases ]; then
  . ~/.bash_aliases
fi

export PS1='\[[1m\]<\h:\W >\[[m\]'
export RSYNC_RSH='ssh'

# add the current directory to the PATH

PATH=${PATH}:.
# set the umask for user,group, but not other writes
umask 002
export PATH

LS_COLORS=$LS_COLORS:'di=31'
export LS_COLORS

export XAMINE_COLORTABLE_DIR=~/colortables
export PATH=/departments/AcceleratorPhysics/HLA/daq/bin:$PATH
```

Make sure to save before you exit!

6) *Restart* Putty and run the command “spectk”



```
Last login: Tue Sep 27 16:05:08 2022 from 10.100.31.65
[haak@pike ~]$ spectk
%
```

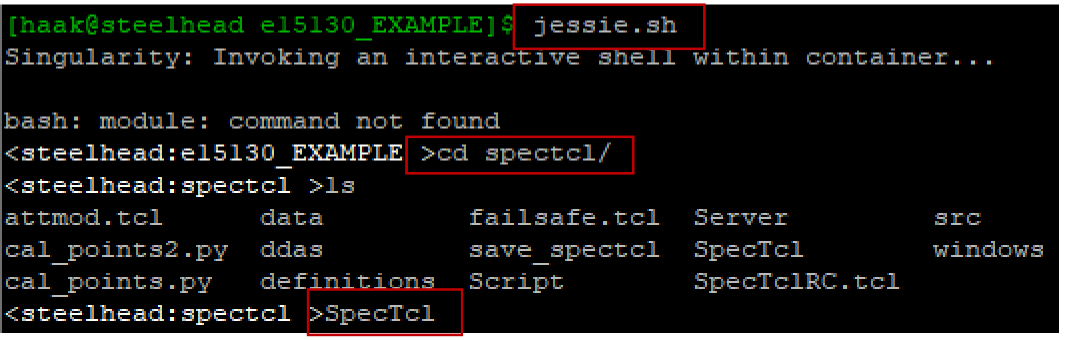


SpecTk, the spectra displayer application should appear

Setup

7) Run SpecTcl in the shell environment given to you by the research group

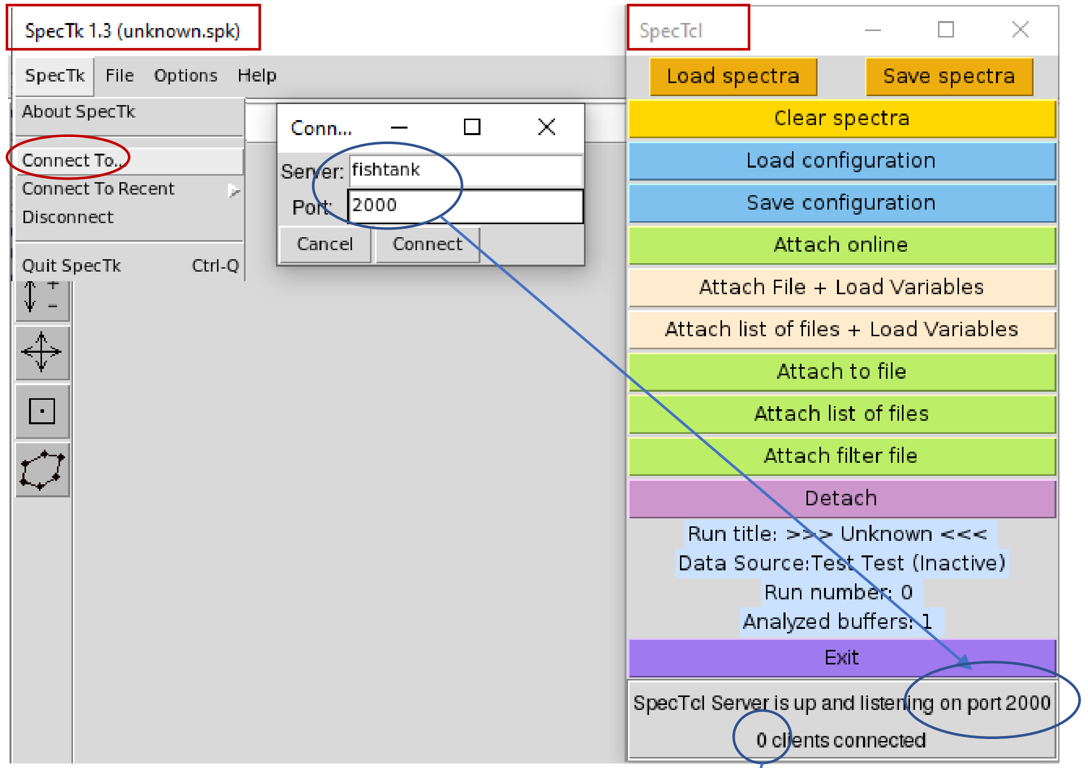
Open another instance of fishtank
Run appropriate shell and navigate to SpecTcl executable location



```
[haak@steelhead e15130_EXAMPLE]$ jessie.sh
Singularity: Invoking an interactive shell within container...

bash: module: command not found
<steelhead:e15130_EXAMPLE >cd spectcl/
<steelhead:spectcl >ls
attmod.tcl       data          failsafe.tcl  Server         src
cal_points2.py   ddas          save_spectcl  SpecTcl        windows
cal_points.py    definitions   Script        SpecTclRC.tcl
<steelhead:spectcl >SpecTcl
```

Run SpecTcl with the SpecTcl command

8) In SpecTk, connect to port number shown in SpecTcl GUI

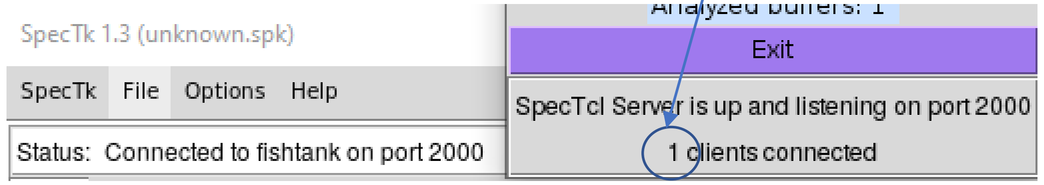


Server: fishtank
Port: 2000

Run title: >>> Unknown <<<
Data Source:Test Test (Inactive)
Run number: 0
Analyzed buffers: 1

SpecTcl Server is up and listening on port 2000
0 clients connected

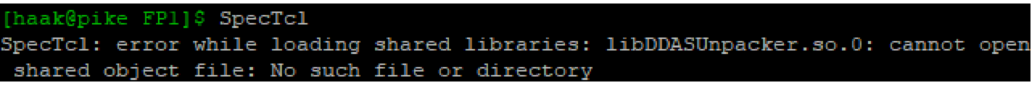
- You may also use "localhost" as a server name, I use "fishtank"
- The port can vary from build to build, but you can set the exact number yourself if you desire (it's a line of code in Server/Server.Tcl file)



Status: Connected to fishtank on port 2000

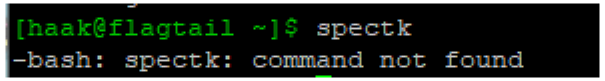
SpecTcl Server is up and listening on port 2000
1 clients connected

A Couple Common Issues



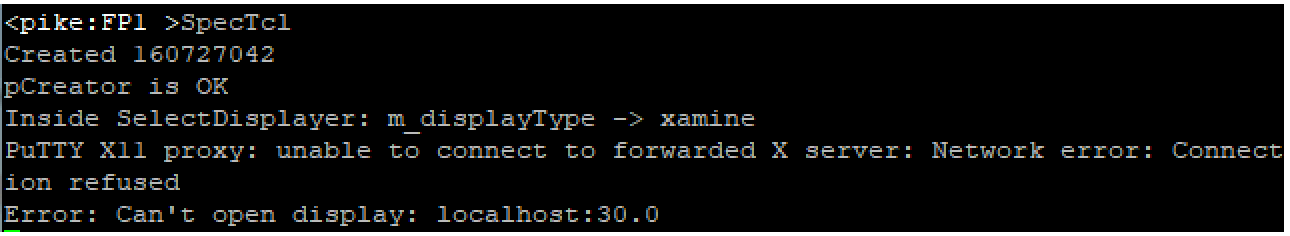
```
[haak@pike FP1]$ SpecTcl
SpecTcl: error while loading shared libraries: libDDASUnpacker.so.0: cannot open
 shared object file: No such file or directory
```

Problem: Attempting to run SpecTcl in the wrong virtual environment.
Solution: Run a shell script to setup the proper environment



```
[haak@flagtail ~]$ spectk
-bash: spectk: command not found
```

Problem: SpecTk path not set in .bashrc file
Solution: See step 5

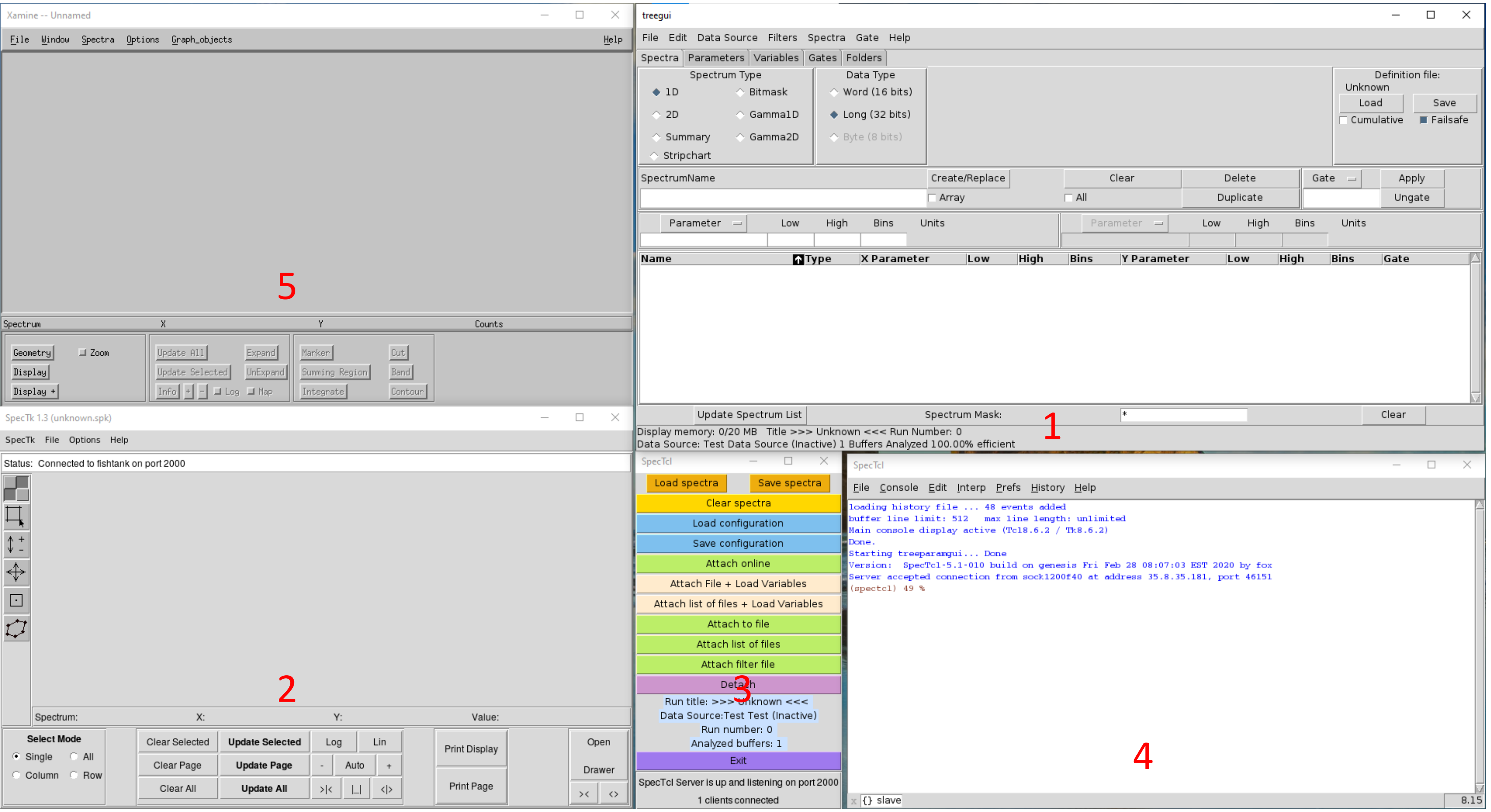


