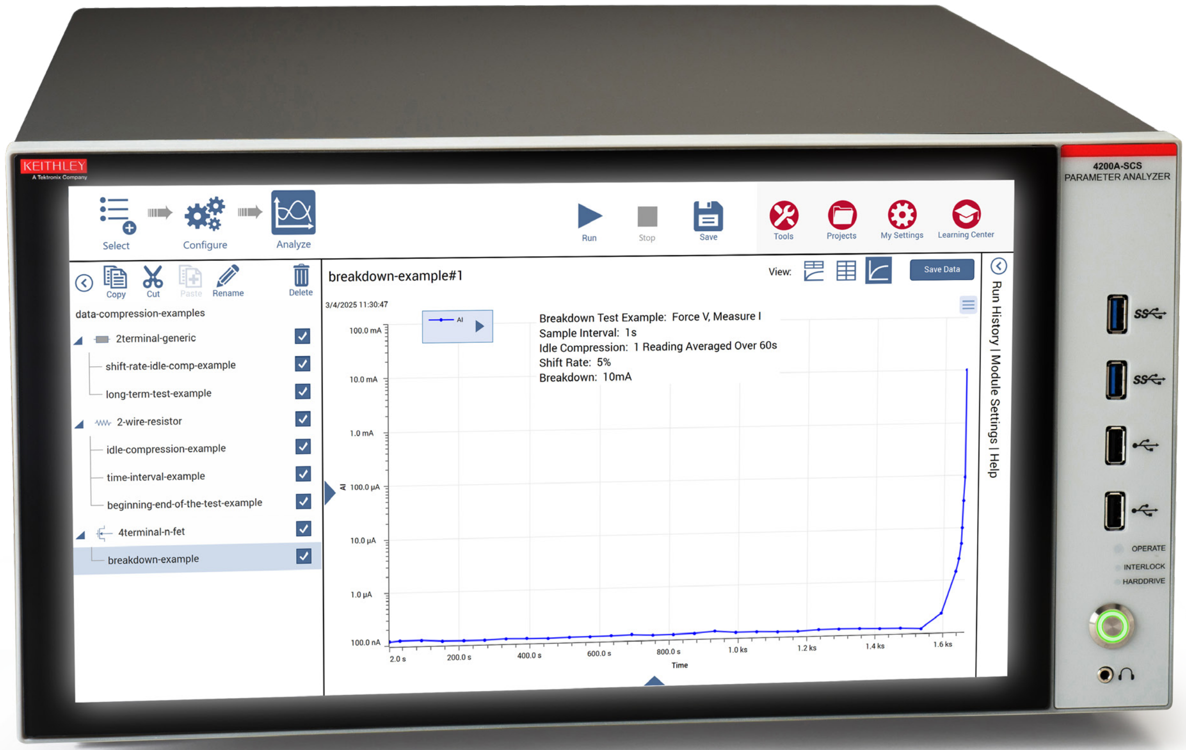
Long-Term Data Collection and Breakdown Testing Using Data Compression in the Keithley 4200A-SCS Parameter Analyzer

APPLICATION NOTE



Breakdown Test Example: Force V, Measure I
Sample Interval: 1s
Idle Compression: 1 Reading Averaged Over 60s
Shift Rate: 5%
Breakdown: 10mA

Introduction

The Keithley 4200A-SCS Parameter Analyzer, combined with the built-in easy-to-use interactive Clarius Software, is an essential tool for semiconductor device testing. The 4200A-SCS is a customizable and fully integrated parameter measurement system that enables current-voltage (I-V), capacitance-voltage (C-V) and ultra-fast pulsed I-V characterization. Some semiconductor test applications, such as breakdown tests, require long-term data collection. However, most test and measurement software are limited to a certain number of data points in a test. Managing the returned data in long-term tests can be challenging. Ideally, only the most relevant data should be returned instead of all the data.

Introduced in Clarius V1.14, the Data Compression feature of the 4200A-SCS Parameter Analyzer addresses this need by allowing users to select important data from areas of interest while compressing, or reducing, the amount of less significant information. This capability greatly reduces the number of data points measured and returned by allowing the user to define exactly which measurements to keep by defining data compression rules.

This application note provides an overview on data compression, explains how each rule type works, shows how to configure the built-in SMU compressed data library tests and guides users on how to create a new user module or modify an existing user module to customize data compression.

Data Compression Rules and Examples

Long-term tests, such as breakdown tests or monitoring the response of a sensor, can run for days and generate hundreds of thousands of data points. However, only the crucial data points, such as those near a breakdown, need to be returned. The Clarius Software user test modules (UTMs) are limited to a maximum of 65,535 data points. The Data Compression feature of the 4200A-SCS addresses this need by allowing users to select and preserve important data.

Data compression is controlled by configuring rules that determine how data is compressed and when uncompressed data is returned. There are five types of rules for data compression, with a limit of ten total compression rules that can be used in a particular test.

1. **Shift Rate:** This rule compares data points to previously returned data. If the change ratio of a selected data series at one point exceeds the shift rate percentage compared to the previously returned data, the data is returned. Multiple shift rate rules can be set.
2. **Idle Compression:** This rule applies compression based on time or number of samples and returns one single averaged data point. This rule compresses data when no other rules are active. For example, if there are multiple shift rate rules, this rule compresses data when there is no percent change in the data returned. Only one idle compression rule can be set. Idle Compression is often used with the other rules if compression is not enabled in those rules.
3. **Beginning of the Test:** This rule enables or disables compressed data in the initial part of the test. The beginning section can be defined by either a duration in number of seconds or by the number of samples. Only one beginning of the test rule can be set.
4. **End of the Test:** This rule enables or disables compressed data at the end of a test. The end part of the test can be defined by either a duration in number of seconds or by the number of samples. Only one end of the test rule can be set.
5. **Time Interval:** This rule compresses data within a defined time. The time can be set by defining the start and stop times in HH:MM:SS format. Multiple time interval rules can be set.

The Data Compression rules are best shown through illustrations. Multiple rules can be combined in a test. The next section shows three examples that illustrate the Shift Rate and Idle Compression, Beginning of the Test and End of the Test, and Time Interval rules.

The first example, shown in **Figure 1**, combines the Shift Rate and Idle Compression rules. This is a typical use case for long-term tests that are waiting for an event to occur, such as breakdown, where more data points are needed during a transition period. In this test, the total test time is 120 s and the sample interval is 100 ms. The Shift Rate is defined to occur when there is a 10% change to the output. The Idle Compression is set to a duration of 10 s, which means that one data point is the average of readings taken at 100 ms sample rate for 10 s.

The next example shows compressed data at only the beginning and end of a test by using the Beginning of the Test and End of the Test rules, as shown in **Figure 2**. The test is configured to run for 80 s with a 1 s interval time. Both the Beginning of the Test and End of the Test rules are set to apply for a duration of 30 s to average 10 samples.

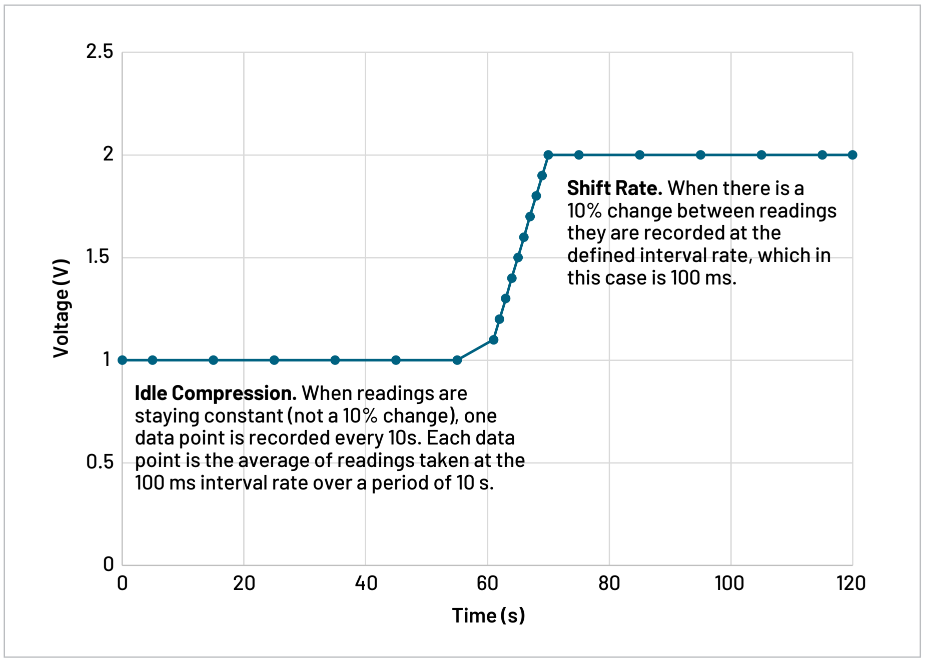


Figure 1. Example data taken with Shift Rate and Idle Compression rules.

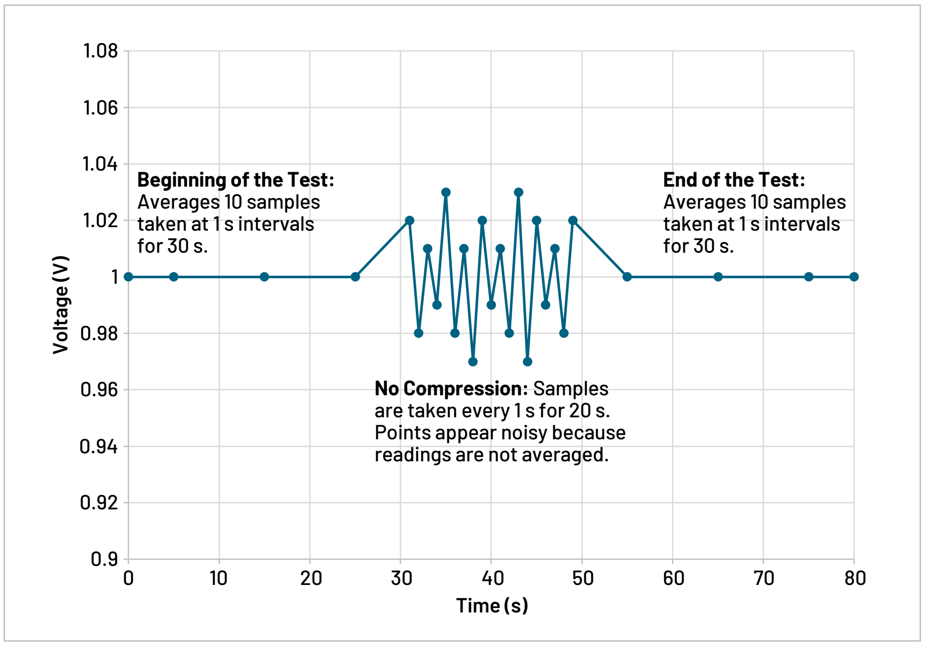


Figure 2. Example data taken with Beginning of the Test and End of the Test compression rules.

The final example shows how the Time Interval rule can compress data. The data is compressed between two specified times in the format hh:mm:ss. In the graph shown in **Figure 3**, the total test time is 60 s, and the sample interval time is set to 2 s. The data is compressed between time=10 s (00:00:10) and time=50 s (00:00:50). During this time, the data is compressed by an average of 5 samples.

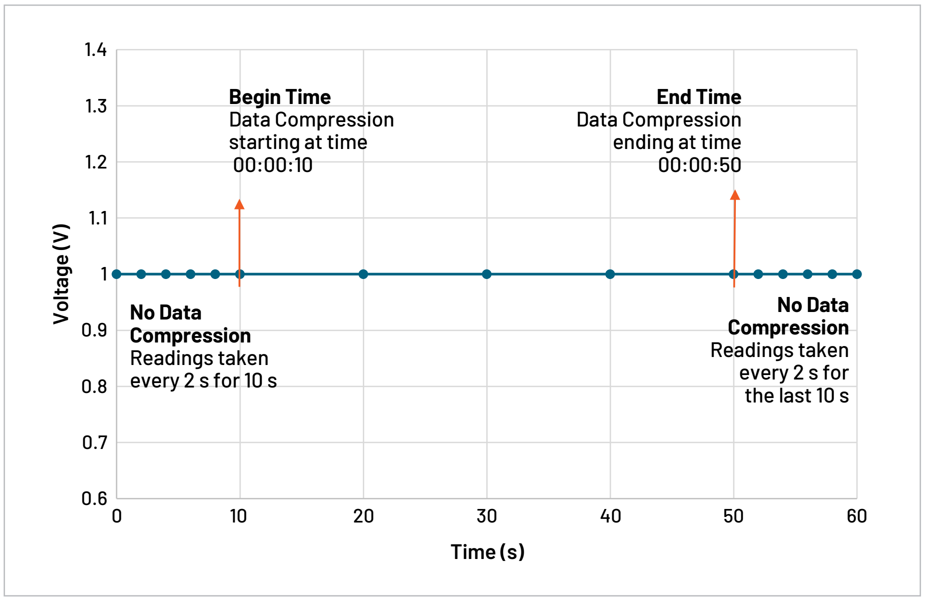


Figure 3. Example data taken with Time Interval rule.

