



## Navigating the PCS Interface

#### Devon Battaglia, Keith Erickson, Dennis Mueller

#### NSTX-U Physics Operators Training September 15, 2015







### Starting the PCS Interface

- Become a member of PCS Unix group - Submit a ticket to helpdesk
- Log in to the real deal when sitting at the P.O. station
  - ssh -Y user@pcs-rt-3
  - ssh -Y user@pcs-srv-3 (preferred once it is ready)
  - Navigate to folder /opt/pcs/INTEL\_DIR/
    - ./run\_wave
- Or run stand-alone PCS anywhere
  - Set up PCS for future shots or run simulations
  - Keith or Roman can help you get this code in your area
  - ssh -Y user@pcspool
  - Navigate to folder /trunk/INTEL\_DIR/
    - ./runsa

## You get the purple box

- File – Version
  - Quit
- Control
  - -Next shot
  - Future shot
  - Shared shot
- Utilities
  - -View PCS log



#### PCS log provides useful feedback on status and communication with realtime system

- File – Quit
- Utilities
  - Show pcs log messages
  - Hide pcs log messages
    - This is the default
  - -PCS process table
    - Shows what computers are up
  - Show realtime errors only
- I like to keep this window open on the desktop

PCS State : Time Stamp: Status: Data: Shot Cycle: Pending Next Shot 0 100 15:17:15-1-(1k) lockserver ready, PID = 20162 15:17:15-1+(1k) lockserver ready, PID = 20162 15:17:11+(1k) lockserver ready, PID = 2019 15:17:10+(1k) lockserver ready, PID = 20234, 15:17:10+(1k) lockserver ready, PID = 20234, 15:17:10+(1k) lockserver ready, PID = 20234, 15:17:10+(1k) lockserver reads, PID = 20234, 15:17:10+(1k)			🔀 Vi	ew Log		
Time Stamp:       Status:       Data:	File Utilities					
Time Stamp:       Status:       Data:	PCS State :	T				
Shot Cycle: Pending Next Shot	1 00 0 tale .					
<pre>0</pre>	Time Stamp:	Ĭ		Status:	Data:	
<pre>15:17:15-1-(1k) lockserver ready PID = 20152 15:17:15-1+(w) waveserver starting queue handler. 15:17:11-1+(w) Avdbattagi/ccs/first_try/truck/INTEL_DIE/runsa_dir/waveserver_dir/NextShot.wal0: load all ( 15:17:11-1+(w) /w/dbattagi/ccs/first_try/truck/INTEL_DIE/runsa_dir/waveserver_dir/NextShot.wal0: load all ( 15:17:11-1+(w) /w/dbattagi/ccs/first_try/truck/INTEL_DIE/runsa_dir/waveserver_dir/NextShot.wal0: load all ( 15:17:11-1+(w) waveserver ready PID = 20139 15:17:11-1+(w) waveserver port_nuber = 2222 15:17:11-1+(w) gu 3: client_tpg= 0:7, menry = 134217728, host = sunfire47 15:17:11-1+(b) RNNODE STANDALINE 15:17:10-1+(b) RNNODE STANDALINE 15:17:10-1+(b) RNNODE STANDALINE 15:17:10-1+(c) FINISEED INTILL_IZINTORS.FI=0x/7fa85644000 to 0:7fa8564afffc, HOST=0x.7f989f0b0000 to 0:7f8t 15:17:10-1+(c) gu 3: client_tpg= 0:7, menry = 134217728, host = sunfire47 15:17:10-1+(c) GPU 3: host_realtine process started. PID = 20291. 15:17:10-1+(c) GPU 2: host_realtine process started. PID = 20281. 15:17:10-1+(c) GPU 2: hostsreer on fixed port = 2333 15:17:10-1+(c) Setting locksrever NBE/DMRE interaction is OFF *** 15:17:10-1+(c) Setting locksrever NBE/DMRE interaction is OFF</pre>			Shot Cycle: Pe	nding Next Sh	ot	
<pre>15:17:15:H-(1k) lockserver rtcp: count = 3 15:17:11-H-(w) waveserver: starting queue handler. 15:17:11-H-(w) Awyeserver: starting queue handler. 15:17:11-H-(w) /w/dbattsgl/cs/first.try/truk/INTEL_DIR/runsa_dir/waveserver_dir/NextShot.wa10: load all c 15:17:11-H-(w) /w/dbattsgl/cs/first.try/truk/INTEL_DIR/runsa_dir/waveserver_dir/NextShot.wa10: load all c 15:17:11-H-(w) waveserver: roady, PID = 20158 15:17:11-H-(w) cpu 3: client.type = 0x7, memory = 134217728, host = sunfire47 15:17:10-H-(k) RNMODE STANDLOWE 15:17:10-H-(k) FINISHED INITIALIZATIONS: RT=0x7Fa496Fab000 to 0x7Fa496Faaffc, HOST=0x7Fa31999c000 to 0x7fac 15:17:00-H-(k) FINISHED INITIALIZATIONS: RT=0x7Fa496Fab000 to 0x7Fa24508d+fc, HOST=0x7Fa31999c000 to 0x7fac 15:17:00-H-(k) FINISHED INITIALIZATIONS: RT=0x7Fa496Fab000 to 0x7Fa24508d+fc, HOST=0x7Fa31999c000 to 0x7fac 15:17:00-H-(k) RNMODE STANDLOWE 15:17:00-H-(k) STANDLOWE 15:</pre>	0					100
<pre>15:17:15:H-(1k) lockserver rtcp: count = 3 15:17:11-H-(w) waveserver: starting queue handler. 15:17:11-H-(w) Awyeserver: starting queue handler. 15:17:11-H-(w) /w/dbattsgl/cs/first.try/truk/INTEL_DIR/runsa_dir/waveserver_dir/NextShot.wa10: load all c 15:17:11-H-(w) /w/dbattsgl/cs/first.try/truk/INTEL_DIR/runsa_dir/waveserver_dir/NextShot.wa10: load all c 15:17:11-H-(w) waveserver: roady, PID = 20158 15:17:11-H-(w) cpu 3: client.type = 0x7, memory = 134217728, host = sunfire47 15:17:10-H-(k) RNMODE STANDLOWE 15:17:10-H-(k) FINISHED INITIALIZATIONS: RT=0x7Fa496Fab000 to 0x7Fa496Faaffc, HOST=0x7Fa31999c000 to 0x7fac 15:17:00-H-(k) FINISHED INITIALIZATIONS: RT=0x7Fa496Fab000 to 0x7Fa24508d+fc, HOST=0x7Fa31999c000 to 0x7fac 15:17:00-H-(k) FINISHED INITIALIZATIONS: RT=0x7Fa496Fab000 to 0x7Fa24508d+fc, HOST=0x7Fa31999c000 to 0x7fac 15:17:00-H-(k) RNMODE STANDLOWE 15:17:00-H-(k) STANDLOWE 15:</pre>						
15:17:11-H-(w)       waveserver starting queue handler.         15:17:11-H-(w)       Rew problem list is empty.         15:17:11-W(w)       /u/dbattagl/pcs/first_try/trunk/INTEL_DIR/runsa_dir/waveserver_dir/NextShot.wa10: load all c         15:17:11-W(w)       /u/dbattagl/pcs/first_try/trunk/INTEL_DIR/runsa_dir/waveserver_dir/NextShot.wa10: load all c         15:17:11-W(w)       waveserver port number = 22222         15:17:11-W(w)       waveserver port number = 22222         15:17:11-W(w)       queue handler = 20284.         15:17:10-W(w)       queue handler = 120284.         15:17:10-W(w)       loss frammLowe         15:17:10-W(w)       public frammLowe         15:17:10-W(w)						
<pre>15:17:11() Raw problem list is empty. 15:17:11-+() /u/dbattsgl/pcs/first_try/trurk/INTEL_DIR/runsa_dir/waveserver_dir/NextShot.wal0: load all c 15:17:11() /u/dbattsgl/pcs/first_try/trurk/INTEL_DIR/runsa_dir/waveserver_dir/NextShot.wal0: load all c 15:17:10() /u/dbattsgl/pcs/first_try/trurk/INTEL_DIR/runsa_dir/waveserver_dir/Pip/Pip/Pip/Pip/Pip/Pip/Pip/Pip/Pip/Pip</pre>						
