



# Excalibur Standard Libraries Documentation - v 1.4 - (*Report type: Simulation*)

Wednesday 7<sup>th</sup> December, 2016 - 19:54

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>General Description</b>	<b>9</b>
<b>3</b>	<b>Additional Constraints</b>	<b>11</b>
<b>A</b>	<b>Undocumented Messir Specification Elements</b>	<b>13</b>
A.1	Undocumented Primary Types	13
A.1.1	Undocumented Primary Datatype Types	13
A.1.2	Undocumented Primary Primitive Types	13
A.2	Undocumented Operation Specifications	14
<b>B</b>	<b>Messir Specification Files Listing</b>	<b>21</b>
B.1	File /src-gen/messir-spec/.views.msr	21
B.2	File /src-gen/messir-spec/library/calendar.msr	21
B.3	File /src-gen/messir-spec/library/math.msr	23
B.4	File /src-gen/messir-spec/library/primitives.msr	25
B.5	File /src-gen/messir-spec/library/string.msr	27

# List of Figures



# Listings

- B.1 Messir Spec. file `.views.msr`. . . . . 21
- B.2 Messir Spec. file `calendar.msr`. . . . . 21
- B.3 Messir Spec. file `math.msr`. . . . . 23
- B.4 Messir Spec. file `primitives.msr`. . . . . 25
- B.5 Messir Spec. file `string.msr`. . . . . 27



# Chapter 1

## Introduction





## Chapter 2

# General Description



## Chapter 3

# Additional Constraints



# Appendix A

## Undocumented Messir Specification Elements

### A.1 Undocumented Primary Types

#### A.1.1 Undocumented Primary Datatype Types

- `lu.uni.lassy.messir.libraries.calendar.dtDate`
- `lu.uni.lassy.messir.libraries.calendar.dtDateAndTime`
- `lu.uni.lassy.messir.libraries.calendar.dtDay`
- `lu.uni.lassy.messir.libraries.calendar.dtHour`
- `lu.uni.lassy.messir.libraries.math.dtInteger`
- `lu.uni.lassy.messir.libraries.calendar.dtMinute`
- `lu.uni.lassy.messir.libraries.calendar.dtMonth`
- `lu.uni.lassy.messir.libraries.math.dtReal`
- `lu.uni.lassy.messir.libraries.calendar.dtSecond`
- `lu.uni.lassy.messir.libraries.string.dtString`
- `lu.uni.lassy.messir.libraries.calendar.dtTime`
- `lu.uni.lassy.messir.libraries.calendar.dtYear`

#### A.1.2 Undocumented Primary Primitive Types

- `lu.uni.lassy.messir.libraries.primitives.ptBoolean`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger`
- `lu.uni.lassy.messir.libraries.primitives.ptReal`
- `lu.uni.lassy.messir.libraries.primitives.ptString`