```
<pike:FP1 >SpecTcl
Created 160727042
pCreator is OK
Inside SelectDisplayer: m_displayType -> xamine
PuTTY X11 proxy: unable to connect to forwarded X server: Network error: Connect
ion refused
Error: Can't open display: localhost:30.0
```

Problem: Xming not running/enabled.
Solution: See step 2 and 3 (Run Xming, ensure X11 forwarding)

Level 1 – Basic Navigation

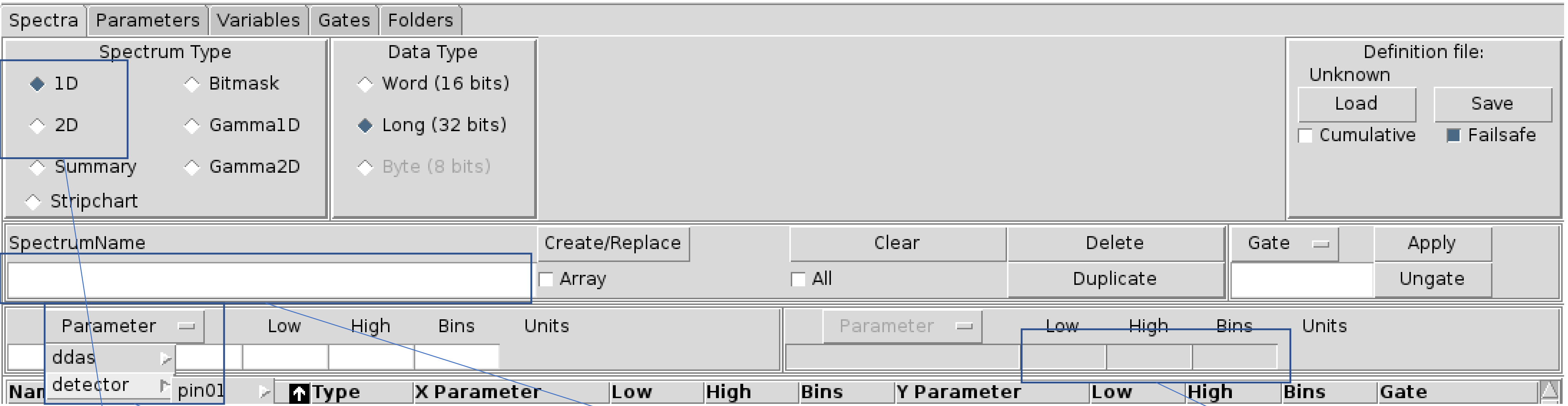
The Windows


Xamine -- Unnamed
File Window Spectra Options Graph_objects Help
5
Spectrum X Y Counts
Geometry Zoom Update All Expand Marker Cut
Display Update Selected UnExpand Summing Region Band
Display + Info + - Log Map Integrate Contour
SpecTk 1.3 (unknown.spk)
SpecTk File Options Help
Status: Connected to fishtank on port 2000
2
Spectrum: X: Y: Value:
Select Mode Single All Column Row
Clear Selected Update Selected Log Lin Print Display Open
Clear Page Update Page - Auto + Drawer
Clear All Update All Print Page
treegui
File Edit Data Source Filters Spectra Gate Help
Spectra Parameters Variables Gates Folders
Spectrum Type 1D Bitmask 2D Gamma1D Summary Gamma2D Stripchart
Data Type Word (16 bits) Long (32 bits) Byte (8 bits)
Definition file: Unknown Load Save Cumulative Failsafe
SpectrumName Create/Replace Array Clear All Delete Duplicate Gate Apply Ungate
Parameter Low High Bins Units
Name Type X Parameter Low High Bins Y Parameter Low High Bins Gate
Update Spectrum List Spectrum Mask: * Clear
1
Display memory: 0/20 MB Title >>> Unknown <<< Run Number: 0
Data Source: Test Data Source (Inactive) 1 Buffers Analyzed 100.00% efficient
SpecTcl
Load spectra Save spectra
Clear spectra
Load configuration
Save configuration
Attach online
Attach File + Load Variables
Attach list of files + Load Variables
Attach to file
Attach list of files
Attach filter file
Detach
3
Run title: >>> Unknown <<<
Data Source:Test Test (Inactive)
Run number: 0
Analyzed buffers: 1
Exit
SpecTcl Server is up and listening on port 2000
1 clients connected
SpecTcl
File Console Edit Interp Prefs History Help
4
{} slave 8.15

```
loading history file ... 48 events added
buffer line limit: 512   max line length: unlimited
Main console display active (Tcl8.6.2 / Tk8.6.2)
Done.
Starting treeparamgui... Done
Version:  SpecTcl-5.1-010 build on genesis Fri Feb 28 08:07:03 EST 2020 by fox
Server accepted connection from sock1200f40 at address 35.8.35.181, port 46151
(spectcl) 49 %
```

1. Main GUI for creating spectra and applying gates
2. SpecTk display window for seeing spectra and drawing contours
3. Radio buttons for attaching data
4. Command line interface which you can write Tcl/Tk commands
5. Default spectra display (you won't use if using SpecTk)

Creating Spectra

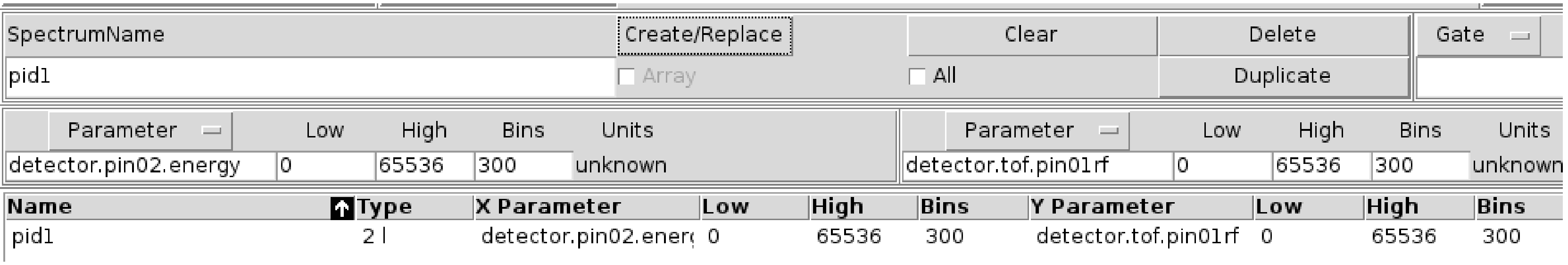


Title the spectra

Set bounds for axis, and bin count

Choose 1D or 2D

Select value to be displayed



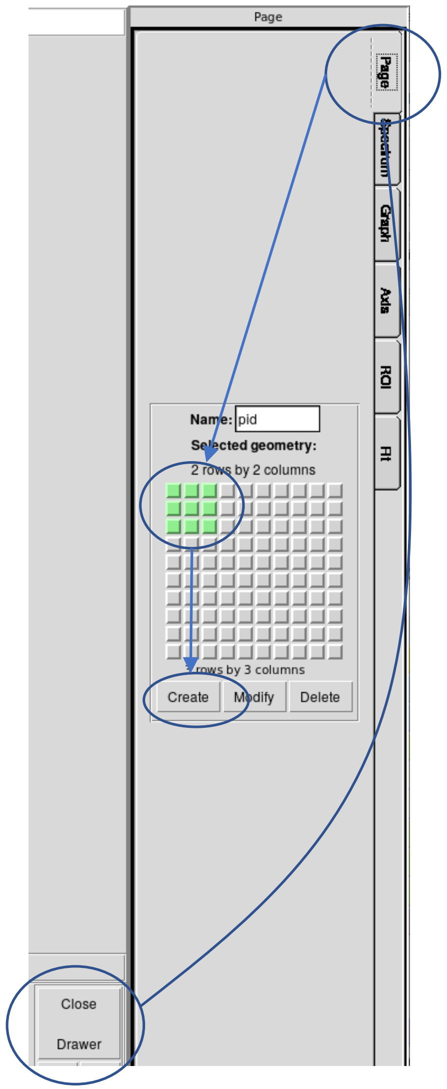
SpectrumName: pid1

Parameter	Low	High	Bins	Units	Parameter	Low	High	Bins	Units
detector.pin02.energy	0	65536	300	unknown	detector.tof.pin01rf	0	65536	300	unknown

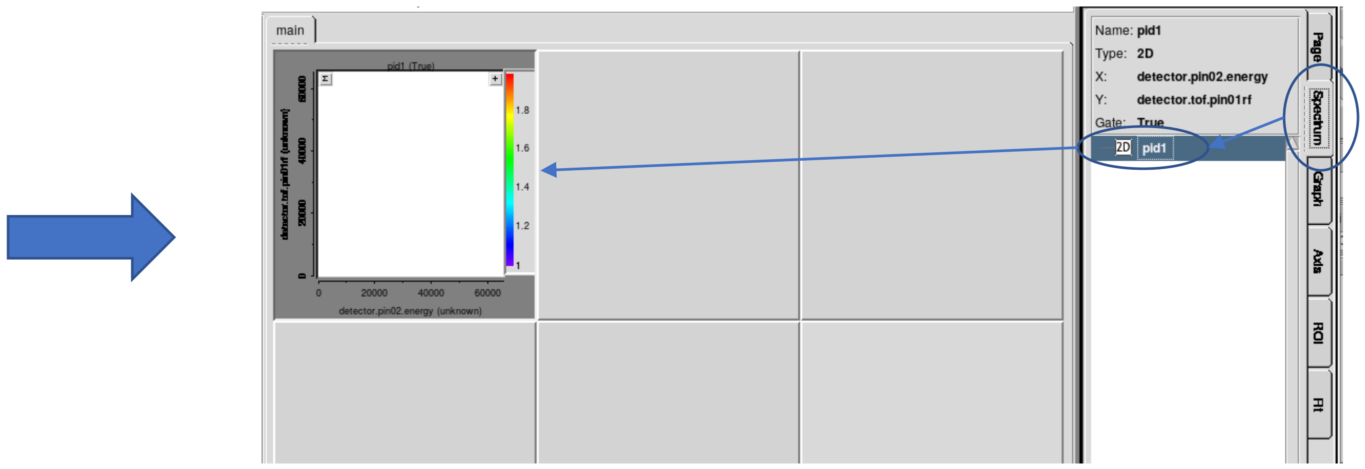
Name	Type	X Parameter	Low	High	Bins	Y Parameter	Low	High	Bins
pid1	2	detector.pin02.energ	0	65536	300	detector.tof.pin01rf	0	65536	300

Note: to save memory, avoid making more than 300 bins on a 2D spectra

Displaying Spectra



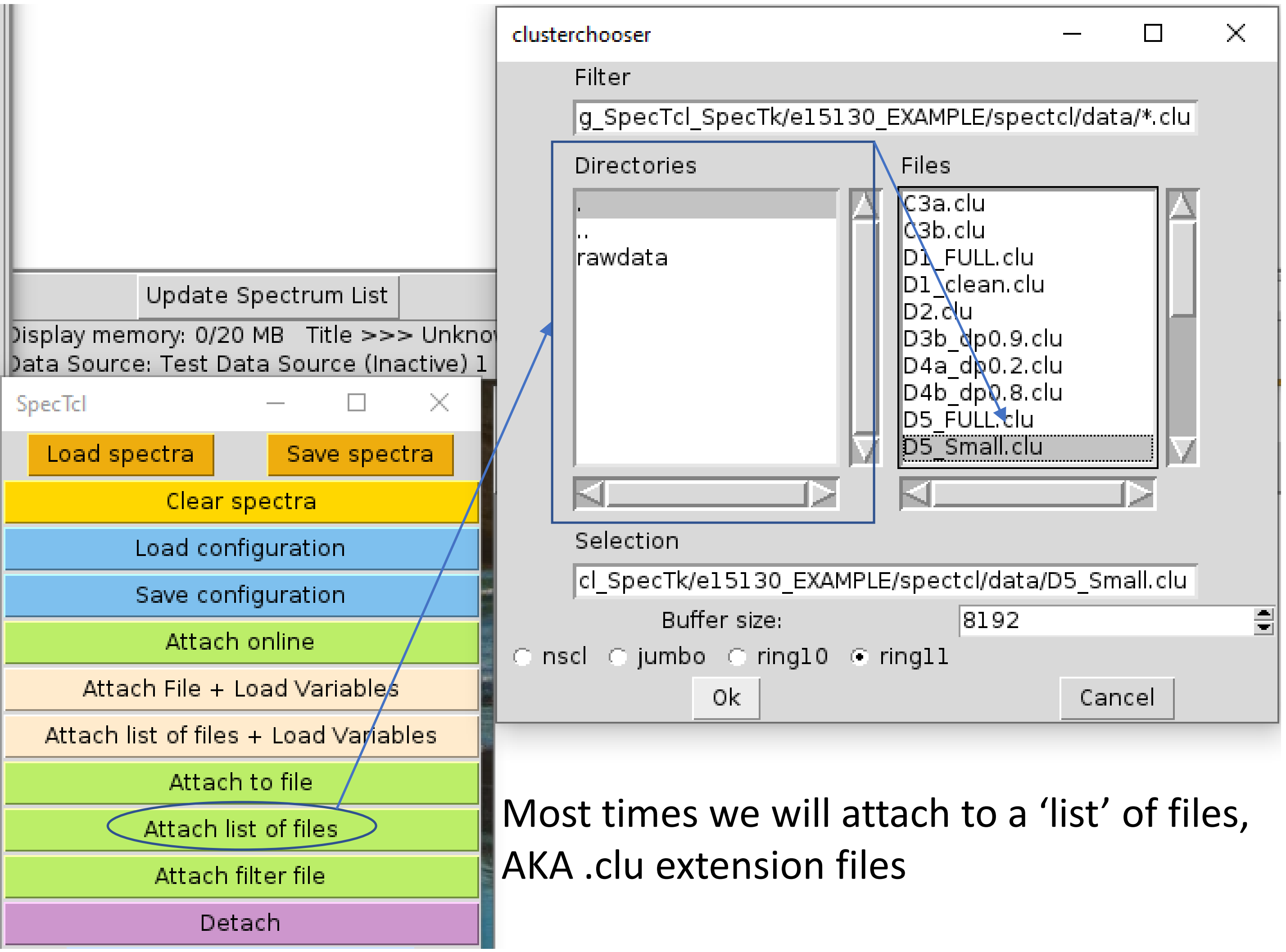
1. Open drawer
2. In **Page** tab click on the grid to make dimensions of the spectra page
3. Click Create


main
pid1 (True)
detector.pin02.energy (unknown)
detector.tof.pin01rf (unknown)
Page
Spectrum
Graph
Axis
ROI
Fit
2D pid1

Name: pid1
Type: 2D
X: detector.pin02.energy
Y: detector.tof.pin01rf
Gate: True

1. Go to **Spectrum** tab
2. Double click on a spectra to display

Loading Data



Update Spectrum List

Display memory: 0/20 MB Title >>> Unkno

Data Source: Test Data Source (Inactive) 1

SpecTcl

Load spectra | Save spectra

Clear spectra

Load configuration

Save configuration

Attach online

Attach File + Load Variables

Attach list of files + Load Variables

Attach to file

Attach list of files

Attach filter file

Detach

clusterchooser

Filter

g_SpecTcl_SpecTk/e15130_EXAMPLE/spectcl/data/*.clu

Directories	Files
.	C3a.clu
..	C3b.clu
rawdata	D1_FULL.clu
	D1_clean.clu
	D2.clu
	D3b_dp0.9.clu
	D4a_dp0.2.clu
	D4b_dp0.8.clu
	D5_FULL.clu
	D5_Small.clu

Selection

cl_SpecTk/e15130_EXAMPLE/spectcl/data/D5_Small.clu

Buffer size: 8192

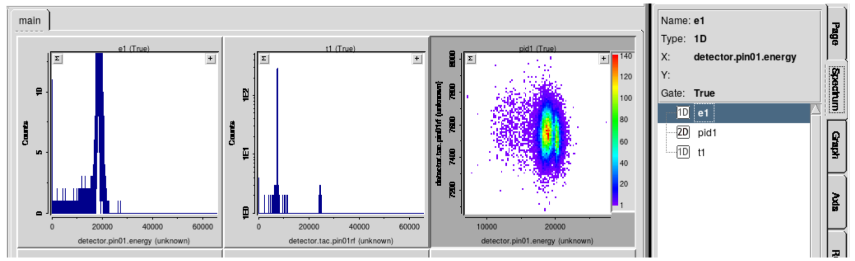
nscl jumbo ring10 [✓ ring11]

Ok Cancel

Navigate by double clicking on left hand side until you find the cluster file you want to load

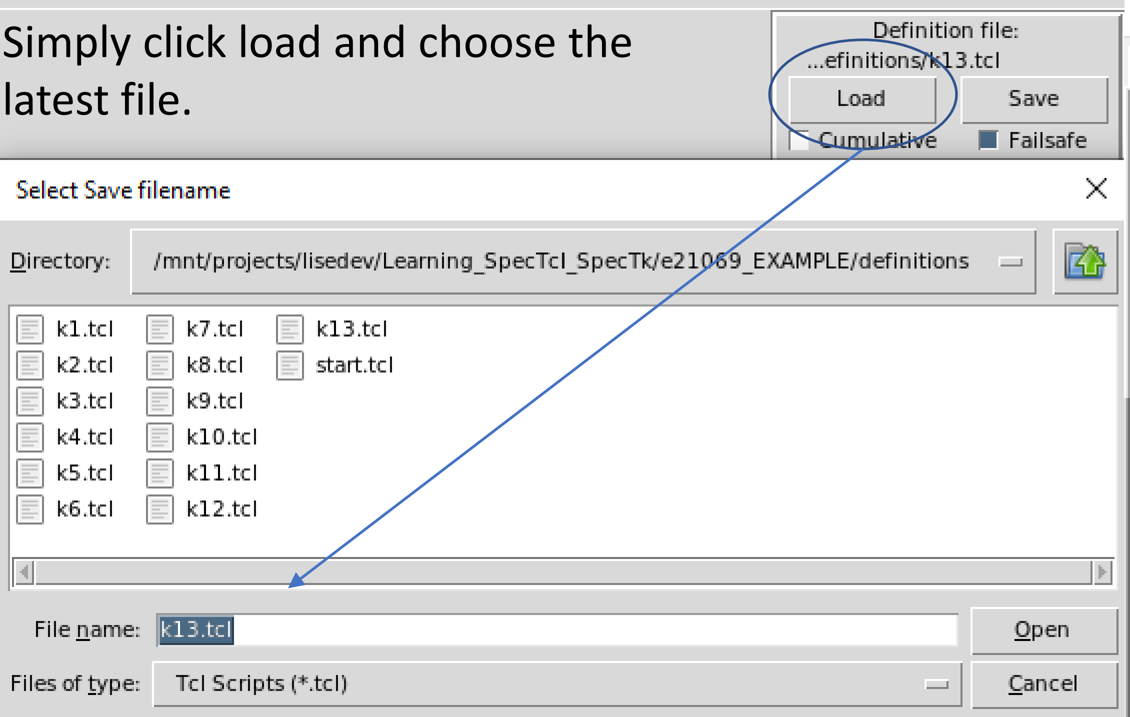
Most times we will attach to a ‘list’ of files, AKA .clu extension files

Data should begin to appear, if not, click **Update Page** in SpecTk window

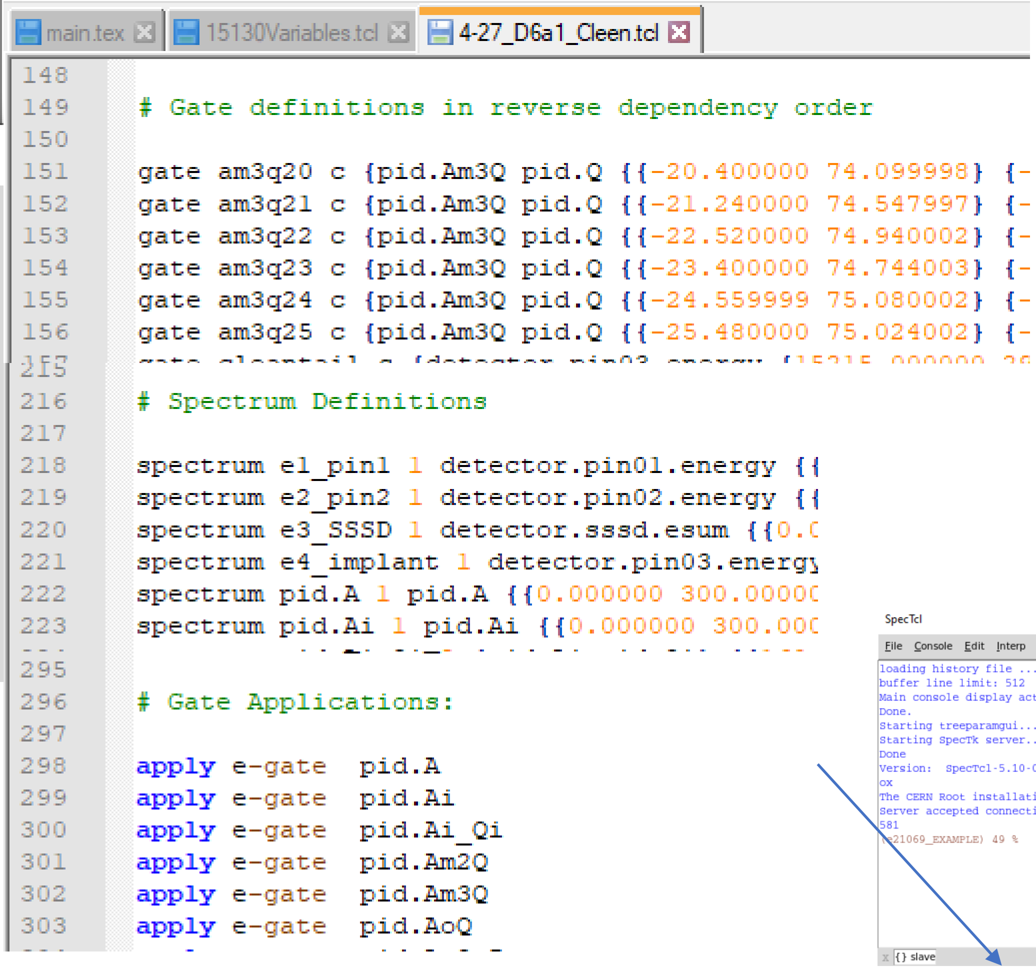


Loading a Definition (Save) File

Simply click load and choose the latest file.



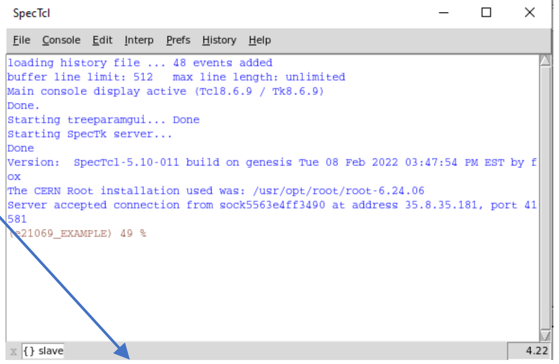
Keep in mind that definitions files are just a list of Tcl/Tk commands



