ExperimentDesigner: A Tcl/Tk INTERFACE FOR CREATING EXPERIMENTS IN EPICS *

H. Shang[†], M. Borland, C.Y. Yao, W. Guo, ANL, Argonne, IL 60439, USA

Abstract

ExperimentDesigner is a Tcl/Tk interface that allows users to easily design and run complicated experiments using a convenient graphical user interface (GUI). Features include: process variable monitoring, which pauses the experiment when values are out of range; user-defined initialization, execution, and finalization sequences; support of complex execution chains containing actions such as setting controls, reading values, running external programs, interacting with the user, etc.; collection of output data for convenient postprocessing; saving and loading of experiment configurations; convenient use of SDDS Toolkit programs; and execution of experiments from the command line without a GUI.

FEATURES

The ExperimentDesigner software is composed of three parts: experiment setup, configuration saving and loading, and experiment execution as described below.

Experiment Setup

ExperimentDesigner has six panels for allowing the user to set up an experiment. Each panel has different functions and features.

The ProcessVariables (PV) panel allows the user to add:

- Readback PVs for monitoring. If any PV is out of the range defined by the user-provided lower and upper limits, the experiment will pause until the out-of-range condition is resolved.

- Equations for calculating the values from PVs and results of other equations.

- Parameters for experiment description or recording any featured PV values.

- Setpoint PVs for setting PV values based on the value of readback PVs or equation values provided in this panel. The equation and setpoint PV features allow the user to set up complicated relationships between PVs.

- Control PVs for changing values during the experiment. These allow the user to change PV values at each experiment step through various methods, such as

  - increment - the value of PV is changed by the given increment each step.

  - specify_range - the value of control PV will be changed from given initial value to final value at a uniform step size based on the number of experiment steps.

  - specify_fixed_value - the user provides a list of one or more comma- or space-separated values. Each value will be sent to the control PV on successive steps of the experiment. If the number of values in the list is less than the number of experiment steps, the last value in the list will be used for the remaining steps. Excess values are ignored.

  - list_value - this is similar to specify_fixed_value except that the number of experiment steps is forced to be equal to the number of values listed.

  - set_from_file - the values for setting the control PVs are obtained from an SDDS file.

Initialization, Execution, and Finalization

An ExperimentDesigner experiment consists of Initialization, Execution, and Finalization, followed by postprocessing. Execution involves performing a series of actions repeatedly in a loop. In contrast, Initialization and Finalization involve performing a series of actions one time.

The Initialization and Finalization panels allow the user to define a series of two types of actions, namely setting PV values and running external scripts. An example of initialization would be setting values of process variables and then waiting for equipment to be ready using a script. An example of finalization would be putting a piece of equipment in a standby or off condition.

The Execution panel allows the user to design a complicated execution chain inside the main loop of the experiment. There are eleven types of actions, which may be used in any order, though some should be given only once:

- TestPV pauses the experiment if any PV is out of range and resumes the experiment when all PVs are within range.

- ChangeControl changes the value of the control PVs specified in the PV panel.

- SetValue sends the setpoint PV values in each experiment step. Recall that these differ from control PVs in that they depend on readback PVs via equations. This should be given only once.

- SetPV sets a given PV to a given value. This could be used, for example, to repeatedly initiate some process in an IOC.

- RunSDDSProgram allows running an SDDS program [1, 2, 3] at each experiment step. Usually, this action type is used to execute SDDS data acquisition programs with automatically-generated commandline arguments.

- RunCustomProgram and RunScript allow execution of programs or scripts. They are very similar, with the former being more general and the latter more convenient for scripts using the OAG Tcl/Tk argument parser.

- ReadValue reads the PV values declared in the PV panel, calculates any equation values, and writes the values to an output file.

- RunStatistics computes statistical values for the readback PVs and writes the values to the statistics output file. This should be given only once.

- WaitTime waits a given period (in seconds).

- WaitForUser pops up a user-provided message and waits for the user to acknowledge the message. It can be used to give the user time to make adjustments or observations that have not been automated.

- QueryUser produces a pop-up dialog that asks the user to input a value. This allows the user to enter data that is not accessible from EPICS, or just to enter a comment for each data point.

Output and Postprocessing

The OutputFiles panel lists all the output files created by the experiment, allowing the user to view and/or process the output files after the experiment. The entries in this panel are created automatically as the user defines the experiment. The widgets containing the output filenames feature a command pulldown menu that allows the user to quickly use the file with a variety of SDDS-based GUIs for data analysis and display.

The Postprocess panel allows the user to add a postprocessing command that is automatically executed after the experiment is done. The user can design the calling syntax for the postprocessing command, which is typically a Tcl/Tk script using the SDDS Toolkit for data analysis and display. The postprocessing command can also be triggered by pressing a button, which is convenient if the user wants to re-run the postprocessing.

Running the Experiment

ExperimentDesigner provides several different execution-related buttons for the user's convenience. For example, the user may run the experiment with or without initialization and also run initialization separately. The user may pause and subsequently resume the experiment at any time. The experiment may also be terminated at any time. The option to restore the original settings after experiment is done is also provided (this can be done with the finalization sequence as well).

ExperimentDesigner provides the capability of saving the current configuration to a set of files, which can be used to repeat the experiment in the future or create a similar experiment by modifying the saved configuration. This feature saves a lot of set-up time when repeating similar experiments. If an experiment has been set up through the interface and the configuration is saved, the user can run the experiment using the saved configuration through the command line without an interface. Thus, an ExperimentDesigner experiment could be run as an action of another ExperimentDesigner experiment.