## A.2 Undocumented Operation Specifications

- `lu.uni.lassy.messir.libraries.calendar.dtDate.close`
- `lu.uni.lassy.messir.libraries.calendar.dtDate.eq`
- `lu.uni.lassy.messir.libraries.calendar.dtDate.fromSecondsQty`
- `lu.uni.lassy.messir.libraries.calendar.dtDate.gt`
- `lu.uni.lassy.messir.libraries.calendar.dtDate.is`
- `lu.uni.lassy.messir.libraries.calendar.dtDate.isNow`
- `lu.uni.lassy.messir.libraries.calendar.dtDate.lt`
- `lu.uni.lassy.messir.libraries.calendar.dtDate.toSecondsQty`
- `lu.uni.lassy.messir.libraries.calendar.dtDateAndTime.close`
- `lu.uni.lassy.messir.libraries.calendar.dtDateAndTime.eq`
- `lu.uni.lassy.messir.libraries.calendar.dtDateAndTime.fromSecondsQty`
- `lu.uni.lassy.messir.libraries.calendar.dtDateAndTime.gt`
- `lu.uni.lassy.messir.libraries.calendar.dtDateAndTime.is`
- `lu.uni.lassy.messir.libraries.calendar.dtDateAndTime.isNow`
- `lu.uni.lassy.messir.libraries.calendar.dtDateAndTime.lt`
- `lu.uni.lassy.messir.libraries.calendar.dtDateAndTime.toSecondsQty`
- `lu.uni.lassy.messir.libraries.calendar.dtDay.close`
- `lu.uni.lassy.messir.libraries.calendar.dtDay.is`
- `lu.uni.lassy.messir.libraries.calendar.dtHour.close`
- `lu.uni.lassy.messir.libraries.calendar.dtHour.is`
- `lu.uni.lassy.messir.libraries.math.dtInteger.acos`
- `lu.uni.lassy.messir.libraries.math.dtInteger.add`
- `lu.uni.lassy.messir.libraries.math.dtInteger.asdtReal`
- `lu.uni.lassy.messir.libraries.math.dtInteger.asin`
- `lu.uni.lassy.messir.libraries.math.dtInteger.asptInteger`
- `lu.uni.lassy.messir.libraries.math.dtInteger.atan`
- `lu.uni.lassy.messir.libraries.math.dtInteger.close`
- `lu.uni.lassy.messir.libraries.math.dtInteger.cos`
- `lu.uni.lassy.messir.libraries.math.dtInteger.eq`

- `lu.uni.lassy.messir.libraries.math.dtInteger.frac`
- `lu.uni.lassy.messir.libraries.math.dtInteger.geq`
- `lu.uni.lassy.messir.libraries.math.dtInteger.gt`
- `lu.uni.lassy.messir.libraries.math.dtInteger.is`
- `lu.uni.lassy.messir.libraries.math.dtInteger.leq`
- `lu.uni.lassy.messir.libraries.math.dtInteger.lt`
- `lu.uni.lassy.messir.libraries.math.dtInteger.mod`
- `lu.uni.lassy.messir.libraries.math.dtInteger.msrebs`
- `lu.uni.lassy.messir.libraries.math.dtInteger.msrddiv`
- `lu.uni.lassy.messir.libraries.math.dtInteger.mul`
- `lu.uni.lassy.messir.libraries.math.dtInteger.neq`
- `lu.uni.lassy.messir.libraries.math.dtInteger.opp`
- `lu.uni.lassy.messir.libraries.math.dtInteger.power`
- `lu.uni.lassy.messir.libraries.math.dtInteger.sin`
- `lu.uni.lassy.messir.libraries.math.dtInteger.sqr`
- `lu.uni.lassy.messir.libraries.math.dtInteger.sqrt`
- `lu.uni.lassy.messir.libraries.math.dtInteger.sub`
- `lu.uni.lassy.messir.libraries.math.dtInteger.tan`
- `lu.uni.lassy.messir.libraries.math.dtInteger.toDeg`
- `lu.uni.lassy.messir.libraries.math.dtInteger.toRad`
- `lu.uni.lassy.messir.libraries.math.dtInteger.todtString`
- `lu.uni.lassy.messir.libraries.calendar.dtMinute.close`
- `lu.uni.lassy.messir.libraries.calendar.dtMinute.is`
- `lu.uni.lassy.messir.libraries.calendar.dtMonth.close`
- `lu.uni.lassy.messir.libraries.calendar.dtMonth.is`
- `lu.uni.lassy.messir.libraries.math.dtReal.acos`
- `lu.uni.lassy.messir.libraries.math.dtReal.add`
- `lu.uni.lassy.messir.libraries.math.dtReal.asdtInteger`
- `lu.uni.lassy.messir.libraries.math.dtReal.asin`
- `lu.uni.lassy.messir.libraries.math.dtReal.asptReal`

