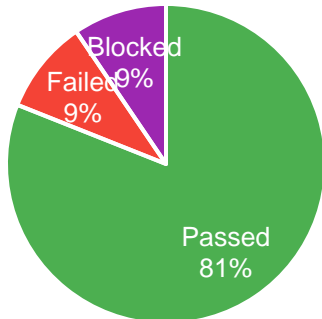# Test project report
Overview

| Tests: 253 | Passed: 205 | Failed: 24 | Blocked: 24 |

Report created: 2023-01-01 00:00:00.000
Version: v1.0.0
Duration: 0:00:08.220000
Test documents: 4
Test suites: 25

Unit Tests

- example/data/example2.xml   passed: 42 / 42
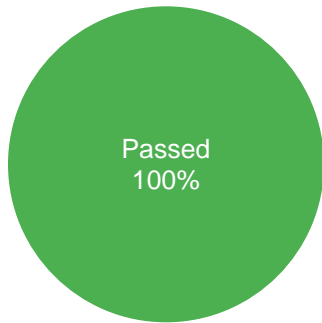- example/data/example3.xml   passed: 114 / 114
- example/data/example4.xml   passed: 25 / 25

Mutation Tests

- example/data/example1.xml   passed: 24 / 72

Unit Tests

Tests: 181	Passed: 181	Failed: 0	Blocked: 0

Passed
100%

Part of: Unit Tests

- example/data/example2.xml   passed: 42 / 42
- example/data/example3.xml   passed: 114 / 114
- example/data/example4.xml   passed: 25 / 25

Test suite "MPL"
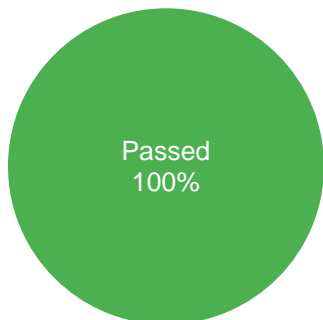
Tests: 9   Passed: 9   Failed: 0   Blocked: 0

Passed
100%

Part of: Unit Tests
File: example/data/example2.xml
Timestamp: 2023-06-03 15:34:53.000
Duration: 0:00:00.000000

- Passed: MPL.AccumulateType
- Passed: MPL.AccumulateValue
- Passed: MPL.Conditional
- Passed: MPL.Join
- Passed: MPL.Filter
- Passed: MPL.FindIf
- Passed: MPL.IsUnique
- Passed: MPL.MinElement
- Passed: MPL.Transform

Test suite "Units"

Passed
100%

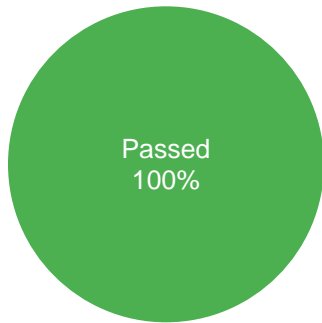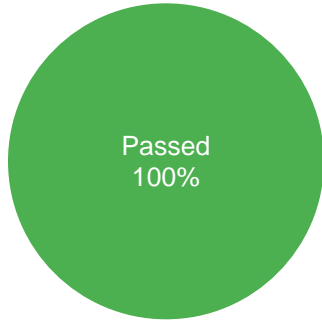| Tests: 31 | Passed: 31 | Failed: 0 | Blocked: 0 |

Part of: Unit Tests
File: example/data/example2.xml
Timestamp: 2023-06-03 15:34:53.000
Duration: 0:00:00.001000

- Passed: Units.QuantityConversion
- Passed: Units.ConvertQuantity
- Passed: Units.ConvertSpeeds
- Passed: Units.ScaleQuantityWithScalar
- Passed: Units.InvertQuantity
- Passed: Units.TrigonometricFunctions
- Passed: Units.ConvertImperial
- Passed: Units.AddQuantities
- Passed: Units.SubtractQuantities
- Passed: Units.MultiplySameQuantities
- Passed: Units.MultiplyDifferentQuantities
- Passed: Units.DivideSameQuantities
- Passed: Units.DivideDifferentQuantities
- Passed: Units.Scalar
- Passed: Units.OperationChain
- Passed: Units.DerivedUnits
- Passed: Units.Degree
- Passed: Units.MinMax
- Passed: Units.Abs
- Passed: Units.Clamp
- Passed: Units.Exp
- Passed: Units.Log
- Passed: Units.Comparisions
- Passed: Units.MultiplicationConversion
- Passed: Units.AdditonConversion
- Passed: Units.ChronoIntegration
- Passed: Units.QuantityCast
- Passed: Units.IntegralConversion
- Passed: Units.Pow
- Passed: Units.Sqrt
- Passed: Units.Traits

Test suite "FormatUnits"

Passed
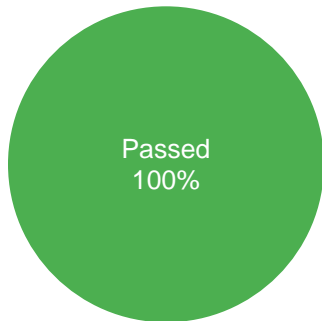100%

| Tests: 2 | Passed: 2 | Failed: 0 | Blocked: 0 |

Part of: Unit Tests
File: example/data/example2.xml
Timestamp: 2023-06-03 15:34:53.000
Duration: 0:00:00.000000

- Passed: FormatUnits.Cout
- Passed: FormatUnits.ToString

Test suite "test-filter"

| Tests: 114 | Passed: 114 | Failed: 0 | Blocked: 0 |

Passed
100%

Part of: Unit Tests
File: example/data/example3.xml
Timestamp: 2023-06-08 12:15:14.000Z
Duration: 0:00:00.007000

- Passed: test-filter.global.Scenario: Brickwall lowpass filters can be initialized/Given: The filter parameters for a 10th order brickwall lowpass filter with 100 Hz sampling rate and 25Hz cutoff/When: The filter is created
- Passed: test-filter.global.Scenario: Brickwall lowpass filters can be initialized/Given: The filter parameters for a 10th order brickwall lowpass filter with 100 Hz sampling rate and 25Hz cutoff/When: The filter is created/Then: The status is ok
- Passed: test-filter.global.Scenario: Brickwall lowpass filters can be initialized/Given: The filter parameters for a 10th order brickwall lowpass filter with 100 Hz sampling rate and 25Hz cutoff/When: The filter is created/Then: The coefficients are correct
- Passed: test-filter.global.Scenario: Brickwall highpass filters can be initialized/Given: The filter parameters for a 10th order brickwall highpass filter with 100 Hz sampling rate and 25Hz cutoff/When: The filter is created
- Passed: test-filter.global.Scenario: Brickwall highpass filters can be initialized/Given: The filter parameters for a 10th order brickwall highpass filter with 100 Hz sampling rate and 25Hz cutoff/When: The filter is created/Then: The status is ok
- Passed: test-filter.global.Scenario: Brickwall highpass filters can be initialized/Given: The filter parameters for a 10th order brickwall highpass filter with 100 Hz sampling rate and 25Hz cutoff/When: The filter is created/Then: The coefficients are correct
- Passed: test-filter.global.Scenario: Brickwall bandstop filters can be initialized/Given: The filter parameters for a 10th order bandstop highpass filter with 100 Hz sampling rate and 20Hz-40Hz cutoff/When: The filter is created
- Passed: test-filter.global.Scenario: Brickwall bandstop filters can be initialized/Given: The filter parameters for a 10th order bandstop highpass filter with 100 Hz sampling rate and 20Hz-40Hz cutoff/When: The filter is created/Then: The status is ok
- Passed: test-filter.global.Scenario: Brickwall bandstop filters can be initialized/Given: The filter parameters for a 10th order bandstop highpass filter with 100 Hz sampling rate and 20Hz-40Hz cutoff/When: The filter is created/Then: The input array is zeroed

