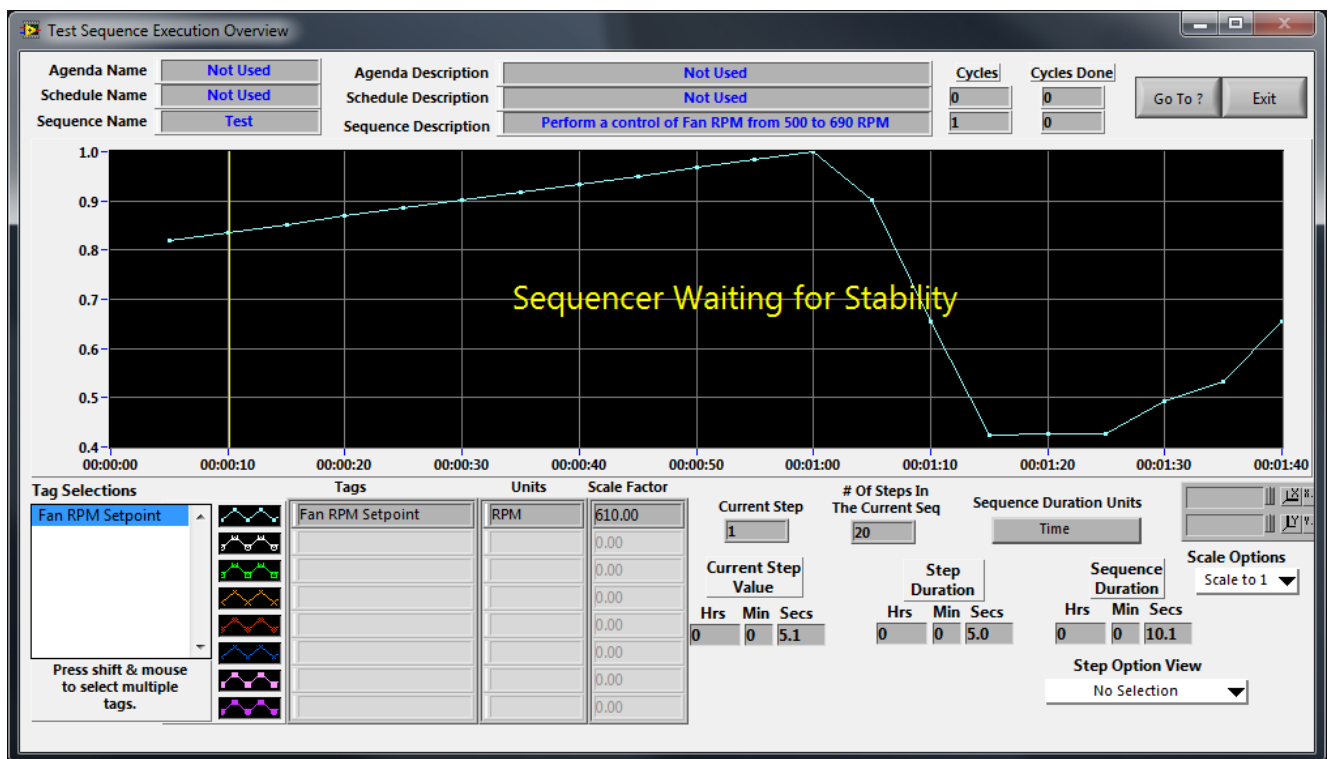


Test Automation Configuration has never been easier. Test SLATE enables the user to quickly and easily build even the most complex test sequences for automated or manual operation.

Test SLATE automated test sequencer.