- `lu.uni.lassy.messir.libraries.math.dtReal.atan`
- `lu.uni.lassy.messir.libraries.math.dtReal.close`
- `lu.uni.lassy.messir.libraries.math.dtReal.cos`
- `lu.uni.lassy.messir.libraries.math.dtReal.eq`
- `lu.uni.lassy.messir.libraries.math.dtReal.frac`
- `lu.uni.lassy.messir.libraries.math.dtReal.geq`
- `lu.uni.lassy.messir.libraries.math.dtReal.gt`
- `lu.uni.lassy.messir.libraries.math.dtReal.is`
- `lu.uni.lassy.messir.libraries.math.dtReal.leq`
- `lu.uni.lassy.messir.libraries.math.dtReal.lt`
- `lu.uni.lassy.messir.libraries.math.dtReal.msrebs`
- `lu.uni.lassy.messir.libraries.math.dtReal.msrdv`
- `lu.uni.lassy.messir.libraries.math.dtReal.msround`
- `lu.uni.lassy.messir.libraries.math.dtReal.mul`
- `lu.uni.lassy.messir.libraries.math.dtReal.neq`
- `lu.uni.lassy.messir.libraries.math.dtReal.opp`
- `lu.uni.lassy.messir.libraries.math.dtReal.power`
- `lu.uni.lassy.messir.libraries.math.dtReal.sin`
- `lu.uni.lassy.messir.libraries.math.dtReal.sqr`
- `lu.uni.lassy.messir.libraries.math.dtReal.sqrt`
- `lu.uni.lassy.messir.libraries.math.dtReal.sub`
- `lu.uni.lassy.messir.libraries.math.dtReal.tan`
- `lu.uni.lassy.messir.libraries.math.dtReal.toDeg`
- `lu.uni.lassy.messir.libraries.math.dtReal.toRad`
- `lu.uni.lassy.messir.libraries.math.dtReal.todtString`
- `lu.uni.lassy.messir.libraries.calendar.dtSecond.close`
- `lu.uni.lassy.messir.libraries.calendar.dtSecond.is`
- `lu.uni.lassy.messir.libraries.string.dtString.close`
- `lu.uni.lassy.messir.libraries.string.dtString.dtStringConcat`
- `lu.uni.lassy.messir.libraries.string.dtString.eq`



- `lu.uni.lassy.messir.libraries.string.dtString.geq`
- `lu.uni.lassy.messir.libraries.string.dtString.gt`
- `lu.uni.lassy.messir.libraries.string.dtString.is`
- `lu.uni.lassy.messir.libraries.string.dtString.length`
- `lu.uni.lassy.messir.libraries.string.dtString.leq`
- `lu.uni.lassy.messir.libraries.string.dtString.lt`
- `lu.uni.lassy.messir.libraries.string.dtString.neq`
- `lu.uni.lassy.messir.libraries.string.dtString.subdtString`
- `lu.uni.lassy.messir.libraries.string.dtString.toLower`
- `lu.uni.lassy.messir.libraries.string.dtString.toUpper`
- `lu.uni.lassy.messir.libraries.string.dtString.toptString`
- `lu.uni.lassy.messir.libraries.calendar.dtTime.close`
- `lu.uni.lassy.messir.libraries.calendar.dtTime.eq`
- `lu.uni.lassy.messir.libraries.calendar.dtTime.fromSecondsQty`
- `lu.uni.lassy.messir.libraries.calendar.dtTime.gt`
- `lu.uni.lassy.messir.libraries.calendar.dtTime.is`
- `lu.uni.lassy.messir.libraries.calendar.dtTime.isNow`
- `lu.uni.lassy.messir.libraries.calendar.dtTime.lt`
- `lu.uni.lassy.messir.libraries.calendar.dtTime.toSecondsQty`
- `lu.uni.lassy.messir.libraries.calendar.dtYear.close`
- `lu.uni.lassy.messir.libraries.calendar.dtYear.is`
- `lu.uni.lassy.messir.libraries.primitives.ptBoolean.close`
- `lu.uni.lassy.messir.libraries.primitives.ptBoolean.eq`
- `lu.uni.lassy.messir.libraries.primitives.ptBoolean.is`
- `lu.uni.lassy.messir.libraries.primitives.ptBoolean.msrand`
- `lu.uni.lassy.messir.libraries.primitives.ptBoolean.msrrnot`
- `lu.uni.lassy.messir.libraries.primitives.ptBoolean.msrror`
- `lu.uni.lassy.messir.libraries.primitives.ptBoolean.msrxor`
- `lu.uni.lassy.messir.libraries.primitives.ptBoolean.neq`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.acos`