- Passed: test-filter.global.Scenario: Brickwall bandstop filters can be initialized/Given: The filter parameters for a 10th order bandstop highpass filter with 100 Hz sampling rate and 20Hz-40Hz cutoff/When: The filter is created/Then: The output array is zeroed
- Passed: test-filter.global.Scenario: Brickwall bandstop filters can be initialized/Given: The filter parameters for a 10th order bandstop highpass filter with 100 Hz sampling rate and 20Hz-40Hz cutoff/When: The filter is created/Then: The coefficients are correct
- Passed: test-filter.global.Scenario: Brickwall bandpass filters can be initialized/Given: The filter parameters for a 10th order brickwall bandpass filter with 100 Hz sampling rate and 20Hz-40Hz cutoff/When: The filter is created
- Passed: test-filter.global.Scenario: Brickwall bandpass filters can be initialized/Given: The filter parameters for a 10th order brickwall bandpass filter with 100 Hz sampling rate and 20Hz-40Hz cutoff/When: The filter is created/Then: The status is ok
- Passed: test-filter.global.Scenario: Brickwall bandpass filters can be initialized/Given: The filter parameters for a 10th order brickwall bandpass filter with 100 Hz sampling rate and 20Hz-40Hz cutoff/When: The filter is created/Then: The input array is zeroed
- Passed: test-filter.global.Scenario: Brickwall bandpass filters can be initialized/Given: The filter parameters for a 10th order brickwall bandpass filter with 100 Hz sampling rate and 20Hz-40Hz cutoff/When: The filter is created/Then: The output array is zeroed
- Passed: test-filter.global.Scenario: Brickwall bandpass filters can be initialized/Given: The filter parameters for a 10th order brickwall bandpass filter with 100 Hz sampling rate and 20Hz-40Hz cutoff/When: The filter is created/Then: The coefficients are correct
- Passed: test-filter.global.Scenario: Butterworth output coefficients can be computed/Given: parameters: fifth order butterworth, 100 Hz sampling rate, 20Hz cutoff/When: the lowpass parameters are computed
- Passed: test-filter.global.Scenario: Butterworth output coefficients can be computed/Given: parameters: fifth order butterworth, 100 Hz sampling rate, 20Hz cutoff/When: the lowpass parameters are computed/Then: status is ok
- Passed: test-filter.global.Scenario: Butterworth output coefficients can be computed/Given: parameters: fifth order butterworth, 100 Hz sampling rate, 20Hz cutoff/When: the lowpass parameters are computed/Then: results are correct
- Passed: test-filter.global.Scenario: Butterworth lowpass filters can be initialized/Given: parameters: fifth order butterworth, 100 Hz sampling rate, 25Hz cutoff/When: lowpass parameters are computed
- Passed: test-filter.global.Scenario: Butterworth lowpass filters can be initialized/Given: parameters: fifth order butterworth, 100 Hz sampling rate, 25Hz cutoff/When: lowpass parameters are computed/Then: status is ok

- Passed: test-filter.global.Scenario: Butterworth lowpass filters can be initialized/Given: parameters: fifth order butterworth, 100 Hz sampling rate, 25Hz cutoff/When: lowpass parameters are computed/Then: coefficients are correct
- Passed: test-filter.global.Scenario: Butterworth highpass filters can be initialized/Given: parameters: fifth order butterworth, 100 Hz sampling rate, 25Hz cutoff/When: highpass parameters are computed
- Passed: test-filter.global.Scenario: Butterworth highpass filters can be initialized/Given: parameters: fifth order butterworth, 100 Hz sampling rate, 25Hz cutoff/When: highpass parameters are computed/Then: status is ok
- Passed: test-filter.global.Scenario: Butterworth highpass filters can be initialized/Given: parameters: fifth order butterworth, 100 Hz sampling rate, 25Hz cutoff/When: highpass parameters are computed/Then: coefficients are correct
- Passed: test-filter.global.Scenario: Butterworth bandstop filters can be initialized/Given: parameters: third order butterworth, 100 Hz sampling rate, 15-30Hz bandstop/When: bandstop parameters are computed
- Passed: test-filter.global.Scenario: Butterworth bandstop filters can be initialized/Given: parameters: third order butterworth, 100 Hz sampling rate, 15-30Hz bandstop/When: bandstop parameters are computed/Then: status is ok
- Passed: test-filter.global.Scenario: Butterworth bandstop filters can be initialized/Given: parameters: third order butterworth, 100 Hz sampling rate, 15-30Hz bandstop/When: bandstop parameters are computed/Then: coefficients are correct
- Passed: test-filter.global.Scenario: Butterworth bandpass filters can be initialized/Given: parameters: third order butterworth, 100 Hz sampling rate, 15-30Hz bandpass/When: bandpass parameters are computed
- Passed: test-filter.global.Scenario: Butterworth bandpass filters can be initialized/Given: parameters: third order butterworth, 100 Hz sampling rate, 15-30Hz bandpass/When: bandpass parameters are computed/Then: status is ok
- Passed: test-filter.global.Scenario: Butterworth bandpass filters can be initialized/Given: parameters: third order butterworth, 100 Hz sampling rate, 15-30Hz bandpass/When: bandpass parameters are computed/Then: results are correct
- Passed: test-filter.global.Scenario: Chebyshev output coefficients can be computed/Given: parameters: fourth order chebyshev, 100 Hz sampling rate, 25Hz cutoff, 3 db ripple, lowpass/When: denominator is computed/Then: the status is ok
- Passed: test-filter.global.Scenario: Chebyshev output coefficients can be computed/Given: parameters: fourth order chebyshev, 100 Hz sampling rate, 25Hz cutoff, 3 db ripple, lowpass/When: denominator is computed/Then: results are correct
- Passed: test-filter.global.Scenario: Chebyshev output coefficients can be computed/Given: parameters: fifth order chebyshev, 100 Hz sampling rate, 20Hz cutoff, 2 db ripple, high pass/When: coefficients are computed/Then: the status is ok