Using Clarius Library Tests for Data Compression

The comprehensive built-in library of the 4200A-SCS includes tests that demonstrate using the rules for data compression and can be modified to fit other applications. These tests are listed in the project tree of the ***data-compression-examples*** project, shown in **Figure 4**. The tests in the project can also be found individually in the Test Library and are further explained in the following paragraphs. The tests and project can be found in the Clarius Library by searching "data compression" in the search bar in the Select view.

These tests were generated using the ***ConstantVoltage*** and ***ConstantCurrent*** user modules in the ***CompressedAcquisitionUlib*** user library.

By selecting one of the tests in the Project Tree, the force and measure settings can be updated in the Configure View, as shown in **Figure 5**. Each of the tests enables up to four SMUs to bias either a current or a voltage, depending on the test. In the Test Settings tab, both the measure range and timing settings are configured. The timing settings include the Total Test Time, Sample Interval, Hold Time and PLC (power line cycles).

The data compression rules for each of these tests can be configured by selecting Compressed Data, which opens the Compressed Data Acquisition Settings dialog, as shown in **Figure 6**. From this dialog, one of the rules can be selected and configured. Configuration of the five rules is provided in the next section.

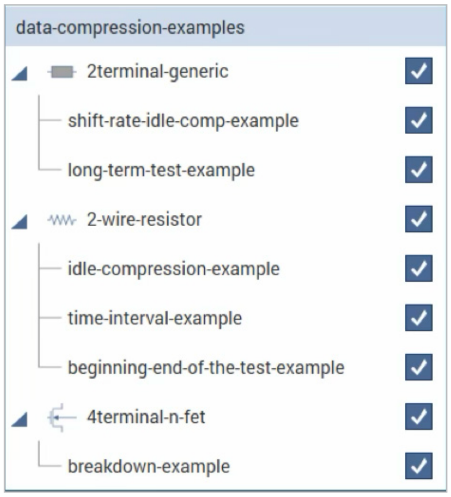
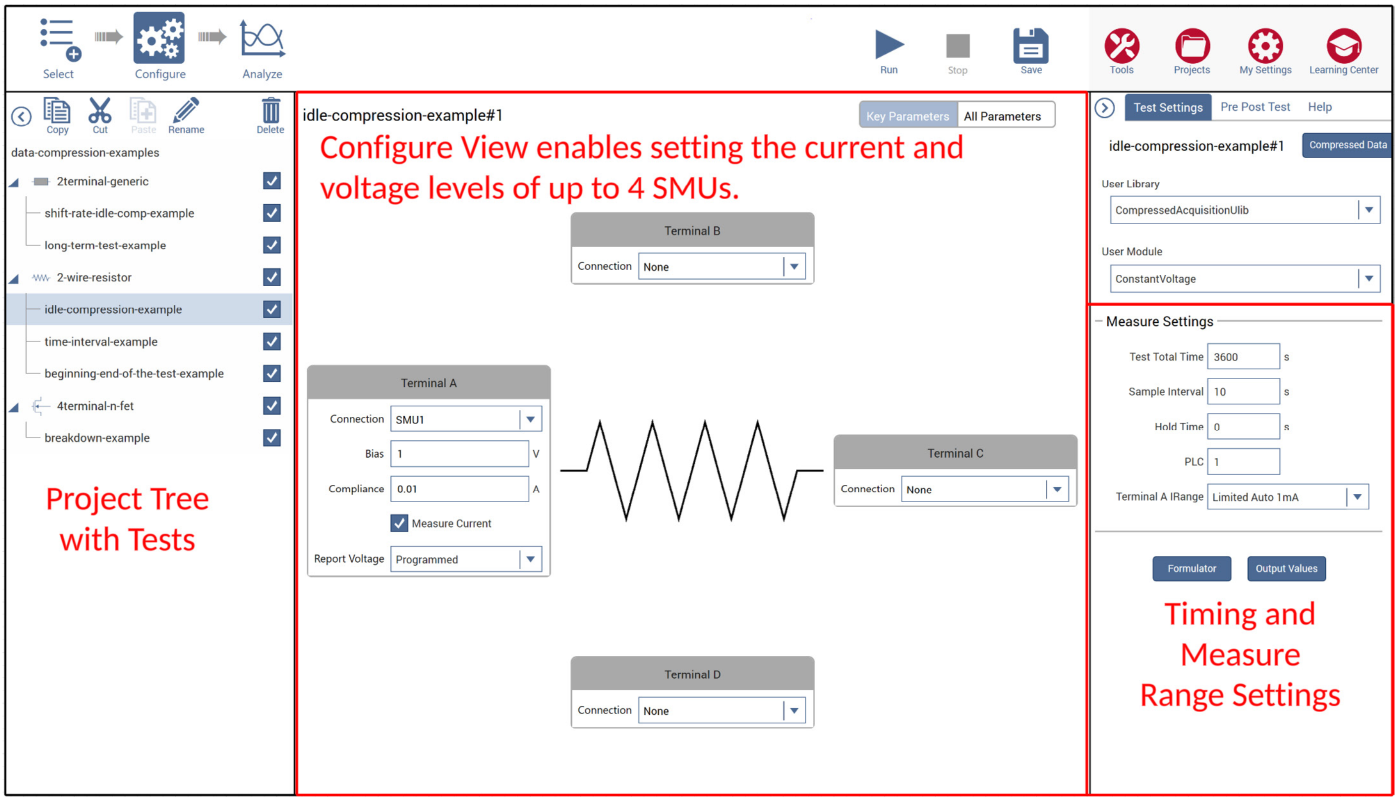

data-compression-examples
- 2terminal-generic ✓
  - shift-rate-idle-comp-example ✓
  - long-term-test-example ✓
- 2-wire-resistor ✓
  - idle-compression-example ✓
  - time-interval-example ✓
  - beginning-end-of-the-test-example ✓
- 4terminal-n-fet ✓
  - breakdown-example ✓

Figure 4. Project Tree of *data-compression-examples* project.



Project tree: data-compression-examples
- 2terminal-generic
  - shift-rate-idle-comp-example
  - long-term-test-example
- 2-wire-resistor
  - idle-compression-example
  - time-interval-example
  - beginning-end-of-the-test-example
- 4terminal-n-fet
  - breakdown-example

Project Tree with Tests

idle-compression-example#1

Configure View enables setting the current and voltage levels of up to 4 SMUs.

Terminal B — Connection: None

Terminal A — Connection: SMU1; Bias: 1 V; Compliance: 0.01 A; ✓ Measure Current; Report Voltage: Programmed

Terminal C — Connection: None

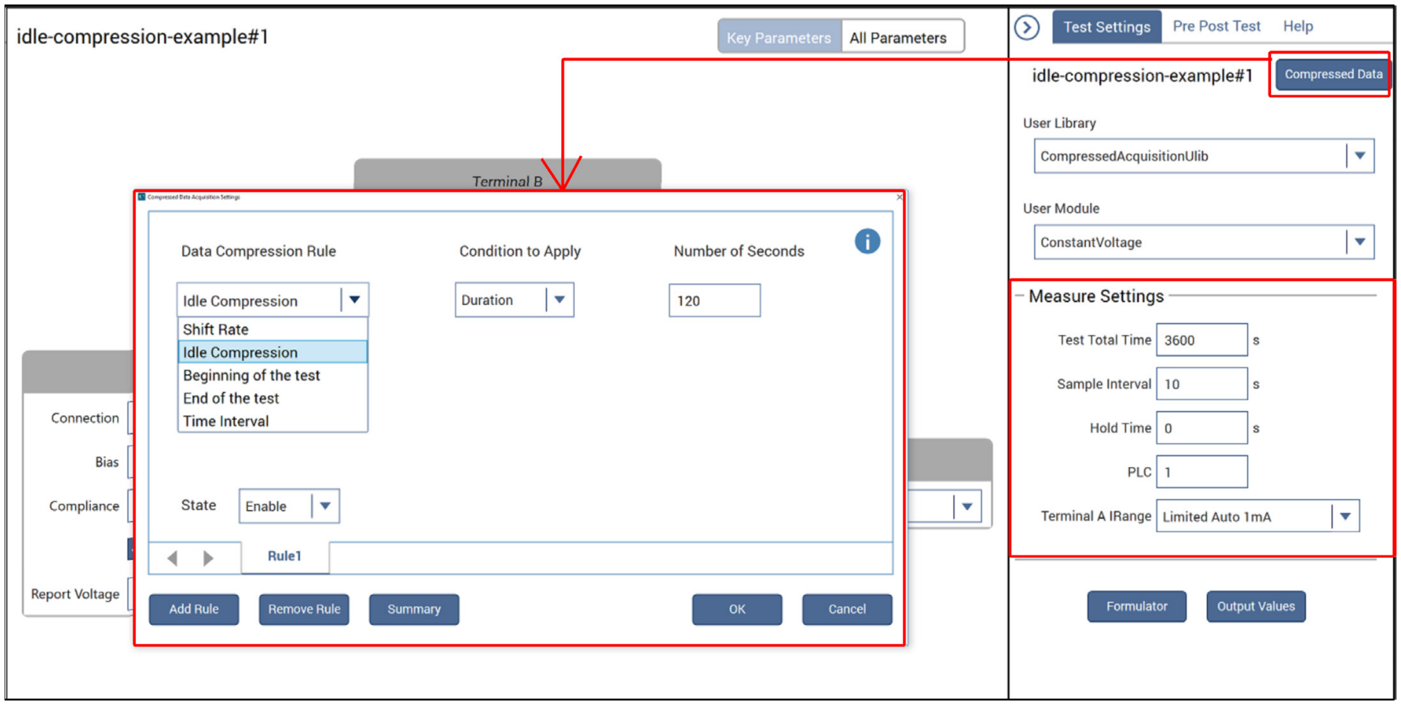
Terminal D — Connection: None

idle-compression-example#1
User Library: CompressedAcquisitionUlib
User Module: ConstantVoltage

Measure Settings
- Test Total Time: 3600 s
- Sample Interval: 10 s
- Hold Time: 0 s
- PLC: 1
- Terminal A IRange: Limited Auto 1mA

Timing and Measure Range Settings

Figure 5. Configure view of a data compression test.



idle-compression-example#1

Terminal B

Data Compression Rule	Condition to Apply	Number of Seconds
Idle Compression	Duration	120

Data Compression Rule options: Shift Rate, Idle Compression, Beginning of the test, End of the test, Time Interval

State: Enable

Connection / Bias / Compliance / Report Voltage

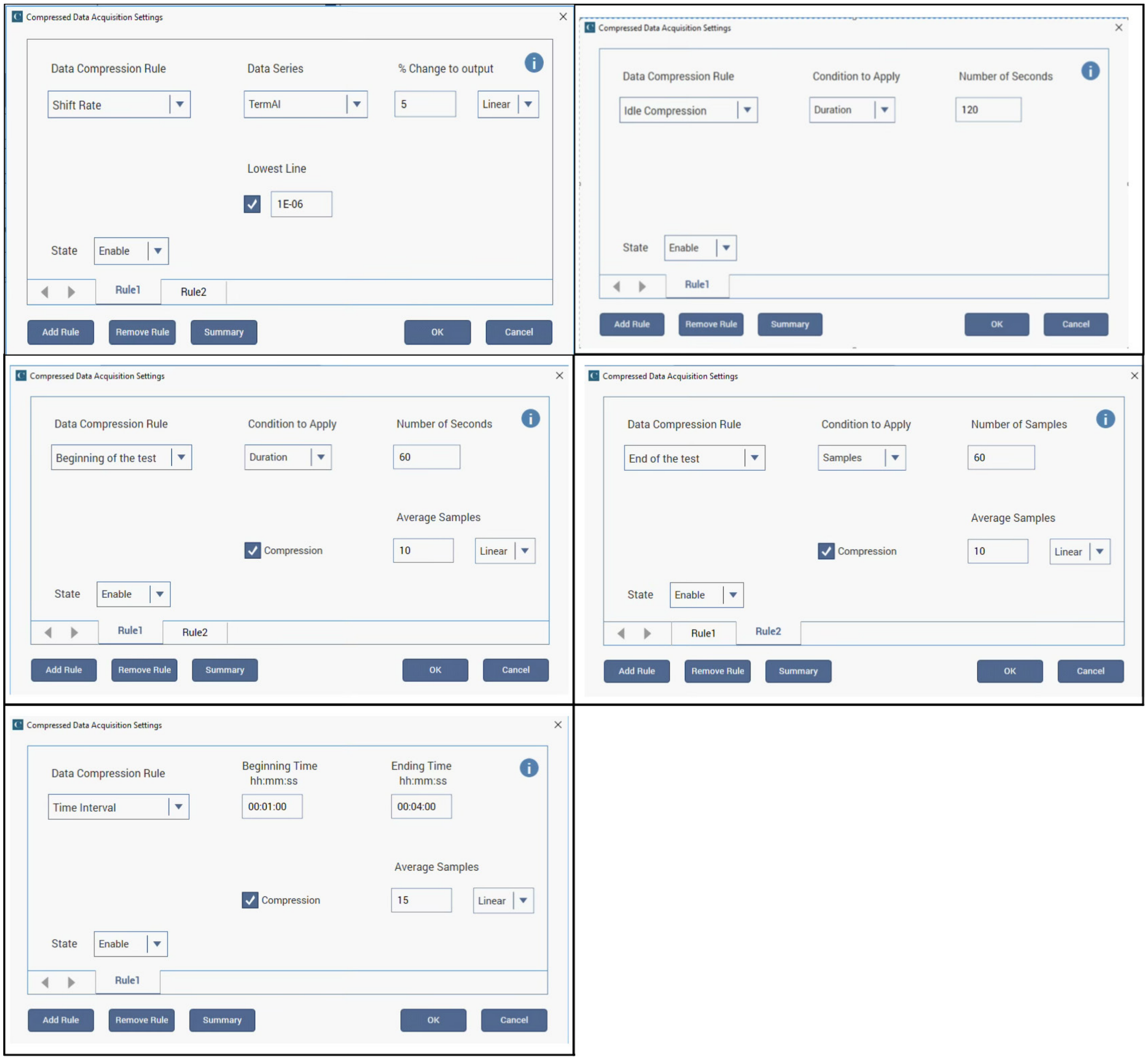
idle-compression-example#1
User Library: CompressedAcquisitionUlib
User Module: ConstantVoltage

Measure Settings
- Test Total Time: 3600 s
- Sample Interval: 10 s
- Hold Time: 0 s
- PLC: 1
- Terminal A IRange: Limited Auto 1mA

Figure 6. Compressed Data Acquisition Settings dialog showing compression rules.