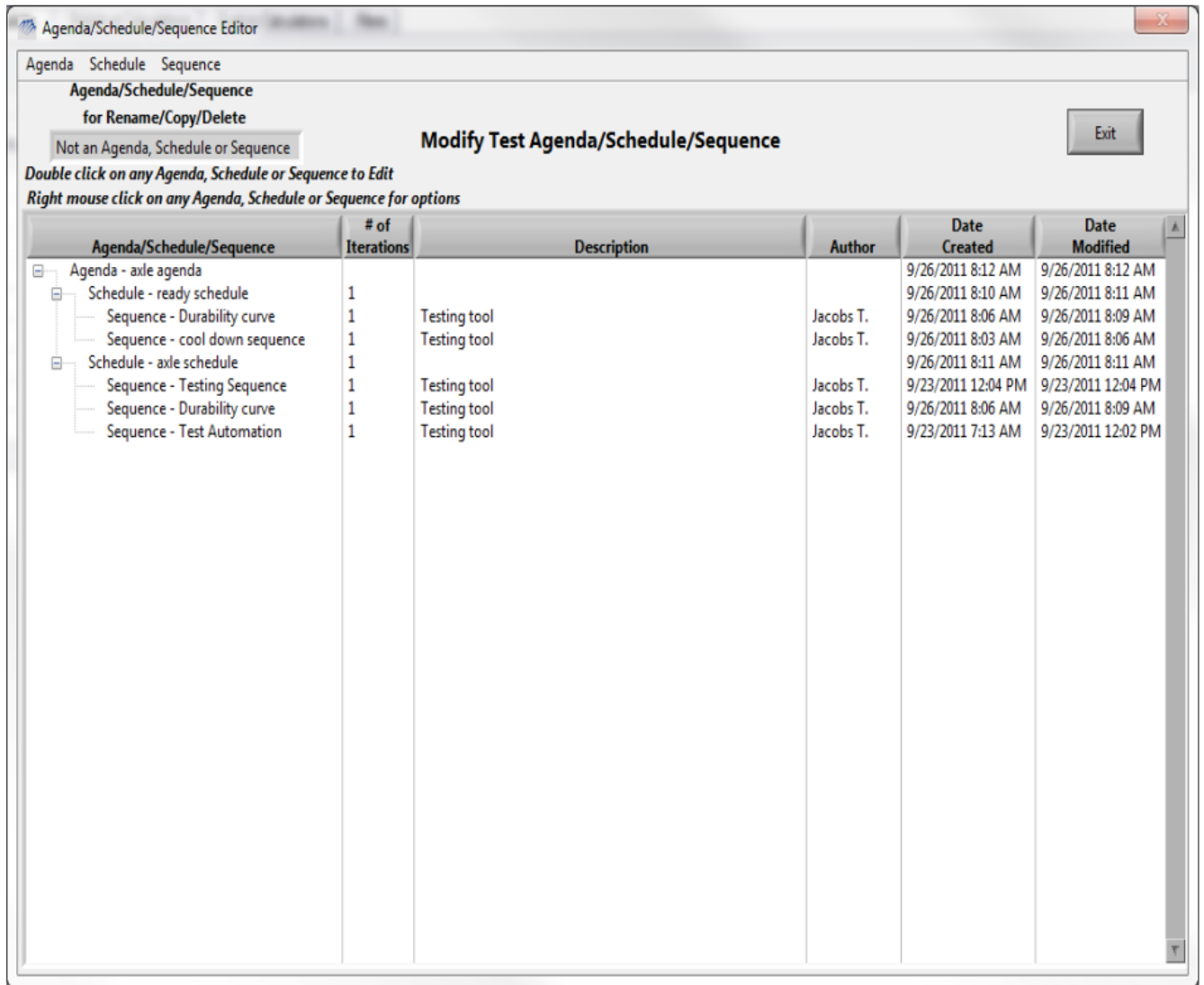
The Test Sequence builder enables the user to configure test sequences to fit specific test applications. The user can define a variety of conditions or events to trigger data storage or other actions. Example configurable items in the test sequence builder include facility or test article setpoints; time to remain at each step; the number of data points to be taken or duration of data storage at each step/sequence; and the conditions or events causing a transition to the next step/sequence. Test SLATE's test sequencer provides multiple analog and digital setpoint generation through profile plots or alphanumeric spreadsheet-type interfaces. Analog setpoints can be generated for internal PID loop control or for output to an external controller.



Monitor progress during testing

Programmable branching logic is included to provide automated intelligence for performing or jumping to other steps or sequences as test variations occur. The test sequencer also provides the capability to perform various functions per step, including alarm/trip checking, PID gain scheduling, pauses, operator messages with optional required response, tag stability with multiple logic conditions and a transfer on-step stability failure, data acquisition criteria with multiple data stability logic conditions, display of any recognized file with automatic removal, and automated reports.

The test sequencer features three nested tiers of test sequences to provide for endurance or repetitive test cycling. A nested tier of sequences is referred to as a Test Schedule, and a nested tier of Test Schedules is referred to as a Test Agenda. Test sequencing messages are stored to a log file for later analysis.