- Passed: test-filter.global.Scenario: Chebyshev output coefficients can be computed/Given: parameters: fifth order chebyshev, 100 Hz sampling rate, 20Hz cutoff, 2 db ripple, high pass/When: coefficients are computed/Then: results are correct
- Passed: test-filter.global.Scenario: Chebyshev output coefficients can be computed/Given: parameters: third order chebyshev, 100 Hz sampling rate, 15, 30Hz cutoff, 3 db ripple, band pass/When: coefficients are computed/Then: the status is ok
- Passed: test-filter.global.Scenario: Chebyshev output coefficients can be computed/Given: parameters: third order chebyshev, 100 Hz sampling rate, 15, 30Hz cutoff, 3 db ripple, band pass/When: coefficients are computed/Then: denomiator is correct
- Passed: test-filter.global.Scenario: Chebyshev output coefficients can be computed/Given: parameters: third order chebyshev, 100 Hz sampling rate, 15, 30Hz cutoff, 3 db ripple, band pass/When: coefficients are computed/Then: numerator is correct
- Passed: test-filter.global.Scenario: Chebyshev output coefficients can be computed/Given: parameters: third order chebyshev, 100 Hz sampling rate, 15, 30Hz cutoff, 3 db ripple, band stop/When: coefficients are computed/Then: the status is ok
- Passed: test-filter.global.Scenario: Chebyshev output coefficients can be computed/Given: parameters: third order chebyshev, 100 Hz sampling rate, 15, 30Hz cutoff, 3 db ripple, band stop/When: coefficients are computed/Then: denomiator is correct
- Passed: test-filter.global.Scenario: Chebyshev output coefficients can be computed/Given: parameters: third order chebyshev, 100 Hz sampling rate, 15, 30Hz cutoff, 3 db ripple, band stop/When: coefficients are computed/Then: numerator is correct
- Passed: test-filter.global.Scenario: Chebyshev low pass filters are correctly initialized/Given: parameters: fourth order chebyshev, 100 Hz sampling rate, 25Hz cutoff, 3 db ripple, lowpass/When: filter is created
- Passed: test-filter.global.Scenario: Chebyshev low pass filters are correctly initialized/Given: parameters: fourth order chebyshev, 100 Hz sampling rate, 25Hz cutoff, 3 db ripple, lowpass/When: filter is created/Then: the status is ok
- Passed: test-filter.global.Scenario: Chebyshev low pass filters are correctly initialized/Given: parameters: fourth order chebyshev, 100 Hz sampling rate, 25Hz cutoff, 3 db ripple, lowpass/When: filter is created/Then: buffers are initialized
- Passed: test-filter.global.Scenario: Chebyshev low pass filters are correctly initialized/Given: parameters: fourth order chebyshev, 100 Hz sampling rate, 25Hz cutoff, 3 db ripple, lowpass/When: filter is created/Then: feedback coefficients are correct

- Passed: test-filter.global.Scenario: Chebyshev low pass filters are correctly initialized/Given: parameters: fourth order chebyshev, 100 Hz sampling rate, 25Hz cutoff, 3 db ripple, lowpass/When: filter is created/Then: feedforward coefficients are correct
- Passed: test-filter.global.Scenario: Chebyshev high pass filters are correctly initialized/Given: parameters: fifth order chebyshev, 100 Hz sampling rate, 20Hz cutoff, 2 db ripple, high pass/When: coefficients are computed
- Passed: test-filter.global.Scenario: Chebyshev high pass filters are correctly initialized/Given: parameters: fifth order chebyshev, 100 Hz sampling rate, 20Hz cutoff, 2 db ripple, high pass/When: coefficients are computed/Then: the status is ok
- Passed: test-filter.global.Scenario: Chebyshev high pass filters are correctly initialized/Given: parameters: fifth order chebyshev, 100 Hz sampling rate, 20Hz cutoff, 2 db ripple, high pass/When: coefficients are computed/Then: buffers are initialized
- Passed: test-filter.global.Scenario: Chebyshev high pass filters are correctly initialized/Given: parameters: fifth order chebyshev, 100 Hz sampling rate, 20Hz cutoff, 2 db ripple, high pass/When: coefficients are computed/Then: feedback coefficients are correct
- Passed: test-filter.global.Scenario: Chebyshev high pass filters are correctly initialized/Given: parameters: fifth order chebyshev, 100 Hz sampling rate, 20Hz cutoff, 2 db ripple, high pass/When: coefficients are computed/Then: feedforward coefficients are correct
- Passed: test-filter.global.Scenario: Chebyshev band pass filters are correctly initialized/Given: parameters: third order chebyshev, 100 Hz sampling rate, 15, 30Hz cutoff, 3 db ripple, band pass/When: coefficients are computed
- Passed: test-filter.global.Scenario: Chebyshev band pass filters are correctly initialized/Given: parameters: third order chebyshev, 100 Hz sampling rate, 15, 30Hz cutoff, 3 db ripple, band pass/When: coefficients are computed/Then: results are correct
- Passed: test-filter.global.Scenario: Chebyshev band stop filters are correctly initialized/Given: parameters: third order chebyshev, 100 Hz sampling rate, 15, 30Hz cutoff, 3 db ripple, band stop/When: coefficients are computed
- Passed: test-filter.global.Scenario: Chebyshev band stop filters are correctly initialized/Given: parameters: third order chebyshev, 100 Hz sampling rate, 15, 30Hz cutoff, 3 db ripple, band stop/When: coefficients are computed/Then: results are correct
- Passed: test-filter.global.Scenario: Chebyshev type 2 design/Given: I need to compute coefficients for a second order chebyshev type2 lowpass at 15 Hz/100Hz with -0.3dB ripple/When: the values are computed/Then: the status is ok
- Passed: test-filter.global.Scenario: Chebyshev type 2 design/Given: I need to compute coefficients for a second order chebyshev type2 lowpass at 15 Hz/100Hz with -0.3dB ripple/When: the values are computed/Then: the feedforward coefficients are correct

- Passed: test-filter.global.Scenario: Chebyshev type 2 design/Given: I need to compute coefficients for a second order chebyshev type2 lowpass at 15 Hz/100Hz with -0.3dB ripple/When: the values are computed/Then: the feedback coefficients are correct
- Passed: test-filter.global.Scenario: Chebyshev type 2 design/Given: I need to compute coefficients for a second order chebyshev type2 highpass at 15 Hz/100Hz with -0.3dB ripple/When: the values are computed/Then: the status is ok
- Passed: test-filter.global.Scenario: Chebyshev type 2 design/Given: I need to compute coefficients for a second order chebyshev type2 highpass at 15 Hz/100Hz with -0.3dB ripple/When: the values are computed/Then: the feedforward coefficients are correct
- Passed: test-filter.global.Scenario: Chebyshev type 2 design/Given: I need to compute coefficients for a second order chebyshev type2 highpass at 15 Hz/100Hz with -0.3dB ripple/When: the values are computed/Then: the feedback coefficients are correct
- Passed: test-filter.global.Scenario: Chebyshev type 2 design/Given: I need to compute a second order bandpass at [30Hz,60Hz] for 200Hz sampling rate with 3db ripple/When: i compute the polynomial/Then: the status is ok
- Passed: test-filter.global.Scenario: Chebyshev type 2 design/Given: I need to compute a second order bandpass at [30Hz,60Hz] for 200Hz sampling rate with 3db ripple/When: i compute the polynomial/Then: the feedforward coefficients are correct
- Passed: test-filter.global.Scenario: Chebyshev type 2 design/Given: I need to compute a second order bandpass at [30Hz,60Hz] for 200Hz sampling rate with 3db ripple/When: i compute the polynomial/Then: the feedback coefficients are correct
- Passed: test-filter.global.Scenario: Chebyshev type 2 design/Given: I need to compute a second order bandstop at [30Hz,60Hz] for 200Hz sampling rate with 3db ripple/When: i compute the polynomial/Then: the status is ok
- Passed: test-filter.global.Scenario: Chebyshev type 2 design/Given: I need to compute a second order bandstop at [30Hz,60Hz] for 200Hz sampling rate with 3db ripple/When: i compute the polynomial/Then: the feedforward coefficients are correct
- Passed: test-filter.global.Scenario: Chebyshev type 2 design/Given: I need to compute a second order bandstop at [30Hz,60Hz] for 200Hz sampling rate with 3db ripple/When: i compute the polynomial/Then: the feedback coefficients are correct
- Passed: test-filter.global.Scenario: A moving average filter can be created/Given: An options structure filled with values for moving average filters/When: the create function is called
- Passed: test-filter.global.Scenario: A moving average filter can be created/Given: An options structure filled with values for moving average filters/When: the create function is called/Then: the buffers are initialized
- Passed: test-filter.global.Scenario: A moving average filter can be created/Given: An options structure filled with values for moving average filters/When: the create function is called/Then: The filter can be initialized