```

# Gate definitions in reverse dependency order

gate am3q20 c {pid.Am3Q pid.Q {{-20.400000 74.099998} {-
gate am3q21 c {pid.Am3Q pid.Q {{-21.240000 74.547997} {-
gate am3q22 c {pid.Am3Q pid.Q {{-22.520000 74.940002} {-
gate am3q23 c {pid.Am3Q pid.Q {{-23.400000 74.744003} {-
gate am3q24 c {pid.Am3Q pid.Q {{-24.559999 75.080002} {-
gate am3q25 c {pid.Am3Q pid.Q {{-25.480000 75.024002} {-

# Spectrum Definitions

spectrum e1_pin1 1 detector.pin01.energy {{
spectrum e2_pin2 1 detector.pin02.energy {{
spectrum e3_SSSD 1 detector.sssd.esum {{0.0
spectrum e4_implant 1 detector.pin03.energy
spectrum pid.A 1 pid.A {{0.000000 300.00000
spectrum pid.Ai 1 pid.Ai {{0.000000 300.000

# Gate Applications:

apply e-gate  pid.A
apply e-gate  pid.Ai
apply e-gate  pid.Ai_Qi
apply e-gate  pid.Am2Q
apply e-gate  pid.Am3Q
apply e-gate  pid.AoQ
```

So

If You copy/paste this code into the SpecTcl command line window it would be the same as loading a definitions file.

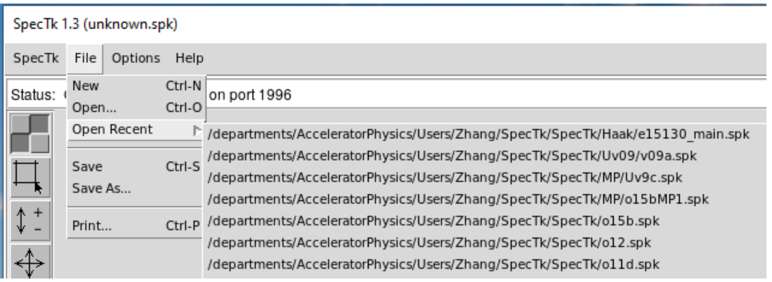