<pre>15:17:11-+(uv) /uv/dbattag1/pcs/first.try/trunt/INTEL_DIR/runsa_dir/waveserver_dir/NextShot.wa10: load all c 15:17:11-+(uv) waveserver port number = 2222 15:17:11-+(uv) cavaserver and the process started. PID = 20294. 15:17:10-+(uv) lockserver waiting for waveserver to get ready 15:17:10-+(uv) cpu 2: client_type = 0x7, memory = 13421728, host = sunfire47 15:17:10-+(uv) cpu 2: client_type = 0x7, memory = 13421728, host = sunfire47 15:17:10-+(uv) cpu 2: client_type = 0x7, memory = 13421728, host = sunfire47 15:17:10-+(uv) cpu 2: client_type = 0x7, memory = 13421728, host = sunfire47 15:17:10-+(uv) cpu 2: client_type = 0x7, memory = 13421728, host = sunfire47 15:17:10-+(uv) cpu 2: client_type = 0x7, memory = 13421728, host = sunfire47 15:17:00-+(uv) cpu 2: client_type = 0x7, memory = 13421728, host = sunfire47 15:17:00-+(uv) cpu 2: client_type = 0x7, memory = 13421728, host = sunfire47 15:17:00-+(uv) cpu 2: client_type = 0x7, memory = 13421728, host = sunfire47 15:17:00-+(uv) cpu 2: client_type = 0x7, memory = 536870912, host = sunfire47 15:17:00-+(uk) RNNODE STANDLONE 15:17:00-+(uk) Not_realtime process started. PID = 20288. 15:17:00-+(uk) StartING UP WAVESERVER PROCESS  File Utilities  PCS State : [ Time Stamp:</pre>	15:17:11-P-(wv)	Raw problem list i	is empty.			
15:17:11-1-(uv)       waveserver ready, PID = 20158         15:17:11-1+(uv)       waveserver port number = 22222         15:17:11-11+(uv)       waveserver port number = 22222         15:17:11-11+(uv)       runveserver port number = 22222         15:17:10-11+(uv)       runveserver port number = 22222         15:17:10-11+(uv)       runveserver port number = 22224         15:17:10-11+(uv)       runveserver to get ready         15:17:10-11+(uv)       runveserver to get ready         15:17:10-11+(uv)       runveserver to get ready         15:17:10-11+(uv)       runver not fixed port = 23333						
15:17:11-H-(n3)       FINISHED INITIALIZATIONS: RT=0x7F368444000 to 0x7F38844dfFfc. H0ST=0x7F40d302000 to 0x7F4         15:17:11-H-(n3)       GPU 3; client_type = 0x7, memory = 134217728, host = sunfire47         15:17:11-H-(n3)       GPU 3; host_realtime process started. PID = 20294,         15:17:11-H-(n4)       lockserver waiting for waveserver to get ready         15:17:10-H-(n2)       RUNNOE STANDLIONE         15:17:10-H-(n2)       FINISHED INITIALIZATIONS; RT=0x7Fa4387ab000 to 0x7Fa4b8FaaFfc, H0ST=0x7F3b9F0b0000 to 0x7F3t         15:17:10-H-(n2)       RUNNOE STANDLIONE         15:17:10-H-(n2)       CPU 2; host_realtime process started. PID = 20291.         15:17:10-H-(n4)       Initint_2ATIONS; RT=0x7Hal508000 to 0x7Hal508000 to 0x7Hal5080000 to 0x7Hal508000 to 0x7Hal508000 to 0x7H	15:17:11-I-(wv)	waveserver ready	. PID = 20158	birer anod_air / wa		iotearot tong att c
15:17:11-H-(w)       opu 3: client.type = 0x7, memory = 134217728, host = surfire47         15:17:11-H-(h3)       RUNNODE STANDALONE         15:17:11-H-(h3)       RUNNODE STANDALONE         15:17:11-H-(h3)       RUNNODE STANDALONE         15:17:11-H-(h2)       RUNNODE STANDALONE         15:17:11-H-(h2)       RUNNODE STANDALONE         15:17:10-H-(h2)       RUN Statististististististististististististist				-0000 to 0v7£7£9		0d702000 to 0v7for
15:17:11-1-(h3)       CPU 3; host_realtime process started, PID = 20294.         15:17:10-H(1k)       lockserver waiting for waveserver to get ready         15:17:10-H(2)       RINNIDE STANDLONE: RT=0.7744967ab000 to 0x77a4b67a67545670b0000 to 0x7791         15:17:10-H(2)       RUNNOE STANDLONE: RT=0.7744967ab000 to 0x77a4b67a67519390000 to 0x7791         15:17:10-H(2)       RUNNOE STANDLONE: RT=0.7744067a6000 to 0x77a4b67a519390000 to 0x77631939000 to 0x763         15:17:10-H(2)       RUNNOE STANDLONE         15:17:10-H(2)       RUN of the ready         15:17:10-H(1)       FINISHED INITIALIZATIONS: RT=0.77d10508000 to 0x77631939000 to 0x7631939000 to 0x763193193000 to 0x76319319191011111111111111111111111111111		cpu 3: client_type	e = 0x7, memory = 134			00302000 08 027184
15:17:10-H-(lk)       lockserver waiting for waveserver to get ready         15:17:10-H-(k)       FINISHED INITIALIZATIONS: RT=0x7fa498fab000 to 0x7fa4b8faaffc, H0ST=0x7f9b9f0b0000 to 0x7f9t         15:17:10-H-(w)       ppu 2: client_type = 0x7, memory = 13421728, host = sunfire47         15:17:10-H-(w)       ppu 2: client_type = 0x7, memory = 13421728, host = sunfire47         15:17:10-H-(w)       ppu 2: client_type = 0x7, memory = 13421728, host = sunfire47         15:17:10-H-(w)       ppu 2: client_type = 0x7, memory = 538870912, host = sunfire47         15:17:10-H-(N)       RNNDE STANDLOW         15:17:00-H-(h)       RNNTALES TANDLOW         15:17:00-H-(h)       RNTALES TANDLOW         15:17:00-H-(k)       Setting lockserver no fixed port = 23333         15:17:00-H-(k)       Setting lockserver PROCESS         Is:17:00-H-(k)       Setting lockserver PROCESS         Is:17:00-H-(k)       Setting lockserver PROCESS         Is:17:00-H-(k)       Setting lockserver PROCESS         Is:17:00-W       Immatch case         Find:       Immatch case				DID 00004		
15:17:10-H-(L2)       FINISHED INITIALIZATIONS: RT=0x7Fa4398 <sup>2</sup> ab000 to 0x7F4b89fa4ffc, H0ST=0x7F9b970b0000 to 0x7F91         15:17:10-H-(L2)       RUNNDE STANDALONE         15:17:10-H-(L2)       RUNNDE STANDALONE         15:17:10-H-(L2)       RUNNDE STANDALONE         15:17:10-H-(L2)       RUNNDE STANDALONE         15:17:10-H-(L2)       PU 2: host_realtime process started. PII = 20291.         15:17:08-H-(L1)       FINISHED INITIALIZATIONS: RT=0x7fdic508000 to 0x7fd24508dffc, H0ST=0x7fc31999c000 to 0x7fc2         15:17:08-H-(LN)       realtime process started. PID = 20281.         15:17:08-H-(LN)       lockserver hog Hcady         15:17:08-H-(LN)       lockserver hog Hcady         15:17:08-H-(LN)       lockserver hog maxeserver to get ready         15:17:08-H-(LK)       lockserver H62IMBR et interaction is 0FF ***         15:17:08-H-(LK)       ****         15:17:08-H-(LK)       status:         15:17:08-H-(LK)       ****         15:17:08-H-(LK)       status:         15:17:08-H-(LK)       ****         15:17:08-H-(LK)       status:         15:17:08-H-(LK)       ****         15:17:08-H-(LK)       ****         15:17:08-H-(LK)       ****         15:17:08-H-(LK)       ****         15:17:08-H-(LK)       ****						