- Passed: test-filter.global.Scenario: A moving average filter can be created/Given: An options structure filled with values for moving average filters/When: the create function is called/Then: a step function can be filtered
- Passed: test-filter.global.Scenario: A moving average filter can be used/Given: A moving average filter struct is manually created/When: a step function is filtered/Then: the filter first returns zero
- Passed: test-filter.global.Scenario: A moving average filter can be used/Given: A moving average filter struct is manually created/When: a step function is filtered/Then: the filter function is called with a step function the output has a linear response
- Passed: test-filter.global.Scenario: A moving average filter can be used/Given: A moving average filter struct is manually created/When: a step function is filtered/Then: the filter function is called with a step function the output has a linear response
- Passed: test-filter.global.Scenario: A moving average filter can be used/Given: A moving average filter struct is manually created/When: a step function is filtered/Then: the filter function is called with a step function the output has a linear response
- Passed: test-filter.global.Scenario: A moving average filter can be used/Given: A moving average filter struct is manually created/When: a step function is filtered/Then: the filter function is called with a step function the output has a linear response
- Passed: test-filter.global.Scenario: A moving average filter can be used/Given: A moving average filter struct is manually created/When: a step function is filtered/Then: the filter function reaches a constant state after 4 calls
- Passed: test-filter.global.Scenario: A moving average filter can be used/Given: A moving average filter struct is manually created/When: a step function is filtered/When: The gain is set to 4.0/Then: the first value in the output coefficients is set to 4.0
- Passed: test-filter.global.Scenario: A moving average filter can be used/Given: A moving average filter struct is manually created/When: a step function is filtered/When: The gain is set to 4.0/When: a filter cycle is run in the steady state/Then: the filter outputs 4 times the input
- Passed: test-filter.global.Scenario: A moving average highpass filter can be used/Given: An options structure filled with values for moving average filters/When: the create function is called
- Passed: test-filter.global.Scenario: A moving average highpass filter can be used/Given: An options structure filled with values for moving average filters/When: the create function is called/Then: the buffers are initialized
- Passed: test-filter.global.Scenario: A moving average highpass filter can be used/Given: An options structure filled with values for moving average filters/When: the create function is called/Then: The filter can be initialized

- Passed: test-filter.global.Scenario: A moving average highpass filter can be used/Given: An options structure filled with values for moving average filters/When: the create function is called/Then: the output slowly drops to zero
- Passed: test-filter.global.Scenario: A moving average highpass filter can be used/Given: An options structure filled with values for moving average filters/When: the create function is called/Then: the output reaches zero
- Passed: test-filter.global.Scenario: An buttwerworth lowpass filter generated on the webpage can be used/Given: Generated code/When: A zero is filtered/Then: Output is 0
- Passed: test-filter.global.Scenario: An buttwerworth lowpass filter generated on the webpage can be used/Given: Generated code/When: A step function is filtered/Then: Output is larger than 0
- Passed: test-filter.global.Scenario: An buttwerworth lowpass filter generated on the webpage can be used/Given: Generated code/When: A step function is filtered/Then: Output is larger than before
- Passed: test-filter.global.Scenario: The coefficients are used in correct order/Given: A filter struct is manually created with different input parameters/When: an impulse input is filtered/Then: the filter first returns coefficient 0 times inmpulse
- Passed: test-filter.global.Scenario: The coefficients are used in correct order/Given: A filter struct is manually created with different input parameters/When: an impulse input is filtered/Then: the filter first returns coefficient 1 times last input
- Passed: test-filter.global.Scenario: The coefficients are used in correct order/Given: A filter struct is manually created with different input parameters/When: an impulse input is filtered/Then: the filter first returns coefficient 2 times first input
- Passed: test-filter.global.Scenario: The coefficients are used in correct order/Given: A filter struct is manually created with different input parameters/When: an impulse input is filtered/Then: the filter first returns coefficient 3 times first input
- Passed: test-filter.global.Scenario: The coefficients are used in correct order/Given: A filter struct is manually created with different input parameters/When: an impulse input is filtered/Then: the filter first returns zero
- Passed: test-filter.global.Scenario: The coefficients are used in correct order/Given: A filter struct is manually created with different output parameters and the last output is set to 1/When: filter is called/Then: the filter returns -(output coefficient 1) times last output
- Passed: test-filter.global.Scenario: The coefficients are used in correct order/Given: A filter struct is manually created with different output parameters and the last output is set to 1/When: filter is called/Then: the filter return value computed from weigthed sum of last two outputs
- Passed: test-filter.global.Scenario: The coefficients are used in correct order/Given: A filter struct is manually created with different output parameters and the last output is set to 1/When: filter is called/Then: the filter return value computed from weigthed sum of last three outputs

- Passed: test-filter.global.Scenario: The coefficients are used in correct order/Given: A filter struct is manually created with different output parameters and the last output is set to 1/When: filter is called/Then: the filter return value computed from weigthed sum of last three outputs
- Passed: test-filter.global.Scenario: An FIR exponential lowpass filter can be used/Given: An options structure filled with values for FIR exponential lowpass filters/When: the create function is called
- Passed: test-filter.global.Scenario: An FIR exponential lowpass filter can be used/Given: An options structure filled with values for FIR exponential lowpass filters/When: the create function is called/Then: the buffers are initialized
- Passed: test-filter.global.Scenario: An FIR exponential lowpass filter can be used/Given: An options structure filled with values for FIR exponential lowpass filters/When: the create function is called/Then: The filter can be initialized
- Passed: test-filter.global.Scenario: An FIR exponential lowpass filter can be used/Given: An options structure filled with values for FIR exponential lowpass filters/When: the create function is called/Then: an impulse function can be filtered
- Passed: test-filter.global.Scenario: An FIR exponential lowpass filter can be used/Given: An options structure filled with values for FIR exponential lowpass filters/When: the create function is called/Then: an impulse function can be filtered
- Passed: test-filter.global.Scenario: An FIR exponential lowpass filter can be used/Given: An options structure filled with values for FIR exponential lowpass filters/When: the create function is called/Then: an impulse function can be filtered
- Passed: test-filter.global.Scenario: An FIR exponential lowpass filter can be used/Given: An options structure filled with values for FIR exponential lowpass filters/When: the create function is called/Then: an impulse function can be filtered
- Passed: test-filter.global.Scenario: An IIR exponential lowpass filter can be used/Given: An options structure filled with values for IIR exponential lowpass filters/When: the create function is called
- Passed: test-filter.global.Scenario: An IIR exponential lowpass filter can be used/Given: An options structure filled with values for IIR exponential lowpass filters/When: the create function is called/Then: the buffers are initialized
- Passed: test-filter.global.Scenario: An IIR exponential lowpass filter can be used/Given: An options structure filled with values for IIR exponential lowpass filters/When: the create function is called/Then: The filter can be initialized
- Passed: test-filter.global.Scenario: An IIR exponential lowpass filter can be used/Given: An options structure filled with values for IIR exponential lowpass filters/When: the create function is called/Then: a step function can be filtered
- Passed: test-filter.global.Scenario: An IIR exponential lowpass filter can be used/Given: An options structure filled with values for IIR exponential lowpass filters/When: the create function is called/Then: a step function can be filtered
- Passed: test-filter.global.Scenario: S-plane pole coefficients can be computed/Given: An array of size 3/When: the s-plane poles are computed/Then: results are correct