Data Compression Rule Settings

Figure 7 shows the Compressed Data Acquisition Settings dialogs for each of the Data Compression rules. Up to ten rules can be added to each test. If multiple rules are set in one test, the order of rule hierarchy is End of the Test, Beginning of the Test, Time Interval, Shift Rate and Idle Compression. The settings for each of the rules are further explained in the following paragraphs.



Compressed Data Acquisition Settings

Data Compression Rule: Shift Rate
Data Series: TermAI
% Change to output: 5 Linear
Lowest Line: ✓ 1E-06
State: Enable
Rule1 Rule2

Compressed Data Acquisition Settings

Data Compression Rule: Idle Compression
Condition to Apply: Duration
Number of Seconds: 120
State: Enable
Rule1

Compressed Data Acquisition Settings

Data Compression Rule: Beginning of the test
Condition to Apply: Duration
Number of Seconds: 60
✓ Compression
Average Samples: 10 Linear
State: Enable
Rule1 Rule2

Compressed Data Acquisition Settings

Data Compression Rule: End of the test
Condition to Apply: Samples
Number of Samples: 60
✓ Compression
Average Samples: 10 Linear
State: Enable
Rule1 Rule2

Compressed Data Acquisition Settings

Data Compression Rule: Time Interval
Beginning Time hh:mm:ss: 00:01:00
Ending Time hh:mm:ss: 00:04:00
✓ Compression
Average Samples: 15 Linear
State: Enable
Rule1

Figure 7. Compressed Data Acquisition Settings dialogs for each of the Data Compression rules.

Shift Rate

In the Shift Rate dialog, a **Data Series** is selected and a **% Change to output** text box is used to enter a percentage from 0.1 to 100 percent as a linear or log value. A **Lowest Line** value can be set to prevent zero noise value changes. When selected, values below the defined value are excluded from the shift rate calculation.

Idle Compression

The **Condition to Apply** of Idle Compression can be selected as either **Samples** or **Duration**.

When **Samples** are selected, the **Number of Samples** are averaged and returned as a single data point. From 1 to 1 e6 samples can be averaged. The readings are taken at the Sample Interval rate in Measure Settings.

When **Duration** is selected, the time interval is chosen in units of **Number of Seconds** from 0 to 3600 s. The average result of the samples within the time interval is returned as a single data point.

Beginning of the Test

The Beginning of the Test rule can be configured based on **Samples** or **Time**, as selected from the **Condition to Apply** list.

The **Compression** checkbox allows users to set the number of **Average Samples** to average down to a single returned data point. If **Compression** is cleared, then no compression is applied during the beginning of the test condition.

End of the Test

The End of the Test rule can be configured based on **Samples** or **Time**, selected from the **Condition to Apply** list.

The **Compression** checkbox allows users to set the number of **Average Samples** to average down to a single returned data point. If **Compression** is cleared, then no compression is applied during the end of the test condition.

Time Interval

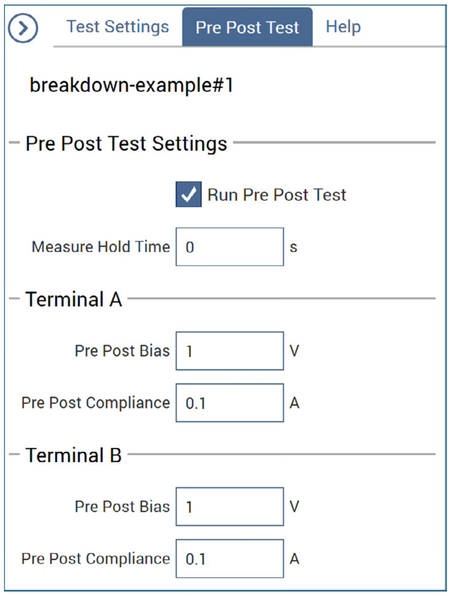
When defining the Time Interval rule, the **Beginning Time** and **Ending Time** text boxes are used to define these times in the HH:MM:SS format.

The **Compression** checkbox allows users to set the number of **Average Samples** to average down to a single returned data point. If **Compression** is cleared, then no compression is applied during the defined time interval.

Timing Settings

Timing settings for the tests are in Measure Settings under the Test Settings tab in the Configure view. These settings are Test Total Time, Sample Interval and Hold Time, as illustrated in **Figure 8**.

The Pre-Test and Post-Test Measure Hold Time is configured in the Pre Post Test tab in the Configure view, as shown in **Figure 9**. The Pre Post Test provides an additional voltage or current bias at the beginning and end of the test and measures the resulting current or voltage. This can be used for breakdown tests or to verify test connections.



breakdown-example#1

Pre Post Test Settings

[✓] Run Pre Post Test

Measure Hold Time	0	s

Terminal A

Pre Post Bias	1	V
Pre Post Compliance	0.1	A

Terminal B

Pre Post Bias	1	V
Pre Post Compliance	0.1	A

Figure 9. Pre Post Test tab settings.

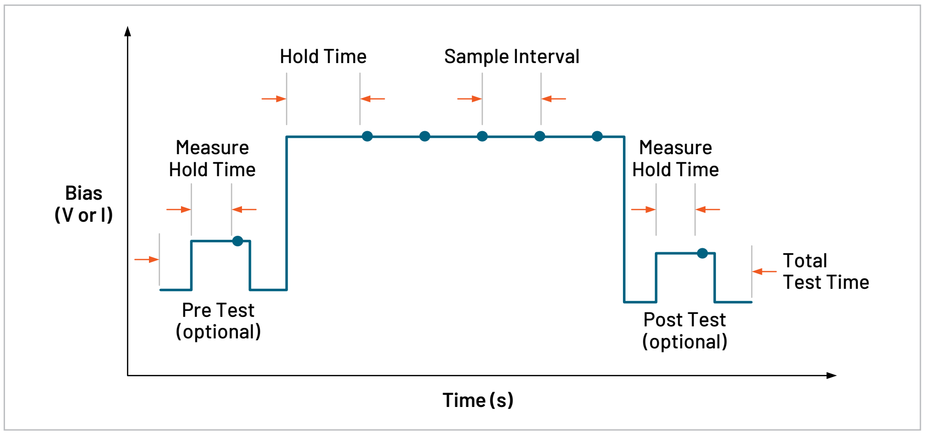


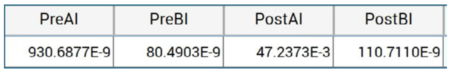
Figure 8. Timing diagram for data compression factory tests.

Output Parameters

Once a test is executed, the time and measured current and voltage from each SMU are returned to the Sheet in the Analyze view. In addition to these parameters, the Pre- and Post-Test and Breakdown parameters are returned.

Pre- and Post-Test Parameters

If the **Run Pre Post Test** box is selected, pre-test and post-test measurements are taken by each SMU (A, B, C and D) configured in the test. **Figure 10** shows the results of the pre-test and post-test measured current from SMUs A and B.

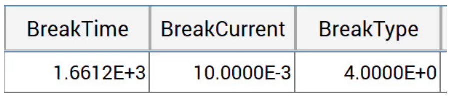


PreAI	PreBI	PostAI	PostBI
930.6877E-9	80.4903E-9	47.2373E-3	110.7110E-9

Figure 10. Pre-test and Post-Test Data in Sheet Analyze View.

Breakdown Parameters

Breakdown parameters are automatically returned to the Sheet. These include the BreakTime, BreakCurrent (or BreakVoltage) and BreakType, as shown in **Figure 11**.



BreakTime	BreakCurrent	BreakType
1.6612E+3	10.0000E-3	4.0000E+0

Figure 11. Breakdown Parameters in Sheet Analyze View.

The BreakTime is the time when the SMU went into compliance. The BreakCurrent (or BreakVoltage) is the value of the current or voltage reading at compliance.

The BreakType is a number from 0 to 5 that indicates how the condition of the test ended. These conditions are derived from JEDEC standard JESD263. They are described as follows:

0 – All passed, no compliance reached.

1 – Pre-test compliance reached.

2 – Catastrophic failure: Compliance reached during the test and post-test.

3 – Masked Catastrophic failure: Compliance reached during the post-test.

4 – Non-Catastrophic failure: Compliance reached during the test but not in pre-test or post-test.

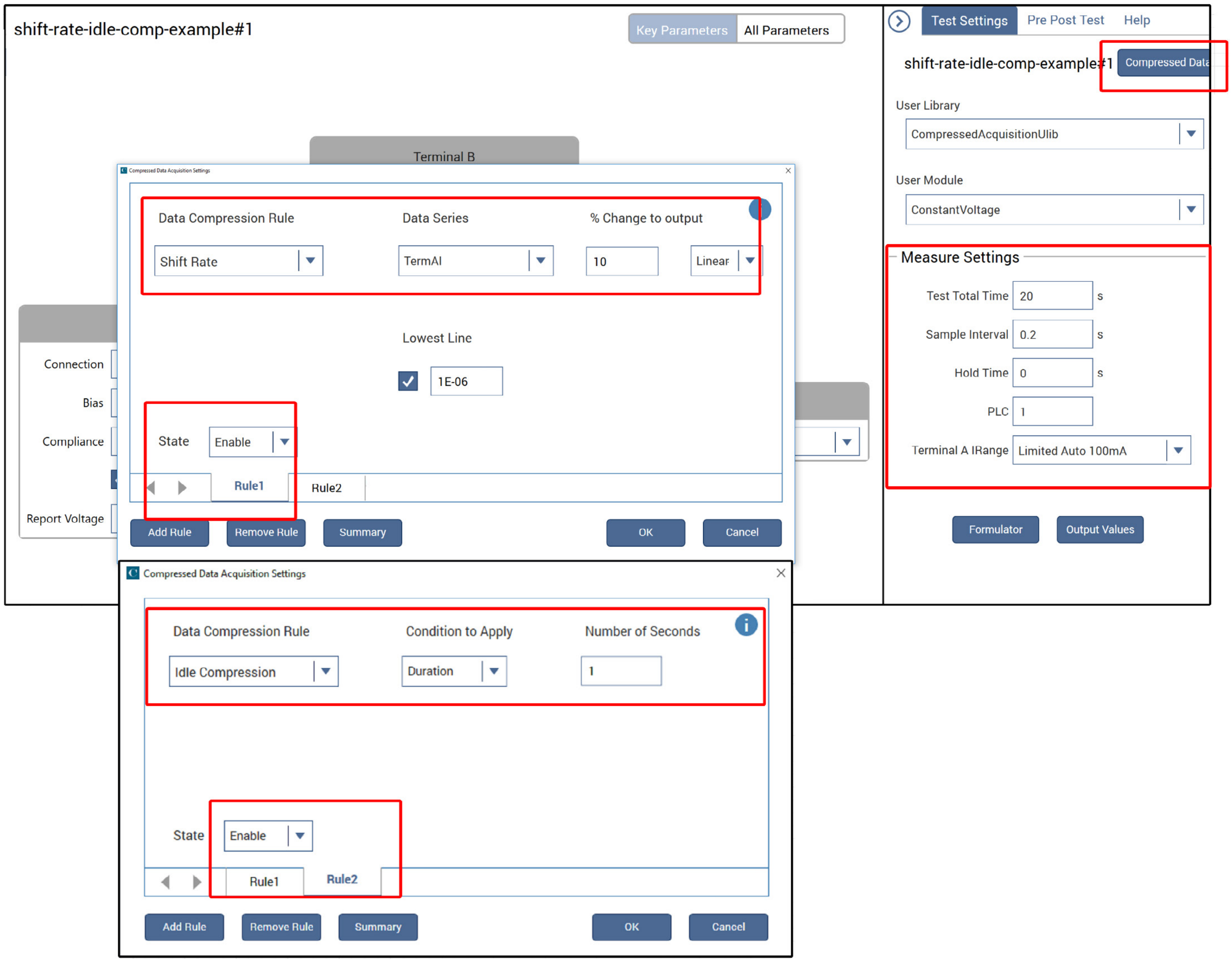
5 – Monitor failed: Compliance reached during the test; post-test was disabled.