151:71:0+H-(w)       cpu 2: client_type = 0x7, memory = 134217728, host = surfire47         151:71:0+H-(w)       cpu 2: host_realtime process started, PID = 20291,         151:70:0+H(w)       cpu 1: client_type = 0x7, memory = 538870912, host = sunfire47         151:70:0+H(w)       cpu 1: client_type = 0x7, memory = 538870912, host = sunfire47         151:70:0+H(w)       cpu 1: client_type = 0x7, memory = 538870912, host = sunfire47         151:70:0+H(h)       RNNEDC STANDLONE         151:70:0+H(h)       Nexterver WARDMARE interaction is OFF ***         151:70:0+H(h)       Statting for waveserver to get pready         151:70:0+H(h)       Statting lockserver on fixed port = 23333         151:70:0+H(h)       Statting lockserver PROCESS         If ind:       Imatch case         Find:       Imatch case         File       Utilities         PCS State :       Imatch case         Image:       Shot Cycle: Pending Next Shot	15:17:10-M-(h2)	FINISHED INITIALIZ	ATIONS: RT=0x7fa498f		Bfaaffc, HOST=0x7f9b	9f0b0000 to 0x7f9t
15:17:10-1-(b2)       CPU 2; host_realtime process started. PID = 20291.         15:17:08-H-(h1)       FINISHED INITIALIZATIONS: RT=0x7fdic5080000 to 0x7fd24508dffe, H0ST=0x7fc31999c000 to 0x7fc3         15:17:08-H-(h1)       FINISHED INITIALIZATIONS: RT=0x7fdic5080001 to 0x7fd24508dffe, H0ST=0x7fc31999c000 to 0x7fc3         15:17:08-H-(h1)       RUMNOE STANDLOWE         15:17:08-H-(h1)       CPU 1; host_realtime process started. PID = 20280.         15:17:08-H-(h1)       IOKserver waiting for waveserver to get ready         15:17:09-H-(hk)       Iockserver waiting for waveserver to get ready         15:17:09-H-(hk)       Iockserver H82MRE interaction is 0FF ***         15:17:09-H-(hk)       ***         15:17:09-H-(hk)       Iockserver H82MRE interaction is 0FF ***         15:17:09-H-(hk)       *** <th></th> <td></td> <td></td> <td>217728 bost - s</td> <td>mfired7</td> <td></td>				217728 bost - s	mfired7	
15:17:08-H-(uw)       cpu 1: client_type = 0x7, memory = 536870912, host = surfire47         15:17:08-H-(h)       RUMMORE STANDLONE         15:17:08-H-(h)       Clickserver waiting process started, PID = 20288, 15:17:05-H-(h)         15:17:08-H-(h)       Setting lockserver on fixed port = 23333         15:17:08-H-(h)       *** lockserver for port = 23333         15:17:08-H-(h)       *** lockserver hord port = 23333         15:17:08-H-(h)       *** lockserver hord met interaction is OFF ***         15:17:08-H-(h)       *** lockserver hord met interaction is OFF ***         15:17:08-H-(h)       *** lockserver hord met interaction is OFF ***         15:17:08-H-(h)       *** lockserver hord met interaction is OFF ***         15:17:08-H-(h)       *** lockserver hord met interaction is OFF ***         15:17:08-H-(h)       *** lockserver hord met interaction is OFF ***         15:17:08-H-(h)       *** lockserver hord met interaction is of Filter:         Find: <ul> <li>match case</li> <li>Filte</li> <li>Filte</li> <li>Utilities</li> <li>PCS State :</li> <li>Ime Stamp:</li> <li>Shot Cycle: Pending Next Shot</li> <li>Shot Cycle: Pending Next Shot</li> </ul>						
15:17:08-H-(h)       RNNODE STANDHLÖNE         15:17:08-H-(h)       CPU 1; host_realtime process started, PID = 20288,         15:17:08-H-(h)       lockserver waiting for waveserver to get ready         15:17:05-H-(h)       lockserver HARDWRE interaction is OFF ***         15:17:05-H-(h)       ***         Find:       Imatch case         File:       Imatch case         File:       Imatch case         File:       Imatch case         File:       Utilities         PCS State :						1999c000 to 0x7fc?
15:17:05-H-(1k)       lockserver valting for waveserver to get ready         15:17:05-H-(1k)       setting lockserver PROMEME interaction is OFF ***         15:17:05-H-(1k)       *** lockserver PROMEME interaction is OFF ***         15:17:04-C-(ww)       STARTING UP WAVESERVER PROCESS         Find: <ul> <li>match case</li> <li>Filter:</li> <li>match case</li> <li>Filter:</li> <li>match case</li> <li>File</li> <li>Utilities</li> <li>PCS State :</li> <li>Ime Stamp:</li> <li>Status:</li> <li>Data:</li> <li>Shot Cycle: Pending Next Shot</li> </ul>				070312, NOSC - S	alli 11 647	
15:17:05-H-(1k)       Setting lockserver on fixed port = 23333         15:17:05-H-(1k)       ****         16:11       ****         17:05       ****         18:12:05       ****         19:12:05       ****         10:05       *****         10:05       *****         10:05       *****         10:05       ******         10:05       ******         10:05       ************************************						
Is:17:04-C-(w)     Status:     Data:     Shot Cycle: Pending Next Shot						
Find: Imatch case   Find: Imatch case   File View Log   File Utilities PCS State :   Imatch case   PCS State :   Imatch case   File Status:    Data:    Shot Cycle: Pending Next Shot		*** lockserver HAR	RDWARE interaction is			
Find: match case   File: View Log File: Utilities PCS State : Time Stamp: Shot Cycle: Pending Next Shot	15:17:04-U-(wv)	STARTING OP WAVESE	RVER PRUCESS			
File Utilities   PCS State :			1			
File       Utilities         PCS State :	Find:		🛛 🗖 match case	Filter:		🗖 match case
File       Utilities         PCS State :	p			,		
File       Utilities         PCS State :						
PCS State : Time Stamp: Data: Data	000		🔀 Vie	ew Log		
Time Stamp: Time Status: Data: Shot Cycle: Pending Next Shot	File Utilities					
Shot Cycle: Pending Next Shot	PCS State :	Ι				
	Time Stamp:	Ĭ		Status:	Data:	
			Shot Cycle <u>: Pe</u>	nding Next Sh	ot	
	0					100