- Passed: test-filter.global.Scenario: S-plane pole coefficients can be computed/Given: An array of size 4/When: the s-plane poles are computed/Then: results are correct
- Passed: test-filter.global.Scenario: Convolution of parameters/Given: Two Arrays of size 3/When: the convolution is computed/Then: the result is correct
- Passed: test-filter.global.Scenario: The cpp bindings can be used/Given: An options structure filled with values for FIR exponential lowpass filters/When: the create function is called/Then: response can be computed
- Passed: test-filter.global.Scenario: The cpp bindings can be used/Given: An options structure filled with values for FIR moving average lowpass filters/When: the create function is called/Then: response can be computed
- Passed: test-filter.global.Scenario: The cpp bindings can be used/Given: An options structure filled with values for FIR moving average highpass filters/When: the create function is called/Then: response can be computed

Test suite "Linux-c++"
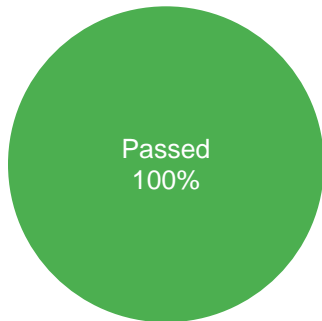
Passed
100%

| Tests: 25 | Passed: 25 | Failed: 0 | Blocked: 0 |

Part of: Unit Tests
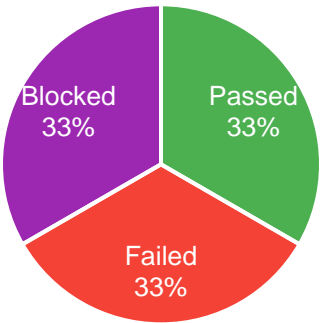File: example/data/example4.xml
Timestamp: 2023-06-08 14:40:14.000
Duration: 0:00:04.000000

- Passed: Scenario: Brickwall lowpass filters can be initialized.Scenario: Brickwall lowpass filters can be initialized
- Passed: Scenario: Brickwall highpass filters can be initialized.Scenario: Brickwall highpass filters can be initialized
- Passed: Scenario: Brickwall bandstop filters can be initialized.Scenario: Brickwall bandstop filters can be initialized
- Passed: Scenario: Brickwall bandpass filters can be initialized.Scenario: Brickwall bandpass filters can be initialized
- Passed: Scenario: Butterworth output coefficients can be computed.Scenario: Butterworth output coefficients can be computed
- Passed: Scenario: Butterworth lowpass filters can be initialized.Scenario: Butterworth lowpass filters can be initialized
- Passed: Scenario: Butterworth highpass filters can be initialized.Scenario: Butterworth highpass filters can be initialized
- Passed: Scenario: Butterworth bandstop filters can be initialized.Scenario: Butterworth bandstop filters can be initialized
- Passed: Scenario: Butterworth bandpass filters can be initialized.Scenario: Butterworth bandpass filters can be initialized
- Passed: Scenario: Chebyshev output coefficients can be computed.Scenario: Chebyshev output coefficients can be computed
- Passed: Scenario: Chebyshev low pass filters are correctly initialized.Scenario: Chebyshev low pass filters are correctly initialized
- Passed: Scenario: Chebyshev high pass filters are correctly initialized.Scenario: Chebyshev high pass filters are correctly initialized
- Passed: Scenario: Chebyshev band pass filters are correctly initialized.Scenario: Chebyshev band pass filters are correctly initialized
- Passed: Scenario: Chebyshev band stop filters are correctly initialized.Scenario: Chebyshev band stop filters are correctly initialized
- Passed: Scenario: Chebyshev type 2 design.Scenario: Chebyshev type 2 design
- Passed: Scenario: A moving average filter can be created.Scenario: A moving average filter can be created

- Passed: Scenario: A moving average filter can be used.Scenario: A moving average filter can be used
- Passed: Scenario: A moving average highpass filter can be used.Scenario: A moving average highpass filter can be used
- Passed: Scenario: An buttwerworth lowpass filter generated on the webpage can be used.Scenario: An buttwerworth lowpass filter generated on the webpage can be used
- Passed: Scenario: The coefficients are used in correct order.Scenario: The coefficients are used in correct order
- Passed: Scenario: An FIR exponential lowpass filter can be used.Scenario: An FIR exponential lowpass filter can be used
- Passed: Scenario: An IIR exponential lowpass filter can be used.Scenario: An IIR exponential lowpass filter can be used
- Passed: Scenario: S-plane pole coefficients can be computed.Scenario: S-plane pole coefficients can be computed
- Passed: Scenario: Convolution of parameters.Scenario: Convolution of parameters
- Passed: Scenario: The cpp bindings can be used.Scenario: The cpp bindings can be used

Mutation Tests

Blocked 33%
Passed 33%
Failed 33%

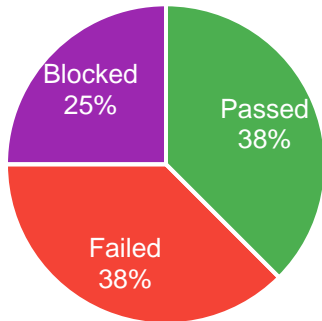| Tests: 72 | Passed: 24 | Failed: 24 | Blocked: 24 |

Part of: Mutation Tests

- example/data/example1.xml   passed: 24 / 72

Test suite "builtin.and"

Tests: 8   Passed: 3   Failed: 3   Blocked: 2

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:03.210000

- Passed: example/source.dart.Line8_builtin.and_0
- Failed: example/source.dart.Line6_builtin.and_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source.dart
  Line: 6
  Original line: if (a == b && (a < c || b > c || b == c)) {
  Mutation: if (a == b || (a < c || b > c || b == c)) {

- Passed: example/source2.dart.Line29_builtin.and_0
- Passed: example/source2.dart.Line31_builtin.and_0
- Failed: example/source2.dart.Line22_builtin.and_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source2.dart
  Line: 22
  Original line: } else if (y == 0.0 && text != '') {
  Mutation: } else if (y == 0.0 || text != '') {

- Failed: example/source2.dart.Line31_builtin.and_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source2.dart
  Line: 31
  Original line: (event.c || event.d || (event.f && event.g)) &&
  Mutation: (event.c || event.d || (event.f && event.g)) ||

- Blocked: example/source2.dart.Line20_builtin.and_0

Type: "timeout"
Message: "The test command timed out after 0.0 s"

File: example/source2.dart
Line: 20
Original line: if (y <= 0.0 && text != '') {
Mutation: if (y <= 0.0 || text != '') {

● Blocked: example/source2.dart.Line30_builtin.and_0

Type: "timeout"
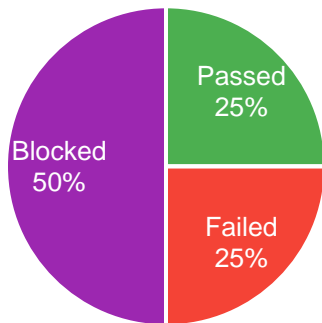Message: "The test command timed out after 0.0 s"

File: example/source2.dart
Line: 30
Original line: event.b &&
Mutation: event.b ||

Test suite "builtin.or"

Tests: 4	Passed: 1	Failed: 1	Blocked: 2

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:01.002000

● Passed: example/source.dart.Line6_builtin.or_0
● Blocked: example/source.dart.Line6_builtin.or_0

Type: "timeout"
Message: "The test command timed out after 0.0 s"

File: example/source.dart
Line: 6
Original line: if (a == b && (a < c || b > c || b == c)) {
Mutation: if (a == b && (a < c && b > c || b == c)) {

● Failed: example/source2.dart.Line31_builtin.or_0

Type: "undetected"
Message: "All tests passed despite changing the code!"

