

## Testing the Phoebus Alarm System for the Neutral Particle Spectrometer

Aaron Brown, Mary Ann Antonioli, Peter Bonneau, Pablo Campero, Brian Eng, George Jacobs, Mindy Leffel, Tyler Lemon, Marc McMullen, and Amrit Yegneswaran  
*Physics Division, Thomas Jefferson National Accelerator Facility, Newport News, VA 23606*  
 July 25, 2023

To test whether the Phoebus alarm system is capable of handling the process variables (PVs) that will be used for the Neutral Particle Spectrometer (NPS) control and monitoring program, a softIOC to generate test PVs was previously developed [1]. A random number generator was developed to provide values to the PVs generated by the softIOC.

The softIOC was developed on the computer that hosts the PVs for testing, and required downloading the configuration, database, and make files. The Python program *db-test-2.py*, Fig. 1, was developed to create the 1958 database records needed to simulate the NPS signals.

```
dbLines = [
'record(al, "pvname")',
'{' ,
'    field(DESC, "Temp at crystal")',
'    field(SCAN, "1 second")',
'    field(HIHI, "30")',
'    field(HIGH, "28")',
'    field(LOW, "20")',
'    field(LOLO, "18")',
'    field(HHSV, "MAJOR")',
'    field(HSV, "MINOR")',
'    field(LSV, "MINOR")',
'    field(LLSV, "MAJOR")',
'}']
```

FIG. 1. Screenshot of portion of *db-test-2.py* program used to generate database records.

The number of PVs is determined by the number of monitoring signals. Each signal PV has several associated PVs—average, standard deviation, high and low signal limits, interlock status, latch status, number of points to average, trip delay values, and enables for sensor, average, interlock, and trip delay.

The *db-test-2.py* program specifies PV names and associated fields for each signal (alarm limits, scan rate, and the severity of each alarm type). There are four alarm limits—HIHI, HIGH, LOW, and LOLO—that are initialized to 30, 28, 20, and 18 (arbitrary units) respectively, and can be changed via the Phoebus screens, Fig. 2.

Detector Frame Alarm Testing																
Temperature																
PV name	Sensor	read	HSA set	HSA read	HSB set	HSB read	LOW set	LOW read	LOLO set	LOLO read	Alarm status	Alarm severity	Scan rate	range [°C]	Min T [°C]	Max T [°C]
temp_hhr_tf_1	Frame 1	20.75	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1.980000	18.00	20.00	30.00
	Frame 2	21.64	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1.980000	18.00	20.00	30.00
	Frame 3	28.35	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	HIGH	MINOR	1.980000	18.00	20.00	30.00
	Frame 4	22.31	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1.980000	18.00	20.00	30.00
Humidity																
temp_hhr_tf_1h	Frame 1	24.24	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1.980000	18.00	20.00	30.00
	Frame 2	23.26	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	HIGH	MINOR	1.980000	18.00	20.00	30.00
	Frame 3	25.50	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1.980000	18.00	20.00	30.00
	Frame 4	25.99	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1.980000	18.00	20.00	30.00
Dew point																
temp_hhr_tf_1p	Frame 1	24.91	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1.980000	18.00	20.00	30.00
	Frame 2	24.70	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1.980000	18.00	20.00	30.00
	Frame 3	20.21	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	HIGH	MINOR	1.980000	18.00	20.00	30.00
	Frame 4	28.25	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	HIGH	MINOR	1.980000	18.00	20.00	30.00

FIG. 2. Screenshot of Detector Frame Alarm Testing Phoebus screen using random values from *randomTestScript.py* Python program.

Of the 1958 database records, only about 650 receive a random number. These are the PVs that require four additional PVs created specifically for the test system—three input PVs

for the scan rate, range, and minimum value of the random number, and one additional PV to calculate and display the maximum value of the random number. Therefore, the total number of PVs generated by the softIOC is ~2550.

The Python program *randomTestScript.py*, Fig. 3, was developed to provide the random numbers using the random Python package. *randomTestScript.py* creates an array of all signal PVs (the variable for this array is named *PVs*) and an array of PVs that sets the range and minimum of the random numbers (the variable for this array is named *rminPVs*). *randomTestScript.py* initializes the range and minimum values for each signal PV to 0. The range and minimum values can be changed by the user.

```
try:
while True:
    j = 0
    for i in range(len(PVs)):
        rang = caget(rminPVs[j])
        j = j + 1
        rmin = caget(rminPVs[j])
        rmax = rang + rmin + 1
        x = "%.2f" % random.uniform(rmin, rmax)
        val = caput(PVs[i], float(x))
        j = j + 1
except:
    print("Interrupted")
```

FIG. 3. Screenshot of part of *randomTestScript.py* Python program.

Next, *randomTestScript.py* sends the Channel Access command *caget* to retrieve the minimum value of the random number and a second *caget* command to retrieve the value for the range of the random number. These two values are used to calculate the maximum value for the random number as follows:  $\text{range} + \text{minimum} + 1 = \text{maximum}$ . The one is added so that the maximum value will be included in the possible random number values (e.g., if the minimum value is 0 and the range is 5 the maximum value of the random number will be 5; however, for 5 to be a possible value, the minimum and maximum values passed to the *random.uniform* Python command need to be 0 and 6).

To conclude, a softIOC has been developed to simulate all signals from the NPS. The simulated values are provided by a random number generator with range and minimum values.

[1] A. Brown, et al., *Development of the EPICS SoftIOC for Phoebus Alarm Testing*, DSG Note 2023-12, 2023.