

root

[Go Up](#)

Table of Contents

example.ecl
Basic Example with :
example_10.ecl
example_11.ecl
example_2.ecl
Basic Inheritance documentation : mod_3 inherits both mod_1 and mod_2
example_3.ecl
Documentation Testing Multiline Title
example_4.ecl
Example : Inheritance across files
example_7.ecl
Basic Type Example
Math.ecl
test.ecl
test module

example

[Go Up](#)

DESCRIPTIONS

MODULE example

	example
--	---------

Basic Example with : records, interface, function, modules, transform, embed, macros and functionmacro

Children

1. [rec_1](#)
 2. [rec_2](#)
 3. [interface_ex](#)
 4. [func_1](#)
 5. [func_2](#)
 6. [mod_1](#)
 7. [mod_2](#)
 8. [cpp_1](#)
 9. [funcmacro_1](#)
 10. [macro_1](#)
 11. [macro_2](#)
-

RECORD rec_1

example \

	rec_1
--	-------

RECORD rec_2

example \

	rec_2
--	-------

INTERFACE interface_ex

example \

	interface_ex
--	--------------

Children

1. iface_v3

ATTRIBUTE iface_v3

example \ interface_ex \

STRING25	iface_v3
----------	----------

FUNCTION func_1

example \

	func_1
(REAL8 x, STRING25 y)	

FUNCTION func_2

example \

DATASET(rec_2)	func_2
(DATASET(rec_1) d)	

MODULE mod_1

example \

	mod_1
(REAL8 a)	

Children

1. pi_w
-

ATTRIBUTE pi_w

example \ mod_1 \

	pi_w
--	------

MODULE mod_2

example \

	mod_2
--	-------

Children

1. pi_wo

ATTRIBUTE pi_wo

example \ mod_2 \

	pi_wo
--	-------

EMBED cpp_1

example \

DATA	cpp_1
(REAL8 varcpp)	

MACRO funcmacro_1

example \

	funcmacro_1
(num)	

MACRO macro_1

example \

	macro_1
(num_1, num_2)	

MACRO macro_2

example \

	macro_2
--	---------

example__10

[Go Up](#)

IMPORTS

intest |

DESCRIPTIONS

MODULE example__10

	example__10
--	-------------

Children

- 1. [mod_1](#)

MODULE mod_1

[example__10](#) \

	mod_1
--	-------

INHERITED True

example__11

[Go Up](#)

IMPORTS

Inintest | [Example__3](#) | intest.[Example__3](#) | intest.inintest.[Example__3](#) |
Inintest.[Example__3](#) |

DESCRIPTIONS

MODULE [example__11](#)

	example__11
--	-----------------------------

Children

1. [Example__3](#)

MODULE [Example__3](#)

[example__11](#) \

	Example__3
--	----------------------------

INHERITED True

example__2

[Go Up](#)

DESCRIPTIONS

MODULE example__2

	example__2
--	------------

Basic Inheritance documentation : mod__3 inherits both mod__1 and mod__2 . Inherits v2__m1, v2__m2, Overrides v1__m1, new locals v2__m3 . Interface Inheritance : mod__4 inherits interface iface__1, overrides v1__i1

Children

1. [rec__1](#)
2. [rec__2](#)
3. [rec__3](#)
4. [mod__1](#)
5. [mod__2](#)
6. [mod__3](#)
7. [iface__1](#)
8. [mod__4](#)

RECORD rec__1

[example__2 \](#)

	rec_1
--	-------

RECORD rec_2

[example_2](#) \

	rec_2
--	-------

RECORD rec_3

[example_2](#) \

	rec_3
--	-------

MODULE mod_1

[example_2](#) \

	mod_1
--	-------

Children

- 1. [v1_m1](#)
- 2. [v2_m1](#)

ATTRIBUTE v1_m1

[example_2](#) \ [mod_1](#) \

real8	v1_m1
-------	-------

ATTRIBUTE v2_m1

example_2 \ mod_1 \

	v2_m1
--	-------

MODULE mod_2

example_2 \

	mod_2
--	-------

Children

- 1. v1_m1
 - 2. v2_m2
-

ATTRIBUTE v1_m1

example_2 \ mod_2 \

	v1_m1
--	-------

ATTRIBUTE v2_m2

example_2 \ mod_2 \

	v2_m2
--	-------

MODULE mod_3

example_2 \

	mod_3
--	-------

Children

- 1. v2_m1
- 2. v2_m2
- 3. v1_m1
- 4. v2_m3

ATTRIBUTE v2_m1

example_2 \ mod_3 \

	v2_m1
--	-------

INHERITED True

ATTRIBUTE v2_m2

example_2 \ mod_3 \

	v2_m2
--	-------

INHERITED True

ATTRIBUTE v1_m1

[example_2 \ mod_3 \](#)

	v1_m1
--	-------

OVERRIDE True

ATTRIBUTE v2_m3

[example_2 \ mod_3 \](#)

	v2_m3
--	-------

INTERFACE iface_1

[example_2 \](#)

	iface_1
--	---------

Children

1. [v1_i1](#)
-

ATTRIBUTE v1_i1

example_2 \ iface_1 \

real8	v1_i1
-------	-------

MODULE mod_4

example_2 \

	mod_4
--	-------

Children

1. v1_i1
 2. v2_m4
-

ATTRIBUTE v1_i1

example_2 \ mod_4 \

	v1_i1
--	-------

OVERRIDE True

ATTRIBUTE v2_m4

example_2 \ mod_4 \

STRING20	v2_m4
----------	-------

example__3

[Go Up](#)