File: example/source2.dart
Line: 31
Original line: (event.c || event.d || (event.f && event.g)) &&
Mutation: (event.c || event.d && (event.f && event.g)) &&

● Blocked: example/source2.dart.Line31_builtin.or_0

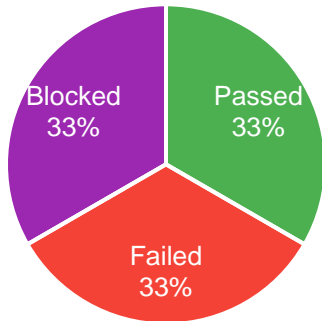Type: "timeout"
Message: "The test command timed out after 0.0 s"

File: example/source2.dart
Line: 31
Original line: (event.c || event.d || (event.f && event.g)) &&
Mutation: (event.c && event.d || (event.f && event.g)) &&

Test suite "builtin.op.eq"

Tests: 3   Passed: 1   Failed: 1   Blocked: 1
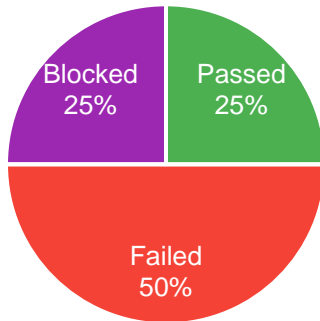
Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

● Failed: example/source.dart.Line6_builtin.op.eq_0

   Type: "undetected"
   Message: "All tests passed despite changing the code!"

   File: example/source.dart
   Line: 6
   Original line: if (a == b && (a < c || b > c || b == c)) {
   Mutation: if (a != b && (a < c || b > c || b == c)) {

● Blocked: example/source.dart.Line6_builtin.op.eq_0

   Type: "timeout"
   Message: "The test command timed out after 0.0 s"

   File: example/source.dart
   Line: 6
   Original line: if (a == b && (a < c || b > c || b == c)) {
   Mutation: if (a == b && (a < c || b > c || b != c)) {

● Passed: example/source2.dart.Line22_builtin.op.eq_0

Test suite "builtin.op.leq"

Tests: 4   Passed: 1   Failed: 2   Blocked: 1

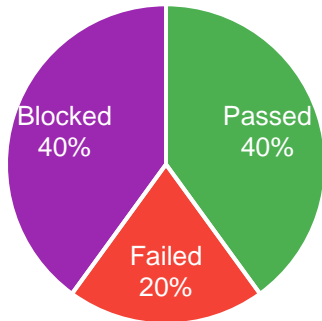Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

● Passed: example/source.dart.Line8_builtin.op.leq_1
● Failed: example/source.dart.Line8_builtin.op.leq_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source.dart
  Line: 8
  Original line: } else if (b <= 0 && c > 0) {
  Mutation: } else if (b == 0 && c > 0) {

● Failed: example/source2.dart.Line20_builtin.op.leq_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source2.dart
  Line: 20
  Original line: if (y <= 0.0 && text != '') {
  Mutation: if (y == 0.0 && text != '') {

● Blocked: example/source2.dart.Line20_builtin.op.leq_1

  Type: "timeout"
  Message: "The test command timed out after 0.0 s"

  File: example/source2.dart
  Line: 20
  Original line: if (y <= 0.0 && text != '') {
  Mutation: if (y < 0.0 && text != '') {

Test suite "builtin.if"

Blocked 40%
Passed 40%
Failed 20%

Tests: 5   Passed: 2   Failed: 1   Blocked: 2

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

- Passed: example/source.dart.Line8_builtin.if_0
- Blocked: example/source.dart.Line6_builtin.if_0

  Type: "timeout"
  Message: "The test command timed out after 0.0 s"

  File: example/source.dart
  Line: 6
  Original line: if (a == b && (a < c || b > c || b == c)) {
  Mutation: if (!(a == b && (a < c || b > c || b == c))) {

- Passed: example/source2.dart.Line22_builtin.if_0
- Failed: example/source2.dart.Line20_builtin.if_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source2.dart
  Line: 20
  Original line: if (y <= 0.0 && text != '') {
  Mutation: if (!(y <= 0.0 && text != '')) {

- Blocked: example/source2.dart.Line29_builtin.if_0

Type: "timeout"
Message: "The test command timed out after 0.0 s"
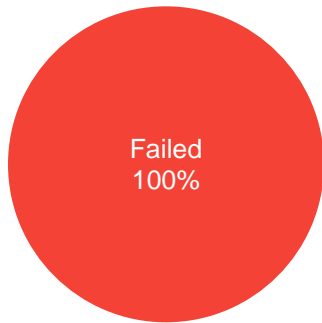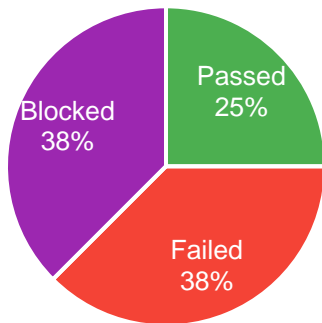
File: example/source2.dart
Line: 29
Original line: if (event.a &&
    event.b &&
    (event.c || event.d || (event.f && event.g)) &&
    event.e) {
Mutation: if (!(event.a &&
    event.b &&
    (event.c || event.d || (event.f && event.g)) &&
    event.e)) {

Test suite "builtin.logical.or_chain"

Failed
100%

| Tests: 2 | Passed: 0 | Failed: 2 | Blocked: 0 |

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

- Failed: example/source.dart.Line6_builtin.logical.or_chain_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source.dart
  Line: 6
  Original line: if (a == b && (a < c || b > c || b == c)) {
  Mutation: if (a == b && (a < c ||!(b > c)|| b == c)) {

- Failed: example/source2.dart.Line31_builtin.logical.or_chain_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source2.dart
  Line: 31
  Original line: (event.c || event.d || (event.f && event.g)) &&
  Mutation: (event.c ||!(event.d)|| (event.f && event.g)) &&

Test suite "builtin.logical.and_chain2"

Blocked 38%
Passed 25%
Failed 38%

Tests: 8   Passed: 2   Failed: 3   Blocked: 3

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

- Failed: example/source.dart.Line8_builtin.logical.and_chain2_1

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source.dart
  Line: 8
  Original line: } else if (b <= 0 && c > 0) {
  Mutation: } else if (b <= 0 &&!(c > 0)) {

- Blocked: example/source.dart.Line8_builtin.logical.and_chain2_0

  Type: "timeout"
  Message: "The test command timed out after 0.0 s"

  File: example/source.dart
  Line: 8
  Original line: } else if (b <= 0 && c > 0) {
  Mutation: } else if (!(b <= 0)&& c > 0) {

- Passed: example/source2.dart.Line22_builtin.logical.and_chain2_0
- Passed: example/source2.dart.Line31_builtin.logical.and_chain2_0
- Failed: example/source2.dart.Line20_builtin.logical.and_chain2_1

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source2.dart
  Line: 20
  Original line: if (y <= 0.0 && text != '') {
  Mutation: if (y <= 0.0 &&!(text != '')) {

- Failed: example/source2.dart.Line31_builtin.logical.and_chain2_1

Type: "undetected"
Message: "All tests passed despite changing the code!"