Data Compression Library Test Descriptions

The Data Compression Rules are applied in the library tests. One or more rules are used in each test that is described in the following paragraphs. Readings from a compressed test may be delayed due to the time required to calculate the compression rules.

shift-rate-idle-comp-example Test

The ***shift-rate-idle-comp-example*** test forces 0 V and measures the current and time from an applied square wave. This test uses both the Shift Rate and Idle Compression rules to measure selected data points of a repeated square wave with defined rise and fall times. **Figure 12** shows the Configure view of this test, which has two rules defined.



shift-rate-idle-comp-example#1

User Library: CompressedAcquisitionUlib

User Module: ConstantVoltage

Measure Settings

Test Total Time	20	s
Sample Interval	0.2	s
Hold Time	0	s
PLC	1	
Terminal A IRange	Limited Auto 100mA	

Compressed Data Acquisition Settings (Rule1)

Data Compression Rule	Data Series	% Change to output	
Shift Rate	TermAI	10	Linear

Lowest Line: ✓ 1E-06

State: Enable

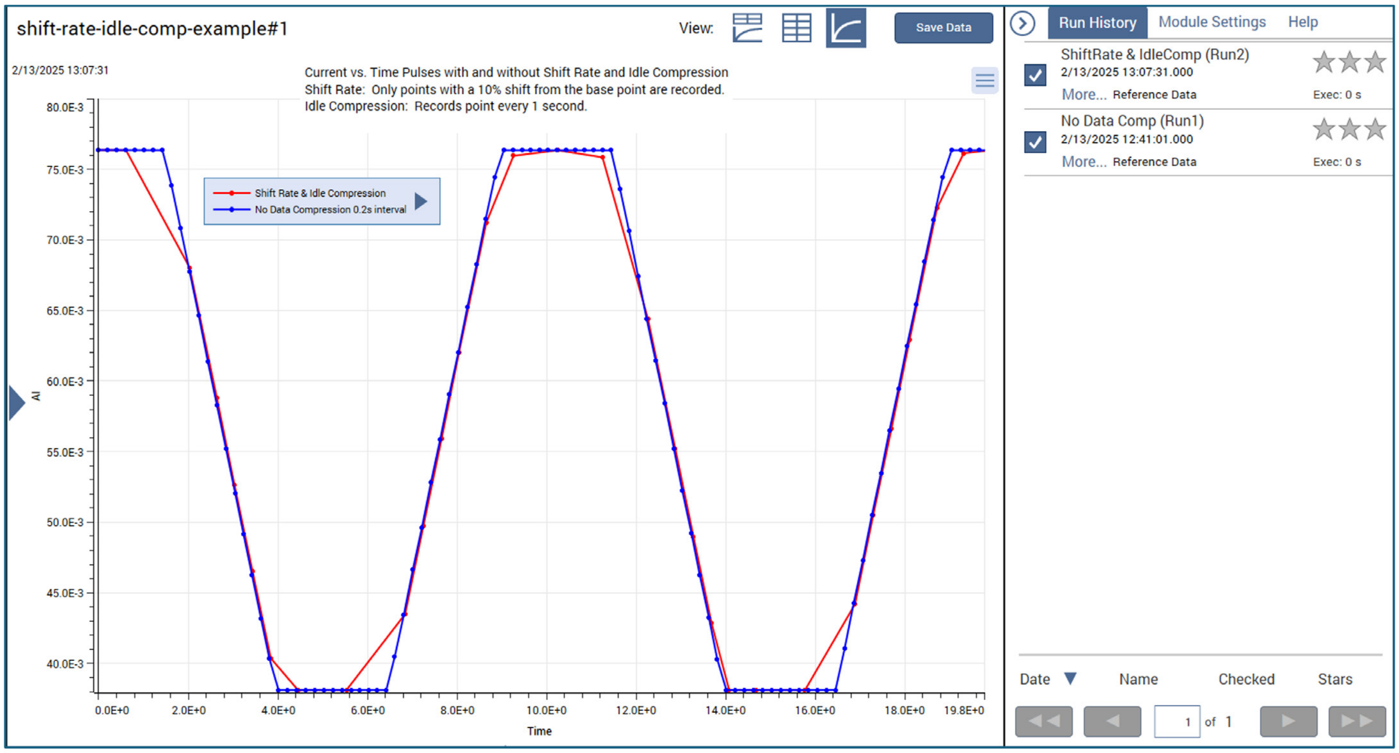
Compressed Data Acquisition Settings (Rule2)

Data Compression Rule	Condition to Apply	Number of Seconds
Idle Compression	Duration	1

State: Enable

Figure 12. Configure view of *shift-rate-idle-comp-example* test with Shift Rate and Idle Compression rules.

This test was executed two times: Run1 with no data compression and Run2 with both Shift Rate and Idle Compression. In **Figure 13**, both Run1 (blue graph with no data compression) and Run2 (red graph with Idle Compression and Shift Rate) are plotted. For the Shift Rate, there are data points only during the rise and fall times because on the flat parts of the square wave the bias is constant so there is no change in the data. For the Shift Rate, data will be taken at the sample rate of 200 ms if there is a 10 percent change in data. For data that isn't shifting 10 percent, data will be recorded every 1 s (idle compression).


shift-rate-idle-comp-example#1

2/13/2025 13:07:31

Current vs. Time Pulses with and without Shift Rate and Idle Compression
Shift Rate: Only points with a 10% shift from the base point are recorded.
Idle Compression: Records point every 1 second.

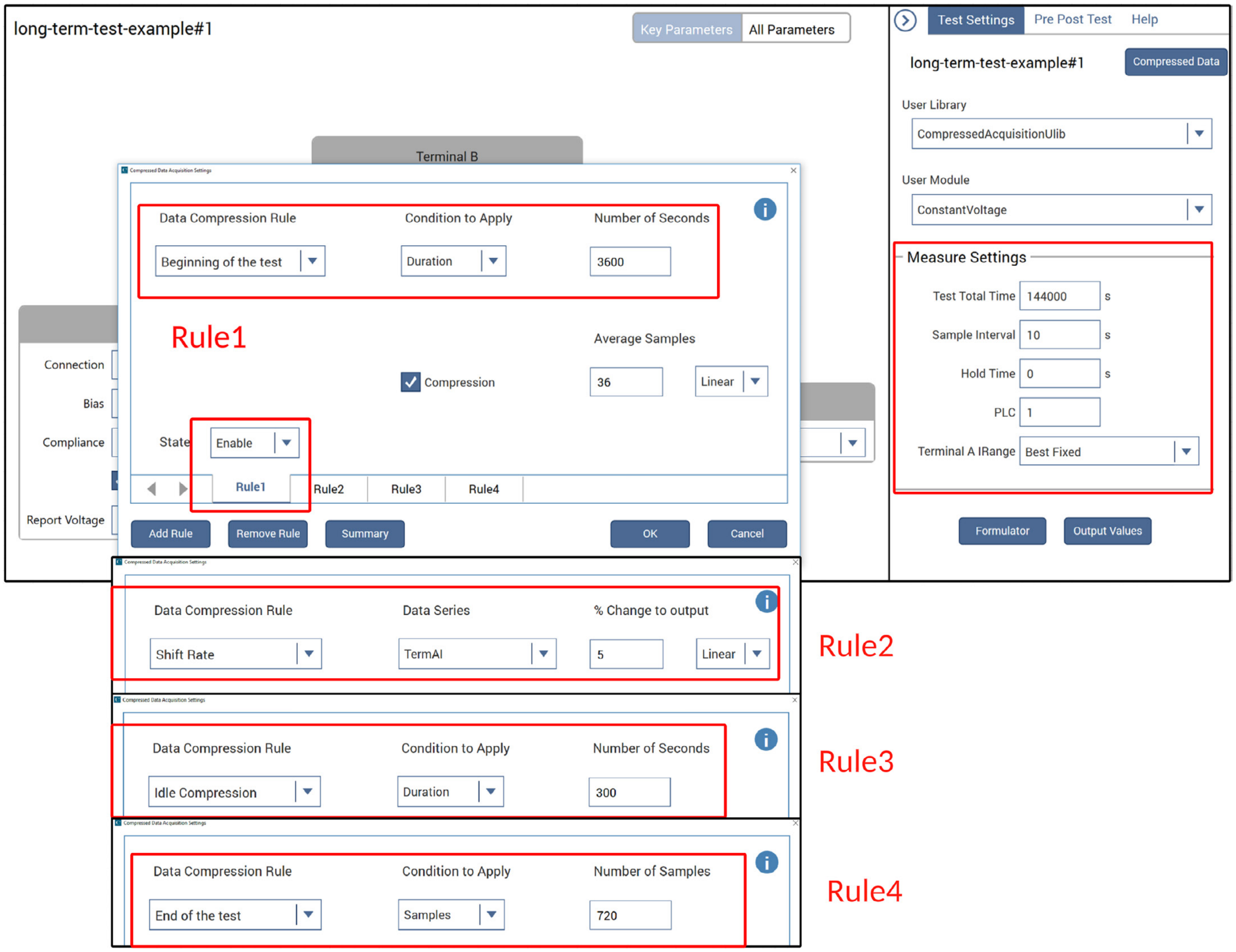
	Name	Date	Exec
✓	ShiftRate & IdleComp (Run2)	2/13/2025 13:07:31.000	Exec: 0 s
✓	No Data Comp (Run1)	2/13/2025 12:41:01.000	Exec: 0 s

Figure 13. Graphical results of both uncompressed data (blue curve) and shift rate with idle compression data (red curve) of time delayed square waves using the ***shift-rate-idle-comp-example*** test.

long-term-test-example Test

In the ***long-term-test-example*** test, SMU1 outputs 0 V and measures the current output of a device that randomly outputs a 20 s pulse every 1 to 3 hours over a 40-hour (144000 s) period. The sample interval is 10 s. As shown in **Figure 14**, this test uses the data compression rules Beginning of the Test, Shift Rate, Idle Compression and End of the Test.

The Beginning of the Test rule compresses the first hour of data, averaging one point every 36 samples. The Shift Rate rule only captures data if there is a 5 percent change in data. An Idle Compression rule returns an averaged reading every 300 s. The End of the Test rule is for the last 720 samples. Every 36 samples are averaged to one point.



long-term-test-example#1

User Library: CompressedAcquisitionUlib
User Module: ConstantVoltage

Measure Settings
Test Total Time: 144000 s
Sample Interval: 10 s
Hold Time: 0 s
PLC: 1
Terminal A IRange: Best Fixed

Rule1 — Data Compression Rule: Beginning of the test; Condition to Apply: Duration; Number of Seconds: 3600; Compression ✓; Average Samples: 36, Linear; State: Enable