```
loading history file ... 48 events added
buffer line limit: 512   max line length: unlimited
Main console display active (Tcl8.6.9 / Tk8.6.9)
Done.
Starting treeparamgui... Done
Starting SpecTk server...
Done
Version:  SpecTcl-5.10-011 build on genesis Tue 08 Feb 2022 03:47:54 PM EST by f
ox
The CERN Root installation used was: /usr/opt/root/root-6.24.06
Server accepted connection from sock5563e4ff3490 at address 35.8.35.181, port 41
581
(e21069_EXAMPLE) 49 %
```

This can sometimes create issues when switching between definitions without restarting SpecTcl (a gate applied when you don't want it to be, etc.)

Loading a Spectra File

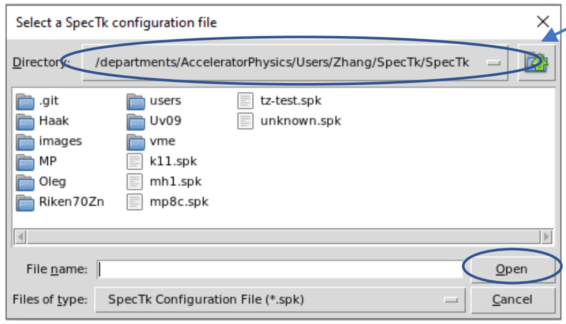
MICHIGAN STATE UNIVERSITY


SpecTk 1.3 (unknown.spk)

on port 1996

/departments/AcceleratorPhysics/Users/Zhang/SpecTk/SpecTk/Haak/e15130_main.spk
/departments/AcceleratorPhysics/Users/Zhang/SpecTk/SpecTk/Uv09/v09a.spk
/departments/AcceleratorPhysics/Users/Zhang/SpecTk/SpecTk/MP/Uv9c.spk
/departments/AcceleratorPhysics/Users/Zhang/SpecTk/SpecTk/MP/o15bMP1.spk
/departments/AcceleratorPhysics/Users/Zhang/SpecTk/SpecTk/o15b.spk
/departments/AcceleratorPhysics/Users/Zhang/SpecTk/SpecTk/o12.spk
/departments/AcceleratorPhysics/Users/Zhang/SpecTk/SpecTk/o11d.spk

The default path for loading .spk files is not a relative one to your experimental directory, thus continuously navigating to this can be repetitive and tedious

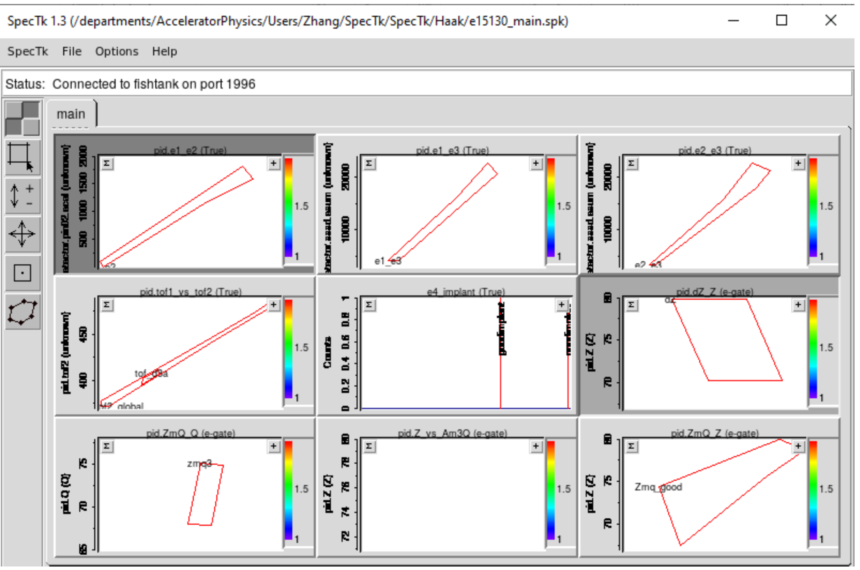

Select a SpecTk configuration file

Directory: /departments/AcceleratorPhysics/Users/Zhang/SpecTk/SpecTk