### Choose the type of control

- Control
  - -Next shot
    - Changes made in GUI are written to the realtime computer
      - On pcspool ... change parameters for next simulation
      - On pcs-rt-3 ... change parameters for next shot
  - Future shot
    - Set up a shot for restoring later
  - Shared shot
    - Change a future shot made by someone else



#### Choosing "Next shot" opens the PCS GUI

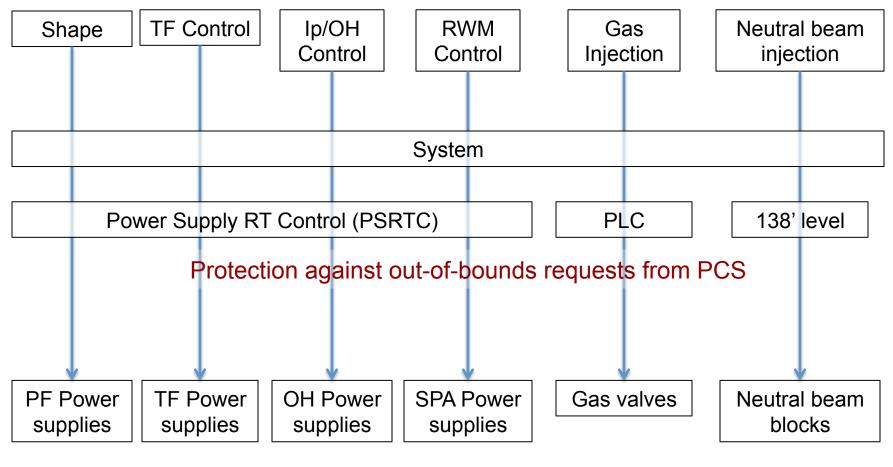
	X PLA	SMA CONTROL FOR TH	IE NEXT SHOT	
File Show Category Pl	hase Defaults Edit	Scales Bookmarks	Restore Information	
Ctgy: Power Supply RT C	Control, Phs: ShotStart, .	Alg: Power Supply Co	ntrol x,y:	<mark>Status</mark> Data
💠 Ctgy 💠 Seq 💠 Phs	🗖 X-phase 🗖 multiple	ot 🗖 select 🗖 add-d	rag add copy cut paste m	erge replace all copy all
♦ Time ♦ Sub ♦ Data	unchanged cancel	apply NEXT SH	DT Trace refresh replot blow	up adjust
Global Settings PF1AU Settings				
PF1BU Settings PF1CU Settings				
PF2U Settings				
PF3U Settings PF4 Settings				
PF5 Settings PF3L Settings				
PF2L Settings PF1CL Settings				
PF1BL Settings				
PF1AL Settings OH Settings				
TF Settings HF Settings				
Numerics mode 📼				
new x:				
new y:				
add delete replace				

Talk will now focus on navigating this window

 But first, let's take a step back

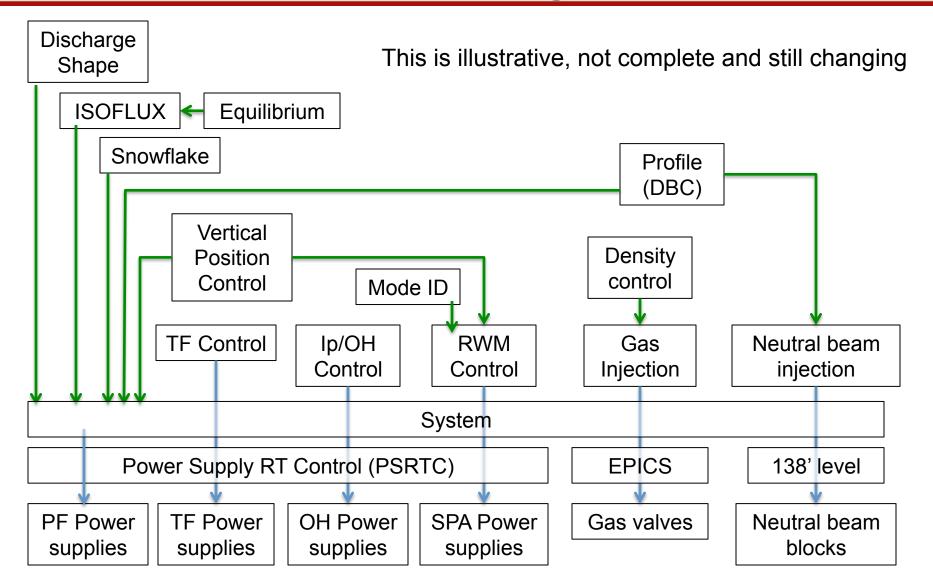
# PCS baseline categories align with control actuators

#### "Baseline Categories" provide control and/or control switches for a single actuator



#### Actuators

## Upper level categories send requests to baseline categories



**NSTX-U** 

### Every category has "Algorithms"

- TF control has two Algorithms
  - **TF current control**: request TF coil current versus time
  - **TF rampdown control**: ramp TF current from initial value to zero over a specified time interval
- Discharge Shape has three Algorithms
  - **Do Nothing**: category is disabled
  - Day0: Current control of PF coils
  - **PCC**: Day0 capability plus  $I_{PF}$  vs  $I_{OH}$ ,  $I_{PF}$  vs  $I_{P}$ , outer gap
    - Categories accrue algorithms as new versions come along

### Every category has "Phases"

- Each phase runs one algorithm with unique parameters P.O. can add/delete phases, cannot add/delete algorithms
- You could have multiple phases calling the same algorithm but with different control parameters
  - Divide pulse into separate experiments where a number of algorithm parameters change
  - Or save different settings for an algorithm
    - For example "rampdown" and "fast rampdown" in the IP/OH category
- Most categories have just one phase for each algorithm
  - For example, TF Control category has two Algorithms and Phases
    - [Phase] ShotStart = TF current control [Algorithm]
    - [Phase] Rampdown = TF rampdown control [Algorithm]

#### Every category has a "Primary" phase sequence

- "Primary" phase sequence sets timing of the phases
  - For example, TF Control Primary Phase Sequence:
    - T = -5s [Phase] ShotStart = TF current control [Algorithm]
    - T = +3s [Phase] Rampdown = TF rampdown control [Algorithm]
- Category may also have "Alternate" phase sequences
  - P.O. cannot add/delete alternate phase sequences
    - P.O. can change the phases in each alternative sequence
  - Algorithms switch to these alternate phase sequences
    - For example, IP/OH category has two alternate phases that are activated by the IP0 algorithm
      - Insufficient lp: Switch to this sequence if l<sub>p</sub> is below a threshold
         » [Phase] Rampdown = IP1 [Algorithm]
      - Ip LOC: Switch to this sequence if OH coil current exceeds bounds
         » [Phase] Fast rampdown = IP1 [Algorithm]