- `lu.uni.lassy.messir.libraries.primitives.ptInteger.add`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.asin`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.asptReal`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.atan`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.close`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.cos`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.eq`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.frac`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.geq`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.gt`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.is`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.leq`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.lt`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.mod`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.msabs`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.msrdv`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.mul`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.neq`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.opp`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.power`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.sin`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.sqr`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.sqrt`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.sub`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.tan`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.toDeg`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.toRad`
- `lu.uni.lassy.messir.libraries.primitives.ptInteger.toptString`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.acos`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.add`

- `lu.uni.lassy.messir.libraries.primitives.ptReal.asin`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.asptInteger`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.atan`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.close`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.cos`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.eq`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.frac`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.geq`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.gt`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.is`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.leq`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.lt`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.msrebs`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.msrdv`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.msround`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.mul`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.neq`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.opp`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.power`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.sin`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.sqr`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.sqrt`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.sub`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.tan`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.toDeg`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.toRad`
- `lu.uni.lassy.messir.libraries.primitives.ptReal.toptString`
- `lu.uni.lassy.messir.libraries.primitives.ptString.close`
- `lu.uni.lassy.messir.libraries.primitives.ptString.eq`
- `lu.uni.lassy.messir.libraries.primitives.ptString.geq`

- `lu.uni.lassy.messir.libraries.primitives.ptString.gt`
- `lu.uni.lassy.messir.libraries.primitives.ptString.is`
- `lu.uni.lassy.messir.libraries.primitives.ptString.length`
- `lu.uni.lassy.messir.libraries.primitives.ptString.leq`
- `lu.uni.lassy.messir.libraries.primitives.ptString.lt`
- `lu.uni.lassy.messir.libraries.primitives.ptString.neq`
- `lu.uni.lassy.messir.libraries.primitives.ptString.ptStringConcat`
- `lu.uni.lassy.messir.libraries.primitives.ptString.subptString`
- `lu.uni.lassy.messir.libraries.primitives.ptString.toLower`
- `lu.uni.lassy.messir.libraries.primitives.ptString.toUpper`

## Appendix B

# Messir Specification Files Listing

### B.1 File ./src-gen/messir-spec/.views.msr

```
1 //
2 //DON'T TOUCH THIS FILE !!!
3 //
4 package uuid7d4b15133efc45b9b0f503fbb2d93068 {
5   Concept Model {}
6 }
```

Listing B.1: Messir Spec. file .views.msr.