```
.git        users      tz-test.spk
Haak        Uv09       unknown.spk
images      vme
MP          k11.spk
Oleg        mh1.spk
Riken70Zn   mp8c.spk
```

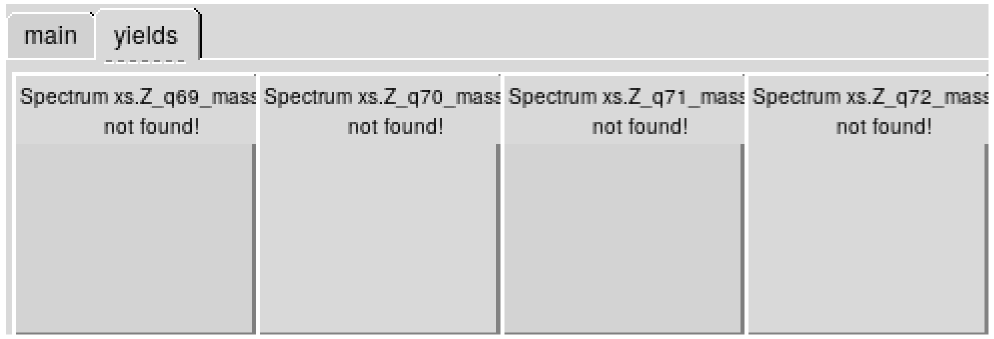
File name:

Files of type: SpecTk Configuration File (*.spk)


SpecTk 1.3 (/departments/AcceleratorPhysics/Users/Zhang/SpecTk/SpecTk/Haak/e15130_main.spk)

Status: Connected to fishtank on port 1996

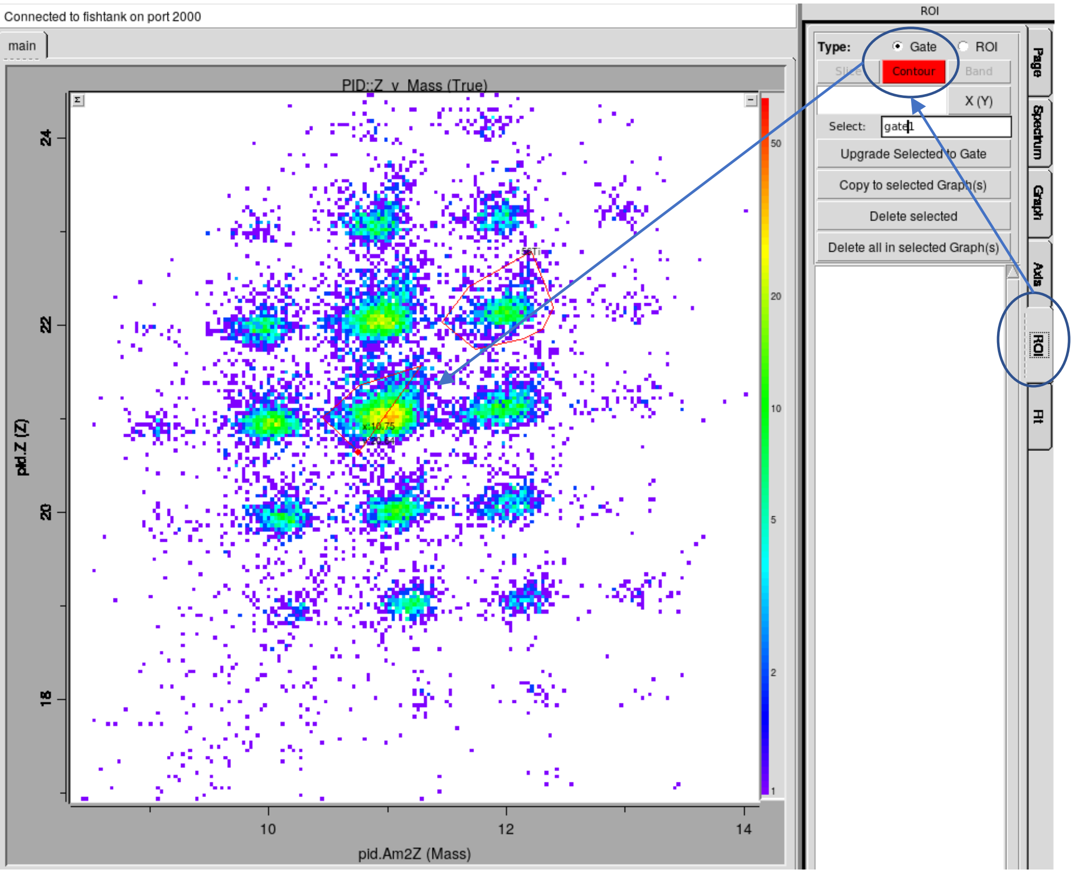
If you load a .spk file with spectra that haven't yet been defined in your SpecTcl you will see this:



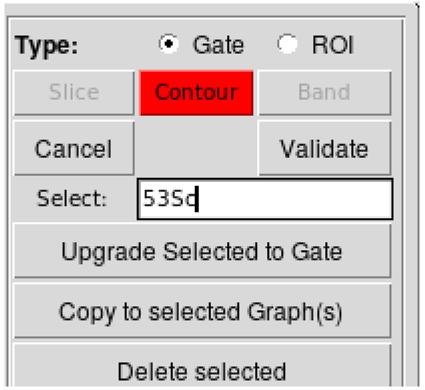
Spectrum xs.Z_q69_mass not found!
Spectrum xs.Z_q70_mass not found!
Spectrum xs.Z_q71_mass not found!
Spectrum xs.Z_q72_mass not found!

Level 2 – Collecting Data

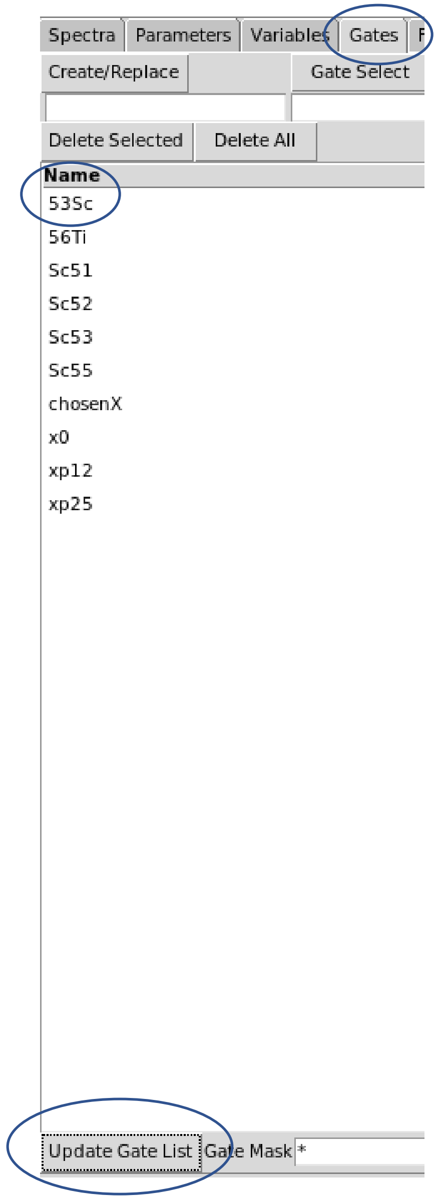
Making a Contour/Gate



Go to **ROI** tab, choose **Gate** and begin clicking on spectra to draw a contour. Double click on your last point to finish the contour.

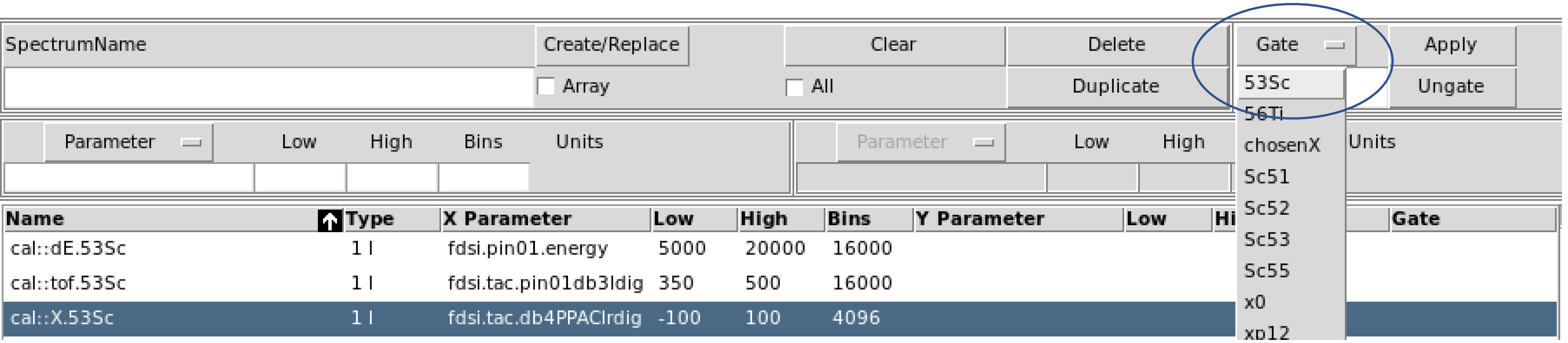


Assign a name to the contour and click validate.



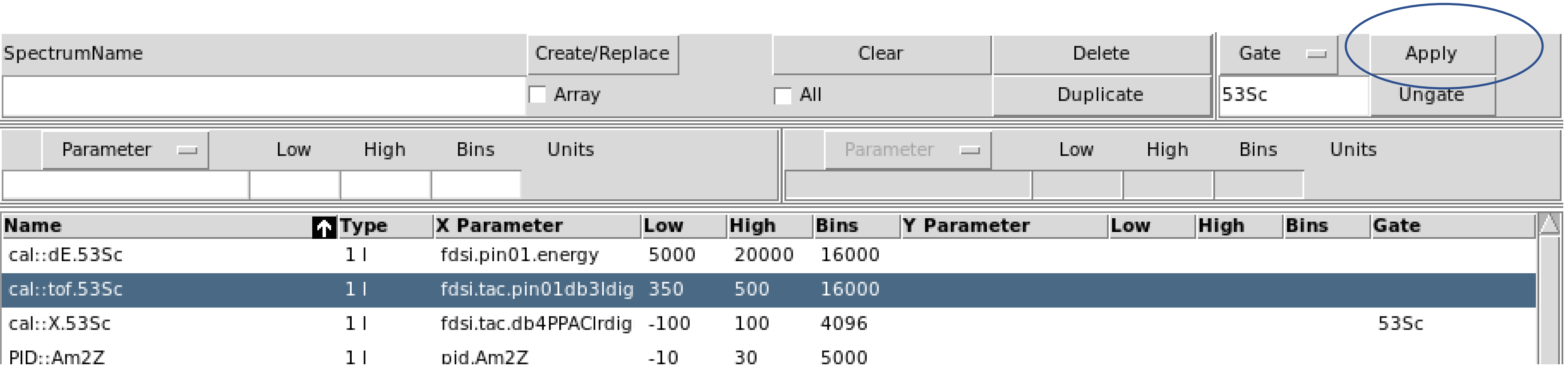
Now it should show up in the **Gates** tab in your main GUI window, after clicking **Update Gate List**

Applying a Gate

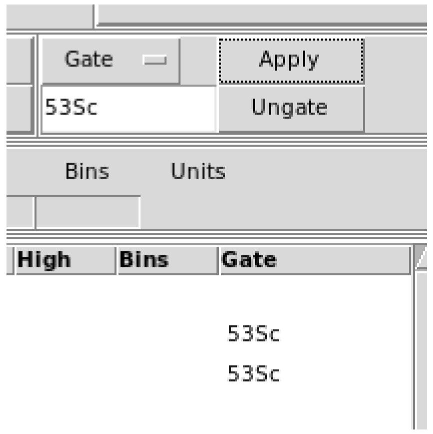


Name	Type	X Parameter	Low	High	Bins	Y Parameter	Low	Hi	Gate
cal::dE.53Sc	1 l	fdsi.pin01.energy	5000	20000	16000				
cal::tof.53Sc	1 l	fdsi.tac.pin01db3ldig	350	500	16000				
cal::X.53Sc	1 l	fdsi.tac.db4PPACIrdig	-100	100	4096				

Select a spectra then go to **Gate** and select the one you want to apply. It will automatically apply.



Name	Type	X Parameter	Low	High	Bins	Y Parameter	Low	High	Bins	Gate
cal::dE.53Sc	1 l	fdsi.pin01.energy	5000	20000	16000					
cal::tof.53Sc	1 l	fdsi.tac.pin01db3ldig	350	500	16000					
cal::X.53Sc	1 l	fdsi.tac.db4PPACIrdig	-100	100	4096					53Sc
PID::Am2Z	1 l	pid.Am2Z	-10	30	5000					



High	Bins	Gate
		53Sc
		53Sc

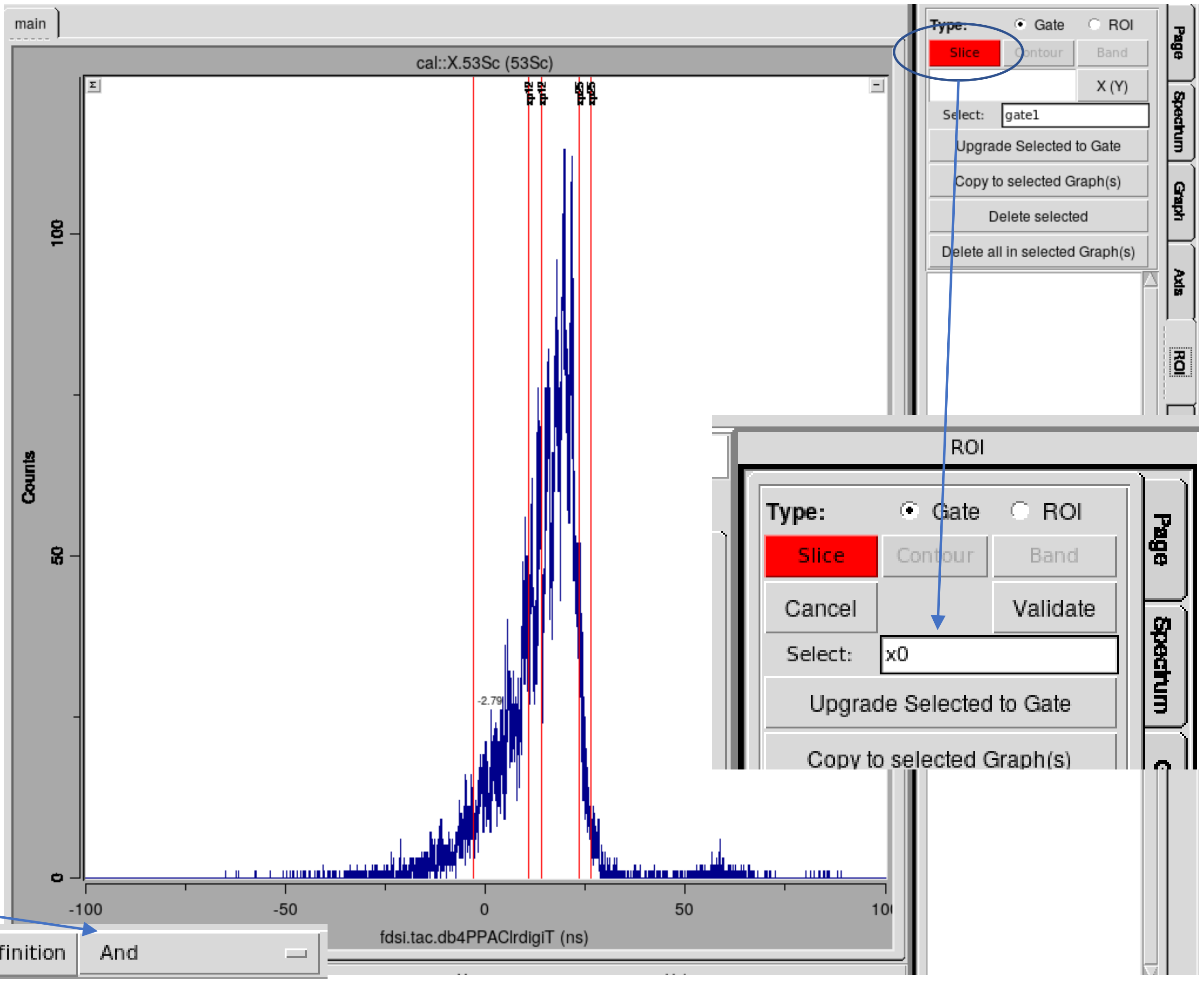
If you already have that gate selected, you can just select a spectra and click **Apply**

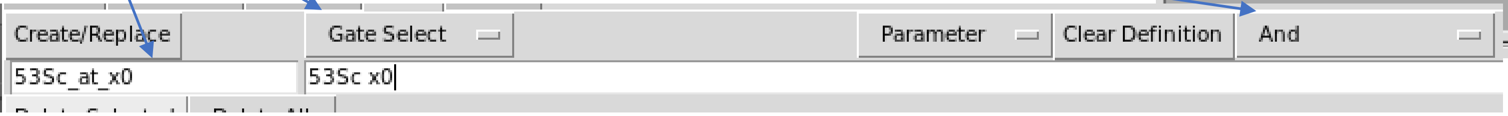
Combining Gates

Say you want energy loss and ToF measurements for calibration for a given fragment BUT at multiple positions. You can:

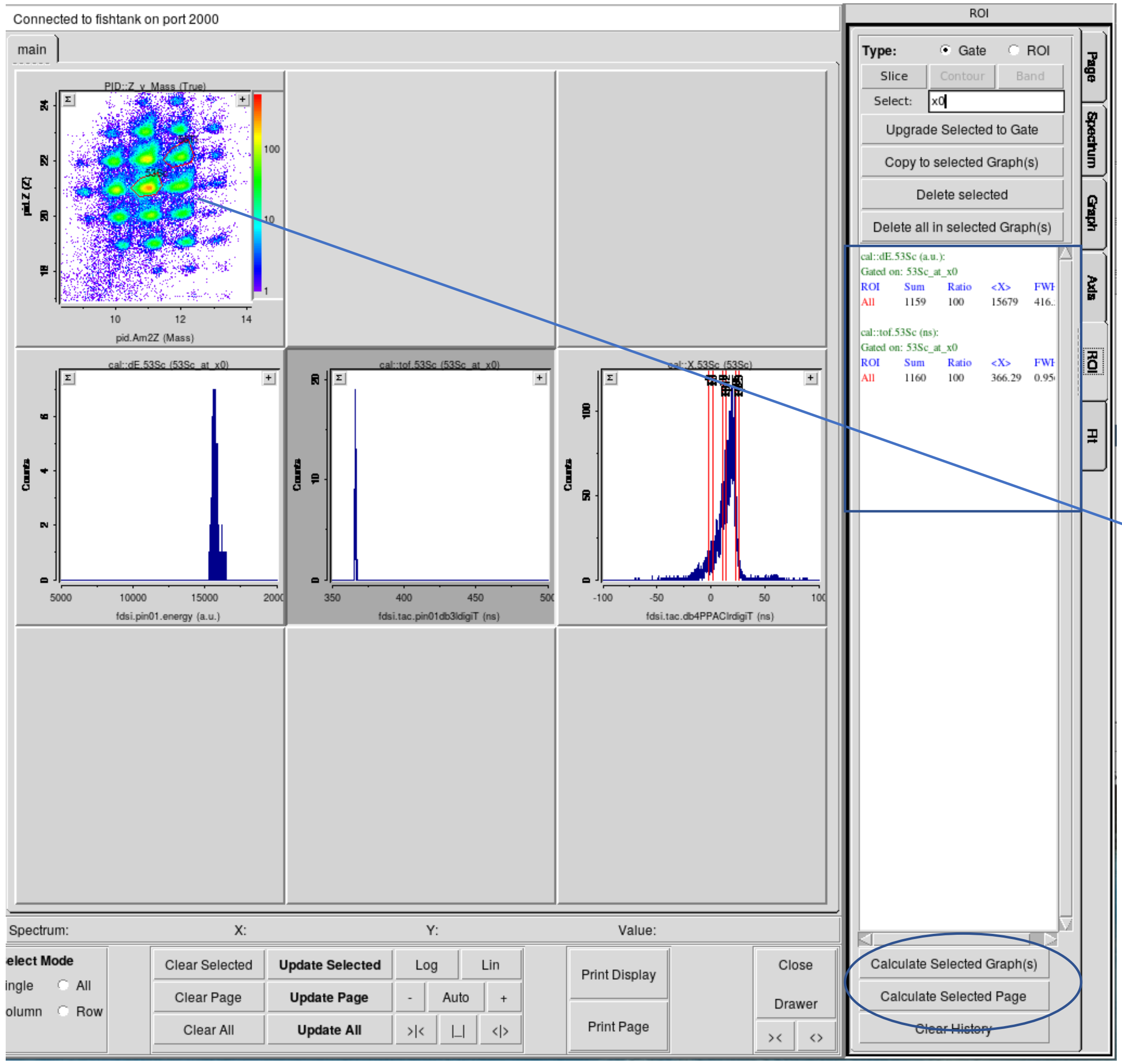
1) Contour the entire fragment
2) Apply that gate to the position spectra
3) Make slices (1D contour) on that spectra
4) Combine the contour with the slice to make an **AND** gate

1. Select gate type **And**
2. add the gates you are combining
3. give it a name and **Create**





Collecting Calibration Data



cal::dE.53Sc (a.u.):
Gated on: 53Sc_at_x0

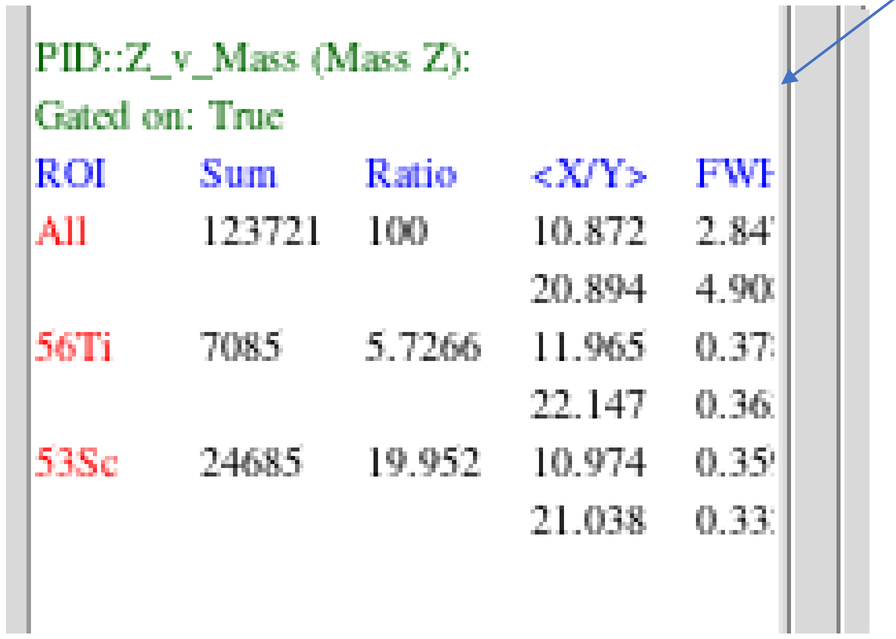
ROI	Sum	Ratio	<X>	FWH
All	1159	100	15679	416.

cal::tof.53Sc (ns):
Gated on: 53Sc_at_x0

ROI	Sum	Ratio	<X>	FWH
All	1160	100	366.29	0.95

In **ROI** tab there are **Calculate Selected ***** buttons. You can use these to get counts, mean spectra value, and distributions widths.

If there are gates on the spectra, these buttons will provide the same information for data within that area



PID::Z_v_Mass (Mass Z):
Gated on: True

ROI	Sum	Ratio	<X/Y>	FWH
All	123721	100	10.872	2.84
			20.894	4.90
56Ti	7085	5.7266	11.965	0.37
			22.147	0.36
53Sc	24685	19.952	10.974	0.35
			21.038	0.33

This is delimiter separated data which can be copy and pasted into excel for further analysis.

Level 3 – Calibrations

Applying Calibrations

There are *many* ways to apply calibration values to your data in SpecTcl.

Simple Methods
1. By hand in the GUI
2. By the command line
3. Upon loading a definitions (save) file

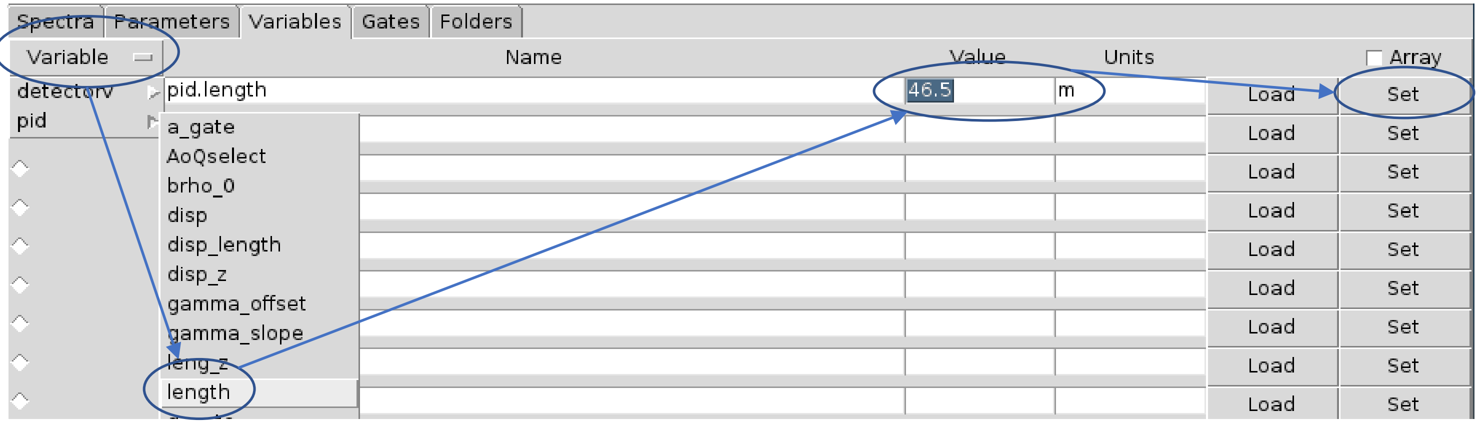
More Complex
4. Upon import of a data file
5. Upon start-up of the application

It all comes down to recognizing that setting calibrations is as simple as running a line of Tcl/Tk code.

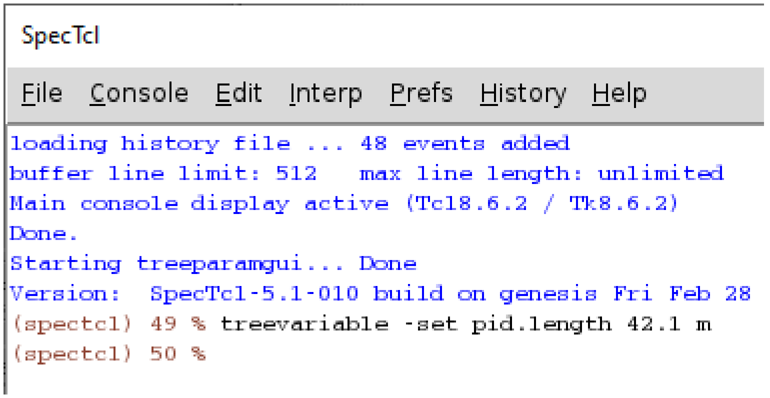
```
treevariable -set pid.length 46.7 m
```

Simple Methods

By hand in GUI

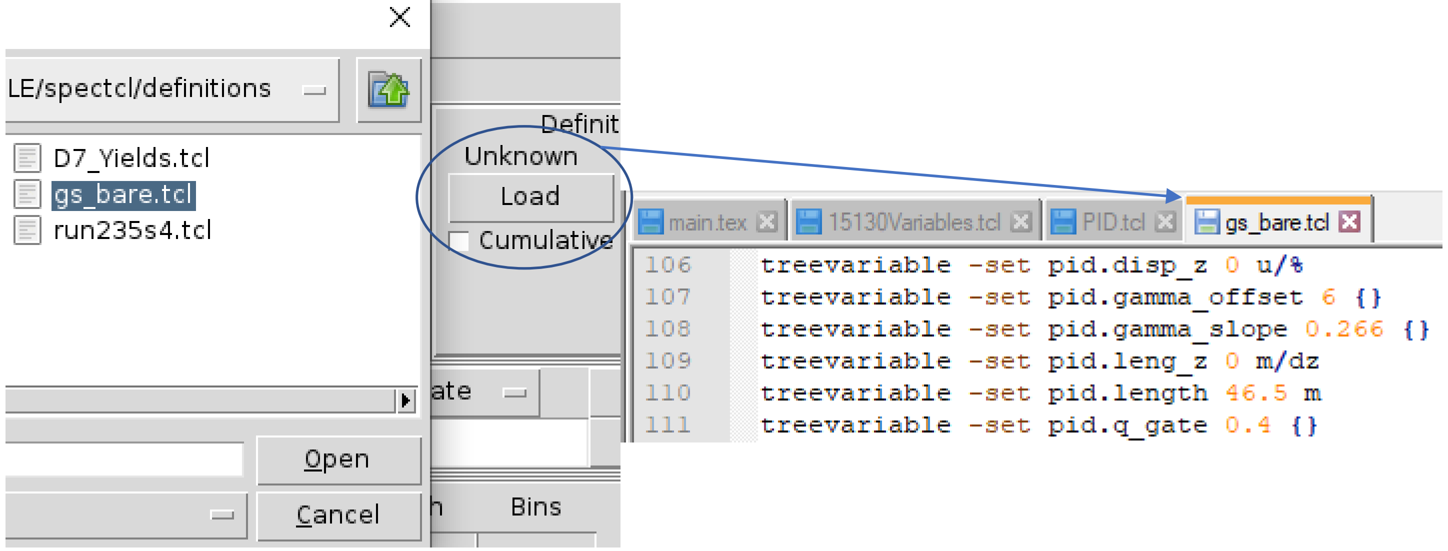


By command line



