Contents

1 Quickstart

libsbml

3 Installation

libsbml

SBML Class	C Class (typedef struct)
SBase	`SBase`

To instantiate (create) an object use either the

void Species_setCompartment (Species_t *s, const char *sname)

Sets the compartment of this Species to a copy of sname.

void Species_setInitialAmount (Species_t *s, double value)

Sets the initialAmount of this Species to value and marks the field as set.

void Species_setUnits (Species_t *s, const char *sname)

Sets the units of this Species to a copy of sname.

void Species_setBoundaryCondition (Species_t *s, int value)

Sets the boundaryCondition of this Species to value (boolean) and marks the field as set.

void Species_setCharge (Species_t *s, int value)

Sets the charge of this Species to value and marks the field as set.

In the case of strings, requiring setter methods also enables clean and simple memory semantics.

int Species_isSetName (const Species_t *s)

Return 1 if the name of this Species has been set, 0 otherwise. In SBML L1, a Species name is required and therefore should always be set. In L2, name is optional and as such may or may not be set.

4.5 Enumerations

SBML as two enumeration types, `UnitKind`

4.6 Abstract Classes

The SBML specification defines three classes that have no representation apart from subclasses that specialize (inherit from) them. In OOP parlance, these types are termed abstract. The abstract SBML classes are:

SBML Class	C Class (typedef struct)
SBase	`SBase_t`
Rule	`Rule_t`
AssignmentRule	`AssignmentRule_t`

Table 2: *Abstract SBML classes their corresponding C class.*

The conventions for abstract classes in the `libsbml` API are similar to that of other classes

```
    int     charge;
  } Species_t;
```

The e□ect is that when the library source is compiled, the first thrt1437(the)-439eld439(is)ofes_t;Spes_t;}

contains a list of rules, but a `Rule`

void SBMLDocument_printFatals (SBMLDocument_t *d, FILE *stream)

Prints all fatals encountered during the parse af this SBMLDocument to the given stream. If no fatals have occurred, that is, if `SBMLDocument_getNumFatals(d) == 0`,

```

  siari = getCurrentMillis();
  d     = readSBML(argv[1]);
  siop  = getCurrentMillis();

  m = d->model;

  printf( "File: %s\n", argv[1]);
  printf( "       model name: %s\n",  m->name );
  printf( "  unitDefinitions: %d\n",  Model_getNumUnitDefinitions(m())) );

```

`SBMLWriter.h` defines the following functions:

7 Handling of Mathematical Formulas and67(of)MathML

libsbml

Note that because the content passed to `readMathMLFromString()` is handed to an XML parser, the string given as argument must be a complete XML document. The following example illustrates the use of this function with a valid MathML input.

```
{
  MathMLDocument_t *D;
```

References

Bornstein, B. (2003). libsbml