File: example/source2.dart
Line: 31
Original line: (event.c || event.d || (event.f && event.g)) &&
Mutation: (event.c || event.d || (event.f &&!(event.g))) &&

● Blocked: example/source2.dart.Line20_builtin.logical.and_chain2_0

Type: "timeout"
Message: "The test command timed out after 0.0 s"

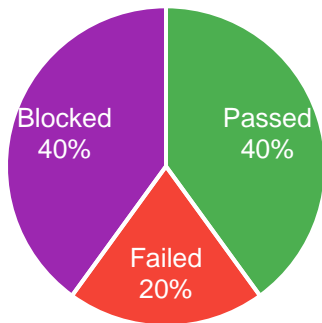File: example/source2.dart
Line: 20
Original line: if (y <= 0.0 && text != '') {
Mutation: if (!(y <= 0.0)&& text != '') {

● Blocked: example/source2.dart.Line22_builtin.logical.and_chain2_1

Type: "timeout"
Message: "The test command timed out after 0.0 s"

File: example/source2.dart
Line: 22
Original line: } else if (y == 0.0 && text != '') {
Mutation: } else if (y == 0.0 &&!(text != '')) {

Test suite "builtin.if.start"

Blocked 40% Passed 40% Failed 20%

| Tests: 5 | Passed: 2 | Failed: 1 | Blocked: 2 |

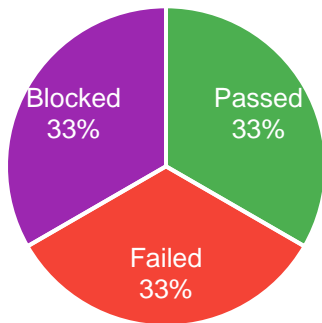Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

● Passed: example/source.dart.Line6_builtin.if.start_0
● Blocked: example/source.dart.Line8_builtin.if.start_0

  Type: "timeout"
  Message: "The test command timed out after 0.0 s"

  File: example/source.dart
  Line: 8
  Original line: } else if (b <= 0 && c > 0) {
  Mutation: } else if (!(b <= 0)&& c > 0) {

● Passed: example/source2.dart.Line29_builtin.if.start_0
● Failed: example/source2.dart.Line22_builtin.if.start_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source2.dart
  Line: 22
  Original line: } else if (y == 0.0 && text != '') {
  Mutation: } else if (!(y == 0.0)&& text != '') {

● Blocked: example/source2.dart.Line20_builtin.if.start_0

  Type: "timeout"
  Message: "The test command timed out after 0.0 s"

  File: example/source2.dart
  Line: 20
  Original line: if (y <= 0.0 && text != '') {
  Mutation: if (!(y <= 0.0)&& text != '') {

Test suite "builtin.if.end"

Tests: 6   Passed: 2   Failed: 2   Blocked: 2

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

● Passed: example/source.dart.Line6_builtin.if.end_0
● Failed: example/source.dart.Line8_builtin.if.end_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source.dart
  Line: 8
  Original line: } else if (b <= 0 && c > 0) {
  Mutation: } else if (b <= 0 &&!(c > 0)) {

● Passed: example/source2.dart.Line22_builtin.if.end_0
● Failed: example/source2.dart.Line31_builtin.if.end_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source2.dart
  Line: 31
  Original line: (event.c || event.d || (event.f && event.g)) &&
  Mutation: (event.c || event.d || (event.f &&!(event.g))) &&

● Blocked: example/source2.dart.Line20_builtin.if.end_0

  Type: "timeout"
  Message: "The test command timed out after 0.0 s"

  File: example/source2.dart
  Line: 20
  Original line: if (y <= 0.0 && text != '') {
  Mutation: if (y <= 0.0 &&!(text != '')) {

● Blocked: example/source2.dart.Line31_builtin.if.end_0

Type: "timeout"
Message: "The test command timed out after 0.0 s"
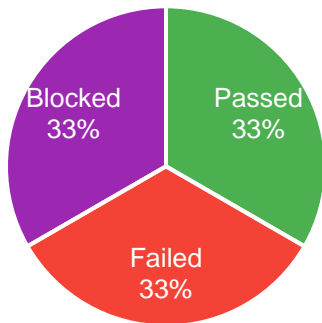
File: example/source2.dart
Line: 31
Original line: (event.c || event.d || (event.f && event.g)) &&
    event.e) {
Mutation: (event.c || event.d || (event.f && event.g)) &&!(
    event.e)) {

Test suite "builtin.logical.chain_not"

Tests: 3   Passed: 1   Failed: 1   Blocked: 1

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

- Blocked: example/source.dart.Line6_builtin.logical.chain_not_0

  Type: "timeout"
  Message: "The test command timed out after 0.0 s"

  File: example/source.dart
  Line: 6
  Original line: if (a == b && (a < c || b > c || b == c)) {
  Mutation: if (a == b &&!(a < c || b > c || b == c)) {
- Passed: example/source2.dart.Line31_builtin.logical.chain_not_0
- Failed: example/source2.dart.Line30_builtin.logical.chain_not_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"
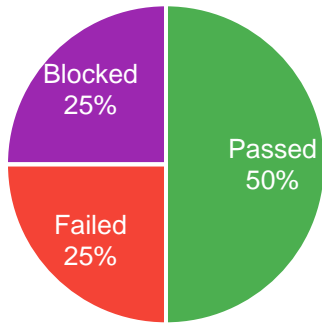
  File: example/source2.dart
  Line: 30
  Original line: event.b &&
      (event.c || event.d || (event.f && event.g)) &&
  Mutation: event.b &&!(event.c || event.d || (event.f && event.g)) &&

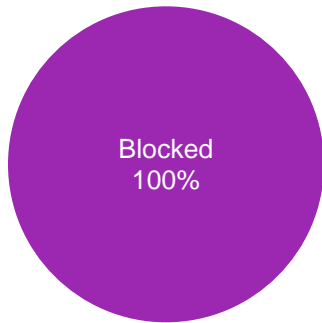# Test suite "builtin.number.negative"

Tests: 4   Passed: 2   Failed: 1   Blocked: 1

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

- Passed: example/source.dart.Line28_builtin.number.negative_0
- Failed: example/source.dart.Line24_builtin.number.negative_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source.dart
  Line: 24
  Original line: return poly(x, y, z, 2.0);
  Mutation: return poly(x, y, z, -2.0);

- Passed: example/source2.dart.Line11_builtin.number.negative_0
- Blocked: example/source2.dart.Line10_builtin.number.negative_0

  Type: "timeout"
  Message: "The test command timed out after 0.0 s"

  File: example/source2.dart
  Line: 10
  Original line: double number1 = 25.0;
  Mutation: double number1 = -25.0;

Test suite "builtin.function.arg2"

Blocked
100%

| Tests: 1 | Passed: 0 | Failed: 0 | Blocked: 1 |

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

● Blocked: example/source.dart.Line32_builtin.function.arg2_0

Type: "timeout"
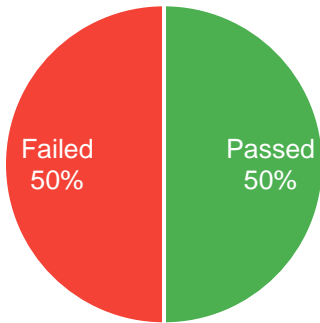Message: "The test command timed out after 0.0 s"

File: example/source.dart
Line: 32
Original line: return inner(x, y);
Mutation: return inner(y,x);

Test suite "builtin.function.arg3"

Tests: 2   **Passed: 1**   **Failed: 1**   **Blocked: 0**

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

- Passed: example/source.dart.Line28_builtin.function.arg3_0
- Failed: example/source.dart.Line28_builtin.function.arg3_1

Type: "undetected"
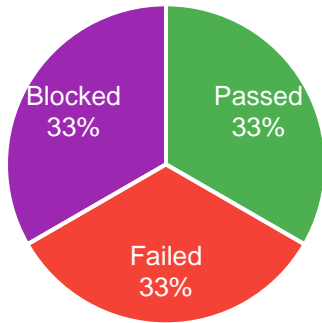Message: "All tests passed despite changing the code!"

