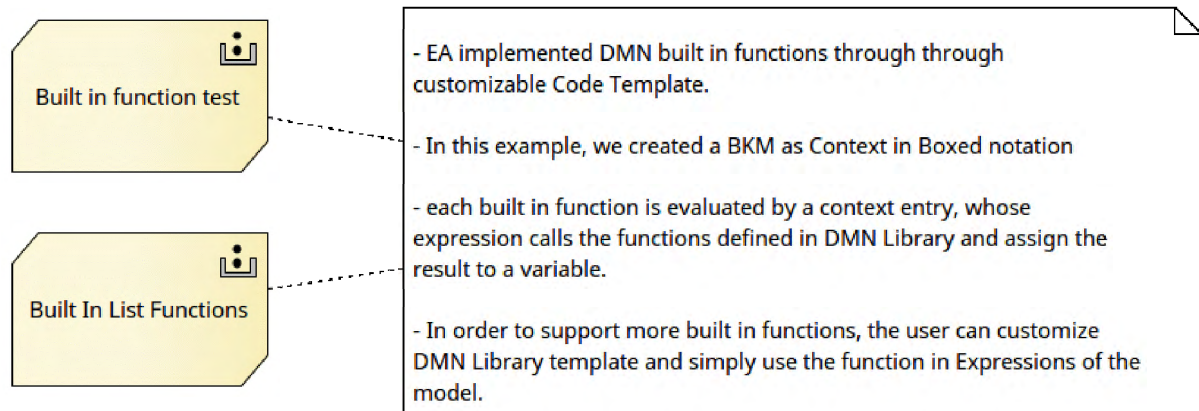


DMN Built In Functions

The DMN Specification defines many built in functions to promote interoperability. Enterprise Architect implements these functions flexibly, through the customizable DMN Library Template.

- Access:
 - DMN Expression View | Simulation Button Menu on Toolbar | Edit DMN Library
 - DMN Simulation View | Simulation Button Menu on Toolbar | Edit DMN Library
 - Ribbon | Code | Configure | Code Template Editor | Language: DMN_JavaScript | DMNSim Library
- In this Package, we have defined two Business Knowledge Models (BKMs) implemented as 'Context' in Boxed notation. Each Context Entry defines a test case for the built in function.



- The first BKM tests built in functions in the 'boolean, number, string' domain and the second BKM tests built in functions for lists.

BusinessKnowledgeModel (Built in function test) □ ×



Built in function test

Input Parameter Values for Simulation

()

PMT test 1	PMT(0.08/12, 10, 10000)
PMT test 2	decimal(PMT(0.08/12, 10, 10000),2)
not test 1	not(true)
not test 2	not(false)
substring test 1	substring("foobar",3)
substring test 2	substring("foobar",3,3)
substring test 3	substring("foobar", -2, 1)
string length tes...	string length("foo")
upper case test 1	upper case("aBc4")
lower case test 1	lower case("aBc4")
substring before ...	substring before("foobar", "bar")
substring before ...	substring before("foobar", "xyz")
substring after t...	substring after("foobar", "ob")
substring after t...	substring after("", "a")
replace test 1	replace("We are all we are.", "we", "you", "gi")
replace test 2	replace("We are all we are.", "we", "you", "g")
replace test 3	replace("We are all we are.", "we", "you", "i")
contains test 1	contains("foobar", "of")
starts with test 1	starts with("foobar", "fo")
ends with test 1	ends with("foobar", "r")
matches test 1	matches("foobar", "^fo*b")
decimal test 1	decimal(1/3, 2)
decimal test 2	decimal(1.5, 0)
decimal test 3	decimal(2.4, 0)
ceiling test 1	ceiling(1.5)
ceiling test 2	ceiling(-1.5)

true

()		
variable01		list contains([1,2,3], 2)
variable02		count([1,2,3])
variable03		min([1,2,3])
variable04		min(1,2,3)
variable05		max([1,2,3])
variable06		max(1,2,3)
variable07		sum([1,2,3])
variable08		sum(1,2,3)
variable09		mean([1,2,3])
variable10		mean(1,2,3)
variable11		and([false,null,true])
variable12		and(false,null,true)
variable13		and([])
variable14		and(0)
variable15		or([false,null,true])
variable16		or(false,null,true)
variable17		or([])
variable18		or(0)
variable19		sublist([1,2,3], 1, 2)
variable20		append([1], 2, 3)
variable21		concatenate([1,2],[3])
variable22		insert before([1,3],1,2)
variable23		remove([1,2,3], 2)
variable24		reverse([1,2,3])
variable25		index of([1,2,3,2],2)
variable26		union([1,2],[2,3], [3,4,5])
variable27		distinct values([1,2,3,2,1])
variable29		flatten([[1,2],[3]], 4)
true		