#### Let's review

- PCS control is divided into categories
- Every category has a library of algorithms
- You create a library of phases – Each phase uses a single algorithm
- You define the timing of phases in the Primary phase sequence
  - You also define the timing of phases in any Alternative (dynamic) phase sequences

# The main task of a P.O. is to change the input parameters for control algorithms

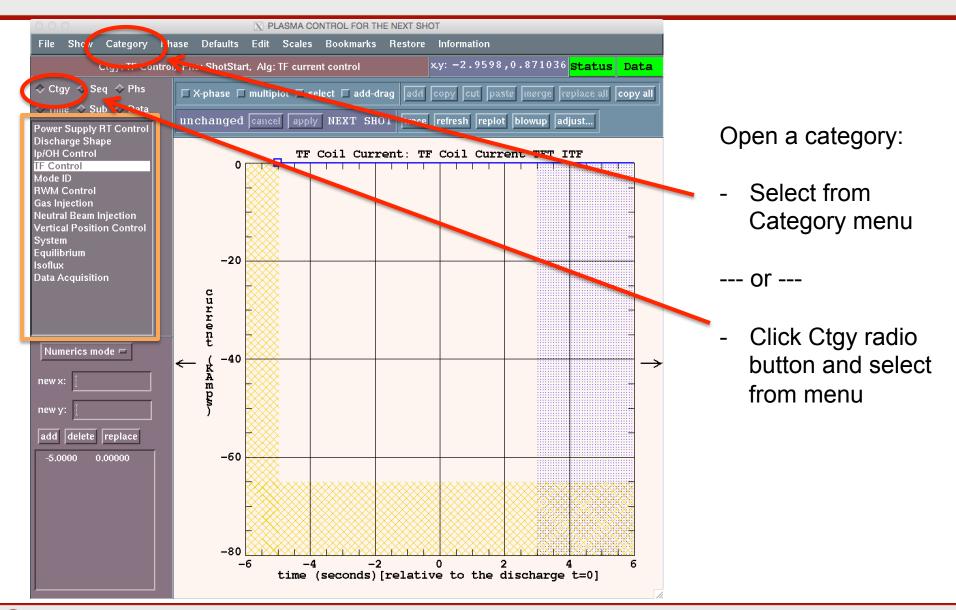
- Inputs to algorithms are organized into subsets
  - [Cat] TF Control, [Phase] ShotStart, [Alg] TF current control
    - Coil Current
      - TF Coil Current
    - TF gains
      - TF Gain, P
      - TF Gain, I
    - Rampdown Info
      - Automatic Rampdown
      - Ip Rampdown Threshold

Subset Waveform, table or GUI

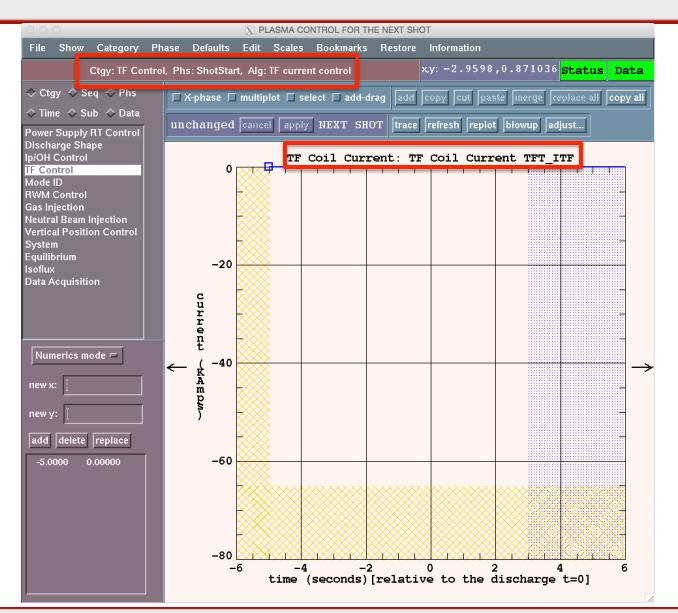
#### We aim to have specs for each algorithm

**NSTX-U** 

#### Let's begin the tour



### Selecting a category

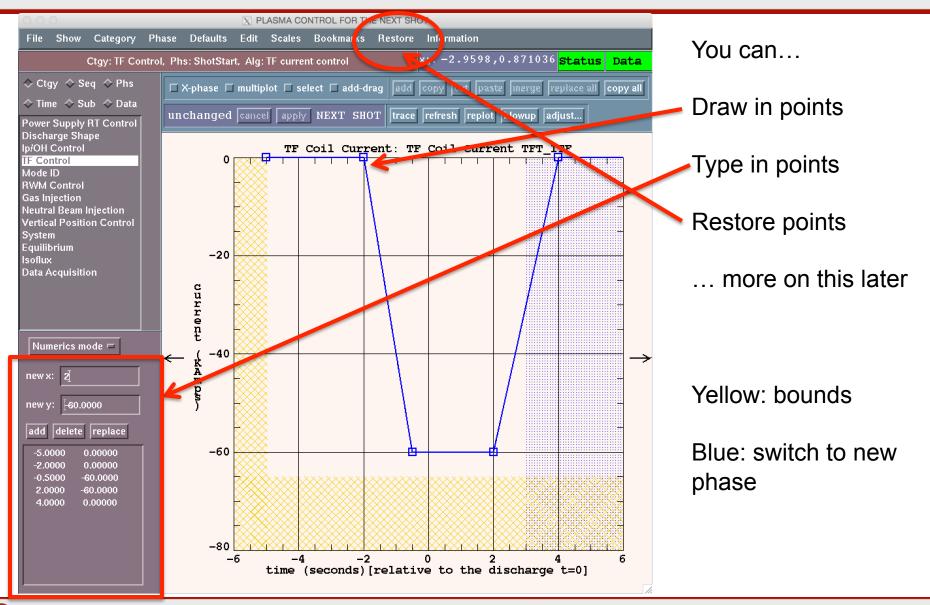


I have selected "TF Control" Category

Now we see the first waveform in the first subset of the algorithm in the first phase of the primary sequence