```
loading history file ... 48 events added
buffer line limit: 512   max line length: unlimited
Main console display active (Tcl8.6.2 / Tk8.6.2)
Done.
Starting treeparamgui... Done
Version:  SpecTcl-5.1-010 build on genesis Fri Feb 28
(spectcl) 49 % treevariable -set pid.length 42.1 m
(spectcl) 50 %
```

Importing definitions file



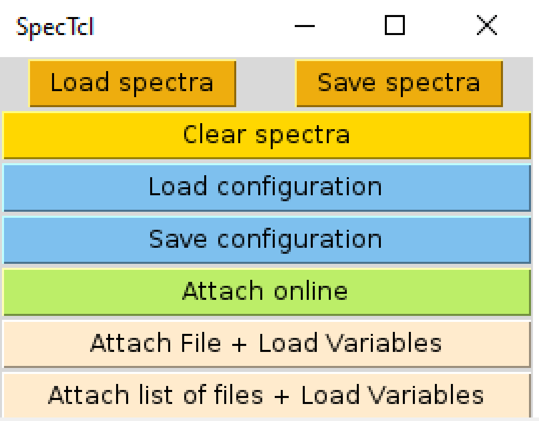
```
treevariable -set pid.disp_z 0 u/%
treevariable -set pid.gamma_offset 6 {}
treevariable -set pid.gamma_slope 0.266 {}
treevariable -set pid.leng_z 0 m/dz
treevariable -set pid.length 46.5 m
treevariable -set pid.q_gate 0.4 {}
```

By hand is good for a quick guess and check when changing a calibration variable (You can also click Load if you think its not set correctly, this will allow you to read the value)

Command line can be convenient because you can copy paste many commands in a row

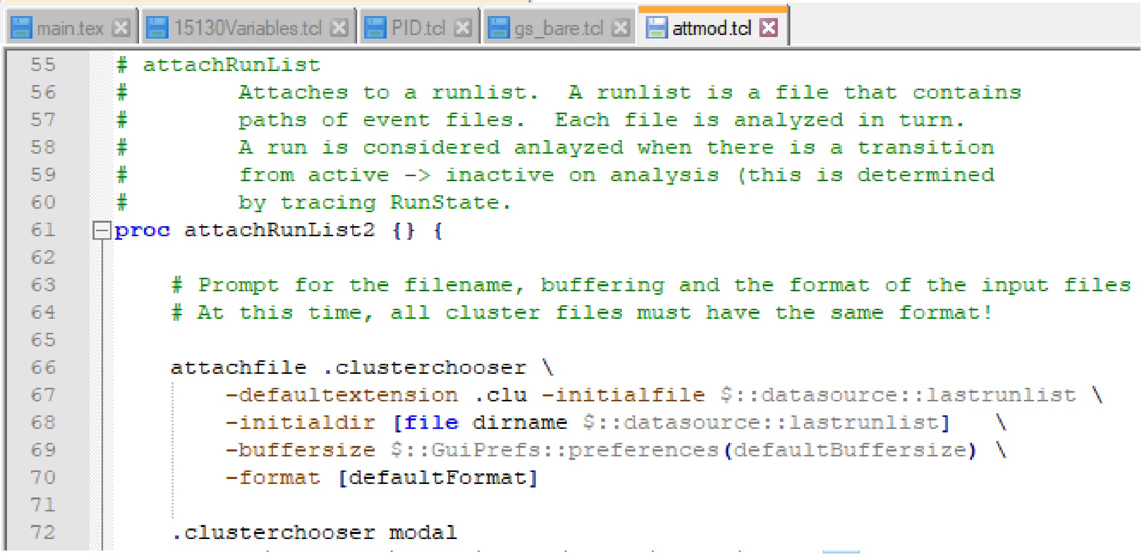
Definitions files can be easily modified by hand and reloaded (or copy/paste into command line)

More Complex Methods

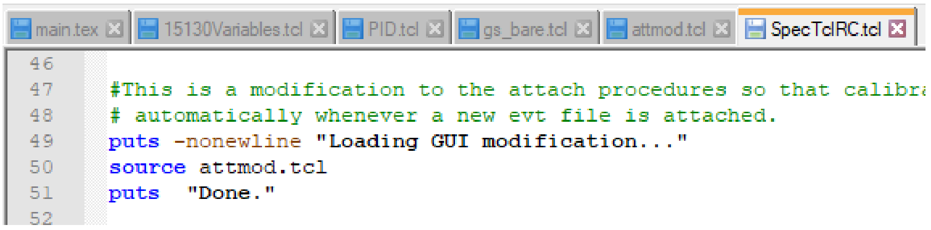


Upon loading a data file (e15130)

Include a GUI related option to load calibration variables specific to a given data set.

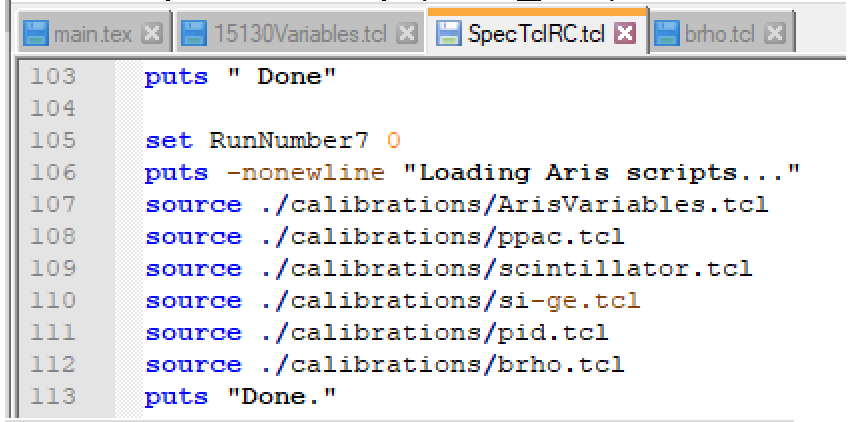


```
# attachRunList
#        Attaches to a runlist.  A runlist is a file that contains
#        paths of event files.  Each file is analyzed in turn.
#        A run is considered anlayzed when there is a transition
#        from active -> inactive on analysis (this is determined
#        by tracing RunState.
proc attachRunList2 {} {

    # Prompt for the filename, buffering and the format of the input files
    # At this time, all cluster files must have the same format!

    attachfile .clusterchooser \
        -defaultextension .clu -initialfile $::datasource::lastrunlist \
        -initialdir [file dirname $::datasource::lastrunlist]    \
        -buffersize $::GuiPrefs::preferences(defaultBuffersize) \
        -format [defaultFormat]