### B.2 File ./src-gen/messir-spec/library/calendar.msr

```
1 /*
2 * Copyright University of Luxembourg
3 *
4 * This file is part of EXCALIBUR.
5 * EXCALIBUR is free software: you can redistribute it and/or modify
6 * it under the terms of the GNU General Public License as published by
7 * the Free Software Foundation, version 3 of the License.
8 *
9 * EXCALIBUR is distributed in the hope that it will be useful,
10 * but WITHOUT ANY WARRANTY; without even the implied warranty of
11 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 * GNU General Public License for more details.
13 *
14 * You should have received a copy of the GNU General Public License
15 * along with EXCALIBUR in the COPYING.txt file.
16 * If not, see <http://www.gnu.org/licenses/>.
17 *
18 * Last Modification:
19 *
20 * @author nicolas.guelfi
21 * @date Mon May 06 18:10:54 CEST 2013
22 */
23
24 package lu.uni.lassy.messir.libraries.calendar{
25
26   import lu.uni.lassy.messir.libraries.primitives
27   import lu.uni.lassy.messir.libraries.math
28
29   Concept Model {
30
31     Primary Types {
32
33       datatype dtHour extends dtInteger {
34         operation is():ptBoolean
35         external operation close() : ptBoolean
36       }
```

```

37  datatype dtMinute extends dtInteger {
38      operation is():ptBoolean
39      external operation close() : ptBoolean
40  }
41  datatype dtSecond extends dtInteger {
42      operation is():ptBoolean
43      external operation close() : ptBoolean
44  }
45
46  datatype dtTime {
47      attribute hour:dtHour
48      attribute minute: dtMinute
49      attribute second: dtSecond
50
51      operation is():ptBoolean
52      external operation close() : ptBoolean
53
54      // Logical Operations
55      operation lt(AddTime:dtTime):ptBoolean
56      operation gt(AddTime:dtTime):ptBoolean
57      operation eq(AddTime:dtTime):ptBoolean
58      external operation isNow():ptBoolean
59
60      // Conversion Operations
61      operation toSecondsQty():dtInteger
62      operation fromSecondsQty(AddInteger:dtInteger):ptBoolean
63  }
64
65  datatype dtYear extends dtInteger {
66      operation is():ptBoolean
67      external operation close() : ptBoolean
68  }
69  datatype dtMonth extends dtInteger {
70      operation is():ptBoolean
71      external operation close() : ptBoolean
72  }
73  datatype dtDay extends dtInteger {
74      operation is():ptBoolean
75      external operation close() : ptBoolean
76  }
77
78  datatype dtDate {
79      attribute year:dtYear
80      attribute month: dtMonth
81      attribute day: dtDay
82
83      operation is():ptBoolean
84      external operation close() : ptBoolean
85
86      // Logical Operations
87      operation lt(AddDate:dtDate):ptBoolean
88      operation gt(AddDate:dtDate):ptBoolean
89      operation eq(AddDate:dtDate):ptBoolean
90      external operation isNow():ptBoolean
91
92      // Conversion Operations
93      operation toSecondsQty():dtInteger
94      operation fromSecondsQty(AddInteger:dtInteger):ptBoolean
95  }
96
97  datatype dtDateAndTime {
98      attribute date:dtDate
99      attribute time: dtTime
100
101      operation is():ptBoolean
102      external operation close() : ptBoolean
103
104      operation lt(AddDateAndTime:dtDateAndTime):ptBoolean
105      operation gt(AddDateAndTime:dtDateAndTime):ptBoolean
106      operation eq(AddDateAndTime:dtDateAndTime):ptBoolean

```

```

107  external operation isNow():ptBoolean
108
109  // Conversion Operations
110  operation toSecondsQty():dtInteger
111  operation fromSecondsQty(AdtInteger:dtInteger):ptBoolean
112  }
113 }
114 }
115 }

```

Listing B.2: Messir Spec. file calendar.msr.