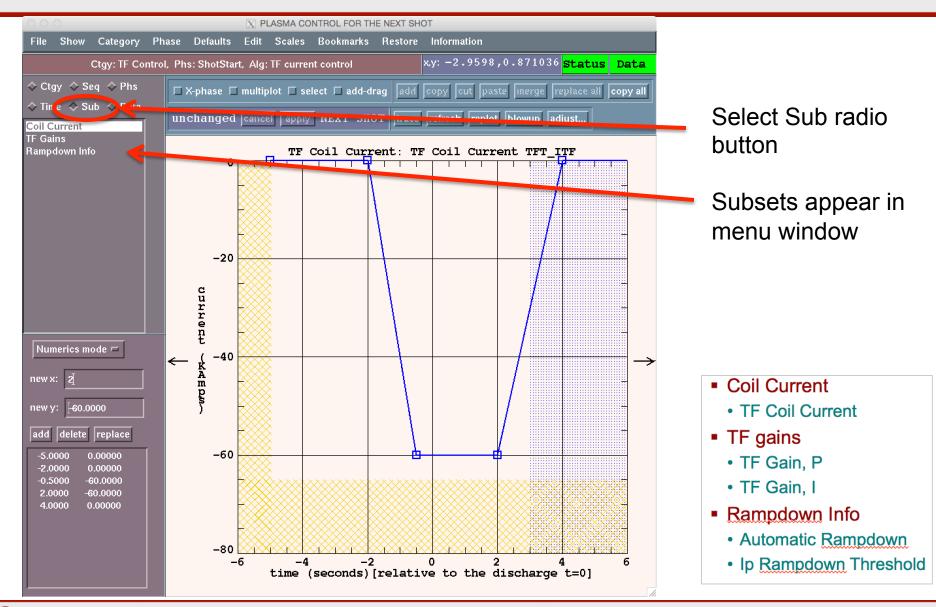
- Coil Current
  - TF Coil Current
- TF gains
  - TF Gain, P
  - TF Gain, I
- Rampdown Info
  - Automatic Rampdown
  - Ip Rampdown Threshold

#### Enter a waveform

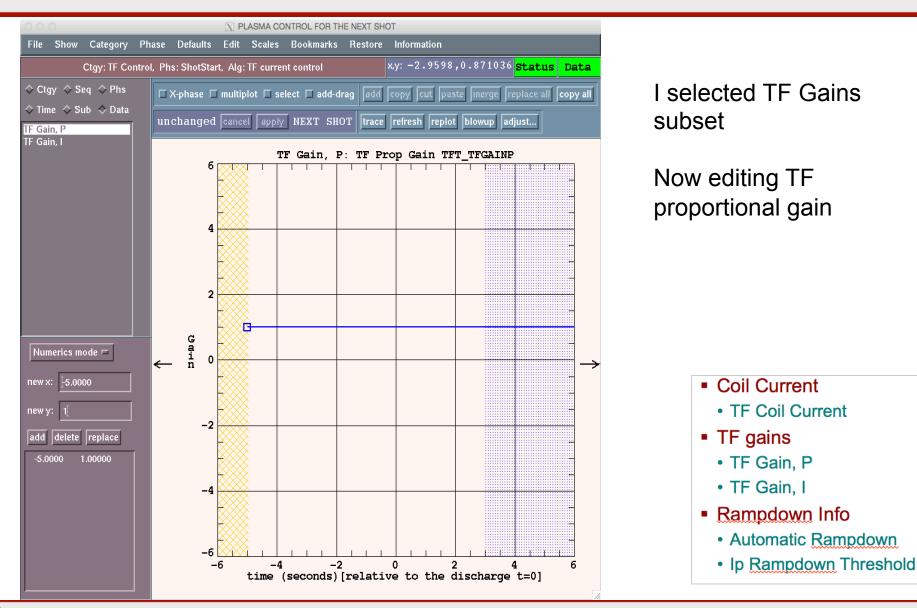


**NSTX-U** 

#### Navigate to a new waveform

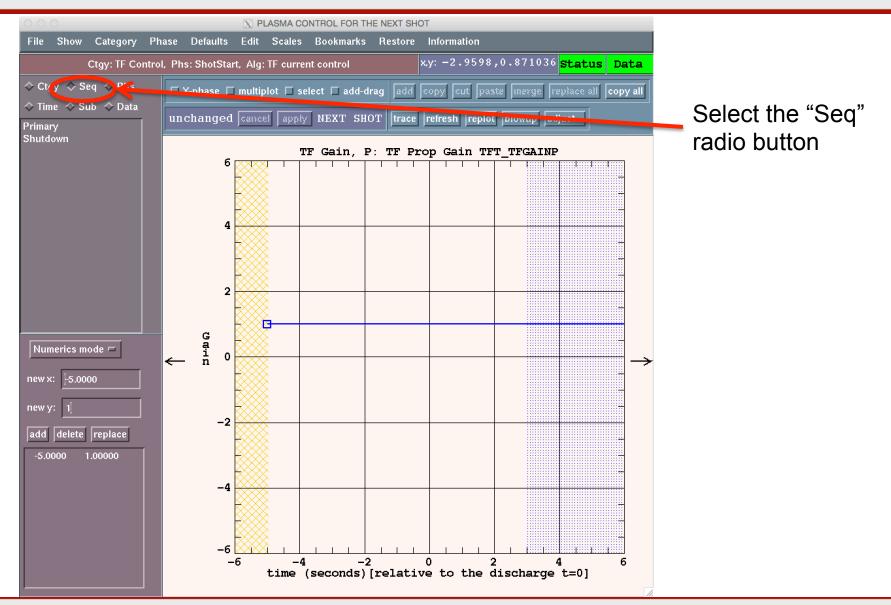


#### Enter values for a new waveform



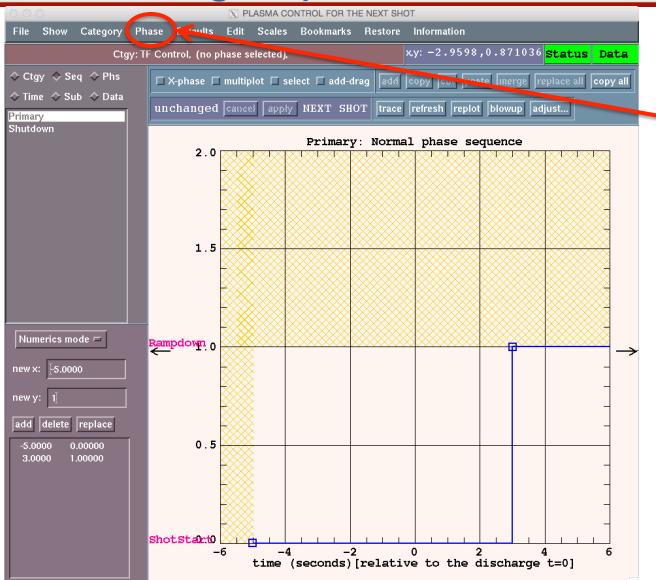
**NSTX-U** 

#### Navigate to Phase sequence



**NSTX-U** 

# Primary sequence describes order and timing of phases within category



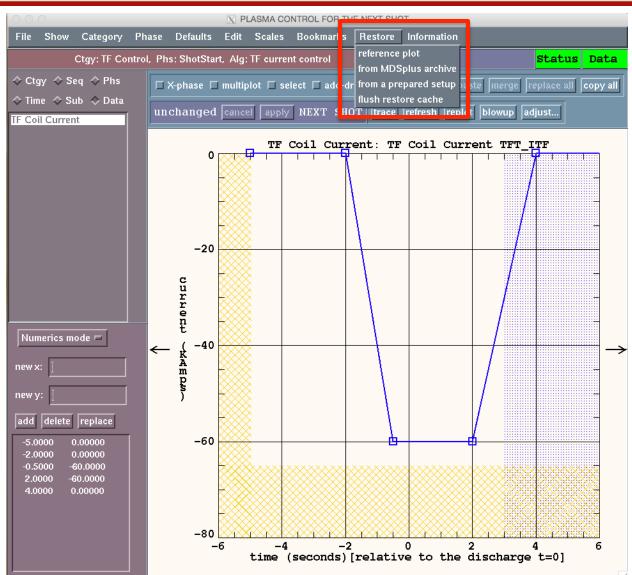
You can add/delete phases in the category using the phase menu

The waveform describes timing of phase switching

TF category has one alternative sequence called Shutdown.