Simulation

These BKM's do not define any parameters. Evaluate each built in function through a *context entry*, whose expression calls the functions defined in the DMN Library and assigns the result to a variable.

Click the 'Simulation' button on the toolbar; the runtime result will be retrieved and rendered on the variables.

BusinessKnowledgeModel (Built in function test)	
Built in function test	
Input Parameter Values for Simulation	
{ }	
PMT test 1 = 1037.0320893591636	PMT(0.08/12, 10, 10000)
PMT test 2 = 1037.03	decimal(PMT(0.08/12, 10, 10000),2)
not test 1 = false	not(true)
not test 2 = true	not(false)
substring test 1 = obar	substring("foobar",3)
substring test 2 = oba	substring("foobar",3,3)
substring test 3 = a	substring("foobar", -2, 1)
string length test 1 = 3	string length("foo")
upper case test 1 = ABC4	upper case("aBc4")
lower case test 1 = abc4	lower case("aBc4")
substring before test 1 = foo	substring before("foobar", "bar")
substring before test 2	substring before("foobar", "xyz")
substring after test 1 = ar	substring after("foobar", "ob")
substring after test 2	substring after("", "a")
replace test 1 = you are all you are.	replace("We are all we are.", "we", "you", "gi")
replace test 2 = We are all you are.	replace("We are all we are.", "we", "you", "g")
replace test 3 = you are all we are.	replace("We are all we are.", "we", "you", "i")
contains test 1 = false	contains("foobar", "of")
starts with test 1 = true	starts with("foobar", "fo")
ends with test 1 = true	ends with("foobar", "r")
matches test 1 = true	matches("foobar", "^fo*b")
decimal test 1 = 0.33	decimal(1/3, 2)
decimal test 2 = 2	decimal(1.5, 0)
decimal test 3 = 2	decimal(2.4, 0)
ceiling test 1 = 2	ceiling(1.5)
ceiling test 2 = -1	ceiling(-1.5)
true	

	()	
variable01 = true		list contains([1,2,3], 2)
variable02 = 3		count([1,2,3])
variable03 = 1		min([1,2,3])
variable04 = 1		min(1,2,3)
variable05 = 3		max([1,2,3])
variable06 = 3		max(1,2,3)
variable07 = 6		sum([1,2,3])
variable08 = 6		sum(1,2,3)
variable09 = 2		mean([1,2,3])
variable10 = 2		mean(1,2,3)
variable11 = false		and([false,null,true])
variable12 = false		and(false,null,true)
variable13 = true		and([])
variable14 = null		and(0)
variable15 = true		or([false,null,true])
variable16 = true		or(false,null,true)
variable17 = false		or([])
variable18 = null		or(0)
variable19 = [2]		sublist([1,2,3], 1, 2)
variable20 = [1,2,3]		append([1], 2, 3)
variable21 = [1,2,3]		concatenate([1,2],[3])
variable22 = [1,2,3]		insert before([1,3], 1, 2)
variable23 = [1,2]		remove([1,2,3], 2)
variable24 = [3,2,1]		reverse([1,2,3])
variable25 = [1,3]		index of([1,2,3,2], 2)
variable26 = [1,2,3,4,5]		union([1,2],[2,3], [3,4,5])
variable27 = [1,2,3]		distinct values([1,2,3,2,1])
variable29 = [1,2,3,4]		flatten([[1,2],[3]], 4)
	true	

- Tips: In order to support more built in functions, you can customize the DMN Library template and simply use the function in Expressions of the model

Note: For functions with "list index" or "position" types of parameter, the values start from 0. For example, sublist(['a','b','c'], 1, 2) returns a list, starting with list[1] up to but

not including list[2], which is ['b']. Similar functions currently include "sublist", "insert before", "remove" and "index of".