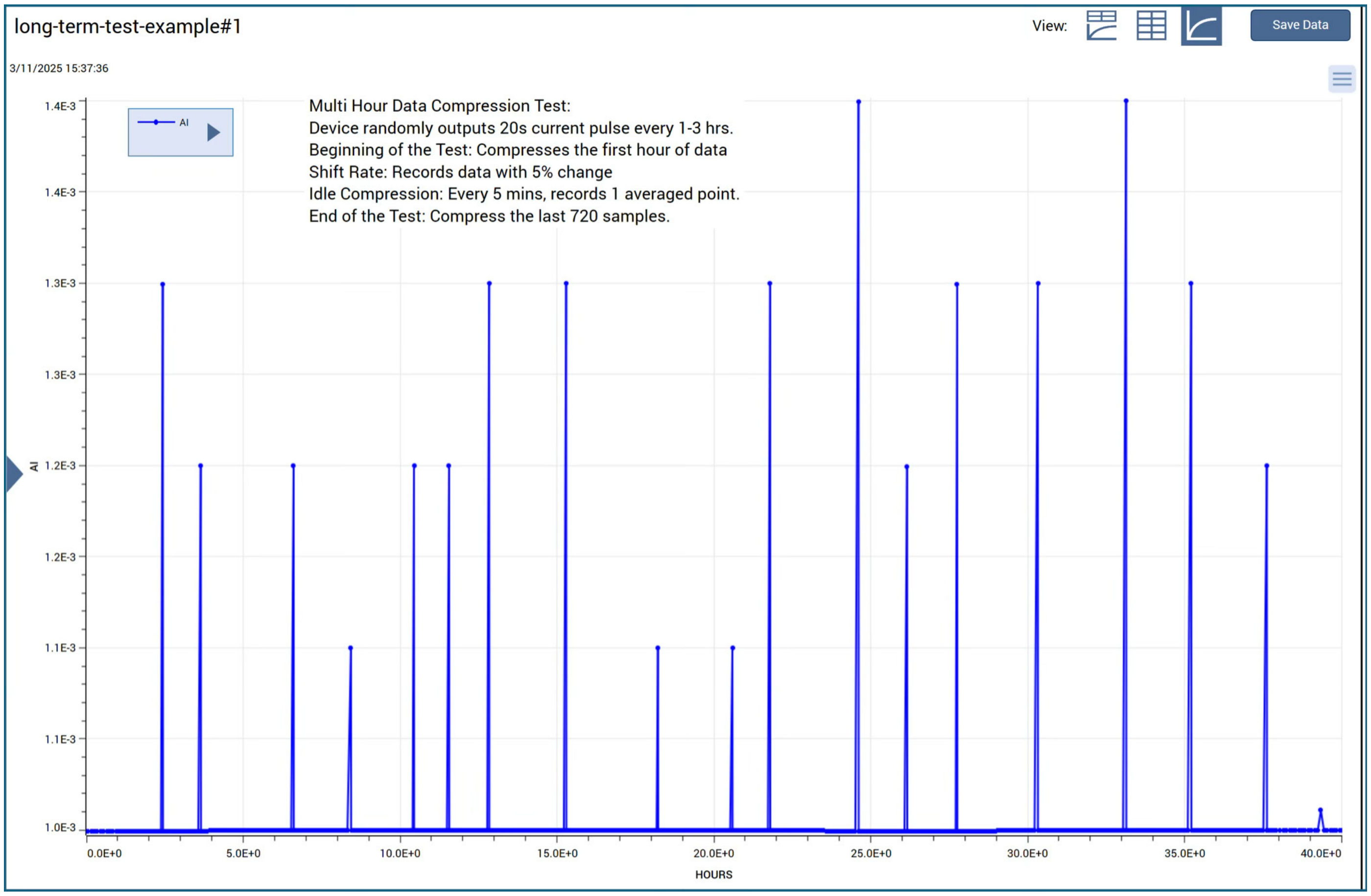
Rule2 — Data Compression Rule: Shift Rate; Data Series: TermAI; % Change to output: 5, Linear

Rule3 — Data Compression Rule: Idle Compression; Condition to Apply: Duration; Number of Seconds: 300

Rule4 — Data Compression Rule: End of the test; Condition to Apply: Samples; Number of Samples: 720

Figure 14. Configure view of *long-term-test-example* showing four rules.

When this 40-hour test is executed and graphed, the results are displayed as shown in **Figure 15** showing the random current peaks.



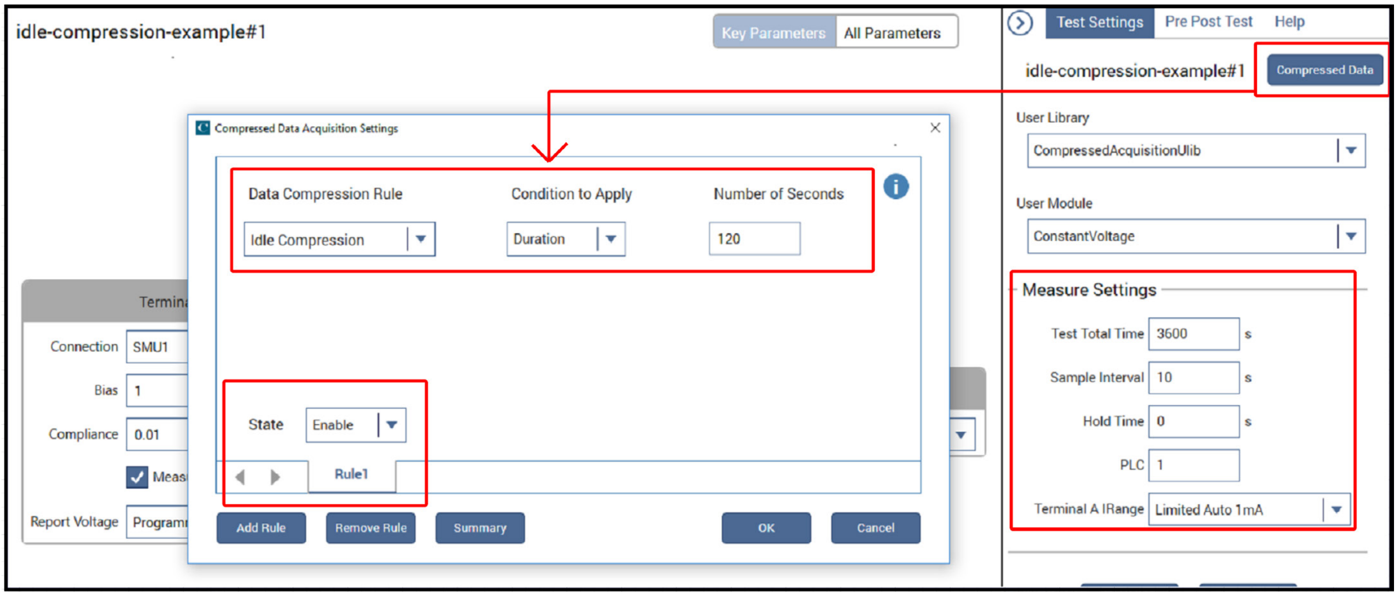
Multi Hour Data Compression Test:
Device randomly outputs 20s current pulse every 1-3 hrs.
Beginning of the Test: Compresses the first hour of data
Shift Rate: Records data with 5% change
Idle Compression: Every 5 mins, records 1 averaged point.
End of the Test: Compress the last 720 samples.

Figure 15. Graphical results of *long-term-test-example* test.

idle-compression-example Test

In the ***idle-compression-example*** test, SMU1 outputs a 1 V bias across a 2 kΩ resistor and the resulting current is measured with a sample interval of 10 s for a total test time of 3600 s (1 hour). **Figure 16** shows these Measure Settings in the Configure view along with the enabled Idle Compression rule.

In this example, one reading is returned to the Sheet every 120 s. This reading is the average of readings taken every 10 s (Sample Interval).



Data Compression Rule: Idle Compression
Condition to Apply: Duration
Number of Seconds: 120
State: Enable

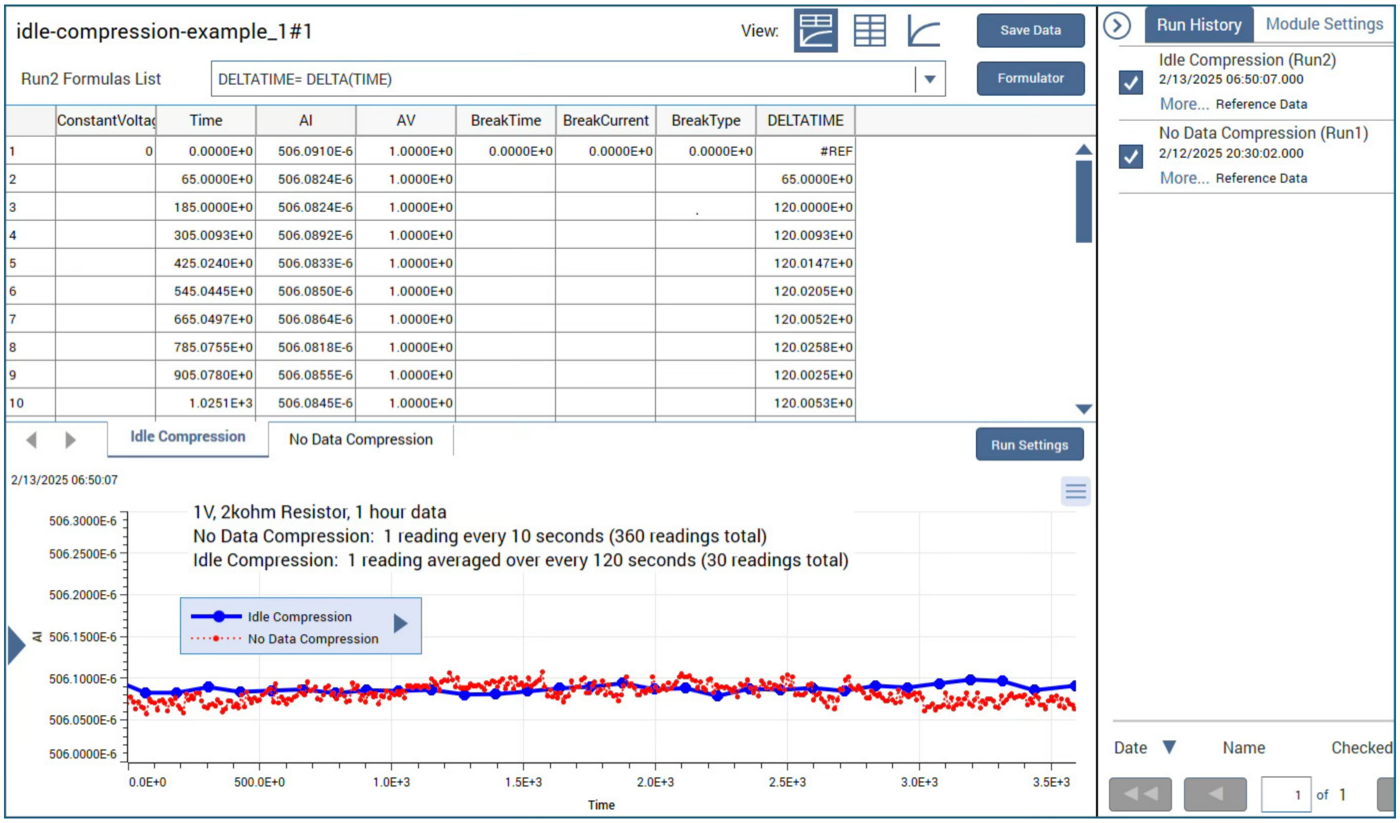
Connection: SMU1
Bias: 1
Compliance: 0.01
Report Voltage: Program

User Library: CompressedAcquisitionUlib
User Module: ConstantVoltage

Measure Settings
Test Total Time: 3600 s
Sample Interval: 10 s
Hold Time: 0 s
PLC: 1
Terminal A IRange: Limited Auto 1mA

Figure 16. Configure view of ***idle-compression-example*** test with the Idle Compression rule.

To show the effects of Idle Compression, this test was run two times, with and without idle compression enabled, as shown in **Figure 17**. In the Analyze view graph, Run1 (red curve) shows all the data points (3600/10=360 points). Run2 (blue curve) shows only the compressed data (3600/120=30 points). Notice that Run2 is much smoother because the readings are averaged. The last column in the Sheet shows the delta time of the readings.



idle-compression-example_1#1

Run2 Formulas List: DELTATIME= DELTA(TIME)

	ConstantVoltag	Time	AI	AV	BreakTime	BreakCurrent	BreakType	DELTATIME
1	0	0.0000E+0	506.0910E-6	1.0000E+0	0.0000E+0	0.0000E+0	0.0000E+0	#REF
2		65.0000E+0	506.0824E-6	1.0000E+0				65.0000E+0
3		185.0000E+0	506.0824E-6	1.0000E+0				120.0000E+0
4		305.0093E+0	506.0892E-6	1.0000E+0				120.0093E+0
5		425.0240E+0	506.0833E-6	1.0000E+0				120.0147E+0
6		545.0445E+0	506.0850E-6	1.0000E+0				120.0205E+0
7		665.0497E+0	506.0864E-6	1.0000E+0				120.0052E+0
8		785.0755E+0	506.0818E-6	1.0000E+0				120.0258E+0
9		905.0780E+0	506.0855E-6	1.0000E+0				120.0025E+0
10		1.0251E+3	506.0845E-6	1.0000E+0				120.0053E+0

Run History:
- Idle Compression (Run2) 2/13/2025 06:50:07.000 ✓
- No Data Compression (Run1) 2/12/2025 20:30:02.000 ✓