## B.3 File ./src-gen/messir-spec/library/math.msr

```

1 /*
2 * Copyright University of Luxembourg
3 *
4 * This file is part of EXCALIBUR.
5 * EXCALIBUR is free software: you can redistribute it and/or modify
6 * it under the terms of the GNU General Public License as published by
7 * the Free Software Foundation, version 3 of the License.
8 *
9 * EXCALIBUR is distributed in the hope that it will be useful,
10 * but WITHOUT ANY WARRANTY; without even the implied warranty of
11 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 * GNU General Public License for more details.
13 *
14 * You should have received a copy of the GNU General Public License
15 * along with EXCALIBUR in the COPYING.txt file.
16 * If not, see <http://www.gnu.org/licenses/>.
17 *
18 * Last Modification:
19 *
20 * @author nicolas.guelfi
21 * @date Mon May 06 18:10:54 CEST 2013
22 */
23
24 package lu.uni.lassy.messir.libraries.math{
25
26 import lu.uni.lassy.messir.libraries.primitives
27 import lu.uni.lassy.messir.libraries.string
28
29 Concept Model {
30
31 Primary Types {
32
33 datatype dtInteger{
34 attribute value:ptInteger
35 // Type checking Operation
36 operation is():ptBoolean
37 external operation close() : ptBoolean
38
39 // Arithmetic Operations
40 operation add(AdtInteger:dtInteger): dtInteger
41 operation sub(AdtInteger:dtInteger): dtInteger
42 operation mul(AdtInteger:dtInteger): dtInteger
43 operation frac(AdtInteger:dtInteger): dtReal
44 operation msrdiv(AdtInteger:dtInteger): dtInteger
45 operation power(AExp:dtInteger): dtInteger
46 operation mod(AdtInteger:dtInteger): dtInteger
47
48 operation sqrt(): dtReal
49 operation msrabs(): dtInteger
50 operation opp(): dtInteger
51 operation sqr(): dtInteger
52
53 // Logical Operations
54 operation eq(AdtInteger:dtInteger): ptBoolean
55 operation neq(AdtInteger:dtInteger): ptBoolean

```

```

56  operation geq(AdtInteger:dtInteger): ptBoolean
57  operation leq(AdtInteger:dtInteger): ptBoolean
58  operation lt(AdtInteger:dtInteger): ptBoolean
59  operation gt(AdtInteger:dtInteger): ptBoolean
60
61  // Trigonometric Operations
62  // default is radian
63  operation cos(): dtReal
64  operation acos(): dtReal
65  operation tan(): dtReal
66  operation atan(): dtReal
67  operation sin(): dtReal
68  operation asin(): dtReal
69  operation toDeg(): dtReal
70  operation toRad(): dtReal
71
72  // Conversion Operations
73  operation asdtReal():dtReal
74  operation todtString():dtString
75  operation asptInteger():ptInteger
76  }
77
78  datatype dtReal {
79      attribute value:ptReal
80
81      // Type checking Operation
82      operation is():ptBoolean
83      external operation close() : ptBoolean
84
85      // Arithmetic Operations
86      operation add(AdtReal:dtReal): dtReal
87      operation sub(AdtReal:dtReal): dtReal
88      operation mul(AdtReal:dtReal): dtReal
89      operation frac(AdtReal:dtReal) : dtReal
90      operation msrdiv(AdtReal:dtReal): dtInteger
91      operation power(AdtReal:dtReal): dtReal
92
93      operation msrround() : dtInteger
94      operation sqrt(): dtReal
95      operation msrabs(): dtReal
96      operation opp(): dtReal
97      operation sqr(): dtReal
98
99      // Logical Operations
100     operation eq(AdtReal:dtReal): ptBoolean
101     operation neq(AdtReal:dtReal): ptBoolean
102     operation geq(AdtReal:dtReal): ptBoolean
103     operation leq(AdtReal:dtReal): ptBoolean
104     operation lt(AdtReal:dtReal): ptBoolean
105     operation gt(AdtReal:dtReal): ptBoolean
106
107     // Trigonometric Operations
108     // default is radian
109     operation cos(): dtReal
110     operation acos(): dtReal
111     operation tan(): dtReal
112     operation atan(): dtReal
113     operation sin(): dtReal
114     operation asin(): dtReal
115     operation toDeg(): dtReal
116     operation toRad(): dtReal
117
118     // Conversion Operations
119     operation asdtInteger():dtInteger
120     operation todtString() : dtString
121     operation asptReal():ptReal
122 }
123 }
124 }

```



125 }

Listing B.3: Messir Spec. file math.msr.