DESCRIPTIONS

MODULE Example__3

	Example__3
--	------------

Documentation Testing Multiline Title. [link@myspace.com](#)

Sentence 1 blablalbla bbbblaaaa

Sentence 2

blablalbla bbbblaaaaa

bbblaaaaaaaaa

PARAMETER first okay__1

PARAMETER second okay__2

PARAMETER third okay__3

FIELD f1 oka_f1

FIELD f2 oka_f2

RETURN rec__1

SEE [example__1.mod__1](#)

Children

1. [mod__1](#)

MODULE mod_1

[Example_3](#) \

	mod_1
--	-------

Children

1. [v1_m1](#) : Doc test 2
2. [v2_m1_ex3](#) : DOC Test 3
3. [long_name](#)

ATTRIBUTE v1_m1

[Example_3](#) \ [mod_1](#) \

	v1_m1
--	-------

Doc test 2. Title end by period not newline

```
ABCD  ||||  
CDEF  ||||
```

ATTRIBUTE v2_m1_ex3

[Example_3](#) \ [mod_1](#) \

	v2_m1_ex3
--	-----------

DOC Test 3 No Period title

FUNCTION long_name

[Example_3](#) \ [mod_1](#) \

	long_name
	(DATASET({REAL8 u}) X, DATASET({REAL8 u}) IntW, DATASET({REAL8 u}) Intb, REAL8 BETA=0.1, REAL8 sparsityParam=0.1 , REAL8 LAMBDA=0.001, REAL8 ALPHA=0.1, UNSIGNED2 MaxIter=100)

example__4

[Go Up](#)

IMPORTS

Inintest.Example__3.mod__1 |

DESCRIPTIONS

MODULE example__4

	example__4
--	------------

Example : Inheritance across files mod__1 in Example__4 inherits mod__1 in Example__3

Children

1. [mod__1](#)

MODULE mod__1

[example__4](#) \

	mod__1
--	--------

Children

1. v2_m1_ex3
2. v2_m1_ex4

ATTRIBUTE v2_m1_ex3

example_4 \ mod_1 \

	v2_m1_ex3
--	-----------

INHERITED True

ATTRIBUTE v2_m1_ex4

example_4 \ mod_1 \

	v2_m1_ex4
--	-----------

example__7

[Go Up](#)

DESCRIPTIONS

MODULE

 example__7

	example__7
--	------------

Basic Type Example Source Code copied from ECL Documentation

Children

- 1. [R](#)

RECORD

 R

[example__7 \](#)

	R
--	---

Math

[Go Up](#)

DESCRIPTIONS

MODULE Math

	Math
--	------

Children

1. [Infinity](#) : Return a real "infinity" value
 2. [NaN](#) : Return a non-signalling NaN (Not a Number)value
 3. [isInfinite](#) : Return whether a real value is infinite (positive or negative)
 4. [isNaN](#) : Return whether a real value is a NaN (not a number) value
 5. [isFinite](#) : Return whether a real value is a valid value (neither infinite not NaN)
 6. [FMod](#) : Returns the floating-point remainder of numer/denom (rounded towards zero)
 7. [FMatch](#) : Returns whether two floating point values are the same, within margin of error epsilon
-

ATTRIBUTE Infinity

[Math](#) \

REAL8	Infinity
--------------	----------

Return a real "infinity" value.

ATTRIBUTE NaN

Math \

REAL8	NaN
-------	-----

Return a non-signalling NaN (Not a Number) value.

FUNCTION isInfinite

Math \

BOOLEAN	isInfinite
(REAL8 val)	

Return whether a real value is infinite (positive or negative).

PARAMETER val The value to test.

FUNCTION isNaN

Math \

BOOLEAN	isNaN
(REAL8 val)	

Return whether a real value is a NaN (not a number) value.

PARAMETER val The value to test.

FUNCTION isFinite

Math \

BOOLEAN	isFinite
(REAL8 val)	

Return whether a real value is a valid value (neither infinite not NaN).

PARAMETER val The value to test.

FUNCTION FMod

Math \

REAL8	FMod
(REAL8 numer, REAL8 denom)	

Returns the floating-point remainder of numer/denom (rounded towards zero). If denom is zero, the result depends on the -fdivideByZero flag: 'zero' or unset: return zero. 'nan': return a non-signalling NaN value 'fail': throw an exception

PARAMETER numer The numerator.

PARAMETER denom The denominator.

FUNCTION FMatch

Math \

BOOLEAN	FMatch
(REAL8 a, REAL8 b, REAL8 epsilon=0.0)	

Returns whether two floating point values are the same, within margin of error epsilon.

PARAMETER a The first value.

PARAMETER b The second value.

PARAMETER epsilon The allowable margin of error.

test

[Go Up](#)

DESCRIPTIONS

MODULE test

	test
--	------

test module