APPLICATIONS

ExperimentDesigner software has been widely used in accelerator experiments at APS because of its many convenient features and the sophisticated automation it supports. Examples of applications are given in this section.

Vertical Decoherence Experiment

ExperimentDesigner was used to study betatron oscillation decay after a vertical kick to explore the relation between the decay pattern and the longitudinal tune, the kick strength, and the chromaticity [4].

In this particular experiment, the monitor PVs are the beam lifetime, vertical emittance, and rf voltage feedback error. The latter is monitored to ensure that the experiment is done safely and correctly. The control PV is the rf gap voltage feedback setpoint, whose value is changed at each experiment step. Actions in the experiment loop include ChangeControl to change the control PV, WaitTime to wait for two seconds after the control PV is changed, ReadValue to read the PV values and write them into a file, and a user-supplied script to change the kicker strength through a series of ten values and record the BPM history for each value.

The experiment is done as follows: Each time, lower the chromaticity manually by 0.5 units and then run ExperimentDesigner with the above setup. The rf gap voltage was changed in ten steps by ExperimentDesigner. After the rf gap was changed, the PV values were read and written into a file, following which the user script was run to vary the kicker strength through ten values. The user script recorded the BPM history after waiting for two seconds. In this way, about 1000 BPM history snapshots were obtained within several hours, which would be impossible to do manually.

PAR Experiments

PAR is a particle accumulator ring with a 30.7-m circumference and a cycle time of 500 ms. It employs a first-

harmonic rf system for capture and a 12th-harmonic system for bunch compression. ExperimentDesigner is used to perform experiments related to routine PAR [5] operation. For example, PAR synchrotron tune scan and bunch-length scan experiments are now used routinely for tuning up PAR [6].

SYNCHROTRON TUNE AND BUNCH LENGTH SCAN: We are interested in the synchrotron tune change during the initial capture and the 12th harmonic rf capture process. The experiment is designed to record the synchrotron tune waveform with variation of the delay trigger. The synchrotron tune waveform is acquired by a vector spectrum analyzer and downloaded to waveform files. A delay generator is used to produce a delayed trigger that scans across the PAR cycle in steps of several ms each. To make the tool useful during SR top-up operation, a top-up warning PV is used to control program execution such that injector returns to normal 30 seconds before the top-up injection starts. A postprocessing script is used to display the scan results at the completion of the experiment. Figure 1 shows an example of such a display after the experiment is done.
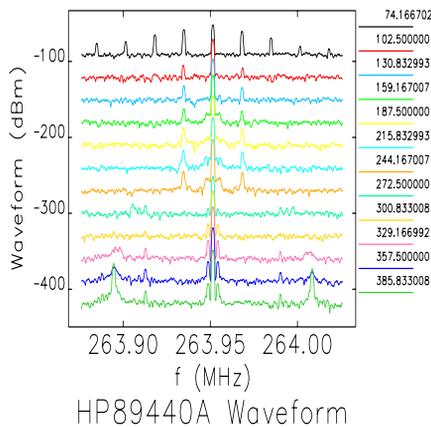
Figure 1: HP waveform curves at different delay times (in ms) shown by different colors.

The above figure can be displayed again after loading the same configuration and running the postprocessing command by pressing the execute button on the postprocess panel.

In order to achieve shorter bunches and better bunch purity, we are also interested in the bunch-length change during the initial capture and the 12th harmonic rf bunch-length compression process. We modified the just-described experiment to acquire streak camera data, which measures the duration of synchrotron light pulses from the beam.

PAR KICKER WAVEFORM SCAN: In this experiment, a fluorescent screen in the PAR to booster (PTB)

beam transport line is inserted and the video signal routed to a digitizer. Process variables provide beam size and centroid position information. By shifting the timing of a target kicker while toggling its strength between two fixed values, the time-dependent kicker waveform shape can be deduced from the differences in the centroid positions.

CONCLUSION

ExperimentDesigner is a powerful tool for interactive design and execution of experiments. Features include:

- The ability to pause execution until PVs are within specified boundaries.

- User-defined initialization, execution, and finalization sequences supporting complex execution chains containing actions such as setting controls, reading values, running external programs, interacting with the user, etc.

- Collection of output data for convenient postprocessing.

- Saving and loading experiment configurations for later re-use.

- Convenient use of SDDS Toolkit programs.

- Support for a user-provided postprocessing command.

- Execution of experiments from the command line without a GUI.

- Optional restoration of the original settings after the experiment.

REFERENCES

[1] M. Borland and L. Emery, "The Self-Describing Data Sets File Protocol and Program Toolkit," Proc. ICALEPCS, p. 653, 1995.

[2] R. Soliday et al., "New Features in SDDS Toolkit," Proc. PAC 2003, p. 3473, http://www.jacow.org.

[3] H. Shang et al., "New Features in SDDS-Compliant EPICS Toolkit," Proc. PAC 2003, p. 3470, http://www.jacow.org.

[4] W. Guo et al., "Generating Picosecond X-ray Pulses with Beam Manipulation in Synchrotron Light Sources," these proceedings.

[5] M. Borland, "Commissioning of the Argonne Positron Accumulator Ring," Proc. PAC 2003, p. 287, http://www.jacow.org.

[6] C. Yao et al., "Results of Preliminary Tests of PAR Bunch Cleaning," these proceedings.