**B.4 File ./src-gen/messir-spec/library/primitives.msr**

```

1 /*
2 * Copyright University of Luxembourg
3 *
4 * This file is part of EXCALIBUR.
5 * EXCALIBUR is free software: you can redistribute it and/or modify
6 * it under the terms of the GNU General Public License as published by
7 * the Free Software Foundation, version 3 of the License.
8 *
9 * EXCALIBUR is distributed in the hope that it will be useful,
10 * but WITHOUT ANY WARRANTY; without even the implied warranty of
11 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 * GNU General Public License for more details.
13 *
14 * You should have received a copy of the GNU General Public License
15 * along with EXCALIBUR in the COPYING.txt file.
16 * If not, see <http://www.gnu.org/licenses/>.
17 *
18 * Last Modification:
19 *
20 * @author nicolas.guelfi
21 * @date Mon May 06 18:10:54 CEST 2013
22 */
23
24 package lu.uni.lassy.messir.libraries.primitives{
25   Concept Model {
26     Primary Types {
27
28     primitive ptBoolean {
29       external operation is() : ptBoolean
30       external operation close() : ptBoolean
31
32       external operation msrnot() : ptBoolean
33       external operation msror(AptBoolean:ptBoolean) : ptBoolean
34       external operation msrxor(AptBoolean:ptBoolean) : ptBoolean
35       external operation msrand(AptBoolean:ptBoolean) : ptBoolean
36       external operation eq(AptBoolean:ptBoolean) : ptBoolean
37       external operation neq(AptBoolean:ptBoolean) : ptBoolean
38     }
39
40     primitive ptInteger {
41
42       operation is() : ptBoolean
43       external operation close() : ptBoolean
44
45       // Arithmetic Operations
46       external operation add(AptInteger:ptInteger) : ptInteger
47       external operation sub(AptInteger:ptInteger) : ptInteger
48       external operation mul(AptInteger:ptInteger) : ptInteger
49       external operation frac(AptInteger:ptInteger) : ptReal
50       external operation msrdiv(AptInteger:ptInteger) : ptInteger
51       external operation power(AptInteger:ptInteger) : ptInteger
52       external operation mod(AptInteger:ptInteger) : ptInteger
53
54       external operation sqrt() : ptReal
55       external operation msrabs() : ptInteger
56       external operation opp() : ptInteger
57       external operation sqr() : ptInteger
58
59       // Logical Operations
60       external operation eq(AptInteger:ptInteger) : ptBoolean
61       external operation neq(AptInteger:ptInteger) : ptBoolean
62       external operation geq(AptInteger:ptInteger) : ptBoolean
63       external operation leq(AptInteger:ptInteger) : ptBoolean

```

```

64  external operation lt(AptInteger:ptInteger) : ptBoolean
65  external operation gt(AptInteger:ptInteger) : ptBoolean
66
67  // Trigonometric Operations
68  // default is radian
69  external operation cos(): ptReal
70  external operation acos(): ptReal
71  external operation tan(): ptReal
72  external operation atan(): ptReal
73  external operation sin(): ptReal
74  external operation asin(): ptReal
75  external operation toDeg(): ptReal
76  external operation toRad(): ptReal
77
78  // Conversion Operations
79  external operation asptReal() : ptReal
80  external operation toptString() : ptString
81 }
82
83 primitive ptReal {
84
85  operation is() : ptBoolean
86  external operation close() : ptBoolean
87
88  // Arithmetic Operations
89  external operation add(AptReal:ptReal) : ptReal
90  external operation sub(AptReal:ptReal) : ptReal
91  external operation mul(AptReal:ptReal) : ptReal
92  external operation frac(AptReal:ptReal) : ptReal
93  external operation msrdiv(AptReal:ptReal) : ptInteger
94  external operation power(AptReal:ptReal) : ptReal
95
96  external operation msrround() : ptInteger
97  external operation sqrt() : ptReal
98  external operation msrabs() : ptReal
99  external operation opp() : ptReal
100 external operation sqr() : ptReal
101
102  // Logical Operations
103 external operation eq(AptReal:ptReal) : ptBoolean
104 external operation neq(AptReal:ptReal) : ptBoolean
105 external operation geq(AptReal:ptReal) : ptBoolean
106 external operation leq(AptReal:ptReal) : ptBoolean
107 external operation lt(AptReal:ptReal) : ptBoolean
108 external operation gt(AptReal:ptReal) : ptBoolean
109
110  // Trigonometric Operations
111  // default is radian
112 external operation cos(): ptReal
113 external operation acos(): ptReal
114 external operation tan(): ptReal
115 external operation atan(): ptReal
116 external operation sin(): ptReal
117 external operation asin(): ptReal
118 external operation toDeg(): ptReal
119 external operation toRad(): ptReal
120
121  // Conversion Operations
122 external operation asptInteger() : ptInteger
123 external operation toptString() : ptString
124 }
125
126 primitive ptString {
127
128  external operation is() : ptBoolean
129  external operation close() : ptBoolean
130
131  external operation length() : ptInteger
132  external operation ptStringConcat(AptString:ptString) : ptString
133  external operation subptString(