It is called if Ip < threshold

# Restoring data from older shots or prepared shots is a common activity



- Restore menu
  - Reference plot
    - Plot data from a shot to compare
  - MDS archive
    - Restore data from an old shot
  - Prepared setup
    - Restore data from a future shot
  - Flush cache

#### Loading data from an old shot or prepared setup

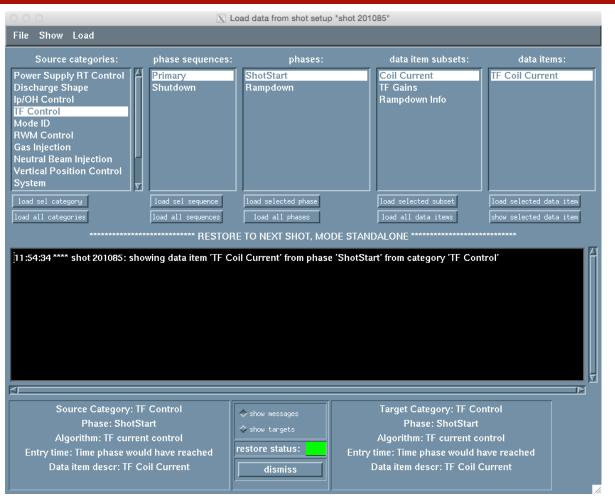


#### Load data from a prepared setup

0.00	∑ Choose a shot setup
lbattagl XMP106_PF5421a lbattagl junk lbattagl nothing lbattagl test	Selected shot setup: dbattagl XMP106_PF5421a Setup Description:
lbattagl test2	
	setup filter: [dbattagl*
	delete all backup files ignore backup files RESTORE TO NEXT SHOT select data to load
	load all categories
	dismiss

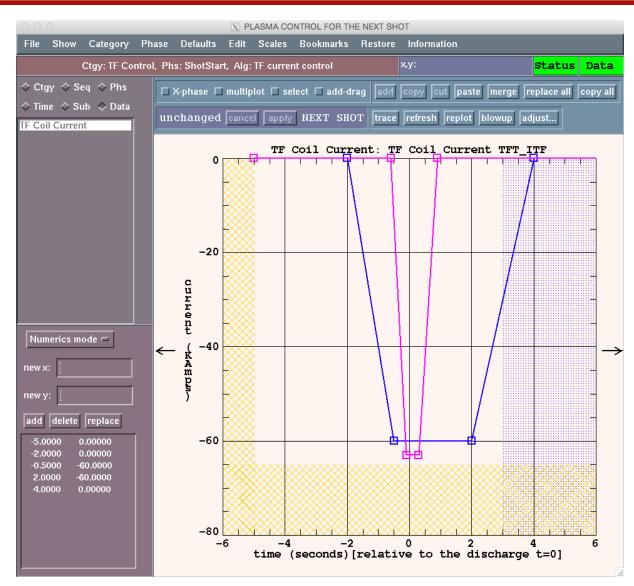
Load all categories: Reload everything it can Load sel category: Just reload selected category Select data to load: Opens a new window

#### "Select data to load" gives full control over what to load and where



- Load selected data item
  - Loads that data item from the source
- Show selected data item
  - Plots that data item for comparison

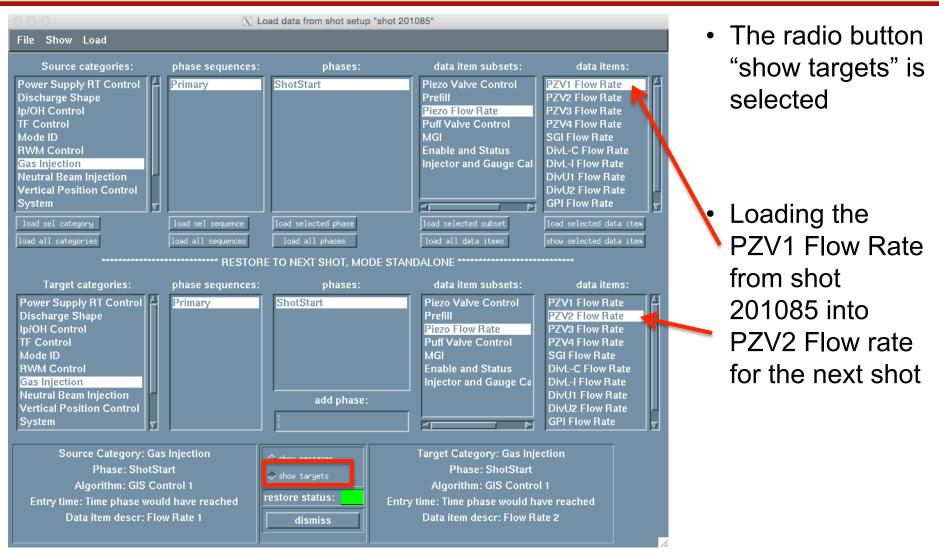
## Using "Show selected data item" before restore is good practice



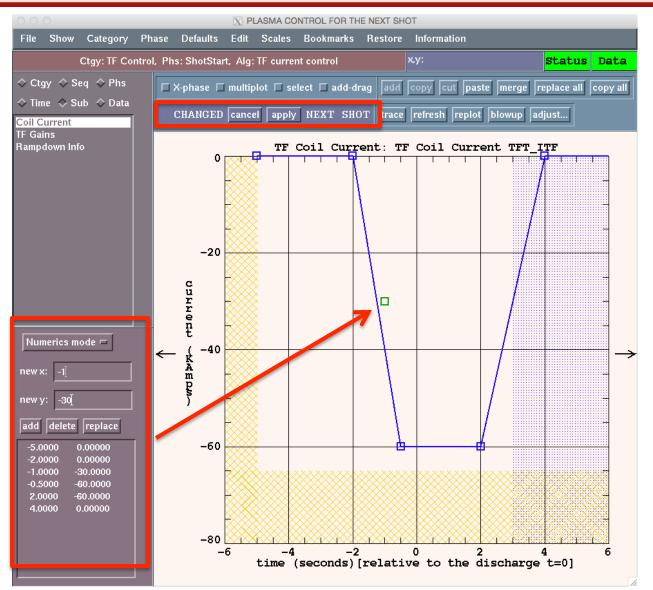
 Comparing the TF current request waveform (blue) to the TF request for shot 201085 (pink)