2/13/2025 06:50:07

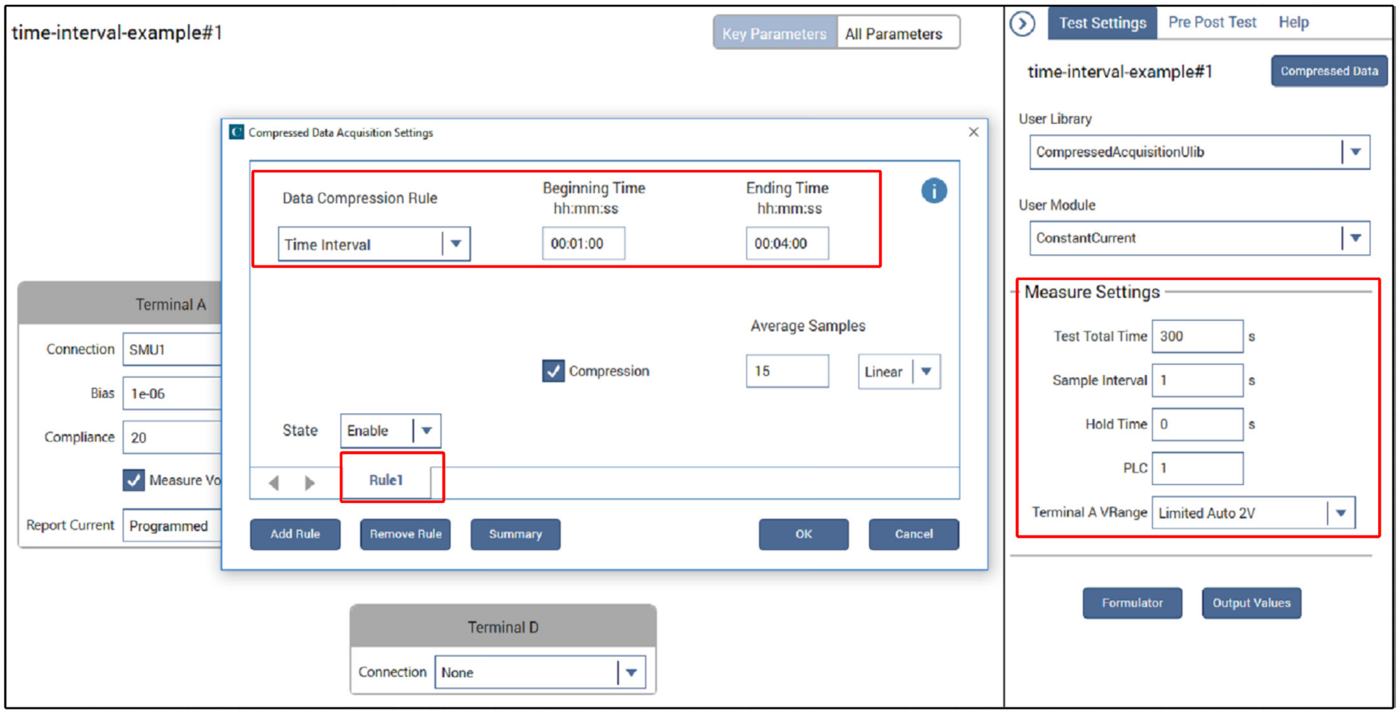
1V, 2kohm Resistor, 1 hour data
No Data Compression: 1 reading every 10 seconds (360 readings total)
Idle Compression: 1 reading averaged over every 120 seconds (30 readings total)

Figure 17. Analyze view of ***idle-compression-example*** test with and without data compression.

time-interval-example Test

The ***time-interval-example*** test defines the start and stop times in HH:MM:SS format using the Time Interval rule, which compresses data within a defined time period. The Configure view of this particular test is shown in **Figure 18**.

The **Total Test Time** is set to 300 s (5 minutes) with a sample interval of 1 s. However, the **Beginning Time** to start data compression is at 00:01:00 (1 minute) and the **Ending Time** is set to 00:04:00 (the 4-minute mark) in the test. During the compression period, 15 samples are averaged for each reading, or every 15 s. During the first and last minute of the test, readings are taken every second (the Sample Interval).



time-interval-example#1

Compressed Data Acquisition Settings

Data Compression Rule: Time Interval
Beginning Time hh:mm:ss: 00:01:00
Ending Time hh:mm:ss: 00:04:00
Compression: ✓
Average Samples: 15, Linear
State: Enable
Rule1

Terminal A
Connection: SMU1
Bias: 1e-06
Compliance: 20
✓ Measure Vo
Report Current: Programmed

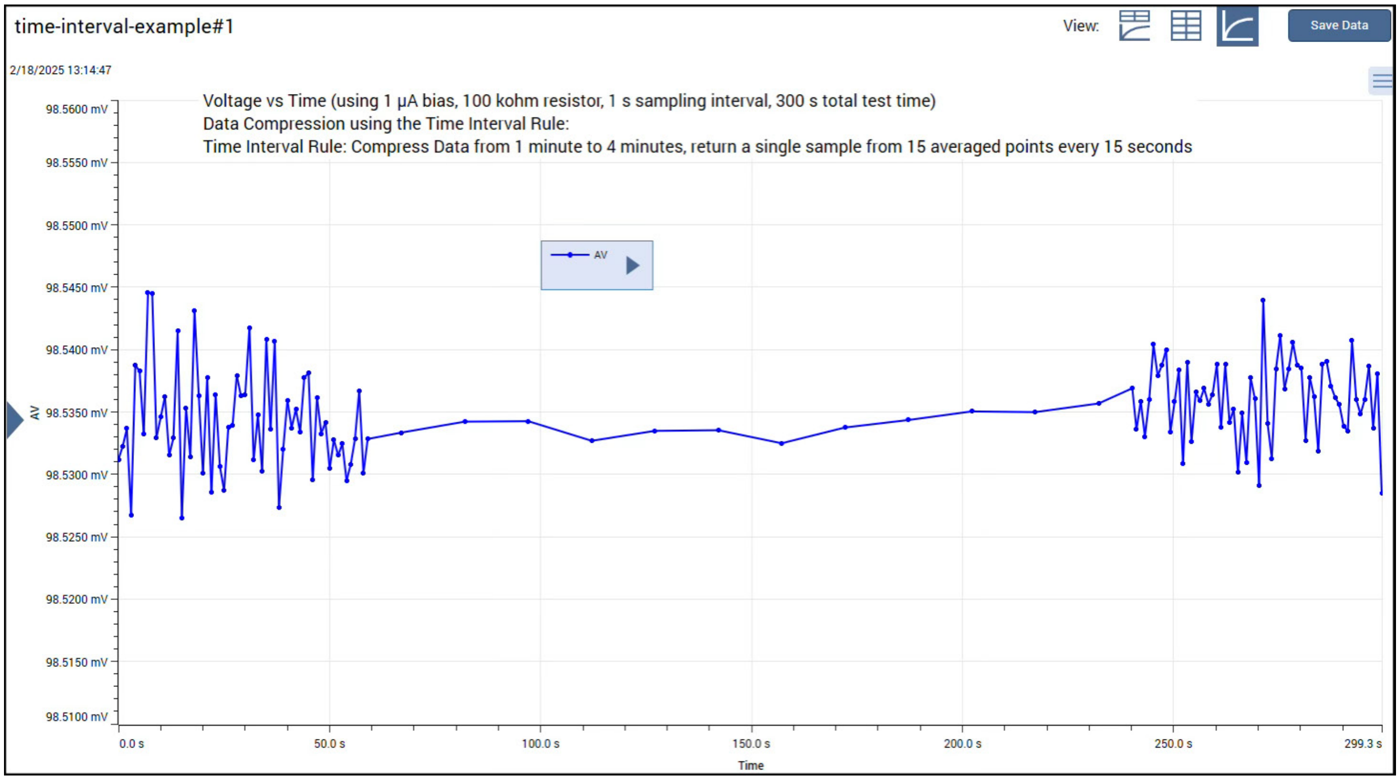
Terminal D
Connection: None

time-interval-example#1
User Library: CompressedAcquisitionUlib
User Module: ConstantCurrent

Measure Settings
Test Total Time: 300 s
Sample Interval: 1 s
Hold Time: 0 s
PLC: 1
Terminal A VRange: Limited Auto 2V

Figure 18. Configure view of *time-interval-example* test.

When this test is executed, the graphical results are shown in **Figure 19**. Notice in the graph that data is only compressed in the three minutes in the middle of the test. During the first and last minute of the test, all the data is returned at a rate of 1 reading per second.

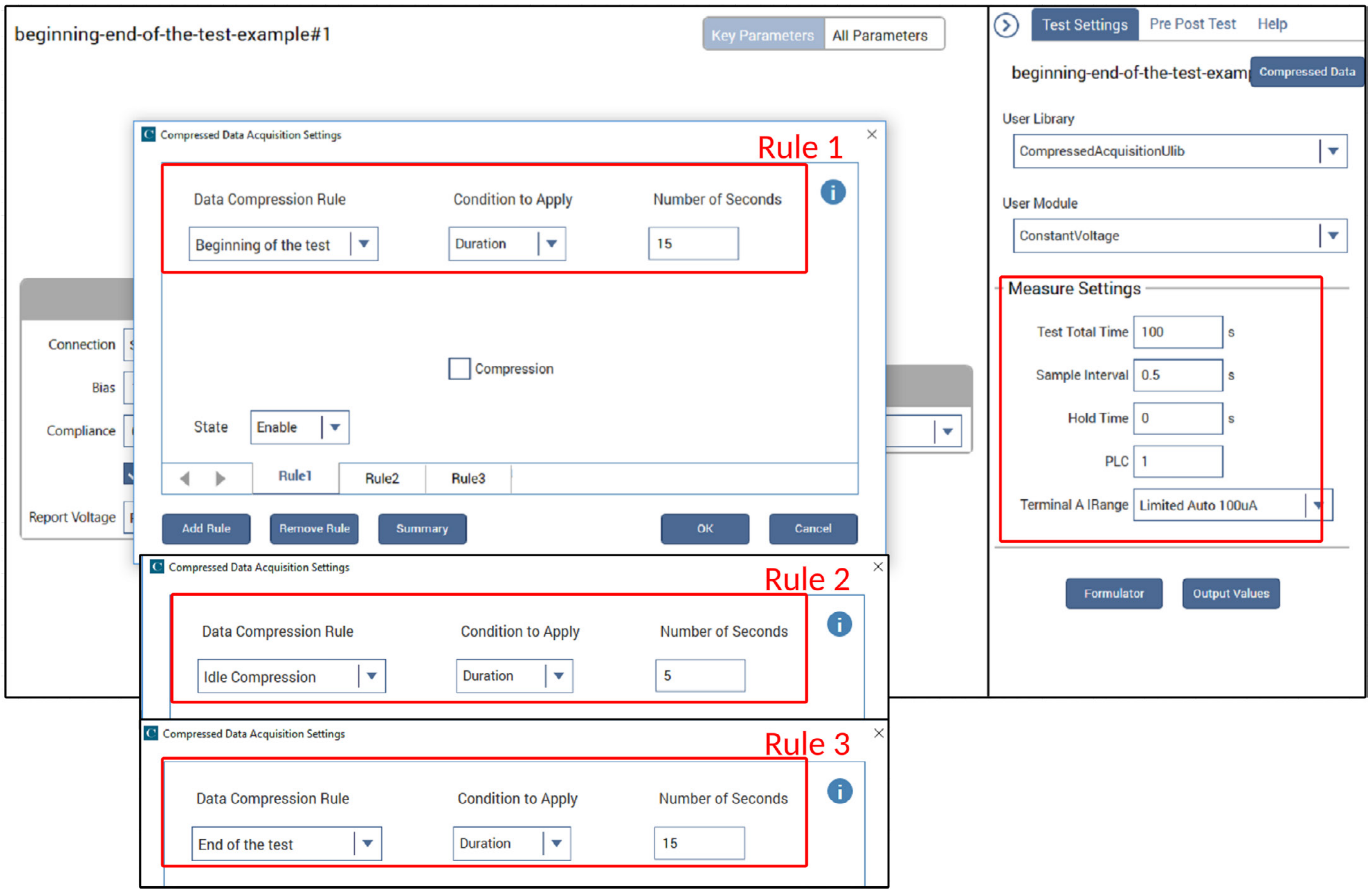


Voltage vs Time (using 1 µA bias, 100 kohm resistor, 1 s sampling interval, 300 s total test time)
Data Compression using the Time Interval Rule:
Time Interval Rule: Compress Data from 1 minute to 4 minutes, return a single sample from 15 averaged points every 15 seconds

Figure 19. Graphical results of *time-interval-example* test.

beginning-end-of-the-test-example Test

The ***beginning-end-of-the-test-example*** test illustrates how to return all data when the test starts and stops but compress data during the rest of the test. This is done using the Beginning of the Test, End of the Test and Idle Compression rules. In this example test, SMU1 outputs a 1 V bias to a variable resistor and the resulting current and time are measured. **Figure 20** shows the three rules in the Configure view of Clarius.



beginning-end-of-the-test-exam

User Library: CompressedAcquisitionUlib

User Module: ConstantVoltage

Measure Settings

Test Total Time	100	s
Sample Interval	0.5	s
Hold Time	0	s
PLC	1	
Terminal A IRange	Limited Auto 100uA	

Rule 1:

Data Compression Rule	Condition to Apply	Number of Seconds
Beginning of the test	Duration	15

State: Enable

Rule 2:

Data Compression Rule	Condition to Apply	Number of Seconds
Idle Compression	Duration	5

Rule 3:

Data Compression Rule	Condition to Apply	Number of Seconds
End of the test	Duration	15

Figure 20. Configure view of *beginning-end-of-test-example* test with Beginning of the Test (Rule1), Idle Compression (Rule2) and End of the Test (Rule3) Compression rules.