File: example/source.dart
Line: 28
Original line: return inner2(x, y, 1.0);
Mutation: return inner2(x, 1.0, y);

Test suite "builtin.function.arg4"

Blocked 33%
Passed 33%
Failed 33%

Tests: 3   Passed: 1   Failed: 1   Blocked: 1

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

- Passed: example/source.dart.Line24_builtin.function.arg4_2
- Failed: example/source.dart.Line24_builtin.function.arg4_1

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source.dart
  Line: 24
  Original line: return poly(x, y, z, 2.0);
  Mutation: return poly(x, z, y, 2.0);

- Blocked: example/source.dart.Line24_builtin.function.arg4_0

  Type: "timeout"
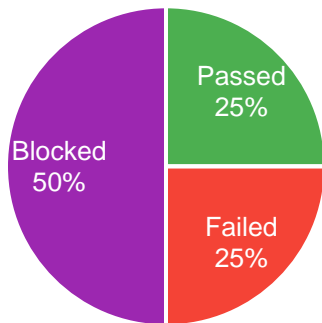 Message: "The test command timed out after 0.0 s"

  File: example/source.dart
  Line: 24
  Original line: return poly(x, y, z, 2.0);
  Mutation: return poly(y,x, z, 2.0);

Test suite "builtin.arith.add"

Tests: 4	Passed: 1	Failed: 1	Blocked: 2

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

Pie chart:
Blocked 50%
Passed 25%
Failed 25%

- Passed: example/source.dart.Line16_builtin.arith.add_0
- Failed: example/source.dart.Line20_builtin.arith.add_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source.dart
  Line: 20
  Original line: return a * x * x + b * x + c;
  Mutation: return a * x * x - b * x + c;

- Blocked: example/source.dart.Line7_builtin.arith.add_0

  Type: "timeout"
  Message: "The test command timed out after 0.0 s"

  File: example/source.dart
  Line: 7
  Original line: return a + c;
  Mutation: return a - c;

- Blocked: example/source.dart.Line20_builtin.arith.add_0

  Type: "timeout"
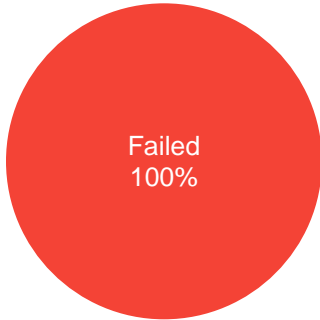 Message: "The test command timed out after 0.0 s"

  File: example/source.dart
  Line: 20
  Original line: return a * x * x + b * x + c;
  Mutation: return a * x * x + b * x - c;

Test suite "builtin.arith.sub"

Failed
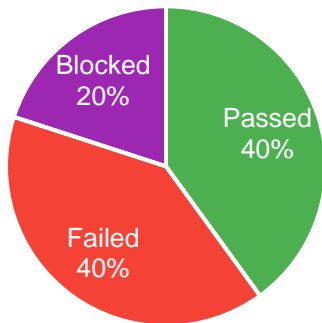100%

| Tests: 1 | Passed: 0 | Failed: 1 | Blocked: 0 |

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

- Failed: example/source.dart.Line9_builtin.arith.sub_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source.dart
  Line: 9
  Original line: return a - b;
  Mutation: return a + b;

Test suite "builtin.arith.mul"

Tests: 5   Passed: 2   Failed: 2   Blocked: 1

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

- Passed: example/source.dart.Line16_builtin.arith.mul_0
- Passed: example/source.dart.Line20_builtin.arith.mul_0
- Failed: example/source.dart.Line20_builtin.arith.mul_0

  Type: "undetected"
  Message: "All tests passed despite changing the code!"

  File: example/source.dart
  Line: 20
  Original line: return a * x * x + b * x + c;
  Mutation: return a * x * x + b / x + c;

- Blocked: example/source.dart.Line20_builtin.arith.mul_0

  Type: "timeout"
  Message: "The test command timed out after 0.0 s"

  File: example/source.dart
  Line: 20
  Original line: return a * x * x + b * x + c;
  Mutation: return a / x * x + b * x + c;

- Failed: example/source2.dart.Line15_builtin.arith.mul_0

  Type: "undetected"
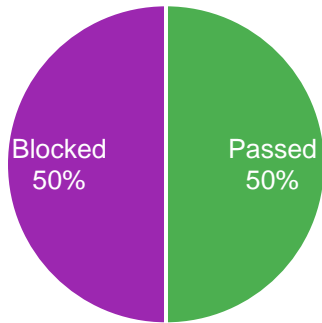 Message: "All tests passed despite changing the code!"

  File: example/source2.dart
  Line: 15
  Original line: return number1 * x / number2;
  Mutation: return number1 / x / number2;

Test suite "builtin.op.neq"

Tests: 2   Passed: 1   Failed: 0   Blocked: 1

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

- Passed: example/source2.dart.Line22_builtin.op.neq_0
- Blocked: example/source2.dart.Line20_builtin.op.neq_0

Type: "timeout"
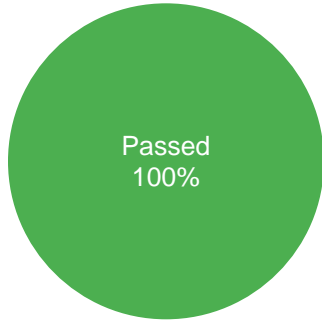Message: "The test command timed out after 0.0 s"

File: example/source2.dart
Line: 20
Original line: if (y <= 0.0 && text != '') {
Mutation: if (y <= 0.0 && text == '') {

Test suite "builtin.logical.and_chain"

Passed
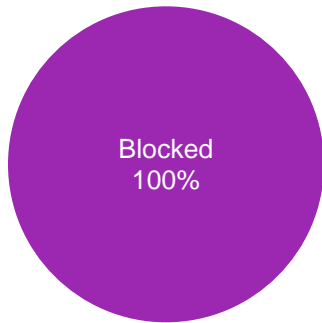100%

| Tests: 1 | Passed: 1 | Failed: 0 | Blocked: 0 |

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

- Passed: example/source2.dart.Line29_builtin.logical.and_chain_0

Test suite "builtin.arith.div"

Blocked
100%

| Tests: 1 | Passed: 0 | Failed: 0 | Blocked: 1 |

Part of: Mutation Tests
File: example/data/example1.xml
Timestamp: 2023-06-03 15:43:06.000
Duration: 0:00:00.000000

- Blocked: example/source2.dart.Line15_builtin.arith.div_0

  Type: "timeout"
  Message: "The test command timed out after 0.0 s"

  File: example/source2.dart
  Line: 15
  Original line: return number1 * x / number2;
  Mutation: return number1 * x * number2;