**NSTX-U** 

## Load a data item into a target with a different name



## Changing a waveform

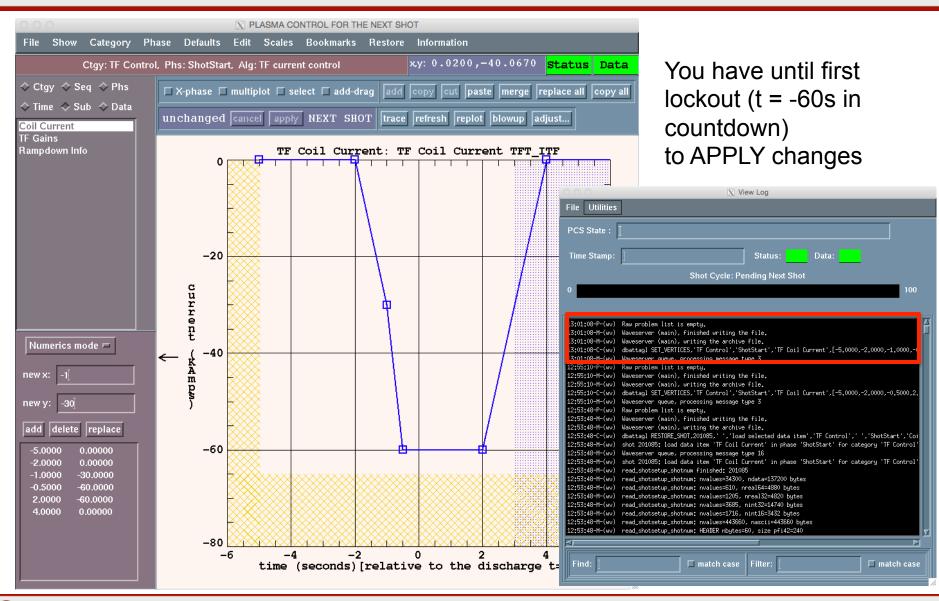


 A point is added by entering the new x and new y coordinates and pressing "add"

THE CHANGE
 IS NOT MADE
 UNTIL YOU
 PRESS APPLY

🚺 NSTX-U

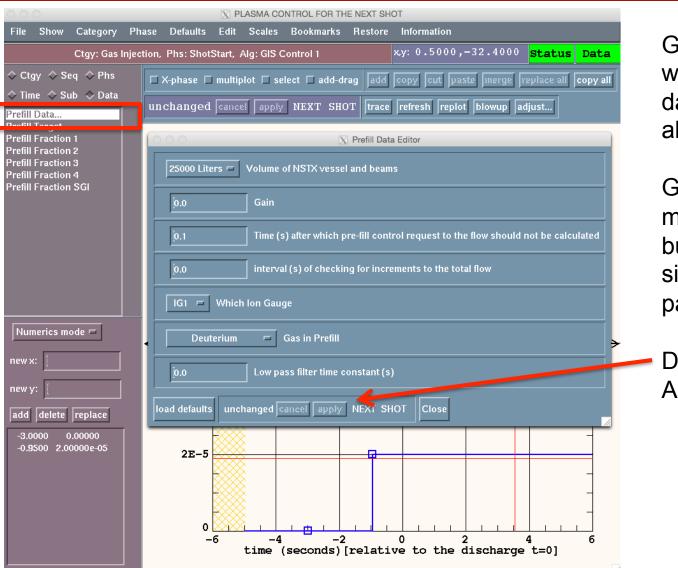
#### Press apply and the change is made



# Entering and manipulating waveforms is best learned on the job

- Mouse control lets you select points, add points, delete points, pan and zoom
- There is a clipboard for cutting, copying and pasting
- Scale or shift selected points
- Overplot references, proposed changes
- Load points from a text file or use a function generator

## Some data items are GUIs instead of waveforms



GUIs are indicated with "…" in the list of data items within a algorithm subset

GUI could have a mix of menus, radio buttons, sliders and single number parameters

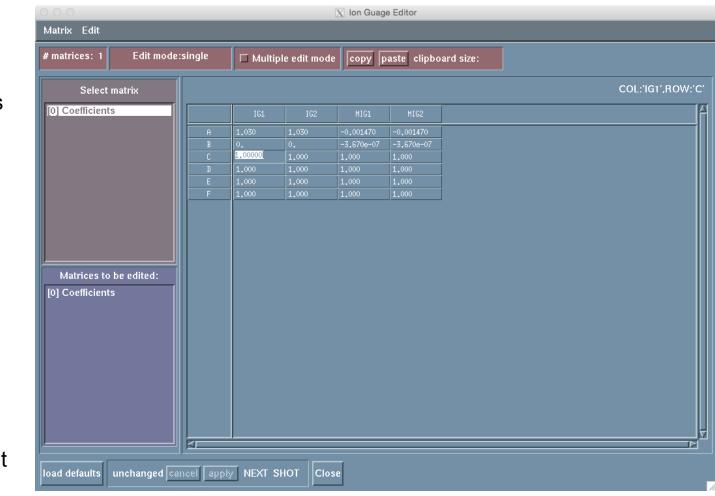
Don't forget to hit APPLY!

### GUIs can also be tables

Multiple versions of the table can exist (a waveform is used to select which table to use)

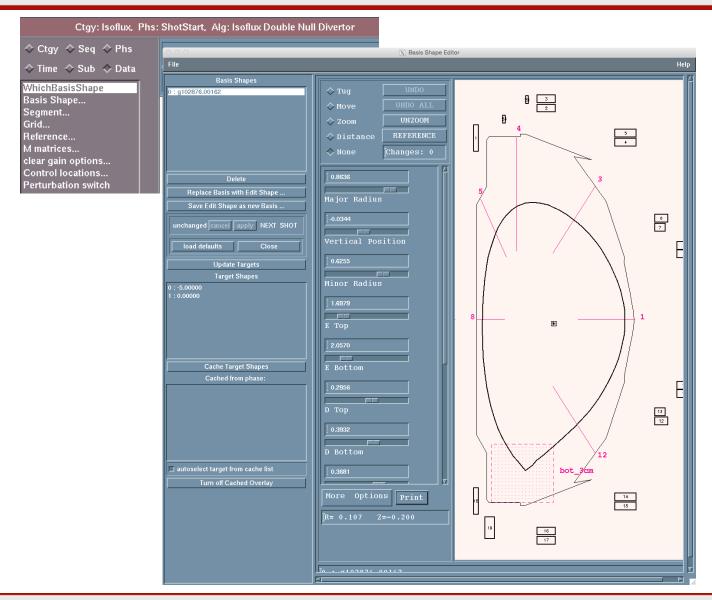
You can edit the same cell in multiple tables

Don't forget to hit APPLY!





#### The ISOFLUX GUIs deserve a separate talk



**NSTX-U** 

#### Indicators on the PCS GUI

Ctg: Power Supply RT Cartrol. Phis: ShotStart. Alg: Power Supply Control       VS       Petatus       window. You could have multiple "Next shot" and "Future shot"         Ctg2: Editings PF180 Settings PF180 Settings PF181 Settings PF181 Settings PF181 Settings PF181 Settings FF181 Se

### If you get a red or yellow Data light ...



- Go to "Information" > "Raw data problems"
  - -Window shows the issue(s)
  - Most likely to occur when restoring a shot



#### Start exploring the PCS GUI

- Get in the PCS Unix group
- Get the stand-alone PCS software
- Refer to:
  - PCS programmers manual:
  - <u>http://nstx.pppl.gov/nstx/controls/pcs/PCS\_doc/master.pdf</u> – NSTX-U PCS specs in the NSTX PCS shared google folder
- Read the specs and navigate through the categories
  - Practice changing things
  - Ask questions

### Summary

- PCS software is divided into Categories
   Each category has a library of control algorithms
- The PCS interface is used to set the sequence of phases where each phase runs a single algorithm
- Each algorithm has waveforms and GUIs that can be edited using the PCS interface
- Dennis: "With power and flexibility comes complexity" – With time and practice, it all makes sense