When the ***beginning-end-of-test-example*** test is executed, the graphical results are shown in **Figure 21**. During the first and last 15 s of the test, all the samples are returned at the rate of 0.5 s/reading. During the rest of the test, idle compression is used to output an averaged reading every 5 s.

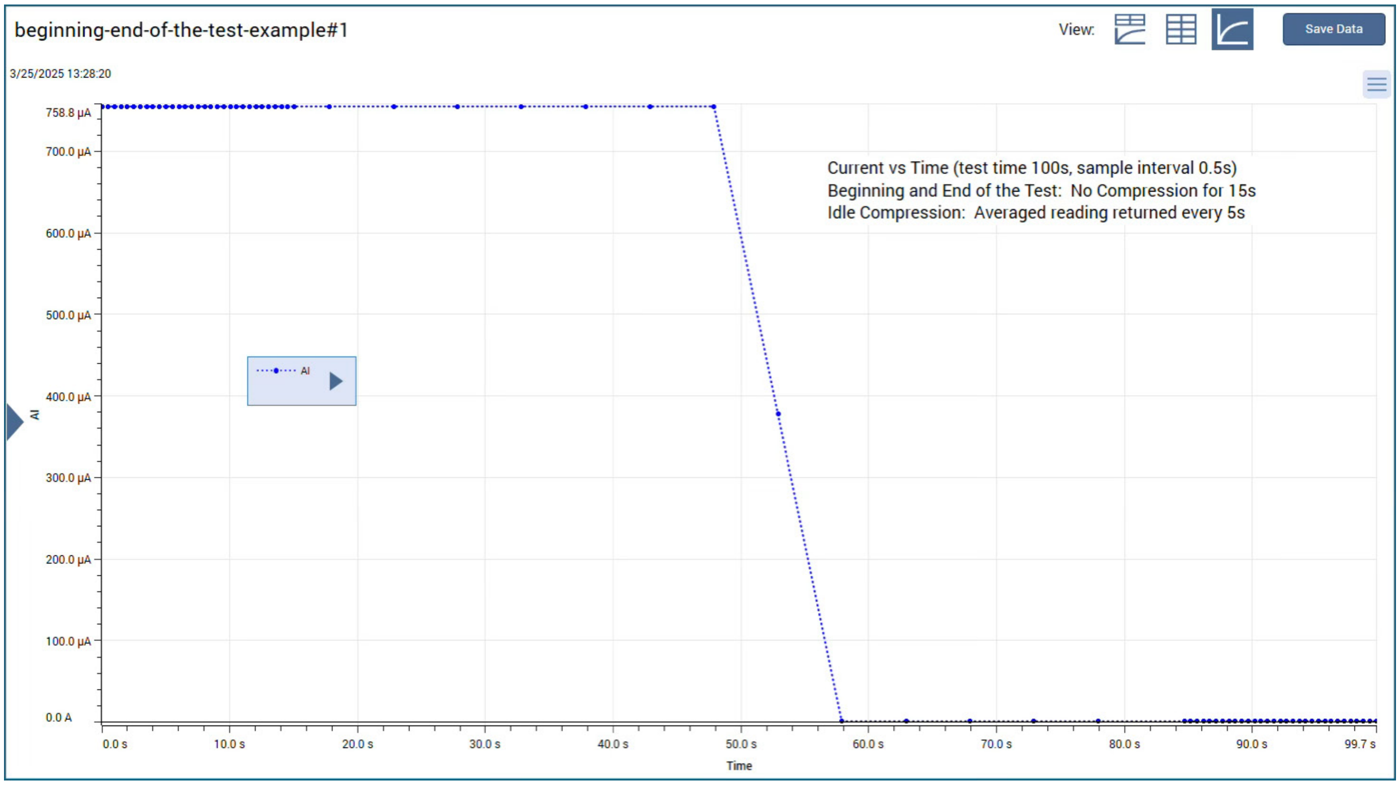


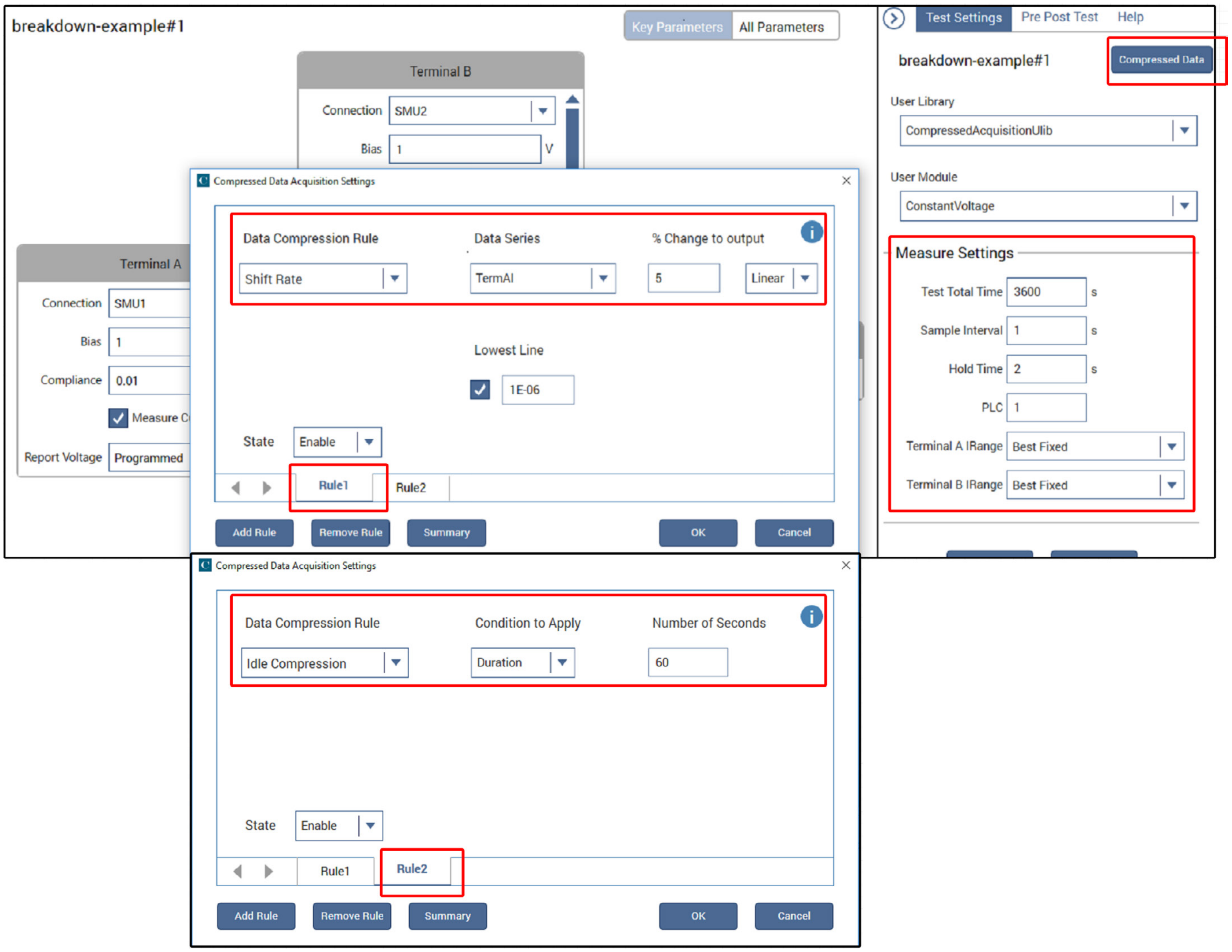
Figure 21. Graphical results of *beginning-end-of-the-test-example*.

breakdown-example Test

The ***breakdown-example*** test demonstrates how data compression can be used for breakdown testing using the Shift Rate and Idle Compression rules. In this example, both SMU1 and SMU2 output a 1 V bias, but the current is measured only on SMU1. This test could be used for a breakdown test such as time-dependent dielectric breakdown (TDDB). The total test time is 3600 s (1 hour). The data taken during the transition of the current measurements is the most important to capture.

Figure 22 shows the rules and measure settings in the Configure view of this test. Even though two SMUs are used, the Shift Rate (Rule1) only applies to data taken at TermA (SMU1). In this example, data is recorded if there is a 5% change between readings. The Shift Rate readings are recorded at the Sample Interval rate of 1 reading/s.

Rule2 defines the Idle Compression rule. If the data is not affected by the Shift Rate (5% change), then one data point is averaged and returned every 60 s. In this case, 60 readings (1 rdg/s) are averaged for each returned data point.



Terminal B: Connection SMU2; Bias 1 V

Terminal A: Connection SMU1; Bias 1; Compliance 0.01; Measure C; Report Voltage Programmed

Compressed Data Acquisition Settings (Rule1): Data Compression Rule: Shift Rate; Data Series: TermAI; % Change to output: 5, Linear; Lowest Line: ✓ 1E-06; State: Enable

User Library: CompressedAcquisitionUlib
User Module: ConstantVoltage

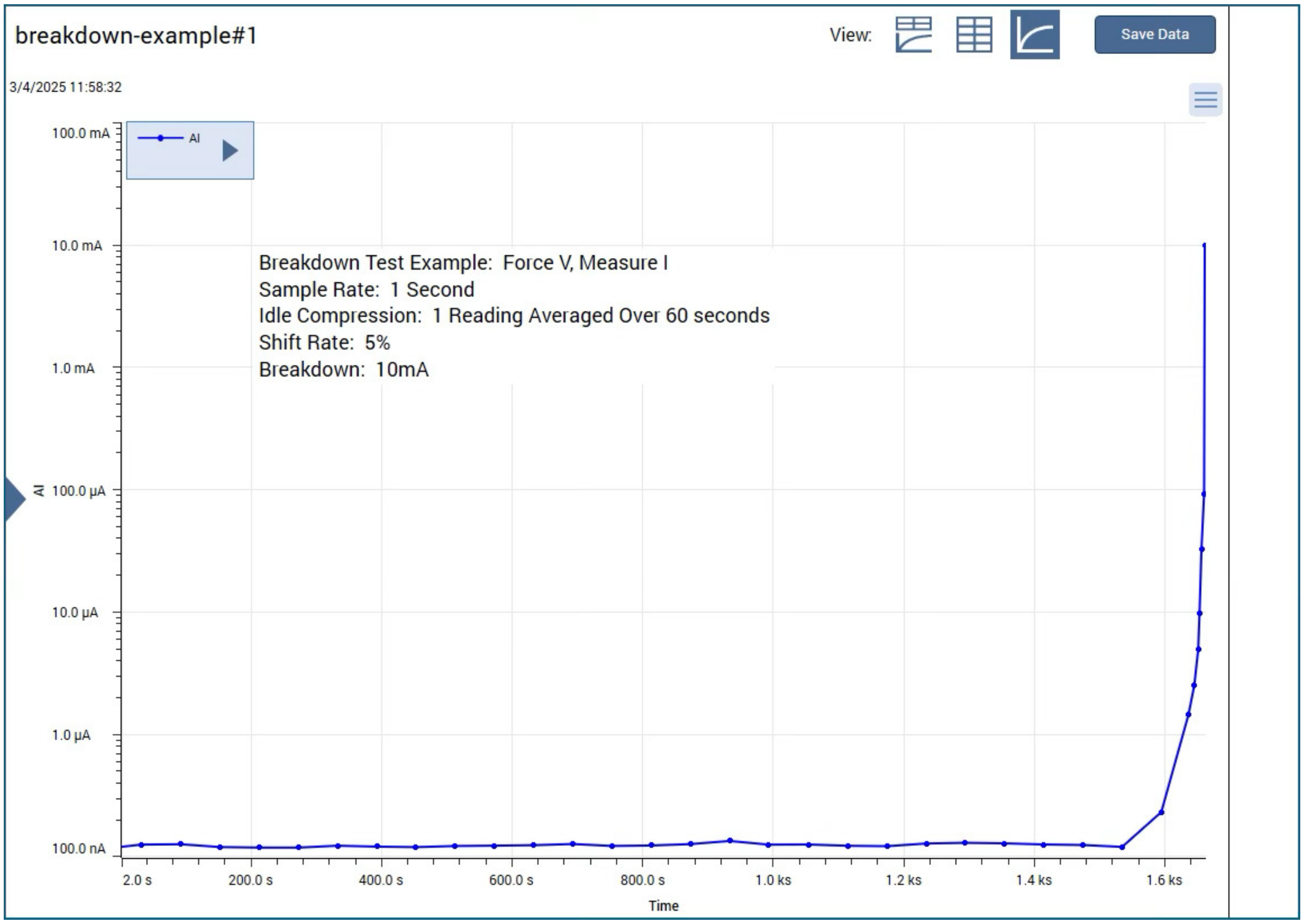
Measure Settings:

Test Total Time	3600 s
Sample Interval	1 s
Hold Time	2 s
PLC	1
Terminal A IRange	Best Fixed
Terminal B IRange	Best Fixed

Compressed Data Acquisition Settings (Rule2): Data Compression Rule: Idle Compression; Condition to Apply: Duration; Number of Seconds: 60; State: Enable

Figure 22. Configure view of ***breakdown-example*** test with Shift Rate and Idle Compression Rules.