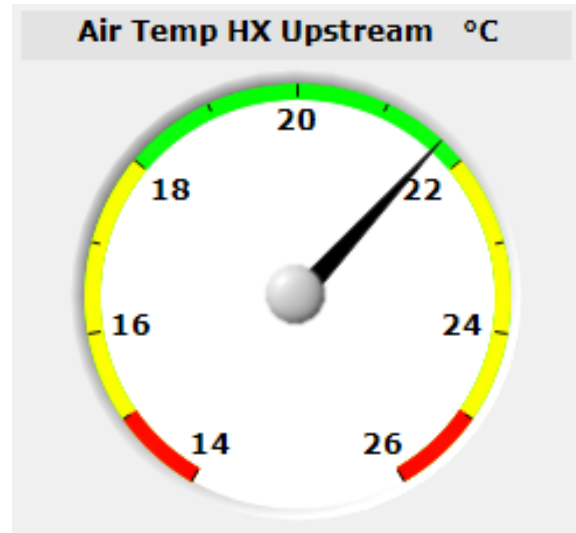
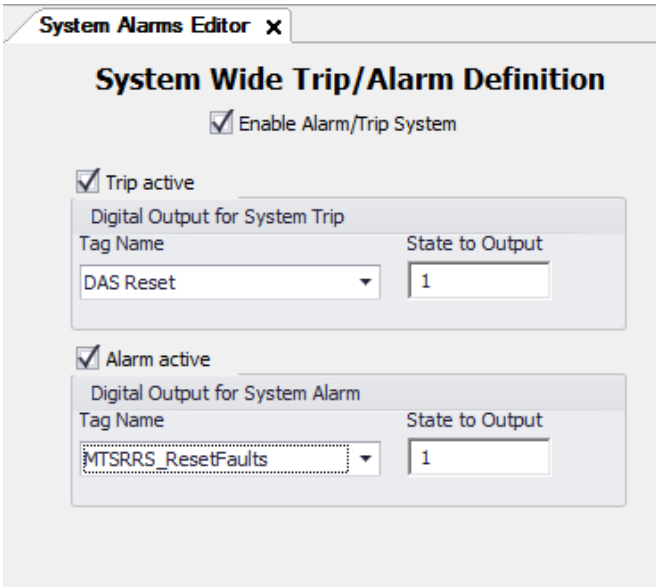


Define multiple layers of sequences

[Test SLATE alarm and trip detection.](#)

Test SLATE provides four levels of windowing alarms/trips with optional operator acknowledgment, two levels of rate of change alarms/trips with optional operator acknowledgment, and alarms/trips based on the variance of a group of analog type tags. Test SLATE also has the ability to alarm/trip digital input tags with optional operator acknowledgment. Digital outputs can be set on any alarm or trip to signal alarm horns, lights, or other logic devices that a fault condition exists. Tags in an alarm or trip state will change to user-defined colors on all displays. Operator messages are generated when an alarm or trip condition changes. Alarms or trips can also be used by the Test Sequencer to cause automatic jumps to shutdown or other sequences in the event of a fault condition.

Alarms and trips are easily displayed

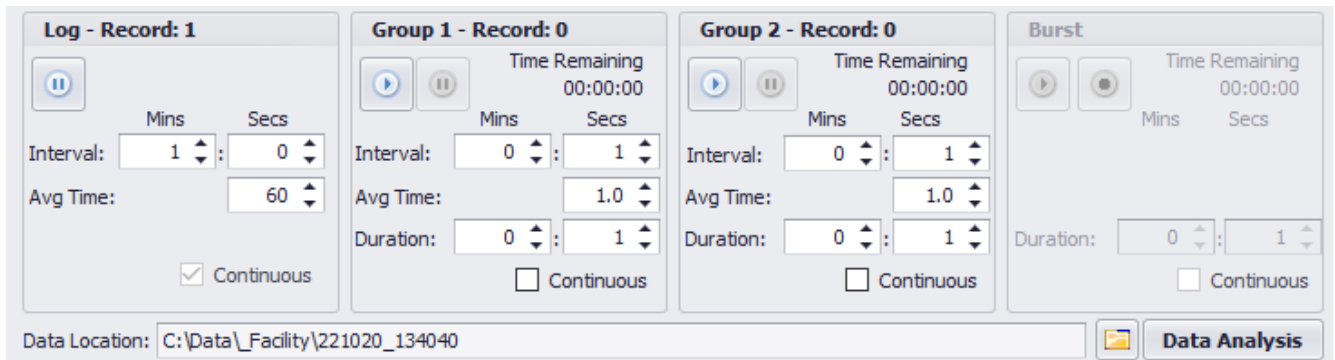


Gauge

System Alarms Editor

Test SLATE data storage.

Test SLATE can support multiple data files that are configured independently. Each data file can store data at different rates, durations, and averaging times. One data file is used as a log file and contains data from the total test length. This allows the capture of data independent of the sequence that is running. There is also a burst data file that can be triggered to take a high rate of data for a specified period of time. This may happen many times during a test. There are also regular data files that are configured and defined independently to allow the flexibility of taking different sets of data at differing rates. All test data is stored in a binary format for maximum speed; each data sample is time-stamped. After the data acquisition period is elapsed, the binary data will be processed, time-aligned, stored into a file, and made available to the local archive import/export software. Data storage can be initiated automatically by a test sequence based on conditional criteria or initiated manually by a user clicking a button on the test execution display.



Manual data storage during testing