libsbml Developer's Manual

Ben Bornstein

Contents

1 Quick Start

libsbml requires a separate XML library for low-level XML tokenizing and Unicode support. It currently supports the Xerces-C++ and Expat XML libraries on Linux, Solaris, Windows and MacOS X. You will first need to ma1(g)-3 sure one of these libraries is installed on y(g)-our system. Many Linux systems provide one or both of these libraries either as part of their standard distribution or as an optional RPM or Debian package. For more information, see http://xml.apache.org/xerces-c/ for Xerces and http://expat.sf.net

```
./configure --with-xerces=/sw
```

During the installation phase (i.e., during `make install`, discussed below), the `libsbml` installation commands will copy header files to `/usr/local/include/sbml` and (shared and static) library files to `/usr/local/lib`, by default. To specify a different installation location, use the `--prefix` argument to `configure`. For example,

```
./configure --prefix=/my/favorite/path
```

Of course, you can combine the flags to `configure`, giving both `--prefix` and `--with-expat` or `--with-xerces`

3.3.2 Memory Tracing

In addition to the unit tests, a custom memory tracing facility is available. It is disabled by

```
typedef struct
{
  SBASE_FIELDS;
  char   *id;
  char   *name;
  char   *compartment;
  union
  {
    double Amount;
    double Concentration;
  } initial;
  char   *substanceUnits;
  char   *spatialSizeUnits;
  int     hasOnlySubstanceUnits;
  int     boundaryCondition;
  int     charge;
  int     constant;
} Species_t;
```

Figure 1: *Example: the definition of SBML's Species in UML (left) and the corresponding* `Species_t` *C struct (right) in* libsbml. `SBASE_FIELDS` *is part of the OOP-liOe style used to implement objects in C; it is a macro that expands into the fields defined by SBase. The use of a union for amount and concentration reflects that these two fields are mutually exclusive in the SBML Species definition.*

To instantiate (create) an object use either the `XXX_create()` or `XXX_createWith()` constructor. To destroy (free) an object use `XXX_free()`.

4.3 Accessing Fields

int Species_isSetCharge (const Species_t *s)

Returns `1` if the `charge` of this Species has been set, `0` otherwise.

4.4 Lists

The last item in the enumeration, `UNIT_KIND_INVALID`

void SBase_setNotes (SBase_t *sb, const char *notes)

Sets the notes field of the given SBML object to a copy of notes. If object already has notes, the existing string is freed before the new one is copied.

void SBase_setAnnotation (S

not have a direct correspondence in SBML Level 1. (But, it is created by libsbml

5.2 XML Schema Validation

To have libsbml validate an SBML document against an SBML (XML) Schema when using a Schema-aware parser such as Xerces requires creating an SBMLReader object and setting the

Many software packages provide users with the ability to express formulas for such things as reaction rate expressions, and these packages' interfaces often let users type in the formulas directly as strings. libsbml provides two high-level functions for working with mathematical

Section 6.4 describes some additional points that are worth knowing about the mathematical formula handling in libsbml. For example, Level 1 formula strings and Level 2 MathML expressions can be interconverted.

to set the level to 2, the model structure at that moment is translated internally so that such things as object names are converted to `id`'s (which do not exist in Level 1).

ni

```
#include <-50stdio.h>
#include "-72)5e5#SdBdMdLdTdydpdeds#(.)-105h)-105>
```

10 Acknowledgments

Thanks to Mike Hucka for updating and editing this manual for version 2.0 of libsbml

A Lists and `ListOf_t`

AST_PLUS	AST_FUNCTION_ARCCOTH	AST_FUNCTION_POWER
AST_MINUS	AST_FUNCTION_ARCCSC	AST_FUNCTION_ROOT
AST_TIMES	AST_FUNCTION_ARCCSCH	AST_FUNCTION_SEC
AST_DIVIDE	AST_FUNCTION_	

Programs manipulating AST node structures should check the type of a given node before calling

References

Ausbrooks, R., Buswell, S., Dalmas, S., Devitt, S., Diaz, A., Hunter, R., Smith, B., Soi□er, N., Sutor, R., and Watt, S. (2001). Mathematical markup language (MathML) version 2.0 (second edition) W3C recommendation 21 October 2003.

Bornstein, B. J. (2004). LibSBML API reference manual. Available on the Internet at http://www.sbml.org/software/libsbml.

Finney, A. M. and Hucka, M. (2003). Systems biology markup language: Level 2 and beyond. *Biochemical Society Transactions*, 31:1472–1473.

[illegible]