    .clusterchooser modal
```

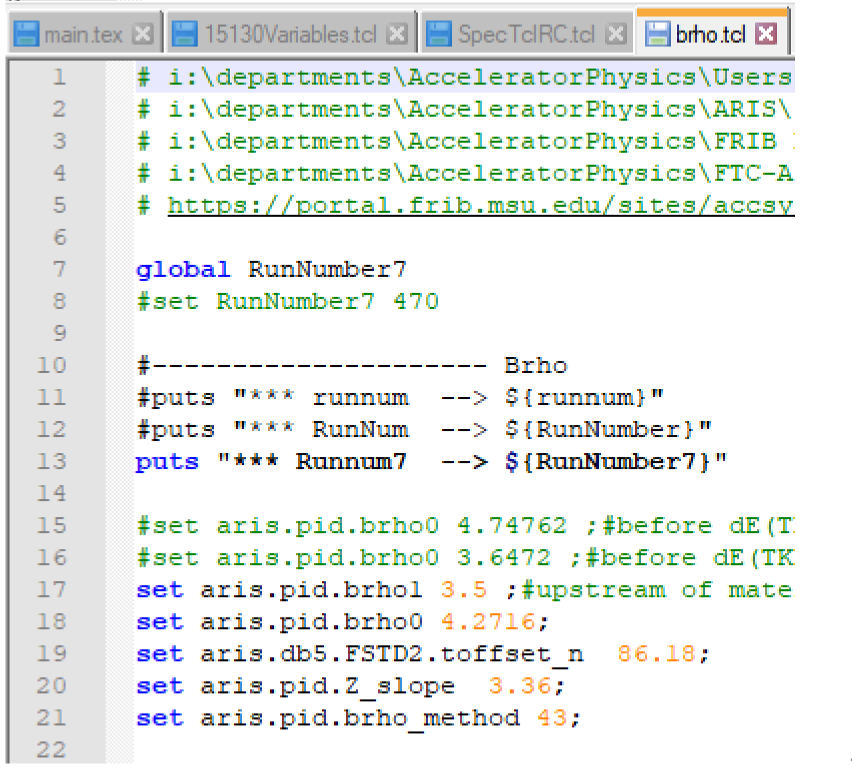


```

#This is a modification to the attach procedures so that calibr
# automatically whenever a new evt file is attached.
puts -nonewline "Loading GUI modification..."
source attmod.tcl
puts  "Done."

```

Upon start-up (ARIS_PID)



```
puts " Done"

set RunNumber7 0
puts -nonewline "Loading Aris scripts..."
source ./calibrations/ArisVariables.tcl
source ./calibrations/ppac.tcl
source ./calibrations/scintillator.tcl
source ./calibrations/si-ge.tcl
source ./calibrations/pid.tcl
source ./calibrations/brho.tcl
puts "Done."
```

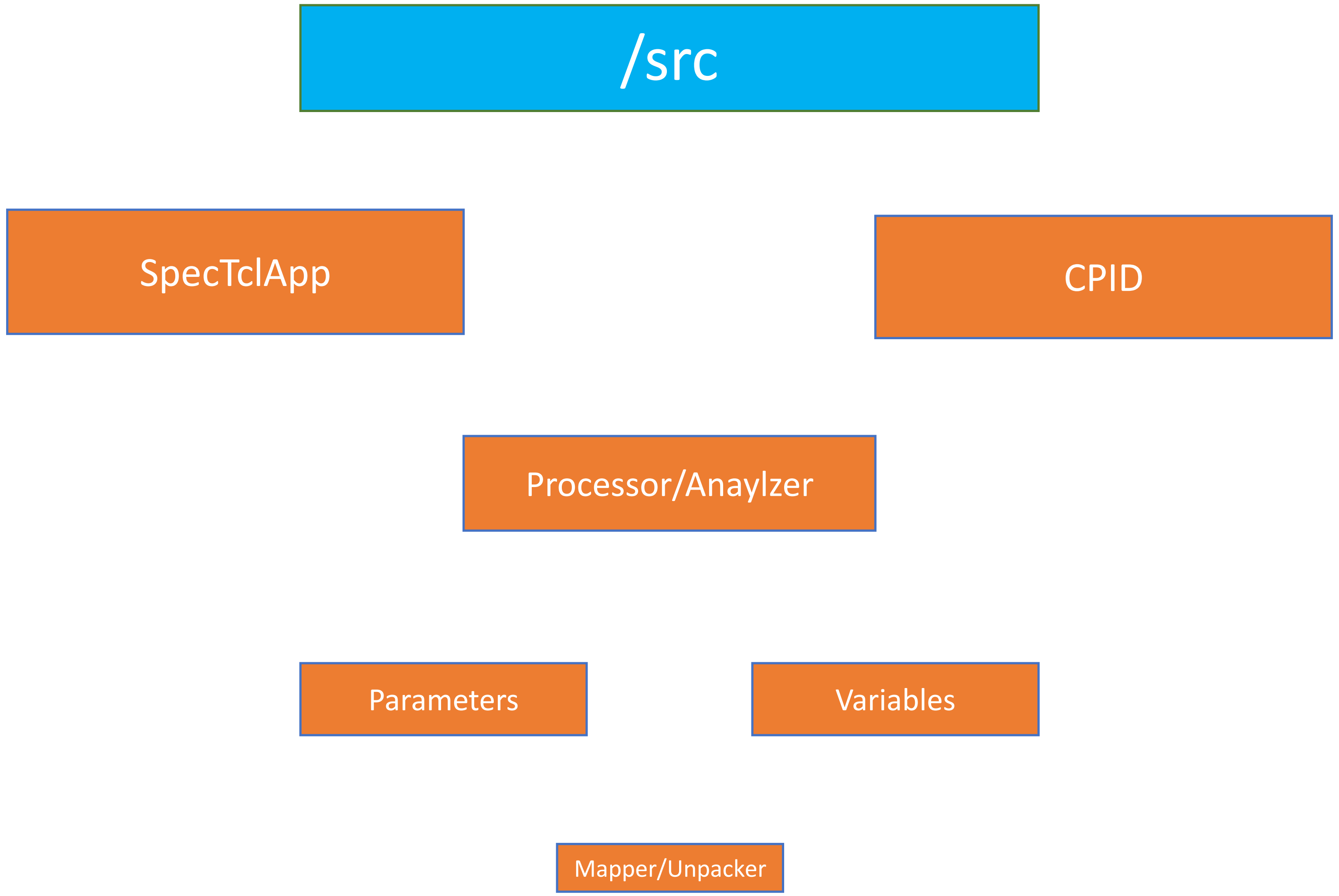


```
# i:\departments\AcceleratorPhysics\Users
# i:\departments\AcceleratorPhysics\ARIS\
# i:\departments\AcceleratorPhysics\FRIB
# i:\departments\AcceleratorPhysics\FTC-A
# https://portal.frib.msu.edu/sites/accsy

global RunNumber7
#set RunNumber7 470

#-------------------- Brho
#puts "*** runnum  --> ${runnum}"
#puts "*** RunNum  --> ${RunNumber}"
puts "*** Runnum7  --> ${RunNumber7}"

#set aris.pid.brho0 4.74762 ;#before dE(T
#set aris.pid.brho0 3.6472 ;#before dE(TK
set aris.pid.brho1 3.5 ;#upstream of mate
set aris.pid.brho0 4.2716;
set aris.db5.FSTD2.toffset_n  86.18;
set aris.pid.Z_slope  3.36;
set aris.pid.brho_method 43;

```

Create a set of definitions to be automatically loaded every time the application is started.

Level 4 – Modifying SpecTcl

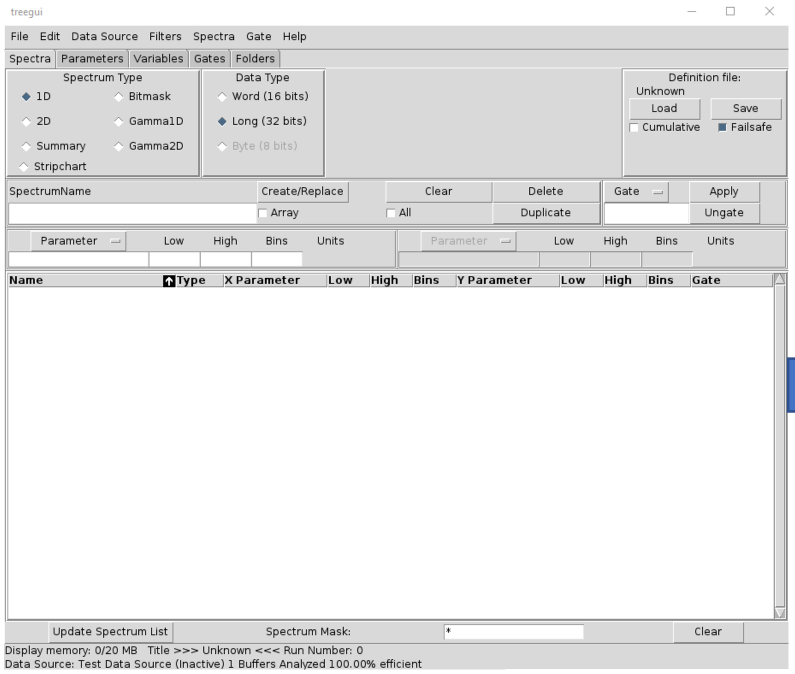
C++ Code Layout



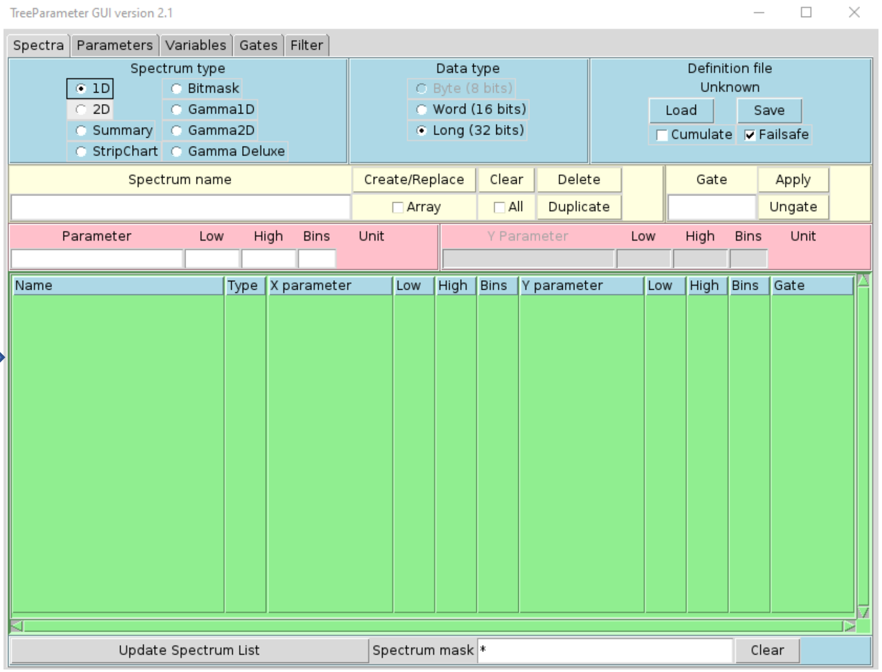
C++ Code Layout 2

- The main file which connects all the code together is ***SpecTclApp** (default is “MySpecTclApp”)
  - If you want to add C++ files to the project you add them here
- When doing PID you will usually only modify the “PID” src code file(s). This is often named **CPID**.
- There are many ways to organize the code. The main calculations for PID calibration may be done *inside* **CPID** or in another C++ file with names like **Analyzer** or **Processor**
- **Parameters** are values that can be plotted on spectra
  - Different for each event
- **Variables** are values which are used in calculations between parameters to make more parameters
  - Fixed for each event
  - ie. You can take ***parameters*** *like dE and ToF* and calibrate them with ***variables*** *like dEslope or ToF offset* to calculate the Z parameter
- The **Mapper/Unpacker** may also have a variety of names and its responsible for identifying which signal corresponding to which measurement

An Aside: GUI Modification

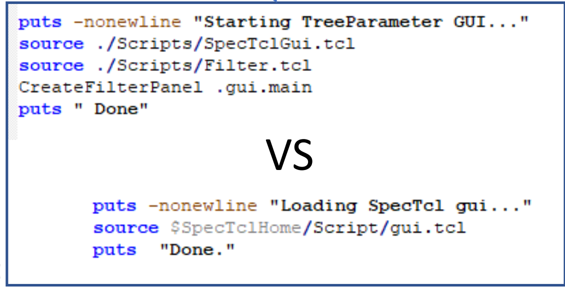

treegui
File Edit Data Source Filters Spectra Gate Help
Spectra Parameters Variables Gates Folders
Spectrum Type
1D Bitmask
2D Gamma1D
Summary Gamma2D
Stripchart
Data Type
Word (16 bits)
Long (32 bits)
Byte (8 bits)
Definition file:
Unknown
Load Save
Cumulative Failsafe
SpectrumName Create/Replace Clear Delete Gate Apply
Array All Duplicate Ungate
Parameter Low High Bins Units Parameter Low High Bins Units
Name Type X Parameter Low High Bins Y Parameter Low High Bins Gate
Update Spectrum List Spectrum Mask: * Clear

```
Display memory: 0/20 MB   Title >>> Unknown <<< Run Number: 0
Data Source: Test Data Source (Inactive) 1 Buffers Analyzed 100.00% efficient
```


TreeParameter GUI version 2.1
Spectra Parameters Variables Gates Filter
Spectrum type
1D Bitmask
2D Gamma1D
Summary Gamma2D
StripChart Gamma Deluxe
Data type
Byte (8 bits)
Word (16 bits)
Long (32 bits)
Definition file
Unknown
Load Save
Cumulate Failsafe
Spectrum name Create/Replace Clear Delete Gate Apply
Array All Duplicate Ungate
Parameter Low High Bins Unit Y Parameter Low High Bins Unit
Name Type X parameter Low High Bins Y parameter Low High Bins Gate
Update Spectrum List Spectrum mask * Clear

GUI modification to this extent is possible.

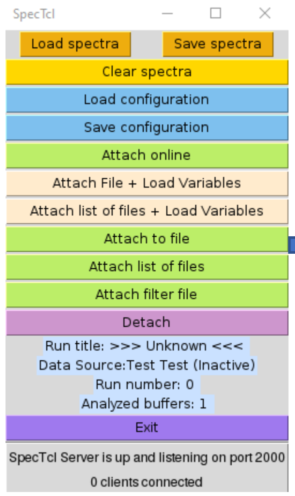
However, this requires redirecting the SpecTclRC to a personal copy of the SpecTcl GUI libraries.



