

# root

---

[Go Up](#)

## Table of Contents

<a href="#">example.ecl</a>
Basic Example with :
<a href="#">example_10.ecl</a>
<a href="#">example_11.ecl</a>
<a href="#">example_2.ecl</a>
Basic Inheritance documentation : mod_3 inherits both mod_1 and mod_2
<a href="#">example_3.ecl</a>
Documentation Testing Multiline Title
<a href="#">example_4.ecl</a>
Example : Inheritance across files
<a href="#">example_7.ecl</a>
Basic Type Example
<a href="#">Math.ecl</a>
<a href="#">test.ecl</a>
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 \

<b>DATA</b>	cpp_1
(REAL8 varcpp)	

---

## **MACRO** funcmacro\_1

example \

	<b>funcmacro_1</b>
(num)	

---

**MACRO** macro\_1

example \

	<b>macro_1</b>
(num_1, num_2)	

---

**MACRO** macro\_2

example \

	<b>macro_2</b>
--	----------------

---

# 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

- [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](#) \

<b>REAL8</b>	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

---