```

```

134         StartIndex:ptInteger,
135         EndIndex:ptInteger
136     ) : ptString
137     external operation toLower():ptString
138     external operation toUpper():ptString
139     external operation eq(AptString:ptString):ptBoolean
140     external operation neq(AptString:ptString):ptBoolean
141     external operation geq(AptString:ptString) : ptBoolean
142     external operation leq(AptString:ptString) : ptBoolean
143     external operation lt(AptString:ptString) : ptBoolean
144     external operation gt(AptString:ptString) : ptBoolean
145 }
146 }
147 }
148 }

```

Listing B.4: Messir Spec. file primitives.msr.

## B.5 File ./src-gen/messir-spec/library/string.msr

```

1 /*
2 * Copyright University of Luxembourg
3 *
4 * This file is part of EXCALIBUR.
5 * EXCALIBUR is free software: you can redistribute it and/or modify
6 * it under the terms of the GNU General Public License as published by
7 * the Free Software Foundation, version 3 of the License.
8 *
9 * EXCALIBUR is distributed in the hope that it will be useful,
10 * but WITHOUT ANY WARRANTY; without even the implied warranty of
11 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 * GNU General Public License for more details.
13 *
14 * You should have received a copy of the GNU General Public License
15 * along with EXCALIBUR in the COPYING.txt file.
16 * If not, see <http://www.gnu.org/licenses/>.
17 *
18 * Last Modification:
19 *
20 * @author nicolas.guelfi
21 * @date Mon May 06 18:10:54 CEST 2013
22 */
23
24 package lu.uni.lassy.messir.libraries.string{
25
26 import lu.uni.lassy.messir.libraries.primitives
27 import lu.uni.lassy.messir.libraries.math
28
29 Concept Model {
30
31     Primary Types {
32
33         datatype dtString {
34             attribute value:ptString
35
36             // Type checking Operation
37             operation is():ptBoolean
38             external operation close() : ptBoolean
39
40             operation length() : dtInteger
41             operation dtStringConcat (AdtString:dtString) : dtString
42             operation subdtString (StartIndex:dtInteger,
43                                     EndIndex:dtInteger
44                                     ) : dtString
45
46             operation toLower():dtString
47             operation toUpper():dtString
48
49             operation eq(AdtString:dtString):ptBoolean

```

```
50  operation neq(AdtString:dtString):ptBoolean
51  operation geq(AdtString:dtString) : ptBoolean
52  operation leq(AdtString:dtString) : ptBoolean
53  operation lt(AdtString:dtString) : ptBoolean
54  operation gt(AdtString:dtString) : ptBoolean
55
56  // Conversion Operations
57  operation toptString():ptString
58  }
59 }
60 }
61 }
```

Listing B.5: Messir Spec. file string.msr.