```
puts -nonewline "Starting TreeParameter GUI..."
source ./Scripts/SpecTclGui.tcl
source ./Scripts/Filter.tcl
CreateFilterPanel .gui.main
puts " Done"
```

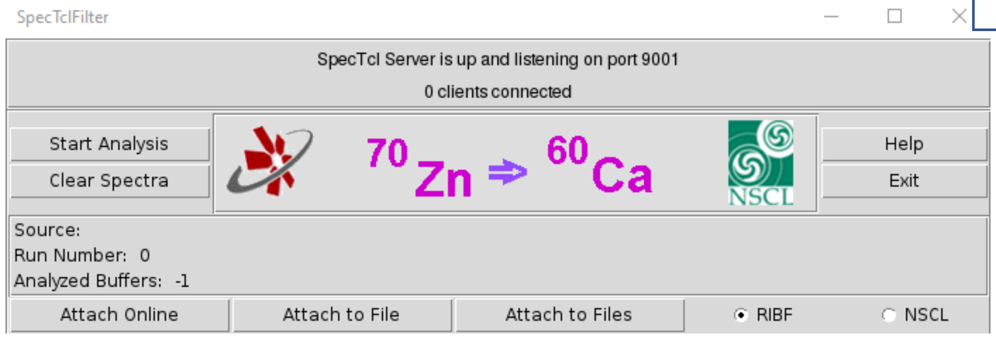
VS

```
puts -nonewline "Loading SpecTcl gui..."
source $SpecTclHome/Script/gui.tcl
puts  "Done."
```



```
Run title: >>> Unknown <<<
Data Source:Test Test (Inactive)
Run number: 0
Analyzed buffers: 1
```

```
SpecTcl Server is up and listening on port 2000
0 clients connected
```



```
SpecTcl Server is up and listening on port 9001
0 clients connected
Source:
Run Number:  0
Analyzed Buffers:  -1
```

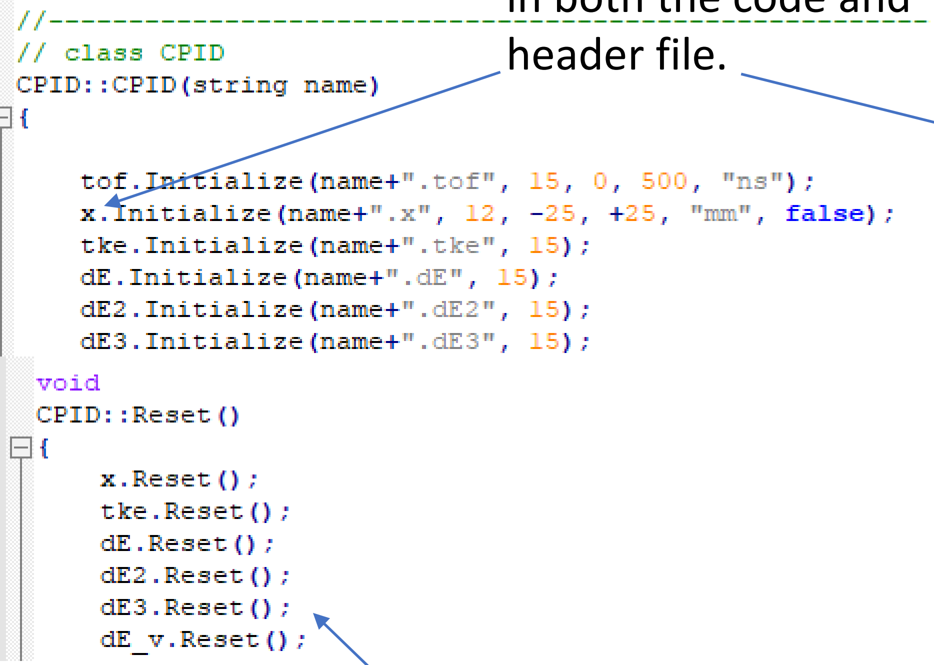
This is a very involved process. Use the Riken70Zn project as a reference if you want to do this.

Adding to PID Source 1

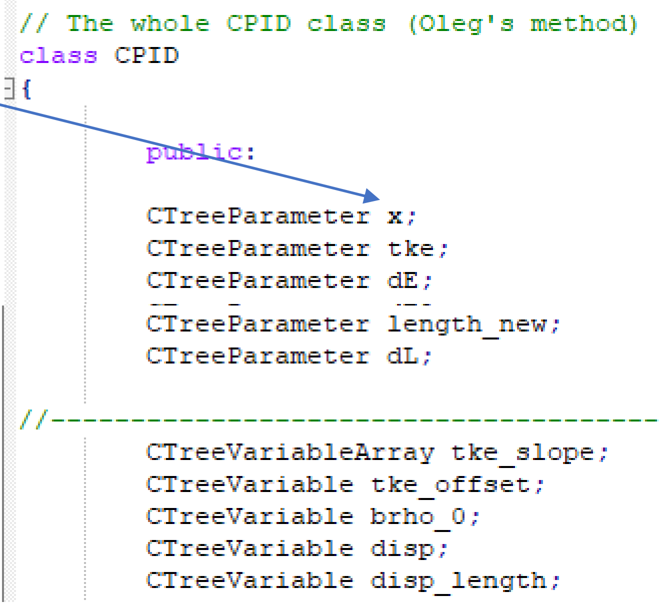
CPID.cpp

You must declare all variables/parameters in both the code and header file.

CPID.h



```
//-----------------------------------------------------------
// class CPID
CPID::CPID(string name)
{

    tof.Initialize(name+".tof", 15, 0, 500, "ns");
    x.Initialize(name+".x", 12, -25, +25, "mm", false);
    tke.Initialize(name+".tke", 15);
    dE.Initialize(name+".dE", 15);
    dE2.Initialize(name+".dE2", 15);
    dE3.Initialize(name+".dE3", 15);
  void
  CPID::Reset()
  {
      x.Reset();
      tke.Reset();
      dE.Reset();
      dE2.Reset();
      dE3.Reset();
      dE_v.Reset();
```



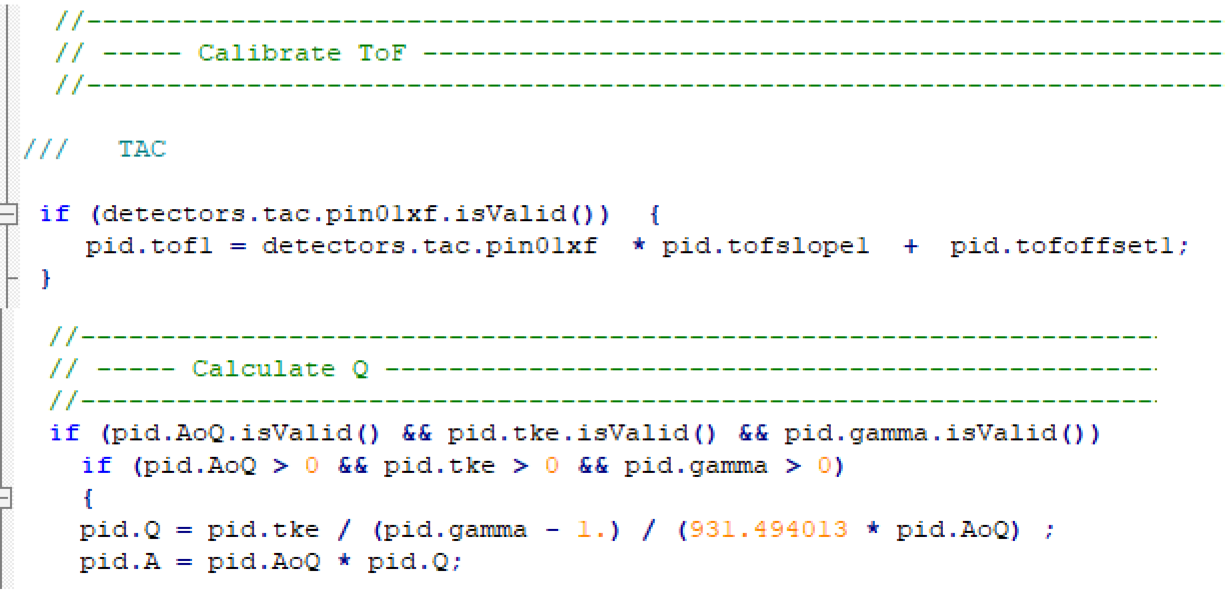
```
// The whole CPID class (Oleg's method)
class CPID
{

        public:

        CTreeParameter x;
        CTreeParameter tke;
        CTreeParameter dE;
        CTreeParameter length_new;
        CTreeParameter dL;

//---------------------------------------
        CTreeVariableArray tke_slope;
        CTreeVariable tke_offset;
        CTreeVariable brho_0;
        CTreeVariable disp;
        CTreeVariable disp_length;
```

Note the difference in declaration. There is a parameter object and a variable object.

Parameters must be reset below in the code file.
Variables are *not* reset.

Adding to PID Source 2

PIDProcessor.cpp



```
  //---------------------------------------------------------------------------
  // ----- Calibrate ToF ------------------------------------------------------
  //---------------------------------------------------------------------------

///   TAC

 if (detectors.tac.pin01xf.isValid())  {
    pid.tof1 = detectors.tac.pin01xf  * pid.tofslope1  +  pid.tofoffset1;
 }

  //------------------------------------------------------------------------
  // ----- Calculate Q ------------------------------------------------------
  //------------------------------------------------------------------------
  if (pid.AoQ.isValid() && pid.tke.isValid() && pid.gamma.isValid())
    if (pid.AoQ > 0 && pid.tke > 0 && pid.gamma > 0)
    {
    pid.Q = pid.tke / (pid.gamma - 1.) / (931.494013 * pid.AoQ) ;
    pid.A = pid.AoQ * pid.Q;
```

You may then manipulate these declarations in the processor file.

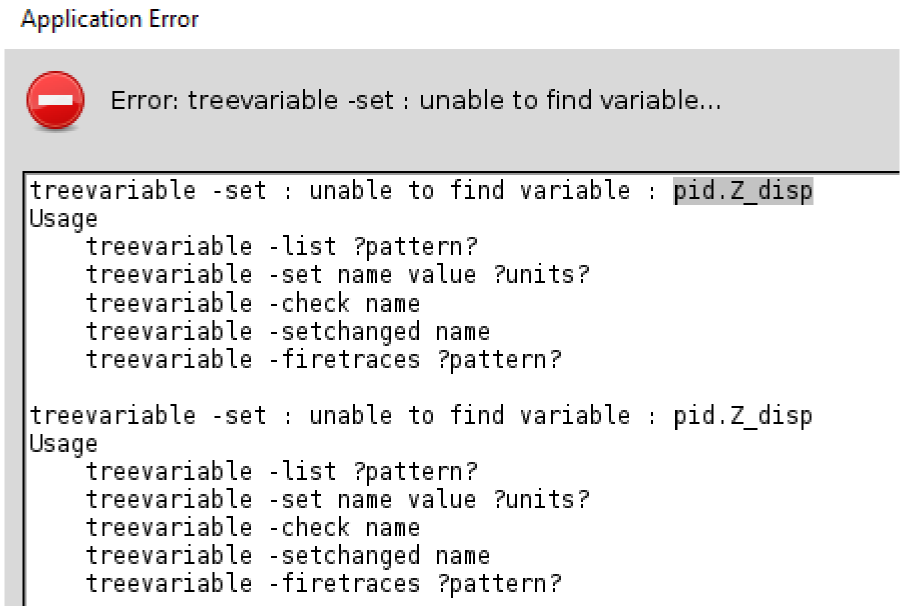
Yay for standard C++ coding!

Or depending on how you organize your code this can be in the same file as CPID (see Riken70Zn).

A Possible Issue

When you add/remove variables or parameters, there is the chance you will make old definitions files incompatible with your new build of SpecTcl.

You will see this error when trying to import/load a definitions (save) file.



Error: treevariable -set : unable to find variable...

```
treevariable -set : unable to find variable : pid.Z_disp
Usage
    treevariable -list ?pattern?
    treevariable -set name value ?units?
    treevariable -check name
    treevariable -setchanged name
    treevariable -firetraces ?pattern?

treevariable -set : unable to find variable : pid.Z_disp
Usage
    treevariable -list ?pattern?
    treevariable -set name value ?units?
    treevariable -check name
    treevariable -setchanged name
    treevariable -firetraces ?pattern?
```