The results of executing the ***breakdown-example*** test are shown in **Figure 23**. Notice that data points are initially graphed every 60 s. Once the device starts to break down, data points are recorded every 1 s if the readings are changing at least 5 percent.



Breakdown Test Example: Force V, Measure I
Sample Rate: 1 Second
Idle Compression: 1 Reading Averaged Over 60 seconds
Shift Rate: 5%
Breakdown: 10mA

Figure 23. Analyze Graph view of *breakdown-example* test.

How to Add Data Compression in User Modules

Data compression can be added to any new or existing user module that collects data. To add data compression, the correct input array parameters must be added to the user module in the Keithley User Library Tool (KULT) and functions must be added to control data compression. The UTM UI Editor can also be used to set default data compression rules.

KULT and the KULT Extension for Visual Studio Code can be used to create and manage user libraries. For more information on KULT, refer to the KULT manual: 4200A-SCS KULT and KULT Extension Programming | Tektronix.

Input Array Parameters

To enable Compressed Data Acquisition within a user module, the following nine input parameters must be included:

1. **int[ ] RuleType:** Sets the type of data compression rule used. The possible values are:
   - ShiftRate = 1
   - BeginningOfTestDuration = 2
   - BeginningOfTestSamples = 3
   - EndOfTestDuration = 4
   - EndOfTestSamples = 5
   - TimeInterval = 6
   - IdleDuration = 7
   - IdleSamples = 8
2. **int[ ] RuleDataSeries:** Sets the index of the UTM output parameter and applies for ShiftRate only.
3. **double[ ] RuleValue1:** Can be set to percent change, duration, number of samples, or beginning time based on RuleType.
4. **double[ ] RuleValue2:** Ending time for the Time Interval rule or linear/log selection flag for ShiftRate rule.
5. **int[ ] RuleOption:** Options include:
   - 0 - No Options
   - 1 - Lowest Line Option
   - 2 - Compression Option
6. **int[ ] RuleOptionMode:** A linear or log selection flag for "Compression Option", used for beginning or end of the test and time interval.
7. **double[ ] RuleOptionValue1:** Data series value for "Lowest Line Option".
8. **int[ ] RuleOptionValue2:** Average count for "Compression Option".
9. **int[ ] RuleEnabled:** Flag for enabling or disabling a rule:
   - 1 - Enabled
   - 0 – Disabled

All these parameters are input arrays, allowing configuration settings for multiple rules. Each rule has a configuration in these arrays. An example template for using data compression is shown in Appendix A.

The Clarius UTM user interface displays the **Compressed Data** button in the top right corner of the UTM Test Settings pane when it detects the above set of **Compressed Data Acquisition** parameters. No special GUI configuration in the UTM UI Editor is required. These nine preconfigured parameters are not available in the UTM UI Editor for use in configuration for any UTM UI group. The user may configure these parameters with the Compressed Data Acquisition configuration dialog and set their defaults within the UTM UI Editor.

CompressedAcqUtils Internal Library

In addition to the nine preconfigured data compression parameters, user modules must call the internal ***CompressedAcqUtils*** library (DLL). ***CompressedAcqUtils*** initializes and manages the data compression engine. Since this is an internal library, it can be used within any user library.

The following functions control the Data Compression engine:

1. **InitializeCompression**
   int InitializeCompression()
   - This function initializes the internal library and must be the first function called from the *CompressedAcqUtils* DLL. If called more than once, it clears any existing data. If this function has not been called, the remaining functions return with an error.
2. **FinalizeCompression**
   int FinalizeCompression(BOOL postData)

- This function tells the internal library that the end of a test has been reached. Any remaining unprocessed data is compressed, as necessary. This is also required for using an **End of the Test** rule.

3. **SetCompressionRules**
   int SetCompressionRules(int RuleType, int* RuleDataSeries, double* RuleValue1, double* RuleValue2, int* RuleOption, int* RuleOptionMode, double* RuleOptionValue1, int* RuleOptionValue2, int* RuleEnabled, int RuleCount)*
   - This function is used to transfer the user-defined information from the **Compressed Data Acquisition Settings** into the internal library. It returns an error if any rule is invalid and cannot be processed.
4. **SetDataSeriesForCompression**
   int SetDataSeriesForCompression(char dataSeries, int* dataType, int seriesCount)*
   - This function passes in the list of UTM outputs separated by commas in dataSeries. This is required before processing any data. The dataType array contains the data type for each data series (double = 0, integer = 1, and time = 2). The size of this array is seriesCount. The function returns an error if the dataSeries array cannot read the array of names. Only one series with the time data type is allowed. A series with the time data type is required to use any time-based data compression rules.
5. **ProcessSampleOfData**
   int ProcessSampleOfData(double dataSample, int sampleSize, BOOL postData)*
   - This function evaluates what to do with a new sample of data. The array size of dataSample is equal to sampleSize and seriesCount. The function returns post data to Clarius if postData is set to true. If post data is set to false, then the user module must be set to return data back with PostDataDouble(). The user module can be set to retrieve the compressed samples by calling GetSampleDataToReturn.
6. **GetSampleDataToReturn**
   int GetSampleDataToReturn(double sampleData, int sampleSize)*
   - This function pulls in the sample data from the selected data series. When data is returned by this function, it is removed from the queue of remaining data. It returns an error if there are no data samples to return.

Configuring the UTM UI Editor with Data Compression Features

To change the data compression defaults in the UTM UI Editor, use the following steps:

1. Close Clarius.
2. From the Windows Start menu, select Keithley Instruments > UTM UI Editor. The UTM UI Editor opens.
3. In the right pane, from the User Libraries list, select the user library.
4. From the User Modules list, select the user module.
5. Select the **Compressed Data** button next to the user module name.

This opens the **Compressed Data Acquisition Default Settings** dialog, similar to the **Compressed Data Acquisition Settings** within a user module. Default rules and configurations can be set from this dialog.

Conclusion

In conclusion, the Data Compression feature introduced in Clarius V1.14 for the Keithley 4200A-SCS Parameter Analyzer significantly enhances the efficiency of long-term semiconductor device testing. By allowing users to focus on critical data points and compress less significant information, this feature ensures tests can be performed without exceeding data limits. Data Compression allows users to maximize the capabilities of the 4200A-SCS Parameter Analyzer.

Appendix A: Data Compression Code Skeleton

Note: This code will not run by itself. It is meant to act as a framework for adding data compression to existing code.

```
/* USRLIB MODULE INFORMATION

    MODULE NAME: CompressedModuleExample
    MODULE RETURN TYPE: int
    NUMBER OF PARMS: 18
    ARGUMENTS:
        RuleType,   I_ARRAY_T,  Input,  ,   ,
        RuleTypeSize,   int,    Input,  ,   ,
        RuleDataSeries, I_ARRAY_T,  Input,  ,   ,
        RuleDataSeriesSize, int,    Input,  ,   ,
        RuleValue1, D_ARRAY_T,  Input,  ,   ,
        RuleValue1Size, int,    Input,  ,   ,
        RuleValue2, D_ARRAY_T,  Input,  ,   ,
        RuleValue2Size, int,    Input,  ,   ,
        RuleOption, I_ARRAY_T,  Input,  ,   ,
        RuleOptionSize, int,    Input,  ,   ,
        RuleOptionMode, I_ARRAY_T,  Input,  ,   ,
        RuleOptionModeSize, int,    Input,  ,   ,
        RuleOptionValue1,   D_ARRAY_T,  Input,  ,   ,
        RuleOptionValue1Size,   int,    Input,  ,   ,
        RuleOptionValue2,   I_ARRAY_T,  Input,  ,   ,
        RuleOptionValue2Size,   int,    Input,  ,   ,
        RuleEnabled,    I_ARRAY_T,  Input,  ,   ,
        RuleEnabledSize,    int,    Input,  ,   ,
    INCLUDES:
#include "keithley.h"
#include "CompressedAcqUtils.h"

    END USRLIB MODULE INFORMATION
*/
/* USRLIB MODULE HELP DESCRIPTION
<!--MarkdownExtra-->
<link rel="stylesheet" type="text/css" href="http://clariusweb/HelpPane/stylesheet.css">

Module: CompressedModuleExample
================================

Description
-----------
This example demonstrates how to set up a user test module that uses data compression with
simulated data.

Inputs
------
```

```
Outputs
------


Return values
-------------

Value  | Description
------ | -----------
0      | OK.
-18001 | Rule inputs must be the same size.

    END USRLIB MODULE HELP DESCRIPTION */
/* USRLIB MODULE PARAMETER LIST */
#include "keithley.h"
#include "CompressedAcquisitionUlib_internal.h"
#include "CompressedAcqUtils.h"

/* USRLIB MODULE MAIN FUNCTION */
int CompressedModuleExample( int *RuleType, int RuleTypeSize, int *RuleDataSeries,
int RuleDataSeriesSize, double *RuleValue1, int RuleValue1Size, double *RuleValue2,
int RuleValue2Size, int *RuleOption, int RuleOptionSize, int *RuleOptionMode, int
RuleOptionModeSize, double *RuleOptionValue1, int RuleOptionValue1Size, int *RuleOptionValue2,
int RuleOptionValue2Size, int *RuleEnabled, int RuleEnabledSize )
{
/* USRLIB MODULE CODE */
    char ErrMsg[255];
    int status = 0;
    // Data Series parameters.
    int seriesCount = 2; //number of outputs is 2: Index and Output1.
    char* dataSeries = "Index,Output1";
    int dataType[2] = { 1, 0 };
    // Data types will be double=0, integer=1 and time=2.
    // Initialize the Compression library.
    status = InitializeCompression();
    if (status != 0)
    {
        return status;
    }
    // Call SetCompressionRules().
    status = SetCompressionRules(RuleType, RuleDataSeries, RuleValue1, RuleValue2,RuleOption,
RuleOptionMode, RuleOptionValue1, RuleOptionValue2, RuleEnabled, RuleEnabledSize);
    if (status != 0)
    {
        return status;
    }
    status = SetDataSeriesForCompression(dataSeries, dataType, seriesCount);
    if (status != 0)
    {
        return status;
    }
    // Call ProcessSampleOfData with simulated data.
    BOOL postData = TRUE; // ProcessSampleOfData will post data if needed to Clarius using
PostDataDouble().
    double dataSample[1] = { 0.0 };
    for (int i = 0; i < MaxDataPoints; i++)
    {
        dataSample[0] = 1e-3*(i + 1);
        status = ProcessSampleOfData(dataSample, seriesCount, postData);
        if (status != 0)
        {
            // Continue the loop even if errors occurred.
            // Log errors if needed.
        }
    }
    // Notify the Compression library that the "test" is complete.
    // This is required to flush any remaining data and allow the end of test rules to be
processed.
    status = FinalizeCompression(postData);
    return status;

/* USRLIB MODULE END  */
}       /* End CompressedModuleExample.c */
```

Contact Information:

Australia	1 800 709 465
Austria*	00800 2255 4835
Balkans, Israel, South Africa and other ISE Countries	+41 52 675 3777
Belgium*	00800 2255 4835
Brazil	+55 (11) 3530-8901
Canada	1 800 833 9200
Central East Europe / Baltics	+41 52 675 3777
Central Europe / Greece	+41 52 675 3777
Denmark	+45 80 88 1401
Finland	+41 52 675 3777
France*	00800 2255 4835
Germany*	00800 2255 4835
Hong Kong	400 820 5835
India	000 800 650 1835
Indonesia	007 803 601 5249
Italy	00800 2255 4835
Japan	81 (3) 6714 3086
Luxembourg	+41 52 675 3777
Malaysia	1 800 22 55835
Mexico, Central/South America and Caribbean	52 (55) 88 69 35 25
Middle East, Asia, and North Africa	+41 52 675 3777
The Netherlands*	00800 2255 4835
New Zealand	0800 800 238
Norway	800 16098
People's Republic of China	400 820 5835
Philippines	1 800 1601 0077
Poland	+41 52 675 3777
Portugal	80 08 12370
Republic of Korea	+82 2 565 1455
Russia / CIS	+7 (495) 6647564
Singapore	800 6011 473
South Africa	+41 52 675 3777
Spain*	00800 2255 4835
Sweden*	00800 2255 4835
Switzerland*	00800 2255 4835
Taiwan	886 (2) 2656 6688
Thailand	1 800 011 931
United Kingdom / Ireland*	00800 2255 4835
USA	1 800 833 9200
Vietnam	12060128

* European toll-free number. If not accessible, call: +41 52 675 3777

Rev. 02.2022

Find more valuable resources at TEK.COM

Copyright © Tektronix. All rights reserved. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specification and price change privileges reserved. TEKTRONIX and TEK are registered trademarks of Tektronix, Inc. All other trade names referenced are the service marks, trademarks or registered trademarks of their respective companies.

040925 SBG 1KW-74148-0